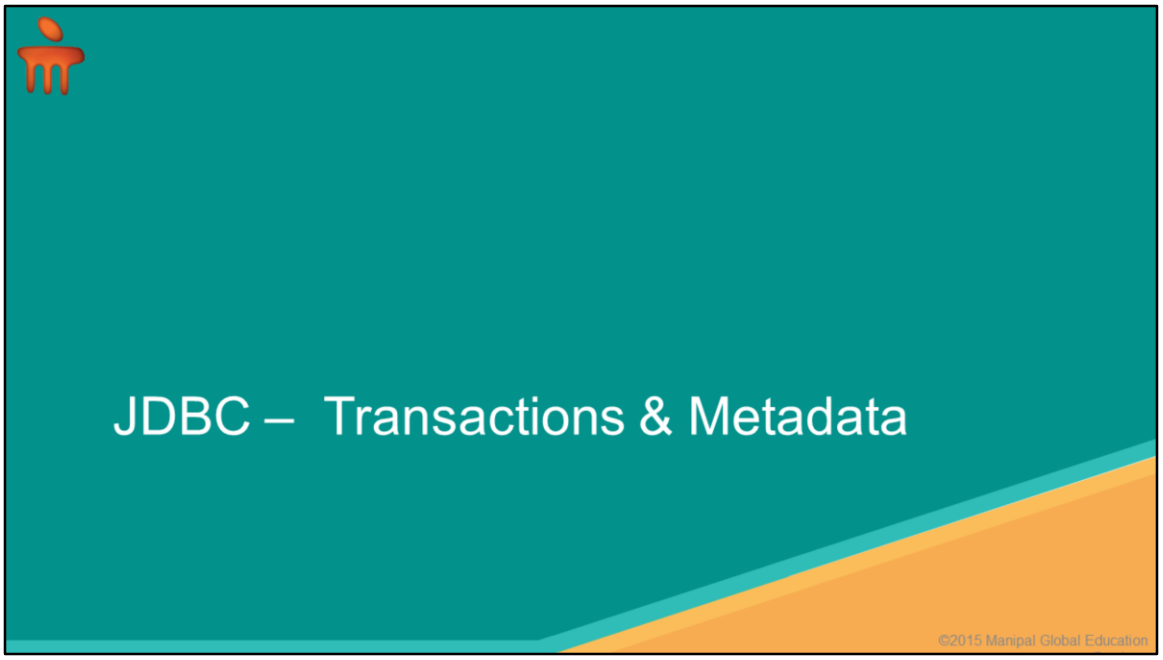

JDBC – Transactions & Metadata
©2015 Manipal Global Education


manipalglobal
education services
INTRODUCTION
JDBC –
Transactions &
Metadata
2
©2015 Manipal Global Education

manipalglobal
education services

LEARNING OBJECTIVES

At the end of this lesson, you will be able to:

- Understand Transaction
- Implement Transaction in JDBC
- Understand Transaction Isolation and its levels
- Get metadata of Database and ResultSet

Refer package **com.mgait.jdbc** in the provided code base for demo programs on the topics covered in this presentation

The demo programs use the 'hr' schema of Oracle Express Edition

CONCEPT

Transactions

manipalglobal
education services

Understanding Transaction

- Transaction represents **a logical unit of work** that contains one or more SQL statements
- A transaction is an atomic unit.
- The effects of all the SQL statements in a transaction can be either all **committed** (applied to the database) or all **rolled back** (undone from the database).

- Example
  - Person-A wants to transfer Rs100 to Person-B. This requires two queries
    - Update query to debit Person-A's account by Rs 100
    - Update query to credit Rs 100 to Person-B's account
  - Both queries should either succeed or fail
  - If first query succeeds and second query fails, data updated will incorrect

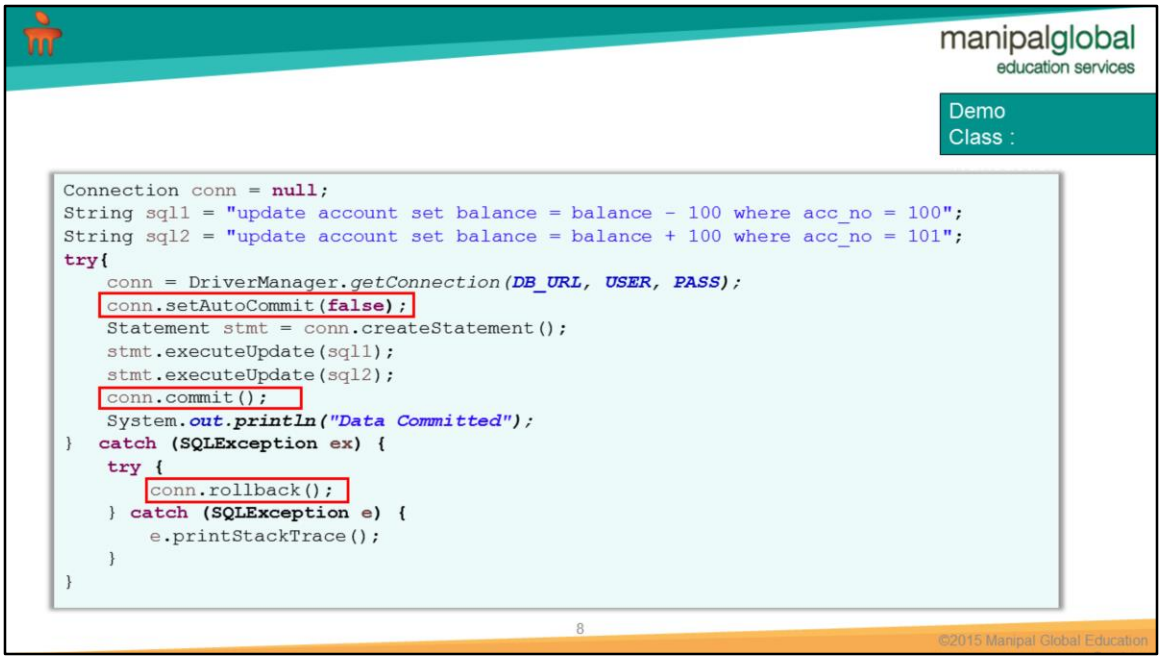
manipalglobal
education services

Transaction Management in JDBC

- By default, a jdbc Connection is in auto-commit mode. It means
  - Each SQL statement is treated as a transaction and is automatically committed after its execution
- Connection interface provides following methods to manage transaction

```
setAutoCommit(boolean) - Sets connection's auto-commit mode
commit()               - commits the transaction
rollback()             - cancels the transaction
```

- In JDBC, two or more statements are grouped into a transaction as below
  - Disable auto-commit mode using setAutoCommit(false)
  - Commit on successful execution of all statements
  - Rollback on failure of any one of the statement in the transaction

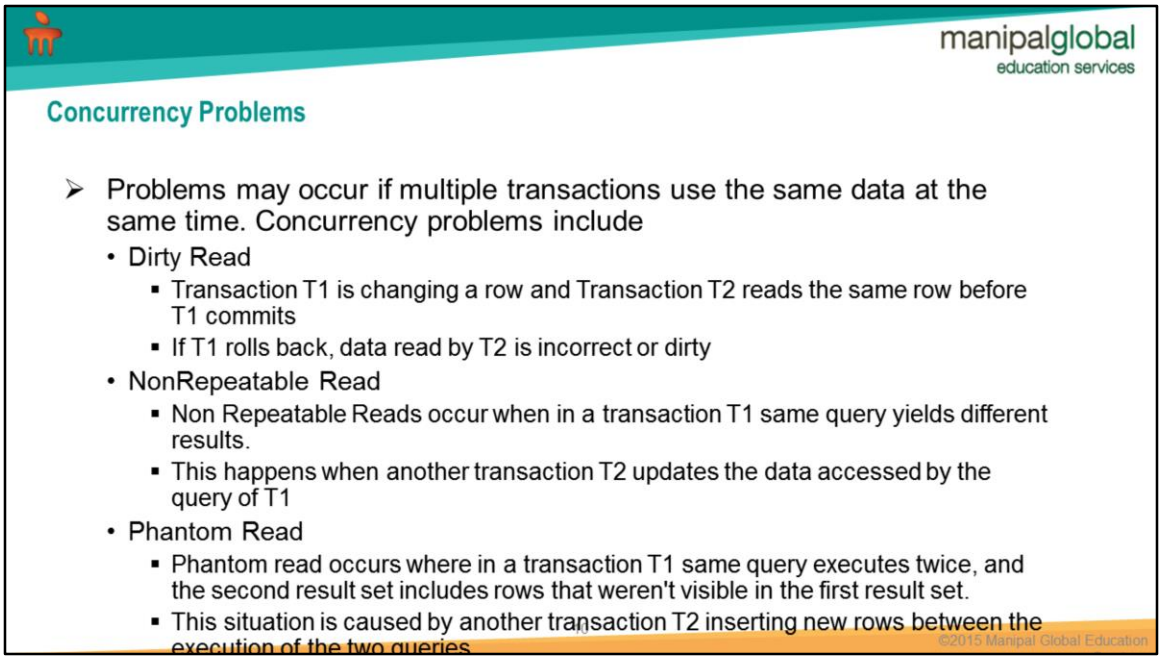

manipalglobal
education services
Demo
Class :
Connection conn = null;
String sql1 = "update account set balance = balance - 100 where acc_no = 100";
String sql2 = "update account set balance = balance + 100 where acc_no = 101";
try{
conn = DriverManager.getConnection(DB_URL, USER, PASS);
conn.setAutoCommit(false);
Statement stmt = conn.createStatement();
stmt.executeUpdate(sql1);
stmt.executeUpdate(sql2);
conn.commit();
System.out.println("Data Committed");
} catch (SQLException ex) {
try {
conn.rollback();
} catch (SQLException e) {
e.printStackTrace();
}
}
8
©2015 Manipal Global Education

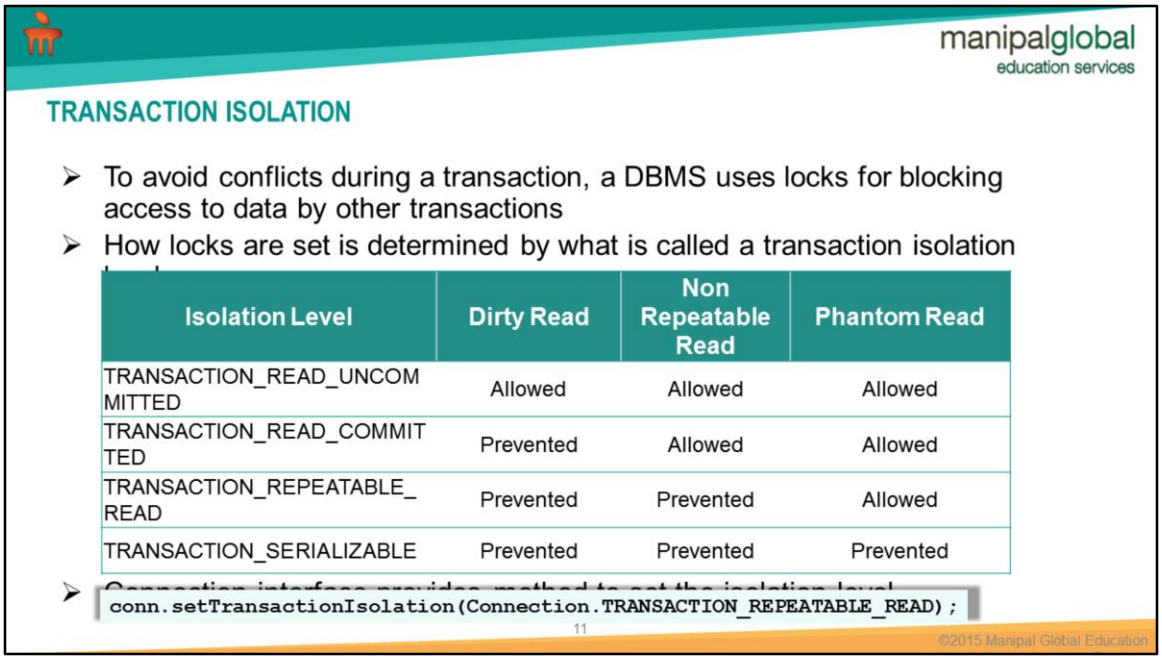
manipalglobal
education services

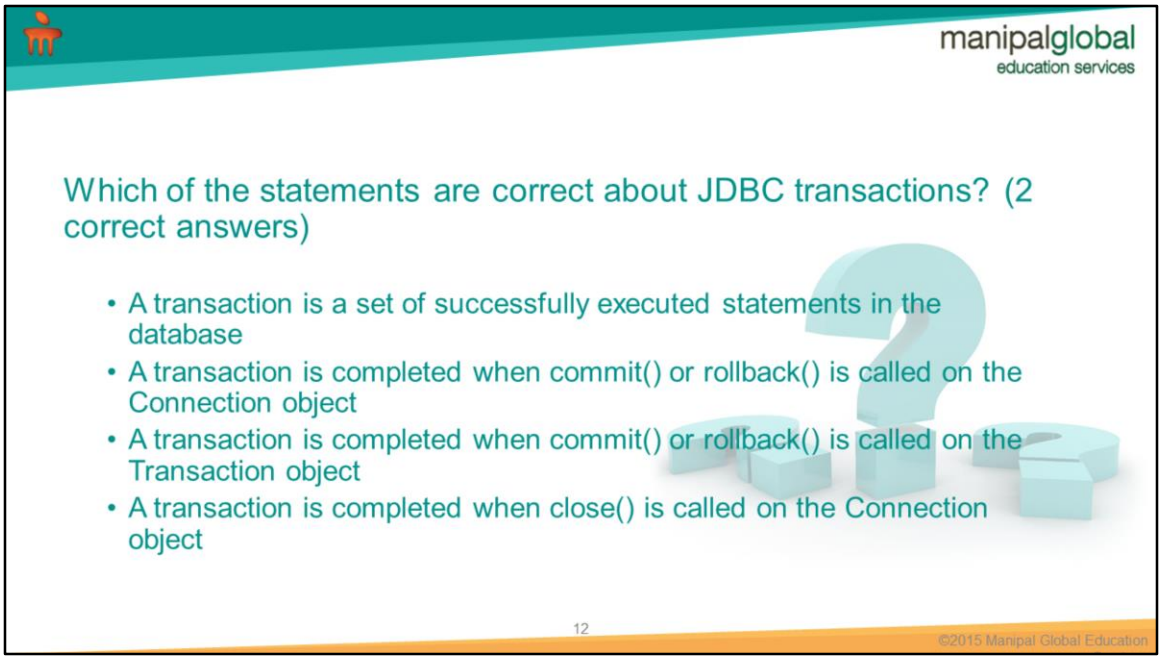
Savepoints in a Transaction

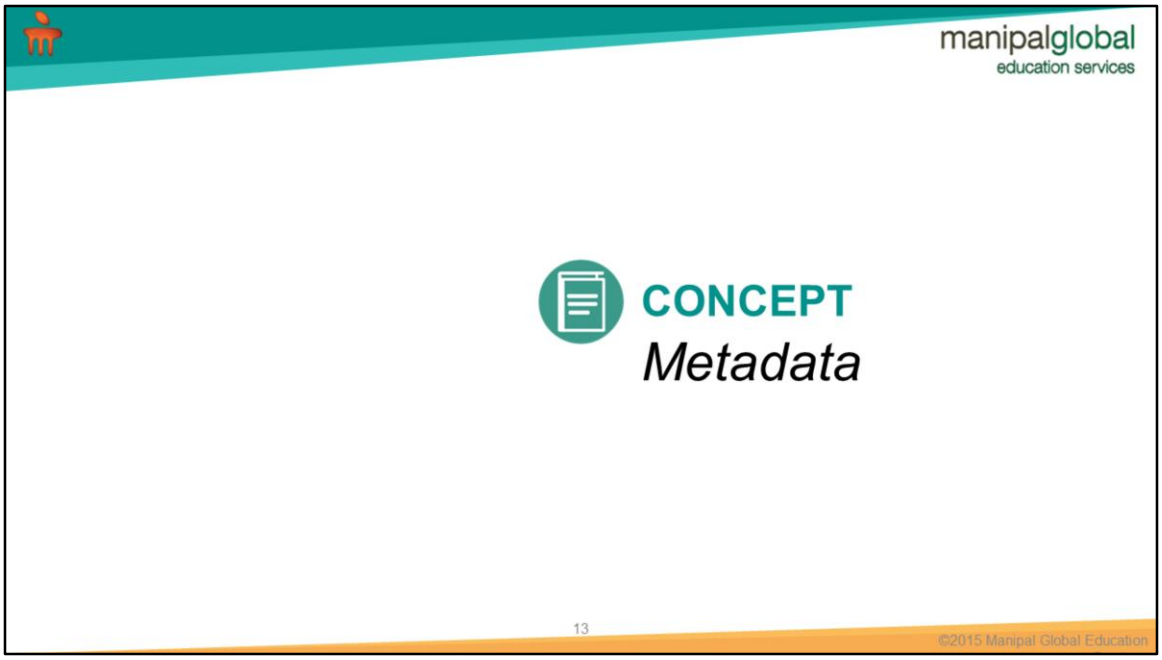
- JDBC offers finer control over transaction commit and rollback through SavePoint
- A named savepoint may be inserted between statements to act as a marker
- The transaction may be rolled back to that marker, leaving all of the

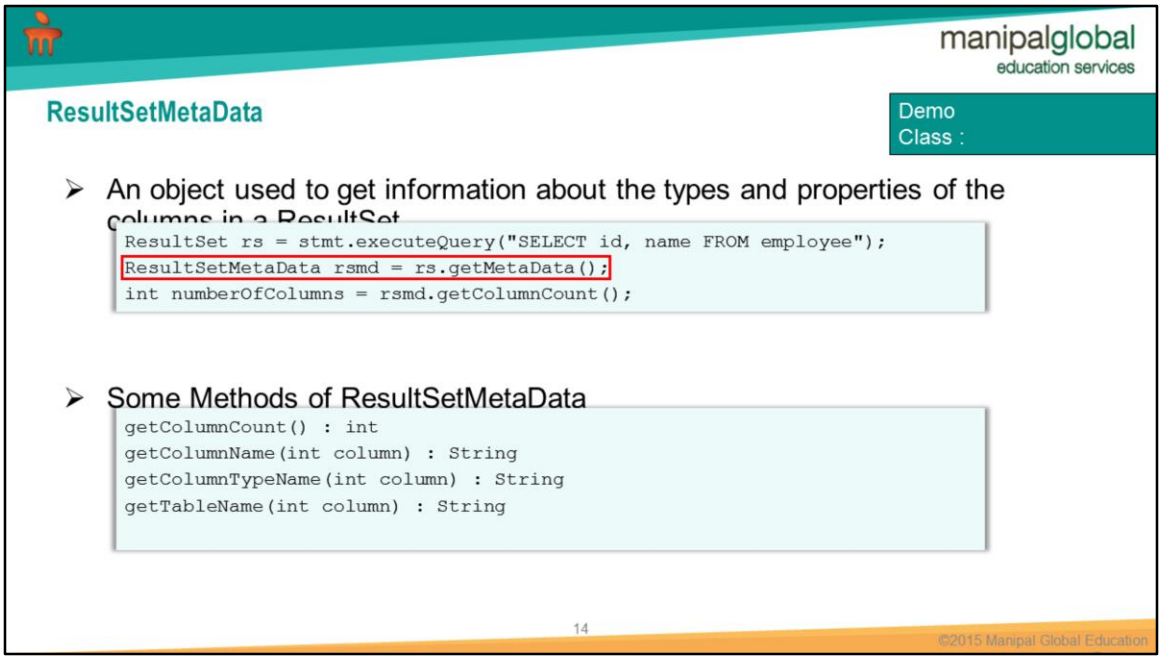
```
con.setAutoCommit(false);
Statement stmt = con.createStatement();
stmt.executeUpdate(sql1);
Savepoint savepoint1 = con.setSavepoint("SavePoint1");
stmt.executeUpdate(sql2);
stmt.executeUpdate(sql3);
..
       con.rollback(savePoint1);
       con.commit();
```

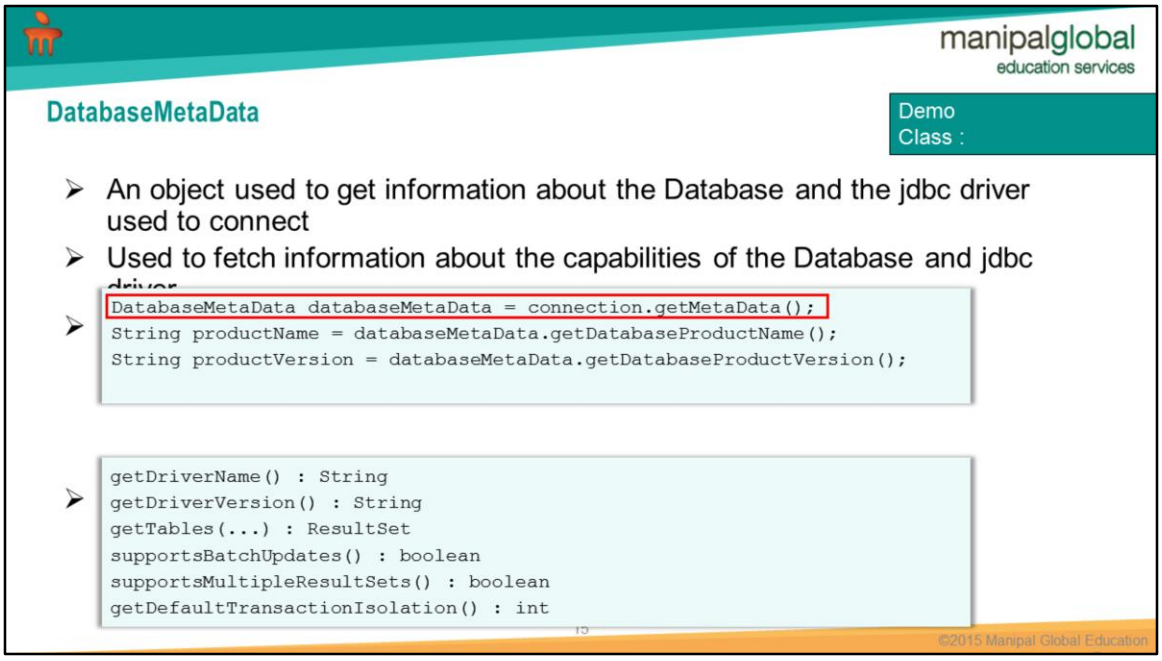

manipalglobal
education services
Concurrency Problems
Problems may occur if multiple transactions use the same data at the same time. Concurrency problems include
Dirty Read
Transaction T1 is changing a row and Transaction T2 reads the same row before T1 commits
If T1 rolls back, data read by T2 is incorrect or dirty
NonRepeatable Read
Non Repeatable Reads occur when in a transaction T1 same query yields different results.
This happens when another transaction T2 updates the data accessed by the query of T1
Phantom Read
Phantom read occurs where in a transaction T1 same query executes twice, and the second result set includes rows that weren't visible in the first result set.
This situation is caused by another transaction T2 inserting new rows between the execution of the two queries
©2015 Manipal Global Education

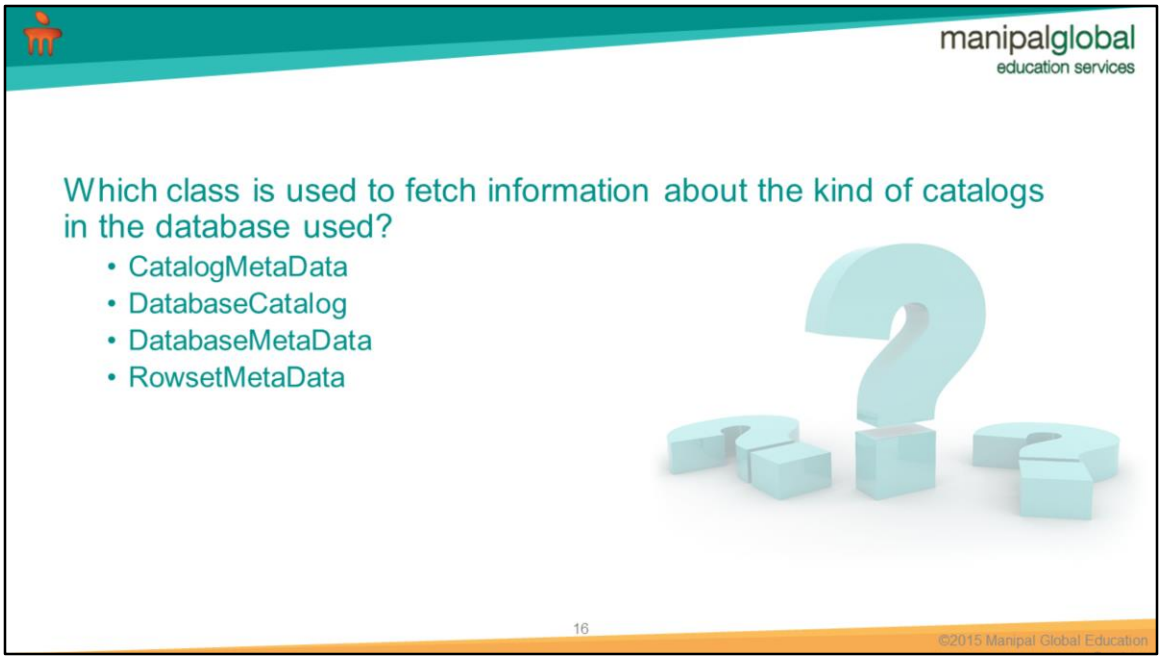

manipalglobal
education services
TRANSACTION ISOLATION
To avoid conflicts during a transaction, a DBMS uses locks for blocking access to data by other transactions
How locks are set is determined by what is called a transaction isolation
Isolation Level | Dirty Read | Non Repeatable Read | Phantom Read
TRANSACTION_READ_UNCOMMITTED | Allowed | Allowed | Allowed
TRANSACTION_READ_COMMITTED | Prevented | Allowed | Allowed
TRANSACTION_REPEATABLE_READ | Prevented | Prevented | Allowed
TRANSACTION_SERIALIZABLE | Prevented | Prevented | Prevented
conn.setTransactionIsolation(Connection.TRANSACTION_REPEATABLE_READ);
11
©2015 Manipal Global Education

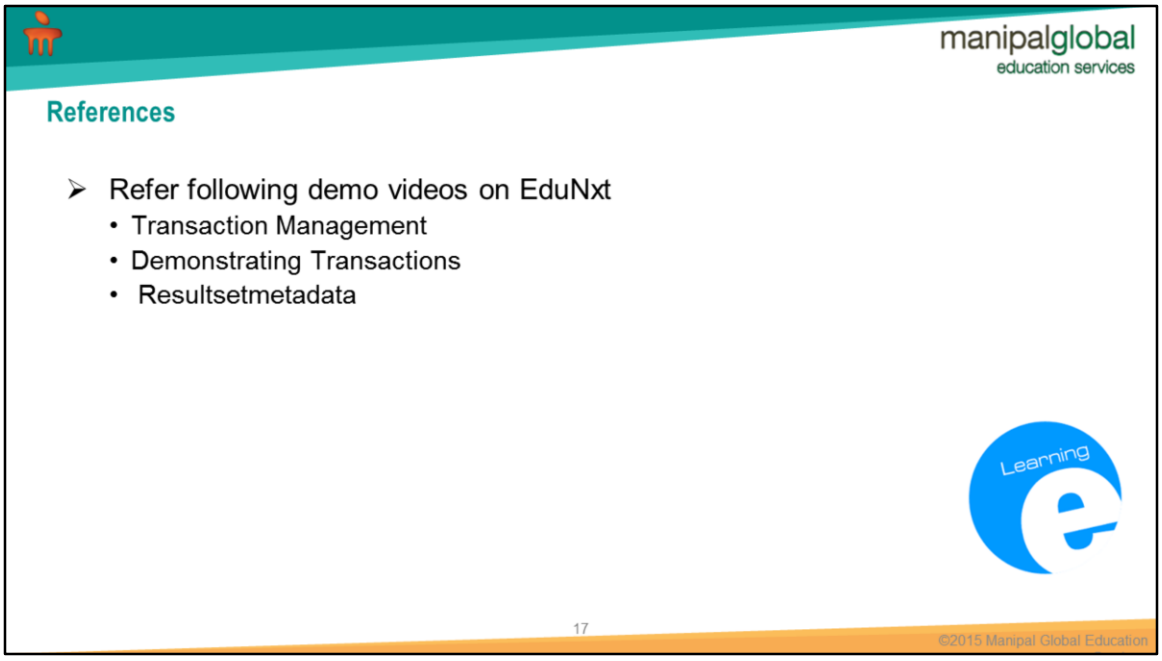

manipalglobal
education services
Which of the statements are correct about JDBC transactions? (2 correct answers)
• A transaction is a set of successfully executed statements in the database
• A transaction is completed when commit() or rollback() is called on the Connection object
• A transaction is completed when commit() or rollback() is called on the Transaction object
• A transaction is completed when close() is called on the Connection object
12
©2015 Manipal Global Education

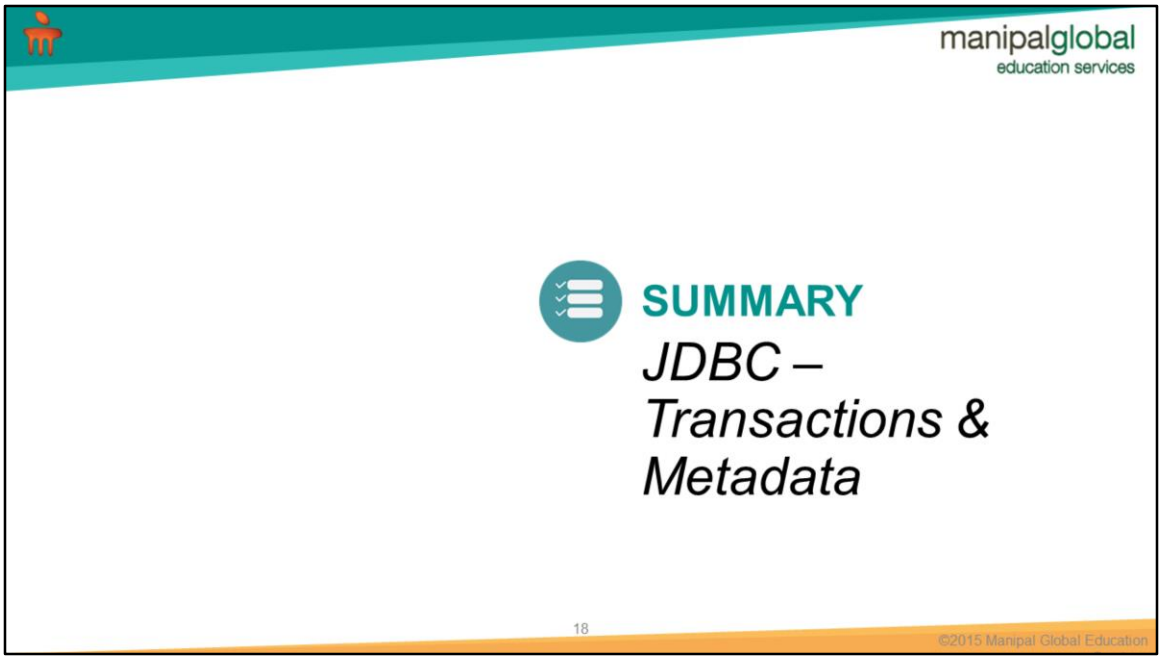

manipalglobal
education services
CONCEPT
Metadata
13
©2015 Manipal Global Education

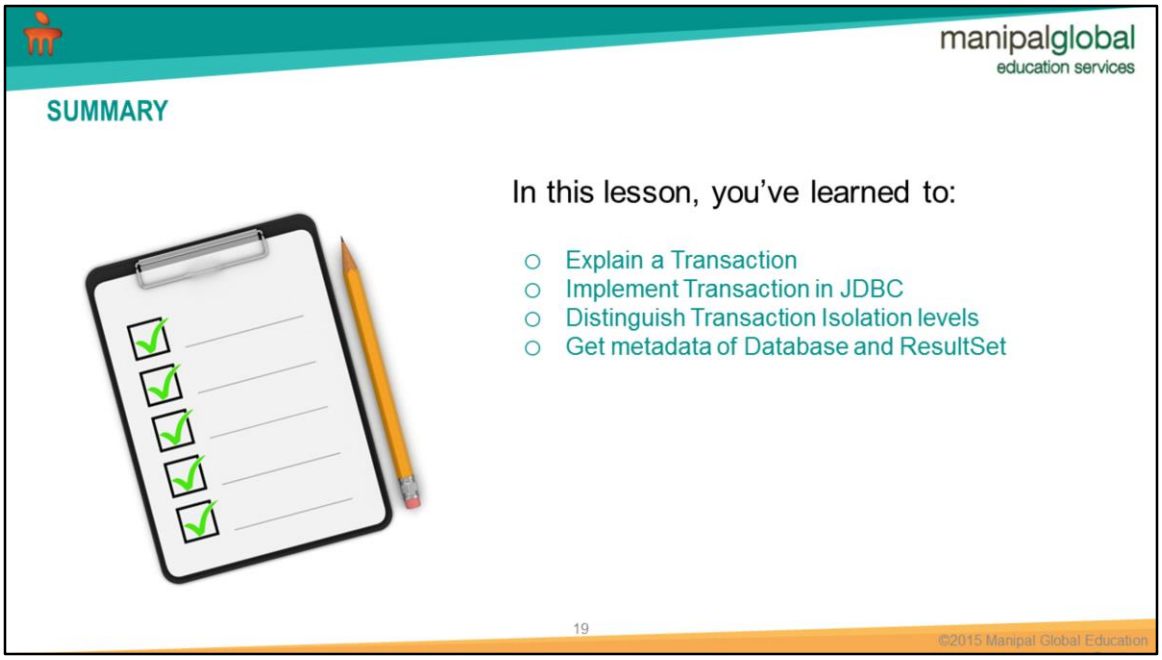

manipalglobal
education services
ResultSetMetaData
Demo
Class :
An object used to get information about the types and properties of the columns in a ResultSet
ResultSet rs = stmt.executeQuery("SELECT id, name FROM employee");
ResultSetMetaData rsmd = rs.getMetaData();
int numberOfColumns = rsmd.getColumnCount();
Some Methods of ResultSetMetaData
getColumnCount() : int
getColumnName(int column) : String
getColumnTypeName(int column) : String
getTableName(int column) : String
14
©2015 Manipal Global Education


manipalglobal
education services
DatabaseMetaData
Demo
Class :
An object used to get information about the Database and the jdbc driver used to connect
Used to fetch information about the capabilities of the Database and jdbc driver
DatabaseMetaData databaseMetaData = connection.getMetaData();
String productName = databaseMetaData.getDatabaseProductName();
String productVersion = databaseMetaData.getDatabaseProductVersion();
getDriverName() : String
getDriverVersion() : String
getTables(...) : ResultSet
supportsBatchUpdates() : boolean
supportsMultipleResultSets() : boolean
getDefaultTransactionIsolation() : int
15
©2015 Manipal Global Education


manipalglobal
education services
Which class is used to fetch information about the kind of catalogs in the database used?
• CatalogMetaData
• DatabaseCatalog
• DatabaseMetaData
• RowsetMetaData
16
©2015 Manipal Global Education


manipalglobal
education services
References
Refer following demo videos on EduNxt
Transaction Management
Demonstrating Transactions
Resultsetmetadata
Learning
17
©2015 Manipal Global Education


manipalglobal
education services
SUMMARY
JDBC –
Transactions &
Metadata
18
©2015 Manipal Global Education


manipalglobal
education services
SUMMARY
In this lesson, you've learned to:
Explain a Transaction
Implement Transaction in JDBC
Distinguish Transaction Isolation levels
Get metadata of Database and ResultSet
19
©2015 Manipal Global Education