MySQL 5.6 Database System

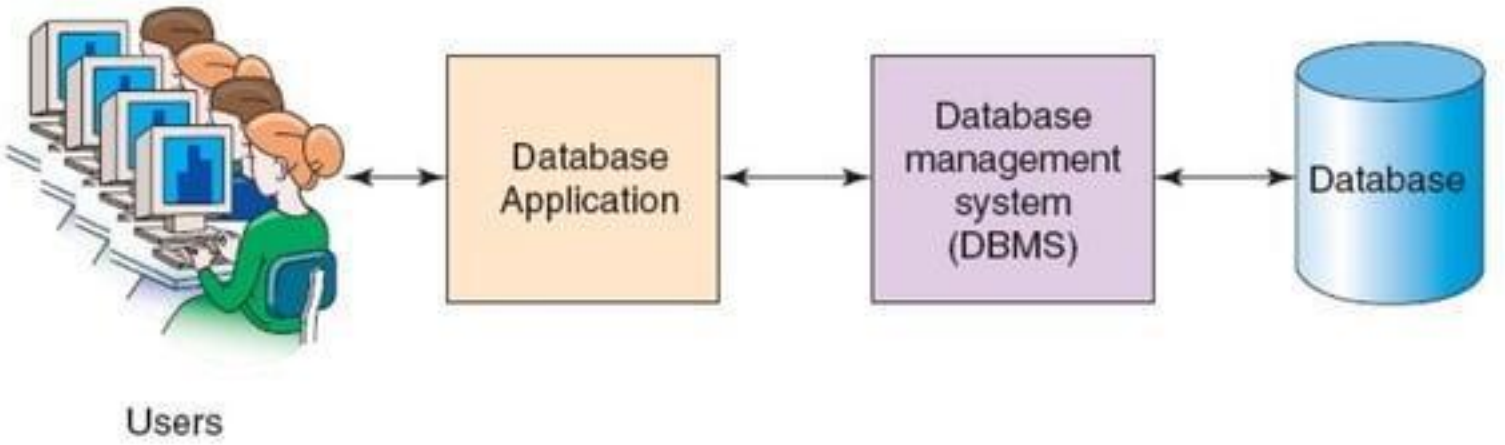
Presented by : Ms. ARSHI

What is RDBMS?

A database management system (DBMS) defines, creates, and maintains a database.

- RDBMS data is structured in database tables, fields and records.

Components of a Database System



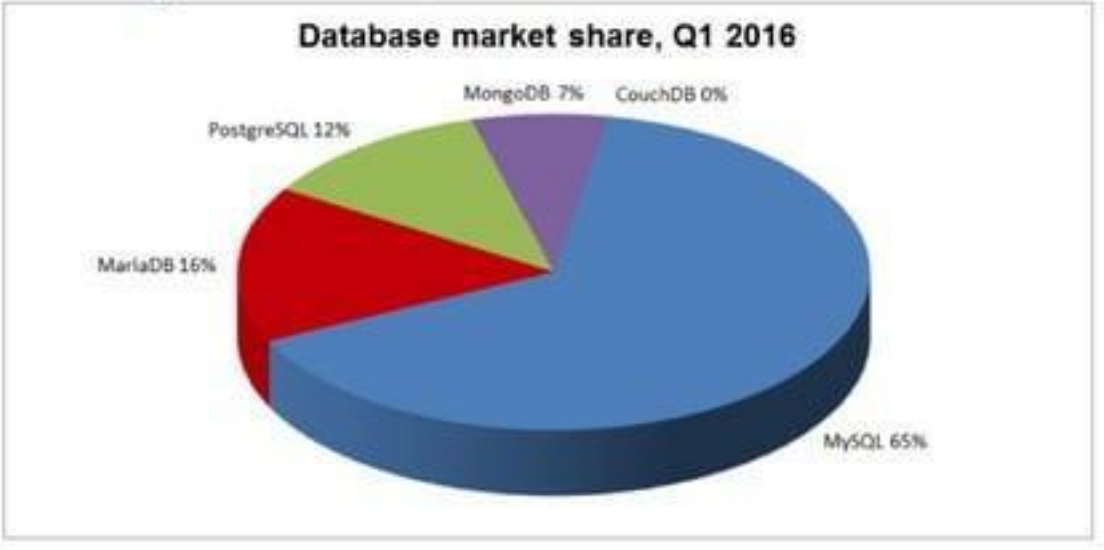
Comparison of Relational database management systems

Who are the main contenders?

- Commercial software
  - Sybase Adaptive Server Enterprise
  - IBM DB2
  - Oracle
  - Microsoft SQL Server
  - Teradata
- Free/GPL/Opensource:
  - MySQL
  - PostgreSQL

Vendor leaders

- For low-medium level servers, Oracle is the leader in the market share, although growth is declining
- IBM DB2 has the highest market share for high-end servers and mainframes, increasing growth in midrange after acquisition of Informix



Operating System Support

	Windows	Mac OS X	Linux	BSD	UNIX
Adaptive Server Enterprise	Yes	Yes	Yes	Yes	Yes
DB2	Yes	No	Yes	No	Yes
Microsoft SQL Server	Yes	No	No	No	No
MySQL	Yes	Yes	Yes	Yes	Yes
Oracle	Yes	Yes	Yes	No	Yes
PostgreSQL	Yes	Yes	Yes	Yes	Yes
Teradata	Yes	No	Yes	No	Yes

Different Types of Database

- Relational Databases
- Operational Databases
- Database Warehouses
- Distributed Databases
- End-User Databases

Why MySQL?

- MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout it's history.
- MySQL is RDBMS which runs a server, providing multi-user access to a number of databases.
- With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for IT in all sectors or domains.

Uses

- Many web applications use MySQL as the database component of a LAMP(Linux (operating system), Apache HTTP Server, MySQL (database software), and PHP, Perl or Python) software stack.
- Its popularity for use with web applications is closely tied to the popularity of PHP , which is often combined with MySQL.
- Several high-traffic web sites includes: Flickr, Facebook, Wikipedia, Google, Nokia and YouTube use MySQL for data storage and logging of user data.

Features of MySQL

- MySQL is written in C and C++ and its SQL parser is written in yacc(**Yet Another Compiler Compiler).**
- MySQL uses only just under 1 MB of RAM on your laptop while Oracle 9i installation uses 128 MB
- MySQL is great for database enabled websites while Oracle is made for enterprises.
- MySQL is portable.
- MySQL default port number is 3306.

Storage Engines

- A storage engine is a software module that a database management system uses to create, read, update data from a database.
- MyISAM, InnoDB, Memory, Merge, Archive, Federated, NDB, CSV, Blackhole, Example.
- By Default: InnoDB for versions after 5.5
  - A transaction-safe (ACID compliant) storage engine for MySQL that has commit, rollback, and crash-recovery capabilities to protect user data.

Database Tables in MySQL

- 5 different tables present in MySQL are

What are Heap Tables?

ISAM(Indexed Sequential Access Method)

Installation Summary

- MySQL can be installed on Linux using RPM packages.
- shell> rpm -qpl MySQL-server-*VERSION*.glibc23.i386.rpm
- This package will install mysql in /var/lib/mysql
- Package can be downloaded from:

http://dev.mysql.com/downloads/mysql/#downloads

Connecting to the Server

- Use a terminal that sets the path to **/var/lib/mysql/bin**
- The following command connects to the server:
  - **mysql -u root -p**
  - you are prompted for the root password.
  - you can now send commands and SQL statements to the server

Client-Server Interaction

Client program can be a MySQL command line client, GUI client, or a program written in any language such as C, Perl, PHP, Java that has an interface to the MySQL server.

MySQL Connectors & API

MySQL Connectors provide connectivity to the MySQL server for client programs

Connector/ODBC	• Connecting to MySQL using the ODBC API
Connector/Net	• to create .NET applications that connect to MySQL
Connector/J	• for connecting to MySQL from Java applications using the standard JDBC API.
Connector/Python	• connecting to MySQL from Python applications
Connector/C++	• enables C++ applications to connect to MySQL
Connector/C	• used for C applications.

Connector/J

- Connector/J is a JDBC Type 4 Driver for connecting Java to MySQL
- Installation is very simple:
  - Download the "Production Release" ZIP file from http://dev.mysql.com/downloads/connector/j/3.1.html
  - Unzip it
  - Put the JAR file where Java can find it
    - Add the JAR file to your CLASSPATH, or
    - In Eclipse: Project --> Properties --> Java Build Path --> Libraries --> Add External Jars...

Connecting to the server

First, make sure the MySQL server is running

In your program,

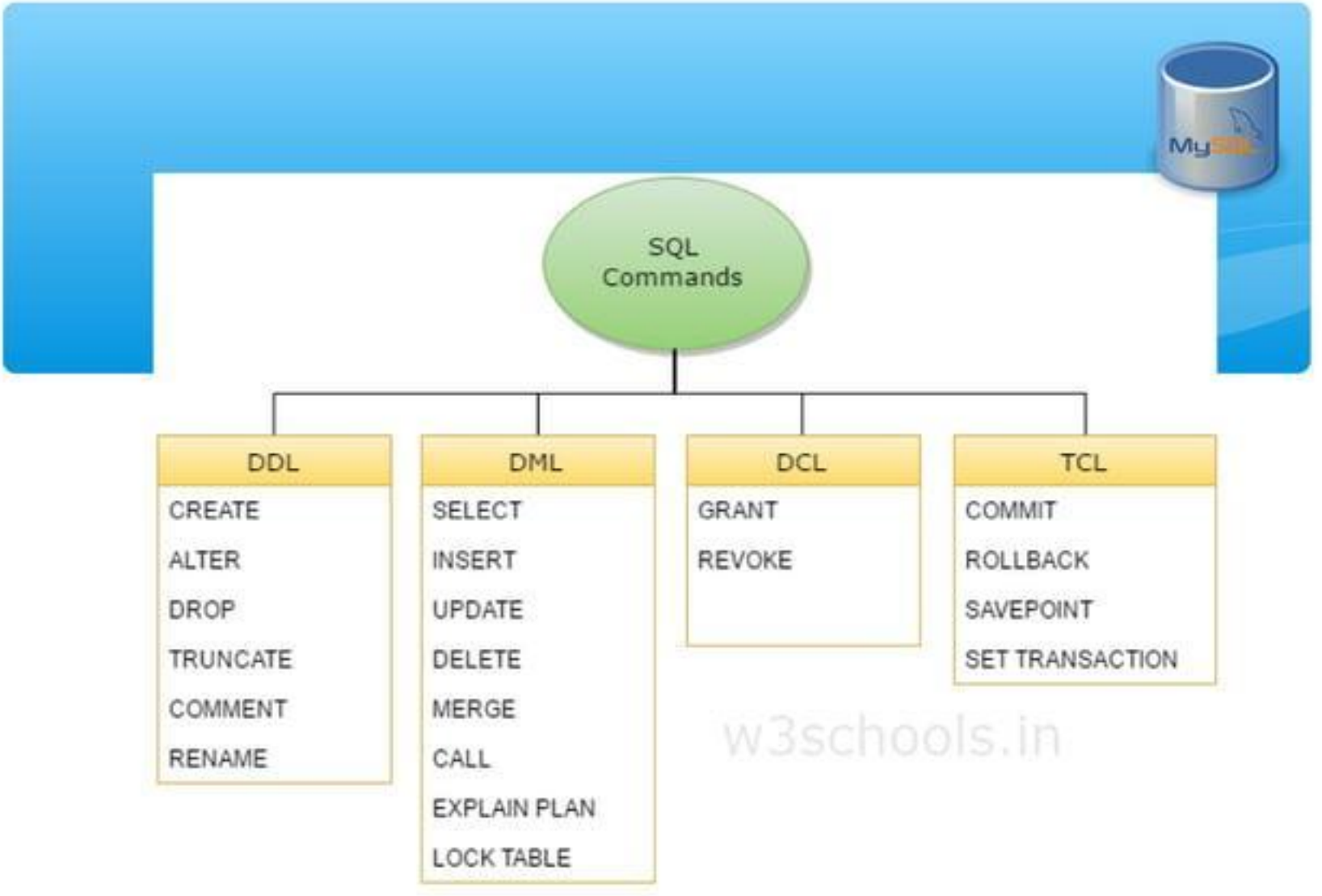
- import java.sql.Connection; // not com.mysql.jdbc...
- import java.sql.SQLException;

- Register the JDBC driver,
  Class.forName("com.mysql.jdbc.Driver").newInstance();

- Invoke the getConnection() method,
  Connection con =
  DriverManager.getConnection("jdbc:mysql:///myDB",
  myUserName,
  myPassword);
- or getConnection("jdbc:mysql:///myDB?user=dave&password=xxx")

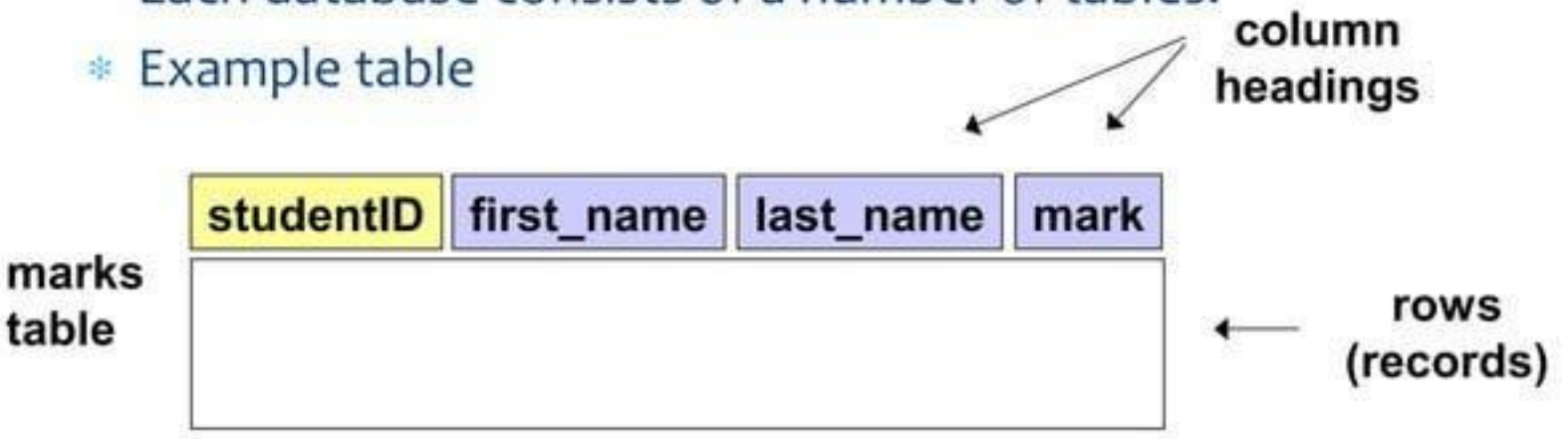
Types of SQL Language statements

* As a language, the SQL standard has three major components:
* A Data Definition Language (DDL) for defining the database structure and controlling access to the data.
* A Data Manipulation Language (DML) for retrieving and updating data.
* A Data Control Language(DCL) concerns with rights, permissions and other controls of the database system.


SQL
Commands
DDL
CREATE
ALTER
DROP
TRUNCATE
COMMENT
RENAME
DML
SELECT
INSERT
UPDATE
DELETE
MERGE
CALL
EXPLAIN PLAN
LOCK TABLE
DCL
GRANT
REVOKE
TCL
COMMIT
ROLLBACK
SAVEPOINT
SET TRANSACTION
w3schools.in

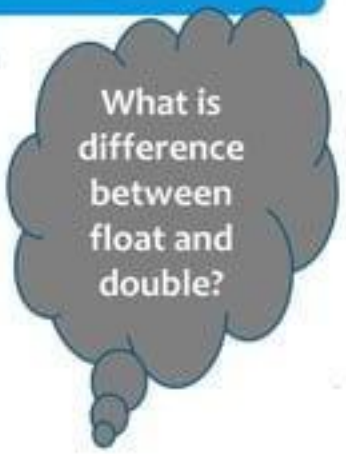
Database concepts

- A relational database management system consists of a number of databases.
- Each database consists of a number of tables.
- Example table



Some SQL data types (1)

- Each entry in a row has a type specified by the column.
- Numeric data types
  - **`TINYINT, SMALLINT, MEDIUMINT,`**
  - **`INT, BIGINT`**
  - **`FLOAT(display_length, decimals)`**
  - **`DOUBLE(display_length, decimals)`**
  - **`DECIMAL(display_length, decimals)`**
    - **`NUMERIC`** is the same as **`DECIMAL`**



Some SQL data types (2)

- Date and time types
  - **DATE**
    - format is YYYY-MM-DD
  - **DATETIME**
    - format YYYY-MM-DD HH:MM:SS
  - **TIMESTAMP**
    - format YYYYMMDDHHMMSS
  - **TIME**
    - format HH:MM:SS
  - **YEAR**
    - default length is 4

SQL data types (3)

- String types
  - **CHAR**
    - fixed length string, e.g., **CHAR(20)**
  - **VARCHAR**
    - variable length string, e.g., VARCHAR(20)
  - **BLOB, TINYBLOB, MEDIUMBLOB, LONGBLOB**
    - same as **TEXT**, **TINYTEXT** ...
  - **ENUM**
    - list of items from which value is selected

SQL commands SHOW, USE

- **SHOW**
  - Display databases or tables in current database;
  - `show databases;`
  - `show tables;`
- **USE**
  - Specify which database to use
  - `use bookstore;`

The CREATE Command (1)

CREATE creates a database table

```
CREATE TABLE table_name
(
   column_name1 column_type1,
   column_name2 column_type2,
   ...
   column_nameN column_typeN
);
```

* Specifying primary keys

```
CREATE TABLE table_name
(
   column_name1 column_type1,
   column_name2 column_type2,
   ...
   column_nameN column_typeN,
   PRIMARY KEY (column_name1)
);
```

The DROP & INSERT Commands

- To delete databases and tables use the DROP command

- Examples
  - DROP DATABASE db_name;
  - DROP TABLE table_name;

What is difference between DROP and DELETE commands ?

- Inserting rows into a table using INSERT comman

```
INSERT INTO table_name
   ( col_1, col_2, ..., col_N)
VALUES
   ( val_1, val_2, ..., val_N);
```

The SELECT Command (1)

- Selecting rows from a table
- Simplest form: select all columns

```
SELECT * FROM table_name;
```

- Select specified columns

```
SELECT column_list FROM table_name;
```

- Conditional selection of rows

```
SELECT column_list FROM table_name
WHERE condition;
```

The SELECT Command (2)

- Specifying ascending row ordering

```
SELECT column_list FROM table_name
WHERE condition
ORDER by ASC;
```

- Specifying descending row ordering

```
SELECT column_list FROM table_name
WHERE condition
ORDER by DESC;
```

The UPDATE Command

Used to modify an existing record

```
UPDATE table_name
SET col_1 = 'new_value1',
..., col_n = 'new_value2';
```

* Conditional update version

```
UPDATE table_name
SET col_1 = 'new_value1',
..., col_n = 'new_value2'
WHERE condition;
```

The Marks Table

```
SELECT * FROM marks;
```

- Selecting the complete table

```
+-----------+------------+-----------+------+
| studentID | first_name | last_name | mark |
+-----------+------------+-----------+------+
|         1 | Fred       | Jones     |   78 |
|         2 | Bill       | James     |   67 |
|         3 | Carol      | Smith     |   82 |
|         4 | Bob        | Duncan    |   60 |
|         5 | Joan       | Davis     |   86 |
+-----------+------------+-----------+------+
5 rows in set (0.00 sec)
```

The WHERE Clause (1)

```
SELECT * FROM marks WHERE studentID > 1
    AND studentID < 5;
```

- Select rows according to some criterion

```
+-----------+------------+-----------+------+
| studentID | first_name | last_name | mark |
+-----------+------------+-----------+------+
|         2 | Bill       | James     |   67 |
|         3 | Carol      | Smith     |   82 |
|         4 | Bob        | Duncan    |   60 |
+-----------+------------+-----------+------+
3 rows in set (0.01 sec)
```

The ORDER BY Clause

```
SELECT * FROM marks ORDER BY mark DESC;
```

- Select rows according to some criterion

```
+-----------+------------+-----------+------+
| studentID | first_name | last_name | mark |
+-----------+------------+-----------+------+
|         5 | Joan       | Davis     |   86 |
|         3 | Carol      | Smith     |   82 |
|         1 | Fred       | Jones     |   78 |
|         2 | Bill       | James     |   67 |
|         4 | Bob        | Duncan    |   60 |
+-----------+------------+-----------+------+
5 rows in set (0.00 sec)
```

Aggregate Functions

COUNT()

- Select count(*) from marks;

SUM()

- Select sum(mark) from marks;

AVG()

- Select Avg(mark) from marks;

MIN()

- Select Min(mark) from marks;

MAX()

- Select Max(mark) from marks;

Cloud SQL

- A fully managed relational mySQL databases on cloud hosted by Google platform.
- Runs on Google infrastructure
- Google + MySQL
- Cloud SQL provides a database infrastructure for applications running anywhere
- Wordpress sites, e-commerce applications, CRM tools, or any other application that is compatible with MySQL.
- It doesn't require any software installation.
- Security:
  - Cloud SQL customer's data is encrypted i.e. Every Cloud SQL instance includes a network firewall.


ANY
QUESTIONS???

Thank you