

Practical MySQL

Session 8

Transactions, Performance Management, and Backup

Session Overview

- Define transaction
- List different commands used in transactions
- Describe transaction using JDBC Driver
- Outline different techniques to improve and manage database performance
- Explain Replication

For Aptech Centre Use Only

Transaction (1-2)

In MySQL, a transaction is a set of logical statements resulting in a single operation. It is a collection of SQL queries or sequential group of statements and operations.

These statements may consist of single or grouped statements of SELECT, INSERT, UPDATE, or DELETE.

Depending upon whether the transaction is successful or not, it will be 'committed' or 'rolled back' respectively.

Committed here means saved and rolled back means reversed to original state before the transaction began.

Transaction (2-2)

Consider the scenario where a customer wants to withdraw \$100 from the savings account of his bank. Customer would have to perform following steps:

Insert the ATM Card and select the Savings account option



Check the account balance and then, select the amount to be withdrawn



Enter the ATM pin and wait during the processing



In a database, two outcomes are possible when there are multiple modifications happening in the database.

Properties of Transactions

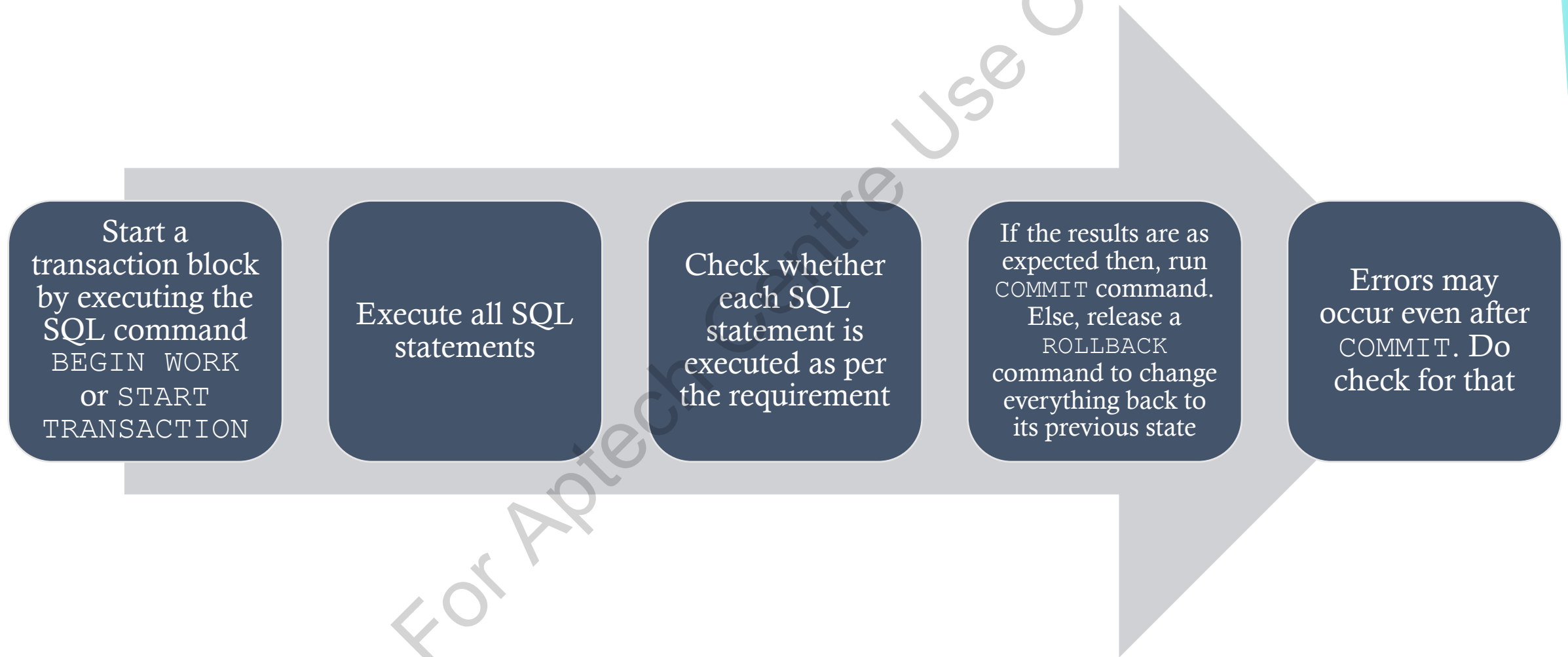
Atomicity: Atomicity ensures that every transaction is either executed successfully or is completely failed.

Isolation: A transaction must be executed without interfering with any other transaction that is under execution. The property of isolation ensures this.

Durability: Durability ensures that if any committed transaction is successful or fails, the results persist in the system throughout even if the system crashes. For this, a transaction must be executed completely.


Consistency: Whenever a transaction takes place, the values at the end of the transaction must be consistent throughout the process.

Starting a Transaction



COMMIT, ROLLBACK, and AUTO COMMIT

COMMIT means that the changes resulting from a transaction become permanent and can also be made visible to other transactions in those sessions. It is a command, which saves the changes executed by the transaction in the database.



ROLLBACK cancels all the modifications made by a transaction. It is used as a command to undo transactions that have not already been saved or committed.



SAVEPOINT creates a point within a transaction in which the statements after the point can be reverted. It is used to roll back the transaction up to a specific point without rolling back the entire transaction.

Commands used in Transaction

SET TRANSACTION: SET TRANSACTION sets a property on a transaction with READ WRITE or READ ONLY.

Syntax: SET TRANSACTION [READ WRITE | READ ONLY] ;

COMMIT: All modifications are made as a unit in the database using this command. It saves all the changes made by the transaction to the database permanently.

Syntax: COMMIT;

ROLLBACK: If any error occurs with any of the SQL grouped statements, all changes will be aborted. This process is called rollback.

Syntax: <transaction statement>; ROLLBACK;

BEGIN TRANSACTION: BEGIN TRANSACTION indicates the start point of a transaction.

Syntax: BEGIN TRANSACTION transaction_name;

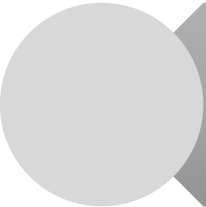
SAVEPOINT: SAVEPOINT works similar to a bookmark.

Syntax: ROLLBACK TO SAVEPOINT_NAME;

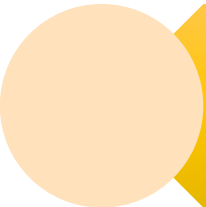
Transaction Using JDBC Driver



Java applications with database-handling tasks use the Java Database Connectivity API.



This API often called as JDBC is used for making connections to the database, creating, and executing SQL queries or for viewing and modifying data in tables.

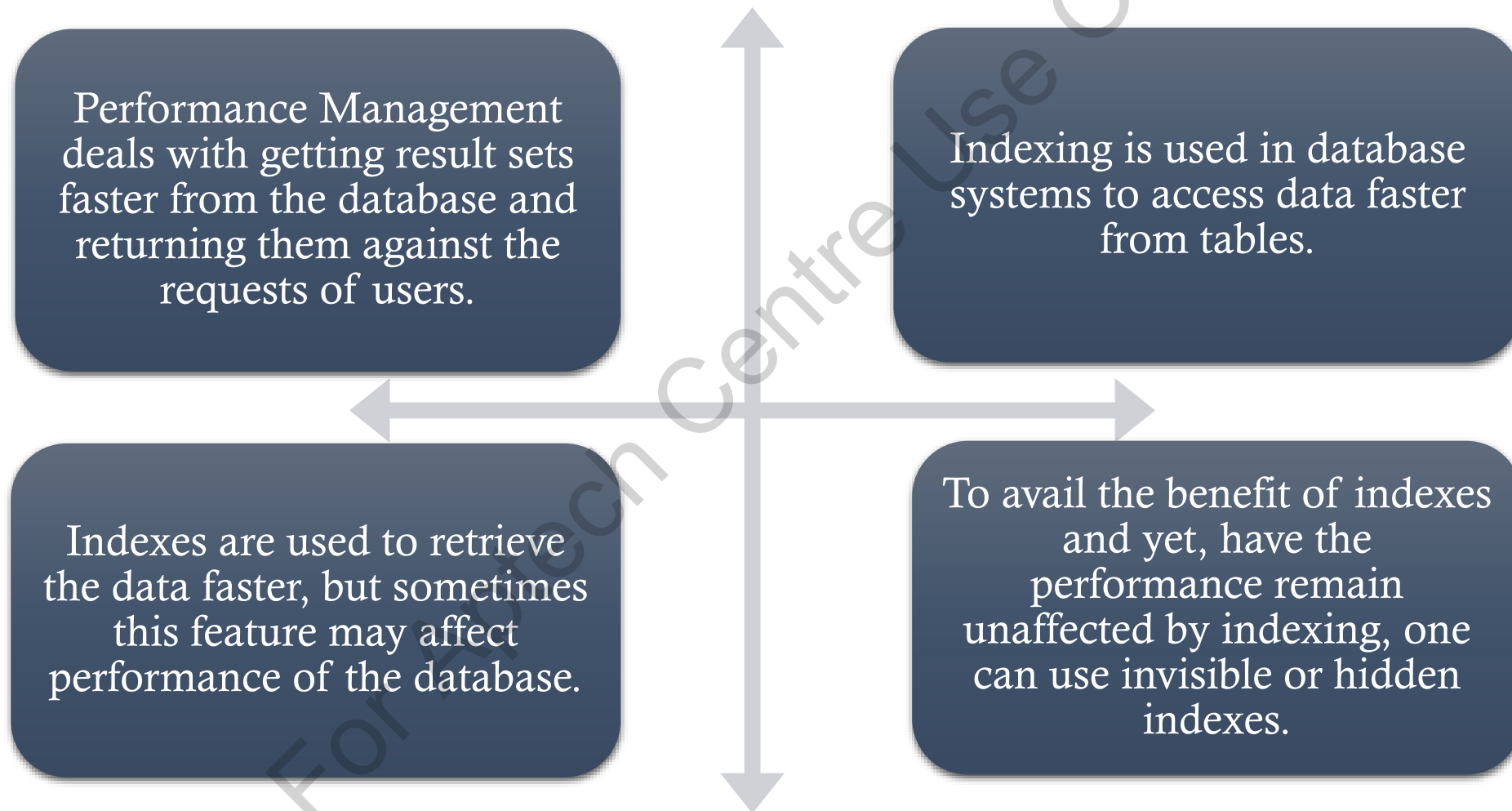


A JDBC driver is used to connect to the database. If a user does not want to use auto commit settings and switch to manual transactions to increase performance and use distributed transactions.

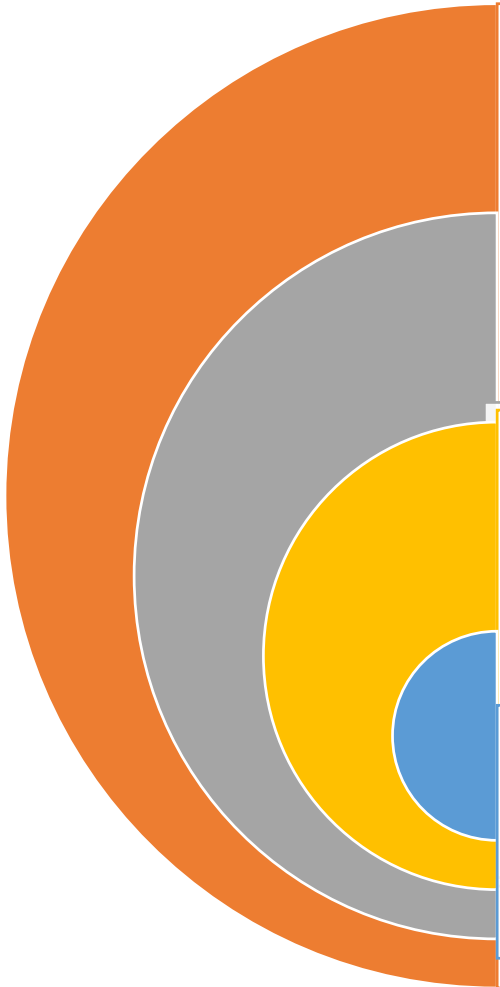


Thus, transaction management can be done by JDBC driver effectively.

Performance Management



Hiding an Indexed Column



Hidden indexes are applicable to indexes other than primary key index. They are not used in query optimization, but can be used to evaluate performance as if there were no indexes. By default, indexes are visible, but to make them invisible, the `INVISIBLE` keyword is used.

Note: Primary key index cannot be hidden.

HAVING with INDEX: `HAVING` is used in conjunction with a `GROUP BY` clause. Indexing is not used frequently with `HAVING`. The most efficient way to process `GROUP BY` is when an index is used to directly retrieve the grouping columns based on some aggregate condition. This is another technique to improve performance.

Setting the Cache: This is also one of the features of MySQL that speeds up data retrieval from a database. The speed of retrieving records from cached data memory that is, Random Access Memory is more than data retrieved from the disk. The individual cache size is 256 KB and total cache size is 10 MB.

Backup and Replication

Backup MySQL Database from Command Line: Database is useful in case of taking business decisions and in handling customer queries. Backups in databases must be used to recover lost data. Backups are always necessary and essential in order to recover from most disasters.

Following is the syntax for taking backup of the database: `mysqldump -u [username] -p[password] [database_name] > [dump_file.sql]`

Other Ways to Back Up the MySQL Database

- Logical backup by using `mysqldump`
- Physical backup
- Hot backup
- Partial

Basics of Replication

Replication is the process where the same data is kept in multiple servers. Replication of data plays an integral part in safeguarding any system. It determines how robust any arrangement can turn out to be. Failures or loss of data can occur at any point and replication can help fix that while allowing the user to enjoy a risk-free experience.

Replication refers to copying data from one MySQL database server which is known as 'Source' to one or more MySQL Database Servers which is known replicas.

MySQL replication allows a server/master to send all changes to server/slave while the slave tries to apply all changes in order to keep up to date with the master.

Summary

- In MySQL, a transaction is a set of statements resulting in a single operation.
- Transaction has four properties namely atomicity, isolation, durability, and consistency.
- Atomicity ensures the status of a transaction to be either successful or failed.
- Isolation property makes sure that each operation in a transaction is executed simultaneously and independent of each other.
- Durability preserves the data in the database even if the server crashes.
- Consistency is ensured when atomicity, isolation, and durability are followed in a system.
- Transaction management can be done in Java applications by JDBC Driver to increase performance of the database.