DB2 Data Management Software

IBM

e business software

Chapter 8: Data Concurrency

IBM DB2 Universal Database V8.1
Database Administration Certification Preparation Course

Maintained by Clara Liu

IBM Software Group

Objectives

- In this section, we will cover:
  - Different Isolation Levels in DB2
  - DB2 Locking Mechanism

DB2 Data Management Software

IBM

e business software

Chapter 8: Data Concurrency

Isolation Levels

Locking Mechanisms

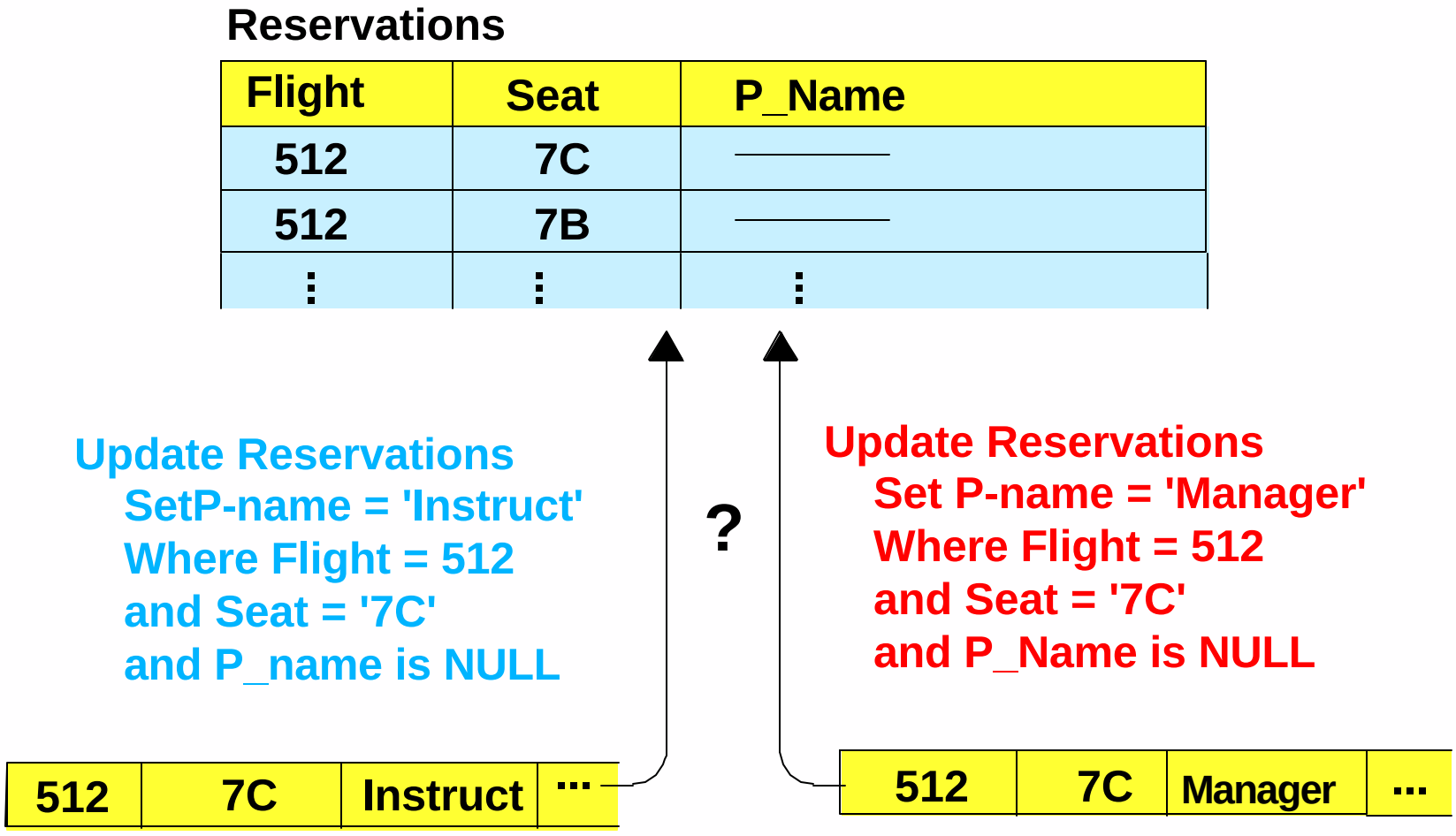
IBM Software Group

Purpose of Locking

- Some undesirable effects may encounter when many users access the same data source:
  - Lost Update
  - Uncommitted Read
  - Non-repeatable Read
  - Phantom Read
- To guarantee the integrity of the data, some sort of modification rules are required to control the use of data

Lost Updates

- Same data is retrieved and updated by two users concurrently
- Last successful change kept, first change overridden



Uncommitted Data

- Can read or view data changed or added that has not committed yet

Reservations

Flight	Seat	P_Name
512	7C	————
512	7B	————
⋮	⋮	

1. **Update Reservations**
   Set P-name = 'Instruct'
   Where Flight = 512
   and Seat = '7C'
   and P_Name is NULL

2. **Select seat**
   From Reservations
   Where P-name is NULL

512	7C	Instruct

3. **Roll back**

4. **Incorrect results set**

Non Repeatable Reads

- Same SELECT statement returns different result set within the same transaction

FLIGHT	SEAT	NAME	DESTINATION	ORIGIN
512	7B	_____	DENVER	DALLAS
⋮				
⋮				
814	8A	_____	SAN JOSE	DENVER
⋮				
134	1C	_____	HONOLULU	SAN JOSE
⋮				⋮

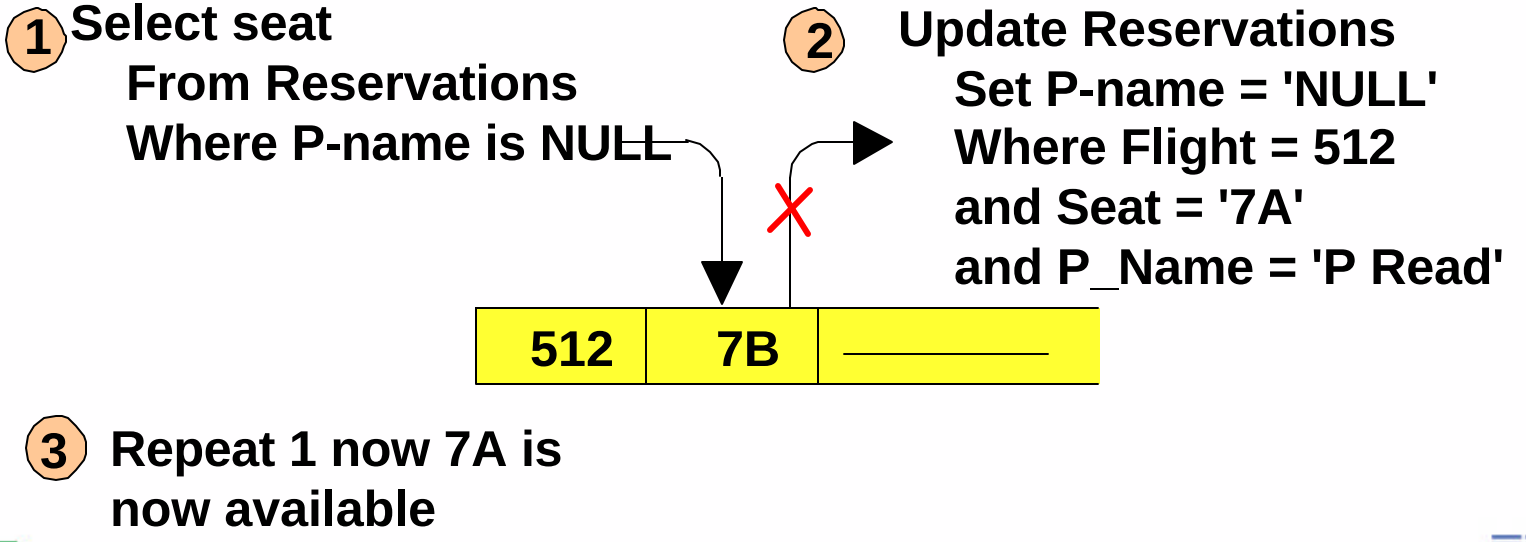
Book a flight from Dallas to Honolulu

Phatom Reads

- Application executes same SQL twice, second result set contains additional rows

Reservations

Flight	Seat	P_Name
512	7B	————
512	7A	P Read
⋮	⋮	



Isolation Levels

- DB2 provides different levels of protection to isolate data
  - Uncommitted Read
  - Cursor Stability
  - Read Stability
  - Repeatable Read
- Cursor Stability is the default isolation level
- Isolation level can be specified for a session, a client connection, or an application before a database connection
- For embedded SQL, the level is set at bind time
- For dynamic SQL, the level is set at run time

Statement Isolation Levels

- New feature allows isolation level to be specified at the statement level
- New WITH [RR RS CS UR] clause
- The following SQL statements support statement-level isolation:

SELECT-statement	searched DELETE
SELECT INTO	searched UPDATE
INSERT	DECLARE CURSOR statement
FOR statement	

- Restrictions
  - cannot be used on subqueries
  - WITH UR applies to only read-only operations
    - if it is used otherwise, the code will automatically upgrade the statement from UR to CS
  - The statement-level isolation level overrides the isolation level specified at the package in which it appears.
  - The default isolation level of the statement is the isolation level of the package in which the statement is bound.

Isolation Levels - Uncommitted Read

- Also known as DIRTY READ
- Lowest level of isolation
- Least restrictive or most concurrency
- May return data that is never committed to the database
- Non-repeatable reads and phantom reads may occur

Isolation Levels - Cursor Stability

- Default isolation level
- Locks any row on which the cursor is positioned during a unit of work
- Lock on the row is held until the new row is fetched or the unit of work is terminated
- If a row is updated, the lock is held until the unit of work is terminated
- Cannot return uncommitted data
- Non-repeatable read and phantom reads may occur

Isolation Levels - Read Stability

- Locks only the rows an application retrieves within a unit of work
- Cannot read uncommitted data
- No other application can change the rows that are locked by this application
- May get 'phantom' rows if application issues the same query more than once within the unit of work

Isolation Levels - Repeatable Read

- Highest isolation level, least concurrency
- Locks held on all rows processed to build the result
- Optimizer may choose to get a TABLE lock
- Same query issued by the application more than once in a unit of work will give the same result each time
- No other application can update, delete, or insert a row that would affect the result table until the unit of work completes

DB2 Data Management Software

IBM

e business software

Chapter 8: Data Concurrency

Isolation Levels

Locking Mechanisms

IBM Software Group

Locking

- Locking controlled by isolation level specified
- Database, table spaces, and tables can be explicitly locked
  - Database lock
    - Example: CONNECT TO dbname IN EXCLUSIVE MODE
  - Table space lock
    - Example: QUIESCE TABLESPACES FOR TABLE tabname INTENT FOR UPDATE
  - Table lock
    - Example: LOCK TABLE tabname IN EXCLUSIVE MODE
- Database, tables, and rows can be implicitly locked
  - Database lock
    - During full database restore
  - Table lock
    - Lock escalation
  - Row lock
    - Through normal data modification

Row Level and Table Level Locking

- By default, DB2 uses row level locking
- Can change to table level locking with
  - ALTER TABLE ... LOCKSIZE TABLE
- Affects all applications and users that access the table
- Use table level locking until row level locking is engaged again
  - ALTER TABLE ... LOCKSIZE ROW
- Changes minimum point-in-time recovery for table's table space

Locking Type Compatibility

Row locking:

S - Share

U - Update

X - Exclusive

Table locking:

IN - Intent None

IS - Intent Share

IX - Intent Excl.

U - Update

X - Exclusive

State being	State of Held Resource												
Requested	none	IN	IS	NS	S	IX	SIX	U	NX	X	Z	NW	W
none	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
IN	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	yes	yes
IS	yes	yes	yes	yes	yes	yes	yes	yes	no	no	no	no	no
NS	yes	yes	yes	yes	yes	no	no	yes	yes	no	no	yes	no
S	yes	yes	yes	yes	yes	no	no	yes	no	no	no	no	no
IX	yes	yes	yes	no	no	yes	no	no	no	no	no	no	no
SIX	yes	yes	yes	no	no	no	no	no	no	no	no	no	no
U	yes	yes	yes	yes	yes	no	no	no	no	no	no	no	no
NX	yes	yes	no	yes	no	no	no	no	no	no	no	no	no
X	yes	yes	no	no	no	no	no	no	no	no	no	no	no
Z	yes	no	no	no	no	no	no	no	no	no	no	no	no
NW	yes	yes	no	yes	no	no	no	no	no	no	no	no	yes
W	yes	yes	no	no	no	no	no	no	no	no	no	yes	no

Locking Behaviour

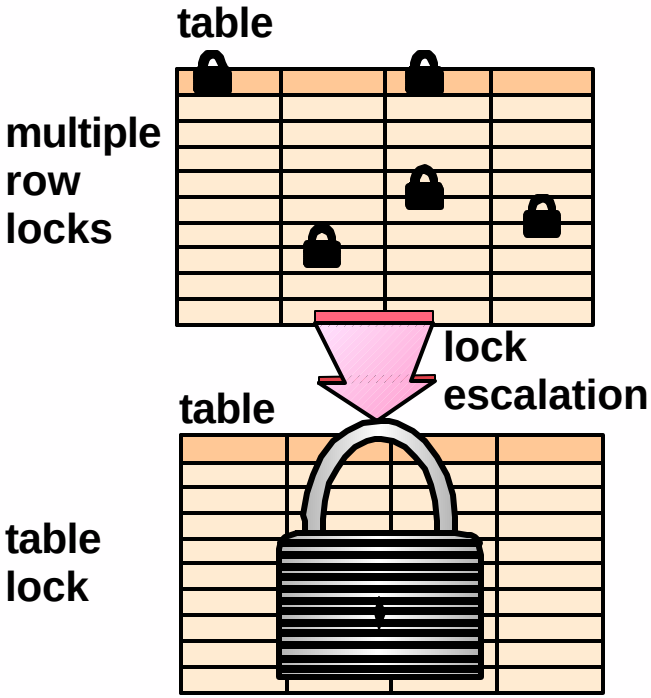
- Lock conversion
  - When an application already locked a data object and requires a more restrictive lock

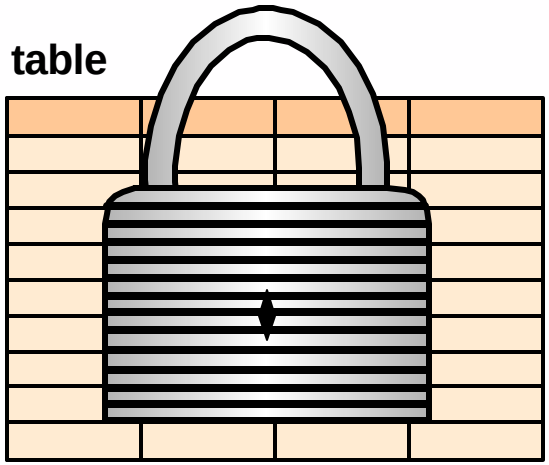
```
SELECT * FROM db2cert.test_taken
WHERE seat_no = '1' AND date_taken = CURRENT DATE
FOR UPDATE OF seat_no;

UPDATE db2cert.test_taken
SET seat_no = '2'
WHERE seat_no = '1' AND date_taken = CURRENT DATE
```

Locking Behaviour

- Lock escalation
  - If application changes many rows in one table, better to have one lock on the entire table
  - Each DB2 lock consumes same amount of memory
  - Avoid resource problem
  - Decrease concurrency
  - Parameters affect lock escalation: MAXLOCKS, LOCKLIST





Lock Waits and DeadLocks

- If the application "hogging" the locks, doesn't COMMIT or ROLLBACK, other applications wait until lock is available or timeout exceeded
- Set LOCKTIMEOUT parameter to avoid this
- LOCKTIMEOUT default is -1 or infinite wait.
- Locktimeout error is SQL0911N with subcode "68" sqlstate: 40001
- DeadLock error is SQL0911N with subcode "2" sqlstate: 40001