www.DBTechNet.org

Education and Culture DG

Lifelong Learning Programme

DBTech EXT
Virtual Laboratory Workshop on Concurrency Control in Databases

Martti Laiho & Kari Silpiö
Haaga-Helia University of Applied Sciences

ADBIS 2010, Novi Sad, Serbia, 21-24.9.2010

Disclaimer
This project has been funded with support from the European Commission. This publication [communication] reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

www.DBTechNet.org

HAAGA-HELIA
University of Applied Sciences

Education and Culture DG
Lifelong Learning Programme

Kari Silpiö

Information Technology Department
HAAGA-HELIA University of Applied Sciences
Helsinki, Finland

DBTechNet *member since 1997*
DBTech EXT *member 2009-*

Workshop on Concurrency Control in Databases

How ?

- Using the free **DBTech Virtual Laboratory Workshop** (*available on the DBTechNet portal, http://dbtech.uom.gr*)

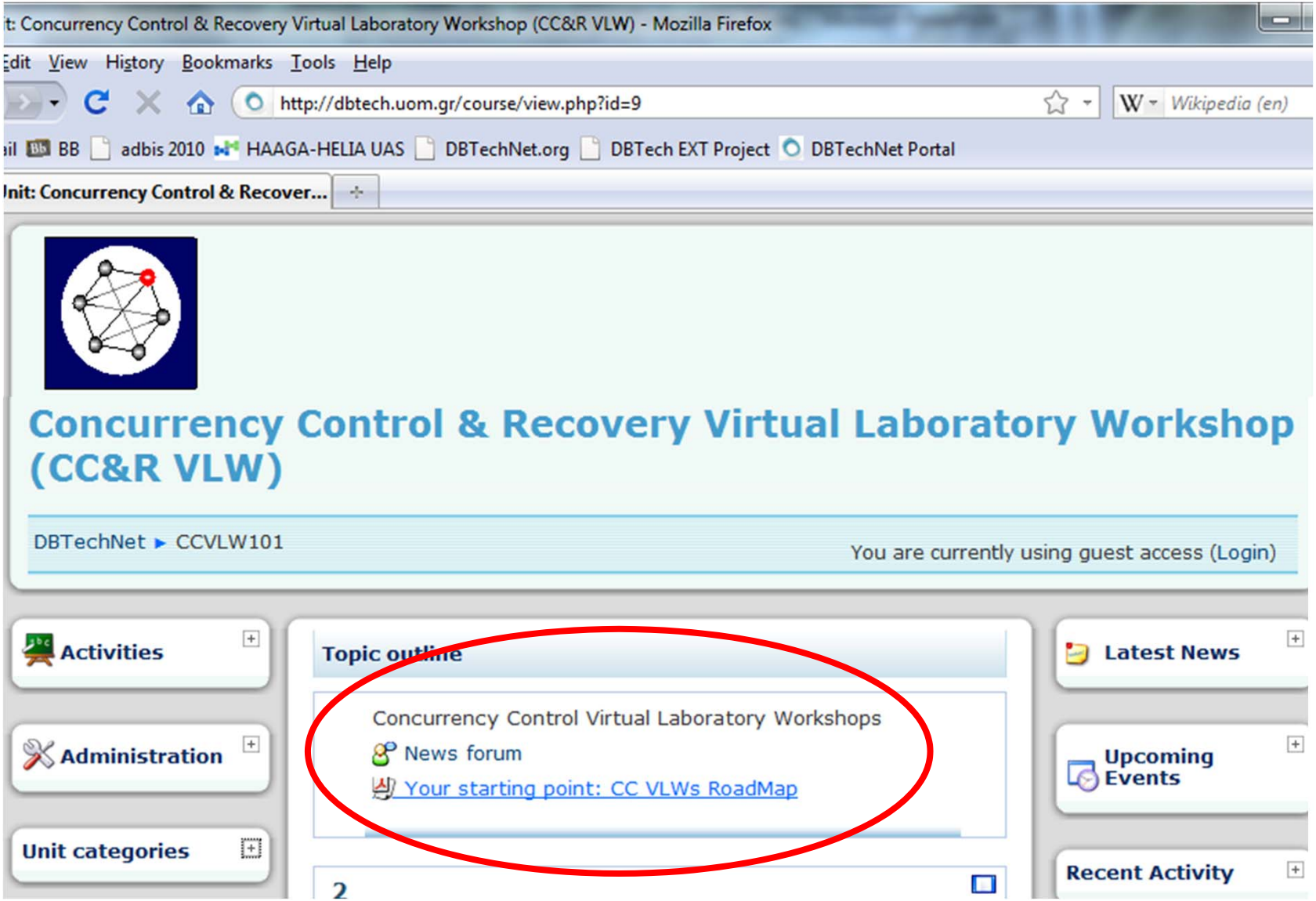
Why ?

- To provide learners with **hands-on experiments** on the basic concurrency control issues (normally learned from textbooks)
- To provide learners with basic **knowledge and skills necessary in solving typical concurrency related problems** when using a mainstream RDBMS product

What ?

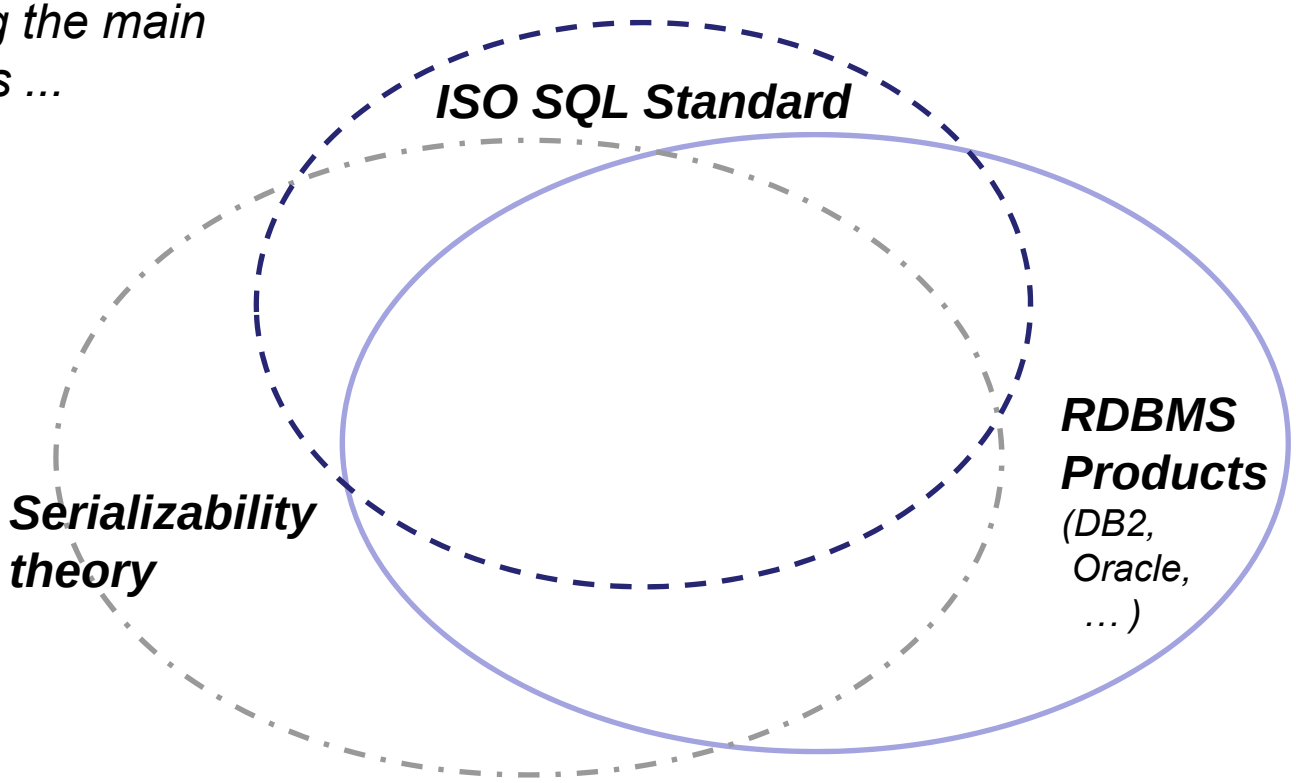
- A set of 4 laboratories (*the first one is already available*) with tutorials, review questions, lab instructions, lab tasks, and virtual laboratory environments
- Focusing on the CC implementations in the following mainstream RDBMS products: DB2, Oracle, and SQL Server

How? / CC VLW on the DBTechNet Portal

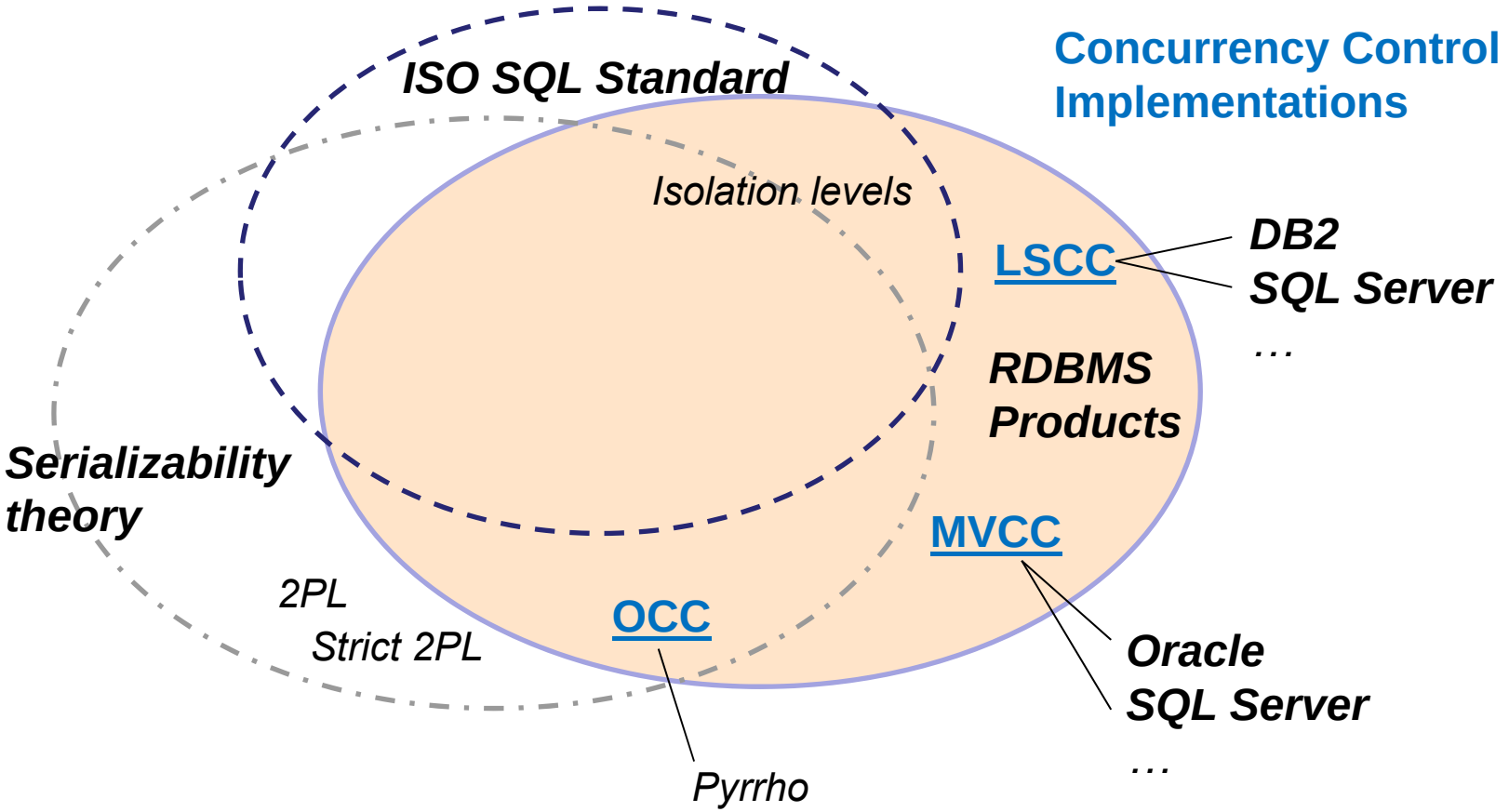


Why? / Transaction Concurrency Coverage

The workshop starts with a tutorial for clarifying the main concepts ...



Concurrency Control of Transactions



Preserving Database Integrity?
Performance?

CC Implementations in RDBMS Products

Typical concurrency control (CC) implementations in RDBMS products:

Locking Scheme *Concurrency Control* (**LSCC**)

- Locks on multi-granular structures, indexes, schemas
- In a competition situation, the winner depends on current locks
- If a deadlock occurs, the deadlock detector chooses the victim

Multi-Versioning *Concurrency Control* (**MVCC**)

- In a competition situation, first writer wins

Optimistic *Concurrency Control* (**OCC**)

- In a competition situation, first one to COMMIT wins

Locking Scheme Concurrency Control (**LSCC**)

- Sample variants of multi-granular lock compatibility matrices

Lock granules:

database

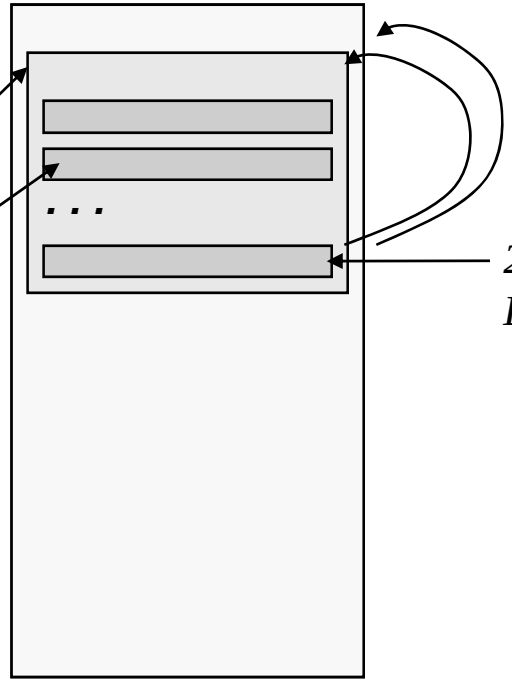
(tablespace)

table

(extent)

page

row



1. Intent locks
IS for S on row
IX for X on row

2.
Lock on row

Lock requested:	Lock already granted to some other process				
	IS	IX	S	SIX	X
IS	grant	grant	grant	grant	wait
IX	grant	grant	wait	wait	wait
S	grant	wait	grant	wait	wait
SIX	grant	wait	wait	wait	wait
X	wait	wait	wait	wait	wait

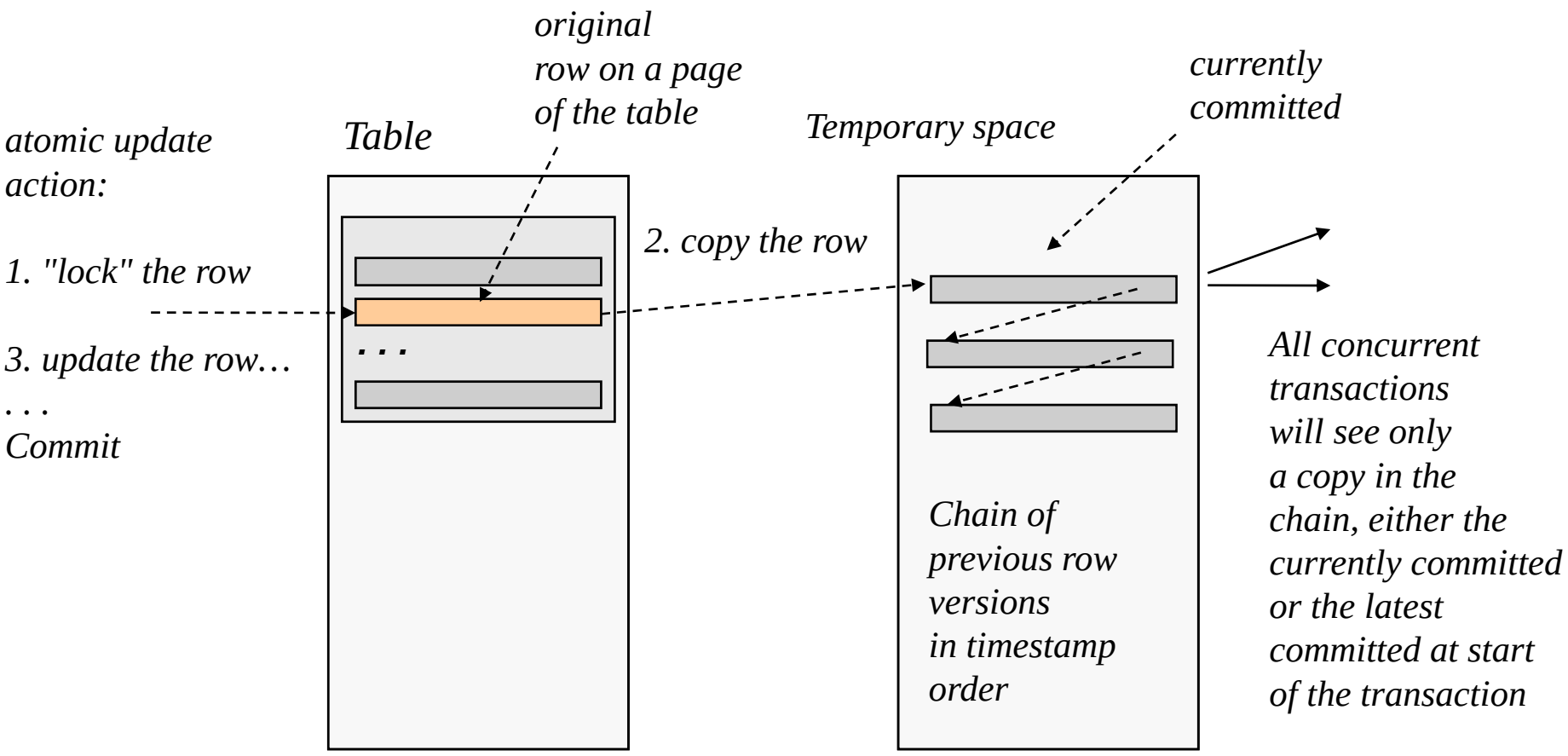
SIX = S + IX

Lock requested:	Lock already granted to some other process			
	none	S	U	X
S	grant	grant	grant[3]	wait
U	grant	grant	wait	wait
X	grant	wait	wait	wait

***S**hared locks (S) allow reading.*
*e**X**clusive locks (X) allow writing and are kept up to end of transaction eliminating lost updates.*

Other locks on index ranges, schemas

Multi-Versioning Concurrency Control (**MVCC**)



Optimistic Concurrency Control (**OCC**)

Transaction phases: *Read*
Validate
Write

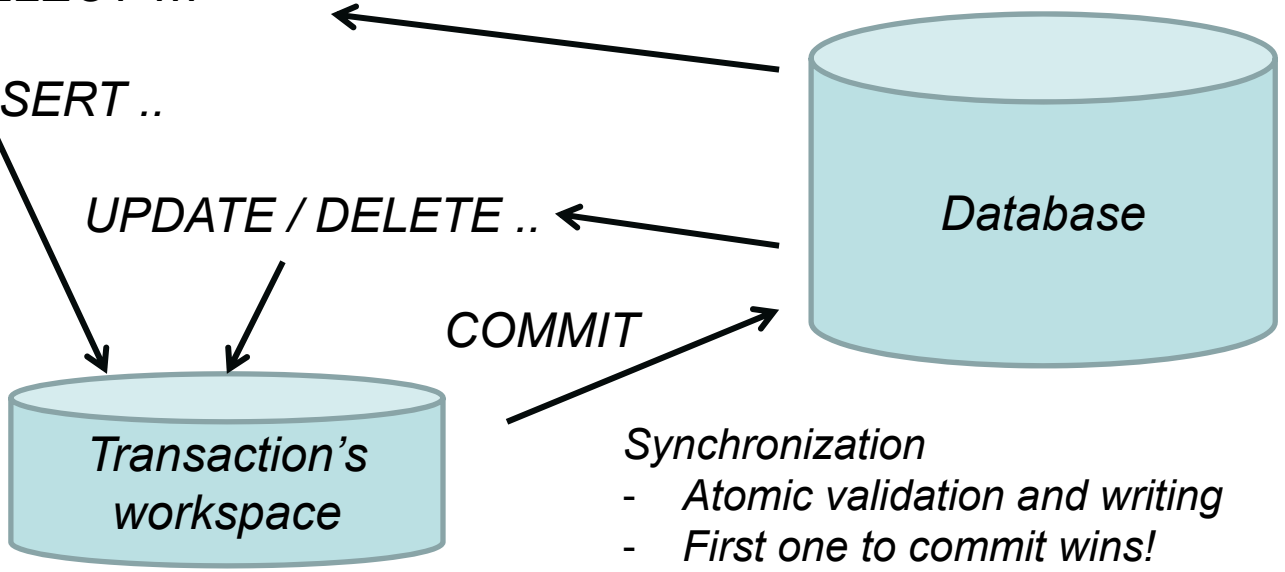
Transaction:

SELECT …

INSERT ..

UPDATE / DELETE ..

COMMIT



Synchronization
- *Atomic validation and writing*
- *First one to commit wins!*

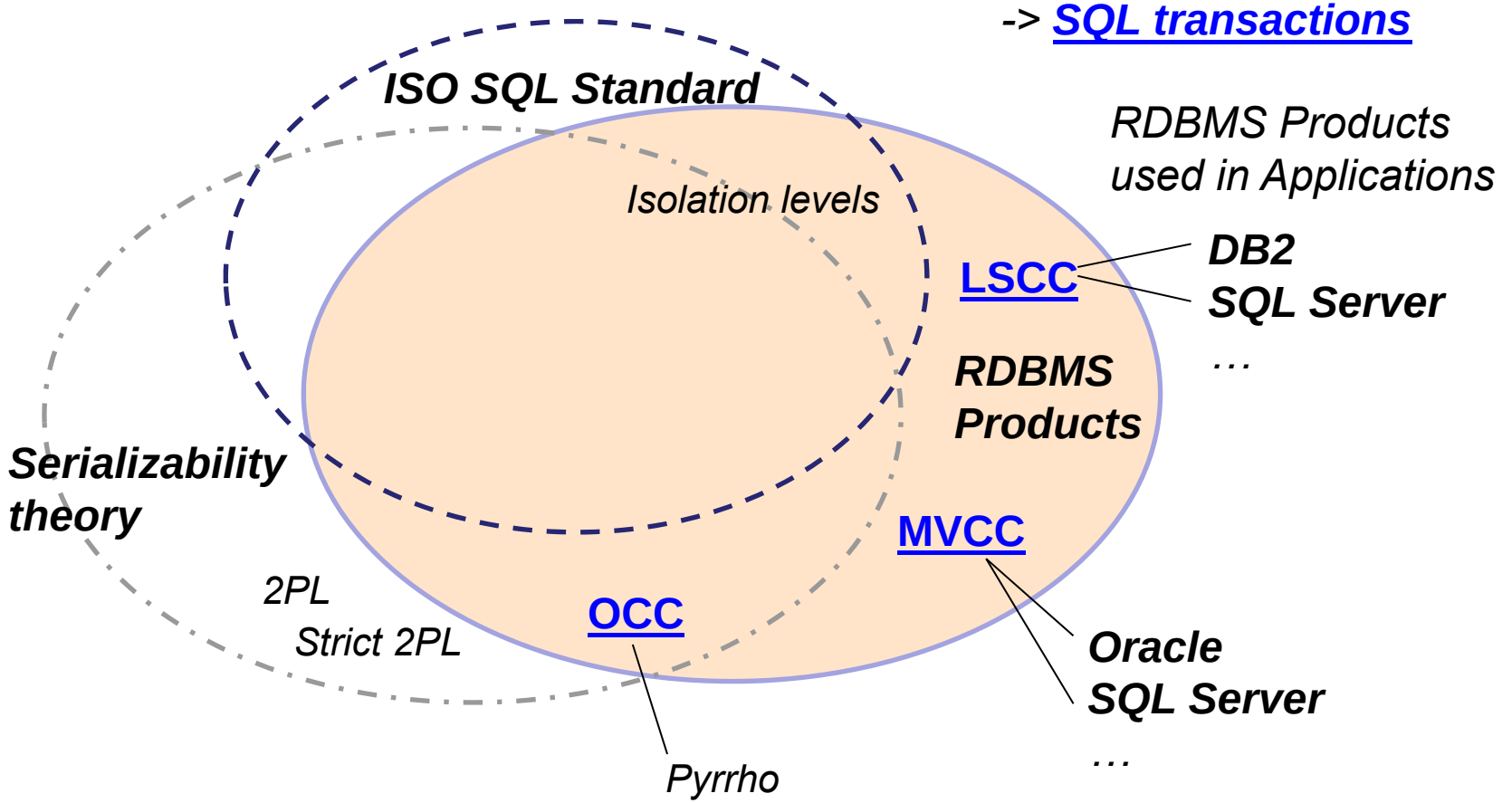
ROLLBACK ?
≈ just forget the workspace!

Server-side OCC provides strict isolation, but it has not been implemented in any commercial mainstream RDBMS product !

Concurrency Control and Context of Transactions

Stand-alone SQL transactions comprise basic knowledge & skills, but …

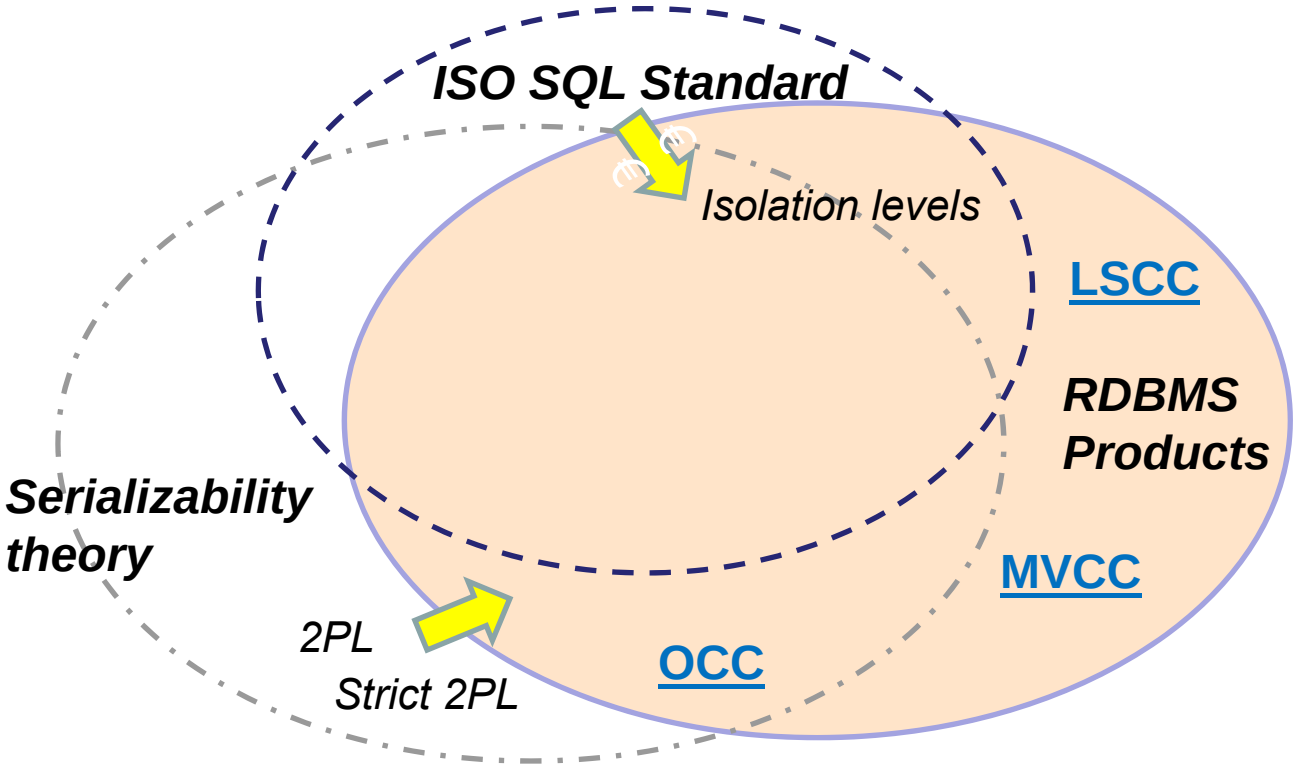
*Real world **Business Transactions** run in Applications*
*-> **User Transactions***
*-> **SQL transactions***



Patterns and Best Practices ?

Software Engineering View:

By studying applications we can find *generic patterns* of user transactions and *best practices* for application development for the industry



What ? / Concurrency Control Labs

DBTech EXT Concurrency Labs (CC Labs 1 - 4)

CC Lab 1: Isolation Levels and Concurrency Technologies

Tutorial on SQL Concurrency Technologies

- Introducing the basic concurrency concepts and problems
- Introducing the concurrency services and problems of the mainstream RDBMS products used by the industry

Hands-on Lab (*also suitable for self-study*)

- Executing SQL scripts (in DB2, Oracle, SQL Server, Pyrrho) to
  - Create the sample database to be used in the lab
  - Study basics of transaction isolation levels and study how the DBMS behaves when different kinds of concurrency conflicts occur
  - Study differences between concurrency control implementations in RDBMS products

Concurrency Control Labs

(Under construction...)

CC Lab 2: **Multiuser Environment with TPC-A Transactions**

Hands-on Lab

- Using DBTechNet software (written in Java) that makes it possible to observe and study
  - Concurrency conflicts, problems, and performance issues
  - Differences between RDBMS products (accessed via JDBC)
  - Transaction isolation levels
  - Client-Side SQL vs. Stored Procedures
  - etc.
- Uses TPC-A Benchmark Database with sample data
- Two different lab implementations
  - **TPCA Lab**: Multiple concurrent learners execute single threads
  - **TPCA 2.1 Lab**: Single learner executes multiple concurrent threads

CC Lab 2: TPCA Lab

TR Client - SQL Server 2005 <server> [jdbc:microsoft:sqlserver://rolle:1433]

File Data Source Help

TEST PARAMETERS FOR TPCA_transaction

Number of transactions

Number of transactions in a test run: 1

Number of test runs : 1

Connections

Open and close a connection for every transaction

Concurrency control

Isolation level

Read Committed Repeatable Read

Locking

Lock the selected rows: Vendor-specific select for upd...

Serialization failure handling

Maximum number of transaction restarts: 0

Debug

Print debug messages Test using a single hot s...

Change Data Source New test RUN

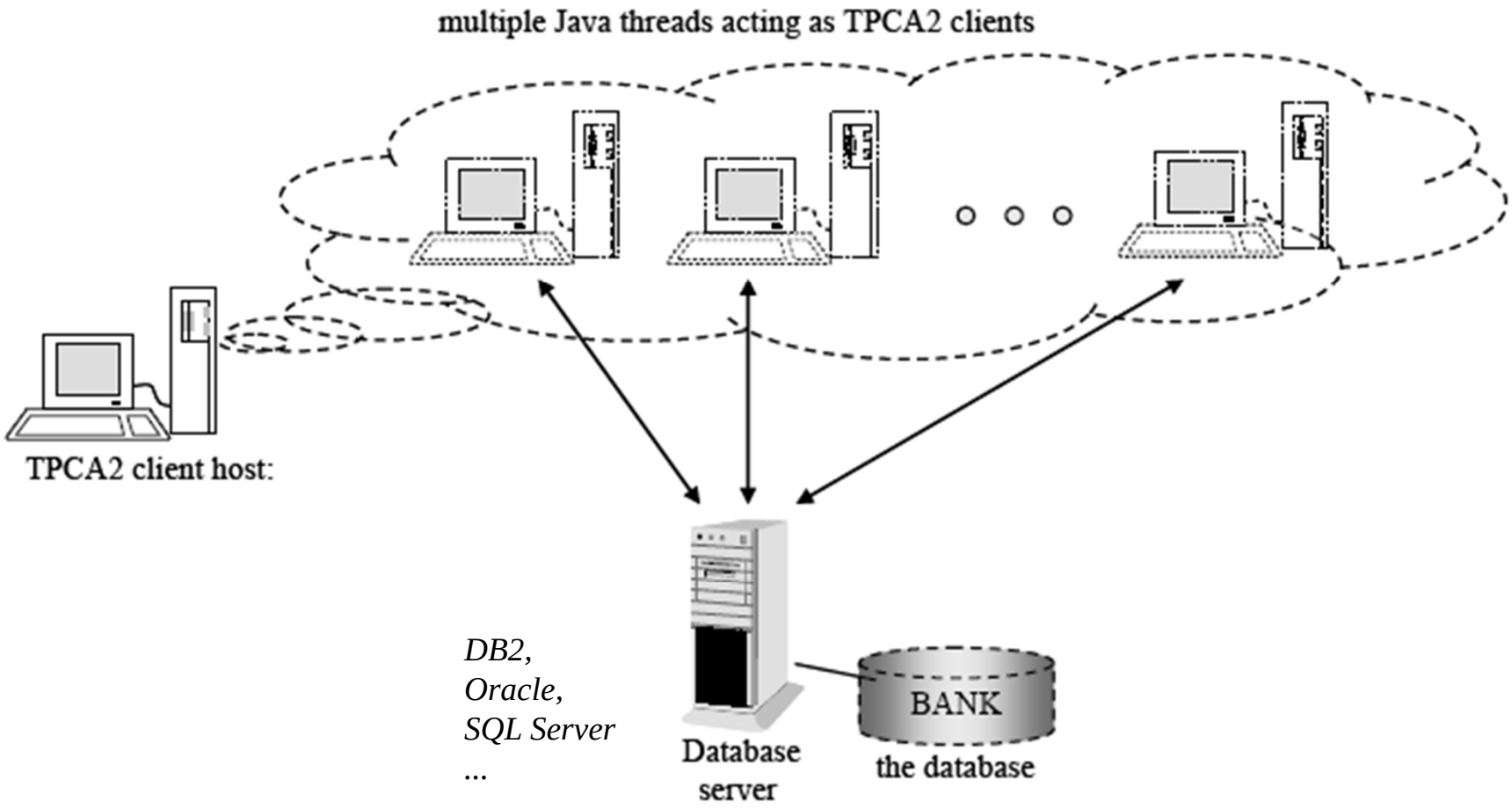
TEST RESULTS

Current test

Transactions	First	Second	Third
Committed:			
Failed:			
Restarted transactions:			
Total of restarts:			
First connect time:			
Execution time (seconds)			
Total:			
Average:			
TPS:			

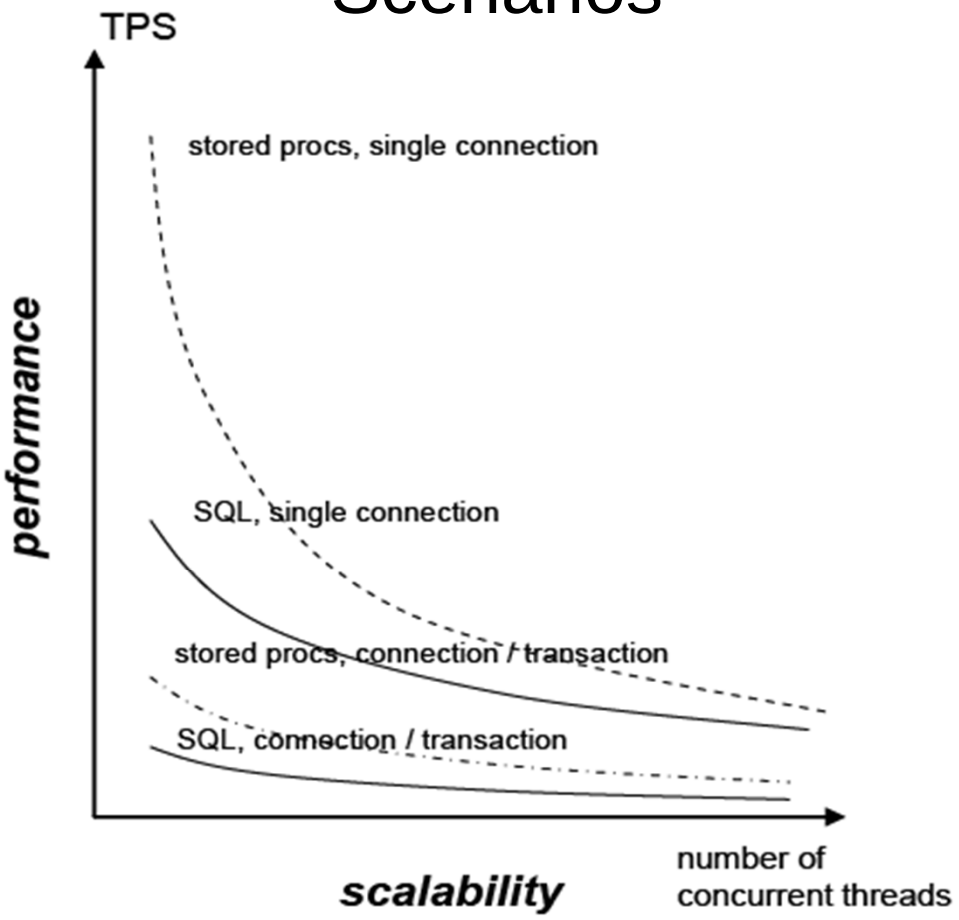
Previous tests

CC Lab 2: TPCA 2.1



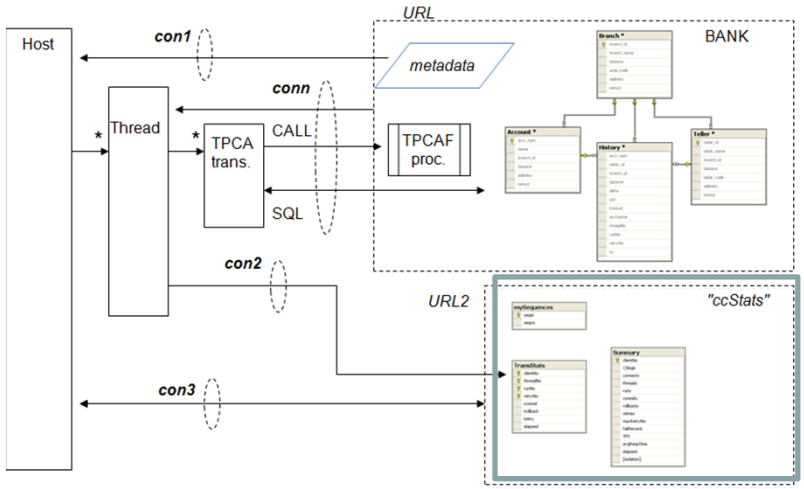
TPCA2 client host using multithreading for client connections

CC Lab 2: Expected Performance/Scalability Scenarios



Hypothetical Performance/Scalability scenarios of some DBMS on depending on number of concurrent TPCA transactions (and threads)

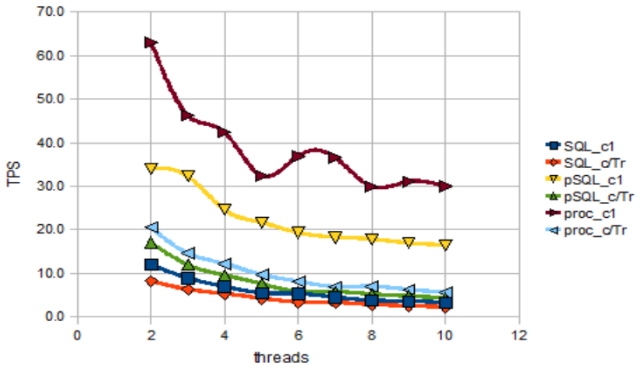
CC Lab 2: Analyzing the Results



Excel or Calc

VBA / SQL

Threads	TPS statistics Conn/thread SQL_c1	Conn/trans SQL_c/Tr	DB2 Express-C 9.7 PreparedStatements Conn/thread pSQL_c1	Conn/trans pSQL_c/Tr	VirtualBox Debian Stored Procs Conn/thread proc_c1	Conn/trans proc_c/Tr
1						
2	12,0	8,2	34,0	17,1	63,0	20,6
3	8,9	6,2	32,3	12,0	46,3	14,6
4	6,9	5,2	24,6	9,5	42,5	12,2
5	5,4	4,2	21,6	7,6	32,3	9,7
6	5,2	3,3	19,3	5,7	36,9	8,1
7	4,4	3,2	18,2	5,7	36,6	6,8
8	3,7	2,7	17,7	5,1	29,9	6,9
9	3,5	2,4	16,9	4,7	31,0	6,2
10	3,2	2,2	16,4	4,2	29,9	5,5



Concurrency Control Labs

(Under construction...)

CC Lab 3: **Data Access Patterns and RVV Discipline**

Tutorial

- Data Access in Modern Application Architectures
- Approaches, services and programming technologies for **row version control** in DB2, Oracle, and SQL Server environment

Lab

- Row Version Verifying (RVV) Data Access Discipline for avoiding Blind Overwriting of Data

CC Lab 4: **Distributed Transactions**

- *To be developed in the future...*

To sum it up...

The free **DBTech Virtual Laboratory Workshop** on CC in Databases is available on the **DBTechNet portal** (http://dbtech.uom.gr)

- A set of 4 laboratories (*the first one is already available*) with tutorials, review questions, lab instructions, lab tasks, and virtual laboratory environments
- Focusing on the CC implementations in the mainstream RDBMS products: DB2, Oracle, and SQL Server

Objectives

- To provide learners with **hands-on experiments** on the basic concurrency control issues (normally learned from textbooks)
- To provide learners with basic **knowledge and skills necessary in solving typical concurrency related problems** when using a mainstream RDBMS product

www.DBTechNet.org

Lifelong Learning Programme

Thank you for your attention!