



IBM Software Group

# DB2 Express-C 9 overview course

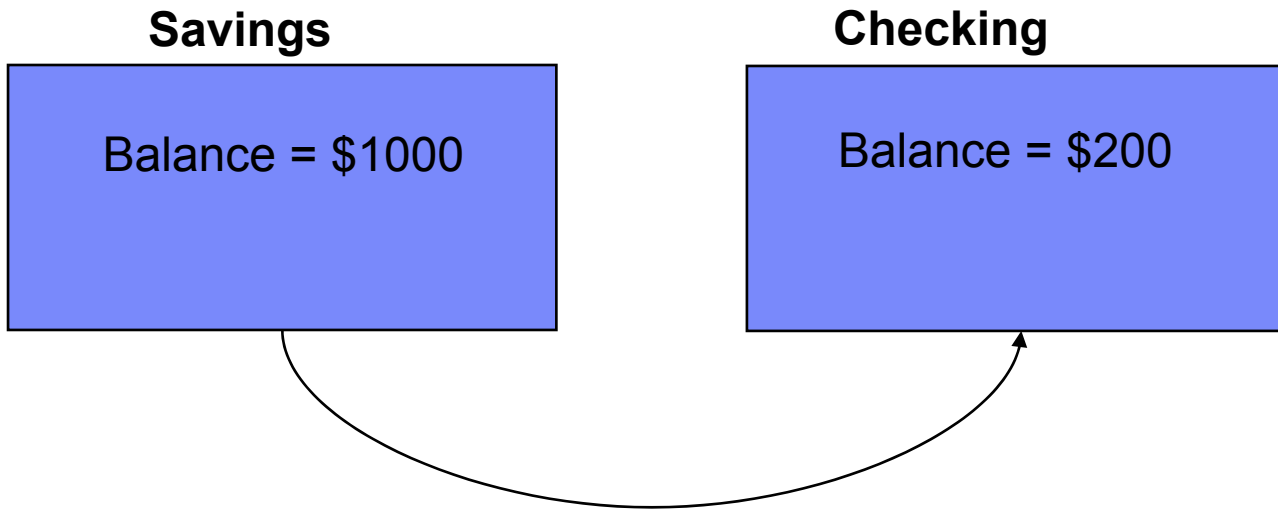
## *Concurrency and Locking*

**DB2** Information Management Software

A horizontal decorative bar spanning the width of the slide, composed of various colored squares and rectangles in shades of green, yellow, red, and blue.

**ON DEMAND BUSINESS™**

# What is a Transaction?

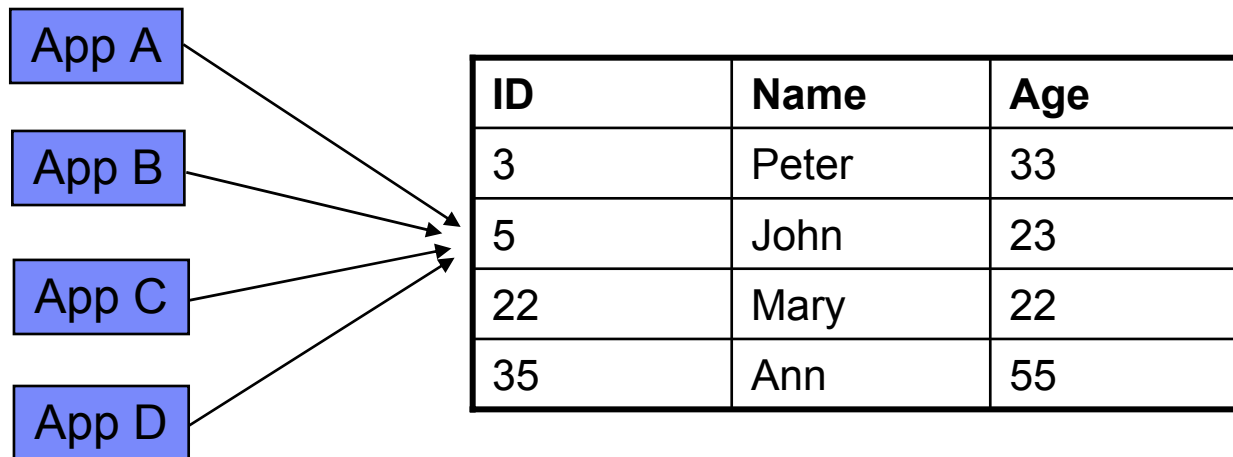


Transfer \$100 from Savings to Checking:

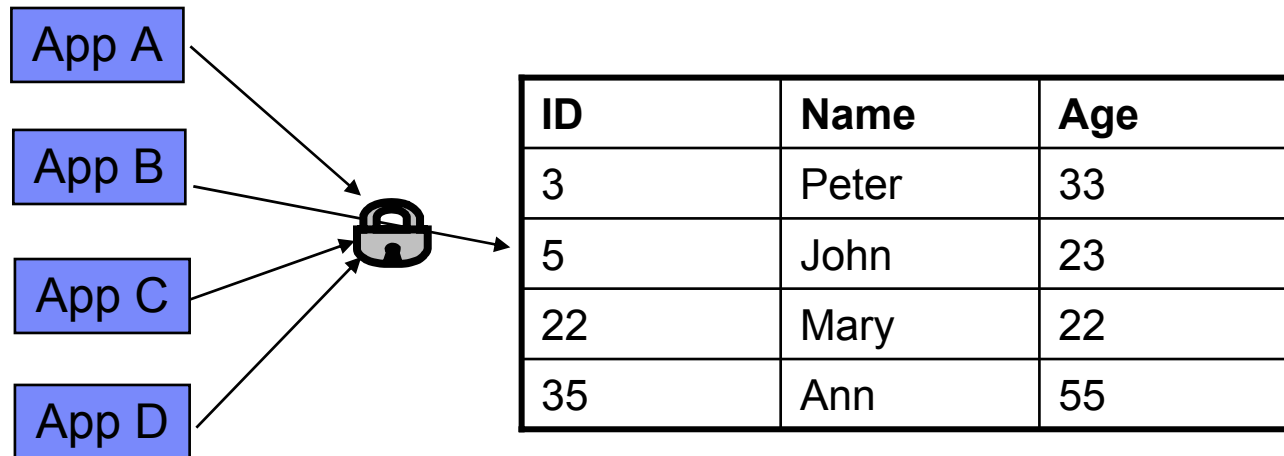
- Debit \$100 from Savings account
- Credit \$100 to Checking account



# Concurrency



# Locking

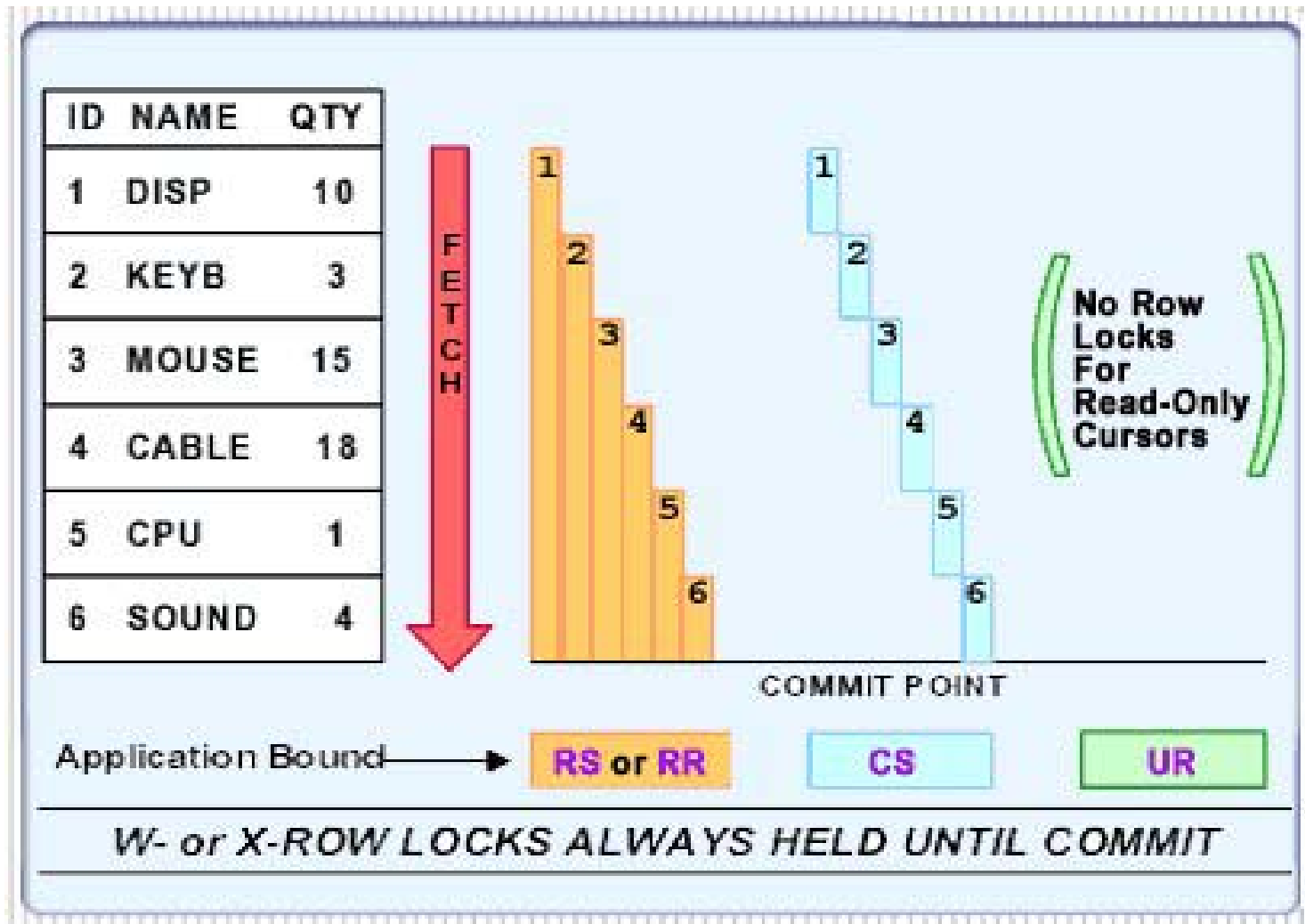


# Isolation Levels

- A policy about how/when to set locks
- DB2 provides different levels of protection to isolate data
  - ▶ Uncommitted Read (UR)
  - ▶ Cursor Stability (CS)
  - ▶ Read Stability (RS)
  - ▶ Repeatable Read (RR)



# Comparing isolation levels



# Comparison of Isolation Level Terminolgy

DB2	.NET	JDBC
Uncommitted Read (UR)	ReadUncommitted	TRANSACTION_READ_UNCOMMITTED
Cursor Stability (CS)	ReadCommitted	TRANSACTION_READ_COMMITTED
Read Stability (RS)	RepeatableRead	TRANSACTION_REPEATABLE_READ
Repeatable Read (RR)	Serializable	TRANSACTION_SERIALIZABLE



# Statement Level Isolation

SELECT ... WITH {UR | CS | RS | RR}

- Example Scenario:
  - ▶ Application needs to get a "rough" count of how many rows are in table. Performance is of utmost importance.

SELECT COUNT(\*) FROM tab1 WITH UR



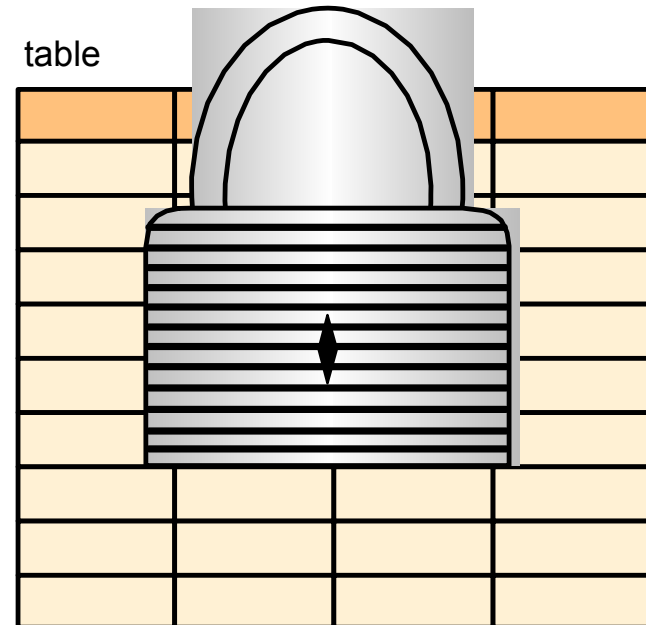
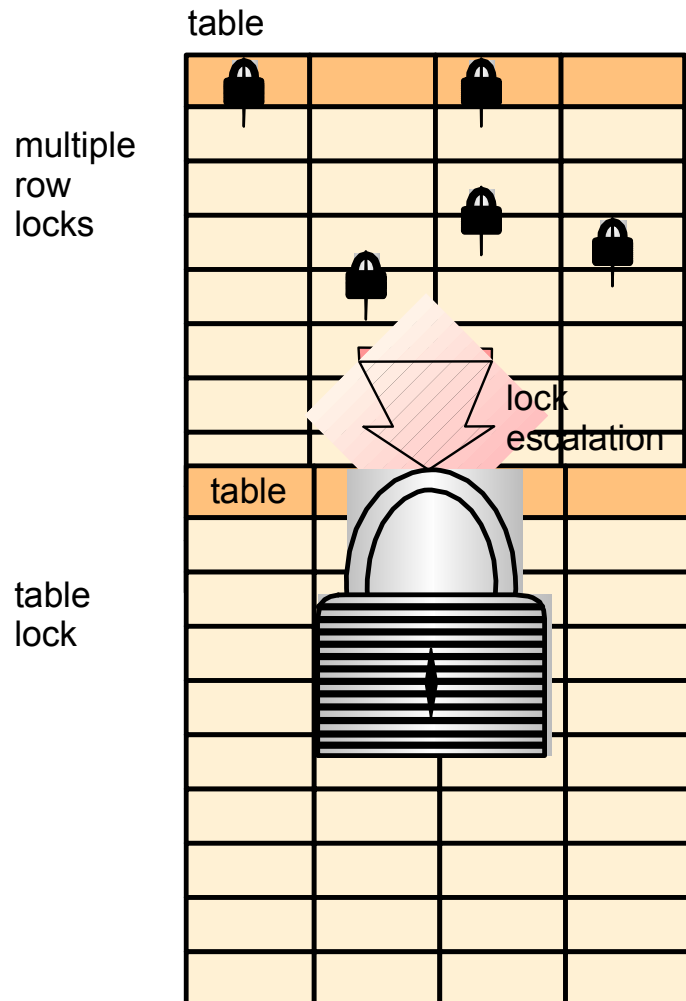


# Lock Wait

- By default, an application waits indefinitely to obtain any needed locks
- **LOCKTIMEOUT** (db cfg):
  - ▶ Default value is -1 or *infinite* wait
  - ▶ Change to specify the number of seconds to wait for a lock
- A database connection also has a user-definable **CURRENT LOCK TIMEOUT** register
  - ▶ Use the **SET LOCK TIMEOUT** statement to change its value
  - ▶ Once it is set for a connection, it persists across transactions
  - ▶ e.g. **SET LOCK TIMEOUT=WAIT *n***



# Lock Escalation



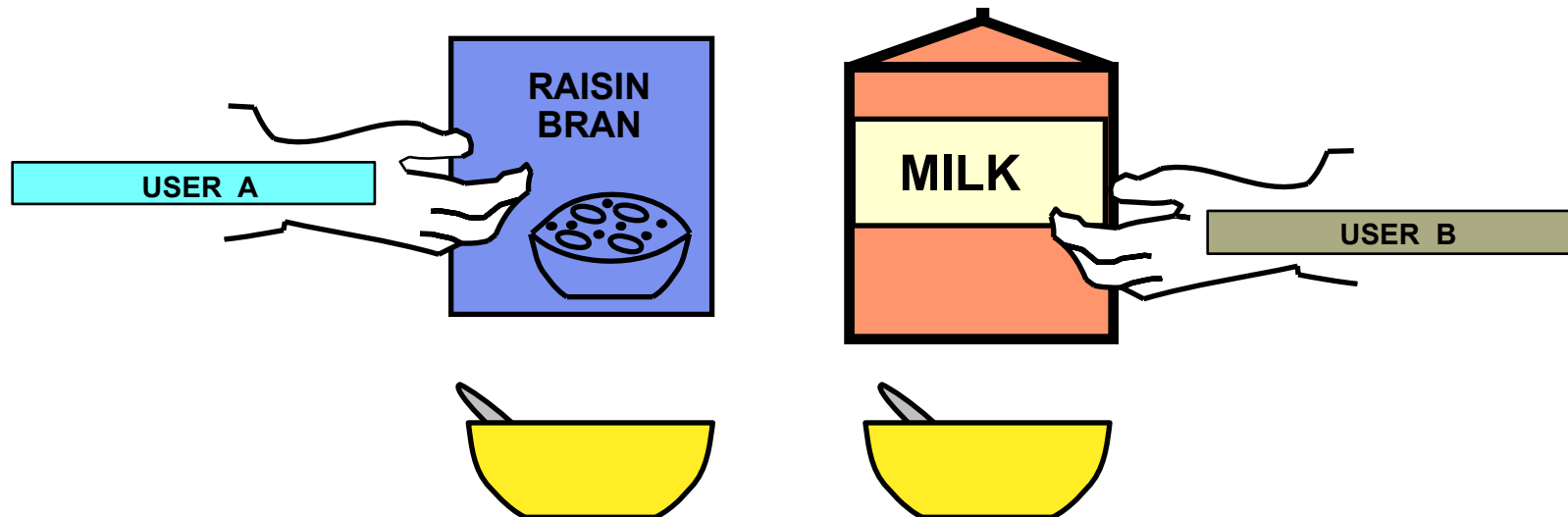
# Lock Escalation

- When optimizer thinks it is better to have one lock on the entire table, rather than multiple row locks
- Database configuration parameters that affect lock escalation:
  - ▶ **LOCKLIST** – the amount of memory (4k pages) to manage locks for all connected applications
    - Default is 50 \* 4K pages on Windows
  - ▶ **MAXLOCKS** –Max percentage of the entire lock list that a single application can use up
    - Default is 22 percent



# Deadlock Causes and Detection

- A deadlock occurs when two or more applications connected to the same database wait indefinitely for a resource
- The waiting is never resolved because each application is holding a resource that the other needs



within a UNIT OF WORK (UOW)

INSERT CEREAL AND MILK into BOWL



# Deadlock Settings

- Deadlocks are an application design issue most of the time
- **DLCHKTIME** (db cfg) sets the time interval for checking for deadlocks
  - ▶ It defines the frequency that the database manager checks for deadlocks among all the applications connected to a database
- If you are experiencing many deadlocks, you should re-examine your existing transactions and see if any re-structuring is possible

