

# **Practical MySQL**

## **Trainer Guide**

For Aptech Centre Use Only

# **Practical MySQL**

## **Trainer Guide**

**© 2022 Aptech Limited**

All rights reserved.

No part of this book may be reproduced or copied in any form or by any means – graphic, electronic or mechanical, including photocopying, recording, taping, or storing in information retrieval system or sent or transferred without the prior written permission of copyright owner Aptech Limited.

All trademarks acknowledged.

### **APTECH LIMITED**

Contact E-mail: [ov-support@onlinevarsity.com](mailto:ov-support@onlinevarsity.com)

Edition 1 - 2022



## PREFACE

MySQL is one of the most popular Database Management System that enables us to enter, organize, and select data from a database. MySQL is open source and available in free as well commercial editions. This book explains how to create and manage databases, tables, and database objects in MySQL. The book also describes queries, joins, functions, and stored procedures. New enhancements in MySQL 8.0 such as those made to server security, MySQL database maintenance, backup, replication, transaction management, performance optimization, and storage systems are also explored in this book. Features such as the new schemas, spatial data types, and so on are also discussed.

The faculty/trainer should teach the concepts in the theory class using the slides. This Trainer's Guide will provide guidance on the flow of the module and also provide tips and additional examples wherever necessary. The trainer can ask questions to make the session interactive and also to test the understanding of the students.

The knowledge and information in this book is the result of the concentrated effort of the Design Team, which is continuously striving to bring to you the latest, the best and the most relevant subject matter in Information Technology. As a part of Aptech's quality drive, this team does intensive research and curriculum enrichment to keep it in line with industry trends and learner requirements.

# Contents

Session 1 – Introduction to MySQL

Session 2 – Essentials of MySQL

Session 3 – Working with Data in MySQL

Session 4 – Joins

Session 5 – Subqueries

Session 6 – Functions and Stored Procedures

Session 7 – MySQL Clauses and Indexes

Session 8 – Transactions, Performance Management, and Backup

Session 9 – Replication and Scalability

Session 10 – Partitioning

Session 11 – Performance Optimization in MySQL

Session 12 – Concept of Storage System and Management

## Session 1: Introduction to MySQL

### Slide 2

#### Session Overview

- Define database
- Explain and list different elements of database management system
- List different types of database models
- Describe the principles of MySQL
- Explain how to connect to Database Server using MySQL Workbench
- Explain features, limitations, and deployment of MySQL
- Explain different elements of Normalization in DBMS

#### Instruction(s) to the trainer:

Show slide 2 and give a brief overview on principles of MySQL. Explain features, limitations, and deployment of MySQL. Inform about the concepts of database and different elements of database management system. Finally, discuss with the students about different elements of Normalization of DBMS.

## Database and Database Model

The diagram illustrates the classification of database models. A central orange box labeled "Database Model" branches out to five smaller orange boxes, each representing a different model: "Hierarchical Database Model", "Network Model", "Relational Database Model", "Object Oriented Database Model", and "Distributed Database Model". To the left of the main title, there is an icon of three stacked cylinders representing a database. Next to it is a green rounded rectangle containing the text: "Database can be defined as a collection of logically related data and information stored in a standardized format."

© Aptech Limited

Practical MySQL / Session 1 / 3 of 16

### Instruction(s) to the Trainer:

Begin the session by asking the question:

Do you know what is a database?

Encourage participants to respond. Appreciate the responses.

- Then, show slide 3 and explain to students about Database and Database Model.
- Say that database is a collection of logically related data and information stored in a standardized format and database model is the logical structure of the database.
- Discuss about some of the common Database Models and their advantages and disadvantages.

### 1. Hierarchical Database Model

This model is like a tree structure where records form the nodes and fields form the branches. A record in a database can be defined as collection of fields with different data types. It is a natural method of implementing records and their relationship. However, it has limitations in representing the relationships. It does not adapt to the new format that occur in real world.

### 2. Network Model

In this, record type can have multiple owners and the data is represented as records using links. There are three components of the network model, namely - Network Schema, Sub-schema, and Data Management Language. It is useful when there is a requirement for showing a many-

to-many relationship. However, database structure will become very complex as pointers are used to maintain all records.

### **3. Relational Database Model**

In this model, data is organized in the form of rows and columns similar to a table. There are certain terminologies used in this model, where tables are called as ‘Relations’ or ‘instance’ with rows and columns. This model is easy to use and helps in improve the performance. Drawback in this model is It requires skilled human resources to implement.

### **4. Object-Oriented Database Model**

It supports Object-Oriented Programming features such as Object, class, Abstraction, Encapsulation, Inheritance, and Polymorphism that will make it easy to understand. It is used when there is a relationship among data elements that are more important than data items. One of the disadvantage of this model is its complexity and increases the cost.

### **5. Distributed Database Model**

In this database model, the computers communicate and exchange data using data analytics, cloud computing, and other means of communication. This model is easier to expand. However, it creates a large overhead on entire system.

#### **Additional Information:**

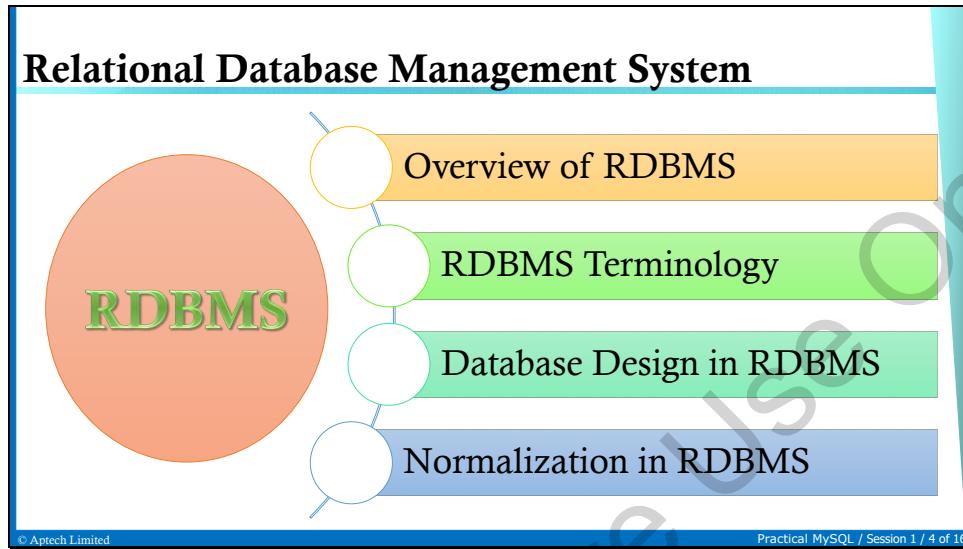
Refer to following links for more information:

- <https://www.javatpoint.com/mysql-create-database>
- <https://dev.mysql.com/doc/workbench/en/wb-database-synchronization.html>

Ask students following question. Wait for a response before you answer.

**In-Class Question:** What is the full form of DBMS?

**Answer:** Database Management System



**Instruction(s) to the Trainer:**

Show slide 4 and explain Relational Database Management System (RDBMS) in detail. Tell students that RDBMS helps to maintain security, accuracy, consistency, and integrity for data. It is based on the principles of E.F. Codd. Examples of most popular RDBMS are MySQL, Oracle, IBM DB2, and SQL Server.

Discuss about the overview of RDBMS and its terminology. Inform students that RDBMS enable users to implement a database with tables and indexes. It also maintains the referential integrity between rows of various tables. Each table contains records. Each record defines a specific number of fields.

Discuss about the database design and normalization in RDBMS. Inform that database designing is a process of facilitating the designing, development, implementation, and maintenance of DBMS. Tell students that the database designer decides how the data elements correlate with each other and which data must be stored. It produces logical and physical design models of database.

Data Modelling can be defined as the process of analyzing data objects and their relation with other objects. Data can be modeled at different levels of abstraction. This process starts by collecting information from stakeholders and the end-users of the system about business requirements. It is important that the database should be designed accurately to maintain

information consistency, eliminate redundant data from the table, execute queries efficiently on the database, and also improve the database performance.

Tell students that normalization is a design technique that helps the user to efficiently organize data in a database and reduce data redundancy. It is important to improve usability of the dataset allowing databases to take less space.

Database Normalization is categorized into following types:

- 1NF (First Normal Form)
- 2NF (Second Normal Form)
- 3NF (Third Normal Form)
- BCNF (Boyce-Codd Normal Form)
- 4NF (Fourth Normal Form)
- 5NF (Fourth Normal Form)

**Normalization Definition:** Normalization is the process of organizing the data in the database. It removes the redundancy from a relation or set of relations.

**Additional Information:**

Refer to following links for more information:

- <https://www.javatpoint.com/mysql-create-database>
- <https://www.tutorialspoint.com/sql/sql-databases.htm>

Ask students following question. Wait for a response before you answer.

**In-Class Question:** Mention the systems that use RDBMS.

**Answer:** A) IBM

B) Oracle

C) Microsoft SQL Server

## Overview of RDBMS

### *Difference between DBMS and RDBMS*

DBMS	RDBMS
Data is contained as a file	Data is contained in table format
Data is stored in either a hierarchical form or a navigational form	The tables have identifiers called as Primary Keys
Normalization is not possible	Normalization is possible
There is no security for data in DBMS	There is an integrity constraint
Does not support distributed database	Supports distributed database
Used for storing small data used for small business	Used for handling a large amount of data and supports multiple users

### **Instruction(s) to the Trainer:**

Using slide 5, discuss the differences between DBMS and RDBMS.

### **Additional Information:**

Refer to following links for more information:

- <https://www.javatpoint.com/difference-between-dbms-and-rdbms>
- <https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/>

## Uses of MySQL

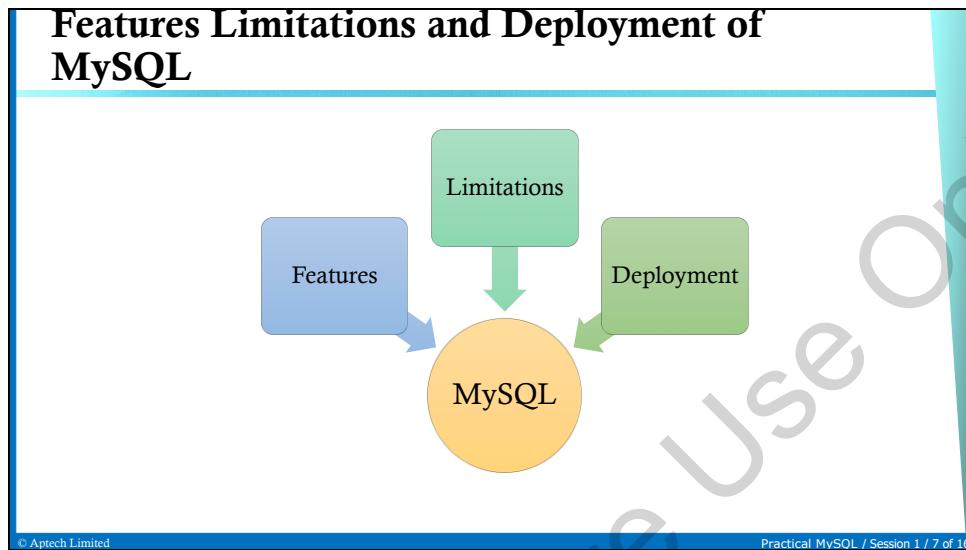
Used on multiple platforms

Provides comprehensive support for Web Application Development	Used for popular Web applications such as Twitter, Instagram Facebook, YouTube, and Google	Used by database-driven applications and Content Management Systems (CMS) such as WordPress, Drupal, Joomla, phpBB
--	--	--

© Aptech Limited Practical MySQL / Session 1 / 6 of 16

**Instruction(s) to the Trainer:**

Show slide 6 and explain the uses of MySQL. Say that MySQL can be used on multiple platforms (Operating Systems).



**Instruction(s) to the Trainer:**

Using slide 7, discuss about the features, limitations, and deployment of MySQL.

Say that MySQL is a widely used RDBMS that primarily works on relational database. Inform students that it also has some limitations when it comes to big data analysis.

**Additional Information:**

Refer to following links for more information:

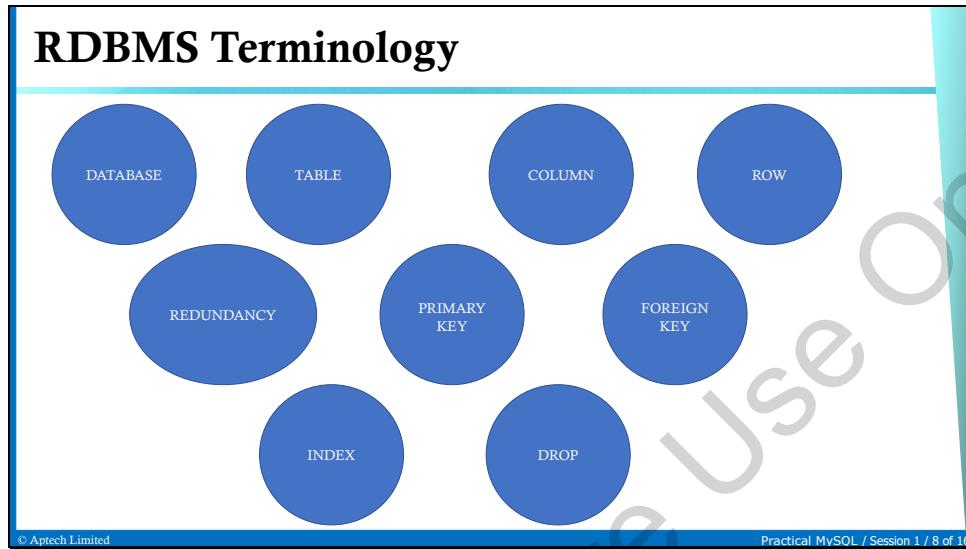
- <https://www.javatpoint.com/mysql-create-database>
- [https://www.w3schools.com/mysql/mysql\\_rdbms.asp](https://www.w3schools.com/mysql/mysql_rdbms.asp)

Ask students following question. Wait for a response before you answer.

**In-Class Question:** What are the features of MySQL?

**Answer:** MySQL is widely used RDBMS that primarily works on relational database model. It is highly quick and reliable.

## Slide 8



### Instruction(s) to the Trainer:

Show slide 8, discuss about the common terminologies used in RDBMS in detail.

**Database:** In simple words, it is a collection of data and information stored in a standardized format.

**Table:** A table is used to store data of multiple data types in form of rows and columns. It can be sorted too.

**Column:** This refers to the series of cells that are aligned vertically in a table.

**Row:** This refers to a series of cells aligned horizontally in a table.

**Redundancy:** This refers to storing data twice or duplicating data.

**Primary Key:** A primary key is used to maintain uniqueness in the table and avoid duplicate data.

**Foreign Key:** A foreign key is used to relate one table with another. It can accept duplicate values.

**Index:** It retrieves data faster. It is similar to an index page on a book.

**Drop:** Drop is an operation to remove or delete a database or database object. For example, a database can be dropped.

### Additional Information:

Refer to following links for more information:

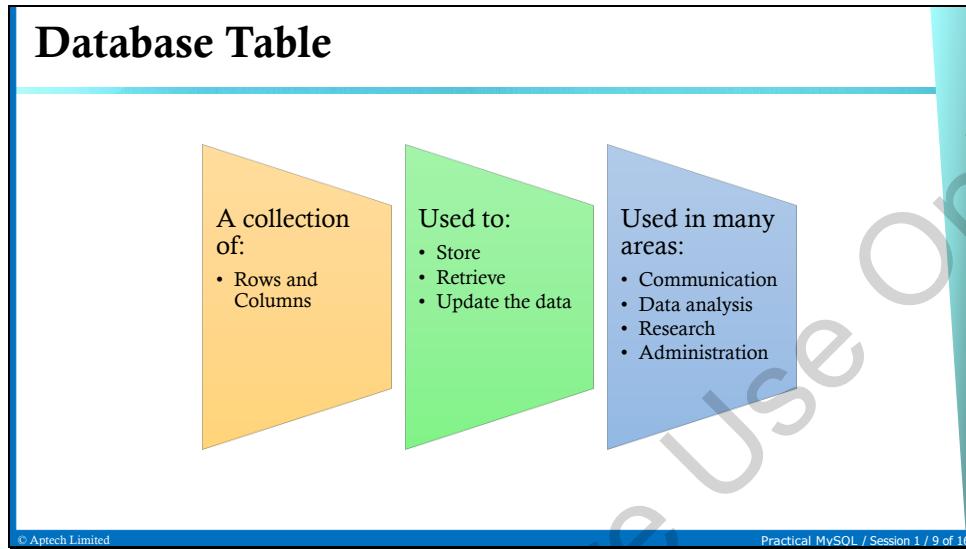
- <https://www.javatpoint.com/mysql-create-database>
- [https://www3.ntu.edu.sg/home/ehchua/programming/sql/MySQL\\_Beginner.html](https://www3.ntu.edu.sg/home/ehchua/programming/sql/MySQL_Beginner.html)

Ask the students following question. Wait for a response before you answer.

**In-Class Question:** What is the key to represent relationship between tables?

**Answer:** Foreign Key

## Slide 9



### Instruction(s) to the Trainer:

Show slide 9 and explain Database Table. Tables in database can be defined as a collection of rows and columns. The data is stored in tables and consists of related data entries in multiple columns and rows.

Discuss about its uses. It is used to store, retrieve, and update the data. Tables are widely used in many areas such as communication, data analysis, research, administration, and so on.

Tell students one can perform many operations on a table such as create, drop, delete, alter, and rename. A table should have name, rows, columns, and data types of the columns.

### Additional Information:

Refer to following links for more information:

- <https://www.tutorialspoint.com/sql/sql-rdbms-concepts.htm>
- <https://www.javatpoint.com/dbms-sql-table>

## Normalization in RDBMS

Normalization is the process of organizing the data in the database.

Database Normal Forms	Keys and Constraints	INF Rules
-----------------------	----------------------	-----------

© Aptech Limited Practical MySQL / Session 1 / 10 of 16

**Instruction(s) to the Trainer:**

Show slide 10 and explain the students about Normalization in RDBMS.

Discuss about the Database normal forms, keys and constraints and INF rules.

**Additional Information:**

Refer to following links for more information:

- <https://www.javatpoint.com/dbms-normalization>
- <https://www.guru99.com/database-normalization.html>

Ask students following question. Wait for a response before you answer.

**In-Class Question:** Why is Normalization required in RDBMS?

**Answer:** It is important that a database is normalized to minimize redundancy (duplicate data) and to ensure only related data is stored in each table.

## Database Design in RDBMS

Database Designing is a process of facilitating the designing, development, implementation, and maintenance of DBMS

Data Modelling	Types of Data Modelling: Conceptual Data Model Logical Data Model Physical Data Model	Importance of Database Design
----------------	--	-------------------------------

© Aptech Limited Practical MySQL / Session 1 / 11 of 16

**Instruction(s) to the Trainer:**

Show slide 11 and discuss about the Database design in RDBMS.

Discuss about Data Modelling, types of Data Modelling, and then explain the importance of Database Design.

**Additional Information:**

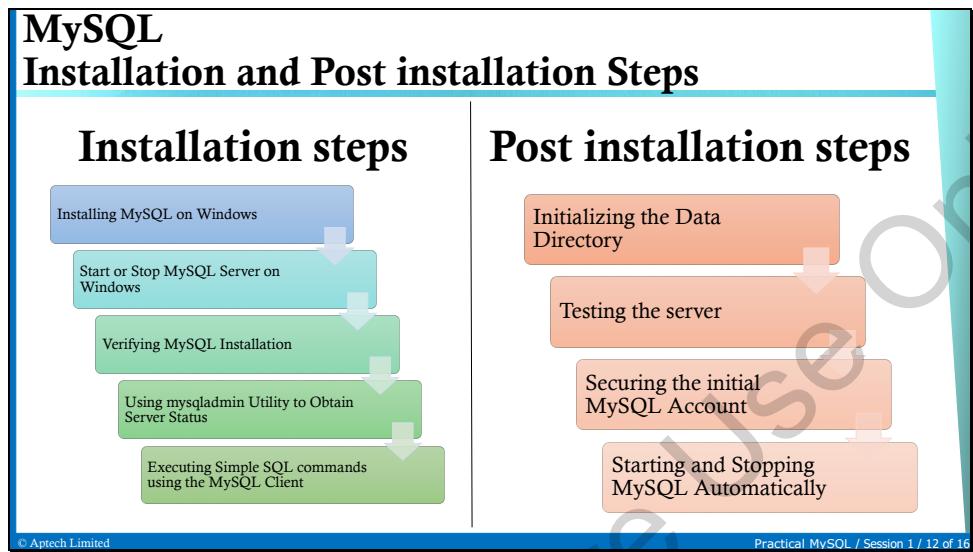
Refer to following links for more information:

- <https://www.javatpoint.com/database-design>
- <https://www.guru99.com/database-design.html>

Ask students following question. Wait for a response before you answer.

**In-Class Question:** Define Data Modelling.

**Answer:** Data Modelling can be defined as a process of analyzing data objects and their relation with other objects.



**Instruction(s) to the Trainer:**

Using slide 12, discuss various steps to be followed for MySQL Installation and post installation.

**Additional Information:**

Refer to following links for more information:

- <https://www.javatpoint.com/how-to-install-mysql>
- <https://www.mysqltutorial.org/install-mysql/>

## Running MySQL at Boot Time

**Step 1:** From Applications → Ubuntu Software Center, search for 'boot up manager'.

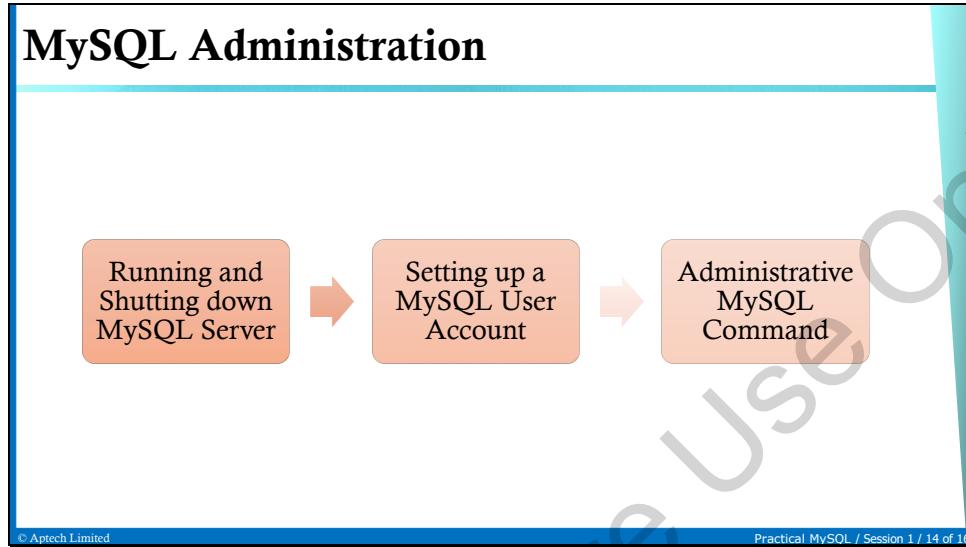
**Step 2:** Click 'boot up manager', go to System → Administration → BootUP-Manager.

**Step 3:** Open webmin browser, type <https://localhost:10000/> and start working on the program.

© Aptech Limited Practical MySQL / Session 1 / 13 of 16

**Instruction(s) to the Trainer:**

Show slide 13 and explain the steps to be followed to run the MySQL at Boot time.



**Instruction(s) to the Trainer:**

Show slide 14 and discuss about MySQL Administration that enables the DBAs to perform a wide range of functions.

Say that the user will find information on server start-up and shutdown, setting up user account, and administrative commands.

Following are the important MySQL Administrative commands, which can be used as per the user's requirement:

- **USE Databasename:** Select a database in the MySQL work area.
- **SHOW DATABASES:** It shows the lists of databases that are accessible.
- **SHOW TABLES:** It shows the list of tables in the database.
- **SHOW COLUMNS FROM tablename:** It shows the attributes, key information, types of attributes, and other information for a table.
- **SHOW INDEX FROM tablename:** It shows the details of all indexes on the table.
- **SHOW TABLE STATUS LIKE tablename\G:** It reports details of the MySQL DBMS performance and statistics.

**Additional Information:**

Refer to following links for more information:

- <https://www.tutorialspoint.com/mysql/mysql-administration.htm>
- <https://www.mysqltutorial.org/mysql-administration>

## MySQL Connection Types

MySQL Server Connection using Command-Line Client

Connect to Database Server using MySQL Workbench

© Aptech Limited Practical MySQL / Session 1 / 15 of 16

**Instruction(s) to the Trainer:**

Show slide 15 and explain MySQL Connection types in detail.

Inform students that user can connect with MySQL Database Server on the same machine or a remote machine using `mysql` binary at the command prompt. After connecting with the MySQL server, the user must connect to the database. Mention that there are multiple ways to connect with the database server. Once the MySQL server is installed, it can be connected using any of following client programs:

- MySQL Server Connection using Command-Line Client
- Connect to Database Server using MySQL Workbench

**Additional Information:**

Refer to following links for more information:

- <https://www.javatpoint.com/mysql-create-database>

## Summary

- MySQL is an open source RDBMS developed, distributed, and supported by Oracle.
- MySQL administration is used to perform administrative tasks such as – monitoring, configuring, managing users, and so on.
- Database design is the technique of organizing the data. A well designed database provides access to accurate and up-to-date information.
- Commands in MySQL are powerful directives used to perform specific tasks.
- Normalization is the process of organising the data to reduce data redundancy and avoid anomalies such as update, insert, delete, and so on.
- Normalization in MySQL reduces data redundancy by eliminating insertion and updating records.

Use slide 16 to summarize the session. End the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

## Session 2: Essentials of MySQL

### Slide 2

#### Session Overview

- List MySQL data types
- List and explain different commands in MySQL
- Explain the basics of creating a database
- Identify ways to create and drop a database in MySQL
- Identify ways to select a database in MySQL
- Outline how to create, alter and drop a table

#### Instruction(s) to the trainer:

Show slide 2 and give students a brief overview of the current session in the form of session objectives. Inform students that the session begins by introducing the list of MySQL data types. Next, it explains the different types of commands used in MySQL. The session further explains the basics of creating a database. It also discusses various ways of creating, dropping and selecting a database in MySQL. Finally, the session explains how to create, alter, and drop a table in MySQL.

## Slide 3

# SQL Commands

**SQL commands are instructions used to communicate with the database.**

Data Definition Language (DDL)

Data Manipulation Language (DML)

© Aptech Limited Practical MySQL / Session 2 / 3 of 15

### In-Class Explanations

#### Instruction(s) to the trainer:

Use slide 3 to introduce students to SQL commands and the two basic types of SQL commands - DDL and DML. Tell them that SQL commands are instructions that are used to communicate with the database. Inform students that both these commands are not case-sensitive.

Tell them that DDL commands are used to modify the structure of databases and database objects. These commands are auto-committed and database tables and objects will be permanently stored in the database.

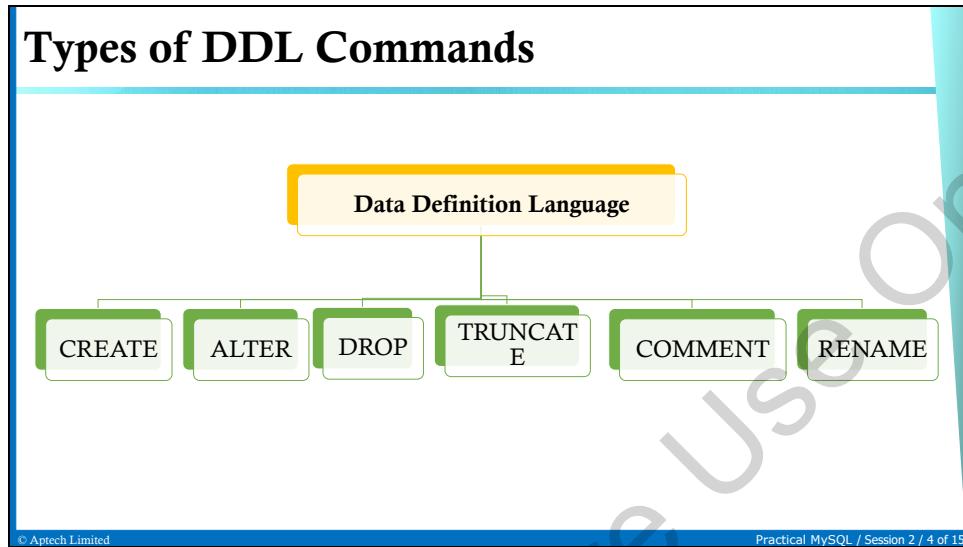
DML commands are used to manipulate data in a database table. DML operations are performed by used SQL statements such as SELECT, INSERT, DELETE, and so on.

#### Additional Information:

Refer to following links for more information:

<https://www.javatpoint.com/dbms-sql-command>

<https://www.edureka.co/blog/sql-commands>



**Instruction(s) to the trainer:**

Use slide 4 to introduce students to different types of DDL commands and the purposes they are used for in MySQL queries.

- **CREATE** command is used to create databases as well as objects in a database such as a table, view, index, and so on.
- **ALTER** command is used to restructure the database, table, and database objects.
- **DROP** command is used to delete databases as well as database objects from the database such as a table, view, index, and so on.
- **TRUNCATE** command is used to remove all rows from the table and at the time, keeping structure of the table the same.
- **COMMENT** command is used to specify some more information about the query. It will not be considered as a SQL query.
- **RENAME** command is used to change the name of the table.

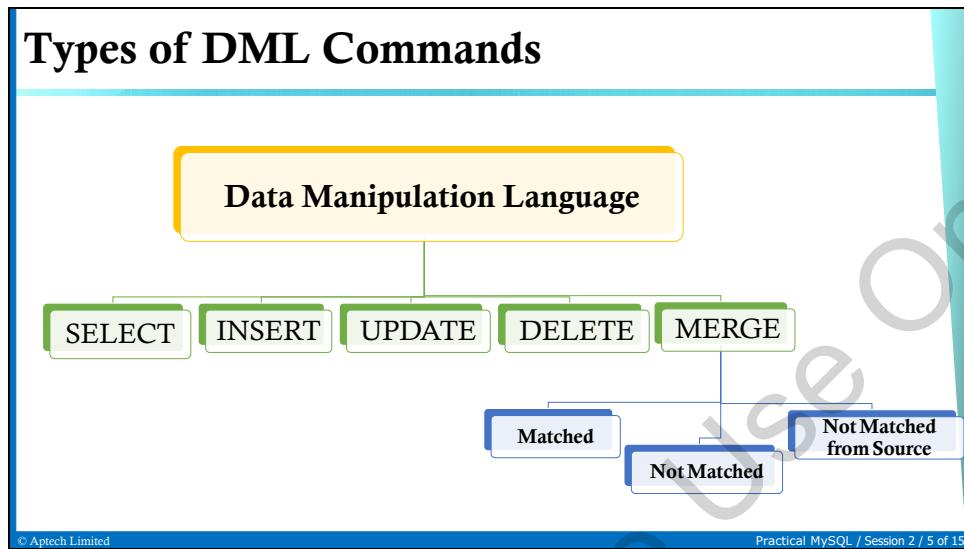
**Additional Information:**

Refer to following links for more information:

<https://www.edureka.co/blog/sql-commands>

<https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/>

## Slide 5



### Instruction(s) to the trainer:

Use slide 5 to introduce students to different types of DML commands and the purposes they are used for in MySQL queries.

- . SELECT command is used to fetch or retrieve records from tables.
- . INSERT command is used to insert the records into the table.
- . UPDATE command is used to modify data in a table. Tell students that if the WHERE clause is not specified, all records in the specified table will be modified.
- . DELETE command is used to delete all records or specific records from a table.
- . MERGE command is used to combine INSERT, UPDATE, and DELETE commands into a single SQL statement.
  - . There are three types of Merge commands - Matched, Not Matched, and Not Matched with Source.

Ask students following question. Wait for a response before you answer.

**In-Class Question:** Which command should be used to combine INSERT, UPDATE, and DELETE commands into a single SQL Statement?

**Answer:** The MERGE command can be used to do this task.

**Additional Information:**

Refer to following links for more information:

<https://www.edureka.co/blog/sql-commands>

<https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/>

## Slide 6

Types of the Merge Command		
Matched	Not Matched	Not Matched by Source
<p>The user must define matched rows that pass the MERGE condition. If the values are matched, one must use UPDATE query to update the values of one table's column to another table's columns.</p>	<p>If rows of source table do not match with any equivalent rows present in target table, the user must use the INSERT query.</p>	<p>It defines the table rows in the given target table which do not have any equivalent rows in the particular source table. Apply the condition to remove or delete rows from the given target table to coordinate the target table with the records from the source one. The DELETE query can be used here.</p>

© Aptech Limited

Practical MySQL / Session 2 / 6 of 15

### Instruction(s) to the trainer:

Show slide 6 and tell students about the three conditions of the MERGE command in MySQL. These three conditions are:

- Matched:** In this case, the user should define matched rows that must pass the MERGE condition.
- Not Matched:** The user must use the INSERT query in case the rows of source table do not match with any equivalent rows present in target table.
- Not Matched by Source:** This condition should be applied to remove or delete rows from the given target table to coordinate the target table with the records from the source one.

### Additional Information:

Refer to following links for more information:

<https://www.edureka.co/blog/sql-commands>

<https://www.sqlshack.com/understanding-the-sql-merge-statement/>

<https://www.sqlservertutorial.net/sql-server-basics/sql-server-merge/>

## Slide 7

### Operations on a Database

A database is a collection of data and information in a standardized format.

Create	Select	Drop
By using DDL command	From the Command Prompt	By using mysqladmin
By using mysqladmin	By using Raw SQL	By using Raw SQL

© Aptech Limited Practical MySQL / Session 2 / 7 of 15

#### Instruction(s) to the trainer:

Use slide 7 to show how to perform some basic database operations in MySQL. A database in MySQL stores multiple tables and other related objects such as views, functions, and so on. Besides using the DDL command CREATE to create a database, one can also use mysqladmin to create databases.

The mysqladmin command is used to perform administrative operations. A database can also be created using mysqladmin.

By dropping a database, we mean deleting the database from the server. A database can be dropped by using mysqladmin command as well as raw SQL command.

Tell students that they can select a database using raw SQL or from the command prompt.

#### Additional Information:

Refer to following links for more information:

[https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp)

<https://www.javatpoint.com/sql-tutorial>

## Slide 8

# MySQL Data Types

Some basic characteristics to determine the data type are:

- Type of value (fixed or variable) it represents
- Storage space of fixed or variable length
- Indexed or non-indexed values

© Aptech Limited Practical MySQL / Session 2 / 8 of 15

### Instruction(s) to the trainer:

Show slide 8 and explain students about some basic characteristics to determine data type. Inform them that to identify a data type, they must know about:

- . The type of value it represents
- . The storage space or length of the variable
- . Whether it contains indexed or non-indexed values

Some common data types in MySQL are categorized as Numeric, String, Date and Time, Spatial, and JSON. These are further subcategorized as floating-point, integer, boolean, character, and so on.

### Additional Information:

Refer to following links for more information:

[https://www.w3schools.com/sql/sql\\_datatypes.asp](https://www.w3schools.com/sql/sql_datatypes.asp)  
<https://www.tutorialspoint.com/sql/sql-data-types.htm>  
<https://www.javatpoint.com/sql-data-types>

## Slide 9

### Different Types of Data Types in MySQL 1-2

**Numeric Data Type:** Stores numerals and integers

**Floating-Point Data Type:** Stores approximate data values

**Temporal Data Type:** Stores temporal information such as time, date, and so on

**Large Object Data Type:** Stores large amounts of unstructured or semi-structured data

#### Instruction(s) to the trainer:

Show slide 9 and list some of the most common types of data types in MySQL.

Tell students that Numeric data types represent whole numbers. For example, number of employees in an organization, number of people in a family, population of an entire country, and so on.

Inform students that floating-point data types can store a high range of decimal numbers for accuracy. These data types are mostly used in columns such as amount of a product, rate of an item, weights, measurements, and so on.

Temporal data types are used for storing information such as time, date, and so on. Every temporal data type has a valid range of values as well as zero.

A Large Object Data Type (LOB) is used for storing large amounts of unstructured or semi-structured data.

#### Additional Information:

Refer to following links for more information:

[https://www.w3schools.com/sql/sql\\_datatypes.asp](https://www.w3schools.com/sql/sql_datatypes.asp)

<https://www.tutorialspoint.com/sql/sql-data-types.htm>

<https://www.javatpoint.com/sql-data-types>

## Different Types of Data Types in MySQL 2-2

- String Data Type:** Stores data as long text or strings
- Binary Data Type:** Stores binary large objects that hold a variable amount of data
- Spatial Data Type:** Stores geometrical and geographical values
- JSON Type:** Provides automatic validation of JSON documents

© Aptech Limited Practical MySQL / Session 2 / 10 of 15

### Instruction(s) to the trainer:

Show slide 10 and explain that string data type can store either fixed or varied number of characters. Entity names such as Customer names, product names, cities, order status, and many other such textual data can be stored as strings.

Binary data type is used to maintain binary large objects that hold a variable amount of data. These differ only in terms of the maximum length of the values.

Spatial data is used to determine specific locations based on the coordinate data. Such data is typically stored in the spatial data types. Geometric data is highly used in Google Maps to display precise directions. Various important shapes such as rectangular tiles, pyramids, football pitches, and road designs require specific multi-polygonic data types to store values.

JSON data type provides automatic validation of JSON documents. It generates an error for invalid documents.

**In-Class Question:** Which data type is used to store geometrical and geographical values?

**Answer:** Spatial data type

**Additional Information:**

Refer to following links for more information:

[https://www.w3schools.com/sql/sql\\_datatypes.asp](https://www.w3schools.com/sql/sql_datatypes.asp)  
<https://www.tutorialspoint.com/sql/sql-data-types.htm>  
<https://www.javatpoint.com/sql-data-types>

For Aptech Centre Use Only

## Creating Tables 1-2

A table is created by entering the CREATE TABLE command in the MySQL command prompt.

With Primary Key

With Foreign Key

### Instruction(s) to the trainer:

Show slide 11 and explain how tables can be created with primary key and foreign key respectively.

Primary keys can be used to relate one table to another. Adding primary key/s while creating a table maintains consistency, uniqueness, and avoids duplicate data. It must be noted that only one primary key is allowed in a table.

To establish a relationship with a child table or any other table, a foreign key must be added along with the column containing only the values as those of the parent table's primary key values.

A foreign key is also known as referencing key and is used to link multiple tables. The foreign key field in a table refers to the primary key field of another table. The foreign keys can be defined in two ways using CREATE TABLE and ALTER TABLE statement.

### Additional Information:

Refer to following links for more information:

[https://www.w3schools.com/sql/sql\\_create\\_table.asp](https://www.w3schools.com/sql/sql_create_table.asp)

<https://www.javatpoint.com/sql-tutorial>

The slide has a dark blue header bar with the title 'Creating Tables 2-2'. Below the header is a large blue rectangular area containing the text 'Tables can also be created'. This area is divided into two sections: 'With Check Constraints' on the left and 'With Select Command' on the right. At the bottom of the slide, there is a thin blue footer bar with the copyright notice '© Aptech Limited' on the left and 'Practical MySQL / Session 2 / 12 of 15' on the right.

## Creating Tables 2-2

Tables can also be created

With Check Constraints      With Select Command

© Aptech Limited      Practical MySQL / Session 2 / 12 of 15

**Instruction(s) to the trainer:**

Using slide 12, inform students on how to create a table with the CHECK constraint as well as the SELECT command. The CHECK constraint is used to check the value before inserting in the column. Query that contains CHECK constraint will return error if the value in a particular column does not match the conditions.

A new table can be created from an existing table as well using the SELECT command. Syntax is as follows:

```
CREATE TABLE <TABLE_NAME> AS SELECT COLUMN_NAME,.... FROM  
<TABLE_NAME>
```

**Additional Information:**

Refer to following links for more information:

[https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp)  
<https://www.javatpoint.com/sql-tutorial>  
<https://www.tutorialspoint.com/sql/sql-create-table.htm>

## Working with Tables 1-2

Some of the common operations performed on the tables in a database

Altering a Table

Renaming a table

### Instruction(s) to the trainer:

Show slide 13 and tell students that the ALTER command is used to add and remove a column in an existing table. The data type of the column can also be modified using the ALTER command.

The user can change the size or field of any column, depending on the requirement. ALTER COLUMN keyword is used to increase or decrease the size of the column in MySQL.

A table in MySQL can be renamed by using following command:

```
RENAME TABLE Customer TO New_Customer;  
// Following command will also rename the table  
ALTER TABLE Customer RENAME TO New_Customer;
```

### Additional Information:

Refer to following links for more information:

[https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp)  
<https://www.javatpoint.com/sql-tutorial>

## Working with Tables 2-2

Other operations performed on the tables in a database

Adding Index to a Table	Truncating a Table	Dropping a Table
-------------------------	--------------------	------------------

© Aptech Limited Practical MySQL / Session 2 / 14 of 15

**Instruction(s) to the trainer:**

Use slide 14 to tell students about some other operations that can be performed on a table, such as adding and index, truncating or dropping a table.

Following is the syntax for ALTER TABLE ADD INDEX:

```
ALTER TABLE ADD INDEX `index_name` (`column_name`)
```

The DROP TABLE statement in MySQL is used to drop a table(s) from the database. It permanently erases all data from the table and removes the structure as well. The TRUNCATE command is used to remove all records from an existing table in MySQL. It performs the same function as a DELETE statement without using a WHERE clause. The structure of table remains intact even after executing the TRUNCATE command.

**In-Class Question:** What is the difference between the DELETE command and the TRUNCATE command?

**Answer:** DELETE is used to remove some or all of the records from the table, whereas TRUNCATE will delete entire rows from table.

**Additional Information:**

Refer to following links for more information:

[https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp)

<https://www.javatpoint.com/sql-tutorial>

## Summary

- A database is a collection of data and information in a standardized format and can be accessed by multiple users at the same time.
- Mysqldadmin is a tool used to perform administrative operations.
- DDL commands are used with database and database objects such as tables, whereas, DML commands are used to manipulate data in database objects.
- Adding primary key/s while creating a table maintains consistency, uniqueness, and avoids data duplication.
- A table can be linked with another table by dropping the primary or foreign key.
- The ALTER COLUMN command is used to change the data type of a column in a table.
- The TRUNCATE TABLE statement is used to remove all records from the existing table in MySQL.

### Instruction(s) to the trainer:

Use slide 15 to summarize the session. You will end the session with a summary of what has been taught in the session. Tell students the pointers of the session. This will be a revision of the current session.

## Session 3: Working with Data in MySQL

### Slide 2

#### Session Overview

- Identify the use of `INSERT`, `UPDATE`, and `DELETE` in the manipulation of data.
- Explain the `SELECT` command in detail.
- Explain the use of `LIKE`, `IN`, `BETWEEN`, and `DISTINCT` clauses.
- Explain `MERGE` command and its usage.

#### Instruction(s) to the trainer:

Show slide 2 and give the students a quick overview of the current session and its goals. Inform them that the session will explain the usage of `SELECT`, `INSERT`, `UPDATE`, and `DELETE` commands. They will also learn about some of the keywords and clauses. Finally, the `MERGE` command and its purpose will be discussed in detail.

## Slide 3

# SELECT Commands in MySQL

A SELECT query is used to display results in a structured way with proper column names and organized records.

The SELECT query displays results based on one or more conditions mentioned by the user.

Syntax: `SELECT * FROM table_name;`  
`SELECT column1, column2, column3... columnN FROM table_name;`

© Aptech Limited

Practical MySQL / Session 3 / 3 of 11

### Instruction(s) to the trainer:

Show slide 3 and explain the concept of SELECT commands in MySQL. Tell the students that the SELECT statement is used to pick data from a database. The data is saved in a result table known as the result-set.

Furthermore, describe the following syntax:

```
SELECT column1, column2, ... FROM table_name;
```

where, column1, column2 are the field names of the table from which a user wants to retrieve data.

Inform students that the following syntax is used to pick all of the fields accessible in the table:

```
SELECT * FROM table_name;
```

**In-Class Question :** What does select 0 mean in SQL?

**Answer:** A field name of 0 in the select statement means: **This field is not in the specified table**, so if you want to use the specified table in the subquery with this field, you add the field in this way and initialize the field with 0.

## Slide 4

### Simple SELECT Statements

Information of all the employees can be displayed using the `SELECT` query.

For example, a `SELECT` query can be used in Railways Database to display all information about trains, train numbers, timings and so on.

Following command is used to fetch all the records from the `emp_details` table:

```
SELECT * FROM emp_details;
```

© Aptech Limited Practical MySQL / Session 3 / 4 of 11

#### Instruction(s) to the trainer:

Show slide 4 and provide a brief overview on simple SELECT Statements.

Give the example of creating a table to explain the concept in a better way. By using queries:

```
CREATE TABLE emp_details (
EmployeeID INT,
Name VARCHAR(255),
Location VARCHAR(255),
Salary INT,
Dept VARCHAR(255),
PRIMARY KEY (EmployeeID)
);
```

Explain the following query to insert data into the `emp_details` table:

```
INSERT INTO `emp_details`(`EmployeeID`, `Name`, `Location`,
`Salary`, `Dept`) VALUES
(1001,'Daniel','Princeton', 10000,'Finance');
INSERT INTO `emp_details`(`EmployeeID`, `Name`, `Location`,
`Salary`, `Dept`) VALUES (1002,'Anna','Ohio', 15000,'Sales');
```

```
INSERT INTO `emp_details`(`EmployeeID`, `Name`, `Location`,  
`Salary`, `Dept`) VALUES (1003,'Emma','New Jersey',  
12500,'Marketing');  
INSERT INTO `emp_details`(`EmployeeID`, `Name`, `Location`,  
`Salary`, `Dept`) VALUES (1004,'Charles','Mexico',  
20000,'Finance');  
INSERT INTO `emp_details`(`EmployeeID`, `Name`, `Location`,  
`Salary`, `Dept`) VALUES (1005,'Kipp','Florida', 25000,'Legal');
```

Tell students that the information of all employees can be displayed using the **SELECT** query. For example, a **SELECT** query can be used in company database to display all information about employees, employee id, name, and so on.

Following command is used to fetch all the records from the **emp\_details** table:

*Command:* **SELECT \* FROM emp\_details;**

*Output:* The user will get all the details in a tabular form.

Discuss with students that only the distinctive (different) results are returned using the **SELECT DISTINCT** command.

A column in a table frequently has multiple duplicate values and a user may only wish to list the distinctive values.

Explain the following syntax:

```
SELECT DISTINCT column1, column2, ... FROM table_name;
```

## Slide 5

### WHERE Clause with Operators

#### MySQL WHERE Clause with a single condition:

- WHERE clause with a single condition returns records based on that particular condition.

#### MySQL WHERE Clause with AND operator:

- WHERE clause with AND operator includes two conditions and returns results fulfilling both the conditions.

#### MySQL WHERE Clause with OR operator:

- WHERE clause with OR operator includes two conditions and returns data fulfilling either of the two conditions.

#### MySQL WHERE Clause with a combination of AND and OR conditions:

- Using AND and OR clauses with a WHERE clause filters the results even more.

#### Instruction(s) to the trainer:

Show slide 5 and describe the WHERE Clause used in MySQL. The WHERE clause is used to filter records. It is also used to retrieve only those records that meet a specific set of criteria.

#### Explain the following Syntax:

```
SELECT column1, column2, ... FROM table_name WHERE condition;
```

**Note:** Inform students that the WHERE clause is used not just in SELECT statements, but also in UPDATE, DELETE, and other statements.

Discuss the following pointers related to the MySQL AND, OR and NOT Operators:

- To filter records based on several conditions, the AND and OR operators are used.
- If all of the conditions separated by AND are TRUE, then the AND operator displays a record.
- If any of the conditions separated by OR is TRUE, then the OR operator displays a record.
- If the condition(s) are NOT TRUE, then the NOT operator displays a record.

#### Explain the following syntax to students:

- **AND:**

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

- **OR:**

```
SELECT column1, column2, ...
  FROM table_name
  WHERE condition1 OR condition2 OR condition3 ...;
```
- **NOT:**

```
SELECT column1, column2, ...
  FROM table_name
  WHERE NOT condition;
```

**Additional Information:**

Refer to following links for more information:

[https://www.w3schools.com/sql/sql\\_where.asp](https://www.w3schools.com/sql/sql_where.asp)  
<https://www.guru99.com/where-clause.html>

## Slide 6

### Keywords in MySQL

LIKE

- LIKE keyword is used to fetch records that match a given beginning character, ending character, or in between characters of a column.
- Following command displays information of employees whose name starts with the letter 'A':  
`SELECT * FROM emp_details WHERE Name LIKE 'A%'`

IN

- IN keyword is used to retrieve records by checking the possible values from a given column. It is used with a single column.
- Following command displays information of employees who belong to the department of Finance, Sales, or Legal:  
`SELECT * FROM emp_details WHERE dept IN ('Finance', 'Sales', 'Legal');`

BETWEEN

- The BETWEEN operator chooses values from a specified range.  
`SELECT column_name(s) FROM table_name  
WHERE column_name BETWEEN value1 AND value2;`

#### Instruction(s) to the trainer:

Show slide 6 and discuss about the keywords in MySQL,

Tell the students that in a WHERE clause,

. LIKE keyword is used to look for a certain pattern in a column. There are two wildcards that are frequently used with the LIKE operator:

- . The percent sign (%) denotes zero, one, or several characters.
- . The underscore symbol (\_) denotes a single character.
- . The underscore and the percent sign can also be used together.

Further, define the appropriate syntax: `SELECT column1, column2, ...`

```
FROM table_name  
WHERE columnN LIKE pattern;
```

. IN: In a WHERE clause, the IN operator allows a user to define several values. Multiple OR conditions are represented by the IN operator.

Describe the syntax:

```
SELECT column_name(s)  
      FROM table_name  
      WHERE column_name IN (value1, value2, ...);
```

**BETWEEN:** The BETWEEN operator chooses values from a specified range. Numbers, text, and dates can all be used as values. The BETWEEN operator takes both the start and finish values into account.

Define the syntax to students:

```
SELECT column_name(s)  
      FROM table_name  
 WHERE column_name BETWEEN value1 AND value2;
```

**In-Class Question:** Is rank a keyword in MySQL?

**Answer:** The rank is a MySQL reserved word defined in MySQL version 8.0.

## Slide 7

### DISTINCT Clause

MySQL also provides basic clauses that can be directly used with SELECT, such as DISTINCT that fetches unique values from a table.

#### MySQL Distinct Clause with Single Expression:

- A DISTINCT clause with a single expression fetches unique values from one column.
- The query will display the names of the departments existing in the organization only once.
- `SELECT DISTINCT Dept FROM emp_details;`

#### MySQL Distinct Clause with Multiple Expressions:

- A DISTINCT clause with multiple expressions means that it fetches unique values from more than one column.
- Following query returns only the distinct values of the columns combined: `SELECT DISTINCT Name, Location FROM emp_details;`

#### Instruction(s) to the trainer:

Show slide 7 and explain about the DISTINCT clause. Only distinctive (different) results are returned using the SELECT DISTINCT command.

A column in a table frequently has multiple duplicate values. Thus, a user might only want to list the distinctive values.

Explain the following syntax to students:

```
SELECT DISTINCT column1, column2, ... FROM table_name;
```

#### Additional Information:

Refer to following links for more information:

<https://www.techonthenet.com/mysql/distinct.php#:~:text=Description,be%20used%20with%20SELECT%20statements>.

<https://www.javatpoint.com/mysql-distinct>

## INSERT Commands in MySQL

The `INSERT` query is used to add new records to the table.

`TIMESTAMP` is a data type that contains the current date and time stored in the system.

```
SELECT TIMESTAMP("2017-07-23", "13:10:11");
```

Following is the syntax of the `INSERT` query:

```
INSERT INTO table_name (column1, column2, column3,  
...) VALUES (value1, value2, value3, ...);  
OR  
INSERT INTO table_name VALUES  
(value1, value2, value3, ...);
```

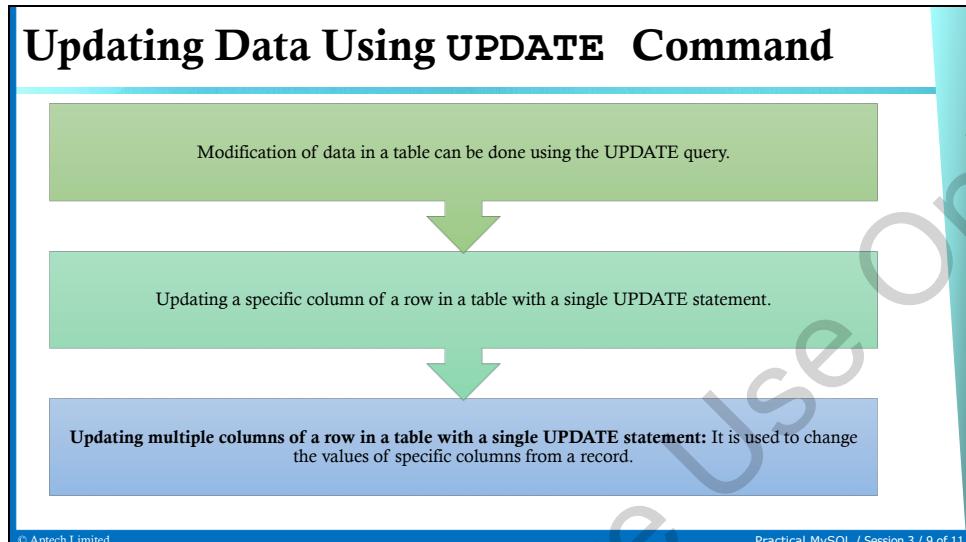
### Instruction(s) to the trainer:

Show slide 8 and give a brief outline on the `INSERT` Commands in MySQL. Tell students that the `INSERT INTO` statement is used to add new records to a table.

Inform them that there are two methods to write the `INSERT INTO` statement.  
Both the column names and the values to be added must be specified.

*Syntax:* `INSERT INTO table_name (column1, column2, column3, ...)`  
`VALUES (value1, value2, value3, ...);`

2. A user does not have to include the column names in the SQL query if the values are being added to all of the table's columns. A programmer must ensure that the values are in the same order as the table's columns. The `INSERT INTO` syntax in this case would be as follows:  
`INSERT INTO table_name VALUES (value1, value2, value3, ...)`



**Instruction(s) to the trainer:**

Show slide 9 and discuss how to update data in MySQL using UPDATE command, The UPDATE statement is used to change the contents of a table's existing records.

Following is the command to update a single column (Location) of the employees belonging to the Finance department:

```
UPDATE emp_details SET Location= 'Delaware' WHERE Dept = 'Finance';
```

Following is the command to update the salary and location of the employee having id 1002:

```
UPDATE emp_details SET Salary = 26000, Location= 'New York'  
WHERE EmployeeId = 1002;
```

Introduce students to its syntax:

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

**Note:** A user must stay cautious while updating records in a table! In the UPDATE statement, one must take note of the WHERE clause. The WHERE clause determines which records should be updated. If the programmer does not utilize the WHERE clause, all of the table's records will be updated.

## DELETE Commands in MySQL

With the `DELETE` query, the user can delete specific records or entire records of the table using the `WHERE` clause.

### Deleting all records from a table in a single `DELETE` command

- Following is the syntax to delete all records of a table at once:

```
DELETE FROM table_name;
```

### Deleting specific rows from a table

- Following is a command to delete details of all employees belonging to the 'Legal' department' in the `emp_details` table:

```
DELETE FROM emp_details  
WHERE Dept = 'Legal';
```

### Instruction(s) to the trainer:

Show slide 10 and discuss how to `DELETE` data in MySQL using the `DELETE` command. The `DELETE` statement is used to erase existing records in a table.

Explain the following syntax:

```
DELETE FROM table_name WHERE condition;
```

**Note:** One must stay alert while deleting records from a table. In the `DELETE` statement, a user must take note of the `WHERE` clause. The `WHERE` clause defines the records to be removed from the database. If a developer does not include the `WHERE` clause, the entire table will be destroyed.

To delete all the records, a user can delete all rows in a table without removing the table. Tell students that the table structure, characteristics, and indexes will all be preserved.

Describe the following syntax:

```
DELETE FROM table_name;
```

## Summary

- A query is an SQL statement to communicate with the database for some data.
- MySQL supports many commands including SELECT, INSERT, UPDATE, and DELETE.
- SELECT command is the most popular command used to display records of a table.
- INSERT command helps to add new records to a table in the database.
- Using UPDATE, users can modify existing records in a table.
- Multiple rows and columns can be updated using the UPDATE command.
- DELETE command can delete a single record, multiple records, or all records at once.

**Instruction(s) to the trainer:**

Use slide 11 to summarize the session. End the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

## Session 4: Joins in MySQL

### Slide 2

#### Session Overview

- Define JOINS
- Explain the use of JOINS
- Identify different types of JOINS
- Explain the use of SET operators
- Outline the differences between JOINS and UNIONS

#### Instruction(s) to the trainer:

Show slide 2 and give students a brief overview of the current session in the form of session objectives. Inform students that the session discusses JOINS and SET operator in MySQL. The session begins with explaining the use of JOINS in MySQL with the help of various examples. Next, it lists different types of JOINS. Further, the session explains the use of SET operators. At the end, the session elaborate the differences between JOINS and UNIONS.

## Slide 3

### Introduction to JOINS

MySQL JOIN is a feature used to combine and retrieve data from more than one table.

The JOIN clause retrieves the data by joining two tables and creating a relationship between them.

A JOIN operation returns the values (rows) whose data matches both tables in that column(s).

### In-Class Explanations

#### Instruction(s) to the trainer:

Show slide 3 and inform students about the JOIN clause in MySQL. Explain to them that JOIN can be used to retrieve data from tables. It works by combining the data of multiple tables by forming a relationship between them.

Tell students that to execute a JOIN clause, tables must have related columns. Join clauses reduce the number of queries, and thus also reduce the load on the server caused by the constant exchanging of data between the server and database.

#### Additional Information:

Refer to following links for more information:

<https://www.sqlshack.com/sql-join-clause-introduction-and-overview/>

<https://www.mysqltutorial.org/mysql-join/?msclkid=57e7cd7eb0eb11eca0e7b2aa183b980a>

## Slide 4

### How to Use Joins?

CustomerID	FirstName	LastName	Country	Age	Gender
214431	Jack	Swanson	Germany	34	M
956437	Joe	Voight	France	22	F
111278	Christian	Gale	Italy	19	M
897867	Karl	Davis	Switzerland	22	M
789023	Berry	Chase	Netherlands	24	F

**Customer Table**

OrderID	CustomerID	Status	Amount
567834	214431	Paid	10000
113425	956437	Unpaid	6780
564721	564721	Unpaid	12000
999982	789023	Paid	4500
453110	897867	Paid	2300

**Orders Table**

© Aptech Limited Practical MySQL / Session 4 / 4 of 14

#### Instruction(s) to the trainer:

Show slide 4 and explain the use of the JOIN clause by citing an example. Consider two tables Customer table and Orders table. The user must create following tables and insert the data into the tables.

Tell students that here the common column is CustomerID, which is also a primary key in the Customer table. Also tell students that CustomerID is a foreign key in the Orders table.

#### Additional Information:

Refer to following links for more information:

[https://www.w3schools.com/mysql/mysql\\_join.asp?msclkid=57e5fa47b0eb11ecb5259393749bc942](https://www.w3schools.com/mysql/mysql_join.asp?msclkid=57e5fa47b0eb11ecb5259393749bc942)

<https://www.mysqltutorial.org/mysql-join/?msclkid=57e7cd7eb0eb11eca0e7b2aa183b980a>

## Slide 5

# JOIN Clause

Example for JOIN Clause

```
SELECT *  
FROM Customer JOIN Orders  
ON Customer.CustomerID = Orders.CustomerID;
```

Output

```
mysql> SELECT *  
-> FROM Customer JOIN Orders  
-> ON Customer.CustomerID = Orders.CustomerID;  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| CustomerID | FirstName | LastName | Country | Age | Gender | OrderID | CustomerID | Status | amount |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| 214431 | Jack | Swanson | Germany | 34 | M | 567834 | 214431 | Paid | 10000 |  
| 956437 | Joe | Voight | France | 22 | F | 113425 | 956437 | Unpaid | 6780 |  
| 789023 | Berry | Chase | Netherlands | 24 | F | 99982 | 789023 | Paid | 4500 |  
| 897867 | Karl | Davis | Switzerland | 22 | M | 453110 | 897867 | Paid | 2300 |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.15 sec)
```

© Aptech Limited Practical MySQL / Session 4 / 5 of 14

### Instruction(s) to the trainer:

Show slide 5 and explain how the JOIN clause works with the example. You can see that the output is displaying the records of both the tables. This means that the JOIN clause has joined the records of both the tables. Also, point out that since there are no order details for CustomerID 111278 in the Orders table, the customer's details for that id are not displayed in the JOIN result.

**In-Class Question:** Give two uses of the JOIN clause.

- Answer:**
1. It is used to filter data from relational tables.
  2. It reduces duplicate records in the final output.

### Additional Information:

Refer to following links for more information:

[https://www.w3schools.com/mysql/mysql\\_join.asp?msclkid=57e5fa47b0eb11ecb5259393749bc942](https://www.w3schools.com/mysql/mysql_join.asp?msclkid=57e5fa47b0eb11ecb5259393749bc942)

<https://www.mysqltutorial.org/mysql-join/?msclkid=57e7cd7eb0eb11eca0e7b2aa183b980a>  
<https://www.guru99.com/joins.html?msclkid=57e80d51b0eb11ecbb3a719455fcf948>

## Why to Use JOINS?

JOINS are used to:

- Filter data from relational tables
- Combine data from two or more tables and filter it based on the conditions specified by the user
- Reduces duplicate records in the combinational result

### Instruction(s) to the trainer:

Show slide 6 and explain the use of JOIN clause. The JOIN clause not only combines the data of two tables but also filters data from relational tables on the basis of the given query.

Thus, they help in retrieving data from two or more related database tables in a single query.

The JOIN clause reduces duplicate records in the final output. The reduced overhead makes query execution faster.

### Additional Information:

Refer to following links for more information:

[https://www.w3schools.com/mysql/mysql\\_join.asp?msclkid=57e5fa47b0eb11ecb5259393749bc942](https://www.w3schools.com/mysql/mysql_join.asp?msclkid=57e5fa47b0eb11ecb5259393749bc942)

<https://www.mysqltutorial.org/mysql-join/?msclkid=57e7cd7eb0eb11eca0e7b2aa183b980a>

## JOINS with Subquery

A subquery can also be used with JOIN operation to increase its usage. Following query will display **Name**, **Age**, and **OrderID** in a Derived Table:

```
SELECT * FROM
(SELECT FirstName, LastName, Age, OrderID
FROM Customer JOIN Orders
ON Customer.CustomerID = Orders.CustomerID) AS Order_Details
mysql> SELECT * FROM
-> (SELECT FirstName, LastName, Age, OrderID
-> FROM Customer JOIN Orders
-> ON Customer.CustomerID = Orders.CustomerID)
-> AS Order_Details
-> ;
+-----+-----+-----+
| FirstName | LastName | Age | OrderID |
+-----+-----+-----+
| Jack      | Swanson | 34  | 567834 |
| Joe       | Voight  | 22  | 113425 |
| Berry     | Chase   | 24  | 999982 |
| Karl     | Davis   | 22  | 453110 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

© Aptech Limited

Practical MySQL / Session 4 / 7 of 14

### Instruction(s) to the trainer:

Show slide 7 and tell students that the functionality of the JOIN clause can be increased by using subqueries with it. Inform them that when a JOIN clause is used with a subquery, the resulting output is known as a Derived table.

The output displays the columns **FirstName**, **LastName**, **Age**, and **OrderID** in a Derived table.

**In-Class Question:** Which clause should be used while creating a subquery with JOINS in MySQL?

**Answer:** When subqueries are used along with JOINS, the subquery is written using the FROM clause.

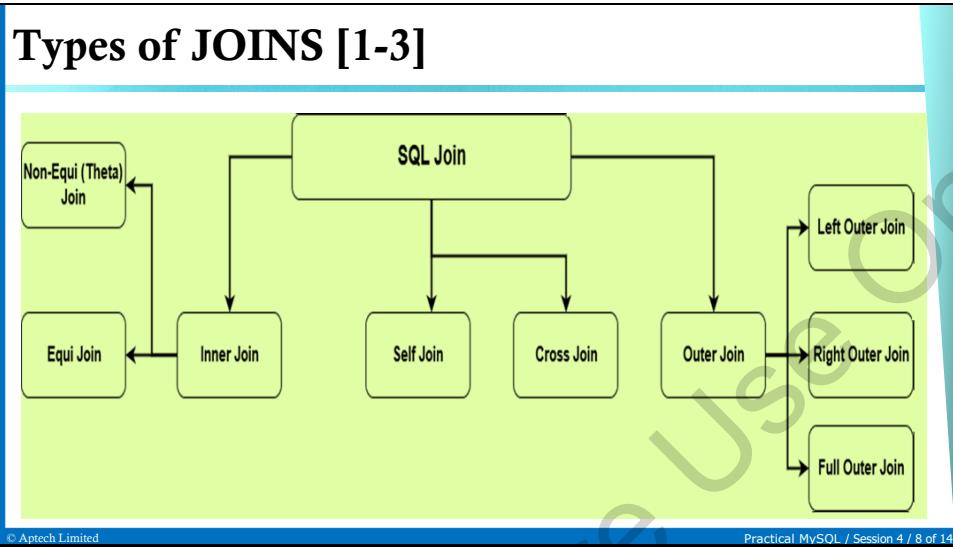
### Additional Information:

Refer to following links for more information:

[https://www.w3schools.com/mysql/mysql\\_join.asp?msclkid=57e5fa47b0eb11ecb5259393749bc942](https://www.w3schools.com/mysql/mysql_join.asp?msclkid=57e5fa47b0eb11ecb5259393749bc942)

<https://stackoverflow.com/questions/26486512>

## Slide 8



#### Instruction(s) to the trainer:

Show slide 8 and explain different types of JOINS. MySQL supports different types of JOINS such as INNER JOIN, OUTER JOIN, SELF JOIN, and CROSS JOIN. Explain how they are connected to each other. Explain students that a JOIN clause links two tables based on a matching column and returns the desired result.

As seen on slide, INNER JOIN, OUTER JOIN, and SELF JOIN are three main types of SQL JOINS. CROSS JOIN can be used to display records of two tables that have no matching rows.

#### Additional Information:

Refer to following links for more information:

[www.w3schools.com/sql/sql\\_join.asp](http://www.w3schools.com/sql/sql_join.asp)  
<https://www.javatpoint.com/types-of-sql-join>  
<https://www.educba.com/types-of-joins-in-sql>

## Types of JOINS [2-3]

INNER JOIN	It displays rows of both tables. It is mandatory to have a common column between the two tables for the results to be non-empty.
LEFT JOIN	Also known as Left Outer Join, it returns all rows of the left table and the matching rows of the right table.
RIGHT JOIN	Also known as Right Outer Join, it returns all rows of the right table and the matching rows of the left table.
CROSS JOIN	It returns a combined result of rows from the first table with that on the second table, irrespective of any matching rows
SELF JOIN	It is a join of a table with itself. It is used to find query results within itself.

### Instruction(s) to the trainer:

Using slides 9 and 10, explain the functionalities of different types of JOINS.

- INNER JOIN is the default JOIN clause that is used to display the rows of both tables.
- LEFT JOIN is used in cases where the records of the table mentioned on the left are to be displayed with the matching records of the table on the right. RIGHT JOIN is basically the inverse of LEFT JOIN.
- CROSS JOIN is used in cases where you need to display records of two tables that have no matching rows. CROSS JOIN returns the records in the form of a cartesian product of two tables.
- SELF JOIN is commonly used with tables having a column with data referring to one of the column of the table itself.

### Additional Information:

Refer to following links for more information:

[www.w3schools.com/sql/sql\\_join.asp](http://www.w3schools.com/sql/sql_join.asp)

<https://www.javatpoint.com/types-of-sql-join>

<https://www.educba.com/types-of-joins-in-sql>

## Types of JOINS [3-3]

DELETE JOIN	It is used when the user wants to delete rows from more than one table.
UPDATE JOIN	It modifies/sets values of rows of more than one table using join.
Equal JOIN	It is performed on tables with matching row values only. It requires an equality operator that checks if the row values of one table match with row values of the other table.
NATURAL JOIN	It returns a Cartesian product of the two tables. It is used to remove the duplicate values and display a single column.

### Instruction(s) to the trainer:

- DELETE JOIN is used to delete rows from more than one table. The WHERE condition can be used to limit the number of rows to be deleted.
- The JOIN statement returns a result containing rows of both tables and the UPDATE clause sets the row values accordingly.
- Equal JOIN requires an equality operator that checks if the row values of one table match with row values of the other table.
- NATURAL JOIN clause returns the result as a Cartesian product of two tables. It also removes duplicate values from the table.

**In-Class Question:** Explain the function of Equal JOIN in MySQL.

**Answer:** Equal JOIN in MySQL is used to check if row values of one table are matching with row values of the second table.

### Additional Information:

Refer to following links for more information:

[www.w3schools.com/sql/sql\\_join.asp](http://www.w3schools.com/sql/sql_join.asp)

<https://www.javatpoint.com/types-of-sql-join>

<https://www.educba.com/types-of-joins-in-sql>

## Slide 11 and 12

### Joining Three Tables [1-2]

StudentID	Student_Name
1001	Mary
1002	Jane

CourseID	Course_Name	StudentID
5001	PHP	1004
5002	JAVA	1001

TeacherID	Teacher_Name	CourseID
3001	Jill	5003
3002	Greul	5002

Syntax to join the three tables using INNER JOIN:

```
SELECT table1.column1_name, table1.column2_name, ...,
       table2.column1_name, table2.column2_name, ...,
       table3.column1_name, table3.column2_name, ...,
  FROM table1
  INNER JOIN table2 ON table1.table1_id = table2.table1_id
  INNER JOIN table3 ON table2.table2_id = table3.table2_id;
```

### Joining Three Tables [2-2]

Following is the example to display all the rows of all three tables using INNER JOIN:

```
SELECT Student.Student_Name, Course.Course_Name,
       Teacher.Teacher_Name
  FROM Student
  INNER JOIN Course ON Student.StudentID = Course.StudentID
  INNER JOIN Teacher ON Course
```

```
mysql> SELECT Student.Student_Name, Course.Course_Name,
   -> Teacher.Teacher_Name
   -> FROM Student
   -> INNER JOIN Course ON Student.StudentID = Course.StudentID
   -> INNER JOIN Teacher ON Course.CourseID = Teacher.CourseID;
+-----+-----+-----+
| Student_Name | Course_Name | Teacher_Name |
+-----+-----+-----+
| Mary         | JAVA        | Greul      |
| Mary         | ASP.NET     | Jill       |
| Jane         | RUBY        | Polo       |
| Jack         | PHP         | Jenkins   |
+-----+-----+-----+
4 rows in set (0.08 sec)
```

#### Instruction(s) to the trainer:

Using slides 11 and 12, explain students how to join three tables using the JOIN clause. Consider an example of three tables, Student, Course, and Teacher. Tell students that in the Course table, StudentID is the foreign key, which is the primary key in the Student table.

CourseID acts as the foreign key in the Teacher table and the primary key in the Course table.

Using Slide 12 explain the example to display all the rows of all three tables - Student, Course, and Teacher using INNER JOIN.

**Additional Information:**

Refer to following links for more information:

<http://codingstatus.com/sql-join-3-tables/>

<https://www.geeksforgeeks.org/join-multiple-tables-using-inner-join>

<https://stackoverflow.com/questions/10195451>

## Set Operators

Set operators join more than two statements and return results.

### Union

- Combines the result of SELECT statements used in the query without returning duplicate values

### Union All

- Executes the queries and returns all the rows. It allows duplicate values

### Intersect

- Combines results of two or more SELECT statements and returns only those rows from the first table which matches the rows of the second table

### Minus

- Returns the rows that are not present in the second statement

### Instruction(s) to the trainer:

Using slide 13, explain different types of Set operators.

- The Union operator is used to combine distinct results of two or more SELECT statements. UNION is useful when normalized tables are combined to find out results. The WHERE clause is used with UNION to filter specific rows in a query.
- The Union All operator is used to combine all the results of two or more SELECT statements, including duplicates. UNION ALL executes the queries and returns all the rows (allows duplicate values).
- The Intersect operator returns only the common records obtained from two or more SELECT statements.
- The Except operator is also known as the Minus operator, It returns only those records which are exclusive to the first table.

**In-Class Question:** Why is the WHERE clause used with UNION?

**Answer:** The WHERE clause is used with UNION to filter specific rows in a query.

### Additional Information:

Refer to following links for more information:

[https://www.w3schools.com/sql/sql\\_union.asp](https://www.w3schools.com/sql/sql_union.asp)

<https://www.databasestar.com/sql-set-operators/>

<https://www.tutorialsteacher.com/sql/sql-union-operator>

<https://www.techonthenet.com/sql/union.ph>

## Summary

- A JOIN clause is used to retrieve combined data sets from more than one table.
- JOINS are more efficient than complex queries and subqueries.
- INNER JOIN and OUTER JOIN are the most used JOINS in MySQL.
- Other types of JOINS include CROSS JOIN, EQUIJOIN, and NATURAL JOIN.
- DELETE JOIN and UPDATE JOIN are used for deleting and updating records from multiple tables respectively.
- SET Operators are used with queries to compare records and form a single set of records.
- The most common types of SET Operators are Union SET Operator, Union All SET Operator, Intersect SET Operator, and Minus SET Operator.
- UNION clause adds two tables and returns the result.
- UNION ALL includes duplicate records whereas, UNION filters out distinct records from the tables.
- INTERSECT provides a result with all distinct records from both queries.
- When EXCEPT is used in queries, it displays the results of the first of the first query after removing the results of the second query.

### Instruction(s) to the trainer:

Use slide 14 to summarize the session. You will end the session with a summary of what has been taught in the session. Tell students the pointers of the session. This will be a revision of the current session.

## Session 5: Subqueries

### Slide 2

#### Session Overview

- Explain sub queries and their usage
- Outline the clauses and keywords used in subqueries
- Describe practical uses of WHERE and FROM clauses in subqueries
- Define IN, NOT IN, EXISTS, and NOT EXISTS keywords
- List and explain different types of subqueries

#### Instruction(s) to the trainer:

Show slide 2 and give students a brief overview of the current session and the session objectives. Inform students that this session explains the concept subqueries, clauses, and keywords in subqueries. It also describes the practical approach of using different clauses like WHERE and FROM in sub-queries. Finally, the session explains the various keywords and types of subqueries present in MySQL.

## Subquery

A subquery is a nested query inside another query. It is also called an Inner Query or Nested Query.

It is also referred to as nested SELECT or Sub-SELECT.

Following is the basic syntax of a subquery:

```
SELECT column_name FROM table_1 WHERE column_name  
expression operator (SELECT COLUMN_NAME FROM  
TABLE_2 WHERE...); //Subquery
```

### Instruction(s) to the trainer:

Show slide 3 and introduce students to the concept of Subquery in MySQL. Explain that a MySQL subquery is a query nested inside another query like SELECT, INSERT, UPDATE, or DELETE. A subquery can also be nested inside of another subquery.

An inner query in MySQL is a subquery, whereas an outer query contains the subquery. A subquery must be enclosed in parentheses and can be used anytime that expression is used.

Elaborate on the concept through an example. The following query uses a subquery to return employees who work in offices in the United States.

```
SELECT lastName, firstName FROM employees WHERE officeCode IN  
(SELECT officeCode FROM offices WHERE country = 'USA');
```

Discuss the following pointers related to this example:

.The subquery retrieves all the office codes for offices in the USA.

.The outer query chooses the last name and first name of employees who work in offices with office codes that appear in the subquery's result set.

**In-Class Question:** Why do we use subquery in SQL?

**Answer:** A Subquery or Inner query or a Nested query is a query within another SQL query and embedded within the WHERE clause. A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

## Slide 4

### When to use Subquery?

Subqueries are used when some result must be fetched from the database that requires multiple query statements. In multiple queries, each query returns a subset of data from the table.

Following are guidelines to use subqueries:

Subquery is used for membership questions.

Subqueries are used when the user requires more than one query.

Multiple queries can be formulated using subqueries

#### Instruction(s) to the trainer:

Show slide 4 and give students information about when to use a Subquery. An inner query is a subquery, while the outer query is the query that contains the subquery. The result of the inner query is passed to the outer query first, and then the main/outer query is executed. Subqueries can be used anywhere in MySQL. However, they must be enclosed in parentheses. MySQL will handle all subquery forms and actions that are supported by the SQL standard.

Describe following rules for using sub-queries with the students:

Subqueries must be enclosed in parenthesis.

If the primary query does not have multiple columns for subqueries, then the SELECT command for a subquery can only have one column.

With a subquery, we can utilise comparison operators like >, =, IN, ANY, SOME, and ALL.

When a subquery returns more than one row, a multiple-row operator comes in handy.

The ORDER BY clause cannot be used in a subquery, although it can be used within the main query.

A subquery cannot be immediately included in a set function.

#### Additional Links:

Refer to following links for more information:

<https://www.mysqltutorial.org/mysql-subquery/#:~:text=A%20MySQL%20subquery%20is%20called,offices%20located%20in%20the%20USA>.

<https://www.w3resource.com/mysql/subqueries/index.php>

## Clauses and Keywords in Subquery

There are several clauses and keywords, which are used with subqueries to solve complex queries.

WHERE and FROM form the basis of clauses used whereas, IN, NOT IN, EXISTS, and NOT EXISTS are the most used keywords.

For a better illustration of clauses and keywords in a subquery, consider two tables: **Department** and **Employee** tables.

The **Department** table must be created and data must be added before creating and adding data to the **Employee** table to avoid a foreign key constraint error.

### Instruction(s) to the trainer:

Show slide 5 explain the topic of clauses and keywords in subquery,

To compare a single value provided by the subquery with the expression in the WHERE clause, a user must use comparison operators like =, >, etc. The following query, for example, returns the client with the highest payment.

```
SELECT  
customerNumber,  
checkNumber,  
amount  
FROM  
payments  
WHERE  
amount = (SELECT MAX(amount) FROM payments);
```

Inform students that a user can also utilize comparison operators like greater than (>), greater than or equal to (>=), less than(<), and less than or equal to (=) in addition to the = operator.

### Keywords:

ADD: To add a column to an existing table, use the ADD command.

ALL: If all of the subquery values fulfil the requirement, the ALL command returns true.

AND: The AND command is used in conjunction with WHERE to only include rows that satisfy both conditions.

ASC: The ASC command sorts the returned data in ascending order.

## Slide 6

### WHERE Clause in a Subquery (1-2)

The WHERE clause in a sub-query is used to filter the rows from set of rows of the resultant set, depending on the condition.

It compares the column of the outer query with the results returned by the sub-query.

WHERE clause is used along with SELECT, INSERT, UPDATE, and DELETE.

#### Instruction(s) to the trainer:

Show slide 6 and tell students that the WHERE clause is used to filter records. It can be utilized to retrieve only those records that meet a specific set of criteria.

Further, explain the syntax with the help of the following example.

The given subquery returns the department numbers for departments on the third floor. The outer query retrieves the names of employees who work on the third floor.

```
SELECT NAME FROM employee WHERE dept IN(SELECT DID FROM dept WHERE floor = 3);
```

Although aggregate functions cannot be directly used in a WHERE clause, they can be used in the SELECT or HAVING clause of a WHERE clause subquery.

**In-Class Question:** Can we use SELECT in WHERE clause?

**Answer:** To filter records and conduct various operations on the data, the WHERE clause can be used with SQL statements like INSERT, UPDATE, SELECT, and DELETE.

## WHERE Clause in a Subquery (2-2)

**SELECT Statement:** The WHERE clause compares the values returned by the subquery with the records in the table which is then displayed by the SELECT statement.

**INSERT Statement:** The WHERE clause compares the returned value by the subquery with the records in the table. It is selected by the SELECT statement and consequently inserted by the INSERT statement.

**UPDATE Statement:** The WHERE clause compares the values returned by the subquery with the records in the table. The records of the selected employees are then updated using the UPDATE statement.

**DELETE Statement:** The DELETE statement is used to delete the returned values of WHERE clause in the subquery.

### Instruction(s) to the trainer:

Show slide 7 and elaborate on WHERE Clause with different statements such as SELECT, INSERT, UPDATE, and DELETE.

**Using Subqueries in SELECT Statements:** Subqueries are most commonly used in SELECT statements. Further, these are mostly used in the WHERE clause of SELECT statements. Still, they can also be used in other clauses such as WHERE, FROM, and HAVING.

Explain the syntax:

```
SELECT name FROM students WHERE name NOT IN ( SELECT student_name FROM college )
```

**Using Subqueries in INSERT Statements:** In an INSERT statement, using a subquery is a common practice. In such cases, the entire result set from a subquery or SELECT statement is inserted into a table.

**Using Subqueries in UPDATE Statements:** The UPDATE statement can have a subquery in several places or clauses (similar to a SELECT query). The most frequent clauses are SET and WHERE in which subqueries are utilized in an UPDATE.

The new value for the column being edited by the UPDATE is defined in the SET clause. A user can get this new value using a subquery, which can come from any table or any legitimate subquery. This can happen as long as a user only gets one record with one column for each record being modified. The data type of the column returned by the subquery must match the data type of the changed column.

**Using Subqueries in DELETE Statements:** Subqueries can only be used inside a WHERE clause with the DELETE statement.

## Slide 8

**FROM Clause**

All the subqueries are placed inside the WHERE clause.

Subqueries used in FROM clause are useful for calculating aggregate values such as SUM(), MAX(), MIN(), AVG(), and COUNT().

For example, in a corporate database system, a user wants to display the average expenses of the company department-wise.

© Aptech Limited Practical MySQL / Session 5 / 8 of 14

### Instruction(s) to the trainer:

Show slide 8 and inform students that the result set returned from a subquery is used as a temporary table when a subquery is utilized in the FROM clause. This table is known as a materialised subquery or derived table.

Provide an example of a subquery that finds the maximum, lowest, and average number of items in sale orders:

```
SELECT  
MAX(items),  
MIN(items),  
FLOOR(AVG(items))  
FROM  
(SELECT orderNumber, COUNT(orderNumber) AS items FROM orderdetails GROUP BY  
orderNumber) AS lineitems;
```

It is worth noting that the FLOOR() function is used to remove decimal places from item average values.

### Additional Links:

Refer to following links for more information:

<https://www.javatpoint.com/mysql-from>

[https://www.w3schools.com/mysql/mysql\\_where.asp](https://www.w3schools.com/mysql/mysql_where.asp)

## Slide 9

### IN and NOT IN keywords in the Subquery

The IN and NOT IN keywords are used to compare more than one value in a list with the records in the table. If a record matches with any value in the list (with IN clause), the particular row/s are returned as a result dataset.

**IN Clause:** Following query fetches the records of the employees who work in departments having a rank of either 1 or 2:

```
SELECT * FROM Employee WHERE Dept IN (SELECT DID FROM Department WHERE DRanking = 1 OR DRanking = 2 OR DRanking = 3);
```

**NOT IN:** Following query fetches the records of the employees who do not work in the departments located in Texas or Alaska:

```
SELECT * FROM Employee WHERE Dept NOT IN (SELECT DID FROM Department WHERE Dlocation = 'Texas' OR Dlocation = 'California');
```

#### Instruction(s) to the trainer:

Show slide 9 and define the IN and NOT IN keywords in subqueries. Tell the students that if a subquery returns several values, a user can use different operators in the WHERE clause, such as IN or NOT IN.

Moreover, explain that subqueries are SQL queries that contain one or more SELECT statements nested within the WHERE clause of another SELECT statement. The outer query is the first statement in this form of query, whereas the inner query is the second statement. When such queries are run, the inner query is assessed first, and the outer query is given the result of the inner query.

Now, give an example where a user will make a schema for our database. Then, the developer will construct a table called geeks data and utilise the IN operator with a sub-query.

**IN:** `SELECT first_name, last_name FROM employee_data WHERE dept IN (SELECT dept_name FROM computer_dept WHERE id = 1);`

For example, a user can utilize a subquery with NOT IN operator to find the customers who have not placed any orders as follows:

```
SELECT customerName FROM customers WHERE customerNumber NOT IN (SELECT DISTINCT customerNumber FROM orders);
```

## EXISTS and NOT EXISTS

EXISTS and NOT EXISTS are operators that check whether rows exist or not in the database. It always returns a boolean value.

### Exist

The EXISTS subquery returns TRUE if it returns a dataset containing one or more records. Following query finds the details of the departments which has at least one employee working under them:

```
SELECT * FROM Department d WHERE EXISTS (SELECT * FROM Employee e WHERE e.Dept = d.DID);
```

### Not Exist

The NOT EXISTS statement combines two operators: EXISTS and NOT. It is the opposite of EXISTS. The subquery returns true if it returns a dataset not containing any record.

```
SELECT * FROM Department d WHERE NOT EXISTS (SELECT * FROM Employee e WHERE e.Dept = d.DID);
```

© Aptech Limited Practical MySQL / Session 5 / 10 of 14

### Instruction(s) to the trainer:

Show slide 10 and explain the keywords EXISTS and NOT EXISTS. In SQL, the EXISTS condition is used to determine whether the result of a correlated nested query is empty (i.e., includes no tuples) or not. EXISTS returns a boolean value of True or False. It can be used in a SELECT, UPDATE, INSERT, or DELETE statement.

### Explain the following syntax:

```
SELECT column_name(s) FROM table_name WHERE EXISTS (SELECT column_name(s) FROM table_name WHERE condition);
```

### Using EXISTS condition with SELECT statement

To fetch the first and last name of the customers who placed at least one order:

```
SELECT fname, lname FROM Customers WHERE EXISTS (SELECT * FROM Orders WHERE Customers.customer_id = Orders.c_id);
```

### Using NOT with EXISTS

To fetch the last and first name of the customers who has not placed any order:

```
SELECT lname, fname FROM Customer WHERE NOT EXISTS (SELECT * FROM Orders WHERE Customers.customer_id = Orders.c_id);
```

### **Using EXISTS condition with DELETE statement**

To delete the record of all the customers from Order Table whose last name is 'Milton':

```
DELETE FROM Orders WHERE EXISTS (SELECT * FROM customers WHERE  
Customers.customer_id = Orders.cid AND Customers.lname =  
'Milton');
```

### **NOT EXISTS:**

Tell students to suppose that a user wishes to choose all pupils with a grade of 9 or higher. He/She may do this by using NOT EXISTS, which contradicts the logic of the EXISTS operator. As a result, if the underlying subquery returns no results, the NOT EXISTS operation returns true. If the inner subquery matches a single record, the NOT EXISTS operator returns false, and the subquery execution can be halted. The user may use the following SQL query to find all student records that have no related student grade with a value less than 9:

```
SELECT  
    id, first_name, last_name  
FROM  
    student  
WHERE NOT EXISTS (  
    SELECT 1  
    FROM  
        student_grade  
    WHERE  
        student_grade.student_id = student.id AND  
        student_grade.grade < 9  
)  
ORDER BY id
```

## Types of Subquery

**Single Row Subquery:** A single row subquery returns no record or a single record to the outer query.

A single row subquery can be formed by placing a `SELECT` statement in a `WHERE` clause, `FROM` clause, or `HAVING` clause of a statement. Such subqueries are always used with single row operators.

Following query displays the details of the employee having the highest salary in the table:

```
SELECT * FROM Employee WHERE Salary = (SELECT MAX(Salary)  
FROM Employee);
```

### Instruction(s) to the trainer:

Show slide 11 and discuss the different types of subqueries present in MySQL.

- Single Row Subquery: In the results, zero or one row is returned.
- Multiple Row Subquery: One or more rows of results are returned.
- Multiple Column Subqueries: Subqueries with several columns.
- Correlated Subqueries: A linked subquery returns one or more columns in accordance with the primary or outer query.
- Nested Subqueries: There are queries within inquiries (inner and outer query).

### Additional Links:

Refer to following links for more information:

<https://data-flair.training/blogs/sql-subquery/>  
<https://www.w3resource.com/mysql/subqueries/index.php>

## Multiple Column Subquery

Following query displays the grouped dataset of maximum salary of each position (Clerk, Manager, and Salesman):

In multiple-column subqueries, the inner query returns a dataset of values of multiple columns to the outer query, which compares values of multiple columns with the dataset.

The dataset returned by the subquery is considered as a separate table. It is called a temporary table or inline view. If the subquery contains a `GROUP BY`, then the grouped data are treated as separate rows.

```
SELECT * FROM Employee WHERE (Position, Salary) IN (SELECT Position, MAX(Salary) FROM Employee GROUP BY POSITION);
```

### Instruction(s) to the trainer:

Show slide 12 explain the concept of Multiple Column Subquery in MySQL. The duplicate WHERE criteria can be combined into a single WHERE clause using multiple-column subqueries. A pair-wise or non-pairwise comparison can be made between columns in a multiple-column subquery.

Elaborate on the topic with the help of the following example:

Display the order number, product number, and quantity of any item in which the product number and quantity match both the product number and quantity of an item in ordid 364.

```
SELECT ordid, prodid, qty FROM item WHERE (prodid, qty) IN (SELECT prodid, qty FROM item WHERE ordid = 364)  
AND ordid = 364 ;
```

**Output:** ORDID PRODID QTY  
364 84 22

## Correlated Sub-query

Correlated subqueries work in an opposite way to how generic subqueries work. In basic subqueries, the inner query returns values that are operated upon by the outer query.

In correlated subqueries, the subquery operates on the values provided by the outer query..

Correlated subqueries cannot be run independently. They contain references to table names mentioned in the outer queries and gives an error on execution.

Query to display the details of the employees with a salary higher than the average salary of their department: `SELECT * FROM Employee E WHERE Salary > (SELECT AVG(Salary) AS avgSal FROM Employee WHERE Dept = E.Dept);`

### Instruction(s) to the trainer:

Show slide 13 and describe the concept of correlated subquery. Discuss the following pointers in detail:

- A correlated subquery refers to a column of a table that is not in the FROM clause of the query. The column can appear in both the Projection and the WHERE clauses. Correlated subqueries, in general, degrade performance.
- For each row processed by the parent statement, a connected subquery is examined once.
- A SELECT, UPDATE, or DELETE statement can be the parent statement.

Describe the working of following query:

`SELECT column1, column2, .... FROM table1 outer WHERE column1 operator (SELECT column1, column2 FROM table2 WHERE expr1 = outer.expr2);`

A correlated subquery is a method of reading each row of a table and comparing the values in each row to related data. It is utilized when a subquery has to deliver a different result or collection of results for each candidate row that the main query considers.

Simply put, a correlated subquery can be used to answer a multipart question whose response is dependent on the value in each row processed by the parent statement.

**Additional Links:**

Refer to following links for more information:

<https://www.javatpoint.com/mysql-subquery#:~:text=A%20correlated%20subquery%20in%20MySQL,row%20in%20the%20outer%20query.>

<https://dev.mysql.com/doc/refman/8.0/en/correlated-subqueries.html>

## Summary

- A subquery is a nested query.
- Basic subqueries return row/s to the outer query for comparison.
- Subqueries can be used with WHERE, HAVING, and FROM clauses.
- The DELETE statement is used to delete the returned values of the WHERE clause in the subquery.
- Subqueries used in FROM clause are useful for calculating aggregate values such as SUM(), MAX(), MIN(), AVG(), and COUNT().
- Single row operators and multi-row operators are important as comparison operators.
- Correlated subqueries cannot run independently as they contain alias names of tables mentioned in the outer query.

### Instruction(s) to the trainer:

Use slide 14 to summarize the session. End the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

## Session 6: Functions and Stored Procedures

### Slide 2

#### Session Overview

- Describe the usage of stored routines
- Explain different types of functions
- Outline the uses of stored procedures and functions
- Explain the use of functions in MySQL
- Explain how functions are created

#### Instruction(s) to the trainer:

- Give brief overview of the objective of the session. Say that students will understand about the use of functions in MySQL and explain how the functions are created. Also, they will be able to explain different types of functions and describe usage of stored routines. Functions and stored procedures explained in this session will help students to write codes faster.

## Introduction to Functions in MySQL

Functions in MySQL can be defined as a program that performs independent operations

Built in functions in MySQL:

- Mathematical functions
- Numeric functions
- Date/Time functions
- Null functions

Users are free to create user-defined functions depending on the requirement

© Aptech Limited

Practical MySQL / Session 6 / 3 of 14

**Instruction(s) to the trainer:**

- Using slide 3, explain to students about functions in MySQL.
- Say that there are many built-in functions in MySQL and users are free to create user-defined functions.
- There are many built-in functions in MySQL such as Numeric functions, Mathematical functions, Date/Time functions, NULL functions, and so on. Users are free to create user-defined functions depending on the requirement.

**Additional Information:**

Refer to following links for more information:

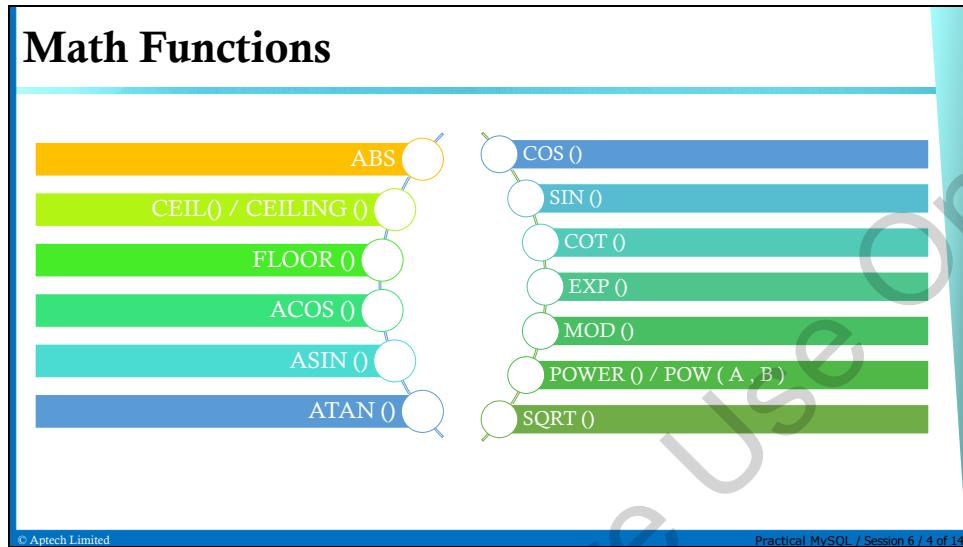
- <https://www.javatpoint.com/mysql-functions>
- <https://www.guru99.com/functions.html>

Ask students following question. Wait for a response before you answer.

**In-Class Question:** Define functions in MySQL?

**Answer:** It can be defined as a program that performs independent operations.

## Slide 4



### Instruction(s) to the trainer:

- Show slide 4 and say that mathematical functions in SQL are numeric functions used to perform calculations by taking input values as arguments.
- Explain the Math functions.
- Discuss about some important Math functions in MySQL such as ABS(), CEIL()/CEILING(), FLOOR(), and so on.

### Additional Information:

Refer to following links for more information.

- <https://www.javatpoint.com/mysql-functions>
- <https://www.guru99.com/functions.html>

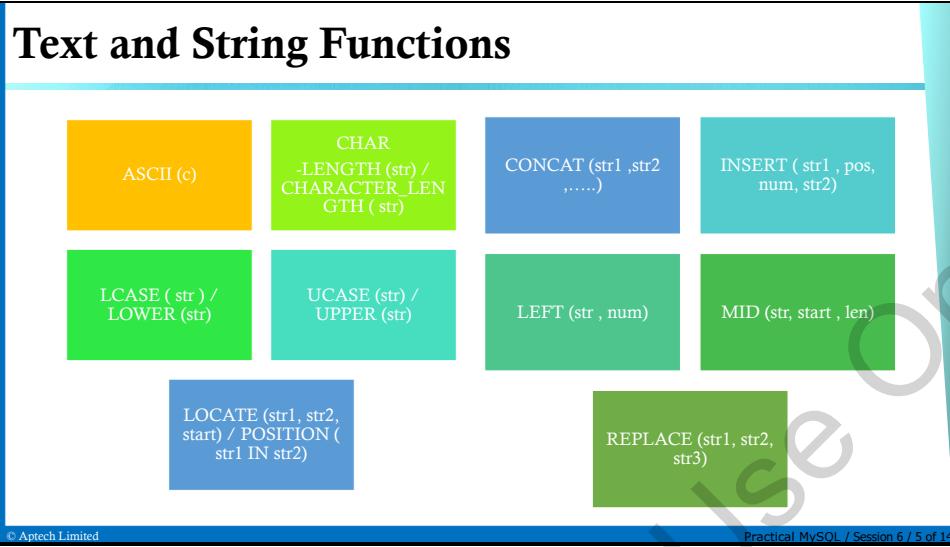
Ask students following question. Wait for a response before you answer.

**In-Class Question:** List the function of COS() and MOD().

**Answer:** COS () – Returns the cosine of the input value. The input must be in radians.

MOD () – Returns the value of the remainder when A is divided by B.

## Slide 5



### Instruction(s) to the trainer:

- Show slide 5 and say that the string functions perform significant operations on string-type input values. It returns either a string or a numeric value.
- Then, discuss text and string functions.
- Finally, explain some important string functions in MySQL such as ASCII(c), CHAR\_LENGTH(str)/CHARACTER\_LENGTH(str), CONCAT(str1,str2,..), INSERT(str1, pos, num, str2), LCASE(str)/LOWER(str), and so on.

### Additional Information:

Refer to following links for more information:

- <https://www.javatpoint.com/mysql-functions>
- <https://www.guru99.com/functions.html>

Ask the students following question. Wait for a response before you answer.

**In-Class Question:** Which function is used to perform specific words/phrases in search engines?

**Answer:** The LOCATE() function is used by search engines when specific words/phrases are typed in the search box. The search results are based on the phrases entered in the Google search box.

## Using Stored Routines

Stored routines in MySQL can be either procedures or functions.

Used as block of SQL statements in the MySQL server for re-use.

Users must refer to the stored routines instead of reissuing the individual statements everytime.

© Aptech Limited Practical MySQL / Session 6 / 6 of 14

**Instruction(s) to the trainer:**

- Show slide 6 and say that the stored routines in MySQL can be either procedures or functions. They are used as a block of SQL statements in the MySQL server for re-use. Users must refer to stored routines instead of reissuing the individual statements every time.
- Discuss the following:
  - How to create a stored routine?
  - How to drop a procedure?
  - How to drop a function?
- Then, list the important points to remember about stored procedures and routines.

**Additional Information:**

Refer to following links for more information:

- <https://www.guru99.com/functions.html>
- <https://www.mysqltutorial.org/mysql-stored-function>
- <https://www.guru99.com/functions.html>

## MySQL NULL Functions

Null functions are used to identify NULL Values.

MySQL IFNULL()	COALESCE()
-------------------	------------

© Aptech Limited Practical MySQL / Session 6 / 7 of 14

**Instruction(s) to the trainer:**

- Show slide 7 and explain about MySQL Null Functions.
- Say that NULL functions are used to identify NULL values. NULL values have no value, no data type, and can be fit into any column anytime. They are placeholders for values that are not available in the table.
- Discuss about few null functions defined in MySQL.

**Additional Information:**

Refer to following links for more information:

- <https://www.javatpoint.com/mysql-functions>
- <https://www.guru99.com/functions.html>

Ask students the following question. Wait for a response before you answer.

**In-Class Questions:** Which function returns NULL if expr1=expr2?

**Answer:** NULLIF()

## Slide 8

### Date and Time Functions

- DATE\_ADD (date, INTERVAL value unit)
- ADD TIME ( datetime1, datetime2 )
- CURDATE () / CURRENT\_DATE ()
- CURTIME () / CURRENT\_TIME ()
- CURRENT\_TIMESTAMP () / NOW ()
- DAY (date) / DAYOFMONTH (date)
- DAYOFWEEK (date)
- TO\_DAYS (date)
- SUBDATE (date, INTERVAL value unit)

© Aptech Limited Practical MySQL / Session 6 / 8 of 14

#### Instruction(s) to the trainer:

- Say that DATE and TIME functions are commonly used in applications involving daily operations of organizations. They are used internally with bulky codes of applications.
- Using slide 8, discuss the date/time functions available in MySQL.

#### Additional Information:

Refer to following links for more information:

<https://www.guru99.com/functions.html>

<https://www.javatpoint.com/mysql-functions>

Ask the students the following question. Wait for a response before you answer.

**In-Class Question:** What is the function of TO\_DAYS (date)?

**Answer:** It returns the total number of days from year 0 to input date.

## Creating Functions

Basic Syntax to Create a Function:

```
CREATE FUNCTION function_name [(parameter data_type [, parameter data_type])]  
RETURNS return_data_type  
BEGIN  
declarations_block  
execution_block  
END;
```

### Instruction(s) to the trainer:

- Say that functions in MySQL allows you to provide multiple parameters, but returns a single value.
- Show slide 9, and explain the syntax for creating functions.

### Additional Information:

Refer to following links for more information:

- <https://www.javatpoint.com/mysql-regexp-replace-function>
- <https://www.techonthenet.com/mysql/functions>

## Drop Function

Basic syntax of DROP Function:

```
DROP FUNCTION function_name;
```

Example:

```
DROP FUNCTION getEmployeeSalary();
```

### Instruction(s) to the trainer:

- Say that the drop function command is used to remove a function (if it exists) from database.
- Using slide 10, explain the syntax for creating drop function with example.
- Say that dropping the database would eventually drop all the routines within it. Also, a user may alter the procedure/function using the ALTER command.
- Mention that the stored procedures and routines correspond to only one database.

### Additional Information:

Refer to following links for more information:

- <https://www.javatpoint.com/mysql-regexp-replace-function>
- <https://www.techonthenet.com/mysql/functions.php>
- [https://www.tutorialspoint.com/mysql/mysql\\_drop\\_function.htm](https://www.tutorialspoint.com/mysql/mysql_drop_function.htm)

Ask students the following question. Wait for a response before you answer.

- **In-Class Question:** What is the syntax of the drop function? Give an example.
- **Answer:** The syntax for the drop function is
- `DROP FUNCTION function_name;`
- Example: `DROP FUNCTION getEmployeeSalary();`

## MySQL REGEXP\_INSTR () Function

Basic Syntax of MySQL REGEXP\_INSTR () Function:

```
REGEXP_INSTR(str, expr[, pos[, occurrence [, return_option[,  
match_type]]]])
```

Example:

```
SELECT REGEXP_INSTR('ppp pppp pppppp','p{5}');  
//Returns 10  
SELECT REGEXP_INSTR('lamp chair lamp', 'lamp');  
//Returns 1
```

### Instruction(s) to the trainer:

- Say that regular expression is a method of specifying a pattern to perform a complex search.
- Using slide 11, explain the syntax for REGEXP\_INSTR () Function with example.

### Additional Information:

Refer to following links for more information:

- <https://www.javatpoint.com/mysql-regexp-replace-function>
- <https://www.geeksforgeeks.org/mysql-regular-expressions-regexp>
- <https://www.tutorialspoint.com/mysql/mysql-regexp.htm>

## MySQL REGEXP\_LIKE () Function

Basic syntax of MySQL REGEXP\_LIKE () Function:

```
REGEXP_LIKE(str, expr[, match_type])
```

Example:

```
SELECT REGEXP_LIKE('PolIsh Boat', 'POLISH');//Returns  
1
```

### Instruction(s) to the trainer:

- Using slide 12, explain the syntax for REGEXP\_LIKE () Function.
- Say that the REGEXP\_LIKE function returns 1 if the string str matches with the regular input expression expr and returns NULL if str or expr is NULL. The REGEXP\_LIKE() returns a Boolean value as a result.

### Additional Information:

Refer to following links for more information:

- <https://www.javatpoint.com/mysql-regexp-replace-function>
- <https://www.geeksforgeeks.org/mysql-regular-expressions-regexp>
- <https://www.tutorialspoint.com/mysql/mysql-regexp.htm>

Ask students the following question. Wait for a response before you answer.

**In-Class Question:** What is the difference between the REGEXP\_LIKE() and REGEXP\_INSTR() function?

**Answer:** The difference is that REGEXP\_LIKE() returns a boolean value as a result whereas, REGEXP\_INSTR() returns an integer value.

## MySQL REGEXP\_REPLACE () Function

Basic syntax Of MySQL REGEXP\_REPLACE () Function:

```
REGEXP_REPLACE(str1,expr,str2[, pos[, occurrence[, return_option[, match_type]]]])
```

Example:

```
SELECT REGEXP_REPLACE('aaa.devJack.com', 'a', 'w');
//Returns www.devJwck.com
SELECT REGEXP_REPLACE('aaaghikaabaa','aa','i');
//Returns iaghikibi
```

### Instruction(s) to the trainer:

- Ask: Can anyone of you state the functionality of the REPLACE() function?
- Encourage the students to respond and appreciate them for the participation.
- Say that that REGEXP\_REPLACE function extends the functionality of the REPLACE function. It allows the users to search a string for a regular expression pattern.
- Then, using slide 13, explain the syntax for REGEXP\_REPLACE () function with example.

### Additional Information:

Refer to following links for more information:

- <https://www.javatpoint.com/mysql-regexp-replace-function>
- <https://www.geeksforgeeks.org/mysql-regular-expressions-regexp>
- <https://www.tutorialspoint.com/mysql/mysql-regexp.htm>

## Summary

- Various types of functions are present in MySQL Library.
- Mathematical functions help on complex calculations in between codes.
- String functions are essentially used for converting one data type to another data type.
- Date/Time functions allow users to precisely store date/time formatted values.
- Stored routines can be functions as well as procedures.
- REGEXP functions are used for pattern matching.

Use slide 14 to summarize the session. End the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

## Session 7: MySQL Clauses and Indexes

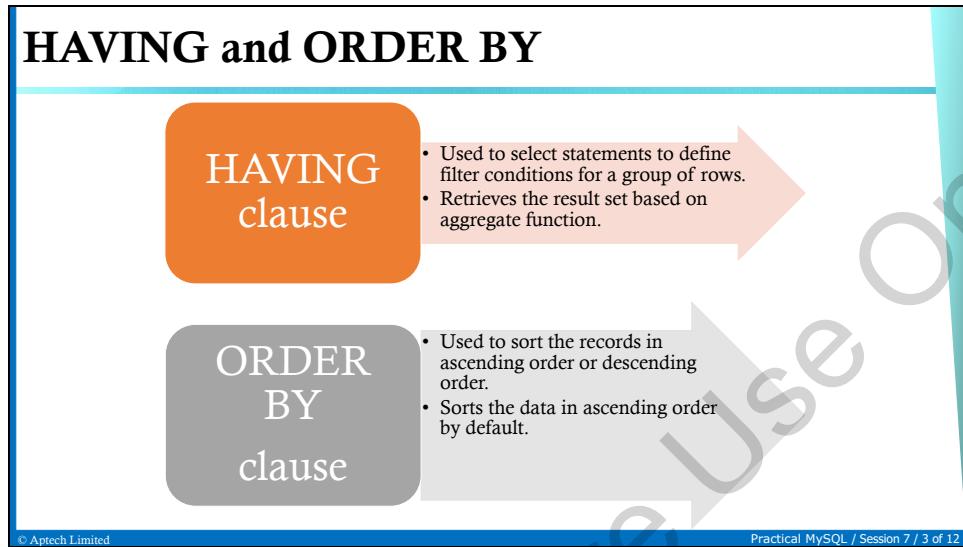
### Slide 2

#### Session Overview

- Describe MySQL HAVING and ORDER BY clause
- Describe MySQL GROUP BY clause
- Define and use ROLLUP Modifier
- Explain Indexes in MySQL

#### Instruction(s) to the trainer:

Show slide 2 and inform the students that in this session they will get a brief overview on MySQL clauses and understand the Use Roll Up modifier. They will also understand the indexes in MySQL. Indexes help users to find rows, eliminate rows, and retrieve rows from other tables, and so on.



**Instruction(s) to the trainer:**

- Say that WHERE keyword cannot be used in aggregate functions, hence, HAVING clause is used in SELECT statements to define filter conditions for a group of rows.
- Using slide 3, discuss the MySQL Having and Order by clauses.

Following is the syntax for ORDER BY clause:

```
SELECT column1, column2, ... FROM table_name ORDER BY column1,  
column2, ... ASC | desc;
```

Inform students that ASC attribute is used to sort data in the ascending order in ORDER BY clause. The desc attribute is utilized to sort and print data in decreasing or descending flow. However, the MySQL ORDER BY clause can be used without any attribute as well.

**Additional Information:**

Refer to following links for more information:

- <https://www.mysqltutorial.org/mysql-having.aspx>
- <https://www.guru99.com/group-by.html>
- <https://www.techonthenet.com/mysql/having.php>

Ask the students the following question. Wait for a response before you answer.

**In-Class Question:** What is the use of ORDER BY clause in MySQL and how does it sort by default?

**Answer:** The ORDER BY clause is used to sort the result set in ascending or descending order. ORDER BY clause sorts the data in ascending order by default.

## MySQL GROUP BY Clause

The diagram consists of two main sections. The left section, with a yellow background, contains the text: "Used for the purpose of categorizing all rows that have the same values." and "Can be combined with functions - known as aggregate functions". The right section, with a blue background, contains the text: "Aggregate functions:" followed by a list: COUNT Function, SUM Function, MIN Function, MAX Function, and AVG Function.

© Aptech Limited

Practical MySQL / Session 7 / 4 of 12

**Instruction(s) to the trainer:**

- Say that the GROUP BY clause is used for the purpose of categorizing all rows that have the same values.
- Using slide 4, explain the MySQL Group By Clause and then, discuss the functions used with the clause.

Following is the syntax for GROUP BY clause:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s);
```

The GROUP BY clause with the COUNT function is used to show the grouping of rows based on the column values and the aggregate function COUNT.

Explain in detail the aggregate functions of GROUP BY with examples to students.

- The GROUP BY clause with the SUM function is used to show the total number of orders received grouped datewise.
- The MIN function is another aggregate function used alongside the GROUP BY clause. It is utilized to retrieve minimum value of a column over each group against a condition.

- Equivalent to the MIN function, the MAX is another function that can be utilized alongside the GROUP BY clause. It is utilized to retrieve maximum value of a column over each group against a condition.
- The AVG is another function that can be utilized with GROUP BY clause. The average can be calculated using this function.

**Additional Information:**

Refer to following links for more information:

- <https://www.javatpoint.com/mysql-group-by>
- <https://www.mysqltutorial.org/mysql-group-by.aspx>
- [https://www.techonthenet.com/mysql/group\\_by.php](https://www.techonthenet.com/mysql/group_by.php)

Ask students following question. Wait for a response before you answer.

**In-Class Question:** What is a Group By clause?

**Answer:** Group By clause groups the selected rows based on identical values in a column or expression.

## Slide 5

### ROLLUP Modifier

Used with GROUP BY clause and provides summary output by including extra rows.

Can build up multiple groups of rows by using one single query.

User must have prior knowledge of grouping of set.

#### Instruction(s) to the trainer:

- Say that the ROLLUP in MySQL is a modifier used to spawn the summary output with extra rows. It helps to construct the sub totals and grand totals for the resultant
- Using slide 5, explain Rollup Modifier in MySQL and say that it is used to spawn the summary output with extra rows.
- Explain the syntax of Rollup Modifier and the method to use it.

Following is the syntax of ROLLUP Modifier:

```
SELECT
    column1, column2, column3, ...
FROM
    table_name
GROUP BY
    column1, column2,... WITH ROLLUP;
```

#### Additional Information:

Refer to the following links for more information:

- <https://www.tutorialspoint.com>
- <https://www.javatpoint.com/mysql-rollup>
- <https://www.mysqltutorial.org/mysql-rollup>

Ask students the following question. Wait for a response before you answer.

**In-Class Question:** Why do we use Rollup modifier in SQL?

**Answer:** Rollup enables a select statement to calculate multiple levels of subtotals across a specified group of dimensions.

## Slide 6

### Indexes in MySQL

Utilized to retrieve or fetch the data from database quickly.

Users can create an index using the CREATE INDEX statement.

Allocated to a data table by using Primary key, Unique key, or Simple key.

© Aptech Limited Practical MySQL / Session 7 / 6 of 12

#### Instruction(s) to the trainer:

- Using slide 6, explain indexes in MySQL.
- Say that it is utilized to retrieve or fetch the data from database quickly.

#### Additional Information:

Refer to following links for more information:

- <https://www.tutorialspoint.com/mysql/mysql-indexes>.
- <https://dev.mysql.com/doc/refman/8.0/en/mysql-indexes.html>
- <https://www.mysqltutorial.org/mysql-index>

Ask the students following question. Wait for a response before you answer.

- **In-Class Question:** What are the types of index in MySQL?
- **Answer:** MySQL has three types of indexes: Primary key, Unique key, and simple key.

## Reasons to Add an Index

Faster retrieval of data	To represent NULL values
Fetch specific values from columns quickly	Avoid searching every row
Optimize the performance of the database	

© Aptech Limited Practical MySQL / Session 7 / 7 of 12

**Instruction(s) to the trainer:**

Using slide 7, discuss with the students about reasons to add an index to a table.

**Additional Information:**

Refer to following links for more information:

- <https://www.tutorialspoint.com/mysql/mysql-indexes>.
- <https://www.javatpoint.com/how-to-create-index-in-mysql>
- <https://www.mysqltutorial.org/mysql-index>

Ask students following question. Wait for a response before you answer.

**In-Class Question:** Why do we need index in MySQL?

- Answer:**
- 1) Faster retrieval of data.
  - 2) Fetch specific values from columns quickly.
  - 3) To represent NULL values

## When to Use Index

- When columns have too many values
- When there are many NULL values
- When the columns are fetched frequently
- When the table data is large

© Aptech Limited Practical MySQL / Session 7 / 8 of 12

**Instruction(s) to the trainer:**

- Using slide 8, discuss the situations in which indexes can be used.

**Additional Information:**

Refer to following links for more information:

- <https://www.tutorialspoint.com/mysql/mysql-indexes.htm>
- <https://www.mysqltutorial.org/mysql-index/mysql-use-index>
- <https://linuxhint.com/create-index-mysql>

## When Not to Use Index

- When the table is small
- When the column is used rarely to retrieve the data
- When the table is non-updated

© Aptech Limited Practical MySQL / Session 7 / 9 of 12

**Instruction(s) to the trainer:**

Using slide 9, discuss the situations in which indexes are not recommended for use.

**Additional Information:**

Refer to following links for more information:

- <https://dev.mysql.com/doc/refman/8.0/en/mysql-indexes.html>
- <https://www.howtoforge.com/when-to-use-indexes-in-mysql-databases>

## MySQL CREATE INDEX Statement

An index can be built using the CREATE INDEX statement

- Syntax

```
CREATE INDEX Index_name on Table_name(Column_1, Column_2,...)
```

### Instruction(s) to the trainer:

- Using slide 10, explain about an index in MySQL.
- Say that indexes helps to retrieve data from the database quickly.
- Say that it can be built by using CREATE INDEX statement.

### Additional Information:

Refer to following links for more information:

- [https://www.w3schools.com/sql/sql\\_create\\_index.asp](https://www.w3schools.com/sql/sql_create_index.asp)
- <https://www.mysqltutorial.org/mysql-index/mysql-create-index>
- <https://www.javatpoint.com/how-to-create-index-in-mysql>

## MySQL Drop Index

DROP INDEX is used with ALTER TABLE command

Using the DROP PRIMARY KEY command it is possible to drop the primary key index

- Syntax

```
ALTER TABLE table_name DROP INDEX index_name;
```

### Instructions to the trainer:

- Using slide 11, discuss about Drop Index in MySQL.
- Say that if an index is not required anymore, the particular index can be dropped.
- Say that the syntax requires the table name to be specified because MySQL allows index names to be reused on multiple tables.

### Additional Information:

Refer to following links for more information:

- <https://www.mysqltutorial.org/mysql-index/mysql-drop-index>
- <https://www.javatpoint.com/mysql-drop-index>
- <https://www.techonthenet.com/mysql/indexes.php>

Ask the students following question. Wait for the response before you answer.

**In-Class Question:** What is a Drop Index?

**Answer:** DROP INDEX is used with ALTER TABLE command. When an index is not required anymore, the particular index can be dropped from the database.

## Summary

- MYSQL clauses are set of rules to understand the concepts.
- An index in MySQL helps in improving the speed of operations in a table.
- The ASC attribute is used to sort data in the ascending order.
- ALTER TABLE is used to drop primary from a table in a database.
- INDEX command is used to retrieve the data from database quickly.
- SHOW INDEX is used to get index information of the table.
- The GROUP BY clause is utilized to combine the rows with exact same values.
- The ROLLUP Modifier in MySQL is utilized to include extra rows as part of summary output.

Use slide 12 to summarize the session. End the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

## Session 8: Transactions, Performance Management, and Backup

### Slide 2

#### 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

#### Instruction(s) to the trainer:

Show slide 2 and give students a brief outline of what they will learn in this session. Tell them that the session covers the concepts including transactions, various commands used in transactions, and transactions using the JDBC driver. The session will also acquaint them with the different techniques to enhance and manage database performance. Finally, they will understand the basics of Replication.

## Slide 3

### 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.

#### Instruction(s) to the trainer:

Show slide 3 and explain the concept of a transaction in MySQL.

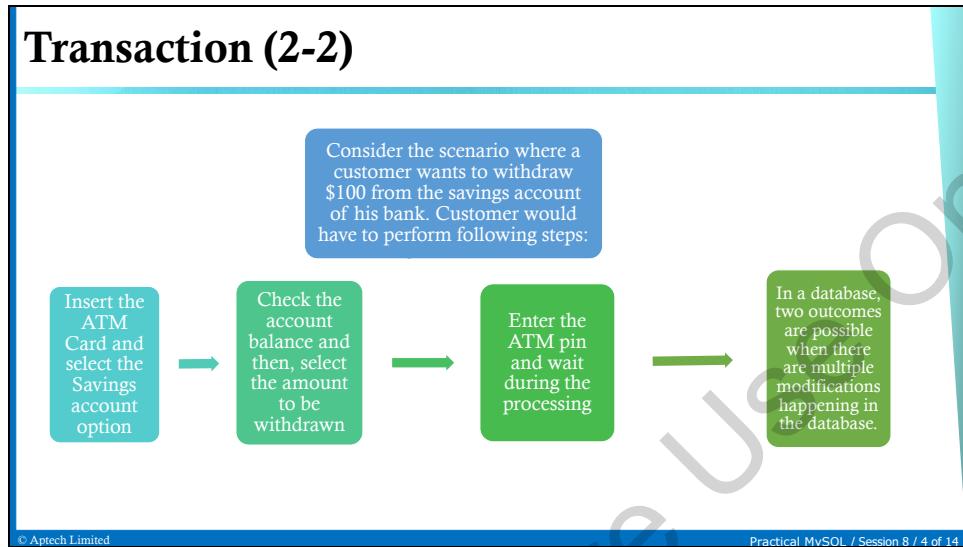
- . In MySQL, a transaction is a set of statements, queries, or operations, such as SELECT, INSERT, UPDATE, or DELETE, that are executed sequentially as a single work unit and can be committed or rolled back.
- . When a transaction makes many database modifications, two things can happen either the transaction is committed or all modifications are successful.
- . When a transaction is rolled back, all changes are undone. In other words, a transaction cannot be successful unless all of the operations in the set are completed. It means that if any statement fails, the transaction activity will fail.
- . In MySQL, a transaction begins with the first SQL statement that may be executed. It then ends when it finds a commit or is rolled back, either expressly or implicitly.

#### Additional Information:

Refer to following links for more information:

<https://dev.mysql.com/doc/refman/8.0/en/commit.html>  
<https://www.mysqltutorial.org/mysql-transaction.aspx>

## Slide 4



#### Instruction(s) to the trainer:

Show slide 4 and elaborate on transactions with the help of an example.

Ask students to consider a banking database for understanding the concept of a transaction in MySQL. Suppose a bank customer wants to transfer funds from one account to another. This can be accomplished through SQL statements. Now, explain following steps in the class:

- The first step is to verify that the requested amount is available in the first account.
- Deduct the money from the first account if it is available. Then, make a change to the first account.
- Finally, make a deposit into the second account. Then, to finalize the transaction, update the second account.
- The transaction will be rolled back to its former state if any of the foregoing processes fail.

## Properties of Transactions

The diagram consists of four orange rectangular boxes arranged horizontally, each representing one of the four properties of a transaction. The boxes are slightly tilted at an angle. A blue vertical bar is positioned to the right of the fourth box.

- 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.

© Aptech Limited Practical MySQL / Session 8 / 5 of 14

### Instruction(s) to the trainer:

Show slide 5 and discuss the properties of a transaction in MySQL.

The transaction primarily consists of four attributes, referred to as the ACID property. The acronym for the ACID property is Atomicity, Consistency, Isolation, and Durability.

• **Atomicity:** This feature guarantees that all the statements or actions within a transaction unit must succeed. Otherwise, if any operation fails, the transaction will be terminated and the data will get rolled back to its previous state. Inform students about its features:

- COMMIT statement
- ROLLBACK statement
- Auto-commit setting
- Operational data from the INFORMATION\_SCHEMA tables

• **Consistency:** This feature ensures that the database only changes state when a transaction is successfully committed. It is also in charge of keeping data safe from crashes. It has the following features:

- InnoDB doublewrite buffer
- InnoDB crash recovery

- **Isolation:** This feature ensures that each transaction unit operation gets performed individually. It also makes sure that the statements are mutually transparent. It has the following features:

- SET ISOLATION LEVEL statement
- Auto-commit setting
- The low-level details of InnoDB locking

- **Durability:** This attribute ensures that the outcome of committed transactions is preserved even if the system fails or crashes. It has the following features:

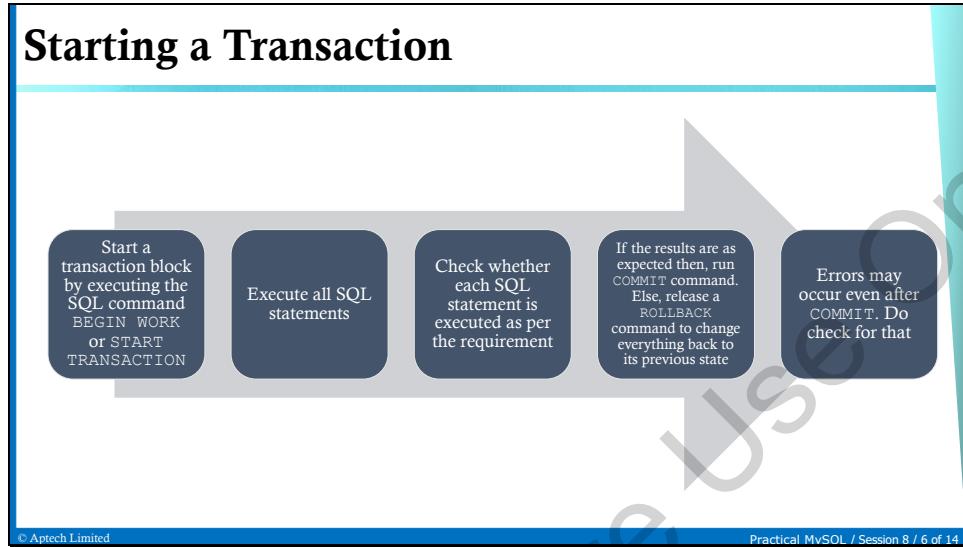
- Write buffer in a storage device
- Battery-backed cache in a storage device
- Configuration option innodb\_file\_per\_table
- Configuration option innodb\_flush\_log\_at\_trx\_commit
- Configuration option sync\_binlog

#### **Additional Information:**

Refer to following links for more information:

<https://www.javatpoint.com/mysql-transaction>

<https://www.tutorialspoint.com/mysql/mysql-transactions.htm>



**Instruction(s) to the trainer:**

Show slide 6 and explain the process of starting a transaction in MySQL. Describe following step-wise procedure in detail:

- The START TRANSACTION command is used to start the transaction in MySQL. It also has an alias for the START TRANSACTION called ‘BEGIN’ and ‘BEGIN WORK’.
- To commit the current transaction, a user can utilize the COMMIT command. It enables the database to make lasting modifications.
- To undo the current transaction, one can use the ROLLBACK statement. It enables the database to undo all modifications and return to its original state.
- To disable/enable the auto-commit mode for the current transaction, a programmer can use the SET auto-commit command.
- The COMMIT statement is executed automatically by default. However, if a user does not want modifications to be committed automatically, he/she can use following statement:

**SET** autocommit = 0;

OR,

**SET** autocommit = OFF:

## Slide 7

### 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.

© Aptech Limited Practical MySQL / Session 8 / 7 of 14

#### Instruction(s) to the trainer:

Show slide 7 and explain the concepts of Commit, Rollback, and Auto Commit in MySQL. All user activity in InnoDB happens within a transaction. If autocommit mode is enabled, each SQL statement is treated as a separate transaction. MySQL starts each new connection's session with autocommit enabled, which means MySQL commits after each SQL statement if the statement does not return an error. If a statement fails, the commit or rollback behavior is determined by the error.

When autocommit is enabled, a session can initiate a multiple-statement transaction with an explicit `START TRANSACTION` or `BEGIN` statement and end it with a `COMMIT` or `ROLLBACK` statement.

A `COMMIT` indicates that the current transaction's changes are made permanent and visible to other sessions. A `ROLLBACK` statement, on the other hand, undoes all of the current transaction's changes. All InnoDB locks that were set during the current transaction are released with both `COMMIT` and `ROLLBACK`.

## Slide 8

### 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;

#### Instruction(s) to the trainer:

Show slide 8 and elaborate on the commands used in transactions.

## 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.

### Instruction(s) to the trainer:

Show slide 9 and describe the concept of a transaction using the JDBC driver. Use following example to deepen students' understanding:

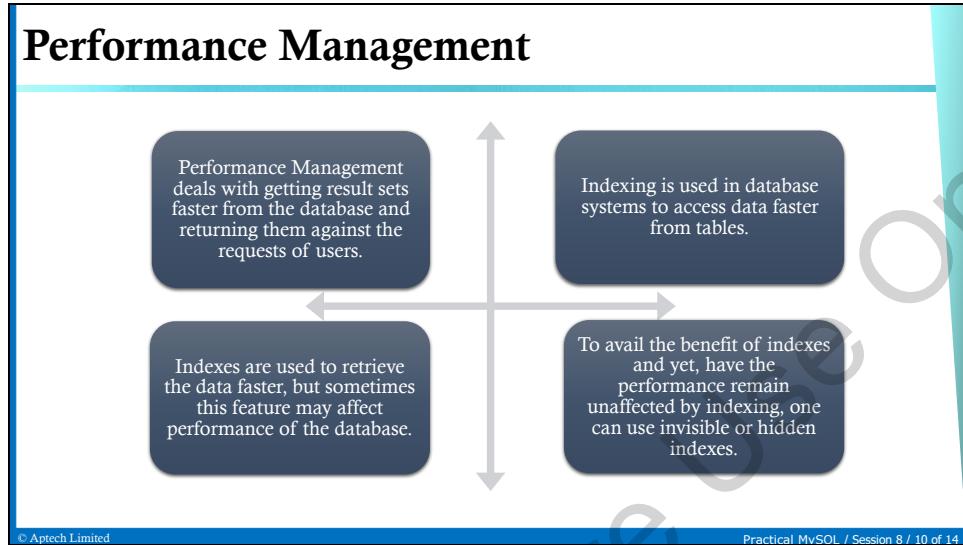
- First, the user creates a new record into the candidates' database and assigns some skills to the newly inserted applicant as an example.
- Within a single transaction, the user will insert an applicant and assign skills.
- Following measure must be taken to get the inserted ID back after inserting a record into the candidate's table.
- In the candidate skills table, a user will enter a set of candidate IDs and Skill IDs.
- Finally, Commit the transaction if all of the foregoing operations were successful; otherwise, roll it back.

### Additional Information:

Refer to following links for more information:

<https://www.mysqltutorial.org/mysql-jdbc-transaction/>

<https://riptutorial.com/mysql/example/20923/transaction-using-jdbc-driver>



**Instruction(s) to the trainer:**

Show slide 10 and tell students about performance management in MySQL. Inform students that there may be a million rows in the tables directly affecting an app's performance. Discuss the different techniques to deal with such performance issues.

**Using Indexes:** The most simple solution to the majority of slow query concerns is using indexes. A user must try this method before moving on to further techniques.

Introduce students to the types of indexes available in MySQL:

- Primary index: It is a one-of-a-kind index in which no value can be null. It is commonly used for the identification column and cannot be modified.
  - Unique index: All column values must be unique in a unique index.
  - Normal index: It can be on one or more columns and is mostly used to find items more quickly.
  - Descending index (MySQL 8+): It is a standard index stored in reverse order. It comes in handy if the user requires to show the most recently inserted data.
  - Full-text index: It is used for text indexing.
- **Using Cache:** It is a useful way to solve performance problems. It can be used at the program level with Memcached services such as Redis or the database level with SQL. On the SQL level,

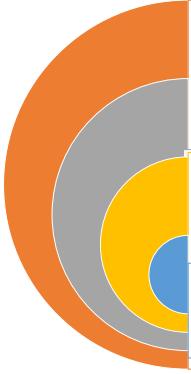
one should be aware that it has been deprecated. Moreover, it has already been removed in version 8 and beyond.

- If a user decides to use a program-level cache, the most difficult part is figuring out how to keep the cache and database consistent. Caching can be used as a bridge between the code and the database or it can be employed in certain situations. It can also be used to store data after it has been manipulated rather than as raw queries.

**In-Class Question:** What is the performance schema in MySQL?

**Answer:** MySQL Performance Schema is a feature that allows you to keep track of how the MySQL Server is performing at a low level. The Performance Schema allows you to inspect the server's internal execution while it is running.

## 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.

© Aptech Limited Practical MySQL / Session 8 / 11 of 14

### Instruction(s) to the trainer:

Show slide 11 and tell students about hiding an indexed column in MySQL.

Hidden indexes can be used on indexes other than the main key indexes. They are not used in query optimization, but they can be used to assess performance as if no indexes were present.

Indexes are visible by default. However, the `INVISIBLE` keyword can be used to make them invisible. It is worth noting that the primary key index can not be disguised.

`HAVING` is used when combined with a `GROUP BY` clause. `HAVING` does not use indexing very much.

The most efficient way to perform `GROUP BY` is using an index to directly get the grouping columns based on some aggregate condition. This is another method for enhancing performance.

### Additional Information:

Refer to following links for more information:

<https://dev.mysql.com/doc/refman/8.0/en/invisible-columns.html>

<https://blogs.oracle.com/mysql/post/how-to-use-invisible-indexes-in-mysql-80>

## 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

### Instruction(s) to the trainer:

Show slide 12 and explain backup and replication in MySQL.

- **Back Up from the Command-Line with mysqldump:** The client utility `mysqldump` can dump a database, including SQL statements needed to recreate it. The SQL commands to restore the tables and data are included in the dump file by default.

Following is the usual syntax for backing up your MySQL database:

```
sudo mysqldump -u [user] -p [database_name] > [filename].sql
```

Replace `[user]` with your username and password (if needed).

The `[database_name]` is the path and filename of the database.

The `>` command specifies the output.

`[filename]` is the path and filename a user wants to save the dump file as.

- **Logical backup:** This backup is saved as a `.sql` file. This type does not involve or require any third-party dependencies. The `mysqldump` command can be used to create a backup.
- **Physical backup:** It is made by duplicating database files. Physical backups take substantially longer to make and restore than logical backups. Large databases should be backed up with it.

- **Hot backup:** If MySQL Server is operating on the InnoDB subsystem, users can create transactional consistent backups without interrupting or restarting the server.
- **Partial:** Partial backups are not utilized to create backups for a database's whole storage. It is used to backup individual tables or databases.

For Aptech Centre Use Only

## 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 are known as replicas.

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

### Instruction(s) to the trainer:

Show slide 13 and discuss the basics of replication. MySQL replication is a technique for keeping data in one or more MySQL database servers, known as slaves, in sync with one master MySQL database server. Replication is useful because it provides for scalability by distributing read access over different servers, helps data backup, and allows data analysis on the slave without requiring access to the master.

### Additional Information:

Refer to following links for more information:

<https://www.digitalocean.com/community/tutorials/how-to-set-up-replication-in-mysql>  
<https://www.educative.io/edpresso/what-is-mysql-replication>

## 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.

**Instruction(s) to the trainer:**

Show slide 14 and provide students with a quick summary of all the topics taught in the session. Discuss the important pointers as this will serve as a revision of the current session.

## Session 9: Replication and Scalability

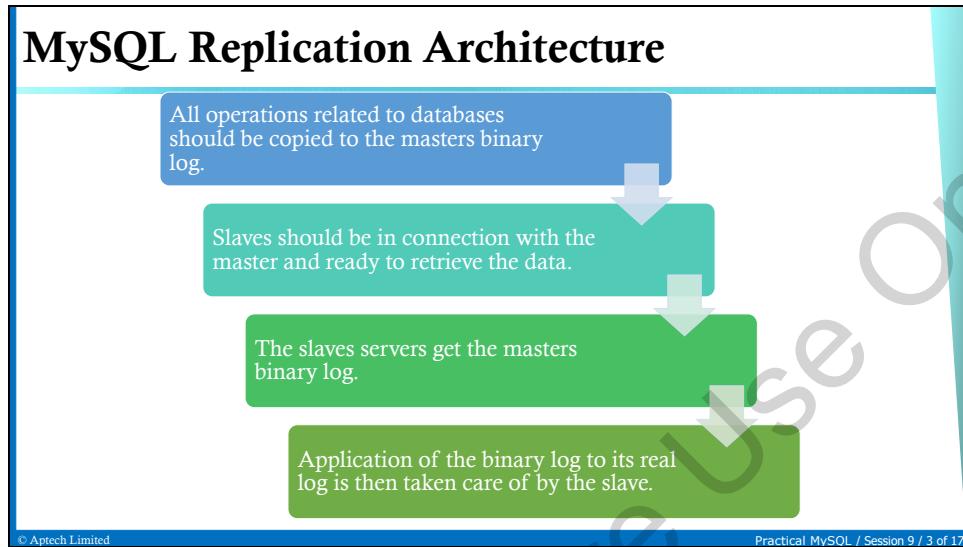
### Slide 2

#### Session Overview

- Explain MySQL replication architecture
- Develop and build on replication
- Describe replication topologies
- List replication use cases
- Summarize types of replications
- Identify advantages and disadvantages of MySQL replication

#### Instruction(s) to the trainer:

Using slide 2, give brief overview on MySQL replication architecture. Discuss the advantages and disadvantages of MySQL replication. Describe replication topologies. Explain how to develop and build on replication. List replication use cases.



**Instruction(s) to the trainer:**

Show slide 3 and discuss with students about the stages involved in MySQL replication process. Say that each directory server picks a replication server and connects to that replication server. All changes processed by the directory server are then sent to the replication server and all changes from other servers are received through the replication server.

**Additional Information:**

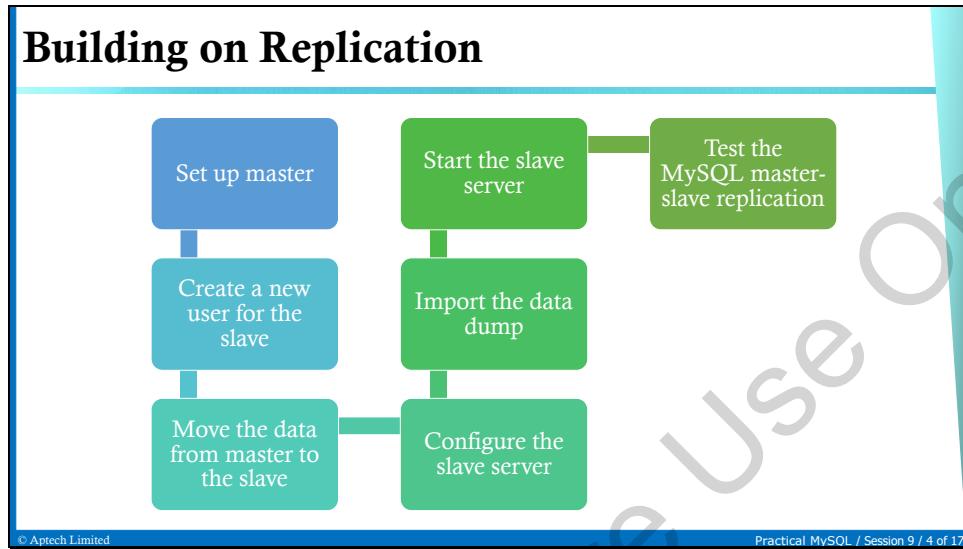
Refer to following links for more information:

- <https://dev.mysql.com/doc/internals/en/replication.html>
- <https://severalnines.com/resources/database-management-tutorials/mysql>

**In-Class Question:** What is MySQL replication architecture?

**Answer:** MySQL replication is a process that enables data from one MySQL database server (the master) to be copied automatically to one or more MySQL database servers (the slaves).

## Slide 4



#### Instructions to the trainer:

Using slide 4, inform students that the procedure of building on replication involves increasing the availability of data. The primary reason for the replication is recovering data. Explain the procedure to adhere to while building a master-slave replication.

#### Additional Information:

Refer to following links for more information:

- <https://dev.mysql.com/doc/mysql-replication>
- <https://www.toptal.com/mysql/mysql-master-slave-replication-tutorial>

## Slide 5

# Replication Topologies

Replication topologies can add another replica to an existing replication configuration without the source server getting stopped.

The replication topology manager helps in following ways:

Store topology and its status	Provide the GUI,CUI, or the API to check the status and perform the required operations	Give support to automatic failure of master while fixing the replication tree	Support topologies from a single master-slave replication to complex multi layered replication trees
-------------------------------	---	---	--

© Aptech Limited Practical MySQL / Session 9 / 5 of 17

### Instruction(s) to the trainer:

Show slide 5 and explain replication topologies. Say that the replication topologies can add another replica to an existing replication configuration without the source server getting stopped. To administer the effects of different topologies, the ‘Orchestrator’ can be utilized. Discuss the advantages of replication topology manager.

Different topologies for MySQL replication are:

- Master with Slaves (Single Replication)
- Master with Relay Slaves (Chan Replication)
- Master with Active Master (Curricular Replication)
- Master with Backup Master (Multiple Replication)
- Multiple Masters to Single Slave (Multi-Source Replication)
- Galera with Replication Slave (Hybrid Replication)
- MySQL Group Replication

### Additional Information:

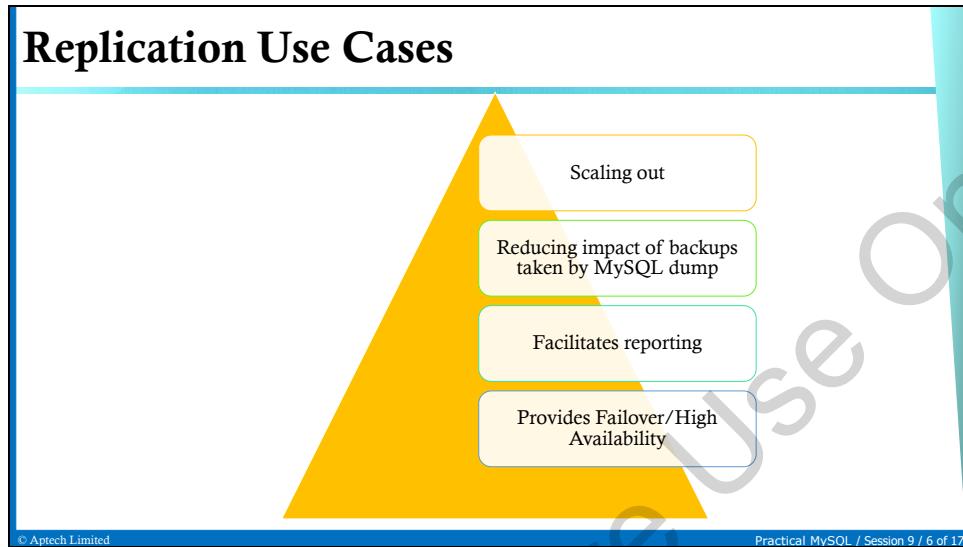
Refer to following links for more information:

- <https://dev.mysql.com/doc/mysql-replication>

- <https://www.slideshare.net/isotopp/mysql-replication-topologies-and-architectures>

**In-Class Question:** What is Replication Topologies?

**Answer:** Replication topologies can add another replica to an existing replication configuration without the source server getting stopped.



**Instruction(s) to the trainer:**

Say that a database replication tool allows one to move or consolidate data from one particular database server to a newer version of that database. Show slide 6 and list the general MySQL replication use cases. Say that replication use cases can differ based on type of replication. Redundancy and replication can help save from ransomware attacks. Finally, describe the use cases for group replication.

Following are the use cases for group replication:

**Elastic Replication:** It is used in places where fluid replication infrastructure is present. In this replication, the number of servers can grow and shrink dynamically, but side effects are minimal.

**Highly Available Shards:** It is one of the most famous approaches to achieving write scale-out. Here, each shard maps to a replication group.

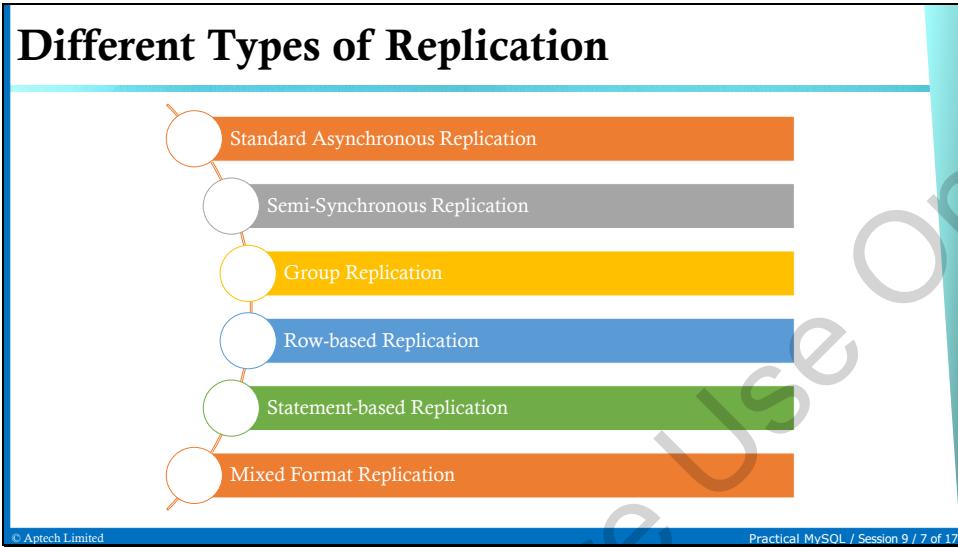
**Alternative to Source-Replica Replication:** In a few circumstances, replication is a single point of contention when utilizing a single source server.

**Autonomic Systems:** Deployment of MySQL group replication can be done purely for automation. The replication protocols have this built into them.

**Additional Information:**

Refer to following links for more information:

- <https://www.oracle.com/technetwork/community/developer-day/msql>
- <https://dev.mysql.com/doc/refman/8.0/en/replication.html>



**Instruction(s) to the trainer:**

Using slide 7, discuss different types of replication. Tell that replications format mainly has two core types. One is statement based and the other one is row based.

There is also a third type called mixed based replication. Explain different types of replication in detail to students.

- Standard asynchronous replication ensures that the local environment transaction is completed and is not influenced by replication slaves.
- Semi-synchronous replication ensures that proper transfer of database transactions occur.
- Group replication is a process by which a pool of servers is established so that they can be used to copy data correctly.
- Row-based replication or RBR is a type of replication in which the binary log stores record level changes which takes place on database tables to the master server.
- Statement-based replication is a type of replication where SQL statements are stored in binary log in order to change databases on the master server.
- Mixed format replication involves a server which can dynamically choose Statement Based Replication (SBR) or Row Based Replication (RBR) as per the condition.

**Additional Information:**

Refer to following links for more information:

- <https://www.oracle.com/technetwork/community/developer-day/mysql>
- <https://severalnines.com/resources/database-management-tutorials/mysql-replication-high-availability-tutorial>

**In-Class Question:** What is Group Replication?

**Answer:** Group replication is a process by which a pool of servers is established so that they can be used to copy data correctly.

## Slide 8

### Difference Between SBR and RBR

Statement-Based Replication	Row-Based Replication
<ul style="list-style-type: none"><li>• All SQL statements which can carry out a modify operation gets logged to the binary log.</li><li>• If the applications creates two update operations, then those two statements get executed on the slave.</li></ul>	<ul style="list-style-type: none"><li>• These represent complete copies of rows which are written to the binary log and later applied to the slaves.</li><li>• If the same row in the application is modified two times, then two copies of that specific row will be written to binary log and then applied to the slaves.</li></ul>

© Aptech Limited Practical MySQL / Session 9 / 8 of 17

#### Instruction(s) to the trainer:

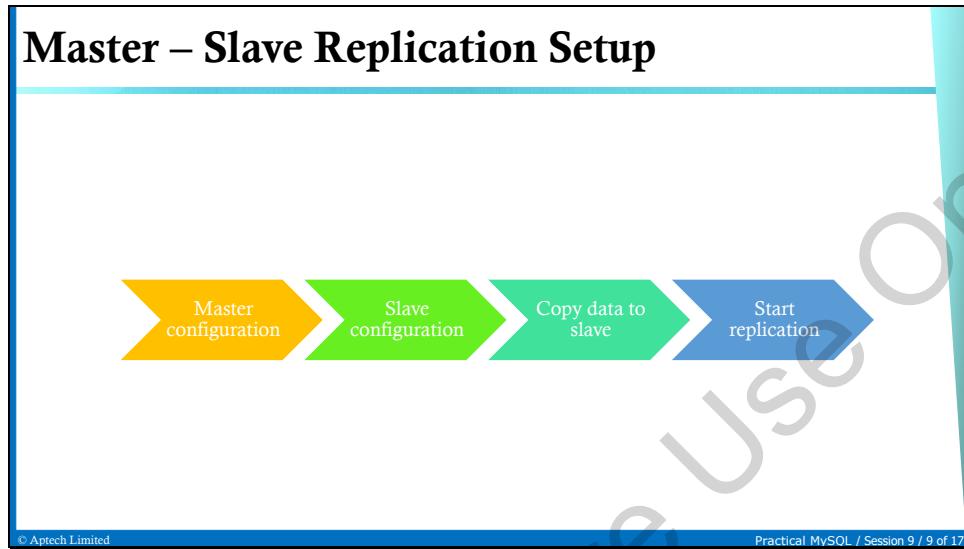
Using slide 8, explain statement-based replication and row-based replication.

Then, discuss the differences between statement-based replication and row-based replication.

#### Additional Information:

Refer to following links for more information:

- <https://www.databasejournal.com/mysql/comparing-mysql-statement-based-and-row-based-replication>
- [https://www.linuxtopia.org/online\\_books/database\\_guides/mysql\\_5.1\\_database\\_reference\\_guide/replication-sbr-rbr.html](https://www.linuxtopia.org/online_books/database_guides/mysql_5.1_database_reference_guide/replication-sbr-rbr.html)



**Instruction(s) to the trainer:**

Using slide 9, explain the process of master-slave replication.

Say that it enables database administrator to replicate or copy the data on more than one server simultaneously.

**Additional Information:**

Refer to following links for more information:

- <https://www.toptal.com/mysql/mysql-master-slave-replication-tutorial>
- <https://dev.mysql.com/doc/refman/5.6/en/replication-howto.html>

## Replication Errors

- MySQL replication error 1146**
  - Generic error when the user sets up the MySQL Replication.
  - The main reason for this error is MySQL queries or non existing SQL Queries.
- Error 1602**
  - This error may occur because of duplicate entry.
- Error 1053**
  - This error may occur because of partial completed query.

© Aptech Limited Practical MySQL / Session 9 / 10 of 17

**Instruction(s) for the trainer:**

Using slide 10, explain that while doing master-slave replication there are possibilities of getting errors.

Discuss about the replication errors.

**Additional Information:**

Refer to following links for more information:

- <https://dev.mysql.com/doc/mysql-replication-excerpt/5.7/en/replication-features-slaveerrors.html>
- <https://riptutorial.com/mysql/example/24095/replication-errors>

## Slide 11

### Advantage and Disadvantages of MySQL Replication

Advantages of MySQL Replications	Disadvantages of MySQL Replications
<ul style="list-style-type: none"><li>• It provides a consistent copy of the data across the nodes to increase data availability.</li><li>• Users can access data relevant to their tasks without interfering with the work of other users.</li><li>• Improves data retrieval of global queries and performance.</li><li>• Data is available on multiple locations and queries are executed faster.</li></ul>	<ul style="list-style-type: none"><li>• Achieving concurrency is hard.</li><li>• Slow updating procedure as one particular update should be performed on multiple places.</li></ul>

© Aptech Limited Practical MySQL / Session 9 / 11 of 17

#### Instructions to the trainer:

Using slide 11, explain the multiple advantages of MySQL replication. Then, list few disadvantages.

#### Additional Information:

Refer to following links for more information:

- <https://www.oracle.com/technetwork/community/developer-day/mysql-replication>
- <https://severalnines.com/resources/database-management-tutorials/mysql-replication-high-availability-tutorial>

## Scalability in MySQL

- Increase the disk size of the MySQL replica with shutdown
- Increase the disk size of the MySQL primary with shutdown
- Increase disk size of the MySQL replica without shutdown
- Increase the disk size of the MySQL primary without shutdown

© Aptech Limited Practical MySQL / Session 9 / 12 of 17

**Instruction(s) of the trainer:**

Using slide 12, explain the different ways to scale MySQL database.

**Additional Information:**

Refer to following link for more information:

- <https://www.oracle.com/technetwork/community/developer-day/mysql-replication-scalability-403030.pdf>

## Advantages and Disadvantages of Row based Replication in MySQL

**Advantages**

- No context-based information is required for performing guaranteed operations which get executed in correct order
- Fewer amounts of row level locks are necessary

**Disadvantages**

- It makes it harder to audit the modifications to the databases
- It can increase disk space as large amounts of data are generated
- It may lead to traffic in the network and IO related operations

© Aptech Limited Practical MySQL / Session 9 / 13 of 17

**Instructions to the trainer:**

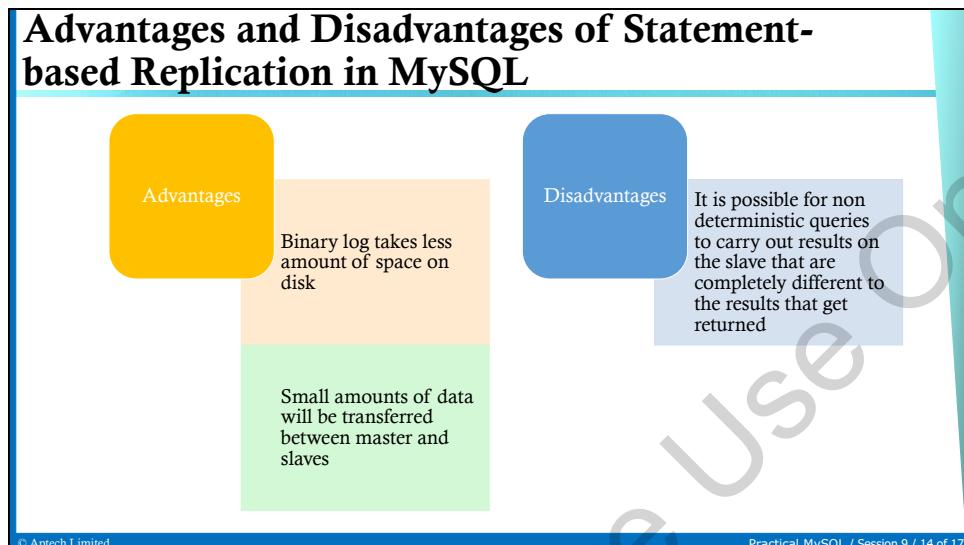
Using slide 13, explain row based replication.

Discuss the advantages and disadvantages of row based replication in MySQL.

**In-Class Question:** What are the advantages of row based replication?

**Answer:** 1) All changes can be replicated.

2) Fewer amount of row level locks are necessary.



**Instruction(s) to the trainer:**

Using slide 14, explain statement-based replication.

Discuss the advantages and disadvantages of statement-based replication in MySQL.

**Additional Information:**

Refer to following links for more information:

- <https://dev.mysql.com/doc/mysql-replication>
- <https://www.exploredatabase.com/2014/08/advantages-and-disadvantages-of-data-replication-in-distributed-databases.html>

## Replication Heartbeat



It is a signal that ensures to one copy of the database that the other is still working and functional.

A heartbeat table can be created on master and this will hold the time at which the row gets inserted on master.

The main process of replication heartbeat involves the source that sends a periodic heartbeat signal to a replica to ensure proper function.

### Instruction(s) to the trainer:

Using slide 15, explain the replication heartbeat.

Say that the main process of replication heartbeat involves the source that sends a periodic heartbeat signal to replica to ensure proper function.

### Additional Information:

Refer to following links for more information:

- <https://dev.mysql.com/doc/mysql-replication-excerpt/5.7/en/replication-administration-status.html>
- <https://docs.oracle.com/cd/E19623-01/820-6169/changing-heartbeat-interval.html>

## Replication Solution

Handling unforeseen stoppage of replica

Use replication with several replica storages and sources

Use appropriate methods for replication as per the requirements

Improve the replication performance by having good hardware resources such as RAM, CPU, and SSD

Switch sources during failover

### Instruction(s) to the trainer:

Using slide 16, discuss the solutions that can be used while performing replication.

Say that it is important to note that while performing scale-out solutions, there might be requirements to make some changes in logic and operations of application.

### Additional Information:

Refer to following links for more information:

- <https://dev.mysql.com/doc/mysql-replication-excerpt/5.7/en/replication-solutions.html>
- <https://www.ibm.com/analytics/data-replication>

## Summary

- Replication allows us to copy data from one MySQL database server which is also known as Source to one or more MySQL Database Servers which is known as replicas.
- By implementing different topologies. It is plausible to add another replica to a replication configuration that already exists without stopping the primary server (source).
- There are mainly two types Of replications format. One is statement Based Replication which replicates the entire SQL statement and the other is Row Based Replication where only the changed rows are replicated.
- Using Replication it is possible to perform Scale out operation on Database.
- Scalability refers to the capability to extend the Database and the load of the application queries across several MySQL Servers.

Use slide 17 to summarize the session. Tell the students pointers of the session. This will be a revision of the current session.

## Session 10: Partitioning

### Slide 2

#### Session Overview

- Define partitioning and its different types
- Describe how partitioning works
- List the benefits of partitioning
- Explain partitioning maintenance, pruning and selection
- Outline the restrictions and limitations of partitioning

#### Instruction(s) to the trainer:

Show slide 2 to the students and give them a quick overview of the current session and its goals. Inform students that the session will explain about partitioning and its different types. They will also learn the benefits and working of partitioning. The session also introduces them to partitioning maintenance, pruning, and selection along with its restrictions and limitations.

## What is Partitioning?

Partitioning is the process of dividing the rows of a table into separate tables in different locations.

In MySQL, Partitioning is used to split or partition the records of a table as rows into separate tables in different locations. Even though it is partitioned, it is treated as a single table.

Partitioning function is the rule that is set to accomplish the division of table data.

Partitioning is supported by MySQL only in InnoDB and NDB storage engines.

### Instruction(s) to the trainer:

Show slide 3 to the students and explain partitioning. In MySQL, partitioning is used to divide or partition a table's rows into separate tables in different places, but the table is still viewed as a single table. It distributes chunks of the table's data over a file system according to the rules defined by the user. A partitioning function is a rule that we used to divide the table data. The chosen function is based on the user-defined partitioning type and takes as an argument as user-supplied phrase. Depending on the type of partitioning employed, the user expression can be a column value or a function acting on column values.

**In-Class Question:** What is the difference between partition and index?

**Answer:** Indexes are used to speed the search of data within tables. Partitions provide segregation of the data at the HDFS level, creating sub-directories for each partition.

Partitioning allows the number of files read and the amount of data searched in a query to be limited.

## Working of Partitioning

The user must ensure that the look-up table goes to the correct partition or group of partition.

PARTITION BY divides the result set into partitions and further performs queries on it.

Look-up table can be defined as array of data that maps input values to output values. It contains the data that matches an ID in view.

### Instruction(s) to the trainer:

Show slide 4 and explain the working of partitioning with an example. The construct TABLE or ALTER TABLE statements in MySQL can be used to create a partition. The syntax for generating partitions using the CREATE TABLE command is as follows:

```
CREATE TABLE [IF NOT EXISTS] table_name  
(column_definitions)  
[table_options]  
[partition_options]
```

## Uses of Partitioning

Partitioning improves scalability and optimizes the performance.

It provides a mechanism for dividing data by usage pattern.

Due to partitioning, queries that access only a fraction of the data can run faster.

It stores huge data in one table which can be kept on a single disk or in file system.

It supports to keep more control to manage the data in the database.

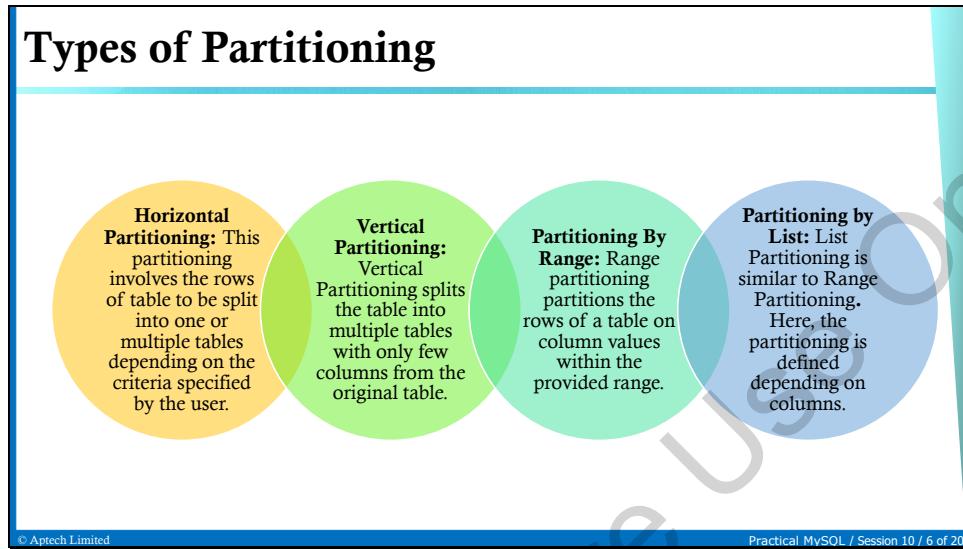
### Instruction(s) to the trainer:

Show slide 5 and discuss the uses of partitioning in MySQL.

Tell them that,

- It improves query performance. When a user runs a query on the table, it only looks at the parts of the table that would fulfil the given statement.
- Large amounts of data can be stored in a single table that can be stored on a single disc or file system partition.
- It gives you more control over your database's data.
- Partitioning gives a database administrator additional flexibility over how data is handled within the database.
- A DBA can simplify the way certain data operations are conducted by intelligently designing partitions. A DBA can, for example, drop selected partitions in a partitioned table while leaving the others alone.

## Slide 6



### Instruction(s) to the trainer:

Show slide 6 and explain the types of partitioning in MySQL.

- **Horizontal Partitioning:** Based on rationale, this partitioning splits a table's rows into numerous tables. Each table in horizontal partitioning has the same number of columns, but there is no need to preserve the same number of rows. It separates the table physically, yet it is logically viewed as a whole. MySQL is the only database that currently supports partitioning. For example: A table containing ten years' worth of historical invoice data might be partitioned into ten unique partitions, each containing a single year's worth of data, as an example of horizontal partitioning.
- **Vertical Partitioning:** The table is partitioned into numerous tables, each with fewer columns than the original table. The remaining columns are stored in a separate table. Currently, MySQL does not support partitioning in this way. For Example: A table with a number of very wide text or BLOB columns that are not frequently addressed could be split into two tables, with the most frequently referenced columns in one table and the rarely referenced text or BLOB data in the other.
- **MySQL RANGE Partitioning:** This partitioning technique allows to divide a table's rows into groups based on column values that fall within a certain range. The specified range is always

continuous, but should not overlap, and the ranges are defined using the VALUES LESS THAN operator. For Example: A DBA might build a partitioned table with three partitions, each containing data from the 1980s, 1990s, and anything after the year 2000.

- **MySQL LIST Partitioning:** In this partitioning, instead of a collection of contiguous values, the division is defined and selected, based on columns matching one of a set of discrete value lists. The PARTITION BY LIST(exp) clause is responsible for this. The exp is a column value or expression that returns an integer value. Each partition will be defined using the VALUES IN(value lists) statement. For Example: A DBA might build a partitioned table with three partitions based on the years 2004, 2005, and 2006.

**In-Class Question:** What is partitioning strategy?

**Answer:** The partitioning approach is a way of breaking down huge additions into smaller, easier-to-do additions. Partitioning includes dividing integers into hundreds, tens, and units before adding them together. Separately, the hundreds, tens, and units are added.

## Partitioning By Hash

In Hash partitioning, data distribution is done on the basis of predefined number of partitions.

In simple words, a table splits into that number of times as the user-defined value.

It distributes the data evenly in partitions. It can be performed by PARTITION BY HASH clause.

The number of partitions into which the table must be partitioned along with the column name to be hashed must be provided.

**Instruction(s) to the trainer:**

Show slide 7 and explain about Hash partitioning. Tell them that Hash partitioning is generally used to ensure an even distribution of data across a set number of partitions.

## Slide 8

### Partitioning By Key

The slide has a light blue header bar with the title 'Partitioning By Key'. Below the title are four colored boxes arranged horizontally, each containing a piece of text:

- Blue box: 'Key partitioning is identical to HASH partitioning. It also requires a user-defined expression. MySQL provides its own hashing function for partitioning using key.'
- Cyan box: 'MySQL uses its internal hashing function to perform PARTITION BY KEY clause with other storage engines.'
- Green box: 'In this case, a key is used instead of HASH. It takes only a list of zero or more column names as parameters.'
- Olive box: 'If no column is specified for partition, then the primary key is automatically considered as partitioning key.'

At the bottom left of the slide is a small copyright notice: '© Aptech Limited'. At the bottom right is a page number: 'Practical MySQL / Session 10 / 8 of 20'.

#### Instruction(s) to the trainer:

Show slide 8 and explain them about the partitioning done by the Key. It is similar to Hash partitioning, in which the user-specified expression is used and the MySQL server provides the hashing function for the key. While using different storage engines, the MySQL server uses its own internal hashing function, which is done with the PARTITION BY KEY clause. User will utilize KEY instead of HASH, which can only accept a list of zero or more column names. If the table has a PRIMARY KEY and no partitioning column is given, the primary key is utilized as the Partitioning key.

Following example explains more clearly:

```
CREATE TABLE AgentDetail (
    agent_id INT NOT NULL PRIMARY KEY,
    agent_name VARCHAR(40)
)
PARTITION BY KEY()
PARTITIONS 2;
```

If the table has unique key, but do not contain the primary key, then a UNIQUE\_KEY is used as a partition key.

```
CREATE TABLE AgentDetail (
    agent_id INT NOT NULL UNIQUE KEY,
```

```
agent_name VARCHAR(40)
)
PARTITION BY KEY()
PARTITIONS 2;
```

**Additional Information:**

Refer to following links for more information:

<https://dev.mysql.com/doc/refman/5.7/en/partitioning-key.html>  
<https://blog.toadworld.com/2017/07/06/table-partitioning-in-mysql>

## Partitioning by Column

In this, multiple columns are used as partitioning keys. It serves the purpose of partitioned rows and determines which partitions are to be validated for matching of rows. It is divided into two parts. Range and List Columns supports following data types:

Integer Types such as INT (INTEGER), MEDIUMINT, BIGINT, TINYINT, and SMALLINT.

String Types such as CHAR, VARCHAR, BINARY, and VARBINARY

Date types such as DATE and DATETIME.

### Instruction(s) to the trainer:

Show slide 9 and explain about partitioning by column. User can use many columns in partitioning keys using this partitioning.

These columns are used to separate the rows and determine which partition will be validated for matching rows. It is mostly separated into two categories:

- RANGE Columns Partitioning
- LIST Columns Partitioning

They allow for the definition of ranges or value lists using non-integer columns.

Following data types are supported:

- **All Integer Types:** TINYINT, SMALLINT, MEDIUMINT, INT (INTEGER), and BIGINT.
- **String Types:** CHAR, VARCHAR, BINARY, and VARBINARY.
- DATE and DATETIME data types.

## Range, List, and Sub Partitioning

Partitioning By Range specifies the partitions using ranges based on values of different columns.

CREATE TABLE part (Column1 INT, Column2 CHAR (5), Column3 INT, Column4 INT)

LIST Columns Partitioning involves single or multiple columns taken as partition keys.

Sub partition also called composite partitioning, splits the partitioned tables into further partitions.

### Instruction(s) to the trainer:

Show slide 10 and describe Range, List, and Sub Partitioning to students.

- **Range Partitioning:** Range partitioning is a sort of relational database partitioning in which the division is based on a preset range for a certain data column, such as IDs, dates, or basic values like currency. A partitioning key column is given a range, and when a data input falls within that range, it is assigned to that partition; otherwise, it is moved to another partition.

Range partitioning characteristics include:

- Each partition has an upper bound that is specific to it.
- Except for the first partition, each partition has a non-inclusive lower bound.

Following example demonstrate how to set up and use RANGE partitioning for MySQL performance:

```
CREATE TABLE customer_contract( cust_id INT NOT NULL, cust_fname  
VARCHAR(30), cust_lname VARCHAR(30), st_dt DATE NOT NULL DEFAULT  
'1970-01-01', end_dt DATE NOT NULL DEFAULT '9999-12-31',  
contract_code INT NOT NULL, contract_id INT NOT NULL ) PARTITION  
BY RANGE (contract_id) ( PARTITION p0 VALUES LESS THAN (50),
```

```
PARTITION p1 VALUES LESS THAN (100), PARTITION p2 VALUES LESS THAN  
(150), PARTITION p3 VALUES LESS THAN (200) );
```

- **LIST Partitioning:** The distinction between RANGE and LIST partitioning is that in LIST partitioning, each partition is based on a certain column's specified list of values. User can do it with PARTITION BY LIST (EXPR), where EXPR is the specified column for list partitioning. Following is an example of LIST partitioning:

```
CREATE TABLE students ( student_id INT NOT NULL, student_fname  
VARCHAR(30), student_lname VARCHAR(30), student_joined DATE NOT  
NULL DEFAULT '1970-01-01', student_separated DATE NOT NULL DEFAULT  
'9999-12-31', student_house INT, student_grade_id INT ) PARTITION  
BY LIST(student_grade_id) ( PARTITION P1 VALUES IN (1,2,3,4),  
PARTITION P2 VALUES IN (5,6,7), PARTITION P3 VALUES IN (8,9,10),  
PARTITION P4 VALUES IN (11,12) );
```

- **SUB Partitioning:** SUB Partitioning, also known as composite partitioning, is a method of partitioning a table by combining RANGE and HASH for better results.

Following example explains SUB Partitioning:

```
CREATE TABLE purchase (id INT, item VARCHAR(30), purchase_date  
DATE) PARTITION BY RANGE( YEAR(purchase_date) ) SUBPARTITION BY  
HASH( TO_DAYS(purchase_date) ) SUBPARTITIONS 2 ( PARTITION p0  
VALUES LESS THAN (2000), PARTITION p1 VALUES LESS THAN (2010),  
PARTITION p2 VALUES LESS THAN MAXVALUE );
```

## Benefits of Partitioning

Partitioning optimizes the query performance by scanning only portions of tables that satisfy the condition or criteria of the query.

Huge amounts of data can be stored in one table rather than on a single hard disk or file system and manage the file organizations as well.

Partitioning makes drive data more secure from malware attacks.

Sub partition also called composite partitioning, splits the partitioned tables into further partitions.

### Instruction(s) to the trainer:

Show slide 11 and list and explain all the benefits of partitioning.

Following are some of the benefits of using partitions:

- More data can be stored in a single table than stored on a single disc or file system partition.
- Deletion: Dropping a useless partition takes seconds, whereas running a traditional DELETE query in a huge table can take minutes.
- Partition Pruning: This feature allows you to exclude non-matching partitions and their data from a search, which speeds up queries.

## Partition Management

**Management of RANGE and LIST Partitions:** It is basically adding and dropping of partitions. RANGE and LIST partitions are used to create tables. Similarly, range and list partitions can be added and dropped.

**By using ALTER TABLE command:** In this management, users can perform various operations on existing partitions such as adding, dropping, redefining, merging, or splitting. All these operations can be performed using ALTER TABLE statement.

**Management of HASH and KEY Partitions:** Partition by HASH and Partition by KEY are similar to each other in making changes in partitioning set up.

### Instruction(s) to the trainer:

Show slide 12 and brief about partition management in MySQL. Tell students that this is the process of partitioning the data in different ways such as RANGE and LISTS Partitions, HASH and KEY partitions, sub-portioning, and so on.

In Management of RANGE and LIST Partitions, adding and dropping of partitions are handled. RANGE and LIST partitions are used to create tables. Similarly, range and list partitions can be added and dropped. Inform students that in this management, users can perform various operations on existing partitions such as adding, dropping, redefining, merging, or splitting. All these operations can be performed using ALTER TABLE statement.

The ALTER TABLE command with the DROP PARTITION option can be used to drop a partition from a table that is partitioned by either RANGE or LIST.

Inform students that partition by HASH and partition by KEY are similar to each other in making changes in partitioning set up. ALTER TABLE statement is used for Management of Hash and Key partitions.

### Additional Information:

Refer to following links for more information:

<https://dev.mysql.com/doc/refman/8.0/en/partitioning-management.html>

<https://dev.mysql.com/doc/refman/5.7/en/partitioning-management.html>

## Partition Maintenance

The diagram consists of five colored boxes arranged in two rows. The top row contains three boxes: an orange box on the left, a grey box in the middle, and a yellow box on the right. The bottom row contains two boxes: a blue box on the left and a green box on the right. Each box contains a title and a brief description.

- Rebuilding Partitions:** The process of partition maintenance includes multiple things such as rebuilding partitions, optimizing the partitions so on.
- Optimizing Partitions:** The partitioned data can be defragmented using optimizing partitions.
- Analyzing Partitions:** Partitions can be analyzed by using SQL Statements such as CHECK TABLE.
- Repairing Partitions:** Partitions which are corrupted during partitions can be repaired using REPAIR PARTITION command.
- Checking of Partitions:** Partitions can contain errors that require checking.

© Aptech Limited Practical MySQL / Session 10 / 13 of 20

### Instruction(s) to the trainer:

Using slide 13, brief about partition maintenance in MYSQL. SQL statements designed for this purpose can be used to perform a variety of table and partition maintenance operations on partitioned tables.

The statements CHECK TABLE, OPTIMIZE TABLE, ANALYZE TABLE, and REPAIR TABLE, which are supported for partitioned tables, can be used to maintain partitioned tables.

User can utilize a number of ALTER TABLE extensions to perform operations of this type directly on one or more partitions, as specified in the following list:

- **Rebuilding partitions:** Rebuilds the partition; this has the same effect as deleting and then reinserting all of the partition's entries. This can be beneficial in terms of defragmentation.  
Syntax: `ALTER TABLE t1 REBUILD PARTITION p0, p1;`
- **Optimizing Partitions:** User can use `ALTER TABLE... OPTIMIZE PARTITION` to reclaim any unused space and defragment a partitioned table with variable-length rows (that is, having VARCHAR, BLOB, or TEXT columns) if you have deleted a large number of rows from a partition or if you have made many changes to a partitioned table with variable-length rows.  
Syntax: `ALTER TABLE t1 OPTIMIZE PARTITION p0, p1;`

- **Analyzing partitions:** This reads and stores the key distributions for partitions. For partitioned tables, ANALYZE TABLE is supported, and ALTER TABLE... To examine one or more partitions, use the ANALYZE PARTITION command.

Syntax: ALTER TABLE t1 ANALYZE PARTITION p3;

- **Repairing partitions:** This restores partitions that have been corrupted.

Syntax: ALTER TABLE t1 REPAIR PARTITION p0,p1;

- **Checking partitions:** The user can use CHECK TABLE to check partitions for problems in the same manner that you can with non-partitioned tables.

Syntax: ALTER TABLE trb3 CHECK PARTITION p1;

**In-Class Question:** Does DB partitioning improve performance?

**Answer:** Partitioning can make managing huge tables easier while also improving scalability and availability. Improved query performance might also be a side effect of partitioning.

## Partition Pruning

Partition pruning is basically the technique of optimizing. It mentions that partitions must not be scanned if there are no matching values.

It removes or cuts undesirable partitions.

The concept of data warehousing uses this as an essential performance feature.

The optimizer checks for useless partitions using WHERE and FROM clause, which can further be removed.

In this process, scanning certain partitions not having any matching conditions can be avoided.

### Instruction(s) to the trainer:

Show slide 14 and explain about partition pruning. The partition pruning optimization is based on a relatively simple notion that can be summarized as 'Do not scan partitions when no matching values are possible.'

Assume following line creates a partitioned table t1:

```
CREATE TABLE t1 ( fname VARCHAR(50) NOT NULL, lname VARCHAR(50)  
NOT NULL, region_code TINYINT UNSIGNED NOT NULL, dob DATE NOT  
NULL ) PARTITION BY RANGE( region_code ) ( PARTITION p0 VALUES  
LESS THAN (64), PARTITION p1 VALUES LESS THAN (128), PARTITION  
p2 VALUES LESS THAN (192), PARTITION p3 VALUES LESS THAN MAXVALUE  
);
```

Assume that following statement is used to get results from a SELECT statement:

```
SELECT fname, lname, region_code, dob FROM t1 WHERE region_code >  
125 AND region_code < 130;
```

It is clear that none of the rows that should be returned are in partitions p0 or p3; therefore, user only need to look in partitions p1 and p2 to discover matching rows. It is feasible to locate matched rows in considerably less time and effort by narrowing the search rather than scanning

all partitions in the table. Pruning is the process of ‘cutting away’ superfluous partitions. When the optimizer can use partition pruning to execute this query, it can be an order of magnitude faster than executing the same query against a non-partitioned table with identical column definitions and data.

```
partition_column = constant  
partition_column IN (constant1, constant2, ..., constantN)
```

The optimizer simply evaluates the partitioning equation for the given value, identifies which partition holds that value, and scans only that partition in the first case. In many circumstances, the equal sign can be replaced with another arithmetic comparison, such as,  $>=$ ,  $\geq$ , and  $>$ . Partition pruning can be used in some queries that employ BETWEEN in the WHERE clause.

## Partition Selection

It is similar to the concept of partition pruning. In this process, only specific partitions are checked for matches, but it may differ in two key aspects that are as follows:

Partitions are checked which are specified by issuer of the statement. It is different from pruning as it is done automatically in pruning.

Partition selection is done for all types of queries and a number of DML statements, whereas, in pruning it is only done for queries.

Following are some SQL statements that support partition selection:



### Instruction(s) to the trainer:

Show slide 15 and explain them about partition selection. SQL statements supporting explicit partition selection are listed here:

- SELECT
- DELETE
- LOAD DATA
- LOAD XML
- INSERT
- UPDATE
- REPLACE

A partition option is used to implement explicit partition selection. This option uses following syntax for all supported statements:

```
PARTITION (partition_names) partition_names: partition_name, ...
```

The name of the table to which the partition or partitions belong is always followed by this option. Partition name is a list of partitions or sub-partitions to utilize, are separated by commas. Each name in this list must be the name of an existing partition or sub-partition of the provided table; if any of the partitions or sub-partitions are not discovered, the statement fails. Partitions and sub-partitions named in partition names can be listed in whatever order they want, and they can overlap.

## Restrictions and Limitations on Partitioning

While doing partitions it is important to understand their limitations and restrictions so that the process is successfully completed. Limitations on partitioning are as follows:

8192 Partitions and sub partitions in MySQL are possible. It is applicable to all versions of MySQL.

There is a limit of 1024 partitions per table as well.

Table partitioning can be performed if a storage engine supports it.

All partitions must be from the same storage engine.

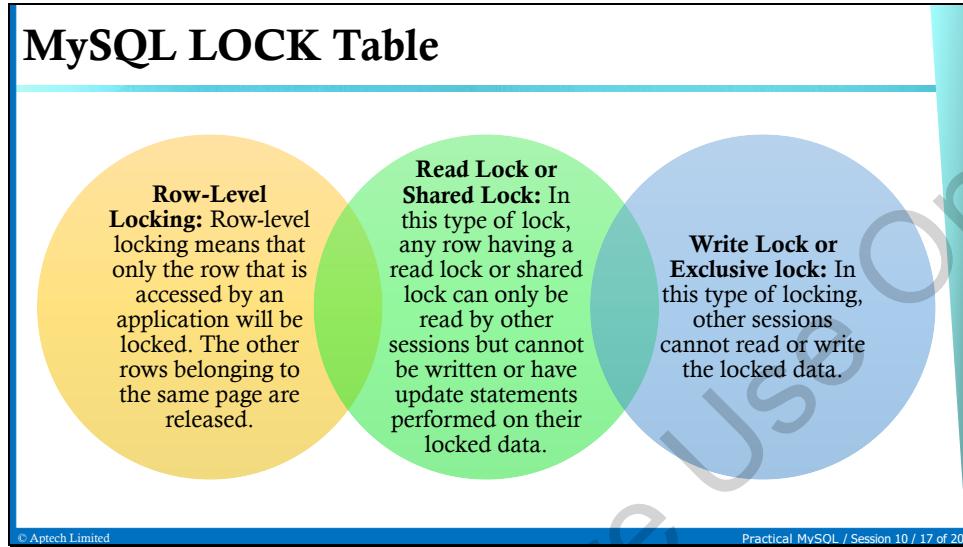
Partition table should not contain any foreign key.

### Instruction(s) to the trainer:

Show slide 16 and discuss about restrictions and limitations while using partitioning. List the limitations given on slide and explain each of them in detail.

List the following restrictions and inform that these restrictions are important to analyze so that the process is successfully completed:

- Stored routines or plugins are not allowed in partitioning expression.
- Variables that have been declared or variables that have been entered by the user.
- Operators for arithmetic and logic. In partitioning expressions, the arithmetic operators +, -, and \* are allowed. However, except in the case of [LINEAR] KEY partitioning, the result must be an integer value or NULL.
- The bit operators |, &, <<, >> are not permitted in partitioning expressions.
- Tables with user-defined partitioning do not keep the SQL mode that was in effect when they were created. The outcomes of several MySQL functions and operators can vary depending on the server SQL mode.



**Instruction(s) to the trainer:**

Show slide 17 and explain MySQL LOCK Table. Inform that,

- Client sessions can deliberately acquire table locks to cooperate with other sessions for table access or to prohibit other sessions from altering tables during periods when a session requires exclusive access to them.
- Only a single session can acquire or release locks.
- A session cannot get locks for another session or release locks that another session has.
- Locks can be used to simulate transactions or to improve the speed with which tables are updated.
- LOCK TABLES obtains table locks for the current client session explicitly.
- For base tables and views, table locks can be obtained.
- For each object to be locked, you must have the LOCK TABLES and SELECT privileges.
- LOCK TABLES adds all base tables used in the view to the set of tables to be locked and locks them automatically when the view is locked.
- LOCK TABLES verifies that the view definer (for SQL SECURITY DEFINER views) or invoker (for all views) has the appropriate privileges on the tables underlying any locked view.
- If user use LOCK TABLES to lock a table explicitly, any trigger tables are likewise locked implicitly, as discussed in LOCK TABLES and Triggers.

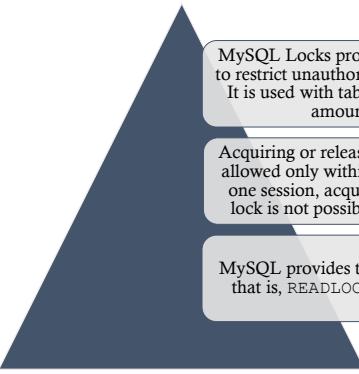
**Additional Information:**

Refer to following links for more information:

<https://dev.mysql.com/doc/refman/8.0/en/lock-tables.html#:~:text=MySQL%20enables%20client%20sessions%20to,release%20locks%20only%20for%20itself>.

<https://dev.mysql.com/doc/refman/8.0/en/table-locking.html>

## MySQL Locks



MySQL Locks provide a security system to restrict unauthorized access to its data. It is used with tables, which have huge amount of data.

Acquiring or releasing locks on a table is allowed only within the session. During one session, acquiring or releasing the lock is not possible for other sessions.

MySQL provides two types of locks and that is, READLOCK and WRITELOCK.

© Aptech Limited

Practical MySQL / Session 10 / 18 of 20

**Instruction(s) to the trainer:**

Show slide 18 and explain them about MySQL Locks. A table lock is a mechanism that prevents unauthorized access to the data stored in the table. MySQL allows a client session to acquire a table lock in order to share the table's contents with other sessions. MySQL also supports table locking, which prevents unauthorized changes to the same table within a specified time period. In MySQL, a session can only acquire or release locks on the table for itself. As a result, one session will not be able to acquire or release table locks for other sessions. It is worth noting that table locking requires TABLE LOCK and SELECT rights. In MySQL, table locking is primarily used to tackle concurrency issues. It will be utilized to complete a transaction.

**In-Class Question:** Does MySQL lock on select?

**Answer:** On InnoDB tables, SELECTs do not generally perform any locking that you worry about. Selects do not lock things when the transaction isolation level is set to the default.

## READ AND WRITE LOCK

The diagram consists of three colored boxes arranged vertically. A large grey arrow points from left to right above the first two boxes, and a large orange arrow points from left to right below the second box. The first box is grey and contains the text "READ LOCK". The second box is orange and contains the text "WRITE LOCK". The third box is red and contains the text "READ VS. WRITE LOCKS". To the right of each box is a list of bullet points describing the lock type.

READ LOCK	• More than one session can acquire a <b>READ LOCK</b> on a single table.
WRITE LOCK	• A session holding the write lock on a table is allowed to read data from the table or write data into the table.
READ VS. WRITE LOCKS	• <b>READLOCKS</b> are shared and hence, they prevent a write lock to become acquired. <b>WRITELOCKS</b> are exclusive locks they allow you to have many read statement, but only one write statement is allowed.

© Aptech Limited Practical MySQL / Session 10 / 19 of 20

### Instruction(s) to the trainer:

Show slide 19 and tell students that there are two types of table locks provided by MySQL.

**READ LOCK:** If a table has a READ LOCK by some session, then other sessions cannot perform write operations in that table. Write operation cannot be performed on a table by sessions not acquiring a READ LOCK. To perform a write operation, the READ LOCK must be released. If a session is terminated by any user or abnormally terminated, MySQL releases all types of locks. The table can be locked for read operations by executing following command:

```
LOCK TABLE Customer READ;
```

**WRITE LOCK:** This lock allows a user to read and write into a table at the same time. It is worth noting that MySQL's default storage engine is InnoDB. Because MySQL uses row-level locking for InnoDB tables, the InnoDB storage engine does not require explicit table locking. As a result, several read and write operations can be done on the same table at the same time without having to wait for each other. Table locking is used by all other storage engines except MySQL.

**Additional Information:**

Refer to following links for more information:

<https://www.geeksforgeeks.org/readwritelock-interface-in-java/>

<https://jenkov.com/tutorials/java-concurrency/read-write-locks.html>

## Summary

- Partition allows users to divide portions of individual tables in a file system or partition according to the rules.
- Vertical Partitioning splits the table into multiple tables with very few columns from the original table.
- Sub partitioning splits each partition in a partition table.
- Rebuilding Partitions is a part of Partition Maintenance.
- Users can perform operations such as CHECK TABLE, REPAIR TABLE, OPTIMIZE PARTITION, REBUILD PARTITION, REPAIR PARTITION, and so on for partition maintenance.
- Partitions are analyzed using SQL statements such as CHECK TABLE.
- Locking mechanism is associated with tables that restrict unauthorized access of data inside the table.

Use slide 20 to summarize the session. End the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

## Session 11: Performance Optimization in MySQL

Slide 2

### Session Overview

- Explain optimizing queries with MySQL query optimization guidelines
- Outline the process of optimizing workload
- Define how to monitor fundamental resources
- Describe restructuring queries
- Describe optimizing subqueries
- Identify scalability problems
- Explain queue cache
- Describe and explain automating of configuration
- Explain specific types of queries

#### Instruction(s) to the trainer:

Show slide 2 and give students a brief overview of the current session and the session objectives. Inform students that this session explains the concept of Performance Optimization in MySQL, describe how to restructure queries, optimizing subqueries, and how to identify scalability problems in MySQL. Inform that they will learn about Queue cache, automating of configuration, and specific type of queries.

## Optimizing Queries with MySQL Query Optimization Guidelines

Optimization refers to the performance increase of any given system. Currently, many applications are database-driven.

If the SQL Queries are not written properly or if they are improperly designed, then it degrades the database application performance.

In such cases, planning for optimizations or troubleshooting the queries and configurations after understanding the crux of the problem is essential.

### Instruction(s) to the trainer:

Show slide 3 and give students an explanation on Optimizing Query practices with MySQL Query.

Tell students that, for optimizing query they should begin by double-checking all WHERE, JOIN, ORDER BY, and GROUP BY clause predicates are indexable. WebSphere Commerce prioritizes indexing predicates to increase SQL performance. As poor indexing of SQL queries can lead to table scans, which can cause locking and other issues.

As a result, indexing all predicate columns for MySQL query optimization can benefit the database. Functions should not be included in predicates. The database does not use an index if a column has a function predefined in it.

**In-Class Question:** Why is query optimization needed?

**Answer:** Query optimization allows more efficient query processing. It has a low cost per query. The database is less stressed as a result. It ensures the system's high performance.

## Benefits of MySQL Database Queries Optimization

- Optimization improves the speed at which queries are executed and aids in managing the resources.
- Optimized queries run faster by taking less time and less computing power.
- Optimization reduces the cost or price of hardware.
- Optimization allows the server to efficiently run by consuming less power and less memory.

© Aptech Limited Practical MySQL / Session 11 / 4 of 15

**Instruction(s) to the trainer:**

Show slide 4 and explain the benefits of using MySQL database queries in Optimization process.

- User prioritizes speed and efficiency. Queries that have been optimized can run faster and use less computational power.
- For starters, it gives the faster results, making the application appear to be more fast.
- As each request takes less time than un-optimized queries, the system can handle more enquiries in the same amount of time.
- Query optimization decreases the amount of impairment on the server's hardware (for example, disc drives) and allows it to run more efficiently (for example, lower power consumption and less memory usage).

**In-Class Question:** What affects SQL query performance?

**Answer:** Data volume and transaction concurrency have an impact on query performance. Executing the same query on a table with millions of records takes longer than doing the same operation on a table with just a few thousand records. SQL Server performance can be harmed by a large number of concurrent transactions.

## Optimizing Queries [1-2]

**By using EXPLAIN statement:** The EXPLAIN statement shows how MySQL executes a statement. It provides information on what the statement would do, how it would be processed, how the table is joined, and so on.

**By normalizing the tables:** Normalization keeps data non-redundant and avoids repetition of lengthy values such as name, categories, and contact details.

**By using the most appropriate data types:** Proper data types should be used in columns so that the columns can take the values that belong only to the specified datatype thereby minimizing the usage of space.

**By avoiding Null values:** Using the NOT NULL constraint for the columns ensures that the column can never have NULL values and ensures that the index mechanism works.

**By using an index for required columns:** Data retrieval time can be reduced by keeping the index on a column that is always in use. It will be easy to find the specific value faster by keeping indexes on the columns.

### Instruction(s) to the trainer:

Show slide 5 and explain students about Optimizing Queries, Index Columns used in the WHERE, ORDER BY, and GROUP BY clauses.

Consider following SQL query, which is being executed from a database with 500 rows and no index:

```
<mysql> select customer_id, customer_name from customers where  
customer_id='665247';
```

Here, the query will cause MySQL to perform a full table scan (from beginning to end) to retrieve the record user is looking for.

Fortunately, MySQL offers a specific EXPLAIN statement that you may use to evaluate your queries alongside SELECT, DELETE, INSERT, REPLACE, AND UPDATE commands.

MySQL shows information from the optimizer about the anticipated execution plan when user attaches the query before a SQL statement.

**Normalizing Tables:** First, even if it means certain trade-offs, standardize all database tables. For example, if user is constructing two databases to store customer data and orders, user should

refer to the customer on the orders table by their customer id rather than repeating the client's name. User's database will bloat when delayed.

**Use Optimal Data Types:** Among the data types supported by MySQL are integer, float, double, date, date-time, varchar, and text. Inform students that when it comes to the table design, remember that 'shorter is usually better.' For example, if the user is creating a system user's table with fewer than 100 users, the user should utilize the 'TINYINT' data type for the 'user id' field because it can handle any value from -128 to 128. Also, if a field expects a date value (for example, sales order date), a date-time data type is ideal since, it eliminates the need to run expensive methods to convert the field to date when obtaining records using SQL.

**Avoid Null Values:** The absence of any value in a column is known as null. This type of number should be avoided as much as possible because it can affect your database findings. For example, if the user wishes to retrieve the sum of all orders in a database but one of the order records has a null amount unless use MySQL's 'ifnull' statement returns an alternate value if a record is null, the intended result may misbehave. If records do not have to have a mandatory value for that particular column/field, the user may need to provide a default value for that field.

**Avoid Too Many Columns:** Wide tables can be very costly and take a lot of CPU time to process. If at all feasible, do not exceed a hundred unless business logic clearly requires it. Inform students to consider dividing the table into logical components rather than one large table. For example, if the user is constructing a customer database and realize that a customer can have several addresses, it is better to create a separate record for addresses that refer back to the customers table via the 'customer id' field.

#### **Additional Links:**

Refer to following links for more information:

<https://www.cloudways.com/blog/mysql-performance-tuning/>

<https://dzone.com/articles/how-to-optimize-mysql-queries-for-speed-and-performance>

## Optimizing Queries [2-2]

**By using wildcard characters:** Wildcard characters are used in conjunction with MySQL LIKE Operator.

**By using LIMIT:** Most often, only a specified number of rows is required from a result set.

**By Using MySQL Query Caching:** Query caching provides database caching functionality that stores the result set.

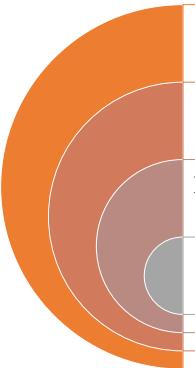
**By converting OUTER JOIN to INNER JOIN:** Convert the OUTERJOIN to INNER JOIN to enhance performance.

### Instruction(s) to the trainer:

Show slide 6 and explain students about Optimizing Queries.

- Leading Wildcards: When a query has a leading wildcard, MySQL is unable to use indexes.
- Optimize Joins: In Join statements, use fewer tables if possible. It is possible that a SQL query with a bad pattern and a lot of joins would not work. As a general guideline, each query should have at least a dozen joins.
- MySQL Query Caching: If a Website or program conducts a lot of select queries, the user should use MySQL query caching. When reading operations are performed, this will improve performance. The technology works by caching the select query as well as the data set. Because the query is fetched from memory if it is executed more than once, it runs faster. However, if the application often refreshes the table, any cached queries and result sets will be invalidated.

## Understanding the Workload



The workload of the MySQL database can be explained by understanding what is currently happening with the database.

In absence of accurate baseline and workload profiling, it is very difficult to identify problems and determine how to use potential optimization measures.

Looking at the workload, the user can find out which query is more expensive and is affecting the performance of the database.

The user can profile or baseline the database performance by using tools such as To do list, Buffer, and so on.

© Aptech Limited Practical MySQL / Session 11 / 7 of 15

**Instruction(s) to the trainer:**

Show slide 7 and give students a brief introduction to the workload in MySQL. Similar to other relational databases, it is a sophisticated database that can come to a halt at any time, leaving applications in the lurch and putting the business at risk. The truth is that most MySQL speed issues are caused by common errors. It is critical to eradicate these errors, which are typically masked by nuance in the workload or a configuration trap, to guarantee that MySQL server runs at full speed and provides steady and consistent performance.

The best approach to figure out how a server spends its time is to look at its workload profile. Users can reveal the most expensive queries for additional tweaking by evaluating their workload. When user runs a query against the server, the only thing that matters is how quickly it completes, time is the most significant metric. The easiest method to profile workload is to use a tool such as MySQL Enterprise Monitor's query analyzer or Percona Toolkit's pt-query-digest. These tools capture server queries and return a table of tasks organized by decreasing response time, bringing the most expensive and time-consuming processes to the top so that users can understand where to focus their efforts.

## Execution Time

- Execution time mainly refers to the time taken for the database to respond to any query.
- Query execution time can be determined by executing the show profiles query.
- Slow queries are logged so that they can be examined later for optimization purposes.
- The MAX EXECUTION TIME (N) hint defines a statement execution timeout of N milliseconds.

© Aptech Limited Practical MySQL / Session 11 / 8 of 15

### Instruction(s) to the trainer:

Show slide 8 and explain students about Execution Time in MySQL.

Only SELECT statements are allowed to use the MAX EXECUTION TIME hint. It sets a timeout value of N milliseconds on how long a statement can run before being terminated by the server.

#### MAX\_EXECUTION\_TIME (N)

MAX\_EXECUTION\_TIME (N) hint specifies an N-millisecond statement execution timeout. If this option is not specified or N is 0, the statement timeout determined by the system variable max execution time is used.

MAX\_EXECUTION\_TIME tip is useful in the following situations:

- MAX\_EXECUTION\_TIME applies to the entire statement and must appear after the first SELECT keyword in statements containing multiple SELECT keywords, such as unions or statements with subqueries.
- It applies to SELECT statements that are read-only.
- Statements that invoke a stored function that affects data as a side effect are not read only.
- It is ignored when it comes to SELECT statements in stored programmes.

## Slide 9

# Monitoring Fundamental Resources

Fundamental resources play a vital role in the optimization of a database and monitoring those resources is a must.

Even if one of these resources is poor or weak, the database performance will suffer.

Organizations should select servers with fast CPU, disk space, and memory.

© Aprech Limited

Practical MySQL / Session 11 / 9 of 15

### Instruction(s) to the trainer:

Show slide 9 and explain students about monitoring fundamental resources in MySQL. Inform students that major technology giants such as Microsoft, Google, and Amazon offer systems that are completely configurable. These systems are called virtual machines. The CPU, memory, disk space, and all facets of the system are changeable even after deployment. Instead of spending on hardware, most companies around the world have started relying on virtual machines for their hardware requirement.

### Additional Links:

Refer to following links for more information:

<https://www.metricfire.com/blog/a-modern-guide-to-mysql-performance-monitoring/>  
<https://www.mysql.com/products/enterprise/monitor-features.html>

## Restructuring Queries

Restructuring queries is another prominent way to achieve query optimization.

**Too many rows:** A common mistake made while writing a query is that it retrieves data.

**Too many columns:** Obtaining unnecessary columns can also affect performance and is considered a common mistake.

**Too much data analysis:** Data analysis can maximize the time for execution.

### Instruction(s) to the trainer:

Show slide 10 and explain students about restructuring queries in MySQL. As user has the ability to restructure problematic queries, first priority should be to find a different way to get the desired outcome. User can convert the query into its equivalent form while processing it, bearing in mind the internal effect in the MySQL server. One option to make in query design is whether to prioritize one complex query over multiple simple queries or vice versa. The traditional database design strategy is to perform as many tasks as feasible with as few queries as possible. The rationale for this is that creating a database connection with a single large/complex query is more cost effective. Network consumption, query processing/optimization, and resource utilization are all advantages of cost reduction in favor of sophisticated queries.

### Additional Information:

Refer to following links for more information:

<https://stackoverflow.com/questions/37400242/need-help-restructuring-a-query>

<https://dev.mysql.com/doc/refman/5.7/en/key-cache-restructuring.html>

## Chopping Queries

Chopping the query means slicing up a big or complex query and running it in smaller chunks so that it affects only a smaller quantity of rows each time..

Chopping the queries is a concept that comes into the picture when the time spent fetching the result set keeps increasing.

There are two types of queries. The very first one is complex queries, and the second one is Simple queries.

Complex queries are the major ones that are mainly to be chopped to get good performance.

### Instruction(s) to the trainer:

Show slide 11 and explain Chopping Queries to the students. Chopping a query refers to slicing up a large or complex query and running it in smaller portions so that only a limited number of rows are affected each time. A larger or more complicated query can produce the necessary results, but it may lock the rows for longer periods of time. The optimizer of the database engine determines how a query must be sliced. While certain database engines can handle longer queries, it is usually advisable to break them up into smaller parts for optimum optimization. When the time spent fetching the result set keeps increasing, the concept of chopping the queries comes into play. There are two different kinds of questions. Complex questions are the first, while simple questions are the second.

## Identifying Scalability Problems



MySQL was designed for a single node system and not for modern data centers.



When considering scalability, scale or enhancement is not the only thing to take care of. What to do with the data and how to keep it while scaling should also be given a thought.



Scalability can be upwards or downwards. In other words, it can mean an increase or decrease in hardware.



In managerial terms, scalability refers to an application that can perform well even though it grows. This must be accounted for as well.

### Instruction(s) to the trainer:

Show slide 12 and explain how to identify scalability problems in MySQL. A monolithic MySQL database with scalability problems can be turned into a distributed MySQL database while maintaining its relational principles utilizing a declarative, policy-based data distribution technique. The current database structure and commands of the program are in line with a well-thought-out data distribution policy. Related data from other tables must be located and aggregated in order to stay localized in a single database instance. ‘Data that play together should stay together,’ to put it another way (and to paraphrase a well-known proverb). Our goal is to be able to effectively complete ‘reads’ and ‘writes’ using only data from one distributed database instance (or shared).

## Query Cache and Automating Configuration

Query Caches is a technique that saves the text from the `SELECT` statement and its resultant output in cache. In the occasion of an identical statement being executed at a later point, the resultant is received from the query cache and displayed to the client or user. This technique works with the `SELECT` statement and is utilized when the table data is seldom changed.

Configuration mainly points to the arrangement or set-up of any system. Automating this process without regular human intervention is what automatic configuration refers to.

### Instruction(s) to the trainer:

Show slide 13 and explain about query cache and automating configuration.

- The query cache keeps track of the text of `SELECT` statements as well as the results that were delivered to the client. If the server receives a similar command later, it obtains the results from the query cache rather than parsing and executing it again. Because the query cache is shared between sessions, a result set generated by one client might be transmitted in response to a query performed by another. The query cache can be useful in situations when tables do not update frequently and the server receives a large number of identical queries. This is a common occurrence for many Web servers that generate a large number of dynamic pages from database content. The query cache does not provide any results.

When `innodb_dedicated_server` is enabled, InnoDB automatically configures following variables:

`innodb_buffer_pool_size`  
`innodb_log_file_size`  
`innodb_log_files_in_group`(as of MySQL 8.0.14)  
`innodb_flush_method`

Only consider enabling `innodb_dedicated_server`, if the MySQL instance resides on a dedicated server where it can use all available system resources. For example, consider enabling if the user

runs MySQL Server in a Docker container or dedicated VM that only runs MySQL. Enabling `innodb_dedicated_server` is not recommended if the MySQL instance shares system resources with other applications.

The information that follows describes how each variable is automatically configured. `innodb_buffer_pool_size` is configured according to the amount of memory detected on the server.

## Optimizing Specific Types of Queries

**Optimizing COUNT () Queries:** `COUNT ()` queries are one of the most frequently used queries in MySQL.

**Simple Optimizations:** Simple Optimization in MySQL is the procedure of selecting an efficient means of executing an SQL statement.

**Optimizing JOIN Queries:** `JOIN` is a concept where a result set can be retrieved from more than one table based on specific conditions. Imprudent use of the `JOIN` statement can result in affecting the data retrieval performance.

**Optimizing LIMIT and OFFSET:** `LIMIT` is mainly used to obtain the specified number of rows and it optimizes the SQL Queries depending on the count of rows to be fetched.

### Instruction(s) to the trainer:

Show slide 14 and explain about optimizing specific types of queries. Tell students that the practice of retrieving the result set of certain queries in less time is known as query optimization. The two ways are as follows:

1. Cost-based Optimization (Physical)
2. Cost-based Optimization (Logical)

- `COUNT ()` queries are one of the most commonly used MySQL queries. Though there is no guaranteed strategy for optimising `COUNT ()`, using it without any other clauses can improve efficiency.
- In MySQL, simple optimization is the process of choosing the most efficient way to execute a SQL statement. When it comes to running complex SQL statements, optimization is crucial. The goals of query optimization are to reduce system resources and provide the proper set to the user at a faster rate.
- A `JOIN` is a notion in which a result set can be fetched from many tables based on certain criteria. Using the `JOIN` statement incorrectly can have an impact on data retrieval performance. When several tables and conditions are used in the statement, data retrieval becomes very slow, and it is worth noting that only `LEFT JOIN` is the fastest of the join types.
- `LIMIT` is mostly used to get a specific number of rows and optimizes SQL Queries based on the number of rows to be fetched.

## Summary

- An application executes multiple queries on the database concurrently and if the SQL Queries are improperly written or if they are badly designed, then it degrades the database application performance.
- Optimization improves the pace at which queries are executed and manages the resources.
- MySQL has a ceiling of 4,096 columns per table.
- In absence of accurate baseline and workload profiling, it is very difficult to discover problems and utilize potential optimization measures.
- Query Caches is a technique that stores the text of the SELECT statement and its result and uses it when a similar statement comes by.
- The procedure of selecting an efficient means of executing an SQL statement in MySQL is called Simple Optimization.

**Instruction(s) to the trainer:**

Use slide 15 to summarize the session. End the session, with a brief summary of what has been taught in the session. Tell the students pointers of the session. This will be a revision of the current session.

## Session 12: Concept of Storage System and Management

Slide 2

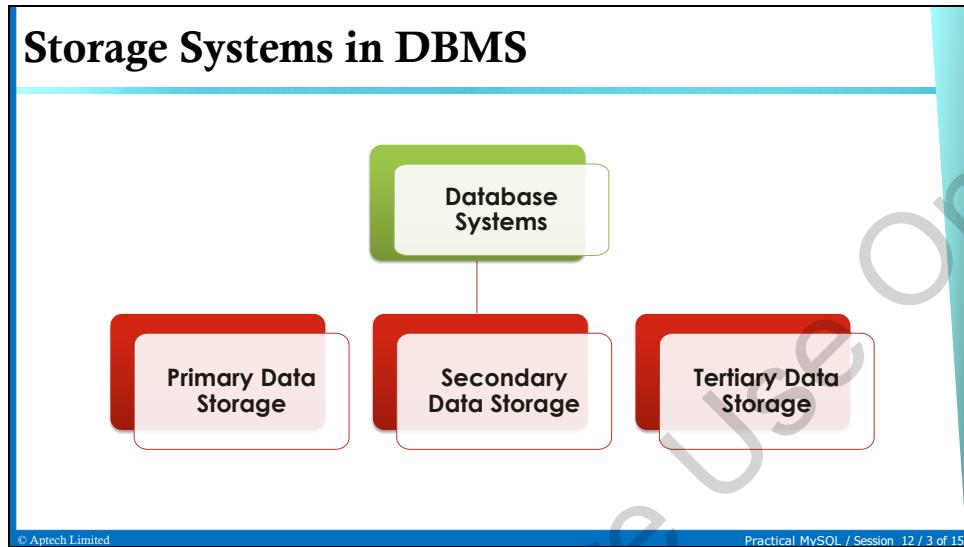
### Session Overview

- Outline the types of data storage
- Explain different types of storage
- Describe demand paging and thrashing
- Explain process management in DBMS
- Define paging and segmentation

#### Instruction(s) to the trainer:

Show slide 2 and give students a brief overview of the current session in the form of session objectives. Inform students that the session begins with an overview of storage systems and memory management in database management systems. The session further explains the types of data storage and discusses virtual memory, demand paging, and thrashing. It also discusses the concept of process management, paging, and segmentation.

## Slide 3



### In-Class Explanations

#### Instruction(s) to the trainer:

Use slide 3 to explain students the concept of data storage in DBMS and the types of data storage systems. Tell them that managing storage systems improves the performance of data storage resources.

Explain to students that based on their speed and accessibility, there are three types of storage systems-primary data storage, secondary data storage, and tertiary data storage.

#### Additional Information:

Refer to following links for more information:

[https://www.tutorialspoint.com/dbms/dbms\\_storage\\_system.htm](https://www.tutorialspoint.com/dbms/dbms_storage_system.htm)

<https://www.seagate.com/in/en/products/storage/data-storage-systems>

## Slide 4

# Primary Data Storage

<b>Main Memory</b> <ul style="list-style-type: none"><li>Main memory is used to operate the data available by the storage medium. It handles the instructions of the machine.</li><li>This is capable to handle gigabytes of data. However, it is small to carry the large database.</li></ul>	<b>Cache</b> <ul style="list-style-type: none"><li>Cache is the costliest and smallest storage media maintained by computer hardware.</li></ul>
--	---

© Aptech Limited Practical MySQL / Session 12 / 4 of 15

### Instruction(s) to the trainer:

Show slide 4 and explain students that primary storage or memory allows the central processing unit of the server to access the data without using other devices. Primary storage is a kind of volatile storage that can access data much faster than any other storage.

Inform students that though primary storage is fast, it is not a permanent storage and data can be lost, if not saved, in case of power cuts or system failures.

Move on to explain two kinds of primary storage in detail: Main memory and Cache.

### Additional Information:

Refer to following links for more information:

<https://www.seagate.com/in/en/products/storage/data-storage-systems>

<https://www.cdw.com/.../en/articles/datacenter/what-is-data-storage.html>

<https://www.educba.com/data-storage-in-database>

## Secondary Data Storage

### Memory devices or Flash Memory

It is used to store data in USB keys that can be plugged into the USB. It is possible to retrieve the data lost during a power cut or system crash if it is stored in USB.

### Magnetic disk

It is an online storage media that is used to store data for a longer period. In this type of storage, the user can store the entire database.

### Instruction(s) to the trainer:

Show slide 5 and discuss secondary storage systems. As these systems allow users to store data permanently, they are also called online storage or non-volatile storage. Tell students that the data stored in secondary storage system does not get lost in case of power failures or system crash.

Different types of secondary storage include memory devices, Flash memory, and magnetic disk drives.

**In-Class Question:** Name a secondary storage device that can be used to store an entire database.

**Answer:** Magnetic disk

## Tertiary Data Storage

### Optical Storage

It can store data in gigabytes or megabytes. For example, a CD can store 600 megabytes of data. Similarly, a DVD can store 5.0 or 7.5 gigabytes of data.

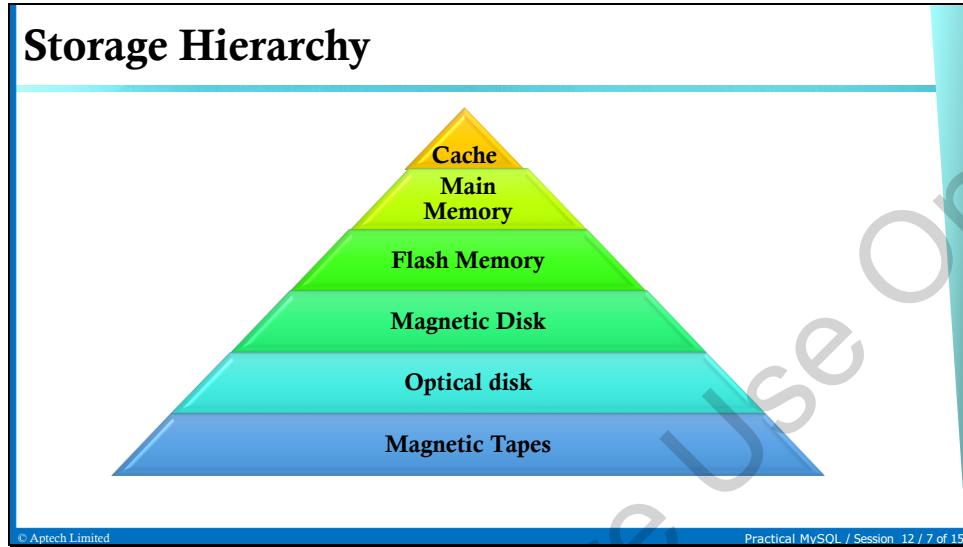
### Tape Storage

It is the cheapest storage media that is slower in processing. It can be used to archive or back up the data.

### Instruction(s) to the trainer:

Slide 6 introduces the concept of tertiary storage systems. It is an offline storage system that can be used to store a huge amount of data. Tell students that as this storage is not a part of the current computer system, it is rather slow in fetching and processing data. Some types of tertiary storage devices include Optical storage and Tape storage.

## Slide 7



### Instruction(s) to the trainer:

Show slide 7 and discuss the storage hierarchy of the memory of a computer system. Different storage types are organized on the basis of their accessing speed of the data, cost per unit of data, and medium's reliability.

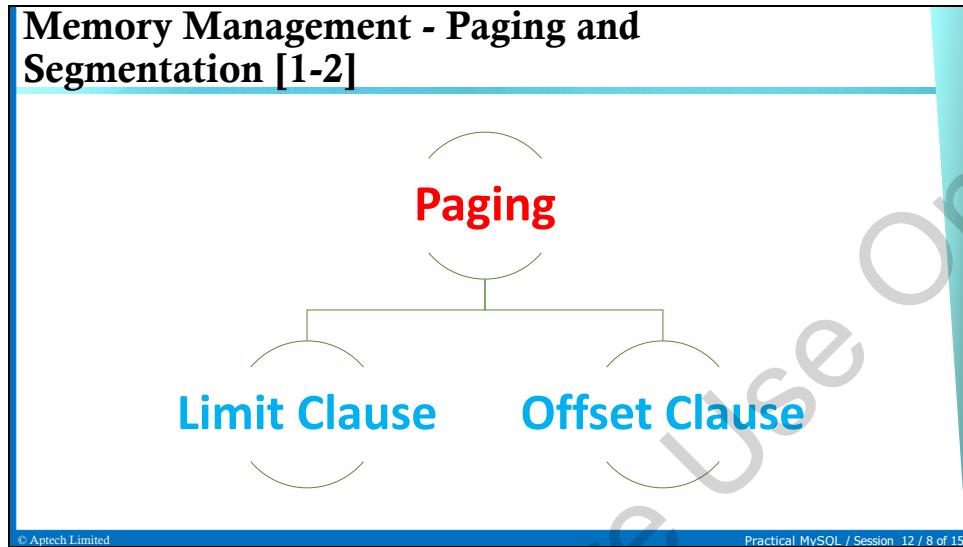
For instance, cache memory, which is the fastest and most easily accessible memory is placed at the top of the pyramid; whereas, magnetic tapes, which are slow in accessing data, are at the bottom.

### Additional Information:

Refer to following links for more information:

<https://www.tutorialspoint.com/Storage-Device-Hierarchy>

<https://www.sciencedirect.com/topics/computer-science/storage-hierarchy>



**Instruction(s) to the trainer:**

Use slide 8 and explain the concept of memory management in DBMS. Tell students that memory management relies completely on the main memory to store and access the data. It reduces the instruction set required to access data by eliminating latency and overhead of hard disk.

Tell students about some basic techniques to effectively manage memory in a database system. These techniques are paging and segmentation.

In Paging, a large dataset is divided into smaller parts, making it easy to read smaller chunks of data spread across multiple pages rather than a huge set of records on a single page.

Pagination can be implemented by using `LIMIT` and `OFFSET` clauses in the `SELECT` query. The `LIMIT` clause is used to restrict the number of rows to be retrieved, while the `OFFSET` clause is used to view a specific number of rows.

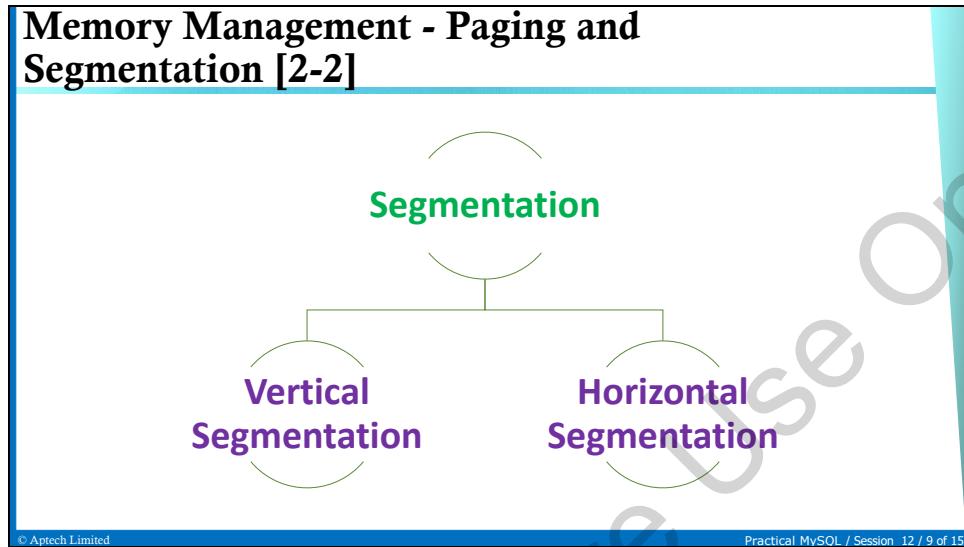
**Additional Information:**

Refer to following links for more information:

[https://www.tutorialspoint.com/operating\\_system/os\\_memory\\_management.htm](https://www.tutorialspoint.com/operating_system/os_memory_management.htm)

<https://www.javatpoint.com/os-memory-management-introduction>

<https://www.studytonight.com/operating-system/memory-management-in-os>



**Instruction(s) to the trainer:**

Show slide 9 and tell students that segmentation is a way to organize the database by grouping contacts and companies based on specific characteristics.

Explain that segmentation is a non-contiguous memory allocation scheme that divides the database into mounted size pages.

The data can be segmented in two ways:

- Vertical segmentation
- Horizontal segmentation

In vertical segmentation, the database is divided into vertical sub-library and vertical sub-table.

In the vertical sub-library, the vertical database stores low-relevancy tables in other databases.

In the vertical sub-table, it splits the columns with low coupling in the table with uncommon fields.

Horizontal segmentation is used to segment huge amounts of data. The data can be divided into horizontal sub-library and horizontal sub-table. Horizontal sub-library splits the single database into multiple databases.

Discuss the advantages and disadvantages of both types of segmentation techniques with students.

**In-Class Question:** You want to split a large database into small databases. Which type of segmentation would you choose?

**Answer:** Horizontal segmentation

For Aptech Centre Use Only

## Virtual Memory

Virtual Memory can be defined as a storage allocation scheme in which secondary memory is addressed.

### Reasons why Operating Systems use Virtual memory:

Virtual memory can be used to read the Data faster. While reading the Data, one cannot assure that Physical Cache works faster. The virtual Cache is used to share the code without keeping several copies of the same code. Without Virtual Memory, one cannot perform more than one operation in the Main Memory.

Virtual Memory ensures Data security. By providing position independence to the Data, the user can store Data at any position in the Main memory.

In memory fragmentation and errors, virtual memory debugs, and checks unallocated memory and NULL pointers.

### Instruction(s) to the trainer:

Use slide 10 to define virtual memory as a storage allocation scheme in which secondary memory is addressed. Tell students that virtual memory helps resolve the problem of memory shortage. It is a temporary memory that maps program addresses into RAM addresses.

Next, explain how virtual memory works to improve the performance of database operations. MySQL allocates buffers and caches. The default configuration of running MySQL Server in virtual memory is approximately 512 MB RAM. One can improve MySQL performance by increasing the values of cache. If there is a lack of memory, the addresses will be mapped into the disk.

Discuss the reasons why all operating systems must use virtual memory.

### Additional Information:

Refer to following links for more information:

<https://www.javatpoint.com/os-virtual-memory>

<https://www.tutorialspoint.com/what-is-virtual-memory>

<https://www.geeksforgeeks.org/virtual-memory-in-operating-system>

## Demand Paging

### STEP 1

If the CPU fails to find the page it is referring to, then it generates an interrupt, indicating a memory access fault.

### STEP 2

The OS puts the interrupted process to blocked. To continue with execution, the OS must bring the required page to the memory.

### STEP 3

OS will search the page in the logical address space.

### STEP 4

The page is brought from Logical address to Physical address. Page replacement algorithm is used in this process.

### STEP 5

After bringing the page from Logical to Physical address, it is updated.

### Instruction(s) to the trainer:

Show slide 11 and explain demand paging. It is a process where the page can be loaded into the main memory on demand. Explain how demand paging actually works with the help of the steps shown on the slide.

In demand paging, a page is fetched into the main memory only when it is needed or demanded by the CPU. Initially, only those pages are loaded that are required by the process immediately. The pages that are never accessed are therefore never loaded into the physical memory. This helps save memory space in DBMS.

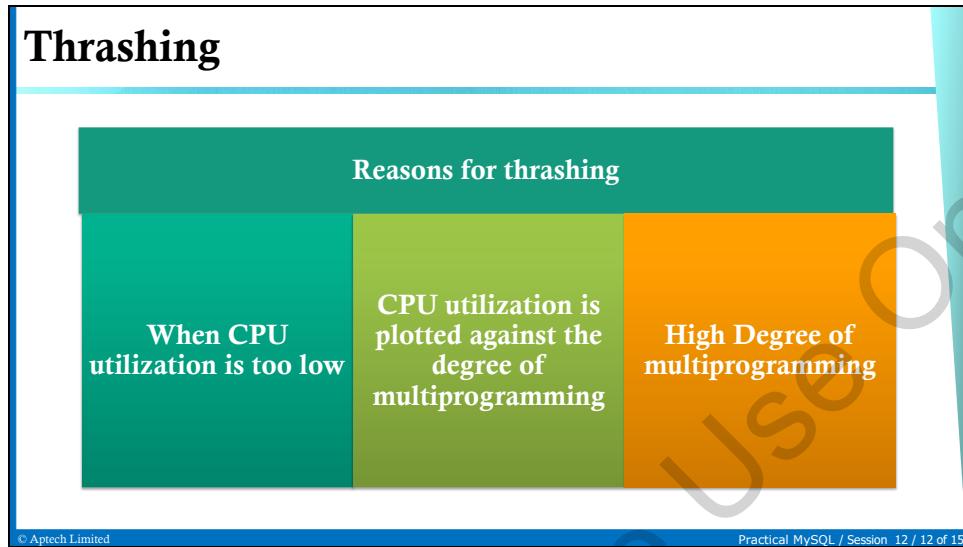
### Additional Information:

Refer to following links for more information:

<http://prepinsta.com/operating-systems/implementation-of-demand-paging/>

<https://www.javatpoint.com/os-demand-paging>

<https://www.tutorialandexample.com/what-is-demand-paging>



**Instruction(s) to the trainer:**

Use slide 12 to explain the concept of thrashing in an operating system. Thrashing mainly occurs when the required page is not allocated and generates a page fault. Inform students that thrashing degrades the computer performance.

Explain students various reasons that lead to thrashing.

Some ways to remove thrashing are as follows:

- Temporary solution for thrashing is to eliminate one or more running applications.
- Provide more memory to the main memory.
- Adjusting the size of swap files.
- Replace the program that takes more memory with the programs that take less memory.

**Additional Information:**

Refer to following links for more information:

<https://www.studytonight.com/operating-system/thrashing>

<https://practice.geeksforgeeks.org/problems/thrashing-in-os>

<https://www.i2tutorials.com/os-introduction/os-thrashing-in-operating-system>

## Storage Engines

Transactional      Non-Transactional

Different storage engines in MySQL include:

InnoDB    MyISAM    CSV    Merge    Archive storage

© Aptech Limited      Practical MySQL / Session 12 / 13 of 15

**Instruction(s) to the trainer:**

Show slide 13 and explain storage engines to students. Storage engines can be defined as MySQL components that are used to handle the SQL operations for different table types to store and manage information in a database.

There are two major types of storage engines in MySQL:

- Transactional
- Non-transactional

The main difference between the transactional and non-transactional storage engines is that in a transactional storage engine, the tables record all the database operations in a log file. This allows you to get your data back even if MySQL crashes. However, this is not the case with non-transactional engines.

List different storage engines in MySQL.

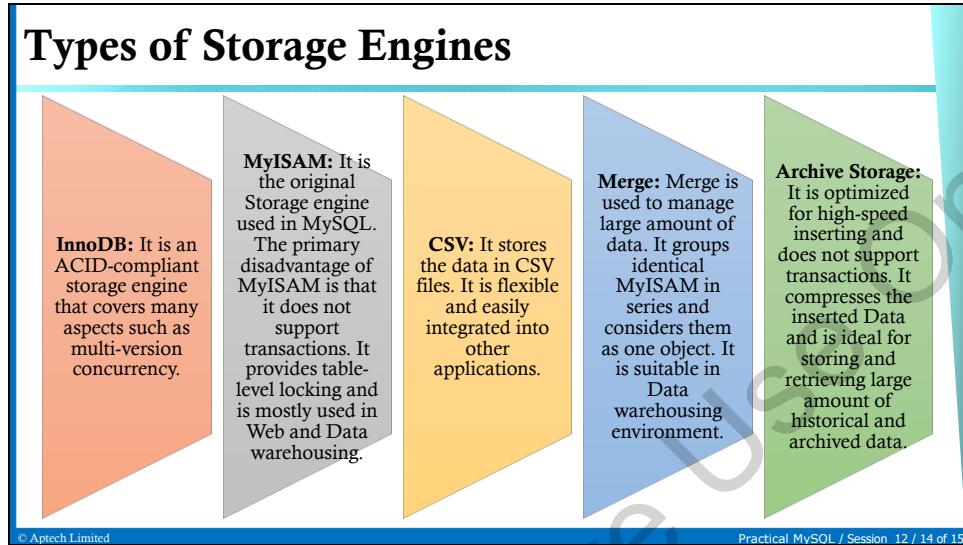
**Additional Information:**

Refer to following links for more information:

<https://www.w3resource.com/mysql/mysql-storage-engines.php>

<https://zetcode.com/mysql/storageengines>

<https://www.javatpoint.com/mysql-table-types-storage-engines>



#### Instruction(s) to the trainer:

Show slide 14 and explain the working of various types of storage engines: InnoDB, MyISAM, CSV, Merge, and Archive Storage. Discuss specific situations and benefits of using each type of storage engine.

Tell students that while creating tables, they can also specify which storage engine to use.

Following command is used to specify or apply the storage engine:

```
CREATE TABLE Customers (Cust_Id INTEGER PRIMARY KEY, Name  
VARCHAR(50), Age INTEGER) ENGINE='MyISAM';
```

#### Additional Information:

Refer to following links for more information:

<https://www.w3resource.com/mysql/mysql-storage-engines.php>

<https://zetcode.com/mysql/storageengines>

<https://www.javatpoint.com/mysql-table-types-storage-engines>

<https://www.thegeekdiary.com/beginners-guide-to-storage-engines-in-mysql>

**In-Class Question:** Which storage engine is capable of managing huge amounts of data and is generally used in data warehousing?

**Answer:** Merge

For Aptech Centre Use Only

## Summary

- The storage management concept refers to the process that improves the performance of Data Storage Resources.
- Main memory operates the Data using the storage medium. It handles the instructions of the machine.
- Tertiary Storage stores a huge amount of Data. They are external to the Computer System and not a part of the local Computer System.
- Paging is the process of dividing a large Dataset into smaller parts.
- OFFSET is used to view a specific number of rows.
- Segmentation is a powerful tool used to create a variable within their SQL Database.
- Thrashing is the situation when the system is spending a major portion of its time in servicing the page faults.
- Transactional and Non-Transactional are two different storage engine types in MySQL.

### Instruction(s) to the trainer:

Use slide 15 to summarize the session. You will end the session with a summary of what has been taught in the session. Tell students the pointers of the session. This will be a revision of the current session.