

**CLASS- XII**  
**COMPUTER SCIENCE WITH**  
**PYTHON**

---

CHAPTER- 13  
MySQL SQL Revision Tour

# DATABASE

---

- It is a collection of data on computer based record keeping system.

## **ADVANTAGES OF DATABASE SYSTEMS**

1. Reduce data redundancy (data duplication)
2. Control data inconsistency
3. Facilitate sharing of data
4. Enforce standards
5. Ensure data security
6. Maintain integrity



# TERMINOLOGIES OF RELATIONAL DATA MODEL

---

- **Relation**- A table storing logically related data.
- **Domain**- A pool of values from which actual values appearing in a column are drawn.
- **Tuple**- A row of a relation.
- **Attribute**- A column of a relation.
- **Degree**- The number of attributes in a relation.
- **Cardinality**- The number of tuples in a relation.
- **View**- The virtual table that doesn't really exist but is derived from one or more base table.

# TERMINOLOGIES OF RELATIONAL DATA MODEL

---

- **Primary key**- A set of one or more attributes that can uniquely identify tuples within the relation.
- **Candidate key**- All attributes combinations inside the relation that can serve as primary key are candidate keys.
- **Alternate key**- A candidate key that is not primary key.
- **Foreign key**- A non-key attribute whose values are derived from primary key of some other table.
- **Referential integrity**- It is a system of rules that a DBMS uses to ensure that relationships between records in related tables are valid and users don't accidentally delete or change related data.

# MySQL and SQL

---

- **MySQL** It is a freely available open source RDBMS that uses SQL.
- **SQL (Structured Query Language)**- It is a language that enables us to create & operate on relational databases which are sets of related information stored in tables.



# CLASSIFICATION OF SQL STATEMENTS

---

1. Data Definition Language (DDL) Commands
2. Data Manipulation Language (DML) Command
3. Transaction Control Language (TCL) Commands
4. Session Control Commands
5. System Control Commands

# 1. Data Definition Language (DDL) Commands

---

- i) **CREATE, ALTER, DROP** - creating, altering & dropping
- ii) **GRANT, REVOKE** - granting & revoking
- iii) **ANALYZE, CHECK, REPAIR, RESTORE** - maintenance commands

## 2. Data Manipulation Language (DML) Command

---

- i) **SELECT** - retrieval of data
- ii) **INSERT** - insertion of data
- iii) **DELETE** - deletion of data
- iv) **UPDATE** - modification of data



### 3. Transaction Control Language (TCL) Commands

---

- i) **COMMIT** - making changes to database
- ii) **ROLLBACK** - undoing changes to database
- iii) **SAVEPOINT** - creating savepoints
- iv) **SET TRANSACTION** - setting properties for current transactions

# DATA TYPES

---

1. **INT**- for numeric type of data
2. **CHAR**- for character type of data
3. **VARCHAR**- for character type of data
4. **DATE**- for date type like YYYY-MM-DD
5. **TIME**- for time type like HH:MM:SS
6. **DATETIME**- for both date and time like YYYY-MM-DD HH:MM:SS

# DIFFERENCE BETWEEN CHAR & VARCHAR

---

CHAR	VARCHAR
Fixed length character string	Variable length character string
All values stored in the column have length of n bytes.	The maximum size a value in column can have n bytes.
If value is shorter than n then blanks are added but size remains n bytes.	Each value added in column stores exactly as we specify no blanks are added if length is shorter than maximum length n.



---

# QUERIES IN SQL

# CREATING DATABASE

---

Syntax- **Create database database-name;**

Example- **Create database db1;**

# ACCESSING DATABASE

---

Syntax- `use database-name;`

Example- `use db1;`



# CREATING TABLES

---

Syntax- Create table table-name

(Column1 datatype (size),  
Column2 datatype (size)  
.....column n datatype(size));

Example- Create table emp

(ecode integer, ename varchar(20), gender char(1));

# INSERTING DATA INTO TABLE

---

- **INSERT QUERY**

Syntax- **Insert into table-name values(value1, value2,..... value n);**

Example- **Insert into emp values (1001, 'Ravi', 'M', 'E4', 4675);**

- **INSERTING NULL VALUES**

Example- **Insert into emp (eno, ename, sal, comm, dno)  
values (8100, "YASH", 6000, NULL,20);**

# SELECT COMMAND QUERIES-1

---

- **SELECT QUERY**

Syntax- **Select column-name from table-name where condition;**

Example- **Select eno, ename from emp where eno=1001;**

- **SELECTING ALL DATA(\*)**

Syntax- **Select \* from table-name;**

Example- **Select \* from emp;**



# SELECT COMMAND QUERIES-2

---

- **SELECT PARTICULAR ROWS**

Example- `Select * from emp where gender='F';`

- **SELECT PARTICULAR COLUMNS**

Example- `Select ename, age from emp where sal>6000;`

# SELECT COMMAND QUERIES-3

---

- **TO ELIMINATE DUPLICATE VALUES (DISTINCT)**

Example- Select **DISTINCT** ename from emp;

- **TO INCLUDE DUPLICATE VALUES (ALL)**

Example- Select **ALL** city from emp;

# SELECT COMMAND QUERIES-4

---

- **VIEWING STRUCTURE OF TABLE (DESC)**

Syntax- **Describe / Desc table-name;**

Example- **Describe / Desc emp;**

- **PERFORMING SIMPLE CALCULATIONS**

Example- **Select 3.14\*6\*6;**

Example- **Select curdate( );**



# SELECT COMMAND QUERIES-5

---

- **USING COLUMN ALIASES**

Syntax- **Select column-name as column-alias from table-name;**

Example- **Select date, type as “Event Type” from emp;**

- **CONDITION BASED ON RANGE (BETWEEN)**

Example- **Select icode, desc from item**

**where QOH BETWEEN 30 AND 50;**

# SELECT COMMAND QUERIES-6

---

- **CONDITION BASED ON LIST (IN / NOT IN)**

Example- **Select \* from emp where city IN ('Delhi', 'Mumbai', 'Kolkata');**

Example- **Select \* from emp where city NOT IN ('Delhi', 'Mumbai', 'Kolkata');**

# SELECT COMMAND QUERIES-7

---

- **SEARCHING FOR NULL (IS NULL / IS NOT NULL)**

Example- **Select eno, ename from emp where dno IS NULL;**

Example- **Select eno, ename from emp where dno IS NOT NULL;**



# SELECT COMMAND QUERIES-8

---

- **CONDITION BASED ON PATTERN MATCH (LIKE/NOT LIKE)**

**1. Percent(%)-** Matches any substring.

Example- **Select Fname, Lname from emp where pin LIKE '13%';**

**2. Underscore(\_)-** Matches any character.

Example- **Select ename from emp where city NOT LIKE '\_ana';**

---

**!!COMPLETED!!**