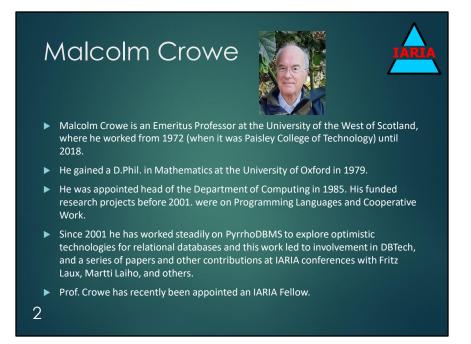


Implementing True Serializable Transactions

THE PYRRHO V7 EXPERIMENT MALCOLM CROWE, FRITZ LAUX

This tutorial is about DBMS implementation, and serializable transactions. Specifically, this experiment looks at whether the extensive use of shareable data structures with a full-featured relational DBMS implementation is feasible. Experiments reported to previous DBKDA conferences seemed to indicate that such a path was possible, but it was not clear how practical it would be.



I'm Malcolm Crowe, a retired professor from the University of the West of Scotland. I started off in Mathematics, transferred to Computing in 1979, and I have been working on database implementation since 2001.

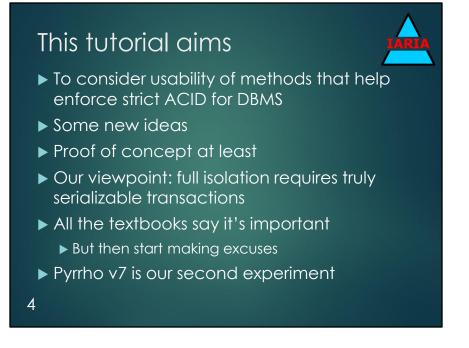
Fritz Laux





- Prof. Dr. Fritz Laux was professor (now emeritus) for Database and Information Systems at Reutlingen University from 1986 - 2015. He holds an MSc (Diplom) and PhD (Dr. rer. nat.) in Mathematics.
- His research focuses on database modeling and technology, transaction processing, data warehousing and data analytics. He has published a number of papers in peer reviewed conferences and journals on the above topics, some of them have received Best Paper Awards. He is a regular contributor and speaker at DBKDA.
- Prof. Laux is a co-founder of DBTechNet, an initiative of European universities and IT-companies to set up a transnational collaboration scheme of higher level education in Databases. Together with colleagues from 5 European countries he was conducting projects supported by the European Union on state-of-the-art teaching and hands-on labs on database technology.
- Prof. Laux received the 2012 Research Award from Reutlingen University and he is an IARIA fellow.

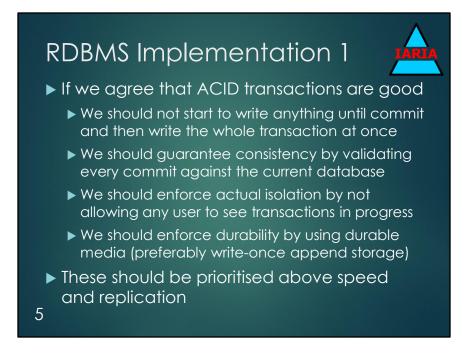
Prof. Dr Fritz Laux is a retired professor from Reutlingen University, also originally a mathematician, who was appointed Professor of Database and Information Systems in 1986. His research papers focus on database modelling and technology, transaction processing, data warehousing and data analytics, and he is a regular contributor to DBKDA.



In this tutorial we will consider some methods that help to enforce strict atomicity, consistency, isolation and durability for database management systems. There are some new ideas and the Pyrrho experiment brings proof of concept.

Our starting point is that full isolation requires truly serializable transactions.

All database textbooks begin by saying how important serializability and isolation are, but very quickly settle for something much less. This is really our second experiment on using shareability.



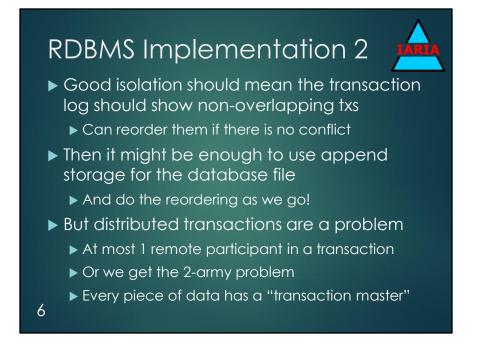
If we agree that ACID transactions are good, then:

First, for atomicity and durability we should not write anything durable until (a) we are sure we wish to commit and (b) we are ready to write the whole transaction.

Second: before we write anything durable, we should validate our commit against the current database.

Third, for isolation, we should not allow any user to see transactions that have not yet been committed.

Fourth, for durability, we should use durable media – preferably write-once append storage.



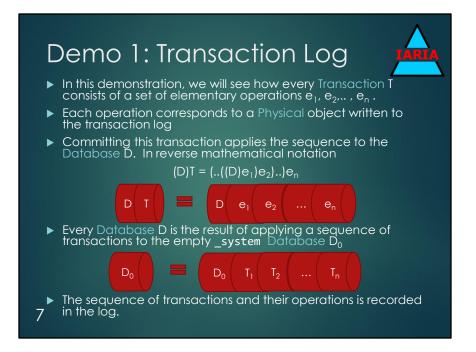
From the last slide, it seems clear that a database should record its durable transactions in a non-overlapping manner.

If transactions in progress overlap in time, they cannot both commit if they conflict: and if they don't conflict, it does not matter which one is recorded first. The simplest order for writing is that of the transaction commit.

If we are writing some changes that we prepared earlier, the validation step must ensure that it depends on nothing that has changed in the meantime, so that our change can seem to date from the time it was committed rather than first considered. Effectively we need to reorder overlapping transactions as we commit them.

These few rules guarantee actual serialization of transactions for a single transaction log (sometimes called a single transaction master). It obviously does not matter where the transactions are coming from.

But if a transaction is trying to commit changes to more than one transaction log, things are very difficult. If messages between autonomous actors can get lost, then inconsistencies are inevitable. One-way dependency can be implemented with a single remote participant.



Our first demonstration is about the transaction log. The transaction log defines the contents of the database.

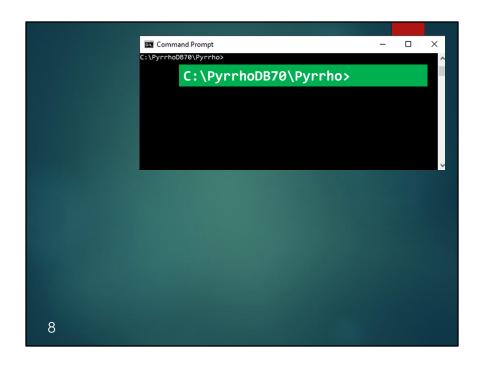
If we think of a transaction commit as comprising a set of elementary operations e, then the transaction log is best implemented as a serialization of these events to the append storage. We can think of these serialized packets as objects in the physical database. In an objectoriented programming language, we naturally have a class of such objects, and we call this class Physical.

So, at the commit point of a transaction, we have a list of these objects, and the commit has two effects (a)

appending them to the storage, (b) modifying the database so that other users can see.

We can think of each of these elementary operations as a transformation on the database, to be applied in the order they are written to the database (so the ordering within each transaction is preserved). And the transaction itself is the resulting transformation of the database.

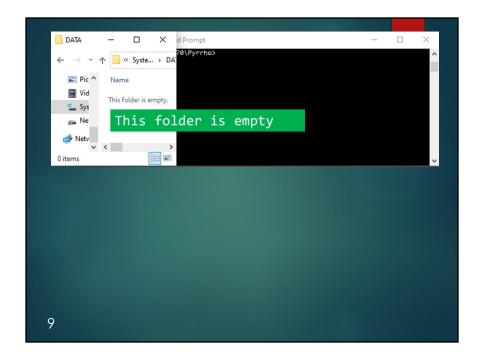
Any state of the database is thus the result of the entire sequence of committed transactions, starting from a known initial database state corresponding to an empty database.



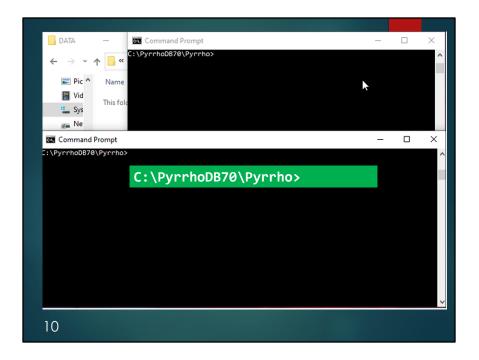
Let's start with a command window, set to whatever folder the distribution is in. Don't worry if yours is different.

The Pyrrho distribution folder contains the server and command line processor executable files, and the PyrrhoLink dll. (You can copy these files to anywhere convenient.)

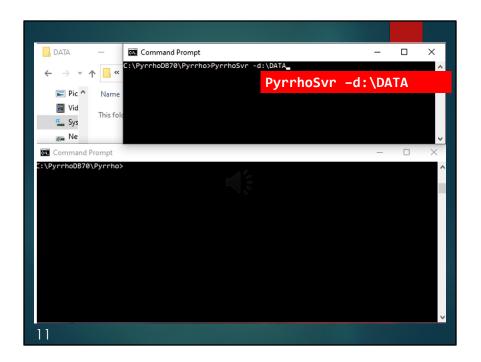
This command window will be for the server.



Let us agree on a folder for the database files. Create a new folder \DATA. To save space, as here, you can overlap the windows a bit.

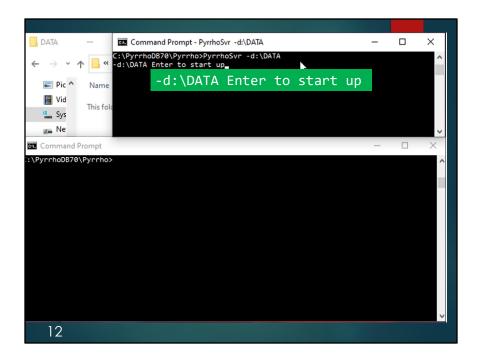


Before we start the server, let's add a command window for the client, also with the same folder. We will see it is better to make it wider than normal.

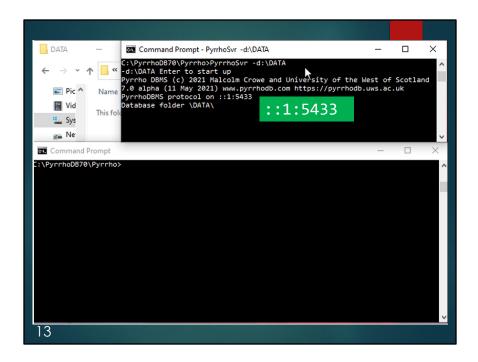


The Pyrrho server is called PyrrhoSvr, and it runs in an ordinary account with no special privileges. You can copy it to anywhere convenient.

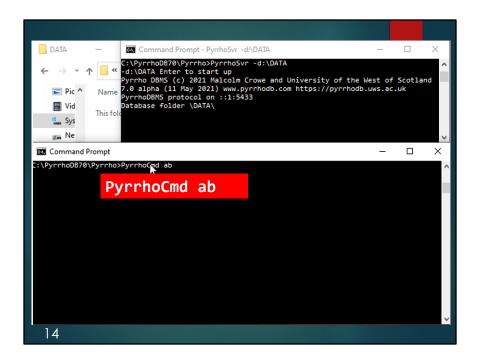
It is easiest just to use it in a command window. When we specify the server, we can provide a folder for it to store database files in.



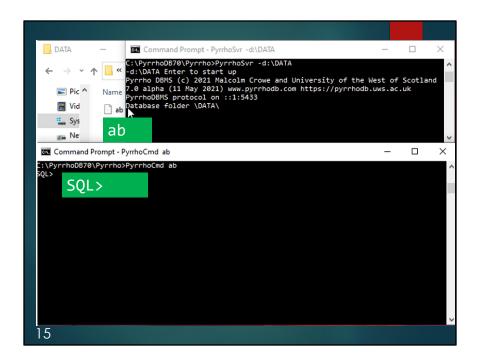
When prompted, we verify that the options specified are the ones we want, and click Enter.



You then need to leave the command window open (but of course you can minimise it). It can be useful for diagnostic information if something goes wrong. On startup, Pyrrho checks that it can access the specified folder, but places nothing in it. Databases are created by users, and the transaction log will be stored by the server on disk, usually in this default folder. It also announces its TCP/IP request port.

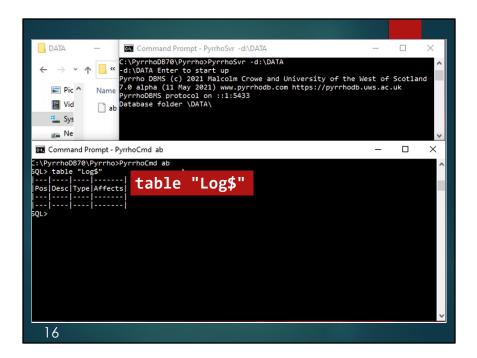


The simplest way to get Pyrrho to do work for you is to connect to it using the command line processor PyrrhoCmd, and when you do that you can specify the database you are connecting to. It can be a new database, as here.

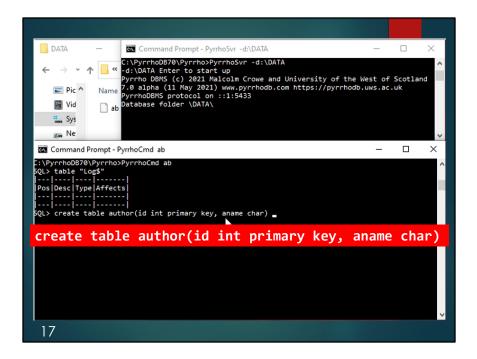


And when you give the command, the server immediately creates an empty database in the database folder.

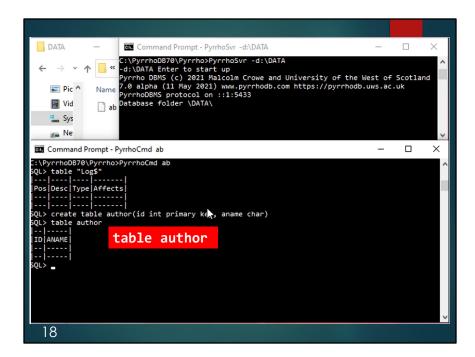
This is a transaction log, and at the moment it contains just 5 bytes, and its opened exclusively by the server, so that we can't look at it unless the server is stopped.



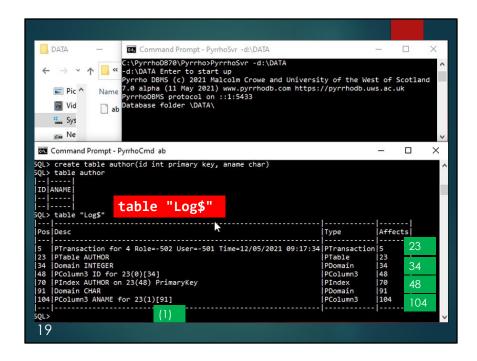
We can examine the contents of the transaction log with the table "Log\$" statement. Log\$ is one of many system tables for inspecting server internals from SQL. It is empty at the moment.



Let's make a base table in the database ab, by giving a CREATE TABLE command, and this is the normal syntax in the SQL standard, but Pyrrho has it own ideas about primitive types. So CHAR, for example, is an unbounded character string, and INT is actually a "bigint" – it's up to 2000 bits

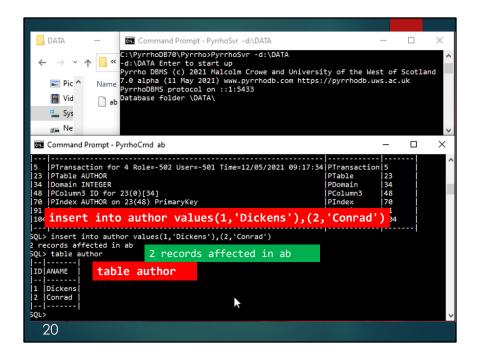


So, we have created the table, but let's look at it: of course it will be empty.

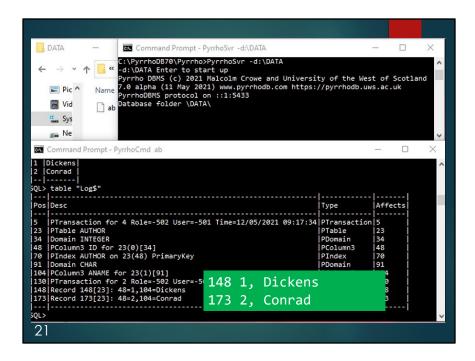


Let's examine how this has been represented in the transaction log. We can see that there are 7 objects in the transaction log, all wrapped in a single transaction. The AUTHOR table is mentioned, and the domains of the two columns, and the two column names, and the primary key as defined in the create table request, is defined there along with its key.

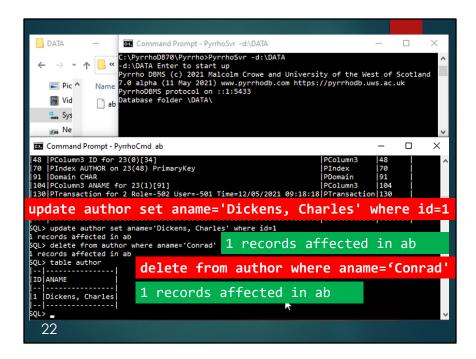
You will notice that the file positions that are here (these are actual byte positions in the file) are used as unique identifiers for objects in the database. Pyrrho uses them throughout: names can be changed but the file position of the definition can't. So here the table AUTHOR is referred to as 23, the domain INTEGER is 34, ID is 48, and that becomes the primary key, and so on. The numbers in brackets give the ordering of the columns in the table. The log shows the identity of the user who made the changes, and the role they were using, but in this database, no users have been defined yet, so these are system defaults.



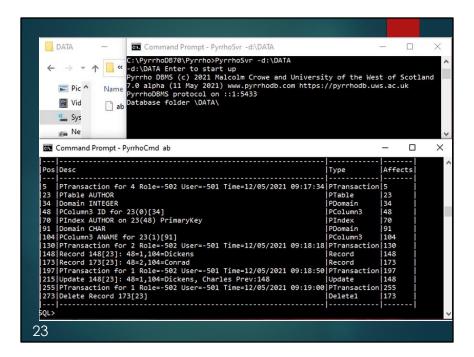
Let's create a couple of authors in this AUTHOR table. We can do this in a single INSERT statement. And then we display the table.



In the transaction log we see that the auto-committed transaction has both records.



We can update, and delete.



Finally let's see the update and delete in the Log.

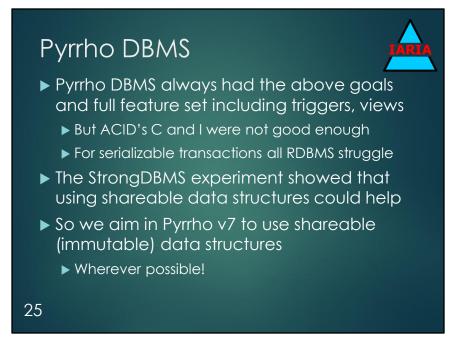
This completes the demonstration showing the use of the transaction log and the transaction markers.



If we agree on a globalization strategy, then the DBMS should be neutral, and not specific to a particular machine, platform, or locale. A database created on one machine, platform or locale should be usable on another.

The DBMS should not impose arbitrary size limits on strings, number of columns etc. Any that are imposed should be huge.

If we agree that security is important, then we should use operating system authentication and no other options. Users should not be simply allowed to say who they are.



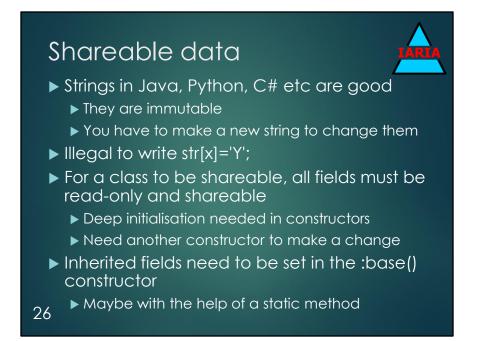
Pyrrho DBMS has always had these goals, and a full feature set including stored procedures, structured types, triggers, and views.

But it turned out that its consistency and isolation in its implementation were not good enough, and it was easily outperformed by StrongDBMS, a much simpler system. In an artificial test with high concurrency and serializable transactions, we found that all DBMS were outperformed by StrongDBMS.

So, in Pyrrho v7, the aim was to re-implement Pyrrho using a lesson learned from StrongDBMS, that in

situations of high transaction concurrency it was best to use shareable, immutable data structures, for as many internal structures as possible.

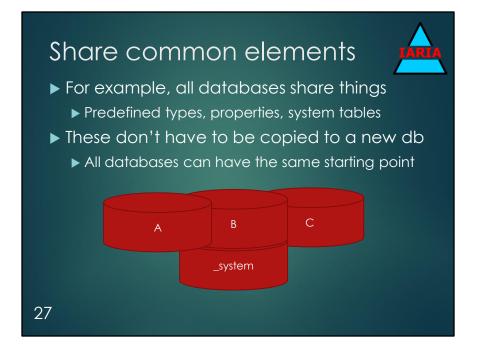
The implementation has been progressing steadily, it is still in at the alpha stage, but anyone can see the progress that has been made, as the source code is on github.



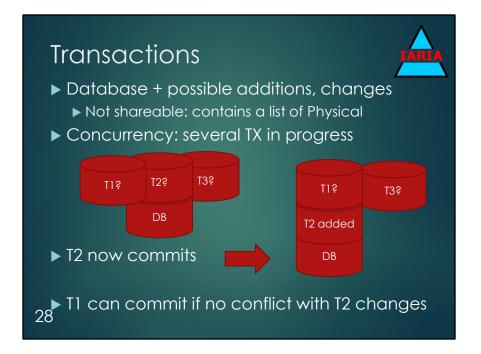
Let's quickly review what is meant by shareable data. Many programming languages currently have shareable implementations of strings. Specifically, strings in Java, Python and C# are immutable: if you make a change, you get a new string, and anyone who had a copy of the previous version sees no change.

In these systems, it is illegal to modify a string by assigning to a character position instead you need to use a library function. The addition operator can be used in these languages to create a sum of strings. This is basically the model for shareable data structures. For a class to be shareable, all fields must be read-only and shareable. Constructors therefore need to perform deep initialisation, and any change to an existing structure needs another constructor.

Inherited fields need to be initialised in the base (or super) constructor.



This is useful for databases because databases share so many things: predefined types, properties, system tables. So as mentioned on a previous slide, all databases have the same starting state.



Even more importantly all transactions can start with the current state of the database, without cloning or copying any internal structures. When a transaction starts, it starts with the shared database state: as it adds physicals, it transforms. Different transactions will in general start from different states of the shared database. In the picture on the slide, we know what the database state is. We don't know if any of the transactions fit on top anymore. In particular, after T2 commits, T1 and/or T3 may no longer be able to commit.

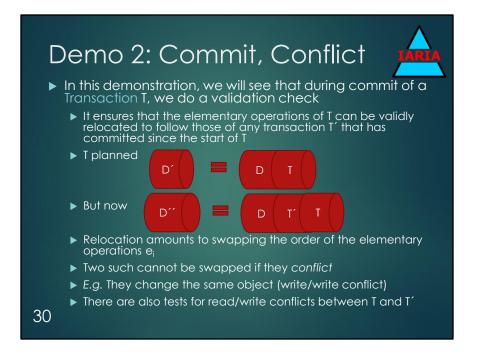
If T1 was able to commit before, then it will still be able

to commit provided it has no conflict with T2's changes.



The details of what constitutes a conflicting transaction are debatable. Most experts would agree with some version of the rules on this slide. The first rule is sound enough, although the condition on reading is very important: we would need to include things like aggregation in our definition of reading. The first rule is also very easy to implement, especially if tables are shareable structures, as a simple 64-bit comparison is sufficient!

For the other rules, we would need to be very clear on what a specified row is, and the non-existence of a row might only be determined by reading the whole table. Such debate is beyond the scope of today's tutorial.



In the second demonstration, we look in detail at the Commit() method for a transaction, and the detection of conflicts.

At the start of Transaction Commit, there is validation check, to ensure that the transaction still fits on the current shared state of the database, that is, that we have no conflict with transaction that committed since our transaction started.

If that is the case, we can relocate all our proposed changes to come after the committed transactions.

The tests for write-write conflicts involve comparing our list of physicals with those of the other transactions.

For checking read-write conflicts, we collect "read constraints" when we are making Cursors.

PyrrhoDB70\Pyrrho>	C:\PyrrhobB70\Pyrrhob	← →	~ 🛧 🧧 « Syste
C:\PyrrhoD	B70\Pyrrho>		Name This folder is empty.
Window Help		PyrrhoSvr	- 🗆 X
0 • 0 8 • 🖕 🗎	📲 🦻 + 🖓 + 🗼 Stat - 🗮 🗃 👳		🖄 Live Share 🔗
PyrrhoSvr* + ×		- ¢	Solution Explorer • 9
Application Build	Configuration: Active (Debug) V Platform: Active (Any CPU) V		Search Solution
Build			
Application	Start action		
Build Build Events	Start action		
Build Build Events Debug*	Start action		 ✓ □ Solution 'Pyrr ▲ ✓ □ PyrrhoSv ▷ □ Refere ▷ □ Comrr ▷ □ Level1
Build Build Events Debug* Resources	Start action		
Build Build Events Debug* Resources Services	Start action Start project Start external program:		✓ [G] Solution 'Pyrr ▲ ✓ (🖙) PyrrhoSv ▷ ↓ ▷ ↓ ▷ ↓ ▷ ↓ ▷ ↓ ▷ ↓ ▷ ↓ ▷ ↓ ▷ ↓ ▷ ↓ ▷ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
Build Build Events Debug* Resources Services Settings	Start action Start action Start project Start sternal program: Start options		Golution 'Pyrr Comr Comr Comr Comr Comr Comr Comr Court Court
Build Build Events Debug" Resources Services Settings Reference Paths	Start action		✓ □ Solution 'Pyn ▲ ✓ ○ PyrthoSv ▷ → ▷ Comm ▷ Comm ▷ Level1 ▷ Level2 ▷ Level3 ▷ Proper □ Proper □ Proper □ Proper □ Proper □ Proper
Build Events Debug* Resources Services Settings Reference Paths Signing	Start action Start action Start project Start sternal program: Start options		
Build Build Events Debug* Resources Services Setvings Reference Paths Signing Security	Start action Start action Start project Start sternal program: Start options		○ ○ → ■ Search Solution P ● > ● > > > > > <td< td=""></td<>

The slide shows two client command windows, the database folder, and Visual Studio. We will get the Visual Studio debugger to run the server for us, as this will enable us to look at the validation step of Transaction.Commit().

Visual Studio has opened the solution in Pyrrho's src\Shared folder, and I have set the database folder to \DATA for convenience, using the Debug properties of the PyrrhoSvr project. In the command window that the start-up creates, I clicked the Enter key to start the server as in demo 1. I have minimised the server window to get it out of the way.

We confirm that the database folder I have nominated is empty.

		Jeck			Syst
Pyr	rhoCmd t10	P	yrrhoCmo	1 t10	
SQL>		S	QL>		t10
File Edit Vier Window Help O • O ☆ • 🍰	w Git Project Build Debug		nsions Search P	PyrrhoSv	T → □
Application	Configuration: Active (Debug)	Platform: Active	(Any CPU)		
Build	Coningulation. Active (Debug,	Pladofin. Active	(Any CPO)		Search Solution Explor
					Solution 'Pyrrho!
Build Events	Start action			~	
Build Events Debug*				^	✓ ✓ ✓ ✓ ✓ ✓ ✓
	 Start project 			^	PyrrhoSvr PrrhoSvr P Reference P Common
Debug*	 Start project Start external program: 				PyrrhoSvr P Reference Common Level1
Debug* Resources	 Start project 				PyrrhoSvr PrrhoSvr P Reference P Common
Debug* Resources Services	 Start project Start external program: 				C PyrrhoSvr P *** Reference Common D *** Level1 D *** Level2 D *** Level3 D *** Level4
Debug* Resources Services Settings	 Start project Start external program: Start browser with URL: 	-d\DATA			C: PyrrhoSvr ParhoSvr ParhoSvr Common Level1 Level2 Level4 Properties
Debug* Resources Services Settings Reference Paths	Start project Start external program: Start browser with URL: Start options	-d:\DATA			
Debug* Resources Services Settings Reference Paths Signing	Start project Start external program: Start browser with URL: Start options	-d:\DATA			VCE PyrrhoSvr VCE PyrrhoSvr VCE Properties VCE Verla VCE

The command we want to run in both windows is the same: PyrrhoCmd t10. Once we do one of them, the DBMS immediately creates the database as we have seen in demo 1 - I have hidden the folder window.

1 records affected in t1	Int primary keys char ext(4).(ife, the Universit) ext(2).(ife) the Universit) ext(2).(ife) the heypert) (ife) the intervention of the extension of the exte	→ × ↑ Syste >
File Edit Vie	nsert into RDC values(1,'Life,the Ur dt Wojet build Debug let Analyze loois Extensions Cruck of Damage nsert into RDC values(2,'Weeks in th	
Application Build Build Events Debug" Resources Services Settings	able RDC Configuration: Active (Debug) V Platform: Active (Any CPU) V Start action © Start project Start external program: Start browser with URL:	Solution Explorer
Reference Paths Signing Security Publish Code Analysis	Start options Command line arguments: -dt\DATA Working directory:	▶ Level4 ▶ Properties ■ App.config ▶ < © Pocument.cs ▶ < ∞ ■ Document.cs ▶ < ∞ ■ Serverico ▶ < ∞ ■ Serverico ▶ < ∞ ■ Start.cs
□ Ready 33	↑ 2 🖋 99* 🚸 Shared	Ϋ́master 🔺 🦣 🛛

In the blue window we do some work on the database. We create a table and insert a couple of records. We can display the table and see just what we have created there.

Command Prompt - Pyr SQL: create table ROC(A in SQL: sinsert into NOC value into NOC value 10(L: insert into NOC value 1 (L: insert into NOC value 1 (L: into NOC val	t primary key,B chan) (42,'Life, the Universe') (52,'Weeks in the yean')	<pre>C:Vyyrh00570(byrh0c)Syyrh0C' SQL sable R0C 1 s s s2 Ueeks in the year s2 Weeks in the year s2 Weeks in the year s2 Life, the UNLyears s3 Life, the UNLyears s4 Life, the UNLY s4 Life, the U</pre>	table R	× DATA	- □ × * ↑ ■ « Syste → 1 t10
File Edit View Window Help	Git Project Build Debug	Test Analyze Tools Exte	nsions Search P	PyrrhoSvr	- 🗆 ×
🕴 O - O 🔞 - 🖕 🖬	📲 🛛 🕫 🗣 🕨 🕨 Continue	- ≥ ≥ = ð →	* @ * 🗗 - 🚽		🖄 Live Share 🛛 🕅
PyrrhoSvr* ⊕ ×				- ¢ s	olution Explorer 💌 👎 🗙
Application Build	Configuration: Active (Debug)	 Platform: Active 	: (Any CPU) V	s	ෙ ව 🔐 🖓 🐻 - " earch Solution Explor 🔑 -
Build Events	Start action				Solution 'PyrrhoSvr' A
Debug*	 Start project 			- 11	References
Resources	 Start external program: 				Common
Services Settings	O Start browser with URL:			51	 Level2 Level3
Reference Paths	Start options				Eevel4 Properties
Signing	Command line arguments:	-d:\DATA		^	a Properties
Security					↓ ✓ C# Document.cs ↓ ✓ C# HttpService.c
Publish				~	C* Httpservice.c C* Resx.cs
Code Analysis	Working directopy			>	a ⊇ Server.ico ▷ ✓ C# Start.cs ▼
Ready			nt 2 💉 99*	🚯 Shared	🖓 master 🔺 🦣
34					

That has all been committed, because we are running in auto-commit mode, so the green window will pick it up. It is also running in auto-commit mode, so a new command starts a new transaction, and it will check the current state of the database.

This completes the set up for this demo. We will repeat this part of the demo later.

Command Prompt - P			X 🔒 DA	ATA — 🗆 🗡
records affected in t10		C:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10 SQL> table RDC	^ <	→ × ↑ 📙 « Syste
records affected in t10	es(52,'Weeks in the year')	A B	_	
QL> table RDC		42 Life, the Universe		 Name
A B	k	52 Weeks in the year 		🗋 t10
42 Life, the Universe 52 Weeks in the year		SQL>		
 SQL> begin transaction SQL-T>				
File Edit View	🗰 begin tr	ansaction tensions Search	PyrrhoS	vr – 🗆 >
Window Help	008211 01			
🕴 😋 🔹 💿 🛛 📸 🖓		<mark>; •</mark> • • • • • • • • • • • • • • • • • •	;	🖄 Live Share 🛛 🕅
PyrrhoSvr* ⇔ ×	SQL-T>		- ¢	Solution Explorer 💌 👎
· ·			_	0 0 G 🖓 🗐 To -
Application		V Platform: Active (Any CPU)		
0.31	Configuration: Active (Debug)	 Platform: Active (Any CPU) 	<u>^</u>	Search Solution Explor
Build		Platform: Active (Any CPO)	<u> </u>	
Build Events	Start action	Viatrorm: Active (Any CPU)	^	
Build Events Debug*		Platform: Active (Any CPU)	^	Solution 'PyrrhoSvr' C:: PyrrhoSvr C:: PyrrhoSvr S:: References
Build Events Debug* Resources	Start action	Platform: Active (Any CPU)	^	Solution 'PyrrhoSvr' C PyrrhoSvr References Common
Build Events Debug* Resources Services	Start action © Start project O Start external program:	Plattorm: Active (Any CPU)		Solution 'PyrrhoSvr' C:: PyrrhoSvr C:: PyrrhoSvr S:: References
Build Events Debug* Resources	Start action	Plattorm: Active (Any CPU)		 ✓ Ig3 Solution 'PyrrhoSvr' ✓ (🖙 PyrrhoSvr → 🐨 References → Common → Level1 → Level2 → Level3
Build Events Debug* Resources Services	Start action © Start project O Start external program:	Plattorm: Active (Any CPU)		✓ [4] Solution 'PyrrhoSvr' ✓ ∞ PyrrhoSvr ▷ Common ▷ Common ▷ Level1 ▷ Level2 ▷ Level4
Build Events Debug* Resources Services Settings	Start action ③ Start project ③ Start external program: ③ Start browser with URL:	-dt/DATA		 ✓ Ig3 Solution 'PyrrhoSvr' ✓ (🖙 PyrrhoSvr → 🐨 References → Common → Level1 → Level2 → Level3
Build Events Debug* Resources Services Settings Reference Paths	Start action ③ Start project ③ Start external program: ③ Start browser with URL: Start options			✓ Im Solution 'PyrthoSvr' ✓ Im PyrthoSvr ✓ Im PyrthoSvr ✓ Im References ✓ Im Common ✓ Im Level ✓ Im Level2 ✓ Im Level3 ✓ Im Level4 ✓ Im Properties ✓ Im App.config ✓ C Documentces
Build Events Debug* Resources Services Settings Reference Paths Signing	Start action ③ Start project ③ Start external program: ③ Start browser with URL: Start options			
Build Events Debug* Resources Services Settings Reference Paths Signing Security	Start action ③ Start project ③ Start external program: ③ Start browser with URL: Start options Command line arguments:			✓ Im Solution 'PyrthoSvr' ✓ Im PyrthoSvr ✓ Im PyrthoSvr ✓ Im References ✓ Im Common ✓ Im Level ✓ Im Level2 ✓ Im Level3 ✓ Im Level4 ✓ Im Properties ✓ Im App.config ✓ Cm Documentces
Build Events Debug" Resources Services Setrings Reference Paths Signing Security Publish	Start action ③ Start project ③ Start external program: ③ Start browser with URL: Start options		v	
Build Events Debug" Resources Services Setrings Reference Paths Signing Security Publish	Start action ③ Start project ③ Start project ③ Start external program: ③ Start browser with URL: Start options Command line arguments: Working directors		v >	>

Now we start an explicit transaction in the blue window. If we just relied on auto-commit mode it is very difficult to synchronise an overlap of transactions. For a demo, it works very well to use explicit transactions.

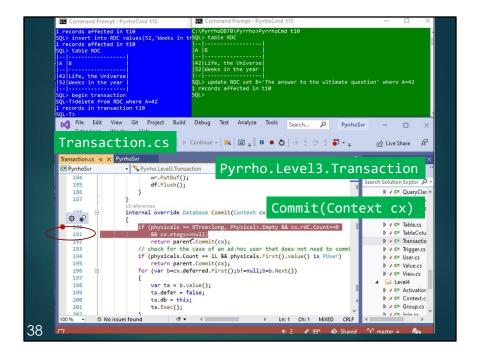
Notice that in an explicit transaction the prompt changes to SQL-T>. As long as the transaction is running, we will get these prompts. When the transaction is over, either because of commit, or rollback, or because we have done something wrong, it will go back to SQL>.

SQL Command Prompt - Py SQL> insert into ROC value 1 records affected in t10 SQL> table ROC 1	s(52, Weeks in the year") c(1/SynthobByOL	X DATA - X ← → • ↑ • • Syste → Name 1 t10
SQL-TS Window Help O - O R - C - C - C - C - C - C - C - C - C -	³⁰ delete from RDC where GR Project Bu 1 records in transaction t10	A=42
PyrrhoSvr* ≄ × Application Build	SQL-T> Configuration: Active (Debug) VPlatform: Active (Any CPU)	Solution Explorer ▼ ↓ × ○ ○ ☆ ↓ □ ○ · □ Search Solution Explor ♪ ~
Build Events	Start action	✓ kq Solution 'PyrrhoSvr' ▲ ▲ ✓ C℡ PyrrhoSvr
Debug* Resources Services	Start project Start external program:	References Common Level1 Level2
Settings Reference Paths	Start options	▷ ► Level3 ▷ ► Level4 ▷ ► Properties
Signing Security Publish	Command line arguments: -d\DATA	 App.config ✓ C[™] Document.cs ✓ C[™] HttpService.c
Publish Code Analysis	Working directory	
37 Ready	<u>↑</u> 2 Ø9	9* 🚯 Shared 🖓 master 🔺 🦣

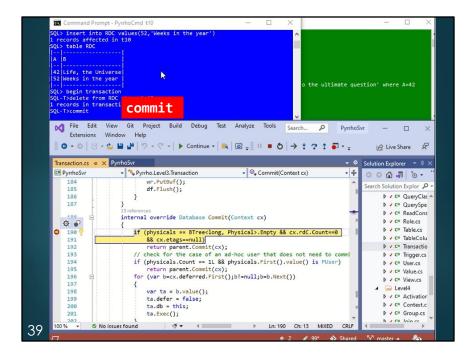
We will request conflicting changes to table RDC in these two clients. In the blue window, let's "delete from RDC where A=42", to delete the first row. That's in a transaction, so nothing has been written to disk yet.

the ultimate ques	
rds affecte	ed in t10 @ Live Share \$
	👻 🕸 Solution Explorer 💌 👎
J) ~	C C A A A C -
	 Solution 'PyrrhoSv Image: Construction of the second seco
	References
	Common
	Eevel1
	Level3
	 Level4 Properties
^	a Properties
	♦ ✓ C [#] Document.
~	▷ ✓ C# HttpService ▷ ✓ C# Resx.cs
>	

In the green window, we are still in auto-commit mode and we are going to make an update to the same row, which will be committed straightaway to disk. This should make the blue window unable to commit its transaction because of the conflict.



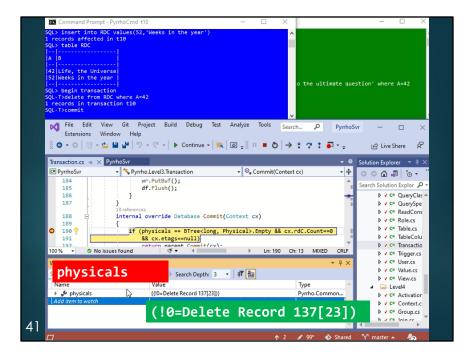
In order to see what happens, let us set a breakpoint in the debugger, at the start of the Transaction.Commit() method.



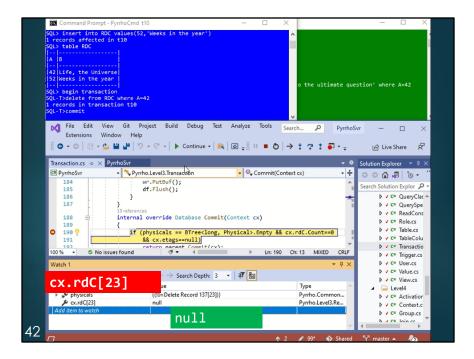
Then when we issue the commit command in the blue window, we hit the breakpoint.

	ommand Prompt - PyrrhoCmd t10				
1 reco	nsert into RDC values(52, We nds affected in t10 able RDC	eks in the year		Â	
A B					
	fe, the Universe eks in the year			o the ultimate	question' where A=42
SQL-T> 1 reco	egin transaction delete from RDC where A=42 rds in transaction t10 commit			,	
M	File Edit View Git Project		Test Analyze Tools		rhoSvr — 🗆 🗡
	Windows	+	Breakpoints Exception Settings	Ctrl+Alt+B	R Live Share
	Continue Break All	F5 Ctrl+Alt+Break	 Exception Settings Output 	Ctrl+Alt+E	Pre cive share X
	Stop Debugging	Shift+F5	(o) XAML Binding Failure	c	🍄 Solution Explorer 🔻 🕂
	Detach All	Shinters	Show Diagnostic Tools		• 0 0 û 🗐 'o •
	Terminate All		Set GPU Threads		Search Solution Explor 🔎
5	Restart	Ctrl+Shift+F5	Tasks	Ctrl+Shift+D. K	♦ 🗸 C# QueryClas
V.E.	Apply Code Changes	Alt+F10	Parallel Stacks	Ctrl+Shift+D, K	▷ ✓ C* QuerySpe ▷ ✓ C* ReadCons
	Performance Profiler	Alt+F2	Parallel Watch	Cur+snint+D, s	▶ ✓ C* Role.cs
🖾 🕽	Relaunch Performance Profiler		Walch		↓ C ⁿ Table.cs
000	Attach to Process 🐼 Watch 1	Ctrl+Alt+W, 1	Autos	Ctrl+Alt+V. A	♦ ✓ C [#] TableColu ♦ ✓ C [#] Transactio
	Other Debug Tare	Ctrl+Alt+W, 2 Ctrl+Alt+W, 3	[%] Locals	Ctrl+Alt+V, L	↓ C* Trigger.cs
*	Step Into	Ctrl+Alt+W, 3 Ctrl+Alt+W, 4	D Immediate	Ctrl+Alt+I	♦ 🗸 C# User.cs
?	Step Over		Live Visual Tree		♦ ✓ C# Value.cs ♦ ✓ C# View.cs
1	Step Out	Shift+F11	Live Property Explorer		∠ Level4
69	QuickWatch	Shift+F9	Call Stack	Ctrl+Alt+C	♦ ✓ C# Activation
	Toggle Breakpoint	F9	Call Stack	Ctrl+Alt+H	
	New Breakpoint	+	Modules	Ctrl+Alt+U	► P ✓ C# Group.cs
1 8	Delete All Breakpoints	Ctrl+Shift+F9	Processes	Ctrl+Alt+Z	
	Disable All Breakpoints				ed 🖓 master 🔺 🧞

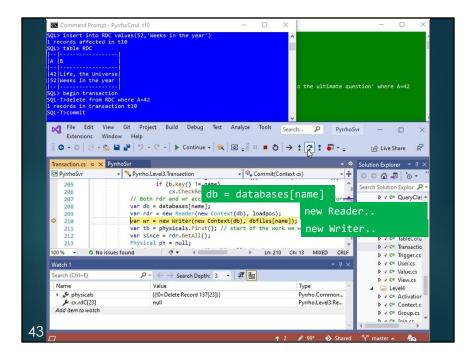
Let us open the Watch window in the debugger, so we can examine details.



Let's look at the physicals. **physicals** is the list of Physical records that the Transaction wishes to commit, and it's just the single Physical record to delete row 137 in the database, "Life, the Universe".

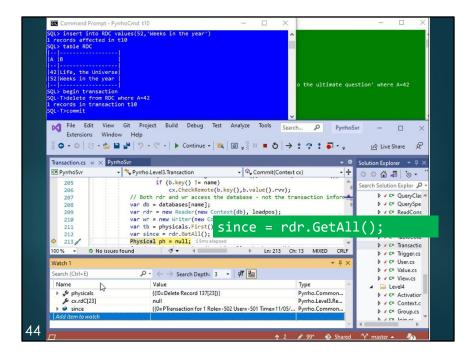


We will also look at cx.rdC[23], which is to do with the ReadConstraints for this particular Transaction. In this case there aren't any, as this transaction hasn't read anything. (In the next experiment we will see it is the other way round.) Step Over a few times to get to line 208.

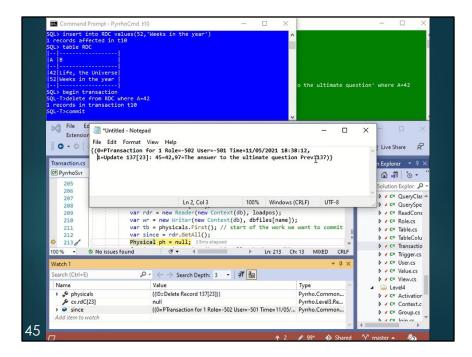


We Step Over a few lines to line 208. The validation step will use the up-to-date copy of the database, as it was left by the green window. We also open a Reader and a Writer: a Reader to look at this database, specifically at the records that have been committed since the start of our Transaction; and a Writer, where we will prepare the records that we are going to add to the database if our validation succeeds.

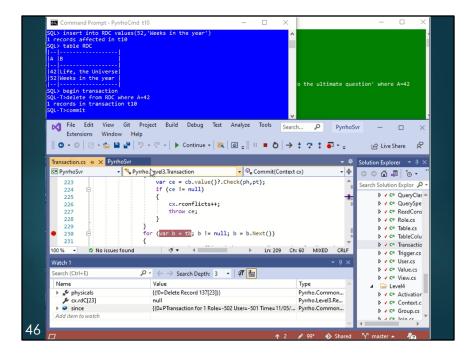
These are not shareable and are subject to locking protocols. They also work on the same FileStream. (Transaction Commit() repeats the validation step after locking the FilleStream.)



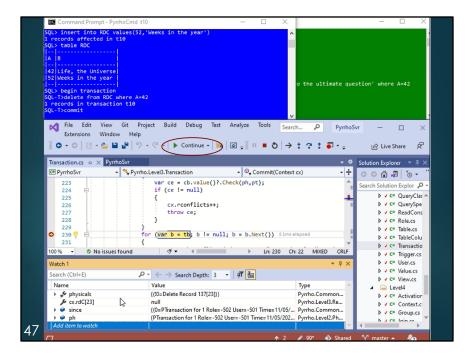
Step Over to line 213, just after a line saying "since=rdr.GetAll();" This gets the records that have been committed to the database since the start of our transaction. Use the Watch window to examine **since**. Right-click, Copy Value, and paste it into a Notepad.



We can see we have got the transaction marker and the update that the green window has made.

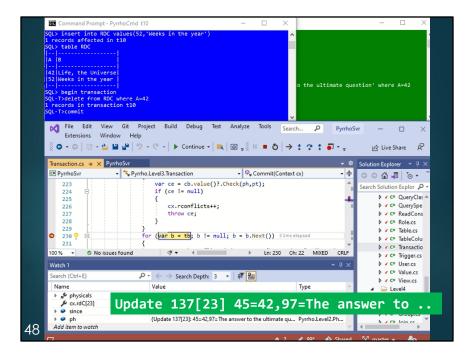


Set a breakpoint at line 230. Click Continue.

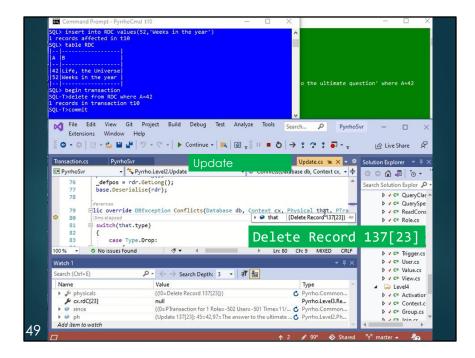


The first time we hit this particular line, ph is the transaction record, which is not very interesting.

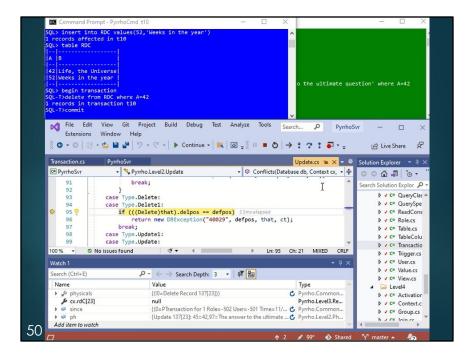
Click Continue



The second time, ph is the Update, and we call the Conflicts() method. Step Into (Visual Studio breaks in evaluation of parameters, so step out, and then in again).



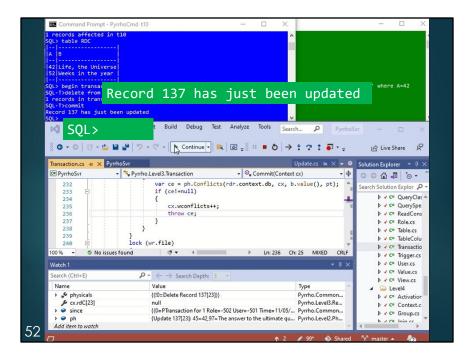
In the Conflicts method, we see that **that** is our Delete, and **this** is the Update. We are looking at the Update that was done by the green window, and comparing it with our Delete record.



Step Over: we reach line 95. For your delete to conflict with an update, it just means that the position you are trying to delete matches the position that has been updated. We can see this is the case, and we will get an Exception. Just click Continue.

Command Prompt - PyrrhoC records affected in \$10	imd t10 — 🗆	_	Prompt - PyrrhoCm			× D.	ATA —	
L> table RDC		SQL> table RDC				^ ←	→ * ↑	« Syste
В		AB					 Name 	
2 Life, the Universe 2 Weeks in the year		42 Life, the U	niverse					
L> begin transaction		jj		to the	ultimité é	105	🗋 t10	
L-T>delete from RDC where A= records in transaction t10 L-T>commit	42	tion' where A=4: 1 records affect SOL>						
🕅 File Edit View Git	t Project Build D	ebug Test Analyze	Tools Extensions	Sea	rch 🔎	PyrrhoS	Svr —	
Window Help								
0-0 13-00 14	・	ntinue 🗸 💼 🙀 🖬 🗸	• ð → : ·	2 1	• • •		🖻 Live S	ihare 🖇
Transaction.cs PyrrhoSv	/r* Continue				Update.cs	≝ × - ≎	Solution Explo	rer 🝷 🕂
CIII PyrrhoSvr •	% Pyrrho.Level2.Update	1	 O Conflicts(Dat 	abase d	b, Context	x, Phys → 💠	000	- o -
92	}					*	Search Solutio	
93	case Type.Delet case Type.Delet							QuerySpi
95 Q		e)that).delpos == de	fpos) ≤1ms elapsed					ReadCon
96		new DBException("400	29", defpos, tha	t, ct)	;			Role.cs
97	break;					1	Þ 🗸 C=	Table.cs
98 100 % - Ø No issues four	case Type.Updat	e1:	h In	95 (Ch: 21 MI	ED CRLF	Þ 🗸 C4	TableCol
			- Lii	33 (.n. 21 - Wi		Þ 🗸 C	Transacti
Watch 1	·					- 4 ×		Trigger.cs
Search (Ctrl+E)	$P \cdot \leftarrow \rightarrow$ Search	h Depth: 3 🔹 🖣 👪						User.cs
Name	Value				Туре			Value.cs
cx.physicals	{(!0=Delete	Record 137[23])}			Pyrrho.Cor	nmon.BT	Þ 🗎 Le	View.cs
	null				Pyrrho.Lev			operties
👂 🤗 since		action for 1 Role=-502 User						operties
▶ 🧼 ph	{Update 13	7[23]: 45=42,97=The answe	r to the ultimate ques	či 🖒	Pyrrho.Lev	el2.Physi		ocument.c
Add item to watch								ttpService.
						Y	A	l ≯

In this case we find there is a conflict, so let's just click Continue



We see that we get a complaint back in the blue window that record 137 has just been updated. The transaction has been rolled back, as we see from the prompt.

That completes the first experiment that we want to do in this demonstration.

l records affected i	n t10 values(52,'Weeks in	he L A B http://www.second.com/ 42[Life, the Universe] 52[Weeks in the year	
A B 42 Life, the Univer 52 Weeks in the yea 		sous	
File Edit View Extensions Win	dow Help	Id Debug Test Analyze Tools <u>Search</u> P PymhoSvr — □ ▷ Continue - R I :	े रू
PyrrhoSvr ⊕ ×		👻 🌣 Solution Explorer	
Application	Configuration: Act		
Build	-	Search Solution Exp	
Build Build Events	Platform: Active (A	Any CPU) V Search Solution Exp	eryClas
	-	Search Solution Exp	eryClas erySpe
Build Events	Platform: Active () Start action	Any CPU) Any CPU Any	eryClas erySpe dCons e.cs le.cs
Build Events Debug Resources	Platform: Active () Start action	Any CPU) Any CPU) Search Solution Exp C Qu C C C Qu C C C C C C C C C C C C C C C C C C C	eryClas erySpe dCons e.cs le.cs leColu nsactio
Build Events Debug Resources Services	Platform: Active () Start action	Any CPU) Any CPU) Any CPU Any	eryClas erySpe dCons e.cs le.cs leColu nsactio ger.cs
Build Events Debug Resources Services Settings	Platform: Active () Start action ③ Start project ③ Start externa	Any CPU) Any CPU Any CPU Any CP	eryClas erySpe dCons e.cs le.cs leColu nsactio ger.cs r.cs
Build Events Debug Resources Services Settings Reference Paths	Platform: Active (/ Start action Start project Start externa Start browse Start options	Any CPU) Search Solution Exp C Qu C C C Qu C C Qu C C C Qu C C C C C C C C C C C C C C C C C C C	eryClas erySpe dCons e.cs le.cs leColu nsactio ger.cs r.cs ue.cs
Build Events Debug Resources Services Settings Reference Paths Signing	Platform: Active (/ Start action © Start project © Start externa © Start browse	Any CPU) Search Solution Exp Search Solution Exp C Qu C	eryClas erySpe dCons e.cs le.cs leColu nsactio ger.cs r.cs ue.cs w.cs

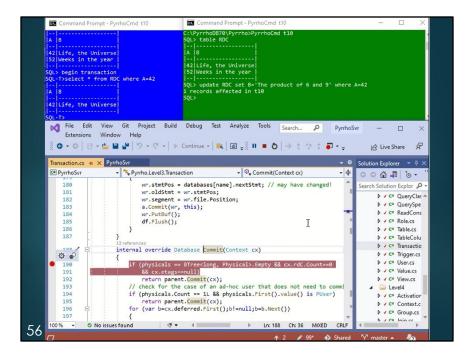
For the second part of the demonstration, for simplicity we will restore the state we had at slide 35: stop the server, delete the database, and repeat the set up to that point.

42 Life, the Univer 52 Weeks in the yea	begin transaction	
SQL> begin transacti SQL-T>select * from	on	
A B	select * from RDC where A=42	
42 Life, the Univer	se	
SQL-T>	w Git Project Build Debug Test Analyze Tools Search Q Durb	
File Edit View	w Git Project Build Debug Test Analyze Tools Search P Pyrrho dow Help	oSvr — 🗆 🗙
	🔛 🔐 り - C - ▶ Continue - 🔍 函 _ 🖁 🗉 = 🗿 → ‡ 🖓 ‡ 🖡	R Live Share
		E civesitate X
PyrrhoSvr 🕫 🗙		🗿 Solution Explorer 🛛 🔻 🖓
Application	Configuration: Active (Debug)	© ⊂ 🖓 🖉 •
Application Build	Configuration: Active (Debug)	Search Solution Explor 🔎
	Configuration: Active (Debug) Platform: Active (Any CPU)	Search Solution Explor 🔎 🕨 🗸 C# QueryClas
Build	Platform: Active (Any CPU)	Search Solution Explor 🔎
Build Build Events	Platform: Active (Any CPU) Start action	Search Solution Explor P
Build Build Events Debug	Platform: Active (Any CPU) Start action Start project	Search Solution Explor \mathcal{P} \Rightarrow < C" QueryClat \Rightarrow < C" QuerySpe \Rightarrow < C" ReadCons \Rightarrow < C" ReadCons \Rightarrow < C" Role.cs \Rightarrow < C" Table.cs
Build Build Events Debug Resources	Platform: Active (Any CPU) Start action	Search Solution Explor P
Build Build Events Debug Resources Services	Platform: Active (Any CPU) Start action Start project	Search Solution Explor P
Build Build Events Debug Resources Services Settings	Platform: Active (Any CPU) Start action • • Start project • Start external program: • Start browser with URL:	Search Solution Explor 0
Build Build Events Debug Resources Services Settings Reference Paths	Platform: Active (Any CPU) Start action • • Start project • Start external program: • Start browser with URL: Start options •	Search Solution Explor P
Build Build Events Debug Resources Services Settings Reference Paths Signing	Platform: Active (Any CPU) Start action • • Start project • Start external program: • Start browser with URL:	Search Solution Explor /

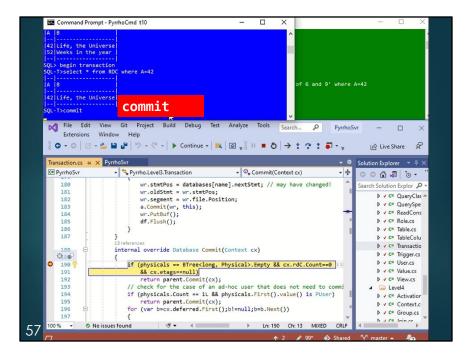
This time, when we start an explicit transaction in the blue window, instead of deleting something, we are going to select a single row from the RDC table, and we stick with A=42 again.

Command Prompt -	PyrrhoCmd t10 Command Prompt - PyrrhoCmd t10	- 0 ×
A B	C:\PyrrhoDB70\Pyrrho>PyrrhoCmd t10 SQL> table RDC	
42 Life, the Univer	 se A B	
	set B='The product of 6 and	9' where A=4
SQL-ISSELECT + +rom		
A B	1 records affected in t10	nere A=42
42 Life, the Univer	se SQL>	
 S0L-T>		
File Edit View	w Git Project Build Debug Test Analyze Tools Search 🔎 💡	yrrhoSvr — 🗆 🗙
	dow Help	
🕴 😋 🗸 😂 🕴 🖄 🚽	$\blacksquare \blacksquare \models \heartsuit \bullet \heartsuit \bullet \models Continue \bullet \blacksquare \blacksquare \models 1 \blacksquare \bullet 1 \Rightarrow \ddagger \heartsuit \bullet 1 \Rightarrow = $	🗧 🖄 Live Share 🖉
PyrrhoSvr + ×		
		000 J To - "
Application	Configuration: Active (Debug)	Search Solution Explor
Application Build	Configuration: Active (Debug) Platform: Active (Any CPU)	C O A II TO - " Search Solution Explor P - ▶ - C= QueryClat
Application Build Build Events		Search Solution Explor ♪ ~
Application Build Build Events Debug		Search Solution Explor \mathcal{P} + \flat + $\mathcal{C}^{=}$ QueryClas \flat + $\mathcal{C}^{=}$ QuerySpe \flat + $\mathcal{C}^{=}$ ReadCons
Application Build Build Events Debug Resources	Platform: Active (Any CPU)	Search Solution Explor ♪ ~
Application Build Build Events Debug Resources Services	Platform: Active (Any CPU) Start action Start project	Search Solution Explor P → ▷ < C* QueryClat
Application Build Build Events Debug Resources Services Settings	Platform: Active (Any CPU) Start action Start project Start external program:	Search Solution Explor P -
Application Build Build Events Debug Resources Services	Platform: Active (Any CPU) Start action Start project	Search Solution Explor P → ▷ < C* QueryClat
Application Build Build Events Debug Resources Services Settings	Platform: Active (Any CPU) Start action Start project Start external program:	Search Solution Explor ♪ ↓ c ⊂ QueryClat ↓ c ⊂ QueryClat ↓ c ⊂ ReadCons ↓ c ⊂ ReadCons ↓ c ⊂ TableCot ↓ c ⊂ TableCot ↓ c ⊂ TableCot ↓ c ⊂ Tansactio ↓ c ⊂ Tansactio
Application Build Build Events Debug Resources Services Services Settings Reference Paths	Platform: Active (Any CPU) Start action • • Start project • Start external program: • Start browser with URL: Start options •	Search Solution Explor ₽
Application Build Build Events Debug Resources Services Services Settings Reference Paths Signing	Platform: Active (Any CPU) Start action • • Start project • Start external program: • Start browser with URL:	Search Solution Explor P → ∀ ⊂ QueryClat ∧ ∀ ⊂ QueryClat ∧ ∀ ⊂ C QueryClat ∧ ∀ ⊂ C Rolecs ∀ ⊂ C Rolecs ∀ ⊂ C TableColu ∀ ⊂ TableColu ∀ ⊂ TableColu ∀ ⊂ C TableColu ∀ ⊂ C Value.cs ∀ ⊂ C Value.cs ∀ ⊂ C Value.cs ∀ ⊂ C Value.cs ↓ ⊂ C Value.cs ↓ ⊂ C Value.cs ↓ ⊂ C Value.cs
Application Build Build Events Debug Resources Services Settings Reference Paths Signing Security	Platform: Active (Any CPU) Start action • • Start project • Start external program: • Start browser with URL: Start options •	Search Solution Explor ₽
Application Build Build Events Debug Resources Services Setrings Reference Paths Signing Security Publish	Platform: Active (Any CPU) Start action • • Start project • Start external program: • Start browser with URL: Start options •	Search Solution Explor P → ▷ < c⊂ QueryClat →

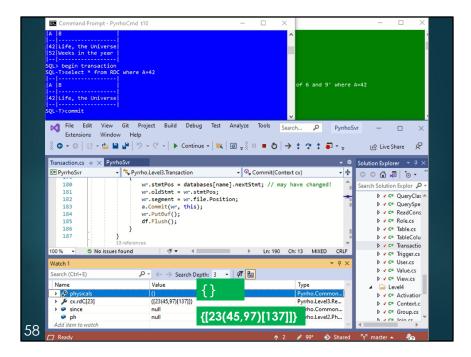
Just as before, the green window makes an update to the same row, which is auto-committed. For the reasons explained earlier, we expect that the blue window will now be unable to commit.



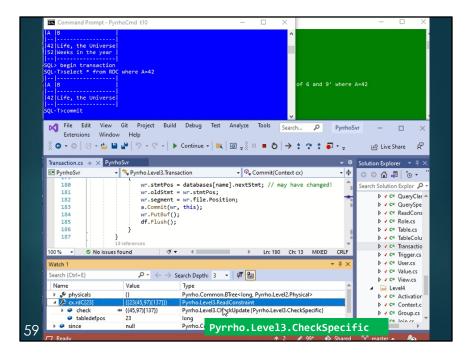
Ensure that we still have a break point in the Transaction.Commit() method.



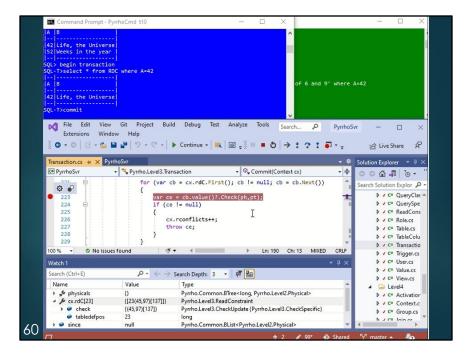
On the commit command, we hit the break point. Open the Watch window again.



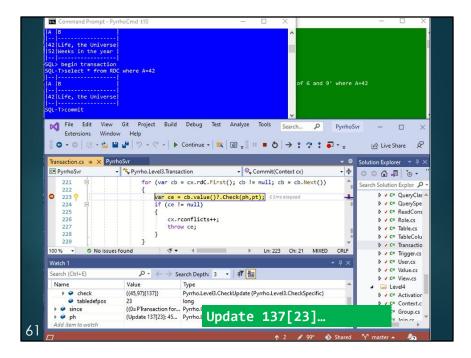
If we look at the Watch window that we had before, we can see that the physicals list is empty this time, but the read constraint has got something in it. (We haven't fetched **since** yet.)



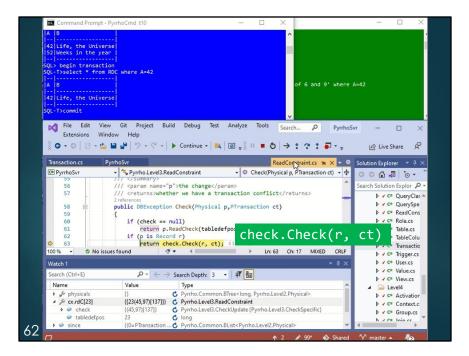
Expand it: we see that this is a CheckSpecific record: the check field shows the columns and the records that have been read.



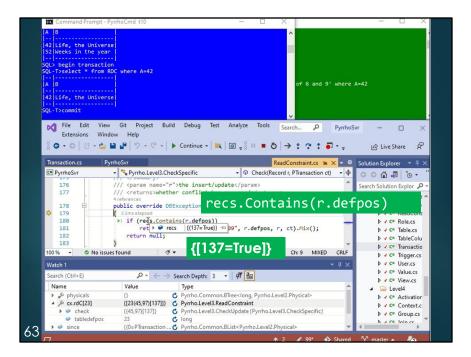
Since rdC is not empty, let's put a break point at line 223



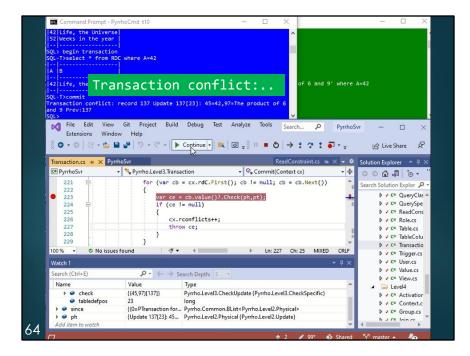
The second time this is hit, ph is the Update that was made by the green window. (The blue window just made a select.) Step Into the Check() call (step out of parameter evaluation, and step in again).



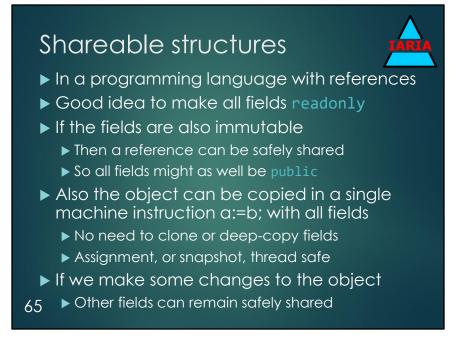
When we step into the ReadConstraint.Check() method, check is the CheckSpecific, and it checks the record r that has been committed by the green window. Step Into check.Check().



This looks at the list of records of the CheckSpecific, the records that were read by the blue window and it contains the record that has just been updated, so an exception occurs. Continue from this point.



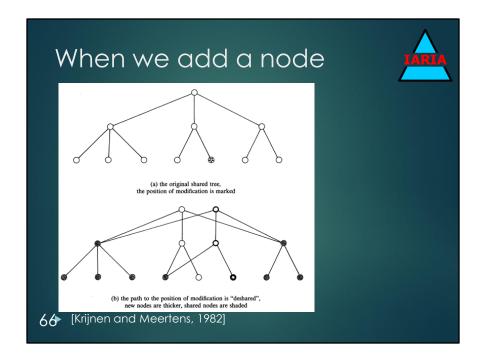
A suitable report is sent to the blue client, and the transaction is rolled back. This concludes the demonstration of transaction conflict.



The next question is how best to implement shareable data structures. In a programming language based on references, such as Java, or C#, we can make all fields in our structure final, or readonly. Then any reference can be safely shared, and all fields might as well be public (unless there are confidentiality issues).

If all of the fields are, in turn, also known to be immutable, then there is no need to clone or copy fields: copying a pointer to the structure itself gives read-only access to the whole thing. For example, if the **Database** class is shareable, and b is a database, then a:=b is a shareable copy of the whole database (we have just copied a single 64-bit pointer). This is also guaranteed to be thread-safe.

Pointers to shareable structures are never updated, but can be replaced when we have a new version to reference. If this new version has some changed fields, it is perfectly fine to continue to use the same pointers for all the unchanged fields.



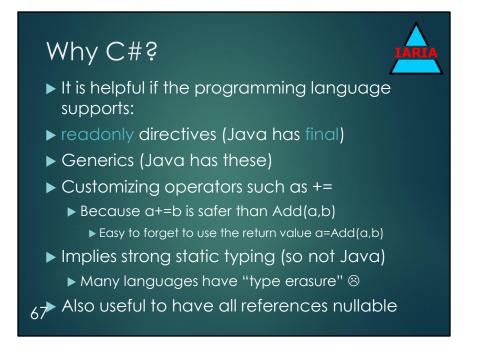
When we add a field located deep in a shareable structure (e.g. a node to a shareable tree structure), we will need to construct a single new node at each level back to the top of the structure. But the magic is that all the other nodes remain shared between the old and new versions.

The picture shows a tree of size 7, and updating (that is, replacing) one leaf node has resulted in just 2 new nodes being added to the tree. This makes shareable B-Trees into extremely efficient storage structures.

In tests, we see that for a suitable minimum number of

child nodes in the B-Tree, the number of new nodes required for a single update to a B-Tree of size N is O(logN), and experimentally, this means that for each tenfold increase in N, the number of new nodes per operation roughly doubles.

Note that we also get a new root node every time (this avoids wearing out flash memory).



Let us choose a popular programming language to recommend. We have readonly or final directives in C# and Java, and generics in both.

But C# allows us to create statically-defined operators such as +=, and this is a great advantage because a+=b allows no mistake, while it is very easy to think Add(a,b) adds b to a. (It does, but only if the result of the method is then assigned to a.)

Java also lacks strong static typing because of type erasure, and C# allows all references to be nullable by

default.

So I prefer C#, which now has been around for 19 years. Java and Python have been with us for over 30 years. However, C# provides no syntax for requiring a class to be shareable.



What data structures in the DBMS can be made shareable?

Database itself, and its subclass, Transaction.

Database Objects such as Table, Index, TableColumn, Procedure, Domain, Trigger, Check, View, Role

Processing objects such as Query, Executable, RowSet, and their many subclasses;

Cursor and most of its subclasses.

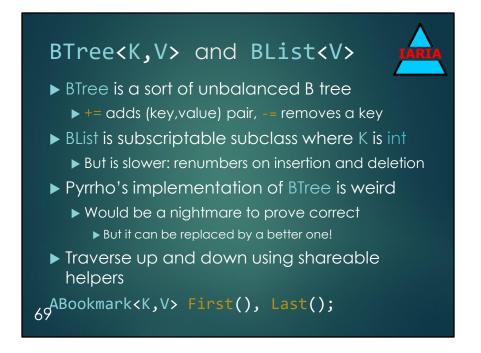
TypedValue and all its subclasses

All of these can be made shareable.

Context and Activation cannot be made shareable because in processing expressions we so often have intermediate values.

Also, something needs to access system non-shareable structures such as FileStreams, HttpRequest.

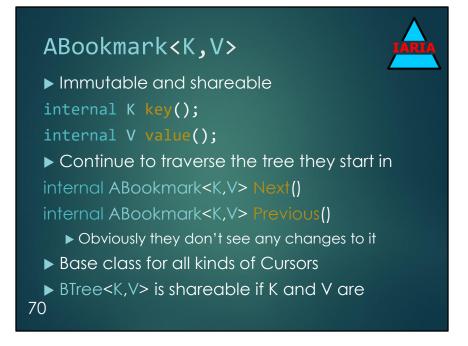
And Physical and its subclasses are used for preparing objects for the database file, so cursors that examine logs are not shareable.



We have already mentioned BTrees. These are used throughout the implementation and have proved very useful. As mentioned above, BTree<K,V> appears to exhibit logarithmic behaviour, It is also used for implementation of BList<V>, but BList is slower in order to ensure that the keys are always 0,1,...

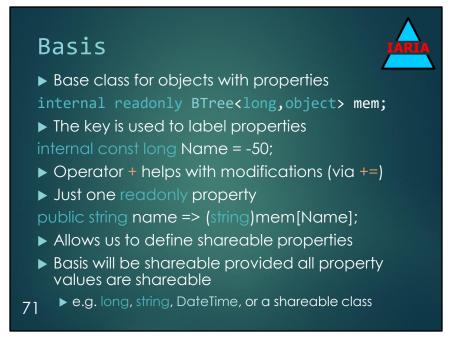
If someone can come up with a better (more provably correct) version of BTree, that would be great!

As described in the StrongDBMS paper, traversal of both BTree and BList is done using shareable bookmarks, and since last year both allow bidirectional traversal.



This slide give some more detail on Bookmarks. Given a Bookmark we can move to the Next, or Previous entry in the tree or list (and we get a new Bookmark of course). They continue to traverse the tree that returned them (which remains immutable). The Cursor class is a subclass of ABookmark and is the base class for numerous Cursor classes.

These structures are shareable provided that K and V are shareable.



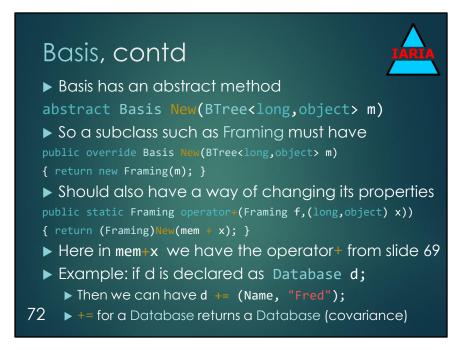
A key building block in the implementation is the Basis class, which is the parent of all of the shareable classes mentioned on slide 69, except TypedValue.

Basis itself contains just one field BTree<long,object> called mem, and has just one shareable property **name**. Object is not shareable: but a Basis object will be shareable if all its property values are.

Its importance comes from the fact that negative keys in mem provide a set of readonly properties. Each subclass of Basis can define its own property keys.

So Name is -50, and negative keys from -50 to -500 are reserved for such properties. Keys below -500 are used for predefined types and system tables.

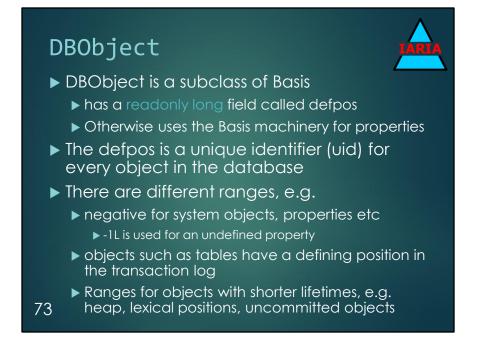
Any subclass of Basis can define operator+ to give useful ways of updating properties.



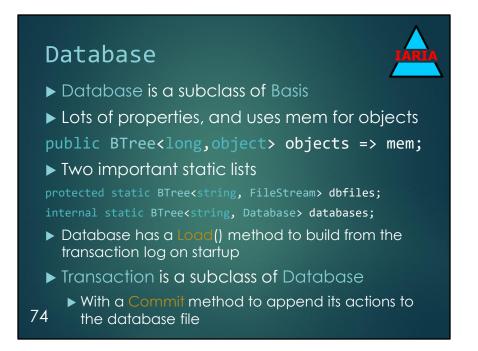
This slide adds some more technical detail for the Basis class and how its subclasses work.

Every subclass must implement New. Provided every subclass also implements + similarly to the one shown, we do not need a new constructor for modifying every kind of field in a shareable subclass.

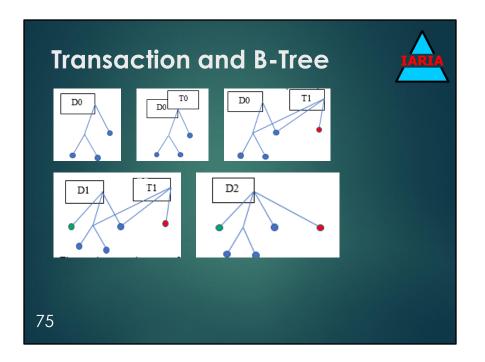
Then whatever subclass x is declared to be, x+(k,v) will be the same class for any long k.



DBObject is a very useful abstract subclass of Basis. It has one new field defpos, which is a readonly long. This is where unique identifiers come in: all defpos are uids. The range of values long.MinValue to long.MaxValue is divided into 6 ranges as described here. Uids are faster to compare than string identifiers, and easy to come up with a next value.



In the demos we have seen reference to databases and dbfiles. Here we see how they fit into the overall design.



The StrongDBMS paper gives a step-by step account of the way Transaction commit works. This is basically a BTree implementation for the process described earlier (slide 27).

Suppose we have a database in any starting state D0.

If we start a transaction T0 from this state, initially the T0 is a copy of D0 (i.e. equal pointers).

As T0 is modified it becomes T1, which shares many nodes with D0, but not all. D0 also evolves as some other transaction has committed, so D1 has a new root and some new nodes.

When T1 commits, we get a new database D2 incorporating the latest versions of everything.

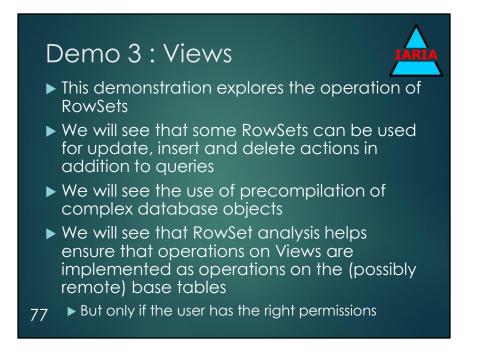


Many other DBMS describe their process of "Query optimisation".

The result of parsing data manipulation language in Pyrrho v7 is not an optimised Query but a possibly updatable RowSet.

So instead of Query Optimisation, Pyrrho v7 has RowSet review. In this process as we will see, filters and aggregations are pushed down from the query level as far as possible towards the table level.

For example, during Rowset Review we replace selection from views and joins by selection underlying tables. This leads to a simple implementation of updatable views and joins.



This demonstration explores the operation of RowSets

We will see that RowSets can be used for update, insert and delete actions in addition to queries

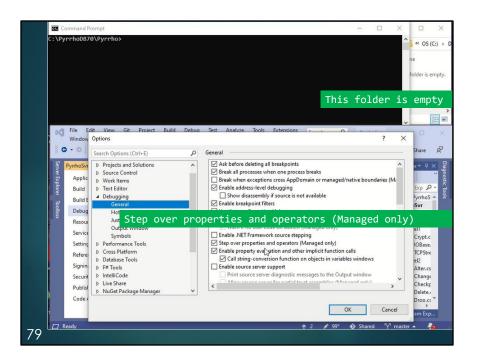
We will see the use of precompilation of complex database objects

We will see that RowSet analysis helps ensure that operations on Views are implemented as operations on the (possibly remote) base tables, but only if the user has the right permissions.

C:\Py	/rrhoDB70\Pyrrho>				^		
						s « OS	(C:)
						ne	
						folder is	emi
						_	
							8==
N	File Edit View Gi	t Project Build Debug Test Analyze Tools Extensions Search 🔎	Pyrrh	noSvr	-		
	Window Help						
G	o - o 🏗 - 🏠 💾 🗳	🎽 🦻 🗸 🖓 👻 🕨 Start 👻 🚉 🖬 🖕			🖻 Liv	e Share	3
ě	PyrrhoSvr + ×		- ¢	Soluti	ion Explo	orer 🖛 🕂	×
Server E			- ¢	1		orer 🔻 🖟	×
Server Explor	PyrrhoSvr + × Application Build	Configuration: Active (Debug) V Platform: Active (Any CPU) V	- ¢	0	o 🙆 🕯		••
ver Explorer	Application		- 0	Searc	C 🟠 🖬	n Exp 🖌	" - C
	Application Build	Start action	- 0	Searc	C C C C C C C C C C C C C C C C C C C	n Exp PyrrhoS hoSvr	" • C
Server Explorer Toolbox	Application Build Build Events	Start sction (a) Start project	- 0	Searc	Solution	n Exp 🖌	•
	Application Build Build Events Debug	Start action	~ @	Searc	Colution Solution Col	I PyrrhoS PyrrhoS hoSvr eferences formon evel1	· · ·
	Application Build Build Events Debug Resources	Start sction (a) Start project	~ @	Searc	Colution Solution Col	n Exp PyrrhoS hoSvr eferences	••• ••
	Application Build Build Events Debug Resources Services	Start action © Start project Start external program:	~ ©	Searc	Solution C Pyrrl R C C C C C C C C C C C C C C	I PyrrhoS hoSvr eferences ommon evel1 Crypt.c IOBase TCPStre	•••
	Application Build Build Events Debug Resources Services Services Settings	Start action Start project Start external program: Start browser with URL: Start options	~ ©	Searc	Contraction Contr	PyrrhoS PyrhoS Pyrh	•
	Application Build Build Events Debug Resources Services Settings Reference Paths	Start action Start project Start external program: Start browser with URL:	~	Searc	Control Contr	I PyrrhoS hoSvr eferences ommon evel1 Crypt.c IOBase TCPStre	• 0 • 0 • 1 • 0 • 1 • 1 • 1 • 1 • 1 • 1 • 1 • 1 • 1 • 1
	Application Build Build Events Debug Resources Services Settings Reference Paths Signing	Start action Start project Start project Start external program: Start proviser with URL: Start options Command line arguments: -d\DATA	~ *	Searc	A solution Solution Or Pyrrl C	Crypt.cc Crypt.cc Crypt.cc TCPStre Crypt.cc TCPStre Crypt.cc Alter.cs Change Checkg	
	Application Build Build Events Debug Resources Services Settings Reference Paths Signing Security	Start action Start project Start external program: Start browser with URL: Start options		Searc		the second	

The slide shows a client command window, a glimpse of a File explorer, and a window for Visual Studio. This

demonstration traces through part of test12 of the PyrrhoTest program using the Visual Studio debugger. In Visual Studio, open the PyrrhoSvr solution in the src\Shared folder of the distribution. Set the debug properties of the PyrrhoSvr project to have -d:\DATA in the Command line arguments.



From the Debug menu, select Options.. and ensure that the "Step over properties and operators (Managed only)" in Debugging/General is checked. Click OK.

				→ · ↑ <mark>·································</mark>
				Quic
	C:\PyrrhoDB70\Pyrrho\src\S	hared\bin\Debug\PvrrhoSvr.exe —		This folder is empty.
	-d:\DATA Enter to start up Pyrrho DBMS (c) 2021 Malcolm C	rowe and University of the West of Sc rrhodb.com https://pyrrhodb.uws.ac.uk		- * * * v <
				ns 🔠 🖬
File Edit View				Svr – 🗆 X
Extensions Wind				ය Live Share 오
Watch 1 PyrrhoSy	/r -			Solution Explorer 🛛 🝷 🕂 🗙
Application				○○☆ Л 'o・ "
Build				Search Solution Explor 🔎 -
Build Events				✓ 🔄 Solution 'PyrrhoSvr' 🔺
Debug				C PyrrhoSvr References
Resources				🖌 🔛 Common 👘
Services	 Start project 			↓ C# BTree.cs ↓ C# CTree.cs
Settings	 Start external program: 			♦ ✓ C [#] DataType.
Reference Paths	O Start browser with URL:			♦ ✓ C# Integer.cs
Signing	Start options			↓ ✓ C# Interfaces ↓ ✓ C# Pyrrho.cs
Security	Command line arguments:			♦ ✓ C [#] Row.cs
Publish	command line arguments:	-d:\DATA		♦ ✓ C# TDocume ♦ ✓ C# Tree.cs
Code Analysis				↓ C+ Iree.cs
1	<		× ×	♦ ✓ C# Crypt.cs

Now click Start in the debugger, and in the popup command window, click Enter. We hide this window because it is not going to do anything interesting during this demo.

We want to ensure that the database folder that we are using for our databases is currently empty before we start the demonstration.

C:\PyrrhoDB70\Pyrrh			^
sqL>	PyrrhoCmd t12		• OS (C:)
SQL>			ne
	•		t12
			112
			t12
File Edit View	Git Project Build Debug Test Analyze Tools Extensions course O		×
File Edit View Window Help	Git Project Build Debug Test Analyze Tools Extensions Search 🔎		
🛛 🖸 🗸 💭 👘 🖓 🖕 🖆	💾 💾 🤚 🗸 🖓 🔹 🕨 Continue - 📴 🐻 💷 🛯 🔳 🙆 🔿 🔅 🖓 🚛 👘		I Live Share
	שו אין	_	
Watch 1 7 PyrrhoSv		- 4	Solution Explorer
Watch 1 # PyrrhoSv Application		- 4	Solution Explorer
Watch 1 7 PyrrhoSv Application Build	л н X	- 4	Solution Explorer
Watch 1 # PyrrhoSv Application	л н X	- <	Solution Explorer Solution Explorer Search Solution Exp Search Solution Pyrrho
Watch 1 7 PyrrhoSv Application Build	n + X Configuration: Active (Debug) V Platform: Active (Any CPU) V Start action	- «	Solution Explorer
Watch 1 9 PyrrhoSv Application Build Build Events	rr • × Configuration: Active (Debug) V Platform: Active (Any CPU) Start action © Start project	• «	Solution Explorer Search Solution Explorer Search Solution Exp Solution Pyrrho Solution PyrrhoSvr Solution PyrrhoSvr Search Solution Exp Solution PyrrhoSvr Solution Explorer Common
Watch 1 9 PyrrhoSv Application Build Build Events Debug	r • × Configuration: Active (Debug) Start action • Start project • Start external program:	· <	Solution Explorer
Watch 1 PyrrhoSv Application Build Build Events Debug Resources	rr • × Configuration: Active (Debug) V Platform: Active (Any CPU) Start action © Start project	• •	Solution Explorer Search Solution Explorer Search Solution Exp Solution Pyrrho Solution PyrrhoSvr Solution PyrrhoSvr Search Solution Exp Solution PyrrhoSvr Solution Explorer Common
Watch 1 9 PyrrhoSv Application Build Build Events Debug Resources Services	r • × Configuration: Active (Debug) Start action • Start project • Start external program:	• •	Solution Explorer ● Search Solution Pyrrho ✓ ✓ ✓ ✓ ✓ ✓ ✓
Watch 1 9 PyrrhoSu Application Build Build Events Debug Resources Services Settings	r • × Configuration: Active (Debug) Platform: Active (Any CPU) Start action Start project Start external program: Start browser with URL: Start options		Solution Explorer ← Constant for the solution of the solution for the solution fore solution for the solution for the solution for the solution for
Watch 1 9 PyrrhoSu Application Build Events Debug Resources Services Settings Reference Paths Signing	Active (Debug) Platform: Active (Any CPU) Start action • • • • •	• • •	Solution Explorer ● Solution Explorer ● Solution Pyrthol ✓ C PyrthoSvr ✓ C PyrthoSvr ✓ C PyrthoSvr ✓ C PyrthoSvr ✓ C Corption ✓ C Corpti
Watch 1 9 PyrmoSu Application Build Build Events Debug Resources Services Setvices Settings Reference Paths Signing Security	r • × Configuration: Active (Debug) Platform: Active (Any CPU) Start action Start project Start external program: Start browser with URL: Start options		Solution Explorer ● Solution Explorer ● Solution Pyrthol CopyrtaSvr B = Common copyrtaSvr copyr copyrtaSv
Watch 1 9 PyrrhoSu Application Build Build Events Debug Resources Services Settings Reference Paths Signing	r • × Configuration: Active (Debug) Platform: Active (Any CPU) Start action Start project Start external program: Start browser with URL: Start options		Solution Explorer Solution Explorer Solution Explorer Solution Pyrrho Solution Pyrrho Common Co

In a command window set to the distribution folder, start the command line client PyrrhoCmd for database t12. We see that the database is created by the server.

C:\Users\Malco>cd	\pyrrhodb70\py*		^ « OS
C:\PyrrhoDB70\Pyrr SQL> create table SQL> create view v SQL> table "Log\$"	ho>PyrrhoCmd t12 p(q int primary key,r char,a int) as select q,r as s,a from p		ne t12
reate ta	ble p(q int primary key,r char,	a i	nt)
	101 9 1020-302 0301-302 1200-207 09/2022 09/39/2019 101302 020019 107-63-		
reate vi	ew V as select q,r as s,a from	р	
DC THE LOR VIEW	A on Project build bedrug lest Analyze loois Extensions Search	PyrrhoSv	r – D
able "Lo	g \mathbb{C}^*		🔓 Live Share
Watch 1 7 Pyrrho	Svr + X	- 0	Solution Explorer
Application	Configuration: Active (Debug) V Platform: Active (Any CPU) V		0044
Build			
Build Events	Start action -	_^	Solution 'P
Build Events Debug			G Solution 'P G Solution 'P G Pyrrhos D S Refe
Build Events Debug Resources	Start action		Solution 'Py C Pyrrhos C Pyrrhos
Build Events Debug Resources Services	Start action		Search Solution E
Build Events Debug Resources Services Settings	Start action		✓ [c] Solution 'P; ▲ ✓ [c] Pyrrho ▷ ::" Refe ▷ :: Con ▲ :: Leve ▷ ✓ c: (▷ ✓ c: ()
Build Events Debug Resources Services Settings Reference Paths	Start action		✓ □ Solution 'P; ▲ ✓ □ Pyrrhos ▷ □ Pyrhos ▷ □ Con ▲ □ Leve ▷ ✓ □ ↓ ▷ ✓ □ ↓ ▷ ✓ □ ↓
Build Events Debug Resources Services Settings Reference Paths Signing	Start action		✓
Build Events Debug Resources Services Settings Reference Paths	Start action		✓ a Solution 'P' ▲ ✓ c Pyrrhos ▷ a Refe ▷ a Com ▲ Com ▲ Com ↓ c a U ▷ ✓ c a U ▷ ✓ c a U

In the command window, enter commands to create a table P with three columns, and a View which uses this table, renaming a column. We also want to look at the transaction log.

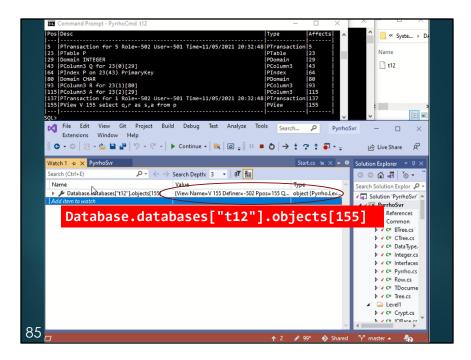
5 PTransaction fo 23 PTable P 29 Domain INTEGER 33 PColumn3 Q for 34 PIndex P on 231 36 Domain CHAR 33 PColumn	23(0)[29] PC	PTransaction 5 PTable 23 Poomain 29 Column3 43 PTodex 64 Promain 80
15 PColum PV1	v Git Project Build Debug Test Analyze Tools	•View 155 ↓ * ↓ < ↓ n ■
	Image: Second secon	
Debug Resources	Start action ① Start project ③ Start external program:	
Services Settings Reference Paths Signing Security Publish	Start browser with URL: Start options Command line arguments: -d\DATA	

In the log file, we see that the View definition in the database file is just recorded as the source of the select definition of the view.

When the server processes this, on load or on definition, it creates some compiled components, so that this statement doesn't have to be parsed every time it is used.

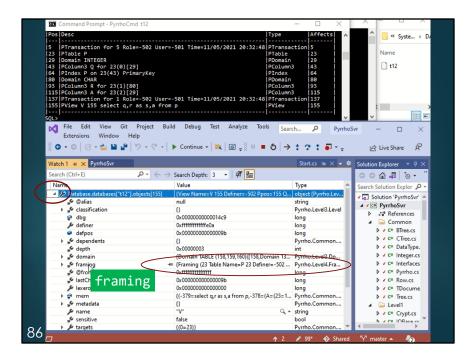
Pos Desc	Туре	Affects ^ ~ ^ A Syste
23 PTable P 29 Domain INTEGER 43 PColumn3 Q for 64 PIndex P on 23(80 Domain CHAR 93 PColumn3 R for 115 PColumn3 A for 137 PTransaction fo	43) PrimaryKey PIndex PDomain 23(1)[80] PColumn3	23 Name 29 Quic 43 t 164 t 180 r 115 r
	Search	PyrrhoSvr - □ Solution Explorer Solution Explorer Search Solution Explorer Search Solution PyrrhoSvr
Debug	Start action	▲ ✓ C■ PyrrhoSvr
Resources Services	Start project Start external program:	Common
Settings	O Start browser with URL:	Þ ✓ C# Integer.c
Settings Reference Paths		h d Cit Interface
-	Start options	↓ ✓ C* Interface
Reference Paths		↓ ✓ C [#] Pyrrho.c ↓ ✓ C [#] Row.cs
Reference Paths Signing	Start optionsCommand line arguments:	♦ 🗸 C# Pyrrho.c

Let's pause the server so we can look at the precompiled objects. Notice I have docked the Watch window as a tabbed document.

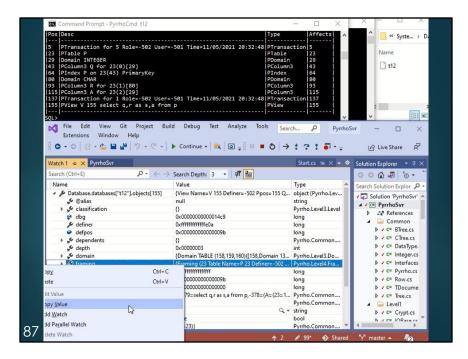


In the Watch window, examine

Database.databases["t12"].objects[155] the View position. We see that there is a View defined in the Value.



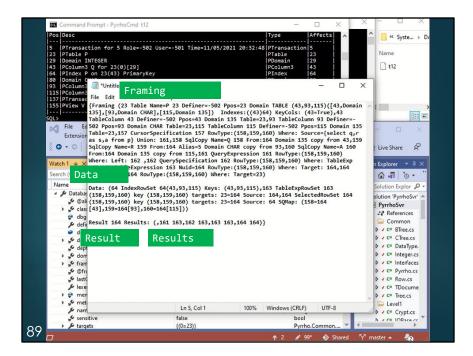
Let's expand that (it takes the server a moment to do it), and we see that there is a Framing field, and this contains the compiled components of the definition.



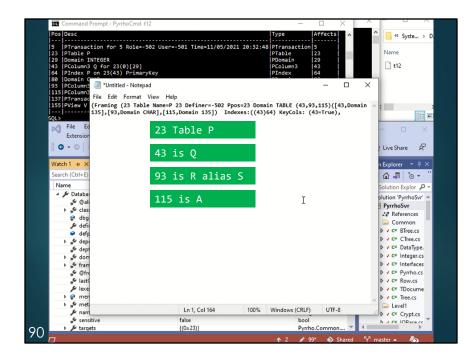
Right click, and Copy the value of the framing field.

	1 3	rhoCmd t12		-		
Pos De	esc			Type A	ffects ^	🍰 🦲 « Syste » D/
23 P 29 D 43 P 64 P	Table P omain INTEGER Column3 Q for 23(Index P on 23(43)		1/05/2021 20:32:4	PTable 2 PDomain 2 PColumn3 4	3	Name
93 P	Column	ed - Notepad Format View Help		100 10	- 0	×
Watch	135], j[93, D TableColum Table201 Table23,1 as s,a fro SqlCopy Na From:164 D Here: Lef 13,163 Ta	(23 Table Name=P 23 Definer= Domain (tAR) [115,0omain 135 mn 43 Definer=-502 Ppos=43 D. 93 Domain (tAR Table=23,115 157 Cursor5pecification 157 I om p] Union: 161,158 SqlCopy ame=R 159 From:164 Alias=5 D. Domain 135 copy from 115,161 ft: 162,162 QuerySpecificat ableExpression 163 Nuid=164	<pre>]) Indexes:((43 omain 135 Table= TableColumn 115 RowType:(158,159 Name=Q 158 From omain CHAR copy QueryExpression ion 162 RowType: RowType:(158,159</pre>)64) KeyCols: (4 23,93 TableColumn Definer=-502 Ppos 1660) Where: Sour :164 Domain 135 c from 93,160 SqlCc 161 RowType:(158 (158,159,160) Whe 1660) Where: Targ	B=True),43 193 Definer= 5=115 Domain cce={select q copy from 43, py Name=A 16 8,159,160) pre: TableExp get: 164,164	135 × 159 Ø Live Share
Nam	e (43,93,115 targets: 2 Databa: targets: 2	<pre>P164 RowType:(158,159,160) 5 Keys: (43,93,115),163 Tab 23=164 Source: 164,164 Sele: (161 163,162 163,163 163,164 (,161 163,162 163,163 163,164)) </pre>	letxpRovSet 163(1) tedRovSet 164(15) 8=164[43],159=16 4 164))	158,159,160) key (8,159,160) key (4[93],160=164[115	(158,159,160) [58,159,160) [])) Result 1	Solution Explor 🔎 👻
Nam	(43,93,115 targets: 2 Datbat targets: 2 \$ © Baik Results: (\$ C class \$ defi \$ defi } defi \$ defi \$ defi } defi } defi \$ defi } defi defi } defi defi defi } defi defi defi } defi defi defi	5) Keys: (43,93,115),163 Tab 23=164 Source: 164,164 Selec 23=164 Source: 64 SQMap: (15	letxpRovSet 164(15 teRdoxSet 164(15 ≋-164[43],159=16 4 164))	158,159,160) key 8,159,160) key (1	(158,159,160 (58,159,160)	 Solution Explor P PyrrhoSvr → PyrrhoSvr → PyrrhoSvr → Common C Biree.cs C Ciree.cs C DataType. C DataType. C DataType. C Integer.cs C Pyrrho.cs C Row.cs C Row.cs C Row.cs C Tocume C Tocume C Tocume C Tee.cs Level
Nam	(43,93,114) targets: 2 Databå targets: 2 \$ Cals \$ day \$ day \$ dep \$ fan \$ solt: \$ last \$ last \$ mer \$ mer \$ mer \$ me	5) Keys: (43,93,115),163 Tab 23=164 Source: 164,164 Selec 23=164 Source: 64 SQMap: (15 (,161 163,162 163,163 163,16	letxpRovSet 163(1) tedRovSet 164(15) 8=164[43],159=16 4 164))	158,159,160) key (8,159,160) key (4[93],160=164[115	(158,159,160) [58,159,160) [])) Result 1	Solution Explor > 64 Jution 'PyrhöSvr' > PyrhöSvr' > References > Common > <

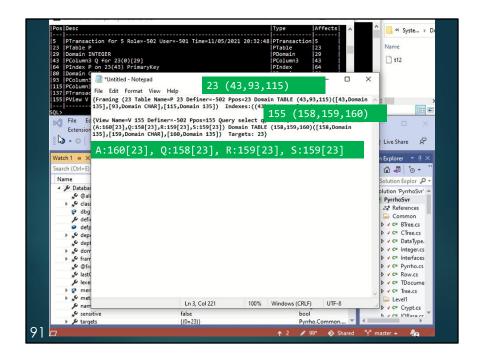
Open a Notepad window, and click Paste. We see there is a fair bit in the framing field. We don't need to look at all this detail. All we want to do here is to see that the Framing is in four parts. We add a little white space in the next slide



We can see there is a section of objects, a Data section for the rowsets, and a list of results.

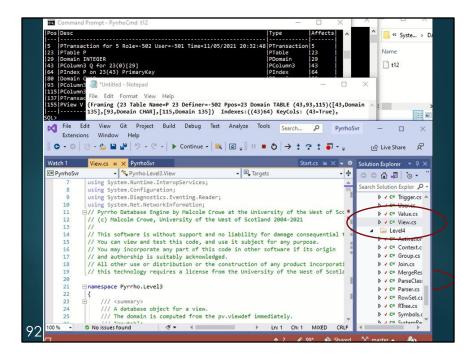


Just now, we will delete everything except the first two lines, which are the details for table P.



Let's also paste the View definition from the Watch window in here, so that we can see what's going on. We notice straightaway that the Domain of the View and the domain of the table don't match.

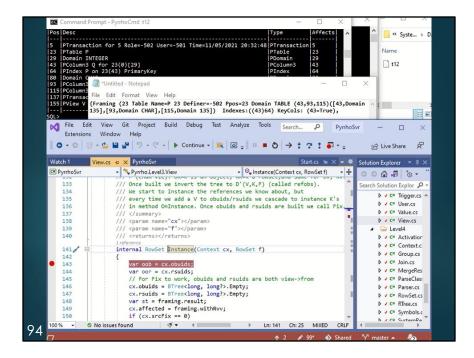
There is a relationship, as the types of the columns match, but the uids are all different. In the table, they are of the columns, but in the View these are some of the compiled objects.



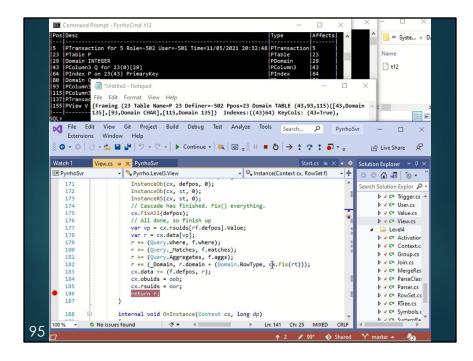
In Solution Explorer, scroll down to find View at the end of Level 3, and double-click.

PoslDesc	and Prompt - PyrrhoCmd t12	Tura		
PosiDesc		Туре	Attects ^	Syste >
	saction for 5 Role=-502 User=-501 Time=11/			
23 PTabl	e P n INTEGER	PTable PDomain	23	Name
43 PColu	mn3 Q for 23(0)[29]	PColumn3	43	🗋 t12
64 PInde 80 Domai	x P on 23(43) PrimaryKey	PIndex	64	
93 PColu	mn 🗉 🦾 *Untitled - Notepad		- 0	×
115 PColu				
155 PView	300	02 Ppos=23 Domain TABLE (43	93,115)([43,Dom	in Ar
	135],[93,Domain CHAR],[115,Domain 135])			1
SQL>	Edit View Git Project Build Debug T	est Analyze Tools Cased		
File Exten		est Analyze Tools Search	PyrrhoSv	n – 🗆 🗙
2020	The second se			-
0.0	View.cs Continue •	The second se	🚰 🖡 🛡 T 👳 👘	🖻 Live Share 🛛 🕅
Watch 1	View.cs + × PyrrhoSvr	Targets	.cs 🖮 🗙 👻 🍄	<u>сыс сы. — П</u> ъ
				Solution Explorer 🛛 🔻 🤻 🛪
C# PyrrhoSv		- E, Targets	<mark>-</mark> ÷	○ ○ ☆ 📲 '⊙ •
7	using Syste		^ <u> </u>	Search Solution Explor 🔎
9	using Syste			Þ ✓ Cª Trigger.cs
10	using Syste @ Instance(Context cx, Row Set f)			♦ ✓ C# User.cs
11	D// Pyrrho D , InstanceOb(Context cx, long dp)	Tong why)		Value.cs
12	// (c) Malc // Instance	Context cx, Row	$S_{0} + f$	♦ ✓ C# View.cs
	// This sof @ Modify(C	concext ex, now	Jet I)	🖌 📄 Level4
15				♦ ✓ C# Activation
	// You can Θ_{\bullet} New(BTree <long, object=""> m)</long,>			
14 15 16	// You may Q New(Context cx BTreeslong ob	iect> m)		♦ ✓ C [#] Context.c
14 15 16 17	// You may // and auth		1	Þ ✓ C# Group.cs
14 15 16 17 18	// You may // and auth // All othe \$\Phi_New(Context cx, BTree <long, ob<br="">\$\Phi_Onlinetance(Context cx, long dp)</long,>			♦ ✓ C# Group.cs ♦ ✓ C# Join.cs
14 15 16 17 18 19	// You may // and auth // All othe // this tec B operator + (View v, (long, object)			 ↓ ✓ C# Group.cs ↓ ✓ C# Join.cs ↓ ✓ C# MergeRes
14 15 16 17 18 19 20	// You may // and auth // All outh // All outh // this tec Pamespace Pamespace Pamespace // Second Second Pamespace Pamespace Pamespace // Second Second Pamespace) x)		 ↓ C= Group.cs ↓ C= Join.cs ↓ C= MergeRes ↓ C= ParseClas:
14 15 16 17 18 19	// You may // and auth // All oth // All oth // this tec manuspace // kip oprator + (View v, (long, object) @, Relocate(long dp) manuspace // Relocate(long dp)		ct> _enu)	 ▷ ✓ C[#] Group.cs ▷ ✓ C[#] Join.cs ▷ ✓ C[#] MergeRes ▷ ✓ C[#] ParseClas: ▷ ✓ C[#] Parser.cs
14 15 16 17 18 19 20 21 22 23	// You may // and auth // All outh // All outh // this tec Pamespace Pamespace Pamespace // Second Second Pamespace Pamespace Pamespace // Second Second Pamespace) x) 30bject from, ABookmark <long, obje<="" td=""><td></td><td> ↓ C^{ass} Group.cs ↓ C^{ass} Join.cs ↓ C^{ass} MergeRes ↓ C^{ass} ParseClas: ↓ C^{ass} Parsec.cs ↓ C^{ass} RowSet.cs </td></long,>		 ↓ C^{ass} Group.cs ↓ C^{ass} Join.cs ↓ C^{ass} MergeRes ↓ C^{ass} ParseClas: ↓ C^{ass} Parsec.cs ↓ C^{ass} RowSet.cs
14 15 16 17 18 19 20 21 22 23 24	// You may 0, New(Context cx, BTre <long, ob<br="">// and auto: // All othe 0, Oninstance(Context cx, long dp // this tec B operator +(View v, (long, object) 0, Relocate(long dp) 1 { 0, RoleClassValue(Transaction tr, DI 0, RoleJavalue(Transaction tr, DI)</long,>) x) 30bject from, ABookmark <long, obje<br="">0bject from, ABookmark<long, objec<="" td=""><td>ct>_enu)</td><td> ↓ C[®] Group.cs ↓ C[®] Join.cs ↓ C[®] MergeRes ↓ C[®] ParseClas: ↓ C[®] Parser.cs ↓ C[®] RowSet.cs ↓ C[®] RTree.cs </td></long,></long,>	ct>_enu)	 ↓ C[®] Group.cs ↓ C[®] Join.cs ↓ C[®] MergeRes ↓ C[®] ParseClas: ↓ C[®] Parser.cs ↓ C[®] RowSet.cs ↓ C[®] RTree.cs
14 15 16 17 18 19 20 21 22 23	// You may 0, New(Context cr, BTre <long, ob<br="">// And auto: 0, Oninstance(Context cr, Jong dp // this tec Bopertor-(View v, (Jong, object) [namespace 0, RoleClassValue(Transaction tr, D [] /// <su 0,="" d<br="" roleclassvalue(transaction="" tr,="">[] /// <su 0,="" d<="" rolezavvalue(transaction="" td="" tr,=""><td>x) 30bject from, ABookmark<long, obje<br="">Object from, ABookmark<long, objec<br="">DBObject from, ABookmark<long, ob<="" td=""><td>ct>_enu)</td><td> V C* Group.cs V C* Join.cs V C* MergeRes V C* ParseClas: V C* Parsec.s V C* RowSet.cs V C* Rifee.cs V C* Symbols.c </td></long,></long,></long,></td></su></su></long,>	x) 30bject from, ABookmark <long, obje<br="">Object from, ABookmark<long, objec<br="">DBObject from, ABookmark<long, ob<="" td=""><td>ct>_enu)</td><td> V C* Group.cs V C* Join.cs V C* MergeRes V C* ParseClas: V C* Parsec.s V C* RowSet.cs V C* Rifee.cs V C* Symbols.c </td></long,></long,></long,>	ct>_enu)	 V C* Group.cs V C* Join.cs V C* MergeRes V C* ParseClas: V C* Parsec.s V C* RowSet.cs V C* Rifee.cs V C* Symbols.c
14 15 16 17 18 19 20 21 22 23 24	// You may 0, New(Context cx, BTre <long, ob<br="">// and auto: // All othe 0, Oninstance(Context cx, long dp // this tec B operator +(View v, (long, object) 0, Relocate(long dp) 1 { 0, RoleClassValue(Transaction tr, DI 0, RoleJavalue(Transaction tr, DI)</long,>	x) 30bject from, ABookmark <long, obje<br="">Object from, ABookmark<long, objec<br="">DBObject from, ABookmark<long, ob<="" td=""><td>ct>_enu)</td><td> ↓ C[#] Group.cs ↓ C[#] Join.cs ↓ C[#] MergeRes ↓ C[#] ParseClas: ↓ C[#] Parser.cs ↓ C[#] RowSet.cs ↓ C[#] RTree.cs </td></long,></long,></long,>	ct>_enu)	 ↓ C[#] Group.cs ↓ C[#] Join.cs ↓ C[#] MergeRes ↓ C[#] ParseClas: ↓ C[#] Parser.cs ↓ C[#] RowSet.cs ↓ C[#] RTree.cs

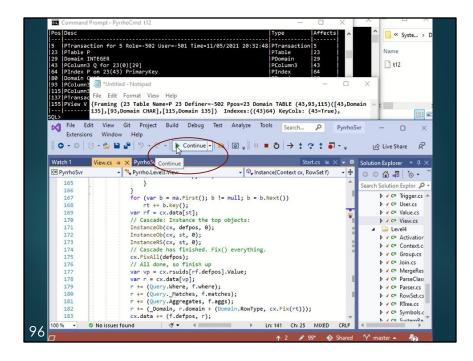
Find the Instance() method, and click on it.



In the Instance() method, set a break point at line 143,



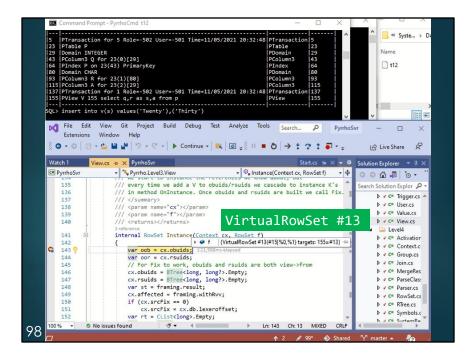
and also a breakpoint at line 186, at the end of the Instance method. The View.Instance() method responds to a reference to the View.



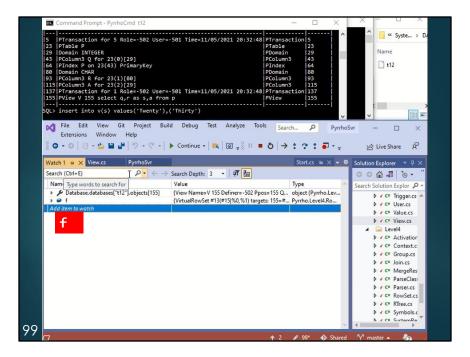
Click Continue, to allow the server to continue execution.

Pos Desc	Туре	Affects	^ Syste >
			- System +
5 PTransaction 23 PTable P	o for 5 Role=-502 User=-501 Time=11/05/2021 20:32:48 PTransact	23	Name
29 Domain INTEC		29	Truthe .
43 PColumn3 Q 1		43] t12
	23(43) PrimaryKey PIndex	64	
80 Domain CHAR	PDomain	80	
93 PColumn3 R +	or 23(1)[80] PColumn3	93	
	+ into w(c) waluec (!Twend		at intervel N
🗄 inser	t into v(s) values('Twen1	ال) ر y.	iirty)
			8=
	v(s) values('Twenty'),('Thirty')	v	
File Edit	View Git Project Build Debug ⁴⁵ Test Analyze Tools Search	P PyrrhoSvr	
Extensions	Window Help		
So.ola.	🍰 💾 🔐 ') + 연 + 🕨 Continue + 🔤 🛛 🚽 🔲 = 🔕 →	at T.	R Live Share
× • • • • • • • • •		e (e) e ● 1 =	Live share X
Watch 1 Vie	w.cs 🛥 🗙 PyrrhoSvr	Start.cs 🖮 🗙 👻 👳	Solution Explorer 🛛 🔫 🔻
C# PyrrhoSvr	Pyrrho.Level3.View Pyrrho.Level3.View Pyrrho.Level3.View		· 0 B. 6 0 0
	12 2		
165	}		Search Solution Explor
165 166	} } }		Search Solution Explor 👂
165 166 167	<pre>} } for (var b = ma.First(); b != null; b = b.Next())</pre>		Search Solution Explor P
165 166 167 168	<pre>rt += b.key();</pre>		Search Solution Explor P
165 166 167 168 169	<pre>rt += b.key(); var rf = cx.data[st];</pre>		Search Solution Explor
165 166 167 168 169 170	<pre>rt += b.key(); var rf = cx.data[st]; // Cascade: Instance the top objects:</pre>		Search Solution Explor
165 166 167 168 169 170 171	<pre>rt += b.key(); var rf = cx.data[st]; // Cascade: Instance the top objects: InstanceOb(cx, defpos, 0);</pre>		Search Solution Explor P
165 166 167 168 169 170	<pre>rt += b.key(); var rf = cx.data[st]; // Cascade: Instance the top objects: InstanceOb(cx, defpos, 0); InstanceOb(cx, st, 0);</pre>		Search Solution Explor P
165 166 167 168 169 170 171 171 172	<pre>rt += b.key(); var rf = cx.data[st]; // Cascade: Instance the top objects: Instance0b(cx, defpos, 0); Instance0b(cx, st, 0); Instance85(cx, st, 0);</pre>		Search Solution Explor P b < C ^a Trigger.cs b < C ^a User.cs b < C ^a Value.cs b < C ^a Value.cs c = Velve.cs c = Velve.cs c = Activatior b < C ^a Context.c
165 166 167 168 169 170 171 172 173	<pre>rt += b.key(); var rf = cx.dat6jt]; // Cascade: Instance the top objects: InstanceOb(cx, defpos, 0); InstanceOb(cx, st, 0); InstanceOb(cx, st, 0); // Cascade has finished. Fix() everything.</pre>		Search Solution Explor P
165 166 167 168 169 170 171 172 173 174	<pre>rt += b.key(); var rf = cx.data[st]; // Cascade: Instance the top objects: Instance0b(cx, defpos, 0); Instance0b(cx, st, 0); Instance85(cx, st, 0);</pre>		Search Solution Explor P b < C ^a Trigger.cs b < C ^a User.cs b < C ^a Value.cs b < C ^a Value.cs c = Vew.cs c = Vew.cs c = Activatior b < C ^a Context.c
165 166 167 168 169 170 171 172 173 174 175	<pre>rt += b.key(); // Cascade: Instance the top objects: InstanceOb(cx, defpos, 0); InstanceOb(cx, st, 0); InstanceOb(cx, st, 0); // Cascade has finished. Fix() everything. cx.FixAll(defpos); // All done, so finish up</pre>		Search Solution Explor P b \checkmark C= Trigger.cs b \checkmark C= User.cs b \checkmark C= Value.cs b \checkmark C= Value.cs c= Value.cs c= Value.cs c= Certext.c b \checkmark C= Context.c b \checkmark C= Group.cs
165 166 167 168 169 170 171 172 173 174 175 176	<pre>rt += b.key(); var rf = cx.data[st]; // Cascade: Instance the top objects: Instance0b(cx, defpos, 0); Instance0b(cx, st, 0); Instance0s(cx, st, 0); // Cascade has finished. Fix() everything. cx.FixAll(defpos);</pre>		Search Solution Explor P b < C* Trigger.cs b < C* User.cs b < C* Value.cs b < C* View.cs c = Level4 b < C* Activation b < C* Activation b < C* Group.cs b < C* Join.cs
165 166 167 168 169 170 171 172 173 174 175 176 177	<pre>rt += b.key(); var rf = cx.data[st]; // Cascade: Instance the top objects: InstanceOb(cx, st, 0); InstanceOb(cx, st, 0); // Cascade has finished. Fix() everything. cx.Fixwl1(defpos); // All done, so finish up var up = cx.rsuids[rf.defpos].value; var = cx.dstd[rf.defpos].value;</pre>		Search Solution Explor P
165 166 167 168 169 170 171 172 173 174 175 176 177 178	<pre>rt += b.key(); var rf = cx.data[st]; // Cascade: Instance the top objects: Instance0b(cx, defpos, 0); Instance0b(cx, st, 0); // Cascade has finished. Fix() everything. cx.FixAll(defpos); // All done, so finish up var vp = cx.rsuid[rf.defpos].Value;</pre>		Search Solution Explor P
165 . 166 . 167 . 168 . 170 . 171 . 172 . 173 . 174 . 175 . 176 . 177 . 178 . 179 .	<pre>rt += b.key(); var rf = cx.data[st]; // Cascade: Instance the top objects: Instance0b(cx, defpos, 0); Instance0b(cx, st, 0); // Cascade has finished. Fix() everything. cx.Fixall(defpos); // All done, so finish up var vp = cx.rsuids[rf.defpos].value; var r = cx.data[vp]; r += (uery.Where, f.where);</pre>		Search Solution Explor D
165 166 167 167 168 169 170 171 172 173 174 175 175 176 177 178 179 180	<pre>rt += b.key(); var rf = cx.data[st]; // Cascade: Instance the top objects: InstanceOb(cx, defpos, 0); InstanceOb(cx, st, 0); // Cascade has finished. Fix() everything. cx.Fixal1(defpos); // All done, so finish up var vp = cx.rsuids[rf.defpos].value; var r = cx.data[vp]; r += (Query.Where, f.where); r += (Query.Wheres, f.matches);</pre>		Search Solution Explor P
165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 181	<pre>rt += b.key(); var rf = cx.datGst]; // Cascade: Instance the top objects: InstanceOb(cx, defpos, 0); InstanceOb(cx, st, 0); // Cascade has finished. Fix() everything. cx.Fixall(defpos); // All done, so finish up var vp = cx.rsuids[rf.defpos].Value; var r = cx.data[vp]; r = (Query.Herce f.where); r += (Query.Jatches, f.matches); r += (Query.Jatches, f.matches); r += (Query.Jatches, f.matches); r = the table table</pre>		Search Solution Explor P

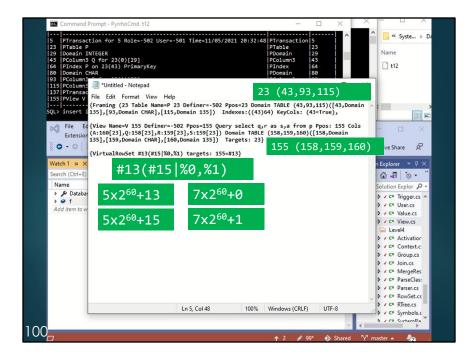
We are going to use the View definition to insert rows into the table P. This is obviously not what views are normally used for, but in this demonstration, we show that we have updatable views. Views can be used, provided the permissions are set correctly, to make modifications to the underlying tables.



When we click Enter for this insert statement, Visual Studio stops at the break point. The Instance() method has a RowSet parameter RowSet f (its uid identifies the position of V in the command): it is a VirtualRowSet.



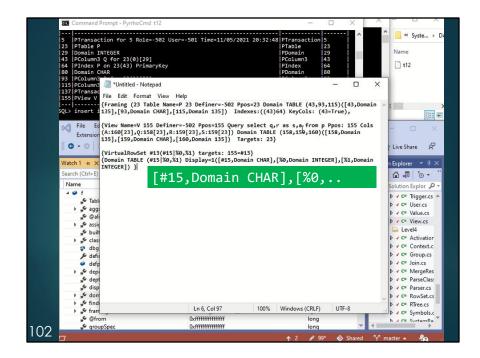
Use the Watch window to examine f. Right-click and copy its value into our Notepad.



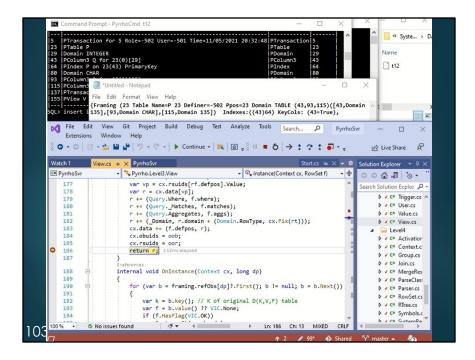
This domain seems different again, and the notation needs explanation. As mentioned, #13 is V's position in the command, and #15 S's position. These notations stand for long integers above 2^60 0x5000000000000 and 0x50000000000000 respectively, and %0 and %1 are 0x70000000000000 and 0x70000000000001 respectively. (We will soon see !0 which is 0x4000000000000.) These are long integers unlikely to clash with file positions.

Command Prompt - PyrrhoCmd t			
5 PTransaction for 5 Role=-5	502 User=-501 Time=11/05/2021 20:32:48 P		Syste >
23 PTable P	P	Table 23	
29 Domain INTEGER		Domain 29	Name
<pre>43 PColumn3 Q for 23(0)[29] 64 PIndex P on 23(43) Primary</pre>		Column3 43 Index 64	1 t12
80 Domain CHAR		Domain 80	
93 PColumn3 C C 03(c) coos			
115 PColumn3 🦾 *Untitled - Notepa	ed 🔉	- 0	×
137 PTransac 155 PView V File Edit Format	View Help		
	e Name=P 23 Definer=-502 Ppos=23 Domain	TABLE (43.93.115)([43.Dor	ain Ar
	HAR],[115,Domain 135]) Indexes:((43)64)		
M File Edit View Git Pro	viect Build Debug Test Analyze Tools		
	ject build Debug lest Analyze loois	Search P Pyrrhos	Svr — 🗆 🗙
Extensions Window Help			
8 G • O 1 🕄 • 🏠 💾 💾 🤊	- 🤆 - 🕨 Continue - 🚉 🔯 🖉 🖬 🔳 🌘	5 → * ? * ₽	IR Live Share
	· · · · · · · · · · ·		
Watch 1 🕫 🗙 View.cs Pyrrh	oSvr	Start.cs 🖮 🗙 👻 🏶	Solution Explorer 📼 🕂 🛪
Search (Ctrl+E)			· · · ·
Name	Value	Type	Search Solution Explor
	(VirtualRowSet #13(#15)%0 %1) targets:		
4 0 1	{VirtualRowSet #13(#15 %0,%1) targets: false		♦ ✓ C= Trigger.cs ✓
TableColsOk	false	155=# Pyrrho.Level4.Ro bool	▷ ✓ C= Trigger.cs ▷ ✓ C= User.cs
TableColsOk		155=# Pyrrho.Level4.Ro bool Pyrrho.Common	 ↓ C= Trigger.cs ↓ C= User.cs ↓ C= Value.cs
↓ ↓ TableColsOk ▶ ↓ aggs ↓ @ alias	false {}	155=# Pyrrho.Level4.Ro bool Pyrrho.Common string	 ↓ C* Trigger.cs ↓ ↓ C* User.cs ↓ C* Value.cs ↓ C* View.cs
TableColsOk	false 0 null	155=# Pyrrho.Level4.Ro bool Pyrrho.Common	 ▷ ✓ C* Trigger.cs ▷ ✓ C* User.cs ▷ ✓ C* Value.cs ▷ ✓ C* View.cs ▲ Level4
→ ↓ ↓ TableColsOk → ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	false O null O	155=# Pyrrho.Level4.Ro bool Pyrrho.Common string Pyrrho.Common	 ↓ C* Trigger.cs ↓ ↓ C* User.cs ↓ C* Value.cs ↓ C* View.cs
↓ ↓ TableColsOk	false () null () false	155=# Pyrrho.Level4.Ro bool Pyrrho.Common string Pyrrho.Common bool Pyrrho.Level3.Level	 ▷ ✓ C* Trigger.cs ▷ ✓ C* User.cs ▷ ✓ C* Value.cs ▷ ✓ C* View.cs ▲ Level4
TableColsOk	false O null O false O	155=# Pyrrho.Level4.Ro bool Pyrrho.Common string Pyrrho.Common bool	 ▷ ✓ C* Trigger.cs ▷ ✓ C* User.cs ▷ ✓ C* Value.cs ▷ ✓ C* View.cs ▲ C* Level4 ▷ ✓ C* Activation
TableColsOk > 5 aggs 5 agg 5 agg 6 agg	false () null () false () 0xx000000000152a	155=# Pyrrho.Level4.Ro bool Pyrrho.Common string Pyrrho.Common bool Pyrrho.Level3.Level long	 ▷ < C* Trigger.cs ▷ < C* User.cs ▷ < C* Value.cs ▷ < C* View.cs ▷ < C* Level4 ▷ < C* Activation ▷ < C* Context.c
TableColsOk ▶ ⊈ aggs ⊈ @alias ▶ ⊈ assig ⊈ built ▶ ⊈ classification ∉ dbg ≠ definer	false () null () false () 0.0000000000152a 0.000000000152a 0.00000000000152a 0.00000000000000000000000000000000000	155=# Pyrrho.Level4.Ro bool Pyrrho.Common string Pyrrho.Common bool Pyrrho.Level3.Level long	 C Trigger.cs C User.cs C User.cs C View.cs C View.cs Level4 C Activation C Context.c C Group.cs
TableColsOk → 5 aggs 5 @alias → 6 walias → 7 wali	false 0 null 0 false 0 0 0 false 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	155=# Pyrrho.Level4.Ro Pyrrho.Loromon string Pyrrho.Common bool Pyrrho.Level3.Level long long long	 C Trigger.cs C User.cs C User.cs C Vieu.cs C Evel4 C Activation C Context.c C Context.c C Group.cs C C Join.cs
TableColsOk → 5 TableColsOk → 5 aggs 5 @alias → 6 waisg 5 built → 6 classification 6 dbg 5 definer 6 defpoos 5 \$ dependents 5 \$ depth	false () null () false () 0.00000000000152a 0.0000000000052a 0.00000000000000000000000000000000000	155=# Pyrrho.Level4.Ro bool Pyrrho.Common string Pyrrho.Common bool Pyrrho.Level3.Level long long long Pyrrho.Common	 C° Trigger.cs C° User.cs C° Value.cs C° Vervet Evel4 C° Context.cc C° Context.cc C° Context.cc C° Context.cs C° Group.cs C° Context.cs C° Context.cs C° Context.cs C° Context.cs C° Group.cs C° Context.cs C° C° Context.cs C° C°
TableColsOk Sequence aggs Calless Sequence aggs Calless Sequence aggs Sequence aggs Seque	false () null () false () 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	155=# Pyrrho.Level4.Ro bool Pyrrho.Common string Pyrrho.Common bool Pyrrho.Level3.Level long long long Pyrrho.Common int it	 C° Triggercs 4 C° Usercs C° Value.cs C° Value.cs C° Vew.cs C° Context.c C° Context.c C° Context.c C° Group.cs C° MergeRes C° C° MergeRes C° C° MergeRes
TableColsOk → 5 TableColsOk → 5 aggs 5 @alias → 6 waisg 5 built → 6 classification 6 dbg 5 definer 6 defpoos 5 \$ dependents 5 \$ depth	faise () null () faise () () () faise () () () 0x00000000000152a () 0x500000000000000000000000000000000000	155=# Pyrrho.Level4.Ro bool Pyrrho.Common bool Pyrrho.Common bool Pyrrho.Level3.Level long long long pyrrho.Common int int (#15,0	 > < C* Trigger.cs > < C* User.cs > < C* Value.cs > < C* Value.cs > < C* Value.cs > < C* Context.cc > < C* Context.cc > < C* Group.cs > < C* Group.cs > < C* ParseClass > < C* ParseClass > < C* ParseClass > < C* CaveStart > < C* C* CaveStart > < C* C* CaveStart > < C* C*
TableColsOk TableColsOk Galias Gali	false () null () false () 0.00000000000152a 0.00000000000152a 0.00000000000000000000000000000000000	155=# Pyrrho.Level4.Ro bool Pyrrho.Common bool Pyrrho.Common bool Pyrrho.Level3.Level long long long pyrrho.Common int int (#15,0	 > < C* Trigger.cs - > < C* User.cs > < C* Value.cs > < C* Value.cs > < C* Value.cs > < C* Verw.cs > < C* Activation > < C* Activation > < C* Context.c > < C* Group.cs > < C* Group.cs > < C* Group.cs > < C* ParseClass > < C* ParseClass > < C* ParseClass > < C* Rifee.cs > < C* Rifee.cs > < C* Rifee.cs
TableColsOk > 5 aggs 5 @alias > 6 assig 5 built > 6 classification addg 6 defnos > 6 defnos > 6 dependents 5 depth 5	faise () null () faise () () () faise () () () 0x00000000000152a () 0x500000000000000000000000000000000000	155=# Pyrrho.Level4.Ro bool Pyrrho.Common byrho.Common byrho.Level3.Level long long Pyrrho.Common int ([#15.D Pyrrho.Level3.Do 51]) Pyrrho.Level3.Do	 > < C* Trigger.ci - > < C* User.cs > < C* Value.cs > < C* Value.cs > < C* Value.cs > < C* Context.c > < C* Activation > < C* Context.c > < C* Group.cs > < <
TableColsOk TableColsOk Sequence aggs Calless Sequence aggs Sequence aggs S	false () null () false () Cxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	155=# Pyrrho.Level4.Ro bool Pyrrho.Common string Pyrrho.Common bool Pyrrho.Level3.Level long long long Pyrrho.Common int int (£15,D] Pyrrho.Level3.Do Pyrrho.Level4.Fra Pyrrho.Level4.Fra	 > < C* Trigger.cs - > < C* User.cs > < C* Value.cs > < C* Value.cs > < C* Value.cs > < C* Verw.cs > < C* Activation > < C* Activation > < C* Context.c > < C* Group.cs > < C* Group.cs > < C* Group.cs > < C* ParseClass > < C* ParseClass > < C* ParseClass > < C* Rifee.cs > < C* Rifee.cs > < C* Rifee.cs

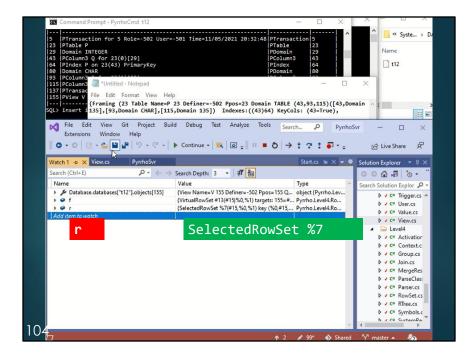
In the Watch window, expand f to get a closer look at the domain. Paste its value into the Notepad.



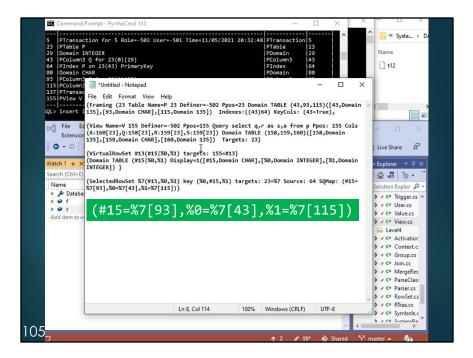
We notice that not only are the uids different, they are also in a different order. Continue to the next break point.



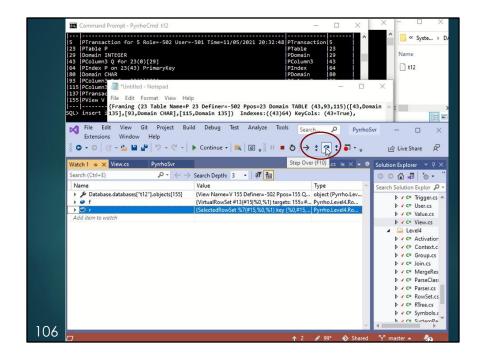
We are at the end of the Instance() method. Now examine the return value from the Instance() method in the Watch window.



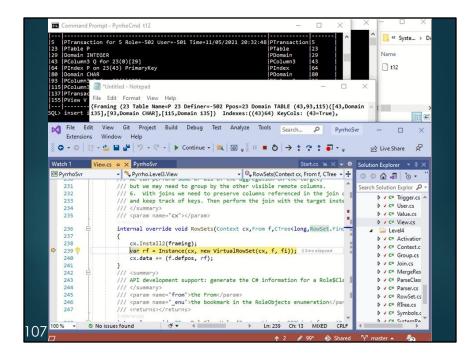
r is a SelectedRowSet. Paste its value into the Notepad.



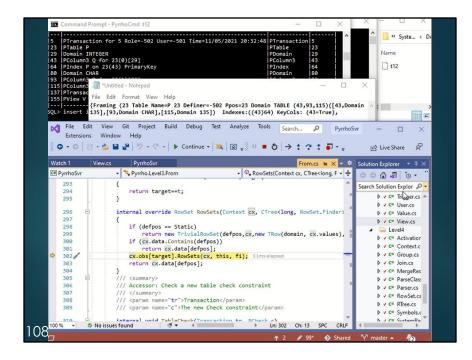
This completes the analysis: We can now see that the SelectedRowSet has a map taking the uids %0, #15, %1 to the table columns 43, 93, 115 of P (not V). As its name implies, SelectedRowSet is normally used for retrieving data but we can use it for insert, update and delete also.



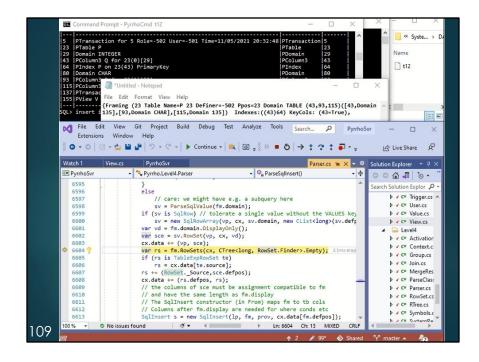
Back in Visual Studio, Step Over



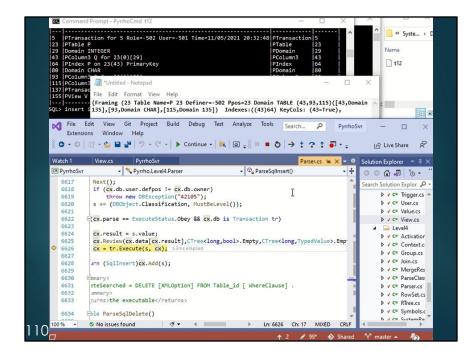
We see that Instance() was called from View.RowSets(). Step Over



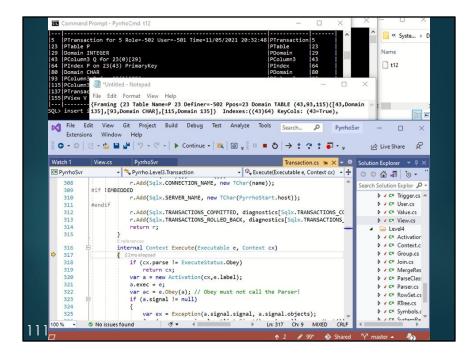
We are now back in From.RowSets(). Step Over



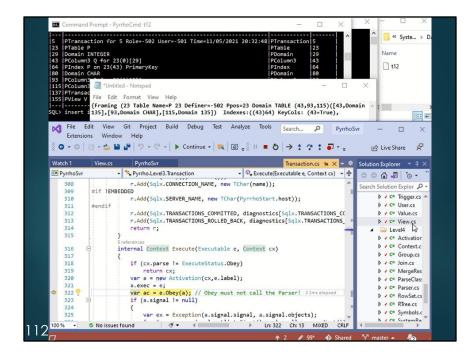
This takes us back to Parser.ParseSqlInsert. Step Over some more until line 6626.



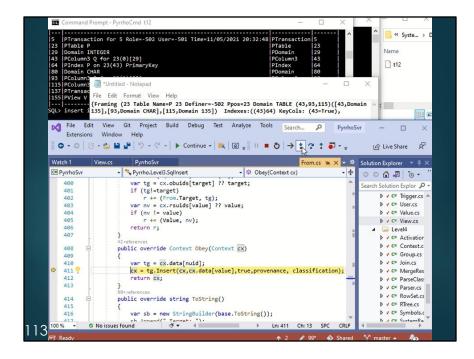
At line 66626, Step Into Transaction.Execute()



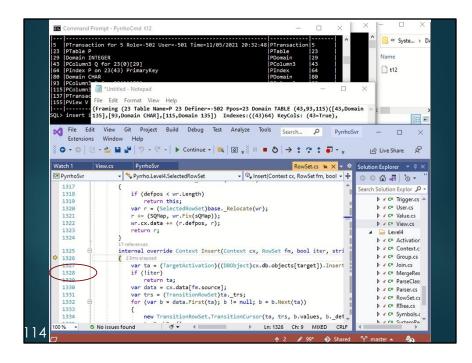
In Transaction.Execute(), Step Over until line 322.



At line 322, Step Into Obey()

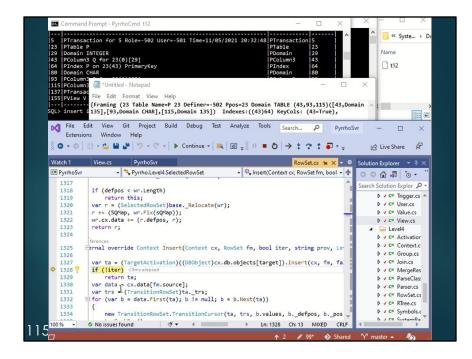


In Sqlinsert.Obey, step into Insert()

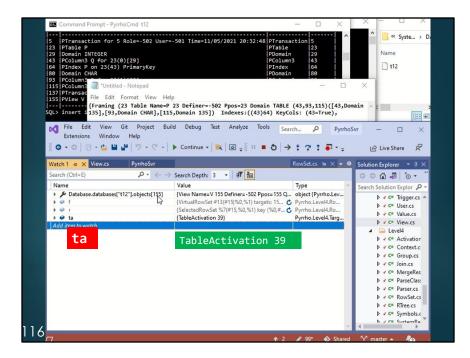


This is where the work is done.

Inserting into a base table often involves Triggers. Triggers are managed by Activations. Step Over the creation of the TargetActivation, to line 1328.



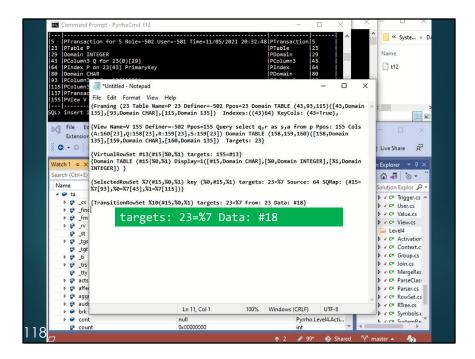
At line 1328, use the Watch window to examine ta



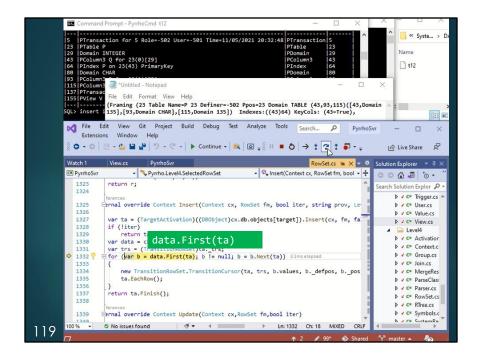
The source code said this would be a TargetActivation. It is a subclass, TableActivation. Expand it.

23 PTable P 29 Domain INTEGER 43 PColumn3 Q for 2		PTable PDomain PColumn3	23 29 43	Name
64 PIndex P on 23(4 80 Domain CHAR	3) PrimaryKey		64 80	🗋 t12
93 PColumn3	SV(4) 5003	Poolaiti		
	itled - Notepad		- 0	×
137 PTransac 155 PView V File Edit	t Format View Help			
{Framing	g (23 Table Name=P 23 Definer=-502			ain A :
SQL> insert i135],[93	3,Domain CHAR],[115,Domain 135])	Indexes:((43)64) KeyCols: (4	3=True),	811
M File Edit View	Git Project Build Debug Test	Analyze Tools Casesh		
		Analyze loois Search	Pyrrhos	Svr — 🗆 🗙
Extensions Windo				
🖞 🖸 🔹 🔘 👘 🖓 – 😂 🖡	💾 🔐 🐬 - 🥂 - 🕨 Continue - 💼		1 🗸 -	Live Share
Watch 1 ↔ X View.cs	PyrrhoSvr	RowSet	.cs 🛎 🗙 👻 🌣	Solution Explorer 🛛 🔻 🖡
Search (Ctrl+E)	$P \star \leftarrow \rightarrow$ Search Depth: 3	- A B		000.00
Name	Value	Type		Search Solution Explor
Name	Value {TableActivation 39	Type }	.Level4.Targ	Search Solution Explor
- Commenter		} Pyrrho	Level4.Targ	Þ ✓ Cª Trigger.cs
🔺 🤪 ta	{TableActivation 39 {Activation 38}	} Pyrrho	.Level4.Con	 ↓ C[#] Trigger.cs ↓ C[#] User.cs
 ✓ Image: a finite and a finite	{TableActivation 39 {Activation 38} {(43= %10[#15],93= {SelectedRowSet %	Pyrrho Pyrrho %10[%0],115=%10[%1],#1 Pyrrho 7(#15,%0,%1) key (%0,#15, Pyrrho	.Level4.Con .Common .Level4.Ro	 ↓ C[#] Trigger.cs ↓ C[#] User.cs ↓ C[#] Value.cs
 ✓ ● ta ▶ ● _cx ▶ ● _finder ▶ ● _fm ▶ ● _rv 	{TableActivation 39 {Activation 38} {(43=%10(#15),93= {SelectedRowSet % null	 Pyrrho Pyrrho %10[%0],115=%10[%1],#1 Pyrrho 7(#15,%0,%1) key (%0,#15, Pyrrho Pyrrho 	Level4.Con .Common .Level4.Ro .Common	 ↓ C[#] Trigger.cs ↓ C[#] User.cs ↓ C[#] Value.cs ↓ C[#] View.cs
 ✓ ♥ ta ▶ ♥ _cx ▶ ♥ _finder ▶ ♥ _fm ▶ ♥ _rv ♥ _st 	{TableActivation 35 {Activation 38} {(43=%10(=15),93= {SelectedRowSet % null	Pyrrho Pyrrho %10[%0],115=%10[%1],#1 Pyrrho 7(#15,%0,%1) key (%0,#15, Pyrrho Pyrrho Pyrrho	.Level4.Con .Common .Level4.Ro .Common .Common	 ↓ ✓ Cⁱⁱ Trigger.cs ↓ ✓ Cⁱⁱ User.cs ↓ ✓ Cⁱⁱ Value.cs ↓ ✓ Cⁱⁱ View.cs ↓ ⊆ Level4
▲	{TableActivation 36 {Activation 38} {(43 = %10 = 15),93 = {SelectedRowSet % null null {}	i} Pyrrho 8/10[%0],115=%10[%1],#1 Pyrrho 7/(#15,%0,%1) key (%0,#15, Pyrrho Pyrrho Pyrrho Pyrrho Pyrrho	Level4.Con .Common .Level4.Ro .Common	 ▷ ✓ C^{III} Trigger.cs ▷ ✓ C^{III} User.cs ▷ ✓ C^{III} Value.cs ▷ ✓ C^{III} Value.cs ▷ ✓ C^{III} Value.cs △ C^{III} Value.cs △ C^{III} Activation
	(TableActivation 35 (Activation 38) ((43-%)(1915))33 (SelectedRowSet % null null () 0.0000000000000000000000000000000000	 Pyrrho Pyrrho Pyrrho Pyrrho Ti5=%10[%1],#1 Pyrrho Pyrrho Pyrrho Pyrrho Pyrrho Pyrrho Pyrrho Pyrrho In ang 	Level4.Con Common Level4.Ro Common Common	 ↓ C* Trigger.cs ↓ C* User.cs ↓ C* Value.cs ↓ C* View.cs ▲ C* View.cs ▲ Level4 ↓ C* Certivation ↓ C* Context.c
▲	(TableActivation 35 (Activation 38) ((43 - % (10 = 15) 33 (SelectedRowSet % null null {) 0x0000000000000000000000000000000000	I) Pyrrho Pyrrho 9yrho 7(#15,%0,%1) key (%0,#15, Pyrrho Pyrrho Pyrrho Pyrrho 17 3 P Domain TABLE (43,93,1 Pyrrho	Level4.Con .Common .Level4.Ro .Common .Common .Common	 ▷ ✓ C^{III} Trigger.cs ▷ ✓ C^{III} User.cs ▷ ✓ C^{III} Value.cs ▷ ✓ C^{III} Value.cs ▷ ✓ C^{III} Value.cs △ C^{III} Value.cs △ C^{III} Activation
	(TableActivation 35 (Activation 38) ((43 - % (10 = 15) 33 (SelectedRowSet % null null {) 0x0000000000000000000000000000000000	 Pyrrho Pyrho Pyrho	Level4.Con .Common .Level4.Ro .Common .Common .Common	 ↓ < C Trigger.cs ↓ < C User.cs ↓ < C Value.cs ↓ < C Activation ↓ < C Activation ↓ < C Group.cs
	(TableActivation 35 (Activation 38) ((43-%)(01=5),93= (5electedRowSet % null 0) 0:00000000000000000000000000000000	1) Pyrrho %10(%0),115=%10(%1),#1 Pyrrho %(10(%0),115=%10(%1),#1 Pyrrho Pyrrho Pyrrho Pyrrho 17 13 P Domain TABLE (43,93, Pyrrho %10(#15,%0,%1) targetz 23 Pyrrho Pyrrho Pyrrho	Level4.Con .Common .Level4.Ro .Common .Common .Common .Level3.Obl .Level4.Ro	 ↓ < C* Trigger.cs ↓ < C* User.cs ↓ < C* User.cs ↓ < C* Vsev.cs ↓ < C* Vsev.cs ▲ Level4 ↓ < C* Activation ↓ < C* Context.c ↓ < C* Group.cs ↓ < C* Join.cs
	(TableActivation 35 (Activation 38) ((43-%)(01=5),93= (5electedRowSet % null 0) 0:00000000000000000000000000000000	 Pyrrho Pyrho Pyrho	Level4.Con .Common .Level4.Ro .Common .Common .Common .Level3.Obl .Level4.Ro	 ▷ < C* Trigger.cs ▷ < C* User.cs ▷ < C* Vseu.cs ▷ < C* View.cs △ C* View.cs △ C* Context.c ▷ < C* Context.c ▷ < C* Context.cs ▷ < C* Join.cs ▷ < C* MergeRes
	(TableActivation 35 (Activation 38) ((43-%)(01=5),93= (5electedRowSet % null 0) 0:00000000000000000000000000000000	b) Pyrtho %10(%0),115=%10(%1),#1 Pyrtho %10(%0),115=%10(%1),#1 Pyrtho 7(#15,%0,%1) key (%0,#15, Pyrtho Pyrtho Pyrtho 13 P Domain TABLE (43,93, Pyrtho %10(#15,%0,%1) targetz 23 Pyrtho Pyrtho Pyrtho Sin Domain TABLE (43,93, Pyrtho Pyrtho Pyrtho Sin Domain TABLE (43,93, Pyrtho Sin Domain TABLE (43,93, Pyrtho	Level4.Con Common Common Common Common Common Level3.Obl Level4.Ro Level2.PTri	 ↓ < C* Trigger.cs ↓ < C* User.cs ↓ < C* Value.cs ↓ < C* View.cs ↓ < C* Activation ↓ < C* Activation ↓ < C* Context.c ↓ < C* Group.cs ↓ < C* MergeRes ↓ < C* MergeRes ↓ < C* ParseClass
	(TableActivation 35 (Activation 33) ((33-%)(1915)33- (SelectedRowSet % null null 0 0:0000000000000 (Oblinfo Namee P 2 Ottinston RowSet Ctrl+C Ctrl+C Ctrl+V Transit	1) Pyrtho	Level4.Con .Common .Level4.Ro .Common .Common .Common .Common .Level3.Pfri lowen .common .Common .Common	
	(TableActivation 35 (Activation 38) ((43-%)(01=5),93= (5electedRowSet % null 0) 0:00000000000000000000000000000000	i) Pyrrho %10(%0,115=%10(%1,1#1 Pyrrho %10(%0,115=%10(%1,1#1 Pyrrho /7(#15,%0,%1) key (%0,#15 Pyrrho Pyrrho Pyrrho 17 Iong 18 P Domain TABLE (43,93,1 Pyrrho \$10(#15,%0,%1) targets: 23 Pyrrho Pyrrho Pyrrho Pyrrho Pyrrho Pyrrho Pyrrho	Level4.Con .Common .Level4.Ro .Common .Common .Common .Level3.Obl .Level3.PTri tommon tommon .Common	

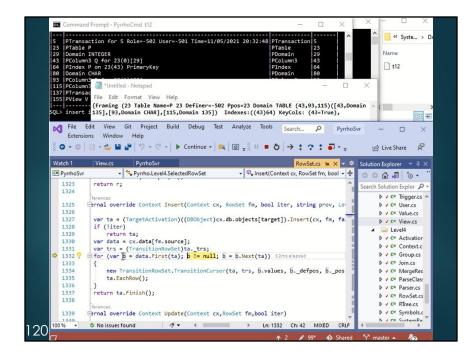
The TableActivation uses a thing called a TransitionRowSet, a concept defined in the SQL standard. Copy its value into our Notepad.



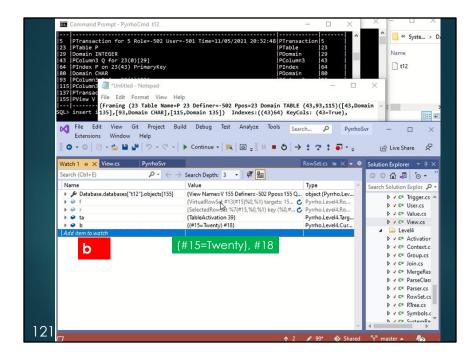
We see the TransitionRowSet knows it has Data, VALUES is at position 18 of the INSERT statement.



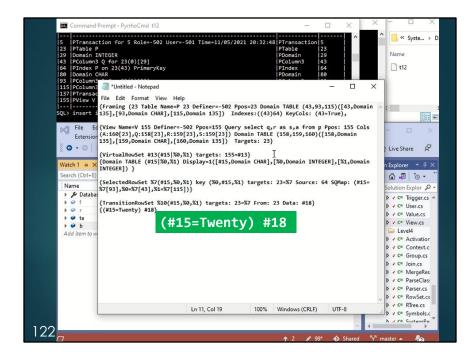
Back in the Insert method, start traversal of the data. Step Over the call to First().



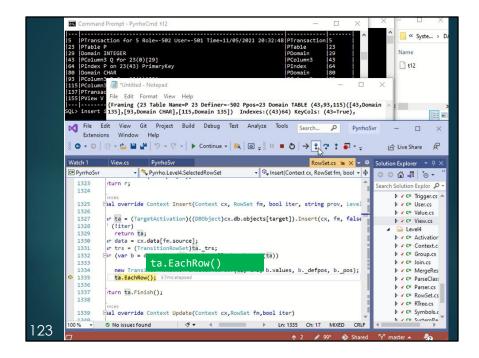
Now use the Watch window to examine Cursor b



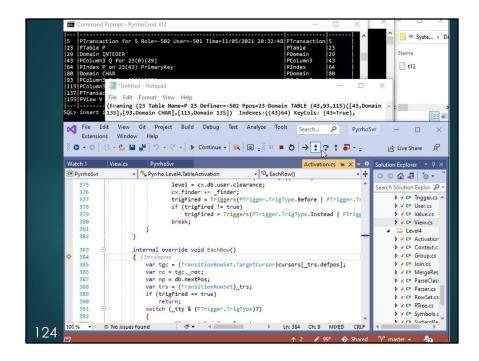
Paste the value of the cursor into the Notepad



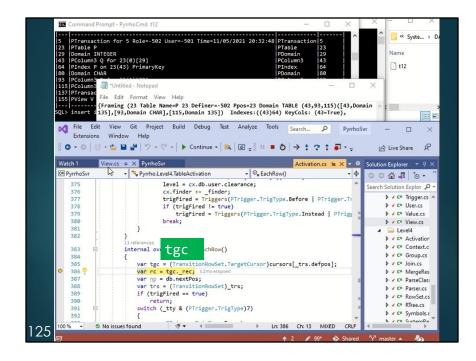
We recognise this value from the INSERT command. Step Over



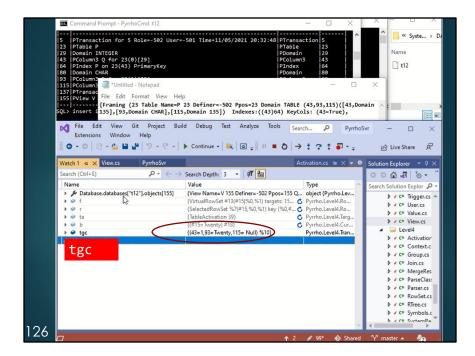
Now Step Into EachRow()



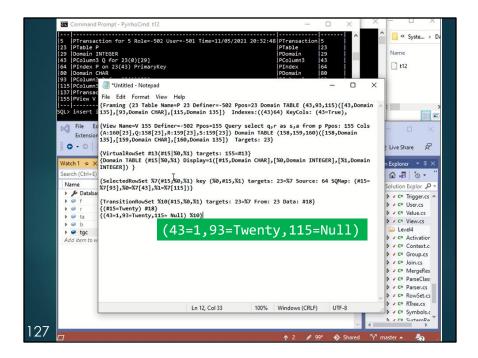
In TableActivation.EachRow(), Step Over twice



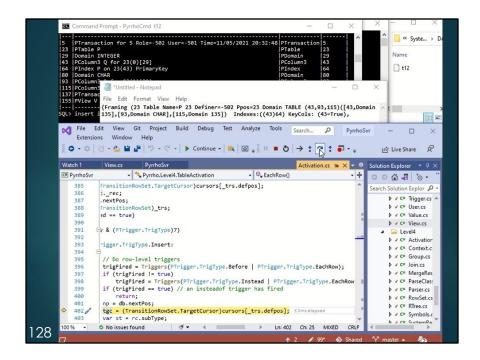
At line 386, use the Watch window to examine the TargetCursor tgc



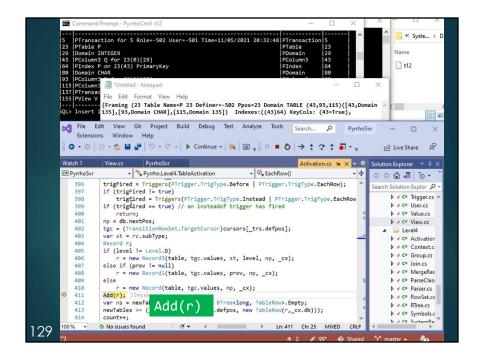
Paste the value of tgc into the Notepad



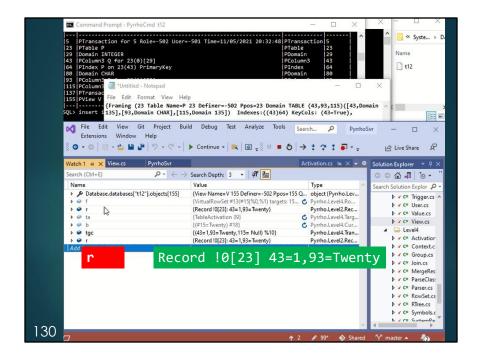
Here we see that Pyrrho's autokey feature has filled in a value for the key column 43.



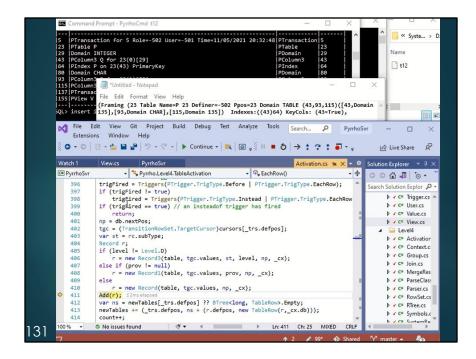
Step Over a bit more. We have no triggers in this example, so after refreshing the value of the target cursor, Pyrrho constructs a record. We will stop at line 411.



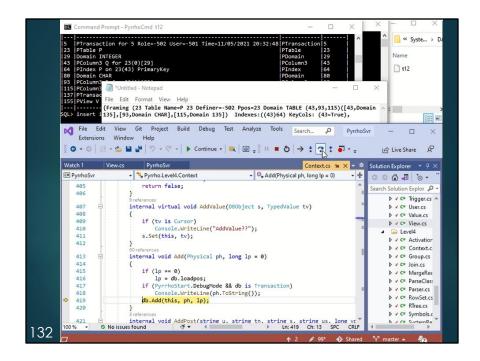
At line 411. A Record r has just been constructed.



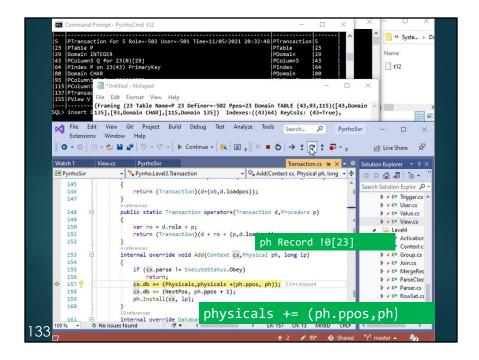
Use the Watch window to examine r. From now on it is all about installing this record in the Transaction.



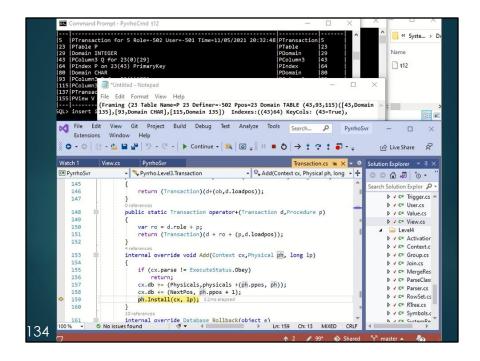
Back in Visual Studio, Step Into Add(r).



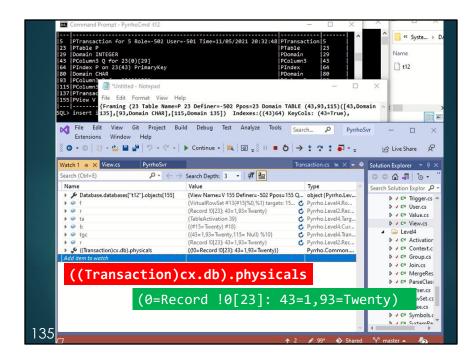
In Context.Add(), step over a few times, then step Into db.Add() to add our Record to the Transaction.



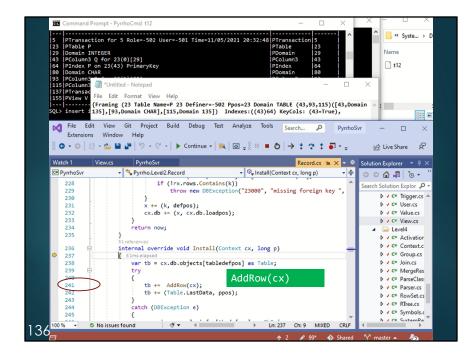
In Transaction.Add, we see how the Physical is added to the physicals list. Step Over line 158.



At line 159, we are about to call ph.Install(). But let's use the Watch window to examine physicals



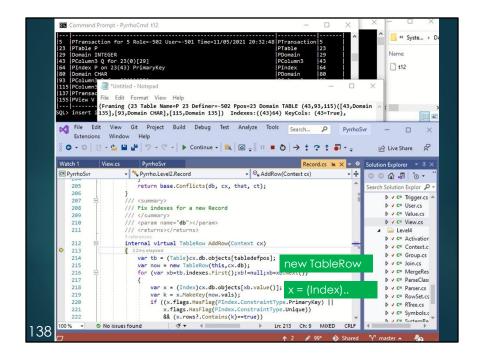
It is a list containing the new Record. Step into ph.Install()



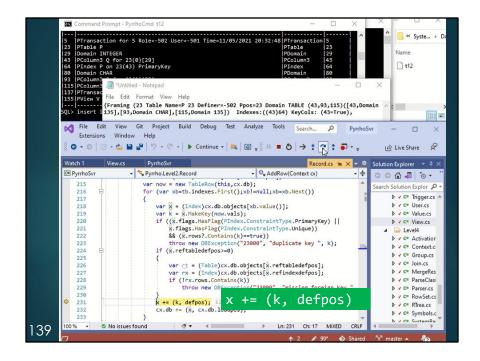
In Record.Install(), Step over to line 241

	n for 5 Role=-502 User=-501 Time=11/05/		n 5	 System
23 PTable P 29 Domain INTE	SER	PTable PDomain	23	Name
43 PColumn3 Q	for 23(0)[29]	PColumn3	43	
64 PIndex P on 80 Domain CHAR	23(43) PrimaryKey	PIndex PDomain	64 80	🗋 t12
93 PColumn3	5 00 (A) 5003	PDomain	00	
	*Untitled - Notepad		- 🗆	×
137 PTransac	Edit Format View Help			
	aming (23 Table Name=P 23 Definer=-502	Ppos=23 Domain TABLE (43,	93,115)([43,Doma	in A c
SQL> insert i 135],[93,Domain CHAR],[115,Domain 135]) I	ndexes:((43)64) KeyCols:	(43=True),	
M File Edit	View Git Project Build Debug Test	Analyze Tools Search		
		Analyze loois Search	PyrrhoSv	r — 🗆
	Window Help			
0 · 0 18 ·	🕍 💾 🚰 👘 - 🤆 - 🕨 Continue - 🗮		👎 1 🗗 t 🚽 👘	🖻 Live Share
Watch 1 Vie	w.cs PyrrhoSvr	Step Into (E11)	ord.cs 🖮 🗙 👻 🌣	Solution Explorer
C# PyrrhoSvr	Pyrrho.Level2.Record	• @ Install(Context cx, Ion		
			+ (q p	0 0 1 - 0 0
228	if (!rx.rows.Contains(· · · · · ·	Search Solution Explo
	throw new DBExcept	ion("23000", "missing for	eign key ,	
				▷ ✓ C [#] Trigg
230	} x += (k, defpos);			
230	<pre>} x += (k, defpos); cx.db += (x, cx.db.loadpos</pre>	.):	1	♦ 🗸 C# User.
230 231	<pre>} x += (k, defpos); cx.db += (x, cx.db.loadpos }</pre>	;);	1	♦ ✓ C# User. ♦ ✓ C# Value
230 231 232		;);		♦ ✓ C# User. ♦ ✓ C# Value
230 231 232 233	<pre>cx.db += (x, cx.db.loadpos } return now; }</pre>	;);	1	 ▷ ✓ C# User. ▷ ✓ C# Value ▷ ✓ C# View. ▲ □ Level4
230 231 232 233 234 235	<pre>cx.db += (x, cx.db.loadpos } return now; } Sireferences</pre>			 ↓
230 231 232 233 234 235 236	<pre>cx.db += (x, cx.db.loadpos } return now; }</pre>			 ↓ C# User. ↓ C# Value ↓ C# View ▲ □ Level4 ↓ C# Activ ↓ C# Cont
230 231 232 233 234 235 236 236 237	<pre>cx.db += (x, cx.db.loadpos } return now; } internal override void Install(Cor {</pre>	text cx, long p)	_	 ↓ C# User. ↓ C# Value ↓ C# View ▲ □ Level4 ↓ C# Activ ↓ C# Cont ↓ C# Grout
230 231 232 233 234 235 236 □ 236 □ 237 238	<pre>cx.db += (x, cx.db.loadpos } return now; } Sireferences internal override void Install(Cor { var tb = cx.db.objects[tablede</pre>	text cx, long p)	_	 ↓ c= User. ↓ c= Value ↓ c= View ▲ Level4 ↓ c= Activ ↓ c= Cent ↓ c= Grou ↓ c= Grou ↓ c= Join
230 231 232 233 234 235 236 237 237 238 238 239	<pre>cx.db += (x, cx.db.loadpos } return now; } internal override void Install(Cor {</pre>	text cx, long p)	_	 C = User. C = Value C = Value C = Vel4 C = Activ C = Cont C = Grou C = Grou C = Merg C = Merg
230 231 232 233 234 235 235 235 236 237 238 239 239 240	<pre>cx.db += (x, cx.db.loadpos } return now; } Sireferences internal override void Install(Cor { var tb = cx.db.objects[tablede try { }</pre>	stext cx, long p) fpos] as Table;	-	 ↓ < C^a Activ. ↓ < C^a Contr ↓ < C^a Group ↓ < C^a Join.c ↓ < C^a Merg ↓ < C^a Parse
230 231 232 233 234 235 236 ⊟ 237 238 239 ⊟ 249 241 ♀	<pre>cx.db += (x, cx.db.loadpos) return now;) sindernass internal override void Install(Cor { var tb = cx.db.objects[tablede try { tb += AddRow(cx); sinsele</pre>	<pre>stext cx, long p) ffpos] as Table; pped</pre>	_	 ▷ < C* User. ▷ < C* View. ▷ < C* View. △ Level4 ▷ < C* Cent ▷ < C* Group. ▷ < C* Parse ▷ < C* Parse ▷ < C* Parse
230 231 232 233 234 235 235 235 236 237 238 239 239 240	<pre>cx.db += (x, cx.db.loadpos } return now; } Sireferences internal override void Install(Cor { var tb = cx.db.objects[tablede try { }</pre>	<pre>stext cx, long p) ffpos] as Table; pped</pre>	_	 ▷ < C* User, ▷ < C* Value ▷ < C* Value ▷ < C* Value ▷ < C* Activ ▷ < C* Activ ▷ < C* Grou ▷ < C* Grou ▷ < C* Grou ▷ < C* Parse
230 231 232 233 234 235 236 237 238 239 240 240 241 ♀ 241 ♀ 242	<pre>cx.db += (x, cx.db.loadpos return now; } } sinternal override void Install(Cor { var tb = cx.db.objects[tablede try { tb += AddRow(cx); finsels tb += (Table.LastData, ppc) }</pre>	<pre>stext cx, long p) ffpos] as Table; pped</pre>	_	 ▷ < C* User, ▷ < C* Value ▷ < C* Value ▷ < C* Activ ▷ < C* Cont ▷ < C* Grou ▷ < C* Grou ▷ < C* Grou ▷ < C* Omerg ▷ < C* Parse ▷ < C* Row ▷ < C* Row ▷ < C* Rive
230 231 232 233 234 235 236 237 238 239 ₽ 240 241 242 242 243	<pre>cx.db += (x, cx.db.loadpos) return now;) sindernass internal override void Install(Cor { var tb = cx.db.objects[tablede try { tb += AddRow(cx); sinsele</pre>	<pre>stext cx, long p) ffpos] as Table; pped</pre>	_	 ↓ < C* User. ↓ < C* Value ↓ < C* View ▲ Level4 ↓ < C* Activ ↓ < C* Grou ↓ < C* Grou ↓ < C* Grou ↓ < C* Join ↓ < C* Merg ↓ < C* Merg ↓ < C* Parse ↓ < C* Parse

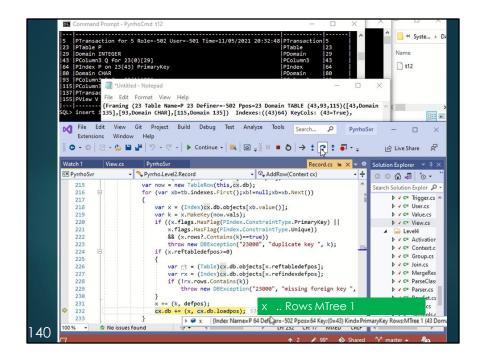
Step Into AddRow().



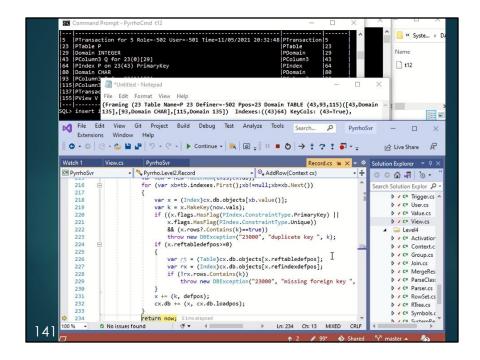
Record.AddRow is all about building a TableRow and adding it to any indexes that are around. Step over down to line 231.



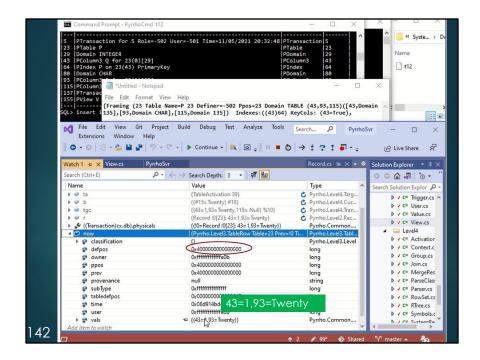
At line 231, we use one of our addition operators to add the key and record position to the Index x.



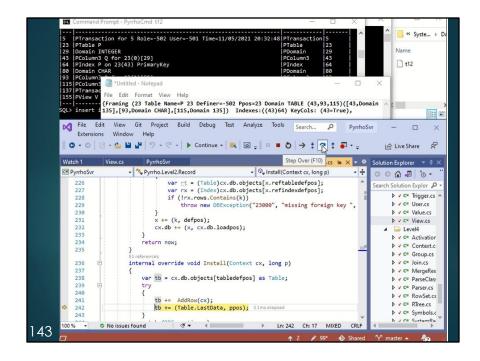
Hovering over x, we see x has a new value and now contains one key. Another addition operator installs the new Index in the transaction.



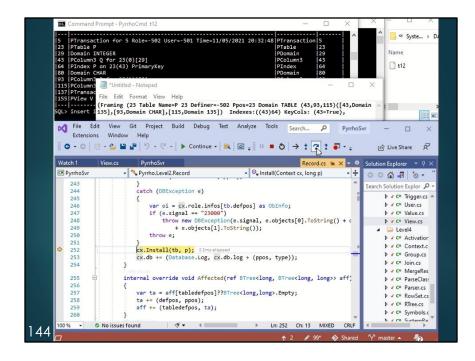
After a journey around the loop we reach the return statement at line 234 with the new TableRow.



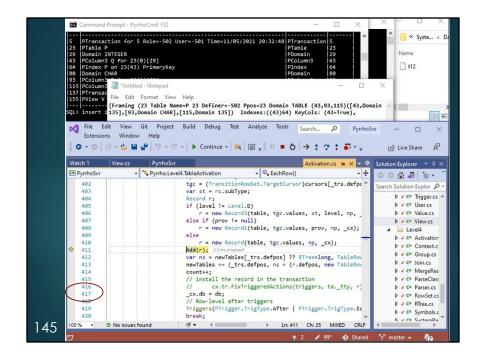
Let's examine now with the Watch window. It has our new values vals. The huge numbers called ppos etc are all !0, the position for the first uncommitted database object in the transaction.



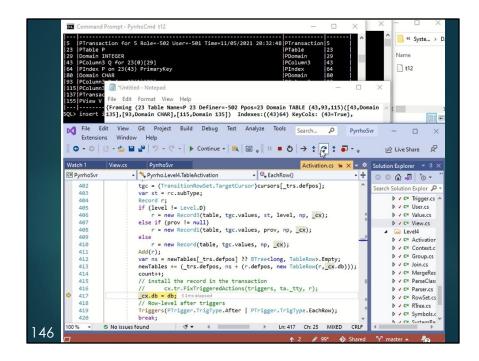
After adding this new row to the table, we add the updated table to the transaction.



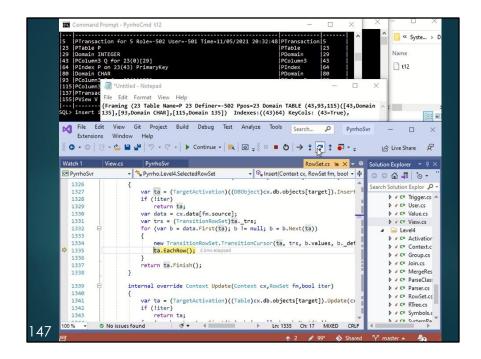
Step Over some more, to get back to EachRow.



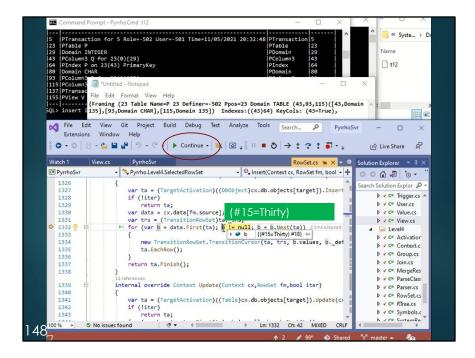
In EachRow() we have just finished Add for our Record. Step over to line 417.



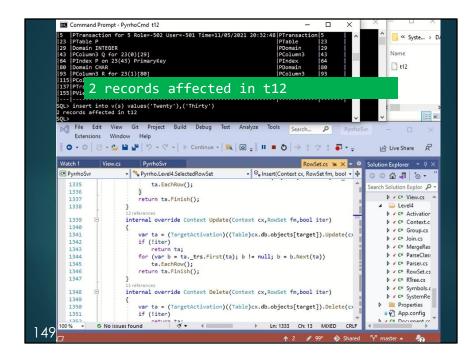
At line 417 we copy our transaction from the TargetActivation into the parent Context. (Just an assignment of a 64-bit pointer.)



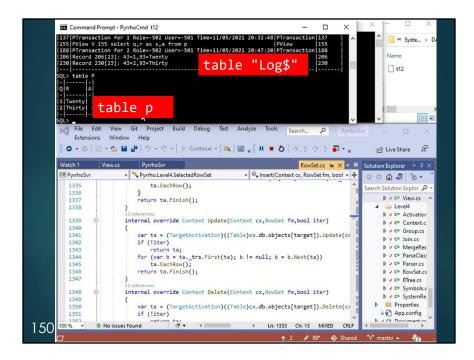
That finishes EachRow for this row of the data. Step Over to see the Cursor for the next row of the data



Hovering over b, we recognise the value Thirty here. The same thing will happen for this row, and then the transaction will Commit(), so just click Continue to finish.

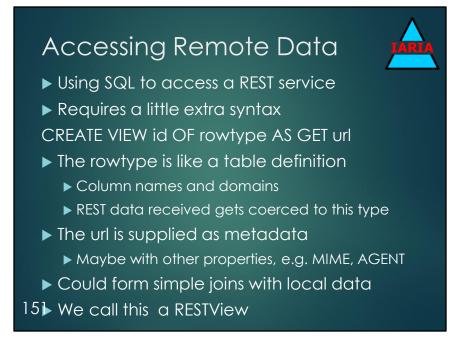


The successful commit is reported to the client.



Confirm the new contents of the table P (this is the real target of the INSERT statement, not V!)

This completes the demonstration.



In the last part of this tutorial we want to demonstrate how these ideas can be extended to remote views (often called View-Mediated data Integration).

If we define a remote view, we need to give the domains of the columns (like in a table definition) that the remote data received from a REST service can be coerced to.

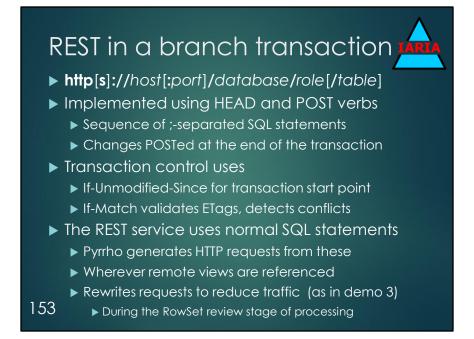
We also need to be told how to access it, e.g. a URL, MIME, locale. This can be done with metadata.

Some extra syntax needs to be added to SQL. We call this RESTView.



Transactions then need some thought

RFC 7232 is helpful, as it explicitly covers the lost update problem It proposes all HTTP1.1 services return entity-tags These can be used for conditional requests We can use these in the RESTView implementation But very few real REST services comply So Pyrrho has a suitable service that does



Pyrrho offers two styles of interaction over HTTP, one using URL-based discrete operations, and another using POST to send a sequence of SQL statements to a remote database.

In both cases any changes should be done as a branch transaction at the commit point of the main transaction. But as we will see, we accumulate information about the steps as we go (like read constraints in demo 2), using the HEAD verb to anticipate the actual change. The information is exchanged using ETags, described next.

The REST service is more natural for an SQL-based client, and the rewriting process can take account of different SQL dialects, with metadata about the view identifying the SQL agent providing the service.

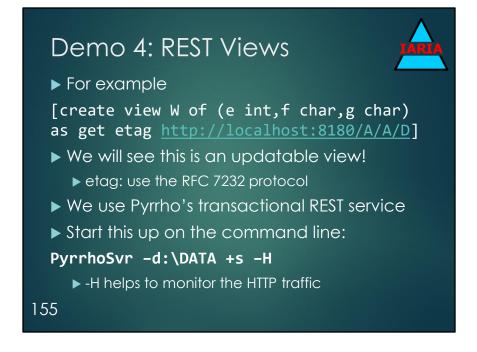


ETags are defined in RFC7232. "*" is an empty ETag, and anything else is opaque, that is, it is generated by the service provider, and not understood by anyone else.

These slides show some simple examples of Pyrrho's ETags. They have the form of a comma-separated list of integers:.

It begins with Table_uid and this is followed by one or more pairs RowDefPos_uid, FilePos_uid.

Thus an ETag can specify (a) a sequence of specific row occurrences, with the defining position and latest position for each, or (b) -1 as a wildcard standing for all rows, and the highwater-mark for the table



In this demonstration we see how the framework developed in this tutorial applies when the data is remote.

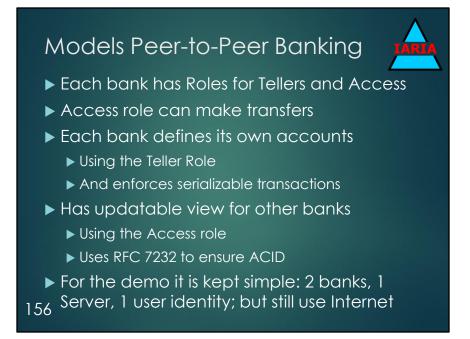
In particular, we demonstrate an updatable RESTView where Pyrrho is used for both server and client (over HTTP, so could be remote).

[create view W of (e int,f char, g

char) as get etag http://localhost:8 180/A/A/D] We will see that this is an updatable view. The etag metadata flag tells Pyrrho to use RFC 7232.

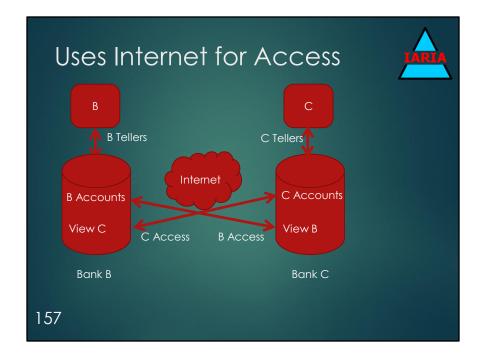
To switch on Pyrrho's REST service, we add the +s flag. It starts an HTTP server (on port 8180 by default).

The –H flag gets diagnostic information in the server window showing HTTP requests and responses.

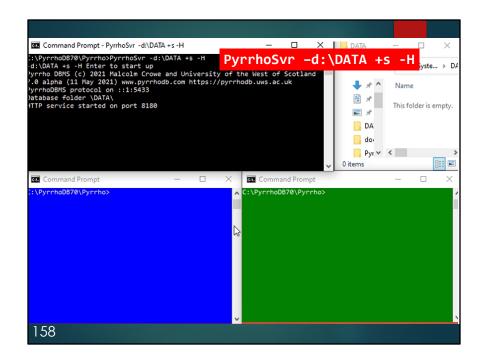


We can think of it as a model of peer-to-peer banking, with roles defined for local accounts and external transfers. Each bank is autonomous, and serializes transactions on their own accounts in the ordinary database way. For remote accounts they use ETags and RFC 7232 conditional requests. There could be dozens of banks, and many users granted these roles.

Users would not actually execute individual SQL statements but use banking applications that manage auditing, journaling, compliance etc.



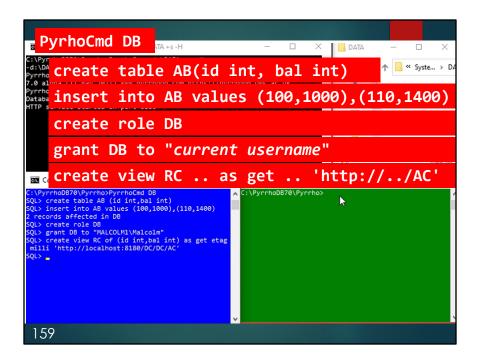
Each bank defines its own accounts, and creates a view of other banks. Views provide only a subset of information (e.g. can validate an account number) and can enable transfers of money in both directions. The diagram shows only a subset of the functionality required of the banking application: just enough to demonstrate the use of the ETag mechanism for transfers between local and remote accounts.



The scenario in this demo is a bank transfer between two banks.

This time the PyrrhoSvr needs to be started up with the command line shown: **PyrrhoSvr –d:\DATA +s –H**

We check the database folder is empty and provide blue and green windows for the two bank clients.

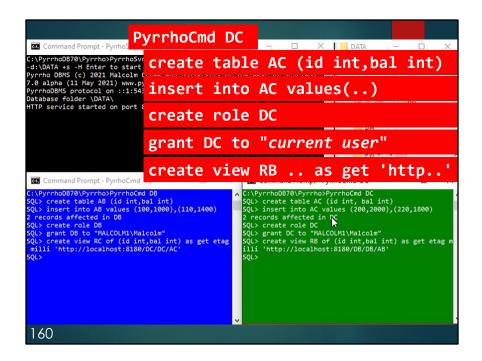


To fit things on the slide we use short identifiers: AB for accounts in bank B, etc. We create an accounts table with two accounts, id 100 with balance 1000, and id 110 with balance 1400.

To make this table accessible over the Internet at least one role and user needs to be defined, so here we use the current user on the device.

And we prepare a remote view for us to access the remote accounts AC in bank C.

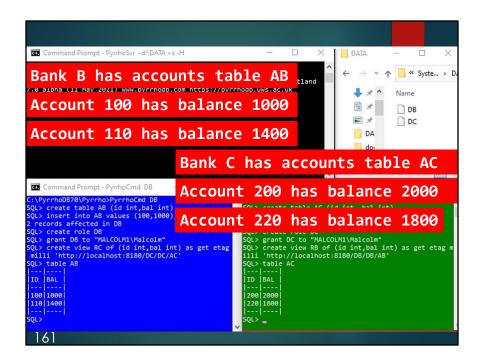
The metadata flags etag and milli tell Pyrrho to enforce RFC 7232 conditional request protocol, but using millisecond accuracy for the If-Unmodified-Since header.



Bank C has a similar set-up.

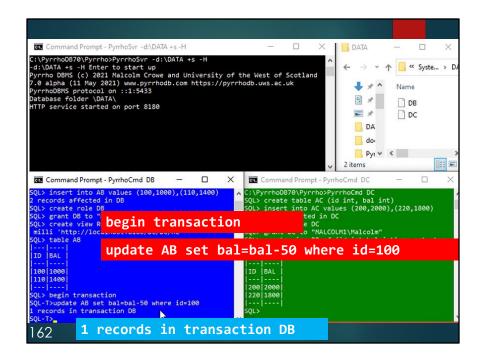
Database DC is the database for bank C. A user is defined who is authorised to connect to role DC.

AC is the table of accounts in bank C, and the RESTView RB gives remote access to accounts in bank B.



Let's show the current state of the accounts in the two banks.

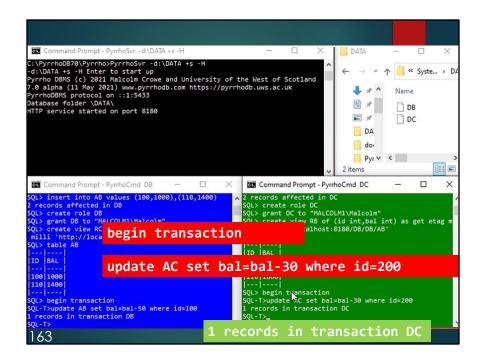
In the next steps, bank B will try to transfer £50 to account 200 in bank C, while bank C tries to transfer £30 from account 200 to an account in bank B.



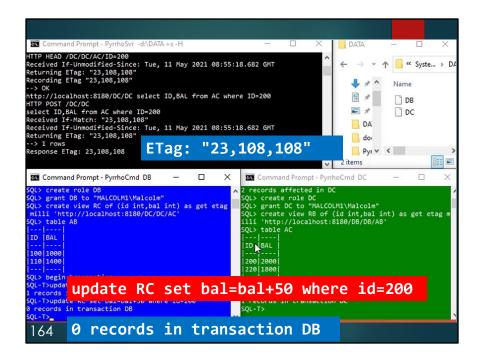
Two overlapping transactions will conflict on account 200.

In this slide, bank B starts a transaction and takes £50 from account 100 to be transferred. Nothing is written to disk yet.

The report on the update makes the point that the new record is in the transaction, rather than the database.

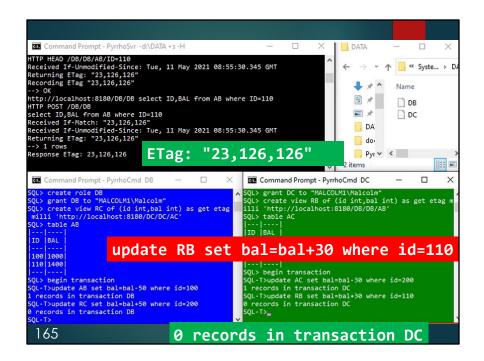


Someone in bank C starts a transaction around the same time and takes £30 from account 200 for another transfer.



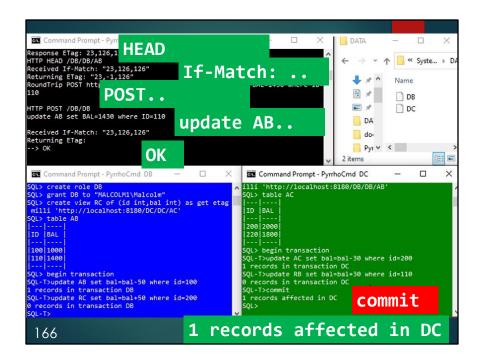
Now the blue teller transfers the £50 to account 200 in bank C using the remote view RC of their accounts. This is still in the transaction, and nothing has been committed. However we see Internet activity as bank B checks that 200 is an account in bank C. we suppose that the balance data will be withheld, because of the permissions, but bank B does receive an ETag. The response on bank B indicates that the changes are not for the local bank.

It is hoped that COUNT(*) can (always) be used instead of selecting columns to create ETags: will become clearer during the next phase of implementation.



The green teller also starts a transfer, of his £30 from account 200 to account 110, and receives an ETag for remote account 110.

At this point both tellers are about to commit their transaction. The first to do so will succeed, but no changes have been made yet.



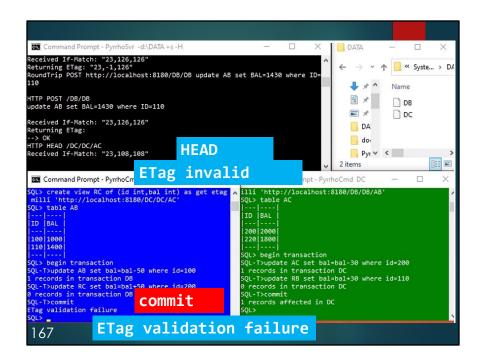
The green teller gets there first, and commits their transaction.

The first step is a HEAD request to verify the ETag sent to the green database is still valid. (It is.)

Then the update is POSTed to bank B, and on receiving OK from DB, the change is committed to local account 200 in bank C.

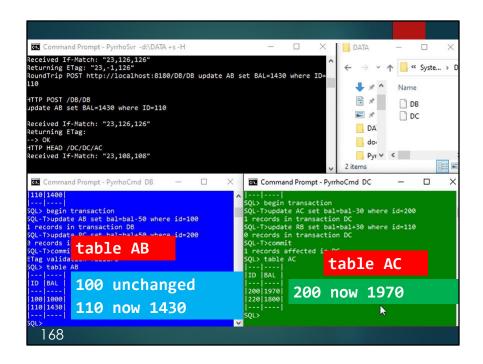
This will invalidate the ETag previously sent to the blue client.

The response to the commit indicates that the change to local account 200 is now recorded durably in the database.

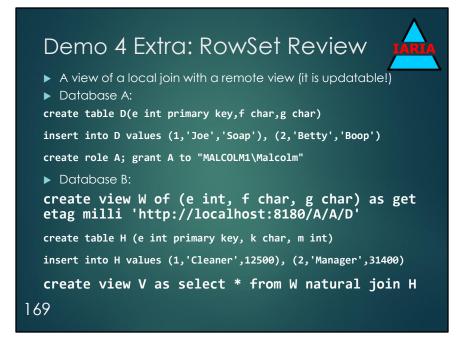


Sure enough, when the blue client attempts to commit, and sends its change to bank C, bank C reports that the ETag is invalid, and the transaction is aborted. The failure of this HEAD roundtrip rolls back the blue transaction entirely.

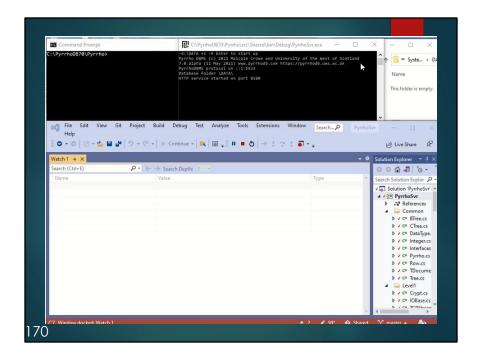
In this case, if the green client committed first, the local update would fail and a normal (non-ETag based) transaction conflict condition would be raised.



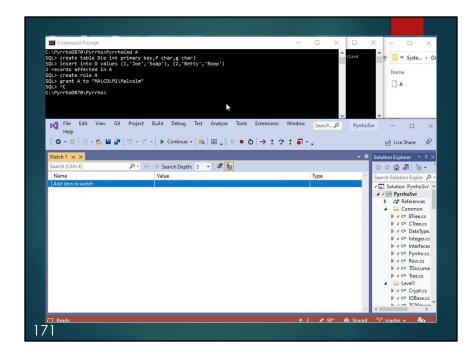
The teller in bank B can verify that the transfer from account 100 has not occurred, while bank C's transaction transferring £30 has been committed in both databases.



As an extra demo, we will use Visual Studio to explore RowSet Review, using as an example a viewed join of a local table with a remote view. This is part of the test23 suite in the Pyrrho V7 alpha distribution, where it is shown to be updatable. The example is a simplified system of a company employee database that records job titles and salaries, but where the employees' names are stored in another database.



We start as before with a debug session on Visual Studio where the command line arguments are again -d:DATA + s - H. The top row shows a command window for the clients, the debugger's server window, a glimpse of a file manager showing the empty database folder.

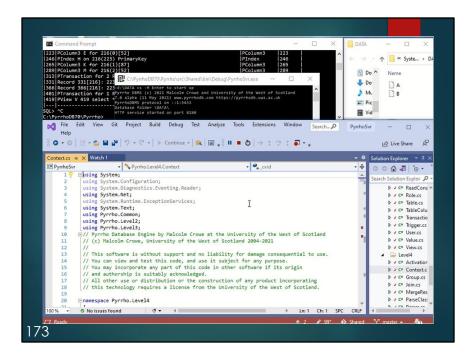


We first create database A with the SQL statements on slide 169.

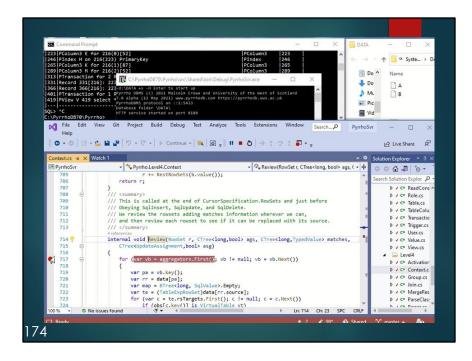
Command Prompt -		×	DATA	-	
1233 PColumn3 1223 1234 PColumn3 1223 1266 PIndex No 1263 1266 PIndex No 1263 1265 PColumn3 1265 PColumn3 1265 1265 PColumn3 462 1265 PColumn3 1265 1269 PColum3 462 1265 PColum3 1265 1280 PColum3 462 1265 PColum3 1265 1313 Promasction 767 280=52 1650 197 Promasction 131 1318 Record 312 1265<-Cleanere, 280=31260		^ —	← →	o ^ Nar o	A
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help	Search	<u></u>	PyrrhoSv		E Share
Toutput					All Pyrrhos ho'Svr Reference: Common BTree.c Crree.c DataTy Integer Interfa: Pyrrho Row.cs Toccur Tree.cs
			100000000000000000000000000000000000000	√ maste	

Now create the database B using the statements on slide 169. We have also had table "Log\$" from which we see that view V is at position 419.

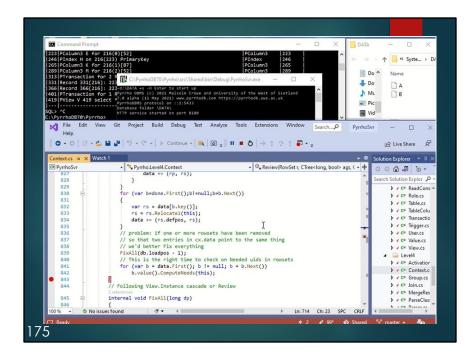
Stop the server: we do this for reproducibility, as compiled object uids are different during the session in which objects are created. Restart the server.



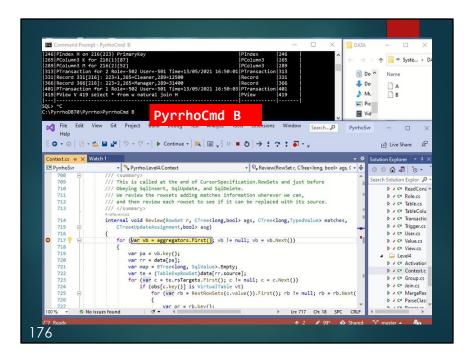
Find Context.cx in Level4 and double-click it. We are going to place breakpoints in the Context.Review() method.



The breakpoint for the start of Context.Review() is at line 717 in Context.cs.



The breakpoint at the end of the Context.Review() method is at line 843 of Context.cs.

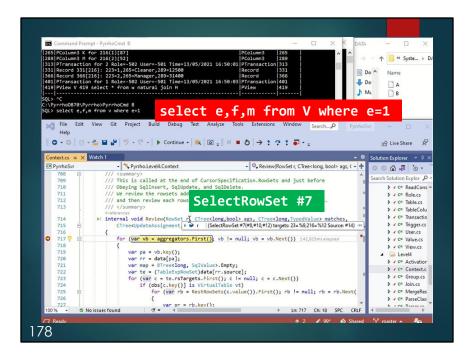


When we reopen database B, it is loaded into the server, and the compilation process for the view calls Context.Review()

Click Continue twice to get the normal SQL> prompt.

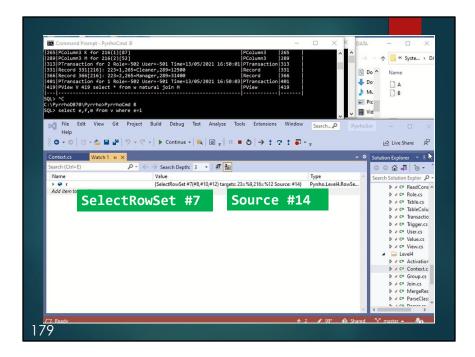
Command Promp	t - PyrrhoCmd B				D	ATA	- 0	>
265 PColumn3 K fo 289 PColumn3 M fo 313 PTransaction 331 Record 331[21 366 Record 36[21 2401 PTransaction	r 216(2)[52] for 2 Role=-502 User=-501 Time=13/05/2021 16:50 6]: 223-1,265=Cleaner,289=12500 6]: 223-2,265=Nanager,289=31400 for 1 Role=-502 User=-501 Time=13/05/2021 16:50	Record 331 Record 366				→ ~ 1	Name	t >
 SQL> ^C	elect * from w natural join H	PView 419 			8	🎝 ML 📰 Pic	В	
C:\PyrrhoDB70\Pyrr SOL>	ho>PyrrhoCmd B					Vid Vid		
Help	fiew G			rch	Pyn	rhoSvr	- 🗆	ہ چ
Context.cs + × W	atch 1				-	Solut	ion Explorer	- a
C# PyrrhoSyr				a, boo	l> ags, (🕶	+	0 A	0.
833	D 0.1			-		-		-
834						Searc	h Solution Exp	lor p
835							Þ ✓ Cª Rea	dCons
836							Þ ✓ C= Rol	e.cs
837							Þ ✓ C≋ Tab	le.cs
838			_	~			Þ ✓ C= Tab	leColu
839	<pre>FixAll(db.loadpos - 1); // This is the right time to check on N</pre>						▶ ✓ Cª Tran	sactio
841	for (var b = data.First(); b != null; b						Þ / C# Trig	aer.cs
842	<pre>b.value().ComputeNeeds(this);</pre>	- Unexc())					Þ ✓ Cª Use	
843						-	Þ ✓ Cª Valu	
844	<pre>// Following View.Instance cascade or Revie 2 references</pre>	w					▷ ✓ C ⁿ Viei I Sevel4	
845 🖃	internal void FixAll(long dp)						Level4	ivation
846	1						P + C" Cor	
847 🖻	<pre>for (var b = obs.PositionAt(dp); b != n</pre>	ull; b = b.Next())				1		
848	1 mar a la land ha						Þ ✓ C= Gro	
849 850	<pre>var p = b.key(); var np = obuids[p] ?? p;</pre>						▷ ✓ C= Joir	
850	<pre>var np = obuids[p] ?? p; var nb = (DBObject)b.value().Fix(th</pre>	2 - X -					¢ ✓ C= Me	
852	<pre>var nb = (ububject)b.value().Fix(th obs += (np, nb);</pre>	13);				*	Þ ✓ C≋ Par	
	issues found		Ln: 843	Ch: 9	SPC CRI	E 4	N CE Das	

We are going to select some items from V.



The given select statement is **select e,f,m from V where e=1**.

(Note the position of e in this statement: it occurs at position 8, so and the equals sign of the where condition is at position 28. We will see these numbers later.) RowSet Review for the select statement stops at the breakpoint we set. We are going to review the given RowSet r and its sources using the Watch window. It is a SelectRowSet.



We see the source is a rowSet #14 . To examine other rowSets we need to look at the **data** BTree in Context.

Command Prompt - Pytrho 265 PColumn3 K for 216(1			~ □ ×
288 PColumn3 M for 216(2) 313 PTransaction for 2 A 313 PTransaction for 2 A 331 Record 331(216): 223 366 Record 366(216): 223 4641 PTransaction for 1 R 419 PView V 419 select * 	[152] PCOLUMD3 289 02e=502 Users-501 Times13/05/2021 16:50:01 Pransaction 313 -1,265-Cleaner,289-31400 Record 331 366	•	↑ Syste > Do ↑ Name Do ↑ A ML ☐ B
C:\PyrrhoDB70\Pyrrho>Pyrr SQL> select e,f,m from v		S (1997)	
File Edit View G Help	P → Continue - K 0 = 0 → 1	Search Pyrrho	년 Live Share 호
Search (Ctrl+E)	P - ← → Search Depth: 3 - 47 🜆		-0 6 5 00
Name	Value	Туре	Search Solution Explor
 ▷ ☞ r ▲ ☞ data ▲ ☞ Count ▲ ☞ root ▲ ☞ count ▶ ☞ gtr ▲ ☞ slots 	[\$electRovGet #7(#3,#10,#2) targets: 23=%3(216=%12 Source=#14) ((246 inder RovGet 24C(23,265,209) Keys (23,265,209),(149)-JoinRo- 0.00000000000011 (Pyrrtho.Common.lnner <long, pyrrho.level4.rowset="">) 0.00000002 0.02 (Pyrrtho.Common.Leaf<long, pyrrho.level4.rowset="">) [\$ytrtho.Common.Leaf<long, pyrrho.level4.rowset="">)</long,></long,></long,>	Pyrrho.Level4.RowSe Pyrrho.Common.BTr long Pyrrho.Common.Bu int byte Pyrrho.Common.Bu	 C ReadCons C Role.cs C Table.cs C TableColu C TableColu C Transaction C Trigger.cs C User.cs
> Slots > ○ [0] ○ [1] ○ Statal > trail	[15ystem.collections.comenck.ety/duce/arxitong.fy/tho.common.bec {{435, Pyrtho.common.Leaf2{System.Int64,Pyrtho.Level4.RowSet}} {{6917529027641081863, Pyrtho.Common.Leaf2{System.Int64,Pyrtho 0x000000000000011	System.Collections.G	 ▷ ✓ C= Value.cs ▷ ✓ C= View.cs ▲ Eevel4 ▷ ✓ C= Activation
Add item to watch			C = Context.c C = Group.cs C = Join.cs C = MergeRes C = ParseClas C = ParseCla

Begin to expand **data**. RowSet #14 will be near the middle. Expand node root.slots[1].

331 Record 331(216): 223-1,265-Rd 3166 Record 56(216): 223-2,265-Rd 3166 Record 56(216): 223-2,265-Rd 401 Piranaction for 1 Role-392 401 Piranaction for 1 Role-392 401 Piranaction for 1 Role-392 501 > fc 502 > fc 502 > fc 503 > fc 504 > fc 504 > fc 505 > fc 505 > fc 506 > fc 507 > fc 508 > fc 508 > fc 509 > fc 509 > fc 500 > fc 501 > fc 502 > fc 503 > fc 504 > fc 505 > fc 505 > fc 506 > fc 507 > fc 508 > fc 508 > fc 509 > fc 500 > fc 501 > fc 502 > fc 503 > fc 504 > fc 505 > fc 506 > fc 507 > fc 508 > fc 509 > fc	Inager, 289-31409 [Record] 366 User-S01 Transaction (40) Natural join H [View 419 PView 419 PV	dow Search Pyrrhos	
Help Help O + O IO + O Context.cs Watch 1 + ×	t Build Debug Test Analyze Tools Extensions Winc [™] - ▶ Continue - ■ ⓐ _♥ [™] ■ • Ô → ↑ ↑ ↑ ↑	dow Search Pyrrhos	Svr — 🗆 (순 Live Share 셔
Help 0 - 0 10 - 20 ■ 10 10 - 0 Context.cs Watch 1 + ×	<pre></pre>	• •	년 Live Share 호
Contraction of the second s		- ¢	Solution Explorer
Search (Ctrl+E)			Joidtion Explorer +
	← → Search Depth: 3 - 🕂 🏭		001 🗸 🐻 •
Name	Value	Туре	Search Solution Explor
	(Pyrtho.CommonLiner <long. pyrtho.level4.rawset="">) 0x0000002 0x02 Pyrtho.CommonLiner<long. pyrtho.level4.rawset="">) [System.Collections.Generic.Key/ValueParkIong. Pyrtho.LocvelARowSet [System.Collections.Generic.Key/ValueParkIong. Pyrtho.Level4.RawSet>) [System.Collections.Generic.Key/ValueParkIong. Pyrtho.Level4.RawSet>) 0x00000000000000000000000000000000000</long.></long.>	 system.Collections.G., ing Pyrtho.Common.Bu., int byte wos. system.Collections.G., 3), System.Collections.G., 245 S., System.Collections.G., 142, System.Collections.G., 142, System.Collections.G., 143, System.Collections.G., 144, System.Collections.G., 144, System.Collections.G., 144, System.Collections.G., 145, System.Collections.G	 > cr: ReadCon > cr: TableCol > cr: TableCol > cr: TableCol > cr: TableCol > cr: Triggercol > cr: Valuecol > cr: Valuecol > cr: Valuecol > cr: Valuecol > cr: Context. > cr: Groupson > cr: Groupson > cr: MergeRe <li< td=""></li<>

Expand the node containing TableRowSet #14

Command Prompt - Pyrrho 7	Intitled - Notepad		- 🗆 X
265 PColumn3 K for 216(1)			
	Edit Format View Help ctRowSet #7(#8,#10,#12) targets: 23=%9,216=%12 Source: #14}		ŕ
File Edit View Git Proj	ect Build Debug Test Analyze Tools Extensions Window	Search P Pyrrhos	5vr — 🗆
Help			
0 • 0 13 • 🖕 🖬 🖉 17 •	$- \mathcal{C} + \models \text{Continue} + \blacksquare \blacksquare \blacksquare = 0 \rightarrow \ddagger \mathcal{C} \stackrel{*}{\rightarrow} = \overline{\mathcal{C}} + \overline{\mathcal{C}} \stackrel{*}{\rightarrow} = \mathcal{$	÷	🖻 Live Share 🖇
Context.cs Watch 1 + X		- Ø	Solution Explorer 💌 👎
	• ← → Search Depth: 3 • # 📲 🐜		
47.21		-	00 🔂 💵 '0 -
Name	Value	Type	Search Solution Explor
→ 🕑 r → 😫 data			♦ ✓ C [#] ReadCon
A Gata	{(246=IndexRowSet 246(223,265,289) Keys: (223,265,289),419=JoinRo 0x00000000000011	Pyrrho.Common.BTr	♦ ✓ C= Role.cs
	{Pyrrho.Common.Inner <long, pyrrho.level4.rowset="">}</long,>	Pyrrho.Common.Bu	♦ ✓ C= Table.cs
4 🤗 root		int	↓ < C= TableColi
v [€] EndPos	0x0000002		♦ ✓ C* Transacti
🥥 count	0x02	byte	
i count i i gtr	0x02 {Pyrrho.Common.Leaf <long, pyrrho.level4.rowset="">}</long,>	byte Pyrrho.Common.Bu	▷ ✓ C= Transaction ▷ ✓ C= Trigger.cs ▷ ✓ C= User.cs
 count i gtr i gi slots 	0x02 {Pyrrho.Common.Leaf <long, pyrrho.level4.rowset="">} {System.Collections.Generic.KeyValuePair<long, pyrrho.common.bu<="" td=""><td>byte Pyrrho.Common.Bu System.Collections.G</td><td>Þ ✓ C= Trigger.cs</td></long,></long,>	byte Pyrrho.Common.Bu System.Collections.G	Þ ✓ C= Trigger.cs
 Gunt gtr slots (0) 	0x02 (Pyrtho.Common.Leaf <long, pyrtho.level4.rowset=""> } (System.Collections.Generic.KeyValuePair<long, pyrtho.common.bu<br="">{{43, Pyrtho.Common.Leaf2[System.Int64,Pyrtho.Level4.RowSet]}</long,></long,>	byte Pyrrho.Common.Bu System.Collections.G System.Collections.G	↓ ✓ C= Trigger.cs ↓ ✓ C= User.cs
	0x02 {Pyrrho.Common.Leaf <long, pyrrho.level4.rowset="">} {System.Collections.Generic.KeyValuePair<long, pyrrho.common.bu<="" td=""><td>byte Pyrrho.Common.Bu System.Collections.G System.Collections.G</td><td>↓ C= Trigger.cs ↓ C= User.cs ↓ C= Value.cs</td></long,></long,>	byte Pyrrho.Common.Bu System.Collections.G System.Collections.G	↓ C= Trigger.cs ↓ C= User.cs ↓ C= Value.cs
 Gunt gtr slots (0) 	0x02 (Pyrtho.Common.Leaf <long, pyrtho.level4.rowset="">) (System.Collections.Generic.Key/ValuePair<long, pyrtho.common.bu<br="">{(435, Pyrtho.Common.Leaf2[System.Int64.Pyrtho.Level4.RowSet]]} ((6917529027641081863, Pyrtho.Common.Leaf2[System.Int64.Pyrtho © 0x60000000000</long,></long,>	byte Pyrrho.Common.Bu System.Collections.G System.Collections.G System.Collections.G	 ↓ C= Trigger.cs ↓ C= User.cs ↓ C= Value.cs ↓ C= View.cs
	0x02 (Pyrtho.Common.Leaf <long, pyrtho.level4.rowset="">) (System.Collections.Generic.Key/ValuePair<long, pyrtho.common.bu<br="">{(435, Pyrtho.Common.Leaf2[System.Int64.Pyrtho.Level4.RowSet]]} ((6917529027641081863, Pyrtho.Common.Leaf2[System.Int64.Pyrtho © 0x60000000000</long,></long,>	byte Pyrrho.Common.Bu System.Collections.G System.Collections.G System.Collections.G long	 ↓ C= Trigger.cs ↓ C= User.cs ↓ C= Value.cs ↓ C= View.cs ↓ C= View.cs
	0x02 (Pyrtha.Common.Lest <long.pyrtha.leveld.rouset>) (System.Collections.Generic.KeyValuePair<long.pyrtha.common.bu (1435, Pyrtha.Common.Lest 72(System.Int64;Pyrtha.Leveld.RouSet)) (159772920761010382, Pyrtha.Common.Lest 72(System.Int64;Pyrtha = 0x600000000000007 (Pyrtha.Common.Lest folgs.Pyrtha.Leveld.RouSet>)</long.pyrtha.common.bu </long.pyrtha.leveld.rouset>	byte Pyrrho.Common.Bu System.Collections.G System.Collections.G long Pyrrho.Common.Bu	 ▷ ✓ C= Trigger.cs ▷ ✓ C= User.cs ▷ ✓ C= Value.cs ▷ ✓ C= View.cs ▲ □ Level4 ▷ ✓ C= Activatio
 i count i gtr i slots i [0] i [1] j Key j Value i EndPos 	0x02 (Pynho.Common.Leaf <long. pynho.level4.rowset="">) (System Collections.Generic.KeyValuePair<long. pynho.common.bu<br="">([435, Pynho.Common.Leaf 2[System.Int64, Pyrho.Level4.RowSet])) ([697125027641081883, Pynho.Common.Leaf 2[System.Int64, Pynho ⇔ 0x6000000000000 (Pynho.Common.Leaf<long. pynho.level4.rowset="">) 0x0000005</long.></long.></long.>	byte Pyrrho.Common.Bu System.Collections.G System.Collections.G long Pyrrho.Common.Bu int byte	 ▷ ✓ C= Trigger.c: ▷ ✓ C= User.cs ▷ ✓ C= Value.cs ▷ ✓ C= View.cs ▲ □ Level4 ▷ ✓ C= Activatio ▷ ✓ C= Context.4
Count Gamma Gamma Gamma Gamma Gamma Gamma Gamma Gamma Gamma Gamma Gamma Gamma Gamma Gamma Gamma Gamm	0x02 (Pyrtho Common Lest < long, Pyrtho Leveld RowSet>) (System, Collections, Generic, Key/MulevBarclong, Pyrtho, Common Bu (1435, Pyrtho, Common, Lest 2[System.Int64, Pyrtho, Leveld, RowSet]) (1591752902761081803, Pyrtho.Common.Lest 2[System.Int64, Pyrtho © 0x600000000007 (Pyrtho Common Lest < long, Pyrtho.Leveld.RowSet>) 0x0000005 0x06 (System, Collections, Generic, Key/MulevBarclong, Pyrtho.Leveld.RowS (1477, JoineNoxSet 347(90, 0), 0, 2, 0, 4, 0, 93) matching (@0c (@3)	byte Pyrtho.Common.Bu System.Collections.G System.Collections.G long Pyrtho.Common.Bul int byte System.Collections.G System.Collections.G	 ↓ < C= Trigger.c: ↓ < C= User.cs ↓ < C= Value.cs ↓ < C= View.cs ↓ < C= View.cs ↓ < C= Activation ↓ < C= Activation ↓ < C= Context.a ↓ < C= Group.cs
$\begin{tabular}{lllllllllllllllllllllllllllllllllll$	0x02 (Pyrrho.Common.Leaf <long.pyrrho.level4.rowset>) (System.Collections.Generic.KeyValuePair<long.pyrrho.common.bu< td=""> [435, Pyrrho.Common.Leaf 2[System.Int64.Pyrrho.Level4.RowSet]) (169712502761031836, Pyrrho.Common.Leaf 2[System.Int64.Pyrrho © 6x000000000000000000000000000000000000</long.pyrrho.common.bu<></long.pyrrho.level4.rowset>	byte Pyrtho.Common.Bu System.Collections.G System.Collections.G long Pyrtho.Common.Bu.L. byte System.Collections.G System.Collections.G System.Collections.G	 ▷ ✓ C= Trigger.c: ▷ ✓ C= User.cs ▷ ✓ C= View.cs ▷ ✓ C= View.cs △ C= View.cs △ C= Activatio ▷ ✓ C= Context.t ▷ ✓ C= Group.cs ▷ ✓ C= Join.cs
	0x02 (Pyrtho Common Leaf < long, Pyrtho Level4 RowSet>) (System Collections Generic Key/bluePair Common Bu	byte Pyrtho.Common.Bu System.Collections.G System.Collections.G long Pyrtho.Common.Bu.Ly int byte System.Collections.G System.Collections.G System.Collections.G	 ↓ c= Trigger.c: ↓ c= User.cs ↓ c= Value.cs ↓ c= View.cs ↓ c= Context. ↓ c= Group.cs ↓ c= Group.cs ↓ c= Mergetes ↓ c= Mergetes
$\begin{tabular}{lllllllllllllllllllllllllllllllllll$	0x02 (Pyrrho.Common.Leaf <long.pyrrho.level4.rowset>) (System.Collections.Generic.KeyValuePair<long.pyrrho.common.bu< td=""> [435, Pyrrho.Common.Leaf 2[System.Int64.Pyrrho.Level4.RowSet]) (169712502761031836, Pyrrho.Common.Leaf 2[System.Int64.Pyrrho © 6x000000000000000000000000000000000000</long.pyrrho.common.bu<></long.pyrrho.level4.rowset>	byte Pyrtho.Common.Bu System.Collections.G System.Collections.G long Pyrtho.Common.Bu.Ly int byte System.Collections.G System.Collections.G System.Collections.G	 C = User.cs C = User.cs C = Value.cs C = Context. C = Context. C = Context. C = MargeRe C = MargeRe<

Open a Notepad window. Scroll to the top and copy the value of r into the Notepad.

265 PColumn3 K for 216(1) 289 PColumn3 M for 216(2) File Edit	d - Notepad Format View Help		- 🗆 X
3331 Record 331(216): 223- 3361 Record 36(216): 223- 401 [Pransaction for 1 Re 431 [Pview V 419 select * (a). > Cr (b). > chore (b). > chore (Socie #14(#8,#10,#12); targets: 22-39,216-312 Source: #14) Socie #14(#8,#10,#12)[30,51] P Cce: %10		23=%9,216=%12
Help O • O 3 • 2 4 4 Context.cs Watch 1 • Search (Ctrl+E) Name			
🖋 EndPos			
@ count			100
Þ 🤗 gtr			UTE-8
✓ ♀ slots ♦ ♀ [0]	[435, Pyrrho.Common.Leaf 2[System.Int64, Pyrrho.Level4.RowSet]]}	100% Windows (CRLF) System.Collections.G	
· · [0]		System.Collections.G	▷ ✓ C= Transactio
J Key	0x60000000000007	long	♦ ✓ C* Trigger.cs
م الع Value	{Pyrrho.Common.Leaf <long, pyrrho.level4.rowset="">}</long,>	Pyrrho.Common.Bu	♦ ✓ C= User.cs
EndPos	0x00000005	int	Value.cs
🥥 count	0x06	byte	View.cs
🔺 🤪 slots	{System.Collections.Generic.KeyValuePair <long, pyrrho.level4.rows<="" td=""><td></td><td>🔺 📄 Level4</td></long,>		🔺 📄 Level4
Þ 🗢 [0]	{{437, JoinRowSet 437(@0,@1,@2,@4,@5 @3) matching (@0=(@3),	System.Collections.G	V C* Activation
▶	{[450, SelectedRowSet 450(@3,@4,@5) targets: 216=450 Source: 246 S		V C* Context.c
▶ ● [2]	{[5764607523034234887, SelectRowSet #7(#8,#10,#12) targets: 23= %9		♦ ✓ C= Group.cs
4 🗢 [3]	{[5764607523034234894, TableExpRowSet #14(#8,#10,#12[%0,%1) key 0x50000000000000e		V C* Join.cs
Key کر Value کر	{TableExpRowSet #14(#8,#10,#12 %0,%1) key (#8,#10) where (#28) m	long Purtha Lavald RowSa	Þ ✓ C= MergeRes
Value		Pynno.Level4.Kowse	♦ ✓ C= ParseClas:
▶ ● [4]	" [5764607523034234899, JoinRowSet %10(#8,#10,#12,%0,%1 %13) key	System.Collections.G *	A Deres of
		er et	Manastar a Ro

Expand the node containing the TableRowSet #14 and copy its value into the Notepad. We see its source is RowSet %10, which we can see in the next node below.

Command Prompt - Pyrrhod	led - Notenad		- 🗆 X
265 PColumn3 K for 216(1)			
	Format View Help		
	pwSet #7(#8,#10,#12) targets: 23=%9,216=%12 Source: #14} pRowSet #14(#8,#10,#12 %0,%1) key (#8,#10) where (#28) m		22-W0 246-W42
366 Record 366[216]: 223= Source:		acches (#8=1) cargets:	23=#9,210=#12
	Set %10(#8,#10,#12,%0,%1 %13) key (#8,#10,%0,%1,#12,%13)		
(#8=(%13),%13=(#8)) targets: 23=%9,216=%12 JoinCond: (%17) match	ing #8=%13 %13=#8 First	: %18 Second:
5QL> ^C *15}			
<pre>C:\PyrrhoDB70\Pyrrho>Pyrrh SQL> select e,f,m from v w</pre>	First	. 9/10	
SQL> Select e, t, m from V W	inRowSet %10	: %TQ	
File Edit View Git			
Help	Second	4. V1E	
	Second	1: %15	
🔆 O - O 🕾 - 🏜 🗳			-
Context.cs Watch 1 -			
Search (Ctrl+E)			
Search (Ctrl+E)			
and a second			
Name			
Name			
Name > (0) > (1) // Key > Value	Ln 3, Col 207	100% Windows (CRLF)	UTF-8
Name	0x00000005	int	V V C- IdDIECO
Name	0x00000005 0x06	int byte	P ✓ C™ Transact
Name ↓ ♥ [0] ↓ ♥ [1] ↓ ♥ [4] ↓ ♥ Value ↓ ♥ IndPos ● count ↓ ♥ slots	0x00000005 0x06 {System.Collections.Generic.KeyValuePair <long, pyrrho.level4.rows<="" td=""><td>int byte System.Collections.G</td><td>▷ ✓ C= Transact ▷ ✓ C= Trigger.</td></long,>	int byte System.Collections.G	▷ ✓ C= Transact ▷ ✓ C= Trigger.
Name	0x0000005 0x06 (System.Collections:Generic.KeyValuePair <long, pyrrho.levela.rows<br="">([437, JoinRowSet 437(@0,@1,@2,@4,@5]@3) matching (@0=(@3)</long,>	int byte System.Collections.G System.Collections.G	D ✓ C= Transact D ✓ C= Trigger. D ✓ C= User.cs
Name →	0x0000005 0x06 (System.Collections.Generic.KeyValuePair <long, pyrrho.level4.rows<br="">([437, JoinRowSet 437(연0, 한 1순고순석 당)(중) matching (연0=(영3) ([450, SelectedRowSet 430(명-24, Ф5) tragets: Tol=435 Source 246 S</long,>	int byte . System.Collections.G System.Collections.G . System.Collections.G	▷ ✓ C= Transact ▷ ✓ C= Trigger. ▷ ✓ C= User.cs ▷ ✓ C= Value.cs
Name →	0x0000005 0x66 (System.Collections.Generic.KeyValuePair <long. pyrrhol.level4.rows<br="">([437], JoinRowSet 437(母)のり、色、ターの 9(音) matching (命に(音)) ([450, SelectedRowSet 450(母)、母、母) targets: 216×450 Source: 246 S. ([576460752303224987, SelectRowSet 年7(母,号) (1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(</long.>	int byte System.Collections.G System.Collections.G System.Collections.G	 C= Transact C= Transact C= Trigger. C= User.cs C= Value.cs C= Value.cs
Name → @ [0] → @ [1] → £ Key → £ Value → € count → @ slots → @ (0] → @ [2] → @ [3]	D:0000005 DoB (System.Collections.Generic.KeyValuePairs/long.Pymho.Level4.RowS. ((437), JoinRowSet 437(@0,@1,@2,@4,@5)=3), matching (@Ce3), (455, SelecteRowSet 430(@3,@4,@5)=targetz 12645 Source 245 ((157460752304224894, TableSenowSet #14(@4)) = (1215(0,5)) (b; c) (157460752304224894, TableSenowSet #14(@4)) = (1215(0,5)) (b; c)	int byte - System.Collections.G - System.Collections.G - System.Collections.G - System.Collections.G	 C= Transact C= Transact C= Trigger. C= User.cs C= Value.cs C= Value.cs C= View.cs Level4
Name →	0x0000005 0x60 (System, Cellections, Generic, KeyValuePair <long, pyrrho.level4,="" rows.,<br="">((437, JoinRowSet 437(空)の), のとのとのも、 (455, SelectedRowSet 450(空), 24(空) 1 anget: 23e5 (15764607523034224885, SelectRowSet #7(#3, #10, #12) target: 23=58, (15764607523034224885, JoinRowSet #7(#3, #10, #12) target: 23=58, (15764607523034224885, JoinRowSet #10(#3, #10, #12) target: 23=58, (15764607523034224885, JoinRowSet #10(#3, #10, #12, %5, %1)% 1), key, (15764607523034224885, JoinRowSet %10(#3, #10, #10, #10, %1)), key, (15764607523034224885, JoinRowSet %10(#3, #10, #10, %5, %1)% 1), key, (15764607523034224885, JoinRowSet %10(#3, #10, #12, %5, %1)% 1), key, (15764607523034224885, JoinRowSet %10(#3, #10, #10, %5, %1)% 1), key, (1576460752303424885, JoinRowSet %10(#3, #10, #10, %5, %1)% 1), key, (1576460752303424885, JoinRowSet %10(#10, #10, #10, %1)), key, (15764607548, %1), key, (157646075485, %1), key, (157646075433042485, JoinRowSet %10(#10, #10, %1)), key, (15764607548), key, (157648), key, (157</long,>	int byte - System.Collections.G - System.Collections.G - System.Collections.G - System.Collections.G - System.Collections.G	 ▷ < C= Transact ▷ < C= Transact ▷ < C= Trigger. ▷ < C= User.cs ▷ < C= User.cs ▷ < C= Value.cs ▷ < C= Value.cs ▷ < C= Value.cs ▷ < C= Activati
Name → @ [0] → @ [1] → £ Key → £ Value → € count → @ slots → @ (0] → @ [2] → @ [3]	Dr.0000005 Dr06 [System.Cellections.Genesic.KeyValueBairciong.Pyrrho.LevelA.RowS. [437].JoinRowSet 437(@0,@1,@2,@4,@5) targets 2156450 Source 2455 [4540_SelectedRowSet 450(@3,@4,@5) targets 2156450 Source 2455 [1574607752304224984].TableSphowSet #14(=#1501-121kg)%15; key [1574607752304224984].TableSphowSet #14(=#1501-121kg)%15; key 0.500000000013	int byte System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G long	
Name → ○ [0] → ○ [1] → ▷ Key → ▷ Value → ▷ count → ○ [0] → ○ [1] → ○ [2] → ○ [3] → ○ [4] → ▷ [4] → ▷ Key → ▷ Key	Dx0000005 Dx60 (System_Cellections.Generic.KeyValuePair <long, pyrrho.level4.rov5.<br="">((43), JoinRov5et 437(40)、01(2)、01(2)、01(2)、01(4) (450, SelectedRov5et 53(0)(2)、44(4)); Targetz 15445 Source 2455 (15764407732034224885, SelectRov5et #714(8) #10, #12) targetz 23 %3. (15764607522034224885, SelectRov5et #714(8) #10, #121(50, %1) key. (15764607522034224885, SelectRov5et #710(9), #10, %10, #121(50, %1) key. (157646075220422485, SelectRov5et #710(9), #10, %10, #12, %1). (15764075220422485, SelectRov5et #710(9), #10, %10, #12, %1). (15764075200000000000000000000000000000000000</long,>	int byte System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G long	 ↓ < ↓ < ← Transact ↓ < ↓ < ← Trigger. ↓ < ↓
Name → @ [0] → @ [1] → Key → Kap → Count → @ [0] → @ [0] → [1] → [Dx0000005 Dx60 (System_Cellections.Generic.KeyValuePair <long, pyrrho.level4.rov5.<br="">((43), JoinRov5et 437(40)、01(2)、01(2)、01(2)、01(4) (450, SelectedRov5et 53(0)(2)、44(4)); Targetz 15445 Source 2455 (15764407732034224885, SelectRov5et #714(8) #10, #12) targetz 23 %3. (15764607522034224885, SelectRov5et #714(8) #10, #121(50, %1) key. (15764607522034224885, SelectRov5et #710(9), #10, %10, #121(50, %1) key. (157646075220422485, SelectRov5et #710(9), #10, %10, #12, %1). (15764075220422485, SelectRov5et #710(9), #10, %10, #12, %1). (15764075200000000000000000000000000000000000</long,>	int byte System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G Iong Pyrtho.Level4.RowSe	 > c= fartisect > c= Triager. > c= User.cs > c= View.cs > c= View.cs > c= cativati > c= Cativati > c= Context > c= Sroup.c > c= Join.cs
Name →	Dr.0000005 Dr.06 [System.Cellections.Genesic.KeyValueBairciong_Pyrrho.LevelA.RowS. [(437, JoinRowSet 437(@0,@1,@2,@4,@5) targets 216-450 Source: 245 (1574640752304224894; TablestpowSet #14(#3;#12) targets 23 = 56. (1574640752304224894; TablestpowSet #14(#3;#12) targets 23 = 56. (1574640752304224894; TablestpowSet #14(#3;#12) targets 23 = 56. (1574640752304224894; TablestpowSet #14(#3;#12) targets 23 = 56. 0.500000000013 = (JoinRowSet %10(#8,#10,#12,%0,%1)%13) key (#8,#10,%0,%1,#12,%1). te.	int byte System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G Iong Pyrtho.Level4.RowSe	 ↓ < ↓ < ← Transact ↓ < ↓ < ← Trigger. ↓ < ↓
Name → @ [0] → @ [1] → Key → Kape → Count → @ [0] → @ [0] →	Dx0000005 0x06 (5ystem.Cellections.Generic.KeyValuePair <long.pyrrhol.evel4.rov5 ([47], JoinRov5et 437(@),@1,@2,@4,@39) matching (@0-(@3), (453, SelectRov5et 430(@),&40) targtst 216455 Source 246 - (1574640752304224894, ToblectRov5et #714(# 10-1215(%)) tby (1574640752304224894, ToblectRov5et #714(# 10-1215(%)) tby (1574640752304224894, ToblectRov5et #714(# 10-1215(%)) tby (1574640752304224894, ToblectRov5et #714(# 10-1215(%)) tby (1574640752304224894, ToblectRov5et #71(# 10-1215(%)) tby (1574640752304224894, ToblectRov5et #71(# 10-1215(%)) tby (157460752302761031863, OrderedRov5et #71@0,@1,@2) key (@) or (16917529027611031863, OrderedRov5et #71@0,@1,@2), key (@) or</long.pyrrhol.evel4.rov5 	int byte System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G Iong .Pyntho.Level4.RowSe System.Collections.G	 > c = fansact > c = fingger. > c = View.cs > c = View.cs > c = View.cs > c = Activati > c = Context > c = Group.c > c = Join.cs > c = MergeR

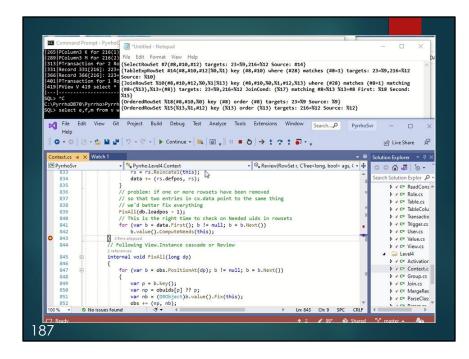
It is a JoinRowSet, with sources %18 and %15. These will both be in the **gtr** node of the **data** tree.

	I - Notepad		- 🗆 X
265 PColumn3 K for 216(1)			- 0 ^
	Format View Help		
	Set #7(#8,#10,#12) targets: 23=%9,216=%12 Source: #14}		
366 Record 366[216]: 223= {TableExpR	owSet #14(#8,#10,#12 %0,%1) key (#8,#10) where (#28) ma al	tches (#8=1) targets:	: 23=%9,216=%12
401 PTransaction for 1 Ro (loinRowSe	t %10(#8,#10,#12,%0,%1 %13) key (#8,#10,%0,%1,#12,%13)	where (#28) matches ((#8=1) matching
	%13=(#8)) targets: 23=%9,216=%12 JoinCond: (%17) matchi	ng #8=%13 %13=#8 Firs	st: %18 Second:
SQL> ^C %15}	wSet %18(#8,#10,%0) key (#8) order (#8) targets: 23=%9	Courses #0)	
C: (Fyrrnobb/e)Fyrrno>Fyrrn	WSet %18(#8,#10,%0) Key (#8) order (#8) targets: 23=%9	Source: %9}	
SQL> select e,f,m from v w		0/0	
File Edit View Git	lanad Day Cat 9/10 SOU	rce: %9	
Help	leredRowSet %18		
and the second			
💿 🗢 💿 🛛 🕄 - 🍰 🔛 💕			
Context.cs Watch 1 🕫			
Search (Ctrl+E)			
Name			
Name Count			
✓ Count ✓ root ✓ EndPos			
Count Count Count Count Count	Ln 4, Col 77	100% Windows (CRLF,	
✓ Count ✓ orot ✓ EndPos ✓ count ✓ gtr	{Pyrrho.Common.Leaf <long, pyrrho.level4.rowset="">}</long,>	Pyrrho.Common.Bu	
✓ Count ✓ root ✓ EndPos ✓ count ✓ gt EndPos	{Pyrrho.Common.Leaf <long, pyrrho.level4.rowset="">} 0x00000006</long,>	Pyrrho.Common.Bu int	♦ C Hablect ♦ ✓ C Transac
✓ Count ✓ root ✓ FodPos ✓ count ✓ gtr ✓ EndPos ✓ count	{Pyrrho.Common.Leaf <long, pyrrho.level4.rowset="">} 0x00000006 0x07</long,>	Pyrrho.Common.Bu int byte	
Count Generation Genera	{Pyrrho.Common.Leaf <long, pyrrho.level4.rowset=""> } 0x0000006 0x07 {System.Collections.Generic.KeyValuePair<long, pyrrho.level4.rows<="" td=""><td>Pyrrho.Common.Bu int byte System.Collections.G</td><td>♦ ✓ C= Transac ♦ ✓ C= Trigger.</td></long,></long,>	Pyrrho.Common.Bu int byte System.Collections.G	♦ ✓ C= Transac ♦ ✓ C= Trigger.
	(Pyrtho.Common.Leaf <long, pyrtho.level4.rowset="">) 0x0000006 0x07 (System.Collections.Generic.Key/ValuePair<long, pyrtho.level4.rows<br="">(16917530027641081864, OrderedRowSet @3(意え意人を) key (き) or</long,></long,>	Pyrrho.Common.Bu int byte System.Collections.G System.Collections.G	 ↓ C = Transac ↓ C = Trigger. ↓ C = User.cs
✓ Count ✓ or root ✓ EndPos ✓ count ✓ gtr ✓ EndPos ✓ count ✓ count ✓ slots	(Pyrrho.Common.Leaf <long, pyrrho.level4.rowset="">) 0x0070006 0x07 (System.Collections.Generic.Key/Value9ai<long, pyrrho.level4.rows<br="">(16917529027641081864, OrderedRowSet @0(@3,@4,@5) key (@3) or (1807045032224732840, TabletexpRowSet %6(€8,#10,%0,%1,#12)%13)</long,></long,>	Pyrrho.Common.Bu int byte System.Collections.G System.Collections.G System.Collections.G	 ↓ C= Transac ↓ C= Trigger. ↓ C= User.cs ↓ C= Value.cs
	(Pyrtho.Common.Leaf <long, pyrtho.level4.rowset="">) 0x0000006 0x07 (System.Collections.Generic.Key/ValuePair<long, pyrtho.level4.rows<br="">(16917530027641081864, OrderedRowSet @3(意え意人を) key (き) or</long,></long,>	Pyrrho.Common.Bu int byte System.Collections.G System.Collections.G System.Collections.G	 > C " Transac > C " Transac > C " Trigger. > C " User.cs > C " Value.cs > C " View.cs
	(Pyrtho.Level4.RowSet>) 0x070 0x07 (System.Collections.Generic,Key/ValuePair <long, pyrtho.level4.rows<br="">(19/17520027641081864, OrderedRowSet @8(@3,@4,@5) key (@3) or (18070450322472884), BathEoryBowSet %9(@3,01,00,10,10,11,12)(s1)), (18070450322472884), BathEoryBowSet %9(@3,01,00,10,10,11,12)(s1)),</long,>	Pyrrho.Common.Bu int byte System.Collections.G System.Collections.G System.Collections.G System.Collections.G	 ↓ < C= Transac ↓ < C= Transac ↓ < C= Transac ↓ < C= User.cs ↓ < C= View.cs ↓ < C= View.cs ↓ < C= View.cs
	(Pyrtho.Common.Leaf <long. pyrtho.level4rowset="">) 0x0070006 0x07 (System.Collections.Generic.Key/ValuePair<long. pyrtho.level4rows<br="">(§071520027541081684, OrderedRowSet @0(@3,@4,@5) key (@3) or. (§07045053224792844), Bable5xPRowSet %0(#8,=10,%0) Key (#8,=10,%0) (§07045053224792844), RentRowSet %0(#8,=10,%0) Key (#8,=10,%0) (§07045053224792844), RentRowSet %0(#8,=10,%0),%1 =12(%3) Sex.</long.></long.>	Pyrrho.Common.Bu int byte System.Collections.G System.Collections.G System.Collections.G System.Collections.G	 ↓ < C* Transac ↓ < C* Transac ↓ < C* Transac ↓ < C* User.cs ↓ < C* Value.cs ↓ < C* Value.cs ↓ < C* Value.st ↓ < C* Activati
	(Pyrtho.Level4.RowSet>) 0x000006 0x000006 0x000006 0x01 (System Collections Generic Key/Value9airclong, Pyrtha.Level4.RowS. (18)7152027541081854, OrdereaRowSet @0(@3,@4,@5) key (@3) or (18)074503224792840, BaletSxp80xvSet %0(@3,@4,@5) key (@3) or (18)074503224792841, RestRowSet %0(@3,@4,@5) key (@3) in (18)074503224792841, JoniRowSet %10(@1,@1,01,%) #12(3) key (18)074503224792844, JoniRowSet %10(@1,%11,%11,%12) key.(%13) (18)074503224792844, JoniRowSet %10(@1,%11,%11,%12) key.(%13) (18)074503224792844, JoniRowSet %10(@1,%11,%11,%12) key.(%13) (18)074503224792844, JoniRowSet %10(@1,%11,%11,%11,%11,%11,%11,%11,%11,%11,%	Pyrrho.Common.Bu int byte System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G	
	(Pyrtho.Level4.RowSet>) 0x0000006 0x07 (System.Collections.Generic.KeyValuePair <long.pyrtho.level4.rows (16971530027541081854, OrderedRowSet @8(@3,@4,@5) key (@3) or (180704503224792841, BastBockSet \$9(@4, #10,%1,%1,#12)k13) (180704503224792841, BastBockSet \$9(@4, #10,%1,%1,#12)k13) (180704503224792844, SoltextedRowSet %16(%13,%1,#12)k13) (180704503224792844, SoltextedRowSet %16(%13,%1,#12)k2) (%13) (180704503224792844, OrderedRowSet %16(%13,%1,#12) key (%13) (180704503224792844, OrderedRowSet %16(%13,%1,#12) key (%13) (180704503224792845, OrderedRowSet %16(%13,%1,#12) key (%13) (180704503224792856, OrderedRowSet %16(%13,\%16) (18070450324792856, OrderedRowSet %16(%13,\%16) (18070450324792856, OrderedRowSet %16(%13,\%16) (18070450324792856, OrderedRowSet %16(%13,\%16) (18070450324792856, OrderedRowSet %16(%13,\%16) (18070450324792856, OrderedRowSet %16(%13,\%16) (18070450324792856, OrderedRowSet %16(%13,\%16)) (180704503247,\%16) (180704503247,\%16) (180704503247,\%16) (180704503247,\%16) (180704503247,\%16) (180704503247,\%16) (180704503247</long.pyrtho.level4.rows 	Pyrrho.Common.Bu int byte System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G Iong	
	(Pyrtho.Level4.RowSet>) 0x000006 0x000006 0x000006 0x01 (System Collections Generic Key/Value9airclong, Pyrtha.Level4.RowS. (18)7152027541081854, OrdereaRowSet @0(@3,@4,@5) key (@3) or (18)074503224792840, BaletSxp80xvSet %0(@3,@4,@5) key (@3) or (18)074503224792841, RestRowSet %0(@3,@4,@5) key (@3) in (18)074503224792841, JoniRowSet %10(@1,@1,01,%) #12(3) key (18)074503224792844, JoniRowSet %10(@1,%11,%11,%12) key.(%13) (18)074503224792844, JoniRowSet %10(@1,%11,%11,%12) key.(%13) (18)074503224792844, JoniRowSet %10(@1,%11,%11,%12) key.(%13) (18)074503224792844, JoniRowSet %10(@1,%11,%11,%11,%11,%11,%11,%11,%11,%11,%	Pyrrho.Common.Bu int byte System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G Iong	 ↓ < C^a Transac ↓ < C^a Transac ↓ < C^a Transac ↓ < C^a User.cs ↓ < C^a Value.cc ↓ < C^a Value.cc ↓ < C^a Value.cc ↓ < C^a Contex ↓ < C^a Group. ↓ < C^a Join.cs
	(Pyrtho.Level4.RowSet>) 0x0000006 0x07 (System.Collections.Generic.KeyValuePair <long.pyrtho.level4.rows (16971530072411081854, OrderedRowSet @8(@3,@4,@5) key (@3) or (18070450322473284, DateExpRowSet \$3(@4,@10,30,31,#12)\$13) (18070450322473284, DateExpRowSet \$3(04,@10,30,31,#12)\$13) (180704503224732844, Contexest \$4(94,@10,30,31,#12)\$13) (180704503224732844, Contexest \$5(14,31,31,#12) key (%3) (180704503224732844, OrderedRowSet \$10(\$41,31,11,21) key (%3) (180704503224732844, OrderedRowSet \$10(\$41,31,11,21) key (%3) (180704503224732845, OrderedRowSet \$10(\$41,31,11,21) key (\$43) (180704503224732847, OrderedRowSet \$10(\$41,31,11,21) key (\$43) (180704503224732847, OrderedRowSet \$10(\$41,31,11,21) key (\$43) (180704503224732847, OrderedRowSet \$10(\$42,81,31,121) key (\$43) (180704503224732847, OrderedRowSet \$10(\$42,81,31,121) key (\$450) 0,70000000000012</long.pyrtho.level4.rows 	Pyrrho.Common.Bu int byte System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G Iong	 ↓ < c= Tansac ↓ < c= Trigger. ↓ < c= Value.c: ↓ < c= Contex ↓ < c= Group. ↓ < c= Group. ↓ < c= Mergefi

%18 is an OrderedRowSet, with source %9. Continue with %15.

Command Prompt - Pyrrhod	an sana lang lang		
265 PColumn3 K for 216(1)	*Untitled - Notepad		- 🗆 X
289 PColumn3 M for 216(2) File	Edit Format View Help		
	lectRowSet #7(#8,#10,#12) targets: 23=%9,216=%12 Source:		A. No. 444 Mag
	<pre>bleExpRowSet #14(#8,#10,#12 %0,%1) key (#8,#10) where (# yea, %10)</pre>	#28) matches (#8=1) targets	s: 23=%9,216=%12
	10,#12,%0,%1 %13) key (#8,#10,%0,%1,	- //-	
rderedRowSe	t %18 targets: 23=%9,216=%12 JoinCond: (%1	Source: %9	
	eredRowSet %18(#8,#10,%0) key (#8) order (#8) targets:	22-%9 Sources %9)	
C:\PyrrhoDB70\Pyrrho>Pyrrh SQL> select e,f,m from v w {Ord	leredRowSet %15(%13,%1,#12) key (%13) order (%13) targets:	25=%9 Source: %9} ts: 216=%12 Source: %12}	
SQL> select e, +, m +rom v w core	() () () () ()	,	
rderedRowSe	+ %15 ug Test Analyze Tools Extensions	Source: %1	2
i dei eurowse		Source. ¹	. 2
X and the Arm Blie			
0.0 8.0 9	$\neg \qquad \frown \qquad \bigcirc \qquad \bigcirc$	° ● * ÷	B Live Share
and the second se			- Decomposition of the second
Context.cs Watch 1 -= ×			Solution Explorer 🛛 🔻 🕂
Search (Ctrl+E)	D ▼ ← → Search Depth: 3 ▼ II Ba		00 🔂 🐺 '0-
Name	Value	Type	County Columbus Country O
€ Count	0x000000000000011	long	
🖌 🧼 root	{Pyrrho.Common.Inner <long, pyrrho.level4.rowset="">}</long,>	long Pyrrho.Common.Bu	Þ ✓ Cª ReadCons
		long	▷ ✓ C™ ReadCons ▷ ✓ C™ Role.cs
✓ ✓ root	{Pyrrho.Common.Inner <long, pyrrho.level4.rowset=""> } 0x00000002 0x02</long,>	long Pyrrho.Common.Bu int byte	 ▷ ✓ C^m ReadCons ▷ ✓ C^m Role.cs ▷ ✓ C^m Table.cs
 ✓ coot ✓ EndPos ✓ count ✓ gtr 	{Pyrrho.Common.Inner <long, pyrrho.level4.rowset=""> } 0x0000002 0x02 (Pyrrho.Common.Leaf<long, pyrrho.level4.rowset=""> }</long,></long,>	long Pyrrho.Common.Bu int byte Pyrrho.Common.Bu	 ▷ ✓ C[#] ReadCons ▷ ✓ C[#] Role.cs ▷ ✓ C[#] Table.cs ▷ ✓ C[#] TableColu
✓ ☞ root	(Pyrrho.Common.Inner <long, pyrrho.level4.rowset="">} 0x0000005 0x02 (Pyrrho.Common.Leaf<long, pyrrho.level4.rowset="">} 0x0000066</long,></long,>	long Pyrrho.Common.Bu int byte Pyrrho.Common.Bu int	 ▷ ✓ C^a ReadCons ▷ ✓ C^a Role.cs ▷ ✓ C^a Table.cs ▷ ✓ C^a TableColu ▷ ✓ C^a Transaction
✓ I root ↓ EndPos I count I gtr ↓ EndPos I count	(Pyrrho.Common.Inner <long, pyrrho.level4.rowset=""> } 0x0000002 0x02 (Pyrrho.Common.Leaf<long, pyrrho.level4.rowset=""> } 0x0000006 0x07</long,></long,>	long Pyrrho.Common.Bu int byte Pyrrho.Common.Bu int byte	 C= ReadCons C= Role.cs C= Table.cs C= TableColu C= Transactio C= Trigger.cs
 ✓ root ✓ EndPos ✓ count ✓ gtr ✓ EndPos ✓ count ✓ count ✓ slots 	(Pyrrho.Common.Inner <long. pyrrho.level4.rowset="">) 0x0000002 0x02 (Pyrrho.Common.Leaf<long. pyrrho.level4.rowset=""> } 0x0000006 0x07 (Sustan Collections</long.></long.>	long Pyrrho.Common.Bu byte Pyrrho.Common.Bu int byte	 V C* ReadCons V C* Role.cs V C* Table.cs V C* TableColu V C* TableColu V C* Transactio V C* Trigger.cs V C* User.cs
	(Pyrrho.Common.Inner <long. pyrrho.level4.rowset="">) 0x000002 (Pyrrho.Common.Leaf<long. pyrrho.level4.rowset="">) 0x007 (System.Collection: ((691752902764108) RestROWSet</long.></long.>	long Pyrrho.Common.Bu int byte Pyrrho.Common.Bu int byte ons.G ons.G	 C= ReadCons C= Role.cs C= Table.cs C= Table.clu C= Transactio C= Transactio C= C= Transactio C= User.cs C= Value.cs
	(Pyrtha Common Inner-Long, Pyrtha Level4.RowSet>) 0x02 0x02 (Pyrtha Common Leaf < long, Pyrtha Level4.RowSet>) 0x02 0x07 (System Collection ((691752902744108 ((8074505324792)	long Pyrrho.Common.Bu int byte Pyrrho.Common.Bu int byte ons.G ons.G ons.G	 C = ReadCons C = Role.cs C = Table.cs C = Table.cs C = Table.club C = Transaction C = Transaction C = Transaction C = User.cs C = Value.cs C = View.cs
	(Pyrrho.Common.Inner <long. pyrrho.level4.rowset="">) 0x0000002 0x02 (Pyrrho.Common.Leaf<long. pyrrho.level4.rowset="">) 0x000006 0x07 (System.Collection) (1691752902764108) (18070405352247922841, RestRowSet %)(8,e10,%0) key (e8,e10</long.></long.>	long Pyrrho.Common.Bu int byte Pyrrho.Common.Bu int byte ons.G ons.G ons.G 0,%0) System.Collections.G.	 C ReadCons C RealCons C Role.cs C Table.cs C Table.cs C Table.ci <l< td=""></l<>
✓ root ✓ fatPos ✓ count ✓ gtr ✓ gtr ✓ gtr ✓ count ✓ gtr ✓ count ✓ count ✓ count ✓ count ✓ count ✓ fatPos ✓ fatPos ✓ fatPos ✓ count ✓ fatPos ✓ fatPos ✓ count ✓ fatPos ✓ count ✓ fatPos ✓ fatPos ✓ count ✓ fatPos ✓ fatPos ✓ fatPos ✓ count ✓ fatPos ✓	(Pyrtha Common.Inner-Long. Pyrtha.Level4.RowSet>) 0x000 0x02 (Pyrtha.Common.Leaf <long. pyrtha.level4.rowset="">) 0x02 (Pyrtha.Common.Leaf<long. pyrtha.level4.rowset="">) 0x02 (System.Collection) (169175202744108) (18070405532247922841) (1807040553247922841, RestRowSet %6(P\$,#10,%0) key (#8,#11) (1807040553247922842, JoinRowSet %6(P\$,#10,%0) key (#8,#11)</long.></long.>	leng Pyrtho.Common.Bu int byte Pyrtho.Common.Bu int byte %9 ons.G ons.G ons.G ons.G ons.G ons.G ons.G ons.G ons.G ons.G	 ↓ < <p>C™ ReadCons ↓ < <p>C™ Role.cs ↓ < <p>C™ Table.cs ↓ < <p>C™ Table.clu ↓ < <p>C™ Table.clu ↓ < <p>C™ Transactio ↓ < <p>C™ Transactio ↓ < <p>C™ Usite.cs ↓ < <p>C™ Value.cs ↓ < <p>C™ Value.cs ↓ < <p>C™ Value.cs ↓ < <p>Level4 ↓ < <p>C Activation </p> </p></p></p></p></p></p></p></p></p></p></p></p>
	(Pyrrho.Common.Inner-long.Pyrrho.Level4.RowSet>) 0x000000 0x001 (Pyrrho.Common.Leaf <long.pyrrho.level4.rowset>) 0x000 0x07 (System.Collection) (69175300276100) (18070459532479204) (18070459532479204) (180704595324792044) (18070459524792044)</long.pyrrho.level4.rowset>	leng Pyrtho.Common.Bu int byte Pyrtho.Common.Bu int byte %9 ons.G ons.G ons.G ons.G ons.G ons.G ons.G ons.G ons.G ons.G	 c = Tablecs c = TableColu c = TableColu c = Tansactio c = Tringarcs c = Valuecs c = Valuecs c = Valuecs c = Level4 c = Activation c = Context.c
root v fatPas count v gtr v fatPas v count v fatPas v count v fatus v tots v v tots v tots v tots v v v tots v tots v v v v tots v	(Pyrtha CommonLeaf <long, pyrtha.level4.rowset="">) 0:000000 0:02 (Pyrtha CommonLeaf<long, pyrtha.level4.rowset="">) 0:00000 0:07 (System Collection (160704503224779284), RestRowSet %5(年8,年10,%0) (ey (年8,年1 1(80704503224779284), JonRowSet %5(10,%1), JonRowSet %5(10,\%1), JonRow</long,></long,>	leng Pyrho.Common.Bu int byte Pyrho.Common.Bu int bote ons.G ons.G ons.G ons.G 0,550 System.Cellections.G (%13) System.Cellections.G	 / C* ReadCont / C* Roless / C* TableCst / C* Tigger.cs / C* User.cs / C* User.cs / C* View.cs / C* View.cs / C* Context.c / C* Context.c / C* Context.c
	(Pyrrha-Common.Inner-long. Pyrrha-Leviel4.RowSet>) 0x00002 (Pyrrha-Common.Leaf <long. pyrrha-leviel4.rowset="">) 0x007 (System.Collection (169175302745102) (169175302745102) (16917530224792) (16917530224792) (16917530224792) (16917530224792) (16917530224792) (16917530224792) (16917530224792) (16917530224792) (16917530224792) (16917530224792) (16917530224792) (16917553024792) (16917553024792) (16917553024792) (16917553024792) (16917553024792) (16917553024792) (1691755324792) (1691755524792) (16917555324792) (1691755524792) (1691755524792) (1691755524792) (1691755524792) (1691755524792) (1691755524792) (1691755524792) (1691755524792) (16917555524792) (16917555524792) (16917555524792) (16917555524792) (16917555524792) (16917555524792) (16917555524792) (16917555524792) (16917555524792) (1691755555755555555555555555555555555555</long.>	leng Pyrho.Common.Bu int byte Pyrho.Common.Bu int bote ons.G ons.G ons.G ons.G 0,550 System.Cellections.G (%13) System.Cellections.G	 / c* ReadCons / c* Roless / c* TableCs / c* Usencs / c* Usencs / c* Valuecs / c* Contextc / c* Contextc / c* Group.cs / c* Group.cs / c* Group.cs
	(Pyrtha Common.Lear(long, Pyrtha.Level4.RowSet>) 0x000 (Pyrtha Common.Lear(long, Pyrtha.Level4.RowSet>) 0x000 (Pyrtha Common.Lear(long, Pyrtha.Level4.RowSet>) 0x000 0x01 (System.Collection) (169175300274100) (180704053224792244) (18070405324792244) (18070405324792244) (18070405324792244) (18070405324	leng Pyrho.Common.Bu int byte Pyrho.Common.Bu int bote ons.G ons.G ons.G ons.G 0,550 System.Cellections.G (%13) System.Cellections.G	 c ReadCont c Roless c TableCst c TableCst
	(Pyrthe.Common.Inner-long.Pyrthe.Level4.RowSet>) 0x000000 0x002 (Pyrthe.Common.Leaf <long.pyrthe.level4.rowset>) 0x000006 0x07 (System.Collection (80704053224792841, RestRowSet \$) (80704553224792841, RestRowSet \$) (80704553224792841, RestRowSet \$) (80704553224792841, RestRowSet \$) (80704553224792841, RestRowSet \$) (80704553224792844, SelectedRowSet \$) (8070455324792844, SelectedRowSet \$)</long.pyrthe.level4.rowset>	Ing Pyrho.Common.Bu int byte Pyrho.Common.Bu int byte ons.G.	 / c* ReadCons / c* Roless / c* TableCs / c* Usencs / c* Usencs / c* Valuecs / c* Contextc / c* Group.cs / c* Group.cs / c* Group.cs / c* MergeRes / c* MergeRes / c* MergeRes
	(Pyrtha Common.Lear(long, Pyrtha.Level4.RowSet>) 0x000 (Pyrtha Common.Lear(long, Pyrtha.Level4.RowSet>) 0x000 (Pyrtha Common.Lear(long, Pyrtha.Level4.RowSet>) 0x000 0x01 (System.Collection) (169175300274100) (180704053224792244) (18070405324792244) (18070405324792244) (18070405324792244) (18070405324	Ing Pyrho.Common.Bu int byte Pyrho.Common.Bu int byte ons.G.	 c ReadCont c Roless c TableCst c TableCst

It is time for a short cut here: %18 has source %9 which we can see is a RestRowSet, and %15 has source %12, which is a SelectedRowSet. So let's pick them up later, at the end of the Review. Click Continue.



This is the breakpoint at the end of Review. Return to the Watch window.

Command Prompt - Pyrrhol			-
265 PColumn3 K for 216(1	*Untitled - Notepad		- 🗆 🗙
289 PColumn3 M for 216(2	File Edit Format View Help		
313 PTransaction for 2 R			
331 Record 331[216]: 223 366 Record 366[216]: 223	{TableExpRowSet #14(#8,#10,#12 %0,%1) key (#8,#10) where (#28) m Source: %10}	atches (#8=1) target:	s: 23=%9,216=%12
401 PTransaction for 1 R	Source: %10} {JoinRowSet %10(#8,#10,#12,%0,%1 %13) key (#8,#10,%0,%1,#12,%13)	where (#28) matches	(#8=1) matching
419 PView V 419 select *	(#8=(%13),%13=(#8)) targets: 23=%9,216=%12 JoinCond: (%17) match		
50L> ^C	%15}	-	
C:\PyrrhoDB70\Pyrrho>Pyrrl	{OrderedRowSet %18(#8,#10,%0) key (#8) order (#8) targets: 23=%9		
SQL> select e,f,m from v v	{OrderedRowSet %15(%13,%1,#12) key (%13) order (%13) targets: 21	6=%12 Source: %12}	
File Edit View Gi	t Project Build Debug Test Analyze Tools Extensions Window	Search Pyrrho	oSvr — 🗆
Help			
10-0 R-0 H	り - で - 🕨 Continue - 🔤 🙆 _ 🔳 🗿 - 🕂 😨 - 🖬 -	-	A Live Share
x = +1 = = = = =		•	
Context.cs Watch 1 🕫	×		Solution Explorer - 4
Search (Ctrl+E)	P - ← → Search Depth: 3 - 📲 👪		000 🖓 🗐 '0-
Name	Value	Туре	Search Solution Explor
🔎 Count	0x00000000000011	long	▶ ✓ C ⁼ ReadCor
🔺 🤗 root	{Pyrrho.Common.Inner <long, pyrrho.level4.rowset="">}</long,>	Pyrrho.Common.Bu	P ✓ C= Role.cs
🖋 EndPos	0x00000002	int	P ✓ C [*] Table.cs
 count 	0x02	byte	↓ C= TableCol
⊿ 🤗 gtr	{Pyrrho.Common.Leaf <long, pyrrho.level4.rowset="">}</long,>	Pyrrho.Common.Bu	▷ ✓ C= Transact
	+= 0x0000006	int	↓ C= Trigger.c
🖋 EndPos		byte	b ✓ C= User.cs
🥥 count	0x07		
🧉 count 4 🤤 slots	{System.Collections.Generic.KeyValuePair <long, pyrrho.level4.rows<="" td=""><td></td><td>b / C= Value cs</td></long,>		b / C= Value cs
 ✓ count ✓ 🖨 slots ▶ 🛩 [0] 	{System.Collections.Generic.KeyValuePair <long, pyrrho.level4.rows<br="">{[6917529027641081864, OrderedRowSet @8(@3,@4,@5) key (@3) or</long,>	System.Collections.G	↓ C= Value.cs
	(System.Collections.Generic.KeyValuePair <long, pyrrho.level4.rows<br="">{[6917529027641081864, OrderedRowSet @8(@3,@4,@5) key (@3) or {[8070450532247928840, TableExpRowSet %8(#8,#10,%0,%1,#12]%13)</long,>	System.Collections.G System.Collections.G	Þ ✓ C= View.cs
 ✓ count ✓ slots ✓ [0] ✓ [1] ✓ [2] 	(System.Collections.Generic.KeyValuePair <long. pymho.level4.rows<br="">(1691752002764103164, OrderedRowSet 登(空急)を4(金) key (受) or (18070450532247928840, TableExpRowSet %8(年8,=10,%0)、%1 = F121% 3) (18070450532247928841, RestRowSet %8(年8,=10,%0) key (生素=10,%0)</long.>	System.Collections.G System.Collections.G System.Collections.G	Þ ✓ C= View.cs ▲ 📄 Level4
Count Count	(System, Callection, Generic KeylaineBair (ong, PyrhoLevel & Book, (16917530027641081864, OrderedRowSet 186(年3日(次の気)) (1807045052247782844)、TableExpRowSet 186(年3日(次の気)), (1807045052247782844)、TableExpRowSet 186(年3日(次の気)), (1807045052247782844)、TableExpRowSet 186(年3日(次の)), (1807045052247782844)、TableExpRowSet 186(年30), (1807045052247782844)、TableExpRowSet 186(年30), (1807045052247782844)、TableExpRowSet 186(年30), (1807045052247782844)、TableExpRowSet 186(年30), (1807045052247782844)、TableExpRowSet 186(年30), (1807045052247782844)、TableExpRowSet 186(年30), (1807045052247782844)、TableExpRowSet 186(186), (1807045052247782844), (1807045052247782844), (1807045052247782844), (1807045052247782844), (1807045052247782844), (180704505247782844), (1807045052247782844), (18070450524782844), (18070450524782844), (18070450524782844), (18070450544), (18070450544), (18070450544), (18070450544), (18070450544), (18070450544), (18070450544), (18070450544), (18070450544), (18070450544), (18070450544), (18070450544), (18070450544), (18070450544), (18070450544), (180704505454),	System.Collections.G System.Collections.G System.Collections.G System.Collections.G	 ▷ ✓ C= View.cs ▲ □ Level4 ▷ ✓ C= Activation
 ✓ count ✓ slots ✓ [0] ✓ [1] ✓ [2] 	(System.Collections.Generic.KeyValuePair <long. pymho.level4.rows<br="">(1691752002764103164, OrderedRowSet 登(空急)を4(金) key (受) or (18070450532247928840, TableExpRowSet %8(年8,=10,%0)、%1 = F121% 3) (18070450532247928841, RestRowSet %8(年8,=10,%0) key (生素=10,%0)</long.>	System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G	 ▷ ✓ C= View.cs ✓ C= View.cs ✓ C= Level4 ▷ ✓ C= Activatio ▷ ✓ C= Context.
	(System:Collections:Generic:KeyValuePair <long, pyrrhol.evel4.rows<br="">(16917530027411081064, OrteetRowSet 8(96年301、%)(18)(18)(18)(18)(18)(18)(18)(18)(18)(18</long,>	System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G	
	[5ytem.Collections.Genetic.Key/bluePaire long. PyrhoLeve4.Rook. (169175302741081864, OrderedRooket 86(49,51,64,46), key (49) ro. (1807045052247782840, TableExpRooket 86(49,51,63,65),17,17,1513), (1807045052247782840, TableExpRooket 86(49,51,63,65),17,17,1513), (1807045052247782842, Jonifoxofet 91,106,46,115,155,17,12153) key, (1807045052247782844, SelectedRooket 815(15),15,15,17,12153) key, (1807045052247782844, SelectedRooket 815(15),15,15,17,12153) key, (1807045052247782844, SelectedRooket 815(15),15,15,17,12153) key, (1807045052247782844, SelectedRooket 815(15),15,15,17,12158) key, (15),25,27,12153) key, (15),27,25,27,27,25) key, (15),25,27,27,25) key, (15),25,27,27,27) key, (15),25,27,27,27) key, (15),25,27,27) key, (15),27,27) key,	System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G long	 V c= View.cs Level4 V c= Activatio V c= Context. V c= Group.cs V c= Join.cs
count solds b ○ [0] b ○ [1] b ○ [2] b ○ [3] b ○ [4] c [5] fey	[57]rtem.Collections.Generic.Key/bluePair-long.PyrhoLeve4.Roox- [6977530274101645, OrderedRoox10 et 69(4):3 (4) (5) (4) (6) (5) [18070450532247728840, TableExpRowSet %8[49]#10,%0,%1,#12[513], [18070450532247728840, TableExpRowSet %8[49]#10,%0,%1,#12[513], [18070450532247728842, JondfowSet %10(49]#10,%0,%1,#12[513], [18070450532247782844, SelecteRoox5et %12[513]%1,#12[513], [18070450532247782844, SelecteRoox5et %15[513]%1,#12[1513], [18070450532247782844, SelecteRoox5et %15[515]%1,%1,#12],bey (%13, [18070450532247782844, SelecteRoox5et %15[515]%1,%1,#12],bey (%13, [1807045052247782844, SelecteRoox5et %15[515]%1,%1,#12],bey (%13, [180704505224782844, SelecteRoox5et %15[515]%1,%1,#12],bey (%13, [180704505224782844, SelecteRoox5et %15[515]%1,%1,#12],bey (%13, [180704505224782844, SelecteRoox5et %15[515]%1,%1,#12],bey (%13, [18070450524782844, SelecteRoox5et %15[515]%1,#12],bey (%13, [1807045052478244, SelecteRoox5et %15[515]%1,#12],bey (%13, [180704505247844, SelecteRoox5et %15[51	System.Collections.G System.Collections.G System.Collections.G System.Collections.G System.Collections.G long	
	[57]rtem.Collections.Generic.Key/bluePair-long.PyrhoLeve4.Roox- [6977530274101645, OrderedRoox10 et 69(4):3 (4) (5) (4) (6) (5) [18070450532247728840, TableExpRowSet %8[49]#10,%0,%1,#12[513], [18070450532247728840, TableExpRowSet %8[49]#10,%0,%1,#12[513], [18070450532247728842, JondfowSet %10(49]#10,%0,%1,#12[513], [18070450532247782844, SelecteRoox5et %12[513]%1,#12[513], [18070450532247782844, SelecteRoox5et %15[513]%1,#12[1513], [18070450532247782844, SelecteRoox5et %15[515]%1,%1,#12],bey (%13, [18070450532247782844, SelecteRoox5et %15[515]%1,%1,#12],bey (%13, [1807045052247782844, SelecteRoox5et %15[515]%1,%1,#12],bey (%13, [180704505224782844, SelecteRoox5et %15[515]%1,%1,#12],bey (%13, [180704505224782844, SelecteRoox5et %15[515]%1,%1,#12],bey (%13, [180704505224782844, SelecteRoox5et %15[515]%1,%1,#12],bey (%13, [18070450524782844, SelecteRoox5et %15[515]%1,#12],bey (%13, [1807045052478244, SelecteRoox5et %15[515]%1,#12],bey (%13, [180704505247844, SelecteRoox5et %15[51	System.Collections.G., System.Collections.G., System.Collections.G., System.Collections.G., System.Collections.G., System.Collections.G., long Pyrrho.Level4.RowSe.,	 C View.cs Level4 C C Activation C C Group.cs C C Group.cs C C Group.cs C MergeR C ParseCla
	(System.Collections.Generic.KeyValuePair-long.Pyrhol.zev4840xx3. (1691753027541081864, OrderedNox6et %0(43, \$4, \$4, \$5), key (\$8) or (180174530224782884), Tablets:pRox6et %16(48, \$10, \$50, \$11, \$12), \$13). (1801745302224782884), Tablets:pRox6et %16(48, \$10, \$50, \$11, \$12), \$13). (1801745302224782884), Tablets:pRox6et %16(48, \$10, \$50, \$11, \$12), \$15]. (180174530224782884), Tablets:pRox6et %16(48, \$10, \$50, \$12), \$12], \$13]. (180174530224782884), Tablets:pRox6et %16(48, \$13, \$16, \$12), \$15]. (180174530224782884), Tablets:pRox6et %16(48, \$13, \$16, \$12), \$15]. (180174530224782884), Tablets:pRox6et %16(58, \$13, \$16, \$12), \$15]. (180174553224782884), Tablets:pRox6et %16(58, \$13, \$16, \$12), \$16]. (180174553224782884), Tablets:pRox6et %15(\$13, \$16, \$12), \$16]. (180174553224782884), SelectedRowSet %15(\$13, \$16, \$12), \$16]. (180174553224782884), SelectedRowSet %15(\$13, \$16, \$12), \$16]. (180174553224782884), SelectedRowSet %15(\$13, \$16, \$12), \$16]. (18017455324782884), SelectedRowSet %15(\$13, \$16, \$12), \$16]. (18017455324782884), SelectedRowSet %15(\$13, \$16, \$12]. (18017455324782884), SelectedRowSet %15(\$16, \$16, \$12]. (19018455324782884), SelectedRowSet %15(\$16, \$16, \$12]. (19018455324782884), SelectedRowSet %15(\$16, \$16, \$12]. (1901845532478284), SelectedRowSet \$15, \$16, \$12]. (190184553	System.Collections.G., System.Collections.G., System.Collections.G., System.Collections.G., System.Collections.G., System.Collections.G., long Pyrrho.Level4.RowSe.,	 > < C* View.cs > Level4 > < C* Activatic > < C* Context. > < C* Group.ct > < C* Join.cs > < C* MergeRe

Visual Studio reports changes to all of these RowSets. r will also be out of date, so repeat the above extraction from the data tree, starting with the modified SelectRowSet #7 (scroll down for it).

Command Prompt - Pyrrhod			
265 PColumn3 K for 216(1)	Untitled - Notepad		- 🗆 X
289 PColumn3 M for 216(2)	File Edit Format View Help		
313 PTransaction for 2 Ro 331 Record 331[216]: 223=	{SelectRowSet #7(#8,#10,#12) targets: 23=%9,216=%12 Source: #14}	4 100 10 10 10	
366 Record 366[216]: 223=	{TableExpRowSet #14(#8,#10,#12 %0,%1) key (#8,#10) where (#28) mat Source: %10}	ches (#8=1) targets	: 23=%9,216=%12
401 PTransaction for 1 Ro	{JoinRowSet %10(#8,#10,#12,%0,%1 %13) key (#8,#10,%0,%1,#12,%13) v	where (#28) matches	(#8=1) matching
419 PView V 419 select *	(#8=(%13),%13=(#8)) targets: 23=%9,216=%12 JoinCond: (%17) matchin		
SQL> ^C	%15}		
C:\PyrrhoDB70\Pyrrho>Pyrrh	<pre>{OrderedRowSet %18(#8,#10,%0) key (#8) order (#8) targets: 23=%9 5 {OrderedRowSet %15(%13,%1,#12) key (%13) order (%13) targets: 216=</pre>		
SQL> select e,f,m from v w	(ordereukowset als(als)als#12) key (als) order (als) targets. 210-	wiz source. wizy	
M File Edit View Git	Project Build Debug Test Analyze Tools Extensions Window		
File Edit View Git Help	Project build bebug lest Analyze loois extensions window	SearchP Pyrrho	Svr — 🗆
G - O 🗄 - 🍰 🔛 🔐	"> - (" - ▶ Continue - 🔍 🐼 🚽 🗉 = 👌 → ‡ 😨 * ,	Ŧ	🖻 Live Share
South Street Str		200 200	
Context.cs Watch 1 🕫	×	÷ 0	Solution Explorer 🛛 🕈 🖗
Search (Ctrl+E)	$P \cdot \leftarrow \rightarrow$ Search Depth: 3 \cdot \overrightarrow{P}		004.10.
Name	Value	Туре 🔺	Search Solution Explor
🕨 🥥 Non-Pub	lic members		↓ C= ReadCor
Þ 🤗 [6]	{[8070450532247928850, RestRowSet %18(#8,#10,%0) key (#8,#10,%0)	System.Collections.G	↓ C= Role.cs
🤗 total		long	A C [■] Table.cs
🔺 🤪 slots	{System.Collections.Generic.KeyValuePair <long, pyrrho.common.bu<="" td=""><td></td><td>A C= TableCo</td></long,>		A C= TableCo
▶ ● [0]		System.Collections.G	♦ ✓ C [®] Transact
4 🤗 [1]	{[6917529027641081863, Pyrrho.Common.Leaf'2[System.Int64,Pyrrho 0x600000000000007	System.Collections.G	Þ ✓ C= Trigger.o
Key نکر Value نکر م		Pvrrho.Common.Bu	Þ ✓ C= User.cs
S EndPos		int	♦ ✓ C= Value.cs
e count		byte	↓ ✓ C* View.cs
A 😭 slots	{System.Collections.Generic.KeyValuePair <long, pyrrho.level4.rows<="" td=""><td></td><td>🖌 📄 Level4</td></long,>		🖌 📄 Level4
Þ 🤗 [0]	{[437, JoinRowSet 437(@0,@1,@2,@4,@5]@3) matching (@0=(@3),		▷ ✓ C= Activation
Þ 🖌 [1]	{[450, SelectedRowSet 450(@3,@4,@5) targets: 216=450 Source: 246 S		Þ ✓ C= Context
Þ 🤗 [2]	{[5764607523034234887, SelectRowSet #7(#8,#10,#12) targets: 23= %1	System.Collections.G	♦ ✓ C= Group.c
Þ 🤗 [3]	{[5764607523034234894, TableExpRovided #14(#8.#10.#121%0.%1) key	System Collections G	h . ce lain as
P 🗣 [5]	{[5764607523034234899, JoinRowSet [[5764607523034234887, SelectRov	//Set #/(#8,#10,#12) targets	s: 23= %18,216= %15 Source
4 @ [4]	y 0x5000000000013	long	Þ ✓ C= ParseCla
▲ 🥥 [4] ۶ Ke		Pyrrho.Level4.RowSe	h / CE Darrow et
ا € [4] لا الح لا الح			
ا € [4] لا الح لا الح	lue {JoinRowSet %10(#8,#10,#12,%0,%1 %13) key (#8,#10,%0,%1,#12,%1 pn-Public me	*	4

We see the changed version: copy its value to the Notepad and continue with its sources etc as we did before.

265 PColumn3 K for 216(1)	where (#28) matches(#	t8=1)
	RowSet #7(#8,#10,#12) targets: 23=%9,216=%12 Source: #14} xpRowSet #14(#8,#10,#12 %0,%1) key (#8,#10) where (#28) matches (#8=1) targets: 23=%9,2	16=%12
366 Record 366[216]: 223= Source:	X10}	
Joinko	wSet %10(#8,#10,#12,%0,%1 %13) key (#8,#10,%0,%1,#12,%13) where (#28) matches (#8=1) ma 3) <u>%13=(#8)) targets: 23=%</u> 9,216=%12 JoinCond: (%17) matching #8=%13 %13=#8 First: %18 5	
	Jizi (")) to be the start sources (") matching we will will we will be the start will be	econu.
C+\PycchoD870\Pyccho>Pycch	dRowSet %18(#8,#10,%0) key (#8) order %#8) targets: 23=%9 Source: %9}	
SQL> select e,f,m from V (Ordere	dRowSet %15(%13,%1,#12) key (%13) order (%13) targets: 216=%12 Source: %12}	
	RowSet #7(#8,#10,#12) targets: 23=%18,216=%15 Source: #14}	
Help Git View Git (TableE Source:	xpRowSet #14(#8,#10,#12 %0,%1) key (#8,#10) where (#28) matches (#8=1) targets: 23=%18, \$10}	216=%15
JoinRo	wSet %10(#8,#10,#12,%0,%1 %13) key (#8,#10,%0,%1,#12,%13) where (#28) matches (#8=1) ma	
(#8=(%1	3),%13=(#8)) targets: 23=%18,216=%15 JoinCond: (%17) matching #8=%13 %13=#8 First: %18	Second:
Context.cs Watch 1 + {RestRo	wSet %18(#8,#10,%0) key (#8,#10,%0) matches (#8=1) targets: 23=%18 http://localhost:818	0/4/4/D
Search (Ctrle E) RemoteC	ols:(66,100,122) RemoteGroups:}	
{Select	edRowSet %15(%13,%1,#12) key (%13,%1,#12) matches (%13=1) targets: 216=%15 Source: 246	SQMap:
d i gtr	5[289],%1=%15[265],%13=%15[223])}	
fr EndDes		
count Se	<pre>lectedRowSet %18 matches(%13=</pre>	:1)
a 😂 slots		- Idulecol
▶ ♥ [0]	{[6917529027641081864, OrderedRowSet @8(@3,@4,@5) key (@3) or System.Collections.G	
		✓ C= Transact
Þ 🛥 [1]	([8070450532247928840, TableExpRowSet %8(#8,#10,%0,%1,#12 %13) System.Collections.G	✓ C= Transact ✓ C= Trigger.c
	【18070450532247928840, TableExpRovSet %8(年巻日) %0,%1,早12 %13) System.Collections.G 【18070450532247928841, RestRowSet %8(年巻日) %0,%ks (考表日)%0, System.Collections.G 【18070450532247928842, InitinovSet %10(年美日) %0,%1,早12 %13) key System.Collections.G ↓	✓ C= Trigger.c ✓ C= User.cs
	[19070450532247928840, TableExpRowSet %8(#축.#10,%0,%1,#12]%13) System.Collections.G [18070450532247928844, RestRowSet %16(#축.#10,%0) key (#축.#10,%0) System.Collections.G [18070450532247928844, AnnoXet %10(#축.#10,%0),#12]%13) System.Collections.G [18070450532247928844, SelectedRowSet %12(%13,%1,#12) key (%13, System.Collections.G	 ✓ C= Trigger.c ✓ C= User.cs ✓ C= Value.cs
▶ ∞ (1) ▶ ∞ (2) ▶ ∞ (3) ▶ ∞ (4) ▲ ∞ (5)	(1007045053224792844)、TableExpRovSet %3(学者、1015%)が、第12(%15))、System Collections.G (1007045053224792844)、RestRovSet %3(学者、1016%)、1016(%1010%)・、System Collections.G (1007045053224792844)、SelectedRovSet %10(%3,1%12)(%3)、Bystem Collections.G (107045053224792844)、SelectedRovSet %10(%3,1%12)(%3)、Bystem Collections.G (107045053224792844)、SelectedRovSet %10(%3)、%14(2)(%3)(%3) System Collections.G (10704505324792844)、SelectedRovSet %10(%3)(%3,14)(2)(%3)(%3) System Collections.G (10704505324792844)、SelectedRovSet %10(%3)(%3,14)(2)(%3)(%3) System Collections.G	✓ C= Trigger.c ✓ C= User.cs ✓ C= Value.cs ✓ C= View.cs
	(1807045053224782884) TableExpRomset %8件巻き10 %0 %1 ≠12[%13] System.Collections.G (1807045053224782884). RestRowSet %9(年巻き10 %0) key (年季10 %0) System.Collections.G (1807045053224782884). Control %5 %10(年9) (130 %%) #12[%13] key System.Collections.G (1807045053224782884). SelectedRowSet %12(%13,%1 +12) key (%13 System.Collections.G (1807045053224782884). SelectedRowSet %12(%13,%1 +12) key (%13 System.Collections.G (1807045053224782884). SelectedRowSet %15(%13,%1 +12) key (%13 System.Collections.G (1807045053224782884). SelectedRowSet %15(%13,%1 +12) key (%13 System.Collections.G (1807045053224782884). SelectedRowSet %15(%13,%1 +12) key (%13 System.Collections.G (180704505324782884). SelectedRowSet %15(%13,%1 +12) key (%13 System.Collections.G (180704505324782847). SelectedRowSet %15(%13,%1 + 12) key (%13 System.Collections.G) (180704505324782847). SelectedRowSet %15(%13,%1 + 12) key (%13 System.Collections.G) (18070450532478847). SelectedRowSet %15(%13,%1 + 12) key (%13 Sys	 ✓ C= Trigger.c ✓ C= User.cs ✓ C= Value.cs ✓ C= View.cs → Level4
▶ ∞ (1) ▶ ∞ (2) ▶ ∞ (3) ▶ ∞ (4) ▲ ∞ (5)	(107045053224792884, TableExpRovGet %3(#8, #10, %0, %1, #12)%13) \$ystem. Collections.G (107045053224792884, Restrictions G (107045053224792884, Restrictions G (107045053224792884, Restrictions G	 C= Trigger.c C= User.cs C= Value.cs C= View.cs Level4 C= Activation
→ → ↓ → ↓ → ↓	(1007045053224782884)、TableExpRovset %3(学者:10 %5 %1; #12)%513 System Collections.G (1807045053224782884)、RestRowSet %3(学者:10 %5 %1; #12)%13)、wy. System.Collections.G (1807045053224782884)、SolicetedRowSet %10(学者:10 %5)、#12)%13) key System.Collections.G (1807045053224782884)、SolicetedRowSet %12(%13,%1,#12) key (%13 System.Collections.G (1807045053224782884)、SolicetedRowSet %15(%13,%1,#12) key (%13 System.Collections.G (1807045053224782884)、SolicetedRowSet %15(%13,%1,#12) key (%13 System.Collections.G (1807045053224782884)、SolicetedRowSet %15(%13,%1,#12) key (%13 System.Collections.G (1807045053224782884)、SolicetedRowSet %15(%13,%1,#12) key (%13 System.Collections.G (180704505324782884)、SolicetedRowSet %15(%13,%1,#12) key (%13 System.Collections.G (180704505324782845)	 ✓ C= Trigger.c ✓ C= User.cs ✓ C= Value.cs ✓ C= View.cs → Level4
→ 0 [1] → 0 [2] → 0 [3] → 0 [4] 4 0 [5] → 0 [4] → 0 [5] → 0 Key → 0 Value → 0 Non-Public memb 4 0 [6] → Key	(107045053224792884), TableExpRovSet %8(#8,#10,%0),#12(%13) System.Collections.G (107045053224792884), TableSxRowSet %8(#8,#10,%0), #12(%13), key System.Collections.G (107045053224792884), ConflowSet %8(#8,#10,%0), key (#8,#10,%0), System.Collections.G (107045053224792884), ConflowSet %10(%1,%1,%1,#12) key System.Collections.G (107045053224792884), SelectedRovSet %12(%1,%1,#12) key.(%13, System.Collections.G (107045053224792884), SelectedRovSet %12(%13,%1,#12) key.(%13, System.Collections.G (107045053224792884), SelectedRovSet %12(%13,%1,#12) key.(%13, System.Collections.G (107045053224792884), SelectedRovSet %12(%13,%1,#12) key.(%13, System.Collections.G (107045053224792884), SelectedRovSet %12(%13,%1,#12) matches (%13 PythoLevel4.RowSe (107045053224792884), SelectedRovSet %15(%13,%1,#12) key.(%13, System.Collections.G (107045053224792884), SelectedRovSet %18(#8,#10,%0), key.(#8,#10,%0), system.Collections.G (107045053224792884), SelectedRovSet %18(#8,#10,%0), key.(#8,#10,%0), system.Collections.G (107045053224792884), RestRovSet %18(#8,#10,%0), key.(#8,#10,%0), system.Collections.G (10704505324792850), RestRovSet %18(#8,#10,%0), key.(#8,#10,%0), system.Collections.G	 C= Trigger.c C= User.cs C= Value.cs C= View.cs Level4 C= Activation C= Context.
$ \begin{array}{c} \bullet & \bullet & [1] \\ \bullet & \bullet & [2] \\ \bullet & \bullet & [3] \\ \bullet & \bullet & [4] \\ \bullet & \bullet & [2] \\ \bullet & & \downarrow & \downarrow \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet &$	[107045053224792884]. @LibleCpRowSet %19(# 0, 10(%);#12(%)3) \$ystem. Collections.G [00704505324792884]. @LibleCpRowSet %19(# 0, 10(%);#12(%)3) \$ystem. Collections.G [007045053224792884]. @LibleCondersMost %19(# 0, 10(%); %12(%);13(%) \$ystem. Collections.G [007045053224792884]. @LibleCondersMost %15(%);13(%);12(%); %12(%);13(%) \$ystem. Collections.G [007045053224792884]. %LibleCondersMost %13(%);13(%);12(%);13(%), \$ystem. Collections.G [007045053224792884]. %LibleCondersMost %13(%);13(%);12(%);13(%), \$ystem. Collections.G [007045053224792884]. %LibleCondersMost %13(%);13(%);12(%);13(%), \$ystem. Collections.G [007045053224792884]. %LibleCondersMost %13(%);12(%);13(%), \$ystem. Collections.G [007045053224792884]. %LibleCondersMost %13(%);12(%);13(%), \$ystem. Collections.G [0070450532247928850]. %LibleCondersMost %13(%);12(%);12(%);13(%), \$ystem. Collections.G [0070450532247928850]. %LibleCondersMost %13(%);12(%);1	 C° Trigger.c C° User.cs C° Value.cs C° View.cs Level4 C° Activatio C° Context. C° Group.cs C° Join.cs C° MergeRe
	(107045053224792884, TabletxpRovGet %19/49, #10, %0, %1, #12/1%13) System Collections.G (107045053224792884, Parket %10, %0, %1, #12/1%13), key System Collections.G (107045053224792884, Parket&RovGet %10(49, #10, %0, %1, #12/1%13), key System Collections.G (107045053224792884, Parket&RovGet %10(49, #10, %0, %1, #12/1%13), key System Collections.G (107045053224792884, Parket&RovGet %12/%13, %1, #12/1 key (%13, System Collections.G Ing (107045053224792884, Parket&RovGet %15/%13, %1, #12/1 key (%13, System Collections.G Ing (107045053224792884, Parket&RovGet %15/%13, %1, #12/1 key (%13, System Collections.G Ing (107045053224792884, Parket&RovGet %18/(%13, %1, #12) matches (%13) Pythol.Level4.RovSe Ing (1070450532247928850, RestRovGet %18/(%13, %1, #12) matches (%13) Pythol.Level4.RovSe Ing (1070450532247928850, RestRovGet %18/(%13, %1, #12) matches (%13) System Collections.G Ing (1070450532247928850, RestRovGet %18/(%13, %1, #12) matches (%13) Ing Ing (1070450532247928850, RestRovGet %18/(%13, %1, #12) matches (%13) Ing Ing (1070450532247928850, RestRovGet %18/(%13, %1, #12) matches (%13) Ing Ing (1070450532247928850, RestRovGet %18/(%13, %1, #12) %0,	 C° Triggence C° User.cs C° Value.cs C° Value.cs C° View.cs Level4 C° Activation C° Context. C° Group.cs C° MergeRation C° ParseCla
→ 0 [1] → 0 [2] → 0 [3] → 0 [3] → 0 [3] → 0 [3] → ∞ Non-Public memb → ∞ Non-Public memb ↓ ∞ Non-Public memb ↓ ∞ Non-Public memb ↓ ∞ Non-Public memb	(107045053224792884, TabletxpRovGet %19/49, #10, %0, %1, #12/1%13) System Collections.G (107045053224792884, Parket %10, %0, %1, #12/1%13), key System Collections.G (107045053224792884, Parket&RovGet %10(49, #10, %0, %1, #12/1%13), key System Collections.G (107045053224792884, Parket&RovGet %10(49, #10, %0, %1, #12/1%13), key System Collections.G (107045053224792884, Parket&RovGet %12/%13, %1, #12/1 key (%13, System Collections.G Ing (107045053224792884, Parket&RovGet %15/%13, %1, #12/1 key (%13, System Collections.G Ing (107045053224792884, Parket&RovGet %15/%13, %1, #12/1 key (%13, System Collections.G Ing (107045053224792884, Parket&RovGet %18/(%13, %1, #12) matches (%13) Pythol.Level4.RovSe Ing (1070450532247928850, RestRovGet %18/(%13, %1, #12) matches (%13) Pythol.Level4.RovSe Ing (1070450532247928850, RestRovGet %18/(%13, %1, #12) matches (%13) System Collections.G Ing (1070450532247928850, RestRovGet %18/(%13, %1, #12) matches (%13) Ing Ing (1070450532247928850, RestRovGet %18/(%13, %1, #12) matches (%13) Ing Ing (1070450532247928850, RestRovGet %18/(%13, %1, #12) matches (%13) Ing Ing (1070450532247928850, RestRovGet %18/(%13, %1, #12) %0,	 C° Trigger.c C° User.cs C° Value.cs C° View.cs Level4 C° Activatio C° Context. C° Group.cs C° Join.cs C° MergeRe

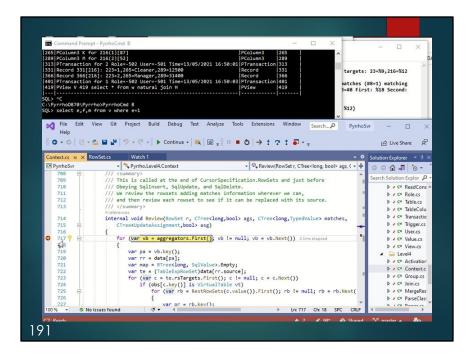
We see that a number of rowsets that were there before have simply been removed: there is no need for %9 or %12 and the OrderedRowSets %18 and %15 have been replaced by RestRowSet %18 and SelectedRowSet %15.

Importantly, the where condition e=1 (where (#28) matches (#8=1)) has been analysed.

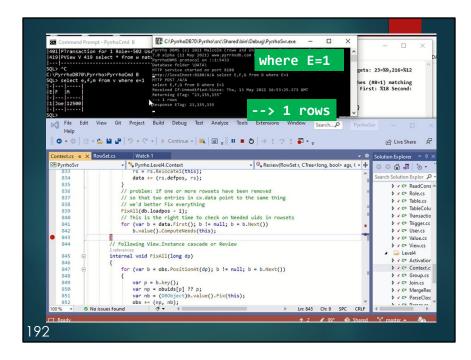
The Review process has noticed that the where-condition collapses both sides of the join to a single row, so that the orderedrowsets are not required.

And the RestRowSet now has the matches condition #8=1 which will be included in the HttpRequest to the remote database.

Click Continue.

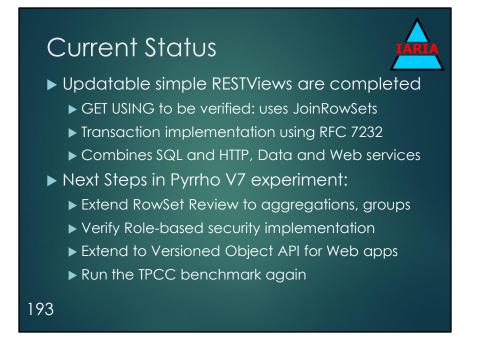


During the HttpService, our breakpoint gets hit again. Click Continue twice (before the timeout!),



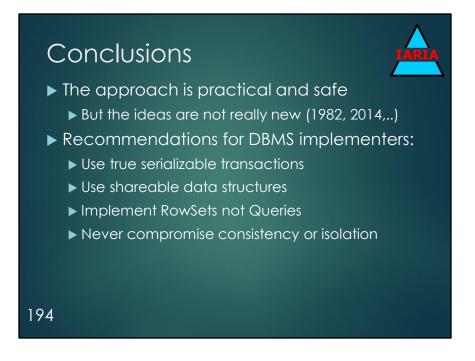
We get the results of course, but importantly we see that only one row was requested from the remote database.

(More generally, aggregations can also result in only one row being returned.) This concludes the demonstration of RowSet Review.



Pyrrho V7 is still at the alpha stage of development, and as this tutorial demonstrates, has a working implementation of simple updatable RESTViews. Inserts and Deletes have been implemented and tested. There is a more complex RESTView mechanism that managed a set of views with similar properties, implemented as a join. This implementation is still to be verified at the time of preparation of this tutorial, but it should follow from the implementation of updatable joins.

Once this is done, the plan is to extend RowSet Review to aggregations and groups with a more comprehensive set of tests. Next will come verification of the role-based security implementation, and the versioned Web API.



Current conclusions from the work are that the approach is practical and safe, and builds on ideas that have been around for a long time, though not in mainstream DBMS implementation.

DBMS implementers should adopt some or all of the following recommendations: serializable transactions, sharable data structures, rowSets not queries; but never compromise consistency or isolation.



I welcome discussion, and will be happy to provide interactive sessions on request.

Further reading



- Crowe, M.K., Laux, F.: Reconsidering Optimistic Algorithms for Relational DBMS, DBKDA 2020
- Crowe, M.K., Matalonga, S., Laiho, M.: StrongDBMS: built from immutable components, DBKDA 2019
- Crowe, M.K., Fyffe, C: Benchmarking StrongDBMS, Keynote speech, DBKDA 2019
- The Pyrrho Manual, SourceIntro, source code on Github.com/MalcolmCrowe/ShareableDataStructures/PyrrhoV7alpha
- T. Krijnen, and G. L. T. Meertens, "Making B-Trees work for B". Amsterdam: Stichting Mathematisch Centrum, 1982, Technical Report IW 219/
- https://pyrrhodb/blogspot.com
 - ▶ Esp: 2017 posts on adapter functions, RESTView