



**Smit IT Institute**  
Gives you wings to fly high

# MySQL

What is MySQL:

MySQL is an Oracle-supported Relational Database Management System (RDBMS) based on structured query language. MySQL supports a wide range of operating systems, most famous of those include Windows, Linux & UNIX.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Complexity

- MySQL is written in C and C++, and its SQL parser is written in yacc.
- YACC stands for Yet Another Compiler Compiler.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Features of MySQL

- Flexible structure
- High performance
- Manageable and easy to use
- Replication and high availability
- Security and storage management
- Drivers
- Graphical Tools
- MySQL Enterprise Monitor
- MySQL Enterprise Security



**Smit IT Institute**

Gives you wings to fly high



## Different Between Data Base and a Table.

There is a major difference between a database and a table. The differences are as follows:

- Tables are a way to represent the division of data in a database while the database is a collection of tables and data.
- Tables are used to group the data in relation to each other and create a dataset. This dataset will be used in the database. The data stored in the table in any form is a part of the database.
- A database is a collection of organized data and features used to access them, whereas the table is a collection of rows and columns used to store the data.





**Smit IT Institute**  
Gives you wings to fly high

# DDL: Data Definition Language

- DDL is short name of Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.
- CREATE - to create a database and its objects like (table, index, views, store procedure, function, and triggers).
- ALTER - alters the structure of the existing database
- DROP - delete objects from the database
- TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
- COMMENT - add comments to the data dictionary
- RENAME - rename an object



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Adding Columns

Adding a column:

- **ALTER TABLE TABLE NAME ADD COLUMN NAME varchar(30);**

Adding Multiple Columns:

- **ALTER TABLE TABLE NAME ADD NEW COLUMN varchar(20)  
AFTER COLUMN NAME;**



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Modify Column

We can modify the column and change its data type size.

It can also force a column to be NOT NULL.

- .
- **ALTER TABLE** Table Name **MODIFY** Column Name **varchar(10);**



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# DROP Table

- What does DROP do?
- In MySQL, the DROP command is used to remove the whole database or table indexes, data, and more. The important part of this command is that it has the ability to permanently remove the table with its contents.

**DROP** Table Name;



**Smit IT Institute**  
Gives you wings to fly high





# TRUNCATE TABLE

- What is TRUNCATE?
- In MySQL, the TRUNCATE command is used to remove all the rows from the table. However, the structure of the table and columns remains the same. It is faster than the DROP command.
- **TRUNCATE TABLE TABLE NAME;**





# Difference between DROP and TRUNCATE

S.No	DROP	TRUNCATE
1.	It is used to eliminate the whole database from the table.	It is used to eliminate the tuples from the table.
2.	Integrity constraints get removed in the DROP command.	Integrity constraint doesn't get removed in the Truncate command.
3.	The structure of the table does not exist.	The structure of the table exists.
4.	Here the table is free from memory.	Here, the table is not free from memory.
5.	It is slow as compared to the TRUNCATE command.	It is fast as compared to the DROP command.





**Smit IT Institute**  
Gives you wings to fly high

# DML: Data Manipulation Language

DML is short name of Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

- SELECT - retrieve data from a database.
- INSERT - insert data into a table.
- UPDATE - updates existing data within a table.
- DELETE - Delete all records from a database table.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# UPDATE

- MySQL UPDATE query is a DML statement used to modify the data of the MySQL table within the database. In a real-life scenario, records are changed over a period of time. So, we need to make changes in the values of the tables also. To do so, it is required to use the UPDATE query.
- UPDATE TABLE NAME SET COLUMN\_NAME WHERE (FILTER CONDITION);



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# DELETE

- MySQL DELETE statement is used to remove records from the MySQL table that is no longer required in the database. **This query in MySQL deletes a full row from the table and produces the count of deleted rows.**
- DELETE FROM TABLE\_NAME WHERE (FILTER CONDITION);



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# DISTINCT

- It generates summaries.
  - It removes the duplicate rows.
- 
- `SELECT DISTINCT COLUMN_NAME FROM TABLE NAME;`



**Smit IT Institute**  
Gives you wings to fly high



# ORDER BY

It Provides an Ascending sequence by default while displaying the records.

“ORDER BY” Keyword is mentioned before the column name which is to be displayed.

```
SELECT COL_NAME FROM TABLE NAME ORDER BY COL_NAME;
```

- WHEN USED WITH THE KEYWORD DESC, PROVIDES DESCENDING RECORDS.

```
SELECT COL_NAME FROM TABLE NAME ORDER BY COL_NAME DESC;
```





**Smit IT Institute**

Gives you wings to fly high

# GROUP BY

- It helps to cluster the records and makes it easy to access and Manipulate them.
- `SELECT COLUMN_NAME FROM TABLE_NAME GROUP BY COLUMN_NAME;`



**Smit IT Institute**

Gives you wings to fly high





**Smit IT Institute**  
Gives you wings to fly high

# Primary Key

**A primary key is a column or a set of columns that uniquely identifies each row in the table.**

The primary key follows these rules:

- A primary key must contain unique values. If the primary key consists of multiple columns
- A primary key column cannot have NULL values. Any attempt to insert or update NULL to primary key columns will result in an error.

**Note that MySQL implicitly adds a NOT NULL constraint to primary key columns.**

- A table can have one and only one primary key.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# PRIMARY KEY

- `CREATE TABLE TABLE_NAME(COL_NAME DATA_TYPE VARCHAR(SIZE));`

Smit IT Institute



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# UNIQUE KEY

A UNIQUE constraint is an integrity constraint that ensures values in a column or group of columns to be unique.

A UNIQUE constraint can be either a column constraint or a table constraint.

```
CREATE TABLE  
TABLE_NAME(COL_NAME VARCHAR(SIZE)  
UNIQUE);
```



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# NOT NULL

- NULL means “a missing unknown value” and it is treated somewhat differently from other values. Because the result of any arithmetic comparison with NULL is also NULL
- `CREATE TABLE TABLE_NAME(COL_NAME DATATYPE(SIZE) NOT NULL)`



**Smit IT Institute**  
Gives you wings to fly high



# MULTI-ROW FUNCTIONS

- **Multiple Row Functions in Oracle**
- The Multiple Row Functions in Oracle are used to return either group of values (or) a single value. These functions are basically operated on a set of rows and return one result or one result per group. This is a powerful feature provided by oracle because these Multiple Row Functions allow us to generate subtotals, max. min, sums, and averages within the SQL that is retrieving the data. The Multiple row function in Oracle is also called group functions or it is also called aggregate functions.





**Smit IT Institute**  
Gives you wings to fly high

# MAX()

- SYNTAX:
- SELECT MAX(COL\_NAME) FROM TABLE\_NAME;
- IT RETURNS MAXIMUM VALUE OF SELECTED COLUMN.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# MIN()

- SYNTAX:
- SELECT MIN(COL\_NAME) FROM TABLE\_NAME;
- IT RETURNS MINIMUM VALUE OF SELECTED COLUMN.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# SUM()

- SYNTAX:
- `SELECT SUM(COL_NAME) FROM TABLE_NAME;`
- It return the total addition of all data in that column.



**Smit IT Institute**  
Gives you wings to fly high





**Smit IT Institute**  
Gives you wings to fly high

# AVG()

- SYNTAX:
- SELECT AVG(COL\_NAME) FROM TABLE\_NAME;
- It return the AVERAGE of all data in that column.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# COUNT()

- SYNTAX:
- `SELECT COUNT(COL_NAME) FROM TABLE_NAME;`
- It return the TOTAL ROW in that column.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# DATE/TIME

- **SELECT SYSDATE();**
- **SELECT NOW();**
- **SELECT DATE("2022-11-23");**
- **SELECT MONTH("2022-11-23");**
- **SELECT MONTHNAME("2022-11-23 10:50:34");**
- **SELECT YEAR("2022-11-23 10:50:34");**
- **SELECT QUARTER("2022-11-22 08:32:50");**
- **SELECT MICROSECOND("2022-11-22 08:32:50.999999");**



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# DATE/TIME

- **SELECT DAYOFMONTH("2022-11-22 08:32:50");**
- **SELECT DAYNAME("2022-11-22 08:32:50");**
- **SELECT DAYOFWEEK("2022-11-22 08:32:50");**
- **SELECT DAYOFYEAR("2022-11-22 08:32:50");**
- **SELECT WEEK("2022-11-22 08:32:50");**
- **SELECT YEARWEEK("2022-11-22 08:32:50");**
- **SELECT LAST\_DAY("2022-11-22 08:32:50");**
- **SELECT EXTRACT(YEAR\_MONTH FROM ("2022-11-22 08:32:50"));**



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# DATE/TIME

- **SELECT EXTRACT(WEEK FROM ("2022-11-22 08:32:50"));**
- **SELECT EXTRACT(MONTH FROM ("2022-11-22 08:32:50"));**
- **SELECT EXTRACT(YEAR FROM ("2022-11-22 08:32:50"));**
- **SELECT EXTRACT(DAY FROM ("2022-11-22 08:32:50"));**
- **SELECT EXTRACT(HOUR FROM ("2022-11-22 08:32:50"));**
- **SELECT EXTRACT(MINUTE FROM ("2022-11-22 08:32:50"));**
- **SELECT EXTRACT(SECOND FROM ("2022-11-22 08:32:50"));**
- **SELECT EXTRACT(MICROSECOND FROM ("2022-11-22 08:32:50.78522"));**



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# CURTIME(FSP)

- DISPLAY CURRENT TIME.
- FSP:-FRACTIONAL SECONDS PARTS
- WITH FSP IT DISPLAYS SPECIFIED NUMBERS OF MICRO-SECONDS.
- RANGE 0-6.
- WITHOUT FSP IT SHOWS ONLY TIME.
- SELECT CURTIME();
- SELECT CURTIME(FSP(0-6));



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# JOINS

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high





**Smit IT Institute**  
Gives you wings to fly high

# CROSS JOIN

CROSS JOIN RETURNS ALL THE RECORD FROM ALL THE TABLES.

IT MATCHES EVERY RECORD OF ONE TABLE WITH EVERY OTHER RECORD SECOND TABLE.(MATCHED RECORDS AS WELL AS UNMATCHED RECORDS) `SELECT * FROM TABL1,TABLE2;` ORACLE SYNTAX.

`SELECT * FROM TABLE1 CROSS JOIN TABLE2;` -> ANSI SYNTAX

- ANSI:-AMERICAN NATIONAL STANDARD INSTITUTE



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# INNER JOIN

- IT IS A TYPE OF JOIN WHERE WE WILL GET ONLY MATCHED RECORDS.
- IN THE INNER JOIN WE WILL GIVE JOIN CONDITION BECAUSE OF IT WE WILL GET MATCHED RECORDS ONLY.

SELECT NAME,DNAME FROM TABLE1\_NAME VAR1,TABLE2\_NAME  
VAR2 WHERE JOIN CONDITION; ->ORACLE

SELECT COL1,CO2 FROM TABLE1 INNER JOIN TABLE2 ON  
TABLE1\_NAME.COL\_NAME=TABLE2\_NAME.COL\_NAME;



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# LEFT Join

- The LEFT OUTER JOIN returns all rows from the left hand table specified in the ON condition and only those rows from the other table where the join where the matching conditions are true.
- **SELECT COLUMNS FROM** table1 LEFT JOIN table2 **ON** table1.column = table2.column;



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# RIGHT JOIN

- The RIGHT OUTER JOIN returns all rows from the RIGHT hand table specified in the ON condition and only those rows from the other table where the join where the matching conditions are true.
- **SELECT COLUMNS FROM** table1 RIGHT JOIN table2 **ON** table1.column = table2.column;



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Foreign Key

The foreign key is used to link one or more than one table together. It is also known as the **referencing** key. A foreign key matches the primary key field of another table. It means a foreign key field in one table refers to the primary key field of the other table. It identifies each row of another table uniquely that maintains the **referential integrity** in MySQL.



**Smit IT Institute**  
Gives you wings to fly high



## Foreign Key

• **CREATE TABLE** TABLE1 ( COL\_NAME DATA\_TYPE,  
COL\_NAME DATA\_TYPE, CONSTRAINT  
FOREIGN KEY(COL\_NAME) REFERENCES  
TABLE1(PRIMARY KEY COLUMN));





**Smit IT Institute**  
Gives you wings to fly high

# Properties of Transaction

The transaction contains mainly four properties, which referred to as ACID property.

Now, we are going to discuss the ACID property in detail.

- Atomicity
- Consistency
- Isolation
- Durability



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Atomicity

**Atomicity:** This property ensures that all statements or operations within the transaction unit must be executed successfully. Otherwise, if any operation is failed, the whole transaction will be aborted, and it goes rolled back into their previous state.



**Smit IT Institute**  
Gives you wings to fly high





**Smit IT Institute**  
Gives you wings to fly high

# Consistency

**Consistency:** This property ensures that the database changes state only when a transaction will be committed successfully. It is also responsible for protecting data from crashes.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Isolation

- **Isolation:** This property guarantees that each operation in the transaction unit operated independently. It also ensures that statements are transparent to each other.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Durability

**Durability:** This property guarantees that the result of committed transactions persists permanently even if the system crashes or failed.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# ACID

## ACID-PROPERTIES

- A-Atomicity
- Consistency
- I-Isolation
- D-Durability



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# MySQL | CREATE USER Statement

- MySQL allows us to specify which user account can connect to a database server. The user account details in MySQL contains two information – username and host from which the user is trying to connect in the format *username@host-name*.
- The **CREATE USER** statement in MySQL allows us to create new MySQL accounts or-in other words, the **CREATE USER** statement is used to create a database account that allows the user to log into the MySQL database.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Create User.

- **Syntax:**
  - CREATE USER user\_account IDENTIFIED BY password;
  - **parameters used:**
  - **user\_account:** It is the name that the user wants to give to the database account. The user\_account should be in the format '**username**'@'**hostname**'
  - **password:** It is the password used to assign to the user\_account. The password is specified in the IDENTIFIED BY clause.
  - e.g CREATE USER gfguser1@localhost IDENTIFIED BY 'jimmy';



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# DCL (Data Control Language)

- DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.
- **GRANT:** This command gives users access privileges to the database.

## Syntax:

*GRANT privileges\_names ON object TO user;*



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# DCL (Data Control Language)

- **privileges\_name:** These are the access rights or privileges granted to the user.
- **object:** It is the name of the database object to which permissions are being granted. In the case of granting privileges on a table, this would be the table name.
- **user:** It is the name of the user to whom the privileges would be granted.



**Smit IT Institute**

Gives you wings to fly high



# DCL (Data Control Language)

- Granting permission

**---GRANT SELECT ON Users TO user.**

- Granting more than one Privilege to a User in a Table ---  
**GRANT SELECT, INSERT, DELETE, UPDATE ON Users TO user.**

- Granting All the Privilege to a User in a Table:

**---GRANT ALL ON Users TO user**



# DCL (Data Control Language)

- **REVOKE:** This command withdraws the user's access privileges given by using the GRANT command.
- Revoke command withdraw user privileges on database objects if any granted. It does operations opposite to the Grant command. When a privilege is revoked from a particular user U, then the privileges granted to all other-users by user U will be revoked.





**Smit IT Institute**

Gives you wings to fly high

# DCL (Data Control Language)

- **.grant insert,**

select on accounts to user

By the above command user has granted permissions on accounts database object like he can query or insert into accounts.

**revoke insert,**

select on accounts from user

By the above command user permissions like query or insert on accounts database object has been removed.



**Smit IT Institute**

Gives you wings to fly high



# DCL (Data Control Language)

S.NO	Grant	Revoke
1	This DCL command grants permissions to the user on the database objects.	This DCL command removes permissions if any granted to the users on database objects.
2	It assigns access rights to users.	It revokes the user access rights of users.
3	For each user you need to specify the permissions.	If access for one user is removed; all the particular permissions provided by that users to others will be removed.
4	When the access is decentralized granting permissions will be easy.	If decentralized access removing the granted permissions is difficult.





**Smit IT Institute**

Gives you wings to fly high

# MySQL CREATE VIEW Statement

- In SQL, a view is a virtual table based on the result-set of an SQL statement.
- A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.
- You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.
- A view is created with the CREATE VIEW statement.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# MySQL CREATE VIEW Statement

- **CREATE VIEW view\_name AS SELECT column1, column2, ...**
- **FROM table\_name WHERE condition;**
- **Or**
- **CREATE VIEW view\_name AS SELECT \* FROM table\_name**



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# TCL Commands in SQL

- In MySQL, TCL stands for Transaction control language.
- A single unit of work in a database is formed after the consecutive execution of commands is known as a transaction.
- There are certain commands present in SQL known as TCL commands that help the user manage the transactions that take place in a database.
- COMMIT, ROLLBACK and SAVEPOINT are the most commonly used TCL commands in SQL.
- Autocommit is by default enabled in MySQL. To turn it off, we will set the value of autocommit as 0.  
mysql> SET autocommit = 0;
- BEGIN / START TRANSACTION command is used to start the transaction.
- mysql> START TRANSACTION;



**Smit IT Institute**  
Gives you wings to fly high



# TCL Commands in SQL

- 1. COMMIT
- COMMIT command in SQL is used to save all the transaction-related changes permanently to the disk. Whenever DDL commands such as INSERT, UPDATE and DELETE are used, the changes made by these commands are permanent only after closing the current session. So before closing the session, one can easily roll back the changes made by the DDL commands. Hence, if we want the changes to be saved permanently to the disk without closing the session, we will use the commit command.
- Syntax: COMMIT;







# TCL Commands in SQL

- 2. SAVEPOINT
- We can divide the database operations into parts. For example, we can consider all the insert related queries that we will execute consecutively as one part of the transaction and the delete command as the other part of the transaction. Using the SAVEPOINT command in SQL, we can save-these different parts of the same transaction using different names.
- **Syntax:** SAVEPOINT savepoint\_name;





# TCL Commands in SQL

- 3. ROLLBACK
- While carrying a transaction, we must create savepoints to save different parts of the transaction. According to the user's changing requirements, he/she can roll back the transaction to different savepoints.
- using the *ROLLBACK* command to the savepoint INS, which we have created before executing the DELETE query.
- Syntax: **ROLLBACK TO** savepoint\_name;





**Smit IT Institute**

Gives you wings to fly high

## Important Questions

### **Describe the difference between MySQL and SQL.**

MySQL is a database management system, whereas SQL is the language used for interacting with data in a database. MySQL implements SQL standards for querying and manipulating data, while SQL itself is a standardized language for managing relational databases.

### **Explain the use of the SELECT statement in MySQL.**

The SELECT statement in MySQL is used to retrieve data from one or more tables in a database. SELECT allows specifying columns to display, filtering conditions, and other parameters to tailor the data output.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

## What are the basic data types in MySQL?

MySQL supports several basic data types, including INT for integers, VARCHAR for variable-length strings, DATE for dates, and FLOAT for floating-point numbers. These data types allow for efficient storage and querying of different forms of data.

## How do you create a database in MySQL?

To create a database in MySQL, use the **CREATE DATABASE** statement followed by the desired database name. This command initializes a new database, preparing it for storing tables and data.

## What is the purpose of the WHERE clause in MySQL?

The WHERE clause in MySQL is used to filter records based on specified conditions. The WHERE clause allows selective retrieval of data from tables by applying criteria to columns.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

### **Explain the use of INSERT INTO statement?**

The INSERT INTO statement in MySQL is used to add new rows of data to a table. The INSERT INTO statement specifies the table to insert into and the values for each column in the new row.

### **How do you update data in a MySQL database?**

Data in a MySQL database is updated using the UPDATE statement, which alters existing records in a table. UPDATE specifies the table, the columns to update, the new values, and optionally, a WHERE clause to limit the rows affected.

### **What is a primary key in MySQL?**

A primary key in MySQL is a unique identifier for each record in a database table. A primary key ensures that each row can be distinctly identified, preventing duplicate entries in the key column.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

### **How do you delete data from a MySQL table?**

To delete data from a MySQL table, use the DELETE FROM statement along with a WHERE clause specifying which records to remove. This command permanently removes the specified rows from the table.

### **Describe the significance of the JOIN statement in MySQL.**

The JOIN statement in MySQL is crucial for combining rows from two or more tables based on a related column. The JOIN statement enables the creation of complex queries and the retrieval of related data from multiple tables.

### **What is normalization in MySQL?**

Normalization in MySQL is the process of organizing database tables and their relationships to minimize redundancy and dependency. Normalization in MySQL enhances database efficiency and integrity by dividing data into logical groupings.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

### **Explain the use of the GROUP BY clause.**

The GROUP BY clause in MySQL is used to group rows that have the same values in specified columns. The GROUP BY clause in MySQL is used in conjunction with aggregate functions like COUNT, SUM, or AVG to perform calculations on grouped data.

### **What is a foreign key in MySQL?**

A foreign key in MySQL is a field in one table that links to the primary key in another table. A foreign key in MySQL establishes a relationship between the tables and enforces referential integrity.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

### **How do you view the structure of a MySQL table?**

To view the structure of a MySQL table, use the DESCRIBE statement followed by the table name. This command displays the column details, data types, and other structural information of the table.

### **Explain the different types of joins in MySQL.**

Different types of joins in MySQL include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN. INNER JOIN returns rows with matching values in both tables, LEFT JOIN includes all rows from the left table and matched rows from the right table, RIGHT JOIN does the opposite, and FULL OUTER JOIN combines the results of both LEFT and RIGHT JOINS.



**Smit IT Institute**  
Gives you wings to fly high





**Smit IT Institute**

Gives you wings to fly high

## **What is indexing and why is it important in MySQL?**

Indexing in MySQL is a process of creating an index, a data structure that improves the speed of data retrieval. Indexing is important as it significantly reduces the amount of data scanned and speeds up query performance.

## **How do you implement subqueries in MySQL?**

Implementing subqueries in MySQL involves placing a SELECT query within another SQL query. Subqueries can be used in various clauses like SELECT, FROM, WHERE, and HAVING to perform complex queries.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

## **Describe the process of normalization and its types.**

The process of normalization in MySQL involves organizing data to reduce redundancy and improve data integrity. Its types include First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), and Boyce-Codd Normal Form (BCNF), each progressively reducing data redundancy and dependency.

## **How does transaction control work in MySQL?**

Transaction control in MySQL involves managing the changes made by DML statements like INSERT, UPDATE, and DELETE. Transaction control in MySQL ensures data integrity and consistency through transactions that are atomic, consistent, isolated, and durable (ACID).



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

### **What is a stored procedure and how is it used?**

A stored procedure in MySQL is a set of SQL statements saved in the database. A stored procedure in MySQL is used to encapsulate complex operations, allowing reuse and better performance, as the query plan is cached after the first execution.

### **Discuss the differences between CHAR and VARCHAR data types.**

The difference between CHAR and VARCHAR data types in MySQL lies in storage and retrieval. CHAR is a fixed-length type, faster for retrieval but may waste storage space, while VARCHAR is a variable-length type, more storage-efficient but slightly slower in retrieval.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

## **What is a view in MySQL and how is it created?**

A view in MySQL is a virtual table based on a SELECT query. A view in MySQL is created using the CREATE VIEW statement and simplifies complex queries, enhances security, and provides a logical view of data.

## **Explain the use of the HAVING clause.**

The HAVING clause in MySQL is used with the GROUP BY clause to filter group results. The HAVING clause specifies conditions for aggregated data, allowing filtering on the grouped records.

## **How do you optimize a MySQL query?**

Optimizing a MySQL query involves techniques like using appropriate indexes, optimizing joins, limiting the use of wildcards, and avoiding unnecessary columns in SELECT statements. These practices enhance performance by reducing the query execution time.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

## **What are the ACID properties in MySQL?**

The ACID properties in MySQL ensure reliable processing of database transactions. They stand for Atomicity, Consistency, Isolation, and Durability, guaranteeing that transactions are processed reliably and protecting the integrity of the database.

## **How is data integrity ensured in MySQL?**

Data integrity in MySQL is ensured through constraints like PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, and CHECK. These constraints enforce the correctness and consistency of data in the database.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

## **What is the difference between DELETE and TRUNCATE commands?**

The difference between DELETE and TRUNCATE commands in MySQL is that DELETE removes rows one at a time and logs each deletion, allowing rollback, whereas TRUNCATE removes all rows in a table without logging individual row deletions, making it faster but irreversible.

## **How do you handle errors in MySQL procedures?**

Handling errors in MySQL procedures involves using the DECLARE handler statement to define conditions and actions to take when specific errors occur. Handling errors in MySQL ensures robust and error-resistant stored procedures.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

## **Describe the use of LIMIT and OFFSET in MySQL.**

The use of LIMIT and OFFSET in MySQL specifies the number of rows to return and the starting point in a result set. LIMIT defines the count of rows to return, and OFFSET specifies the number of rows to skip before starting to return rows.

## **What are MySQL transactions and how are they managed?**

MySQL transactions are sequences of SQL operations treated as a single logical unit. MySQL transactions are managed using statements like START TRANSACTION, COMMIT, and ROLLBACK to ensure data integrity and consistency.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

## 1. lpad() : Count starts from right towards left direction.

```
mysql> SELECT LPAD('APPLE',9,'%') FROM DUAL;
```

```
+-----+  
| LPAD('APPLE',9,'%') |  
+-----+  
| %%%%APPLE          |  
+-----+  
1 row in set (0.04 sec)
```



**Smit IT Institute**  
Gives you wings to fly high





**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

**2.rpad(): count starts from left towards right direction.**

```
mysql> SELECT RPAD('APPLE',9,'%') FROM DUAL;
```

```
+-----+  
| RPAD('APPLE',9,'%') |  
+-----+  
| APPLE%%%%%          |  
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> SELECT LPAD(RPAD('APPLE',9,'&'),13,'#') FROM DUAL;
```

```
+-----+
| LPAD(RPAD('APPLE',9,'&'),13,'#') |
+-----+
| #####APPLE&&&&                |
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

```
mysql> SELECT RPAD('FLOWER',12,'@') FROM DUAL;
```

```
+-----+  
| RPAD('FLOWER',12,'@') |  
+-----+  
| FLOWER@@@@@@@      |  
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> SELECT LPAD('FLOWER@@@@@',16,'$') FROM DUAL;
```

```
+-----+
```

```
| LPAD('FLOWER@@@@@',16,'$') |
```

```
+-----+
```

```
| $$$$FLOWER@@@@@      |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> SELECT LPAD(RPAD('FLOWER',12,'@'),16,'$') FROM DUAL;
```

```
+-----+
| LPAD(RPAD('FLOWER',12,'@'),16,'$') |
+-----+
| $$$$FLOWER@@@@@@@                |
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

-----  
TASK:

1.###SWEET!!!!!!!!!!

2. HELLO\*\*\*\*\*

3.&&&&&HELLO\$\$\$

4.%%%%%%JAVA

---



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

## **lower()—Returns Data in Lower Case**

```
mysql> SELECT LOWER('GRASS') FROM DUAL;
```

```
+-----+
```

```
| LOWER('GRASS') |
```

```
+-----+
```

```
| grass          |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> SELECT LOWER('grass') FROM DUAL;
```

```
+-----+
```

```
| LOWER('grass') |
```

```
+-----+
```

```
| grass      |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
-----
```



**Smit IT Institute**

Gives you wings to fly high





**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

## UPPER()—Returns Data In Upper Case

```
mysql> SELECT UPPER('ABCDE') from dual;
```

```
+-----+  
| UPPER('ABCDE') |
```

```
+-----+  
| ABCDE          |
```

```
+-----+
```

1 row in set (0.00 sec)



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> SELECT UPPER(1234) from dual;
```

```
+-----+
```

```
| UPPER(1234) |
```

```
+-----+
```

```
| 1234      |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
*****
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> SELECT LOWER(1234) from dual;
```

```
+-----+
```

```
| LOWER(1234) |
```

```
+-----+
```

```
| 1234      |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
*****
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

```
mysql> SELECT * FROM ED_RECORD;
```

Will Display the complete Table.

**WORKING ON TABLE COLUMN:**

```
mysql> SELECT NAME FROM ED_RECORD;
```

Will Display Name from the Record.

```
mysql> SELECT LOWER(NAME) FROM ED_RECORD;
```

Try the Output:

```
mysql> SELECT UPPER(NAME) FROM ED_RECORD;
```

Try the Output:

```
mysql> SELECT NAME FROM ED_RECORD;
```

Will Display Names from the record.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> SELECT LOWER(NAME) FROM ED_RECORD;
```

Try the Output:

```
mysql> SELECT UPPER(NAME) FROM ED_RECORD;
```

Try the Output:

```
mysql> SELECT LPAD(NAME,'10','#') FROM ED_RECORD;
```

Try the Output:

```
mysql> SELECT RPAD(NAME,'10','$') FROM ED_RECORD;
```

Try the Output:

```
mysql> SELECT LPAD(RPAD(NAME,'10','#'),20,'*') FROM ED_RECORD;
```

Try the Output:



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

```
mysql> SELECT NAME FROM ED_RECORD;
```

**Will Display the Names of the Employees.**

## **TRIM FUNCTION:**

1.Its a dual function.

2. syntax:

```
SELECT TRIM('CHARACTER' FROM 'STRING') FROM DUAL;
```

3.STARTING->LEADING.

4.END->TRAILING.

5.TRIM() removes the character from leading and trailing.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

\*In Order to trim only the first character we can specify keyword as 'LEADING' IN THE SYNTAX.

```
mysql> SELECT TRIM(LEADING 'A' FROM 'ASHIYANA') FROM DUAL;
```

```
+-----+
| TRIM(LEADING 'A' FROM 'ASHIYANA') |
+-----+
| SHIYANA                          |
+-----+
```

1 row in set (0.00 sec)

\*In Order to trim only the LAST character we can specify keyword as 'TRAILING' IN THE SYNTAX:



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> SELECT TRIM(TRAILING 'A' FROM 'ASHIYANA') FROM DUAL;
```

```
+-----+
```

```
| TRIM(TRAILING 'A' FROM 'ASHIYANA') |
```

```
+-----+
```

```
| ASHIYAN          |
```

```
+-----+
```

```
-----
```



**Smit IT Institute**

Gives you wings to fly high





**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

**BOTH KEYWORD TARGETS 'LEADING' AS WELL AS 'TRAILING'.**

```
mysql> SELECT TRIM(BOTH 'A' FROM 'ASHIYANA') FROM DUAL;
```

```
+-----+  
| TRIM(BOTH 'A' FROM 'ASHIYANA') |  
+-----+  
| SHIYAN                        |  
+-----+
```

1 row in set (0.00 sec)



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

## WORKING ON TABLE COLUMN:

```
mysql> SELECT NAME FROM ED_RECORD;
```

Will Display The Names Of Employees.

```
mysql> SELECT TRIM('S' FROM NAME) FROM ED_RECORD;
```

Try the Output:

```
mysql> SELECT TRIM(TRAILING 'A' FROM NAME) FROM ED_RECORD;
```

Try the Output:



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

**\* TRIM() CONTINUES TO TRIM UNTIL IT FINDS A NEW ELEMENT**

```
mysql> SELECT TRIM('-' FROM '-----A')FROM DUAL;
```

```
+-----+  
| TRIM('-' FROM '-----A') |  
+-----+  
| A                |  
+-----+  
1 row in set (0.00 sec)
```



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

```
mysql> SELECT TRIM('-' FROM '-----A-----')FROM DUAL;
```

```
+-----+  
| TRIM('-' FROM '-----A-----') |  
+-----+  
| A                                |  
+-----+
```

1 row in set (0.00 sec)



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> SELECT TRIM('S' FROM 'SSSSSSSSSURESH')FROM DUAL;
```

```
+-----+
```

```
| TRIM('S' FROM 'SSSSSSSSSURESH') |
```

```
+-----+
```

```
| URESH                |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> SELECT TRIM('S' FROM 'SSSSSSSSURESHSSSSSSSSSS')FROM DUAL;
```

```
+-----+
```

```
| TRIM('S' FROM 'SSSSSSSSURESHSSSSSSSSSS') |
```

```
+-----+
```

```
| URESH          |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

```
mysql> SELECT TRIM(LEADING 'S' FROM 'SSSSSSSSSURESHSSSSSSSSSS')FROM DUAL;
```

```
+-----+  
| TRIM(LEADING 'S' FROM 'SSSSSSSSSURESHSSSSSSSSSS') |  
+-----+  
| URESHSSSSSSSSSS                                     |  
+-----+
```

1 row in set (0.00 sec)



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

```
mysql> SELECT TRIM(TRAILING 'S' FROM 'SSSSSSSSSURESHSSSSSSSSSS')FROM DUAL;
```

```
+-----+  
| TRIM(TRAILING 'S' FROM 'SSSSSSSSSURESHSSSSSSSSSS') |  
+-----+  
| SSSSSSSSURESH                                     |  
+-----+
```

```
1 row in set (0.00 sec)
```

---



**Smit IT Institute**  
Gives you wings to fly high





**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

**REVERSE(): IT RETURNS THE REVERSE ORDER OF THE GIVEN CONTENT:**

```
mysql> SELECT REVERSE('TOMATO') FROM DUAL;
```

```
+-----+  
| REVERSE('TOMATO') |
```

```
+-----+  
| OTAMOT          |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

```
mysql> SELECT REVERSE(NAME) FROM ED_RECORD;
```

Try The Output

```
mysql> SELECT REVERSE(123456789) FROM DUAL;
```

```
+-----+
```

```
| REVERSE(123456789) |
```

```
+-----+
```

```
| 987654321      |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

```
*****
```



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

**floor(): IT Return the closest base integer value.**

```
mysql> select floor(4.9) from dual;
```

```
+-----+  
| floor(4.9) |  
+-----+  
|      4      |  
+-----+
```

1 row in set (0.00 sec)



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> select floor(-4.9) from dual;
```

```
+-----+  
| floor(-4.9) |  
+-----+  
|      -5 |  
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

**ceil(): IT Return the closes highest integer value.**

```
mysql> select ceil(4.1) from dual;
```

```
+-----+
```

```
| ceil(4.1) |
```

```
+-----+
```

```
|      5 |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> select ceil(-4.9) from dual;
```

```
+-----+
```

```
| ceil(-4.9) |
```

```
+-----+
```

```
|      -4 |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

**round():** IT Returns the rounded value of the given decimal value with respect to “5 decimal value”.  
The second parameter after comma specifies the number of decimal values to be displayed after decimal.

```
mysql> select round(479375.12345,1)from dual;
```

```
+-----+  
| round(479375.12345,1) |
```

```
+-----+  
|      479375.1 |
```

```
+-----+  
1 row in set (0.00 sec)
```



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> select round(479375.12345,2)from dual;
```

```
+-----+  
| round(479375.12345,2) |  
+-----+  
|          479375.12 |  
+-----+
```

1 row in set (0.00 sec)



**Smit IT Institute**

Gives you wings to fly high





**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> select round(479375.12345,3)from dual;
```

```
+-----+  
| round(479375.12345,3) |  
+-----+  
|      479375.123      |  
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

**Negative value after comma rounds up the value close the decimal before the decimal.**

```
mysql> select round(479375.12345,-1)from dual;
```

```
+-----+  
| round(479375.12345,-1) |  
+-----+  
|          479380 |  
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> select round(479375.12345,-2)from dual;
```

```
+-----+  
| round(479375.12345,-2) |  
+-----+  
|          479400 |  
+-----+
```

1 row in set (0.00 sec)

```
mysql> select round(479375.12345,-3)from dual;
```

```
+-----+  
| round(479375.12345,-3) |  
+-----+  
|          479000 |  
+-----+
```

1 row in set (0.00 sec)



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> select round(4732.98375) from dual;
```

```
+-----+  
| round(4732.98375) |  
+-----+  
|          4733 |  
+-----+
```

1 row in set (0.00 sec)

```
mysql> select round(4732.98375,1) from dual;
```

```
+-----+  
| round(4732.98375,1) |  
+-----+  
|          4733.0 |  
+-----+
```

1 row in set (0.00 sec)



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

```
mysql> select round(4732.98375,2) from dual;
```

```
+-----+  
| round(4732.98375,2) |  
+-----+  
|      4732.98      |  
+-----+
```

1 row in set (0.00 sec)

## Task:

Referring to ED\_RECORD Table.

- 1.Find the Average Salary of every department and round it up.
- 2.Find the Average commission of every department and display only 2 digits after decimal.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

**substr(): IT Returns the sub part of the targeted String.**

Syntax:

substr("String",Starting\_Point,No. of Characters) ;

```
mysql> select substr("SoftwareTraining",1,4) from dual;
```

```
+-----+  
| substr("SoftwareTraining",1,4) |  
+-----+  
| Soft                          |  
+-----+
```

1 row in set (0.00 sec)



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

```
mysql> select substr("SoftwareTraining",3,9) from dual;
```

```
+-----+  
| substr("SoftwareTraining",3,9) |  
+-----+  
| ftwareTra          |  
+-----+  
1 row in set (0.00 sec)
```



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Dual Functions

## Displaying the First Half of String

```
mysql> select substr("SoftwareTraining",1,length("SoftwareTraining")/2) from dual;
```

```
+-----+
| substr("SoftwareTraining",1,length("SoftwareTraining")/2) |
+-----+
| Software                                     |
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**

Gives you wings to fly high





**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

## Displaying the Second Half of String

```
mysql> select substr("SoftwareTraining",length("SoftwareTraining")/2+1) from dual;
```

```
+-----+  
| substr("SoftwareTraining",length("SoftwareTraining")/2+1) |  
+-----+  
| Training                                     |  
+-----+
```

```
1 row in set (0.00 sec)
```



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# Dual Functions

**\*Number of characters in parameter is optional.**

```
mysql> select substr("SoftwareTraining",4) from dual;
```

```
+-----+  
| substr("SoftwareTraining",4) |  
+-----+  
| twareTraining                |  
+-----+
```

1 row in set (0.00 sec)

## Task: Referring to ED\_RECORD

1.Display the first half of Employee Names and Second Half of the Designation.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Constraints

## 1). Define Constraints?

A constraint is a table column property which performs data validation. Using constraints, you can maintain data integrity by preventing invalid data from being entered.

## 2). What do you understand by Data integrity?

Data integrity is the consistency and accuracy of the data which is stored in a database.

## 3). Can you add constraints to a table that already has data?

Yes, But it also depends on data, like if a column containing null values and you adding not null constraint then first you need to replace all null with some values.

## 4). How many primary keys can exist on a table?

One



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Constraints

## 5). What is a Foreign Key?

A FK in one table points to a PK in another table

It prevents any actions that would destroy links between tables with the corresponding data values

FK are used to enforce referential integrity

## 6). Difference between Primary key and Unique key constraints?

1) Unique constraints will allow a null value. If a field is null able then a unique constraint will allow at most one null value.

2) SQL server allows many unique constraints per table where it allows just primary key per table.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

**7). Can we apply Unique key constraints on multiple columns?**

Yes! Unique key constraints can be applied on a composite of multiple fields to ensure uniqueness of records.

Example : City + State in the StateList table

**8). When you create a Primary key constraints then by default which index will be created by DB?**

Clustered index would be created automatically when you will create a Primary key constraints.

**9). What do you understand by Default constraints?**

A default constraint enters a value in a column when one is not specified in the Insert or Update statement.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# Constraints

## **11). What are the type of data integrity?**

In relational database, there are three type of integrity

1. Domain Integrity( data type, check constraints)
2. Entity Integrity (primary key, unique constraints)
3. Referential Integrity (handled by foreign key constraints)

For Practice Kindly Refer the Constraints

PDF.



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# VIEW

## Why we use View?

MySQL view provides the following advantages to the user:

### Simplify complex query

It allows the user to simplify complex queries. If we are using the complex query, we can create a view based on it to use a simple SELECT statement instead of typing the complex query again.

### Increases the Re-usability

We know that View simplifies the complex queries and converts them into a single line of code to use VIEWS. Such type of code makes it easier to integrate with our application. This will eliminate the chances of repeatedly writing the same formula in every query, making the code reusable and more readable.



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# VIEW

## **Help in Data Security**

It also allows us to show only authorized information to the users and hide essential data like personal and banking information. We can limit which information users can access by authoring only the necessary data to them.

## **Enable Backward Compatibility**

A view can also enable the backward compatibility in legacy systems. Suppose we want to split a large table into many smaller ones without affecting the current applications that reference the table. In this case, we will create a view with the same name as the real table so that the current applications can reference the view as if it were a table.



**Smit IT Institute**  
Gives you wings to fly high





**Smit IT Institute**  
Gives you wings to fly high

# VIEW

## How is the table referred in memory using view?

Like other database management systems, MySQL lets you create views that enable users and applications to retrieve data without providing them direct access to the underlying tables. You can think of a view as a predefined query that MySQL runs when the view is invoked. MySQL stores the view definition as a database object, similar to a table object.

## What is the Syntax to create a View?

Syntax:

```
CREATE [OR REPLACE] VIEW view_name AS  
SELECT columns  
FROM tables  
[WHERE conditions];
```



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high

# VIEW

**Q: To create a view to refer the highest salary in ed\_record;**

```
mysql> create view optimized as select max(salary) from ed_record;  
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from optimized;
```

```
+-----+  
| max(salary) |  
+-----+  
|    155000 |  
+-----+
```

```
1 row in set (0.02 sec)
```



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**  
Gives you wings to fly high

# VIEW

**Q: To create a view to refer to the employees whose working location is not same as Gill?**

```
mysql> create view data as select name from ed_record where deptno in(select deptno  
from dept where loc not in(select loc from dept where deptno in(select deptno from  
ed_record where name="gill")));  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select * from data;  
Will Display all the employees except "Gill".
```



**Smit IT Institute**  
Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high



**Smit IT Institute**

Gives you wings to fly high