Structured Query Language (SQL)

Structured Query Language is a standard Database language which is used to create, maintain and retrieve the relational database.

What is SQL?

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

What Can SQL do?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

What is Relational Database?

Relational database means the data is stored as well as retrieved in the form of relations (tables). Table 1 shows the relational database with only one relation called **STUDENT** which stores **ROLL_NO**, **NAME**, **ADDRESS**, **PHONE** and **AGE** of students.

STUDENT

ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI	9156768971	18

These are some important terminologies that are used in terms of relation.

Attribute: Attributes are the properties that define a relation. e.g.; **ROLL_NO**, **NAME** etc.

Tuple: Each row in the relation is known as tuple. The above relation contains 4 tuples, one of which is shown as:

1	RAM	DELHI	9455123451	18

Degree: The number of attributes in the relation is known as degree of the relation. The **STUDENT** relation defined above has degree 5.

Cardinality: The number of tuples in a relation is known as cardinality. The **STUDENT** relation defined above has cardinality 4.

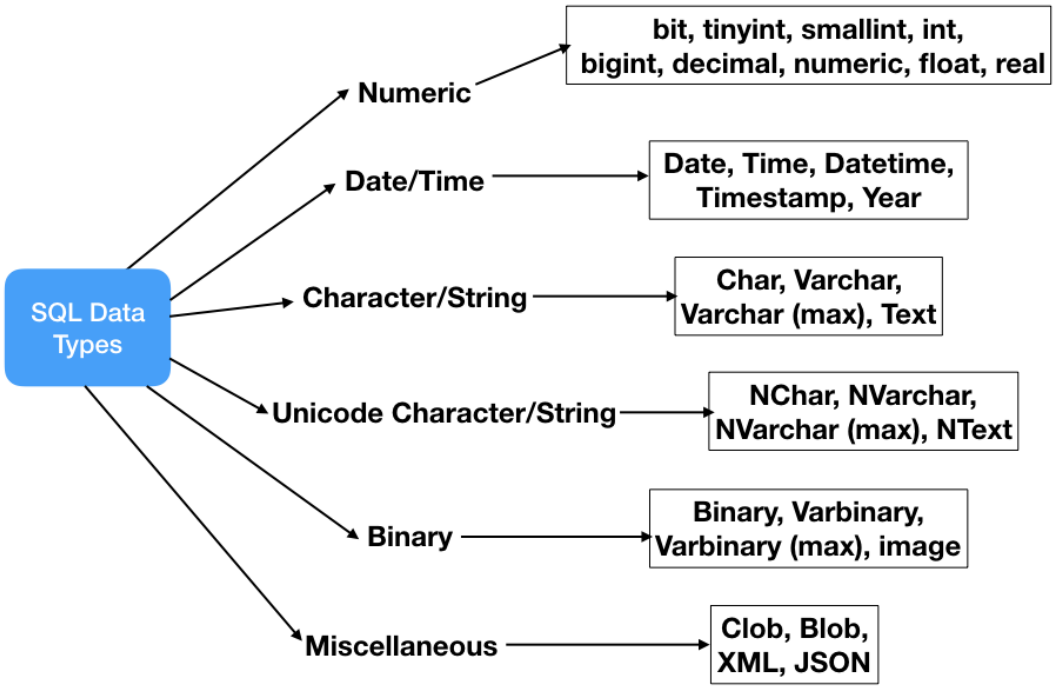
Column: Column represents the set of values for a particular attribute. The column **ROLL_NO** is extracted from relation STUDENT.

ROLL_NO
1
2
3
4

SQL Data Types

SQL data types can be broadly divided into following categories.

1. Numeric data types such as int, tinyint, bigint, float, real etc.
2. Date and Time data types such as Date, Time, Datetime etc.
3. Character and String data types such as char, varchar, text etc.
4. Unicode character string data types, for example nchar, nvarchar, ntext etc.
5. Binary data types such as binary, varbinary etc.
6. Miscellaneous data types – clob, blob, xml, cursor, table etc.



Let's look into different categories of sql data types in detail.

SQL Numeric Data Types

DATATYPE	FROM	TO
bit	0	1
tinyint	0	255

smallint	-32,768	32,767
int	-2,147,483,648	2,147,483,647
bigint	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
decimal	-10^38 +1	10^38 -1
numeric	-10^38 +1	10^38 -1
float	-1.79E + 308	1.79E + 308
real	-3.40E + 38	3.40E + 38

SQL Date and Time Data Types

DATATYPE	DESCRIPTION
DATE	Stores date in the format YYYY-MM-DD
TIME	Stores time in the format HH:MI:SS
DATETIME	Stores date and time information in the format YYYY-MM-DD HH:MI:SS
TIMESTAMP	Stores number of seconds passed since the Unix epoch (‘1970-01-01 00:00:00’ UTC)
YEAR	Stores year in 2 digit or 4 digit format. Range 1901 to 2155 in 4-digit format. Range 70 to 69, representing 1970 to 2069.

SQL Character and String Data Types

DATATYPE	DESCRIPTION
CHAR	Fixed length with maximum length of 8,000 characters
VARCHAR	Variable length storage with maximum length of 8,000 characters
VARCHAR(max)	Variable length storage with provided max characters, not supported in MySQL
TEXT	Variable length storage with maximum size of 2GB data

Note that all the above data types are for character stream, they should not be used with Unicode data.

SQL Unicode Character and String Data Types

DATATYPE	DESCRIPTION
NCHAR	Fixed length with maximum length of 4,000 characters
NVARCHAR	Variable length storage with maximum length of 4,000 characters
NVARCHAR(max)	Variable length storage with provided max characters
NTEXT	Variable length storage with maximum size of 1GB data

Note that above data types are not supported in MySQL database.

SQL Binary Data Types

DATATYPE	DESCRIPTION
BINARY	Fixed length with maximum length of 8,000 bytes
VARBINARY	Variable length storage with maximum length of 8,000 bytes
VARBINARY(max)	Variable length storage with provided max bytes
IMAGE	Variable length storage with maximum size of 2GB binary data

SQL Miscellaneous Data Types

DATATYPE	DESCRIPTION
CLOB	Character large objets that can hold up to 2GB

BLOB	For binary large objects
XML	for storing xml data
JSON	for storing JSON data

That’s all for a quick roundup on SQL data types.

Type of SQL Statements

Type of SQL statements are divided into five different categories: Data definition language (DDL), Data manipulation language (DML), Data Control Language (DCL), Transaction Control Statement (TCS), Session Control Statements (SCS).

Data Definition Language (DDL)

Data definition statement are use to define the database structure or table.

Statement	Description
CREATE	Create new database/table.
ALTER	Modifies the structure of database/table.
DROP	Deletes a database/table.
TRUNCATE	Remove all table records including allocated table spaces.
RENAME	Rename the database/table.

Data Manipulation Language (DML)

Data manipulation statement are use for managing data within table object.

Statement	Description
SELECT	Retrieve data from the table.
INSERT	Insert data into a table.
UPDATE	Updates existing data with new data within a table.
DELETE	Deletes the records rows from the table.
MERGE	MERGE (also called UPSERT) statements to INSERT new records or UPDATE existing records depending on condition matches or not.

LOCK TABLE	LOCK TABLE statement to lock one or more tables in a specified mode. Table access denied to a other users for the duration of your table operation.
CALL EXPLAIN PLAN	Statements are supported in PL/SQL only for executed dynamically. CALL a PL/SQL program or EXPLAIN PATH access the data path.

Data Control Language (DCL)

Data control statement are use to give privileges to access limited data.

Statement	Description
GRANT	Gives privileges to user for accessing database data.
REVOKE	Take back for given privileges.
ANALYZE	ANALYZE statement to collect statistics information about index, cluster, table.
AUDIT	To track the occurrence of a specific SQL statement or all SQL statements during the user sessions.
COMMENT	Write comment to the data table.

Transaction Control Statement (TCS)

Transaction control statement are use to apply the changes permanently save into database.

Statement	Description
COMMIT	Permanent work save into database.
ROLLBACK	Restore database to original form since the last COMMIT.
SAVEPOINT	Create SAVEPOINT for later use ROLLBACK the new changes.

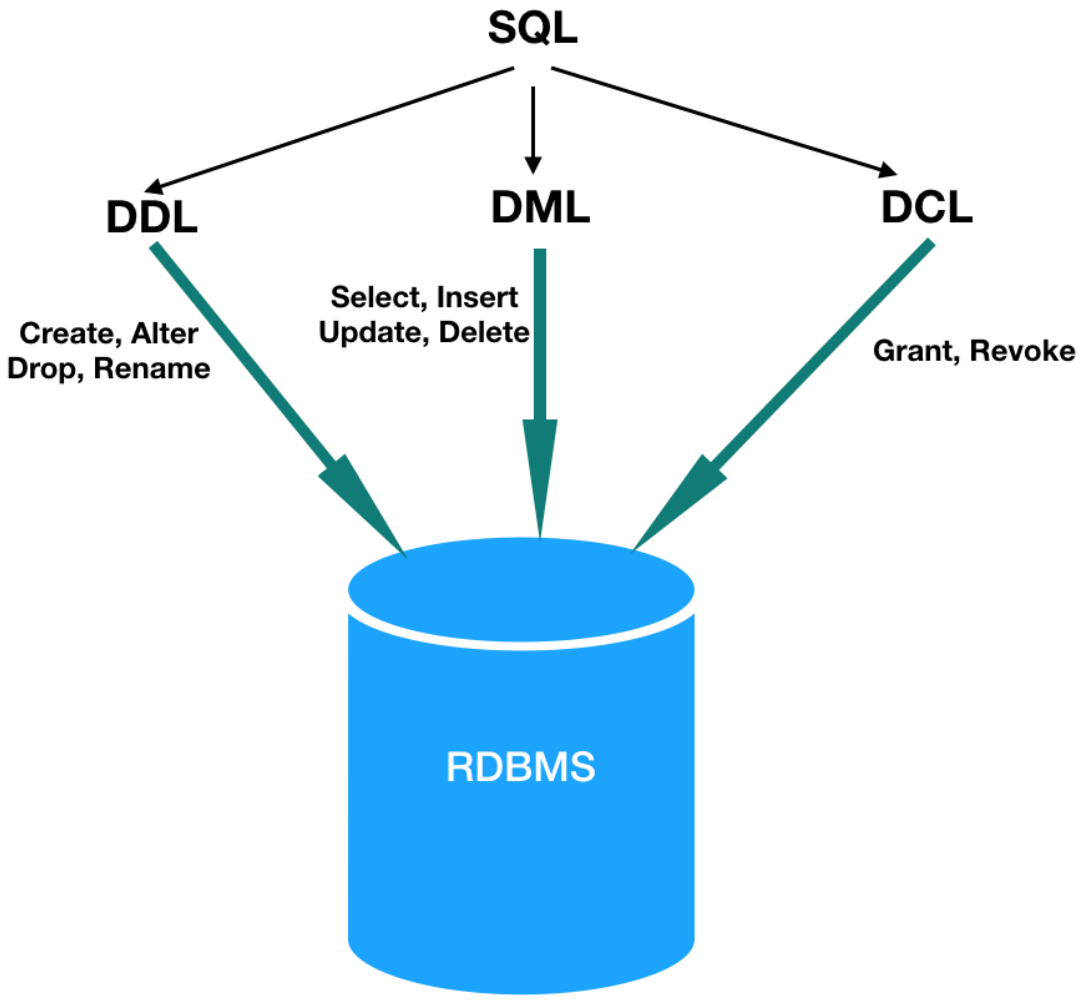
SET TRANSACTION	SET TRANSACTION command set the transaction properties such as read-write/read only access.

PL/SQL Transaction commit, rollback, savepoint, autocommit, Set Transaction read more.

Session Control Statement (SCS)

Session control statement are manage properties dynamically of a user session.

Statement	Description
ALTER SESSION	ALTER SESSION statement to modify conditions or parameters that are affect to your database connection.
SET ROLE	SET ROLE statement to enable or disable the roles that are currently enabled for the session.



SQL Syntax

SQL syntax differs a lot based on type of queries. For example, below is the general syntax for SQL select and insert queries.

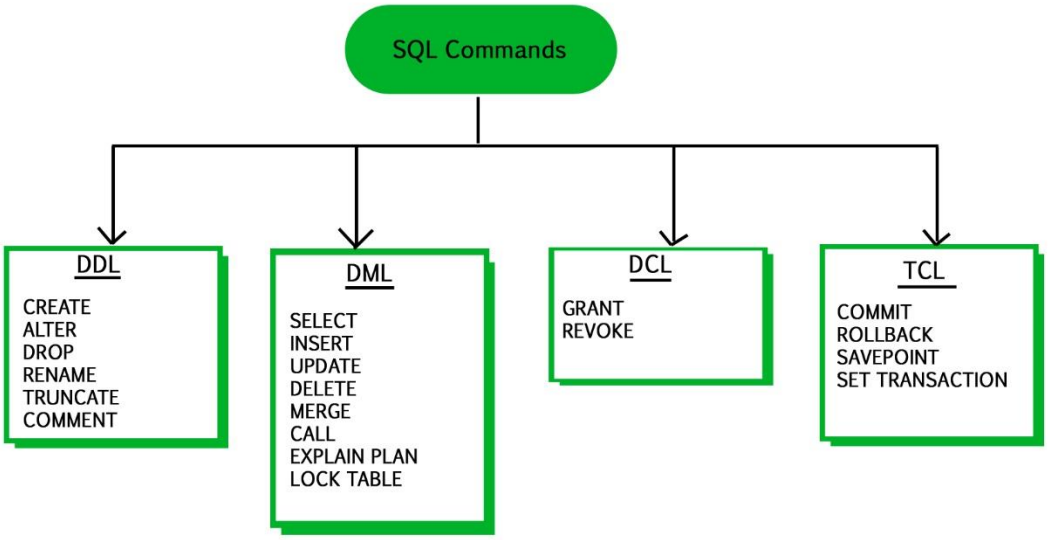
```
Select {fields} FROM {tables/views} WHERE {conditions}
```

```
INSERT INTO {table} ({column_names}) VALUES ({comma separated values})
```

We should be aware of some language specific terminologies.

1. Clause – SQL clauses are the building blocks of sql queries. For example in above syntax examples, Select, Insert, Where are the clauses.
2. Predicate – they are the conditions to limit the query results. In above example, condition in the where clause is called Predicate.
3. Queries – SQL statement is also called queries.

That's all for a quick roundup on SQL.



SQL Operators

- SQL operators are used to perform operations like comparisons and arithmetic operations.
- These Operators are used to specify conditions in an SQL statement.
- SQL operators help us in selecting only specific records from the tables or views.

SQL Operators Types

Broadly SQL operators are classified in following parts.

1. Arithmetic Operators
2. Comparison Operators
3. Logical Operators
4. Bitwise Operators

OPERATOR	DESCRIPTION	EXAMPLE
+ (Addition)	Adds values on both sides of the operator.	SELECT 30 + 20; Output: 50
-(Substraction)	Subtracts values on right side from the value on left side of the operator.	SELECT 30 – 20; Output: 10
*(Multiplication)	Multiplies the values on both sides of the operator	SELECT 30 * 20; Output: 600

/(Division)	Divides left hand side value by right hand side value.	SELECT 30 / 20; Output: 1
%(Modulus)	Divides left hand side value by right hand side value and returns the reminder	SELECT 30 % 20; Output: 10

Let's try to try to understand all the above-mentioned operators one by one.

SQL Arithmetic Operators

SQL Arithmetic operators are the operators which are used for mathematical calculation like addition, subtraction etc. They are used with SQL numeric data types.

SQL Comparison Operators

Comparison operators are the operators which are used for comparison between two values. To understand the comparison operator better, we will take example of Employee table as shown below.

Let's understand usage of comparison operators using the table above as an example.

EMPID	EMPNAME	EMPAGE	EMPSALARY
1	John	32	2000
2	Smith	25	2500

3	Henry	29	3000

OPERATOR	DESCRIPTION	EXAMPLE
= (Equal To)	Checks if the values of two operands are equal, if its equal then condition becomes true.	SELECT EmpName FROM Employee WHERE EmpSalary=2000; Output: John
!= (Not Equal To)	Checks if the values of two operands are not equal, if values are not equal then condition becomes true.	SELECT EmpName FROM Employee WHERE EmpSalary!=2000; Output: Smith Henry
<> (Not Equal To)	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	SELECT EmpName FROM Employee WHERE EmpSalary<>2000; Output: Smith Henry

> (Greater Than)	Checks if the value of left operand is greater than the value of right operand, condition becomes true if it is yes.	SELECT EmpName FROM Employee WHERE EmpSalary > 2000 Output: Smith Henry
< (Less Than)	Checks if the value of left operand is less than the value of right operand, condition becomes true if it is yes.	SELECT EmpName FROM Employee WHERE EmpSalary < 2000 Output: No Records Found
>= (Greater than or Equal To)	Checks if the value of left operand is greater than or equal to the value of right operand, condition becomes true if its yes.	SELECT EmpName FROM Employee WHERE EmpSalary >= 2000 Output: John Smith Henry
<=(Less than or Equal To)	Checks if the value of left operand is less than or equal to the value of right operand, condition becomes true if it is yes.	SELECT EmpName FROM Employee WHERE EmpSalary <= 2000 Output: John

!< (Not Less than)	Checks if the value of left operand is not less than the value of right operand, condition becomes true if it is yes.	SELECT EmpName FROM Employee WHERE EmpSalary !< 2000 Output: Smith Henry
!> (Not Greater Than)	Checks if the value of left operand is not greater than the value of right operand, condition becomes true if it is yes.	SELECT EmpName FROM Employee WHERE EmpSalary !> 2000 Output: – John

SQL Logical Operators

Logical operators are the operators which are used for logical operations. To understand the logical operator better, we will take example of Employee table as shown below.

EmpId	EmpName	EmpAge	EmpSalary
1	John	32	2000
2	Smith	25	2500
3	Henry	29	3000

Let's understand usage of logical operator using the table above as an example.

OPERATOR	DESCRIPTION	EXAMPLE

ALL	ALL operator is used to compare a value to all the values in another set of values.	SELECT EmpName FROM Employee WHERE EmpAge > ALL (SELECT EmpAge FROM Employee WHERE EmpSalary >= 2500); Output: John Smith
AND	AND operator allows the multiple conditions in an SQL statement's WHERE clause.	SELECT EmpName FROM Employee WHERE EmpSalary > 2000 and EmpAge > 28 Output: Henry
ANY	ANY operator is used to compare a value to any applicable value in the list based on the condition.	SELECT EmpName FROM Employee WHERE EmpAge > ANY (SELECT EmpAge FROM Employee WHERE EmpSalary >= 2500); Output: John Smith
BETWEEN	BETWEEN operator is used to search for values that are within a range, given the minimum value and the maximum value.	SELECT EmpName FROM Employee WHERE EmpAge BETWEEN 25 AND 30; Output:

		Smith Henry
EXISTS	EXISTS operator is used to search for the presence of a row in a specified table that meets a certain criterion.	SELECT EmpName FROM Employee WHERE EXISTS (SELECT EmpName FROM Employee WHERE EmpSalary >= 2500); Output: Smith Henry
IN	IN operator is used to compare a value to a list of literal values that have been specified.	SELECT EmpName FROM Employee WHERE EmpSalary IN (2000, 2500); Output: John Smith
LIKE	LIKE operator is used to compare a value to similar values using wildcard operators.	SELECT EmpName FROM Employee WHERE EmpName LIKE 'Jo%'; Output: John
NOT	NOT operator reverses the meaning of the logical operator with which it is used.	SELECT EmpName FROM Employee WHERE EmpSalary IS NOT NULL

		Output: John Smith Henry
OR	OR operator is used to combine multiple conditions in one SQL statement's WHERE clause.	SELECT EmpName FROM Employee WHERE EmpSalary > 2000 OR EmpName IS NOT NULL; Output: John Smith Henry
IS NULL	IS NULL operator is used to compare a value with a NULL value.	SELECT EmpName FROM Employee WHERE EmpSalary IS NULL; Output: No records found
UNIQUE	UNIQUE operator searches every row of a specified table for uniqueness	SELECT UNIQUE(EmpName) FROM Employee WHERE EmpSalary IS NOT NULL; Output: John Smith Henry

SQL Bitwise Operators

Bitwise operators are the operators which are used on bit of data.

OPERATOR	DESCRIPTION
&	Bitwise AND operator
|	Bitwise OR operator
^	Bitwise Exclusive OR operator
<<	Left Shift operator
>>	Right Shift operator

Here is a simple program showing usage of sql bitwise operators.

```
-- 27 = 11011
-- 19 = 10011

select 27 & 19; -- 10011
select 27 | 19; -- 11011
select 27 ^ 19; -- 00100
select 5 << 2; -- 101 to 10100 i.e. 20
select 17 >> 2; -- 10001 to 100 i.e. 4
```

That's all for SQL operators in a nutshell.

SQL KEY

A DBMS key is an attribute or set of an attribute which helps you to identify a row(tuple) in a relation(table). They allow you to find the relation between two tables. Keys help you uniquely identify a row in a table by a combination of one or more columns in that table.

Various Keys in Database Management System
DBMS has following seven types of Keys each have their different functionality:

- Super Key
- Primary Key
- Candidate Key
- Alternate Key
- Foreign Key
- Compound Key
- Composite Key
- Surrogate Key

What is the Super key?

A super key is a group of single or multiple keys which identifies rows in a table. A Super key may have additional attributes that are not needed for unique identification.

Example:

EmpSSN	EmpNum	Empname
9812345098	AB05	Shown
9876512345	AB06	Roslyn
199937890	AB07	James

In the above-given example, EmpSSN and EmpNum name are superkeys.

What is a Primary Key?

A column or group of columns in a table which helps us to uniquely identifies every row in that table is called a primary key. This DBMS can't be a duplicate. The same value can't appear more than once in the table.

Rules for defining Primary key:

- Two rows can't have the same primary key value

- It must for every row to have a primary key value.
- The primary key field cannot be null.
- A table can contain only one primary key constraint.
- The value in a primary key column can never be modified or updated if any foreign key refers to that primary key.

Example:

In the following example, <code>StudID</code> is a Primary Key.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

Main advantage of primary key:

The main advantage of this uniqueness is that we get **fast access**.

SQL primary key for one column:

The following SQL command creates a PRIMARY KEY on the "S_Id" column when the "students" table is created.

MySQL:

```
1. CREATE TABLE students
2. (
3. S_Id int NOT NULL,
4. LastName varchar (255) NOT NULL,
5. FirstName varchar (255),
6. Address varchar (255),
7. City varchar (255),
8. PRIMARY KEY (S_Id)
9. )
```

SQL primary key for multiple columns:

MySQL, SQL Server, Oracle, MS Access:

```
1. CREATE TABLE students
```

```
(
S_Id int NOT NULL,
LastName varchar (255) NOT NULL,
FirstName varchar (255),
Address varchar (255),
City varchar (255),
CONSTRAINT pk_StudentID PRIMARY KEY (S_Id, LastName)
)
```

Note:you should note that in the above example there is only one PRIMARY KEY (pk_StudentID). However it is made up of two columns (S_Id and LastName).

SQL primary key on ALTER TABLE

When table is already created, and you want to create a PRIMARY KEY constraint on the "S_Id" column you should use the following SQL:

Primary key on one column:

```
ALTER TABLE students
ADD PRIMARY KEY (S_Id)
```

Primary key on multiple column:

```
ALTER TABLE students
ADD CONSTRAINT pk_StudentID PRIMARY KEY (S_Id,LastName)
```

When you use ALTER TABLE statement to add a primary key, the primary key columns must not contain NULL values (when the table was first created).

How to DROP a PRIMARY KEY constraint?

If you want to DROP (remove) a primary key constraint, you should use following syntax:

MySQL:

```
ALTER TABLE students
DROP PRIMARY KEY
```

SQL Server / Oracle / MS Access:

```
ALTER TABLE students
DROP CONSTRAINT pk_StudentID
```

What is the Alternate key?

All the keys which not primary key are called an alternate key. It is a candidate key which is currently not the primary key. However, A table may have single or multiple choices for the primary key.

Alternate key is a secondary key it can be simple to understand by an example:

Let's take an example of student it can contain NAME, ROLL NO., ID and CLASS.

Here ROLL NO. is primary key and rest of all columns like NAME, ID and CLASS are alternate keys.

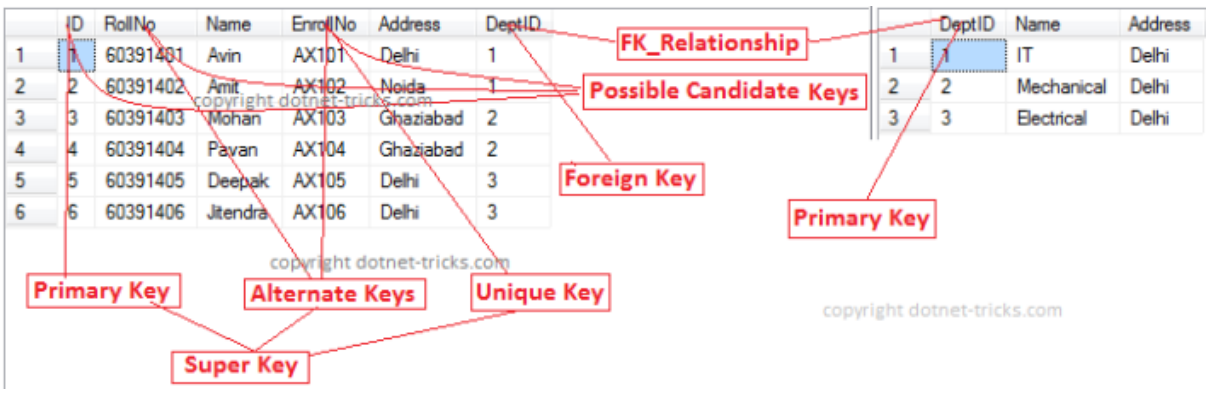
If a table has more than one candidate key, one of them will become the primary key and rest of all are called alternate keys.

In simple words, you can say that any of the candidate key which is not part of primary key is called an alternate key. So when we talk about alternate key, the column may not be primary key but still it is a unique key in the column.

Example: In this table.

StudID, Roll No, Email are qualified to become a primary key. But since StudID is the primary key, Roll No, Email becomes the alternative key.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com



What is a Candidate Key?

A super key with no repeated attribute is called candidate key.

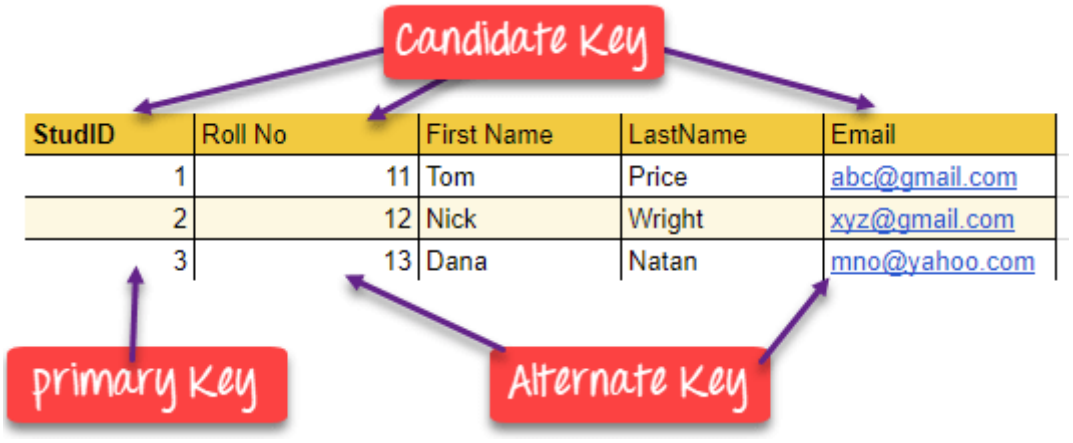
The Primary key should be selected from the candidate keys. Every table must have at least a single candidate key.

Properties of Candidate key:

- It must contain unique values
- Candidate key may have multiple attributes
- Must not contain null values
- It should contain minimum fields to ensure uniqueness
- Uniquely identify each record in a table

Example: In the given table Stud ID, Roll No, and email are candidate keys which help us to uniquely identify the student record in the table.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com



What is the Foreign key?

In the relational databases, a foreign key is a field or a column that is used to establish a link between two tables.

In simple words you can say that, a foreign key in one table used to point primary key in another table.

Let us take an example to explain it:

Here are two tables first one is students table and second is orders table.

Here orders are given by students.

First table:

S_Id	LastName	FirstName	CITY
1	MAURYA	AJEET	ALLAHABAD
2	JAISWAL	RATAN	GHAZIABAD
3	ARORA	SAUMYA	MODINAGAR

Second table:

O_Id	OrderNo	S_Id
1	99586465	2
2	78466588	2
3	22354846	3
4	57698656	1

Here you see that "S_Id" column in the "Orders" table points to the "S_Id" column in "Students" table.

- The "S_Id" column in the "Students" table is the PRIMARY KEY in the "Students" table.
- The "S_Id" column in the "Orders" table is a FOREIGN KEY in the "Orders" table.

The foreign key constraint is generally prevents action that destroy links between tables.

It also prevents invalid data to enter in foreign key column.

SQL FOREIGN KEY constraint ON CREATE TABLE:

(Defining a foreign key constraint on single column)
To create a foreign key on the "S_Id" column when the "Orders" table is created:
MySQL:

```
CREATE TABLE orders
(
O_Id int NOT NULL,
Order_No  int NOT NULL,
S_Id int,
PRIMAY KEY (O_Id),
FOREIGN KEY (S_Id) REFERENCES Persons (S_Id)
)
```

SQL Server /Oracle / MS Access:

```
CREATE TABLE Orders
(
O_Id int NOT NULL PRIMAY KEY,
Order_No int NOT NULL,
S_Id int FOREIGN KEY REFERENCES persons (S_Id)
)
```

SQL FOREIGN KEY constraint for ALTER TABLE:

If the Order table is already created and you want to create a FOREIGN KEY constraint on the "S_Id" column, you should write the following syntax:

Defining a foreign key constraint on single column:

MySQL / SQL Server / Oracle / MS Access:

```
ALTER TABLE Orders
ADD CONSTRAINT fk_PerOrders
FOREIGN KEY(S_Id)
REFERENCES Students (S_Id)
```

DROP SYNTAX for FOREIGN KEY COSTRAINT:
If you want to drop a FOREIGN KEY constraint, use the following syntax:

MySQL:

```
ALTER TABLE Orders
ROP FOREIGN KEY fk_PerOrders
```

SQL Server / Oracle / MS Access:

```
ALTER TABLE Orders
DROP CONSTRAINT fk_PerOrders
```

Difference Between Primary key & Foreign key

Primary Key	Foreign Key
Helps you to uniquely identify a record in the table.	It is a field in the table that is the primary key of another table.
Primary Key never accept null values.	A foreign key may accept multiple null values.
Primary key is a clustered index and data in the DBMS table are physically organized in the sequence of the clustered index.	A foreign key cannot automatically create an index, clustered or non-clustered. However, you can manually create an index on the foreign key.
You can have the single Primary key in a table.	You can have multiple foreign keys in a table.

What is the Compound key?

Compound key has many fields which allow you to uniquely recognize a specific record. It is possible that each column may be not unique by itself within the database. However, when combined with the other column or columns the combination of composite keys become unique.

Example:

OrderNo	PorductID	Product Name	Quantity
B005	JAP102459	Mouse	5
B005	DKT321573	USB	10
B005	OMG446789	LCD Monitor	20
B004	DKT321573	USB	15
B002	OMG446789	Laser Printer	3

In this example, OrderNo and ProductID can't be a primary key as it does not uniquely identify a record. However, a compound key of Order ID and Product ID could be used as it uniquely identified each record.

What is the Composite key?

A key which has multiple attributes to uniquely identify rows in a table is called a composite key. The difference between compound and the composite key is that any part of the compound key can be a foreign key, but the composite key may or maybe not a part of the foreign key.

What is a Surrogate Key?

An artificial key which aims to uniquely identify each record is called a surrogate key. These kind of key are unique because they are created when you don't have any natural primary key. They do not lend any meaning to the data in the table. Surrogate key is usually an integer.

Fname	Lastname	Start Time	End Time
Anne	Smith	09:00	18:00
Jack	Francis	08:00	17:00
Anna	McLean	11:00	20:00
Shown	Willam	14:00	23:00

Above, given example, shown shift timings of the different employee. In this example, a surrogate key is needed to uniquely identify each employee.

Surrogate keys are allowed when

- No property has the parameter of the primary key.
- In the table when the primary key is too big or complicated.

SQL BASIC QUERYS EXAMPLE

-- comment in sql
Single line comments start with --
Multi-line comments start with /* and end with */.

-- FOR DATABASE CREATION
CREATE DATABASE xyz;

SHOW DATABASES;

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| new_try            |
| newformation       |
| performance_schema |
| phpmyadmin         |
| test               |
| xyz                |
+--------------------+
```

USE newformation;
Database changed

SHOW TABLES; -- after creation of persons table by query.

```
+------------------------+
| Tables_in_newformation |
+------------------------+
| persons                |
+------------------------+
```

```
CREATE TABLE persons(
        id INT NOT NULL AUTO_INCREMENT,
        name VARCHAR(15) NOT NULL,
        city VARCHAR(20) NOT NULL,
        pincode INT NOT NULL,
        PRIMARY KEY(id)
);
```

Query OK, 0 rows affected (0.02 sec)

DESC persons;

```
+---------+-------------+------+-----+---------+----------------+
| Field   | Type        | Null | Key | Default | Extra          |
+---------+-------------+------+-----+---------+----------------+
| id      | int(11)     | NO   | PRI | NULL    | auto_increment |
| name    | varchar(15) | NO   |     | NULL    |                |
| city    | varchar(20) | NO   |     | NULL    |                |
| pincode | int(11)     | NO   |     | NULL    |                |
+---------+-------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

INSERT INTO persons(name,city,pincode) VALUES(

```
   "MK",'mira road',401108
);
Query OK, 1 row affected (0.00 sec)

INSERT INTO persons(name,city,pincode) VALUES(
   "Mayur Kadam","Bhayander",401107
);
Query OK, 1 row affected (0.00 sec)

SELECT * FROM persons;
+----+-------------+-----------+---------+
| id | name        | city      | pincode |
+----+-------------+-----------+---------+
|  1 | MK          | mira road |  401108 |
|  2 | Mayur Kadam | Bhayander |  401107 |
+----+-------------+-----------+---------+
2 rows in set (0.00 sec)

DROP TABLE persons;
Query OK, 0 rows affected (0.01 sec)

DROP DATABASE xyz;
Query OK, 0 rows affected (0.00 sec)

-- another way to DROP THE Table if its exists
DROP TABLE IF EXISTS marks;

-- similarely
DROP DATABASE IF EXISTS gangulytech;

/*---------------------------- gangulytech.sql -------------------------------*/

SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| gangulytech        |
| information_schema |
| mysql              |
| new_try            |
| newformation       |
| performance_schema |
| phpmyadmin         |
| test               |
+--------------------+
8 rows in set (0.02 sec)

USE gangulytech;
Database changed

show tables;
+----------------------+
| Tables_in_gangulytech |
+----------------------+
| courses              |
| enrolls              |
| marks                |
| students             |
```

```
+----------------------+
4 rows in set (0.00 sec)
```

select * from marks;

```
+----+------------------------------+-------+-----+
| id | course_name                  | score | sid |
+----+------------------------------+-------+-----+
|  1 | AI In Real World Using Python |    14 |  38 |
|  2 | AI In Real World Using Python |    99 |  51 |
|  3 | GIMP Photo Editing           |    51 |  29 |
|  4 | SQLite Tutorial              |    57 |  13 |
|  5 | Swing GUI In Depth           |    31 |  34 |
|  6 | CSS3                         |    83 |  36 |
|  7 | Codeigniter                  |    22 |  39 |
|  8 | 2D Games Using PyGame        |    60 |  37 |
|  9 | Data Structure In Depth      |    31 |  32 |
| 10 | Amazon Cloud AWS             |    76 |  12 |
| 11 | Amazon Cloud AWS             |    86 |  27 |
| 12 | Computer Networks            |     2 |  34 |
| 13 | Swing GUI In Depth           |    52 |   1 |
| 14 | Dynamic Website Development  |    52 |   3 |
| 15 | CakePHP                      |     2 |  27 |
| 16 | Natural Language Processing  |    55 |   6 |
| 17 | HTML                         |    68 |   6 |
| 18 | Computer Vision Using Python |    72 |  47 |
| 19 | Computer Vision Using Python |    58 |  24 |
| 20 | Data Structure In Depth      |    70 |  14 |
| 21 | Kali Linux                   |    76 |  18 |
| 22 | Java Complete Tutorial       |    67 |  32 |
| 23 | Java Database Connectivity   |     8 |  25 |
| 24 | The C Ninja                  |    39 |  10 |
| 25 | SQLite Tutorial              |    69 |  60 |
+----+------------------------------+-------+-----+
25 rows in set (0.01 sec)
```

select * from marks limit 5;

```
+----+------------------------------+-------+-----+
| id | course_name                  | score | sid |
+----+------------------------------+-------+-----+
|  1 | AI In Real World Using Python |    14 |  38 |
|  2 | AI In Real World Using Python |    99 |  51 |
|  3 | GIMP Photo Editing           |    51 |  29 |
|  4 | SQLite Tutorial              |    57 |  13 |
|  5 | Swing GUI In Depth           |    31 |  34 |
+----+------------------------------+-------+-----+
5 rows in set (0.00 sec)
```

select * from marks limit 9, 5;

```
+----+------------------------------+-------+-----+
| id | course_name                  | score | sid |
+----+------------------------------+-------+-----+
| 10 | Amazon Cloud AWS             |    76 |  12 |
| 11 | Amazon Cloud AWS             |    86 |  27 |
| 12 | Computer Networks            |     2 |  34 |
| 13 | Swing GUI In Depth           |    52 |   1 |
| 14 | Dynamic Website Development  |    52 |   3 |
+----+------------------------------+-------+-----+
5 rows in set (0.00 sec)
```

select distinct course_name from marks;

```
+-------------------------------+
| course_name                   |
+-------------------------------+
| AI In Real World Using Python |
| GIMP Photo Editing            |
| SQLite Tutorial               |
| Swing GUI In Depth            |
| CSS3                          |
| Codeigniter                   |
| 2D Games Using PyGame         |
| Data Structure In Depth       |
| Amazon Cloud AWS              |
| Computer Networks             |
| Dynamic Website Development   |
| CakePHP                       |
| Natural Language Processing   |
| HTML                          |
| Computer Vision Using Python  |
| Kali Linux                    |
| Java Complete Tutorial        |
| Java Database Connectivity    |
| The C Ninja                   |
+-------------------------------+
19 rows in set (0.00 sec)
```

select * from students;

```
+----+----------------------+------------+-------+---------+
| id | name                 | city       | state | pincode |
+----+----------------------+------------+-------+---------+
|  1 | sandeep ganguly      | kanpur     | UP    |  601988 |
|  2 | piyush chandel       | nainital   | UK    |  549386 |
|  3 | divyanshu shukla     | kanpur     | UP    |  940965 |
|  4 | ankita               | kanpur     | UP    |   56669 |
|  5 | brijesh gupta        | gorakhpur  | UP    |  460450 |
|  6 | siddhartha singh     | kanpur     | UP    |  132244 |
|  7 | parvez hasan         | faizabad   | UP    |  279869 |
|  8 | pawan kumar          | banglore   | KA    |    2612 |
|  9 | umesh verma          | kolkata    | WB    |  173453 |
| 10 | ayushi sharma        | jammu      | JK    |  859431 |
| 11 | shameem beg          | mumbai     | MH    |  776793 |
| 12 | Arun Bhatia          | pune       | MH    |  305673 |
| 13 | shiv patel           | surat      | GJ    |  197988 |
| 14 | aman ali             | ajmer      | RJ    |   72920 |
| 15 | varsha singh         | mathura    | UP    |  770636 |
| 16 | deepak yadav         | gurugram   | HR    |  634419 |
| 17 | manjul saini         | dhanbad    | JH    |  860186 |
| 18 | Ankur sharma         | ranchi     | JH    |  397676 |
| 19 | saurabh gupta        | ahemdabad  | GJ    |  407819 |
| 20 | soumya pandey        | srinagar   | JK    |  846069 |
| 21 | digvijay patel       | jamnagar   | GJ    |    6887 |
| 22 | shivani singh        | faridabad  | HR    |  496229 |
| 23 | sarvik roy           | purulia    | WB    |  460485 |
| 24 | mamta banerjee       | kolkata    | WB    |  813736 |
| 25 | dolly ganguly        | howrah     | WB    |  687224 |
| 26 | shubhojeet mukherjee | birbhum    | WB    |  994915 |
| 27 | shubham das          | jhargram   | WB    |  912900 |
| 28 | tapas paul           | bankura    | WB    |  579758 |
| 29 | sbhubendu sarkar     | kolkatta   | WB    |  160089 |
| 30 | kaveri bose          | howrah     | WB    |   61170 |
| 31 | mitali chatterjee    | kolkata    | WB    |  825583 |
| 32 | rupoma biswas        | howrah     | WB    |  944403 |
```

```
| 33 | sujeet ghara         | howrah     | WB    |  245270 |
| 34 | shweta ghara         | kolkata    | WB    |  393138 |
| 35 | nita ganguly         | kolkata    | WB    |  229879 |
| 36 | sumita ganguly       | kolkata    | WB    |  969984 |
| 37 | sumit thakrey        | mumbai     | MH    |  160281 |
| 38 | nana patekar         | mumbai     | MH    |  891455 |
| 39 | nitin gadkari        | nagpur     | MH    |  976432 |
| 40 | dharmesh pradhan     | latur      | MH    |  207796 |
| 41 | rahul gautam         | ballia     | UP    |  109681 |
| 42 | nishi siddiqi        | moradabad  | UP    |  925020 |
| 43 | disha chandok        | ludhiana   | PB    |  296055 |
| 44 | jimmy gill           | chandigarh | PB    |  705216 |
| 45 | deepak sharma        | gurdaspur  | PB    |  637914 |
| 46 | ankur bagga          | chandigarh | PB    |   73924 |
| 47 | asif sheikh          | kulgam     | JK    |  455876 |
| 48 | shahibe alam         | anantnag   | JK    |   57610 |
| 49 | guddu thomas         | imphal     | MN    |  920423 |
| 50 | pradeep gurung       | chandel    | MN    |  429282 |
| 51 | sujeet thapa         | bishnupur  | MN    |  385144 |
| 52 | shankey ale          | imphal     | MN    |  637873 |
| 53 | iti saxena           | nagpur     | MH    |   33934 |
| 54 | chitra chak          | latur      | MH    |  256051 |
| 55 | rinki pal            | jaipur     | RJ    |  178453 |
| 56 | poornima sahay       | pune       | MH    |  124112 |
| 57 | kishan bajpai        | banglore   | KA    |   85200 |
| 58 | sonia dwivedi        | bokaro     | JH    |   53664 |
| 59 | sanjay shukla        | ambala     | HR    |   12721 |
| 60 | sudhir chaudhary     | kutch      | GJ    |  902611 |
+----+----------------------+------------+-------+---------+
60 rows in set (0.01 sec)
```

select distinct state from students;

```
+-------+
| state |
+-------+
| UP    |
| UK    |
| KA    |
| WB    |
| JK    |
| MH    |
| GJ    |
| RJ    |
| HR    |
| JH    |
| PB    |
| MN    |
+-------+
```

select * from marks where score >= 22;

```
+----+--------------------------------+-------+-----+
| id | course_name                    | score | sid |
+----+--------------------------------+-------+-----+
|  2 | AI In Real World Using Python |    99 |  51 |
|  3 | GIMP Photo Editing             |    51 |  29 |
|  4 | SQLite Tutorial                |    57 |  13 |
|  5 | Swing GUI In Depth             |    31 |  34 |
|  6 | CSS3                           |    83 |  36 |
|  7 | Codeigniter                    |    22 |  39 |
|  8 | 2D Games Using PyGame          |    60 |  37 |
|  9 | Data Structure In Depth        |    31 |  32 |
```

```
| 10 | Amazon Cloud AWS              |    76 |  12 |
| 11 | Amazon Cloud AWS              |    86 |  27 |
| 13 | Swing GUI In Depth            |    52 |   1 |
| 14 | Dynamic Website Development   |    52 |   3 |
| 16 | Natural Language Processing   |    55 |   6 |
| 17 | HTML                          |    68 |   6 |
| 18 | Computer Vision Using Python  |    72 |  47 |
| 19 | Computer Vision Using Python  |    58 |  24 |
| 20 | Data Structure In Depth       |    70 |  14 |
| 21 | Kali Linux                    |    76 |  18 |
| 22 | Java Complete Tutorial        |    67 |  32 |
| 24 | The C Ninja                   |    39 |  10 |
| 25 | SQLite Tutorial               |    69 |  60 |
+----+-------------------------------+-------+-----+
21 rows in set (0.00 sec)
```

select * from students where state = "UP"; -- Note == not work in the sql

```
+----+-----------------+-----------+-------+---------+
| id | name            | city      | state | pincode |
+----+-----------------+-----------+-------+---------+
|  1 | sandeep ganguly | kanpur    | UP    |  601988 |
|  3 | divyanshu shukla | kanpur    | UP    |  940965 |
|  4 | ankita          | kanpur    | UP    |   56669 |
|  5 | brijesh gupta   | gorakhpur | UP    |  460450 |
|  6 | siddhartha singh | kanpur    | UP    |  132244 |
|  7 | parvez hasan    | faizabad  | UP    |  279869 |
| 15 | varsha singh    | mathura   | UP    |  770636 |
| 41 | rahul gautam    | ballia    | UP    |  109681 |
| 42 | nishi siddiqi   | moradabad | UP    |  925020 |
+----+-----------------+-----------+-------+---------+
9 rows in set (0.00 sec)
```

select * from students where id >= 10 && id <= 15;

```
+----+--------------+--------+-------+---------+
| id | name         | city   | state | pincode |
+----+--------------+--------+-------+---------+
| 10 | ayushi sharma | jammu  | JK    |  859431 |
| 11 | shameem beg  | mumbai | MH    |  776793 |
| 12 | Arun Bhatia  | pune   | MH    |  305673 |
| 13 | shiv patel   | surat  | GJ    |  197988 |
| 14 | aman ali     | ajmer  | RJ    |   72920 |
| 15 | varsha singh | mathura | UP    |  770636 |
+----+--------------+--------+-------+---------+
6 rows in set (0.00 sec)
```

select * from students where state != "UP";

```
+----+---------------------+------------+-------+---------+
| id | name                | city       | state | pincode |
+----+---------------------+------------+-------+---------+
|  2 | piyush chandel      | nainital   | UK    |  549386 |
|  8 | pawan kumar         | banglore   | KA    |    2612 |
|  9 | umesh verma         | kolkata    | WB    |  173453 |
| 10 | ayushi sharma       | jammu      | JK    |  859431 |
| 11 | shameem beg         | mumbai     | MH    |  776793 |
| 12 | Arun Bhatia         | pune       | MH    |  305673 |
| 13 | shiv patel          | surat      | GJ    |  197988 |
| 14 | aman ali            | ajmer      | RJ    |   72920 |
| 16 | deepak yadav        | gurugram   | HR    |  634419 |
| 17 | manjul saini        | dhanbad    | JH    |  860186 |
| 18 | Ankur sharma        | ranchi     | JH    |  397676 |
| 19 | saurabh gupta       | ahemdabad  | GJ    |  407819 |
```

```
| 20 | soumya pandey        | srinagar   | JK    |  846069 |
| 21 | digvijay patel       | jamnagar   | GJ    |    6887 |
| 22 | shivani singh        | faridabad  | HR    |  496229 |
| 23 | sarvik roy           | purulia    | WB    |  460485 |
| 24 | mamta banerjee       | kolkata    | WB    |  813736 |
| 25 | dolly ganguly        | howrah     | WB    |  687224 |
| 26 | shubhojeet mukherjee | birbhum    | WB    |  994915 |
| 27 | shubham das          | jhargram   | WB    |  912900 |
| 28 | tapas paul           | bankura    | WB    |  579758 |
| 29 | sbhubendu sarkar     | kolkatta   | WB    |  160089 |
| 30 | kaveri bose          | howrah     | WB    |   61170 |
| 31 | mitali chatterjee    | kolkata    | WB    |  825583 |
| 32 | rupoma biswas        | howrah     | WB    |  944403 |
| 33 | sujeet ghara         | howrah     | WB    |  245270 |
| 34 | shweta ghara         | kolkata    | WB    |  393138 |
| 35 | nita ganguly         | kolkata    | WB    |  229879 |
| 36 | sumita ganguly       | kolkata    | WB    |  969984 |
| 37 | sumit thakrey        | mumbai     | MH    |  160281 |
| 38 | nana patekar         | mumbai     | MH    |  891455 |
| 39 | nitin gadkari        | nagpur     | MH    |  976432 |
| 40 | dharmesh pradhan     | latur      | MH    |  207796 |
| 43 | disha chandok        | ludhiana   | PB    |  296055 |
| 44 | jimmy gill           | chandigarh | PB    |  705216 |
| 45 | deepak sharma        | gurdaspur  | PB    |  637914 |
| 46 | ankur bagga          | chandigarh | PB    |   73924 |
| 47 | asif sheikh          | kulgam     | JK    |  455876 |
| 48 | shahibe alam         | anantnag   | JK    |   57610 |
| 49 | guddu thomas         | imphal     | MN    |  920423 |
| 50 | pradeep gurung       | chandel    | MN    |  429282 |
| 51 | sujeet thapa         | bishnupur  | MN    |  385144 |
| 52 | shankey ale          | imphal     | MN    |  637873 |
| 53 | iti saxena           | nagpur     | MH    |   33934 |
| 54 | chitra chak          | latur      | MH    |  256051 |
| 55 | rinki pal            | jaipur     | RJ    |  178453 |
| 56 | poornima sahay       | pune       | MH    |  124112 |
| 57 | kishan bajpai        | banglore   | KA    |   85200 |
| 58 | sonia dwivedi        | bokaro     | JH    |   53664 |
| 59 | sanjay shukla        | ambala     | HR    |   12721 |
| 60 | sudhir chaudhary     | kutch      | GJ    |  902611 |
+----+----------------------+------------+-------+---------+
51 rows in set (0.00 sec)
```

select * from students where state = "MN" && city = 'imphal';

```
+----+--------------+--------+-------+---------+
| id | name         | city   | state | pincode |
+----+--------------+--------+-------+---------+
| 49 | guddu thomas | imphal | MN    |  920423 |
| 52 | shankey ale  | imphal | MN    |  637873 |
+----+--------------+--------+-------+---------+
2 rows in set (0.00 sec)
```

select * from students where state = "MN" || city = 'imphal';

```
+----+----------------+-----------+-------+---------+
| id | name           | city      | state | pincode |
+----+----------------+-----------+-------+---------+
| 49 | guddu thomas   | imphal    | MN    |  920423 |
| 50 | pradeep gurung | chandel   | MN    |  429282 |
| 51 | sujeet thapa   | bishnupur | MN    |  385144 |
| 52 | shankey ale    | imphal    | MN    |  637873 |
+----+----------------+-----------+-------+---------+
4 rows in set (0.00 sec)
```

select * from students where state = "MN" and city = 'imphal';

id	name	city	state	pincode
49	guddu thomas	imphal	MN	920423
52	shankey ale	imphal	MN	637873

2 rows in set (0.00 sec)

select * from students where state = "MN" OR city = 'imphal';

id	name	city	state	pincode
49	guddu thomas	imphal	MN	920423
50	pradeep gurung	chandel	MN	429282
51	sujeet thapa	bishnupur	MN	385144
52	shankey ale	imphal	MN	637873

4 rows in set (0.00 sec)

SELECT * FROM students WHERE NOT city = 'kanpur';

id	name	city	state	pincode
2	piyush chandel	nainital	UK	549386
5	brijesh gupta	gorakhpur	UP	460450
7	parvez hasan	faizabad	UP	279869
8	pawan kumar	banglore	KA	2612
9	umesh verma	kolkata	WB	173453
10	ayushi sharma	jammu	JK	859431
11	shameem beg	mumbai	MH	776793
12	Arun Bhatia	pune	MH	305673
13	shiv patel	surat	GJ	197988
14	aman ali	ajmer	RJ	72920
15	varsha singh	mathura	UP	770636
16	deepak yadav	gurugram	HR	634419
17	manjul saini	dhanbad	JH	860186
18	Ankur sharma	ranchi	JH	397676
19	saurabh gupta	ahemdabad	GJ	407819
20	soumya pandey	srinagar	JK	846069
21	digvijay patel	jamnagar	GJ	6887
22	shivani singh	faridabad	HR	496229
23	sarvik roy	purulia	WB	460485
24	mamta banerjee	kolkata	WB	813736
25	dolly ganguly	howrah	WB	687224
26	shubhojeet mukherjee	birbhum	WB	994915
27	shubham das	jhargram	WB	912900
28	tapas paul	bankura	WB	579758
29	sbhubendu sarkar	kolkatta	WB	160089
30	kaveri bose	howrah	WB	61170
31	mitali chatterjee	kolkata	WB	825583
32	rupoma biswas	howrah	WB	944403
33	sujeet ghara	howrah	WB	245270
34	shweta ghara	kolkata	WB	393138
35	nita ganguly	kolkata	WB	229879
36	sumita ganguly	kolkata	WB	969984
37	sumit thakrey	mumbai	MH	160281
38	nana patekar	mumbai	MH	891455
39	nitin gadkari	nagpur	MH	976432
40	dharmesh pradhan	latur	MH	207796

```
| 41 | rahul gautam         | ballia     | UP    |  109681 |
| 42 | nishi siddiqi        | moradabad  | UP    |  925020 |
| 43 | disha chandok        | ludhiana   | PB    |  296055 |
| 44 | jimmy gill           | chandigarh | PB    |  705216 |
| 45 | deepak sharma        | gurdaspur  | PB    |  637914 |
| 46 | ankur bagga          | chandigarh | PB    |   73924 |
| 47 | asif sheikh          | kulgam     | JK    |  455876 |
| 48 | shahibe alam         | anantnag   | JK    |   57610 |
| 49 | guddu thomas         | imphal     | MN    |  920423 |
| 50 | pradeep gurung       | chandel    | MN    |  429282 |
| 51 | sujeet thapa         | bishnupur  | MN    |  385144 |
| 52 | shankey ale          | imphal     | MN    |  637873 |
| 53 | iti saxena           | nagpur     | MH    |   33934 |
| 54 | chitra chak          | latur      | MH    |  256051 |
| 55 | rinki pal            | jaipur     | RJ    |  178453 |
| 56 | poornima sahay       | pune       | MH    |  124112 |
| 57 | kishan bajpai        | banglore   | KA    |   85200 |
| 58 | sonia dwivedi        | bokaro     | JH    |   53664 |
| 59 | sanjay shukla        | ambala     | HR    |   12721 |
| 60 | sudhir chaudhary     | kutch      | GJ    |  902611 |
+----+----------------------+------------+-------+---------+
56 rows in set (0.00 sec)
```

SELECT * FROM courses;

```
+----+-------------------------------+-----------------+-------+
| id | course_name                   | instructor_name | fees  |
+----+-------------------------------+-----------------+-------+
|  1 | MySQL Database                | Sandeep Ganguly |  4500 |
|  2 | PHP Development               | Dolly Singh     |  1500 |
|  3 | Java Complete Tutorial        | Ramesh Yadav    |  7500 |
|  4 | Swing GUI In Depth            | Guddu Sharma    | 15000 |
|  5 | Computer Vision Using Python  | Narendra Murthy | 25000 |
|  6 | AI In Real World Using Python | Satya Kundu     | 45000 |
|  7 | 2D Games Using PyGame         | Sandeep Ganguly | 18000 |
|  8 | GIMP Photo Editing            | Rachna Mishra   |  5000 |
|  9 | HTML                          | Chatur Singh    |  1000 |
| 10 | CSS3                          | pinky singh     |  1500 |
| 11 | Amazon Cloud AWS              | Ruchi Singhania | 75000 |
| 12 | Hadoop Big Data               | Ankita Ganguly  | 95000 |
| 13 | Natural Language Processing   | Sandeep Ganguly | 45999 |
| 14 | The C Ninja                   | Pradeep Gurung  |  3599 |
| 15 | Java Database Connectivity    | Ratan Tata      |  6599 |
| 16 | Dynamic Website Development   | Girish Patel    |  8599 |
| 17 | Android App Development       | Rishi Khanna    | 17999 |
| 18 | IOS Developer                 | Umesh Verma     | 25000 |
| 19 | Algorithms In Depth           | Arjun Thapa     |  9999 |
| 20 | Data Structure In Depth       | Ashok Kalia     | 15000 |
| 21 | JQuery Ninja                  | James Guido     |  6500 |
| 22 | Twitter Bootstrap             | Mitali Ghosh    | 14999 |
| 23 | Codeigniter                   | Pawan Kumar     |  7599 |
| 24 | Struts Framework              | Umesh Verma     |  7500 |
| 25 | CakePHP                       | Parvez Khan     | 60000 |
| 26 | Machine Learning              | Faisal Qureshi  | 45000 |
| 27 | Computer Networks             | Saleem Khan     | 12599 |
| 28 | C++ STL Library Tutorial      | Kareem Sheikh   | 25000 |
| 29 | Kali Linux                    | Jitan Majhi     |  4500 |
| 30 | SQLite Tutorial               | Nitish Kumar    |  6500 |
+----+-------------------------------+-----------------+-------+
30 rows in set (0.01 sec)
```

-- use BETWEEN when you want to retrive data in range
-- you can also write query SELECT * FROM courses WHERE fees > 5000 AND fees < 15000;
SELECT * FROM courses where fees BETWEEN 5000 AND 15000;

id	course_name	instructor_name	fees
3	Java Complete Tutorial	Ramesh Yadav	7500
4	Swing GUI In Depth	Guddu Sharma	15000
8	GIMP Photo Editing	Rachna Mishra	5000
15	Java Database Connectivity	Ratan Tata	6599
16	Dynamic Website Development	Girish Patel	8599
19	Algorithms In Depth	Arjun Thapa	9999
20	Data Structure In Depth	Ashok Kalia	15000
21	JQuery Ninja	James Guido	6500
22	Twitter Bootstrap	Mitali Ghosh	14999
23	Codeigniter	Pawan Kumar	7599
24	Struts Framework	Umesh Verma	7500
27	Computer Networks	Saleem Khan	12599
30	SQLite Tutorial	Nitish Kumar	6500

13 rows in set (0.00 sec)

-- IN use when you want to retrive data on the basis of present
SELECT * FROM students WHERE state IN('UP','WB'); -- you can also change or add or remove more parameter

id	name	city	state	pincode
1	sandeep ganguly	kanpur	UP	601988
3	divyanshu shukla	kanpur	UP	940965
4	ankita	kanpur	UP	56669
5	brijesh gupta	gorakhpur	UP	460450
6	siddhartha singh	kanpur	UP	132244
7	parvez hasan	faizabad	UP	279869
9	umesh verma	kolkata	WB	173453
15	varsha singh	mathura	UP	770636
23	sarvik roy	purulia	WB	460485
24	mamta banerjee	kolkata	WB	813736
25	dolly ganguly	howrah	WB	687224
26	shubhojeet mukherjee	birbhum	WB	994915
27	shubham das	jhargram	WB	912900
28	tapas paul	bankura	WB	579758
29	sbhubendu sarkar	kolkatta	WB	160089
30	kaveri bose	howrah	WB	61170
31	mitali chatterjee	kolkata	WB	825583
32	rupoma biswas	howrah	WB	944403
33	sujeet ghara	howrah	WB	245270
34	shweta ghara	kolkata	WB	393138
35	nita ganguly	kolkata	WB	229879
36	sumita ganguly	kolkata	WB	969984
41	rahul gautam	ballia	UP	109681
42	nishi siddiqi	moradabad	UP	925020

24 rows in set (0.00 sec)

-- same as IN but it shows not present data
SELECT * FROM students WHERE state NOT IN('UP','WB'); -- you can also change or add or remove more parameter

```
+----+------------------+------------+-------+---------+
| id | name             | city       | state | pincode |
+----+------------------+------------+-------+---------+
|  2 | piyush chandel   | nainital   | UK    |  549386 |
|  8 | pawan kumar      | banglore   | KA    |    2612 |
| 10 | ayushi sharma    | jammu      | JK    |  859431 |
| 11 | shameem beg      | mumbai     | MH    |  776793 |
| 12 | Arun Bhatia      | pune       | MH    |  305673 |
| 13 | shiv patel       | surat      | GJ    |  197988 |
| 14 | aman ali         | ajmer      | RJ    |   72920 |
| 16 | deepak yadav     | gurugram   | HR    |  634419 |
| 17 | manjul saini     | dhanbad    | JH    |  860186 |
| 18 | Ankur sharma     | ranchi     | JH    |  397676 |
| 19 | saurabh gupta    | ahemdabad  | GJ    |  407819 |
| 20 | soumya pandey    | srinagar   | JK    |  846069 |
| 21 | digvijay patel   | jamnagar   | GJ    |    6887 |
| 22 | shivani singh    | faridabad  | HR    |  496229 |
| 37 | sumit thakrey    | mumbai     | MH    |  160281 |
| 38 | nana patekar     | mumbai     | MH    |  891455 |
| 39 | nitin gadkari    | nagpur     | MH    |  976432 |
| 40 | dharmesh pradhan | latur      | MH    |  207796 |
| 43 | disha chandok    | ludhiana   | PB    |  296055 |
| 44 | jimmy gill       | chandigarh | PB    |  705216 |
| 45 | deepak sharma    | gurdaspur  | PB    |  637914 |
| 46 | ankur bagga      | chandigarh | PB    |   73924 |
| 47 | asif sheikh      | kulgam     | JK    |  455876 |
| 48 | shahibe alam     | anantnag   | JK    |   57610 |
| 49 | guddu thomas     | imphal     | MN    |  920423 |
| 50 | pradeep gurung   | chandel    | MN    |  429282 |
| 51 | sujeet thapa     | bishnupur  | MN    |  385144 |
| 52 | shankey ale      | imphal     | MN    |  637873 |
| 53 | iti saxena       | nagpur     | MH    |   33934 |
| 54 | chitra chak      | latur      | MH    |  256051 |
| 55 | rinki pal        | jaipur     | RJ    |  178453 |
| 56 | poornima sahay   | pune       | MH    |  124112 |
| 57 | kishan bajpai    | banglore   | KA    |   85200 |
| 58 | sonia dwivedi    | bokaro     | JH    |   53664 |
| 59 | sanjay shukla    | ambala     | HR    |   12721 |
| 60 | sudhir chaudhary | kutch      | GJ    |  902611 |
+----+------------------+------------+-------+---------+
36 rows in set (0.00 sec)
```

-- order by basically used for sort data in asecding or desending order
-- default order is ascesding
SELECT * FROM courses ORDER BY fees ASC;

```
+----+-------------------------------+-----------------+-------+
| id | course_name                   | instructor_name | fees  |
+----+-------------------------------+-----------------+-------+
|  9 | HTML                          | Chatur Singh    |  1000 |
|  2 | PHP Development               | Dolly Singh     |  1500 |
| 10 | CSS3                          | pinky singh     |  1500 |
| 14 | The C Ninja                   | Pradeep Gurung  |  3599 |
|  1 | MySQL Database                | Sandeep Ganguly |  4500 |
| 29 | Kali Linux                    | Jitan Majhi     |  4500 |
|  8 | GIMP Photo Editing            | Rachna Mishra   |  5000 |
| 21 | JQuery Ninja                  | James Guido     |  6500 |
| 30 | SQLite Tutorial               | Nitish Kumar    |  6500 |
| 15 | Java Database Connectivity    | Ratan Tata      |  6599 |
|  3 | Java Complete Tutorial        | Ramesh Yadav    |  7500 |
| 24 | Struts Framework              | Umesh Verma     |  7500 |
| 23 | Codeigniter                   | Pawan Kumar     |  7599 |
```

```
| 16 | Dynamic Website Development   | Girish Patel    |  8599 |
| 19 | Algorithms In Depth           | Arjun Thapa     |  9999 |
| 27 | Computer Networks             | Saleem Khan     | 12599 |
| 22 | Twitter Bootstrap             | Mitali Ghosh    | 14999 |
|  4 | Swing GUI In Depth            | Guddu Sharma    | 15000 |
| 20 | Data Structure In Depth       | Ashok Kalia     | 15000 |
| 17 | Android App Development       | Rishi Khanna    | 17999 |
|  7 | 2D Games Using PyGame         | Sandeep Ganguly | 18000 |
| 28 | C++ STL Library Tutorial      | Kareem Sheikh   | 25000 |
| 18 | IOS Developer                 | Umesh Verma     | 25000 |
|  5 | Computer Vision Using Python  | Narendra Murthy | 25000 |
|  6 | AI In Real World Using Python | Satya Kundu     | 45000 |
| 26 | Machine Learning              | Faisal Qureshi  | 45000 |
| 13 | Natural Language Processing   | Sandeep Ganguly | 45999 |
| 25 | CakePHP                       | Parvez Khan     | 60000 |
| 11 | Amazon Cloud AWS              | Ruchi Singhania | 75000 |
| 12 | Hadoop Big Data               | Ankita Ganguly  | 95000 |
+----+-------------------------------+-----------------+-------+
30 rows in set (0.00 sec)
```

SELECT * FROM courses ORDER BY fees DESC;

```
+----+-------------------------------+-----------------+-------+
| id | course_name                   | instructor_name | fees  |
+----+-------------------------------+-----------------+-------+
| 12 | Hadoop Big Data               | Ankita Ganguly  | 95000 |
| 11 | Amazon Cloud AWS              | Ruchi Singhania | 75000 |
| 25 | CakePHP                       | Parvez Khan     | 60000 |
| 13 | Natural Language Processing   | Sandeep Ganguly | 45999 |
|  6 | AI In Real World Using Python | Satya Kundu     | 45000 |
| 26 | Machine Learning              | Faisal Qureshi  | 45000 |
|  5 | Computer Vision Using Python  | Narendra Murthy | 25000 |
| 18 | IOS Developer                 | Umesh Verma     | 25000 |
| 28 | C++ STL Library Tutorial      | Kareem Sheikh   | 25000 |
|  7 | 2D Games Using PyGame         | Sandeep Ganguly | 18000 |
| 17 | Android App Development       | Rishi Khanna    | 17999 |
|  4 | Swing GUI In Depth            | Guddu Sharma    | 15000 |
| 20 | Data Structure In Depth       | Ashok Kalia     | 15000 |
| 22 | Twitter Bootstrap             | Mitali Ghosh    | 14999 |
| 27 | Computer Networks             | Saleem Khan     | 12599 |
| 19 | Algorithms In Depth           | Arjun Thapa     |  9999 |
| 16 | Dynamic Website Development   | Girish Patel    |  8599 |
| 23 | Codeigniter                   | Pawan Kumar     |  7599 |
| 24 | Struts Framework              | Umesh Verma     |  7500 |
|  3 | Java Complete Tutorial        | Ramesh Yadav    |  7500 |
| 15 | Java Database Connectivity    | Ratan Tata      |  6599 |
| 21 | JQuery Ninja                  | James Guido     |  6500 |
| 30 | SQLite Tutorial               | Nitish Kumar    |  6500 |
|  8 | GIMP Photo Editing            | Rachna Mishra   |  5000 |
| 29 | Kali Linux                    | Jitan Majhi     |  4500 |
|  1 | MySQL Database                | Sandeep Ganguly |  4500 |
| 14 | The C Ninja                   | Pradeep Gurung  |  3599 |
| 10 | CSS3                          | pinky singh     |  1500 |
|  2 | PHP Development               | Dolly Singh     |  1500 |
|  9 | HTML                          | Chatur Singh    |  1000 |
+----+-------------------------------+-----------------+-------+
30 rows in set (0.00 sec)
```

DESC courses; --its descibe the table;

```
+-----------------+-------------+------+-----+---------+----------------+
| Field           | Type        | Null | Key | Default | Extra          |
+-----------------+-------------+------+-----+---------+----------------+
```

```
| id              | int(11)     | NO   | PRI | NULL    | auto_increment |
| course_name     | varchar(30) | NO   |     | NULL    |                |
| instructor_name | varchar(30) | NO   |     | NULL    |                |
| fees            | int(11)     | NO   |     | NULL    |                |
+-----------------+-------------+------+-----+---------+----------------+
4 rows in set (0.02 sec)
```

SELECT * FROM courses ORDER BY fees ASC LIMIT 5;

```
+----+-----------------+-----------------+------+
| id | course_name     | instructor_name | fees |
+----+-----------------+-----------------+------+
|  9 | HTML            | Chatur Singh    | 1000 |
|  2 | PHP Development | Dolly Singh     | 1500 |
| 10 | CSS3            | pinky singh     | 1500 |
| 14 | The C Ninja     | Pradeep Gurung  | 3599 |
|  1 | MySQL Database  | Sandeep Ganguly | 4500 |
+----+-----------------+-----------------+------+
5 rows in set (0.00 sec)
```

SELECT * FROM courses ORDER BY id DESC;

```
+----+-------------------------------+-----------------+-------+
| id | course_name                   | instructor_name | fees  |
+----+-------------------------------+-----------------+-------+
| 30 | SQLite Tutorial               | Nitish Kumar    |  6500 |
| 29 | Kali Linux                    | Jitan Majhi     |  4500 |
| 28 | C++ STL Library Tutorial      | Kareem Sheikh   | 25000 |
| 27 | Computer Networks             | Saleem Khan     | 12599 |
| 26 | Machine Learning              | Faisal Qureshi  | 45000 |
| 25 | CakePHP                       | Parvez Khan     | 60000 |
| 24 | Struts Framework              | Umesh Verma     |  7500 |
| 23 | Codeigniter                   | Pawan Kumar     |  7599 |
| 22 | Twitter Bootstrap             | Mitali Ghosh    | 14999 |
| 21 | JQuery Ninja                  | James Guido     |  6500 |
| 20 | Data Structure In Depth       | Ashok Kalia     | 15000 |
| 19 | Algorithms In Depth           | Arjun Thapa     |  9999 |
| 18 | IOS Developer                 | Umesh Verma     | 25000 |
| 17 | Android App Development       | Rishi Khanna    | 17999 |
| 16 | Dynamic Website Development   | Girish Patel    |  8599 |
| 15 | Java Database Connectivity    | Ratan Tata      |  6599 |
| 14 | The C Ninja                   | Pradeep Gurung  |  3599 |
| 13 | Natural Language Processing   | Sandeep Ganguly | 45999 |
| 12 | Hadoop Big Data               | Ankita Ganguly  | 95000 |
| 11 | Amazon Cloud AWS              | Ruchi Singhania | 75000 |
| 10 | CSS3                          | pinky singh     |  1500 |
|  9 | HTML                          | Chatur Singh    |  1000 |
|  8 | GIMP Photo Editing            | Rachna Mishra   |  5000 |
|  7 | 2D Games Using PyGame         | Sandeep Ganguly | 18000 |
|  6 | AI In Real World Using Python | Satya Kundu     | 45000 |
|  5 | Computer Vision Using Python  | Narendra Murthy | 25000 |
|  4 | Swing GUI In Depth            | Guddu Sharma    | 15000 |
|  3 | Java Complete Tutorial        | Ramesh Yadav    |  7500 |
|  2 | PHP Development               | Dolly Singh     |  1500 |
|  1 | MySQL Database                | Sandeep Ganguly |  4500 |
+----+-------------------------------+-----------------+-------+
30 rows in set (0.00 sec)
```

SELECT * FROM courses WHERE course_name LIKE "java%";

```
+----+---------------------------+-----------------+------+
| id | course_name               | instructor_name | fees |
+----+---------------------------+-----------------+------+
|  3 | Java Complete Tutorial    | Ramesh Yadav    | 7500 |
```

```
| 15 | Java Database Connectivity | Ratan Tata      | 6599 |
+----+----------------------------+-----------------+------+
2 rows in set (0.00 sec)
```

SELECT * FROM courses WHERE course_name LIKE "%depth";

```
+----+-------------------------+-----------------+-------+
| id | course_name             | instructor_name | fees  |
+----+-------------------------+-----------------+-------+
|  4 | Swing GUI In Depth      | Guddu Sharma    | 15000 |
| 19 | Algorithms In Depth     | Arjun Thapa     |  9999 |
| 20 | Data Structure In Depth | Ashok Kalia     | 15000 |
+----+-------------------------+-----------------+-------+
3 rows in set (0.00 sec)
```

SELECT * FROM courses WHERE course_name LIKE "% IN %";

```
+----+--------------------------------+-----------------+-------+
| id | course_name                    | instructor_name | fees  |
+----+--------------------------------+-----------------+-------+
|  4 | Swing GUI In Depth             | Guddu Sharma    | 15000 |
|  6 | AI In Real World Using Python | Satya Kundu     | 45000 |
| 19 | Algorithms In Depth            | Arjun Thapa     |  9999 |
| 20 | Data Structure In Depth        | Ashok Kalia     | 15000 |
+----+--------------------------------+-----------------+-------+
4 rows in set (0.00 sec)
```

SELECT * FROM courses WHERE course_name LIKE "c_m%";

```
+----+--------------------------------+-----------------+-------+
| id | course_name                    | instructor_name | fees  |
+----+--------------------------------+-----------------+-------+
|  5 | Computer Vision Using Python | Narendra Murthy | 25000 |
| 27 | Computer Networks              | Saleem Khan     | 12599 |
+----+--------------------------------+-----------------+-------+
2 rows in set (0.00 sec)
```

SELECT * FROM courses WHERE course_name LIKE "J_v_ %";

```
+----+----------------------------+-----------------+------+
| id | course_name                | instructor_name | fees |
+----+----------------------------+-----------------+------+
|  3 | Java Complete Tutorial     | Ramesh Yadav    | 7500 |
| 15 | Java Database Connectivity | Ratan Tata      | 6599 |
+----+----------------------------+-----------------+------+
2 rows in set (0.00 sec)
```

SELECT * FROM courses WHERE course_name LIKE "J_v %";
Empty set (0.00 sec)

SELECT * FROM marks WHERE course_name LIKE "C_____e_ %"; -- we can also use underscore more than once

```
+----+--------------------------------+-------+-----+
| id | course_name                    | score | sid |
+----+--------------------------------+-------+-----+
| 12 | Computer Networks              |     2 |  34 |
| 18 | Computer Vision Using Python |    72 |  47 |
| 19 | Computer Vision Using Python |    58 |  24 |
+----+--------------------------------+-------+-----+
3 rows in set (0.00 sec)
```

/*----------Aggregate Function ----------*/

SELECT * FROM marks;

id	course_name	score	sid
1	AI In Real World Using Python	14	38
2	AI In Real World Using Python	99	51
3	GIMP Photo Editing	51	29
4	SQLite Tutorial	57	13
5	Swing GUI In Depth	31	34
6	CSS3	83	36
7	Codeigniter	22	39
8	2D Games Using PyGame	60	37
9	Data Structure In Depth	31	32
10	Amazon Cloud AWS	76	12
11	Amazon Cloud AWS	86	27
12	Computer Networks	2	34
13	Swing GUI In Depth	52	1
14	Dynamic Website Development	52	3
15	CakePHP	2	27
16	Natural Language Processing	55	6
17	HTML	68	6
18	Computer Vision Using Python	72	47
19	Computer Vision Using Python	58	24
20	Data Structure In Depth	70	14
21	Kali Linux	76	18
22	Java Complete Tutorial	67	32
23	Java Database Connectivity	8	25
24	The C Ninja	39	10
25	SQLite Tutorial	69	60

25 rows in set (0.01 sec)

SELECT MIN(score) FROM marks;

MIN(score)
2

1 row in set (0.00 sec)

SELECT MAX(score) FROM marks;

MAX(score)
99

1 row in set (0.00 sec)

SELECT COUNT(score) FROM marks;

COUNT(score)
25

1 row in set (0.00 sec)

SELECT SUM(score) FROM marks;

SUM(score)

```
|       1300 |
+------------+
1 row in set (0.00 sec)
```

SELECT AVG(score) FROM marks;

```
+------------+
| AVG(score) |
+------------+
|    52.0000 |
+------------+
1 row in set (0.00 sec)
```

/*--- subquery with aggregate function ---*/

SELECT * FROM marks WHERE score = (SELECT MIN(score) FROM marks);

```
+----+------------------+-------+-----+
| id | course_name      | score | sid |
+----+------------------+-------+-----+
| 12 | Computer Networks |     2 |  34 |
| 15 | CakePHP           |     2 |  27 |
+----+------------------+-------+-----+
```

SELECT * FROM marks WHERE score = (SELECT SUM(score) FROM marks);
Empty set (0.00 sec)

SELECT * FROM marks WHERE score = (SELECT MAX(score) FROM marks);

```
+----+-------------------------------+-------+-----+
| id | course_name                   | score | sid |
+----+-------------------------------+-------+-----+
|  2 | AI In Real World Using Python |    99 |  51 |
+----+-------------------------------+-------+-----+
1 row in set (0.00 sec)
```

SELECT * FROM marks WHERE score = (SELECT MAX(score) FROM marks);

```
+----+-------------------------------+-------+-----+
| id | course_name                   | score | sid |
+----+-------------------------------+-------+-----+
|  2 | AI In Real World Using Python |    99 |  51 |
+----+-------------------------------+-------+-----+
1 row in set (0.00 sec)
```

SELECT name,pincode, CONCAT(city," ",state) FROM students LIMIT 10;

```
+------------------+---------+------------------------+
| name             | pincode | CONCAT(city," ",state) |
+------------------+---------+------------------------+
| sandeep ganguly  |  601988 | kanpur UP              |
| piyush chandel   |  549386 | nainital UK            |
| divyanshu shukla |  940965 | kanpur UP              |
| ankita           |   56669 | kanpur UP              |
| brijesh gupta    |  460450 | gorakhpur UP           |
| siddhartha singh |  132244 | kanpur UP              |
| parvez hasan     |  279869 | faizabad UP            |
| pawan kumar      |    2612 | banglore KA            |
| umesh verma      |  173453 | kolkata WB             |
| ayushi sharma    |  859431 | jammu JK               |
+------------------+---------+------------------------+
10 rows in set (0.00 sec)
```

SELECT name,pincode, CONCAT(city,",",state) FROM students LIMIT 10;

```
+------------------+---------+-----------------------+
| name             | pincode | CONCAT(city,",",state) |
```

```
+------------------+---------+------------------------+
| sandeep ganguly  |  601988 | kanpur,UP              |
| piyush chandel   |  549386 | nainital,UK            |
| divyanshu shukla |  940965 | kanpur,UP              |
| ankita           |   56669 | kanpur,UP              |
| brijesh gupta    |  460450 | gorakhpur,UP           |
| siddhartha singh |  132244 | kanpur,UP              |
| parvez hasan     |  279869 | faizabad,UP            |
| pawan kumar      |    2612 | banglore,KA            |
| umesh verma      |  173453 | kolkata,WB             |
| ayushi sharma    |  859431 | jammu,JK               |
+------------------+---------+------------------------+
10 rows in set (0.00 sec)
```

-- CREATING ALIASES AS KEYWORD

select fees AS charges from courses;

```
+---------+
| charges |
+---------+
|    4500 |
|    1500 |
|    7500 |
|   15000 |
|   25000 |
|   45000 |
|   18000 |
|    5000 |
|    1000 |
|    1500 |
|   75000 |
|   95000 |
|   45999 |
|    3599 |
|    6599 |
|    8599 |
|   17999 |
|   25000 |
|    9999 |
|   15000 |
|    6500 |
|   14999 |
|    7599 |
|    7500 |
|   60000 |
|   45000 |
|   12599 |
|   25000 |
|    4500 |
|    6500 |
+---------+
30 rows in set (0.00 sec)
```

select count(fees) AS CountOfFees from courses;

```
+-------------+
| CountOfFees |
+-------------+
|          30 |
+-------------+
1 row in set (0.00 sec)
```

-- fully qualified queries is basically usefull in JOIN OPERATIONS.
SELECT courses.course_name, courses.fees From courses; //also known as fully qualified queries.

```
+-------------------------------+-------+
| course_name                   | fees  |
+-------------------------------+-------+
| MySQL Database                |  4500 |
| PHP Development               |  1500 |
| Java Complete Tutorial        |  7500 |
| Swing GUI In Depth            | 15000 |
| Computer Vision Using Python  | 25000 |
| AI In Real World Using Python | 45000 |
| 2D Games Using PyGame         | 18000 |
| GIMP Photo Editing            |  5000 |
| HTML                          |  1000 |
| CSS3                          |  1500 |
| Amazon Cloud AWS              | 75000 |
| Hadoop Big Data               | 95000 |
| Natural Language Processing   | 45999 |
| The C Ninja                   |  3599 |
| Java Database Connectivity    |  6599 |
| Dynamic Website Development   |  8599 |
| Android App Development       | 17999 |
| IOS Developer                 | 25000 |
| Algorithms In Depth           |  9999 |
| Data Structure In Depth       | 15000 |
| JQuery Ninja                  |  6500 |
| Twitter Bootstrap             | 14999 |
| Codeigniter                   |  7599 |
| Struts Framework              |  7500 |
| CakePHP                       | 60000 |
| Machine Learning              | 45000 |
| Computer Networks             | 12599 |
| C++ STL Library Tutorial      | 25000 |
| Kali Linux                    |  4500 |
| SQLite Tutorial               |  6500 |
+-------------------------------+-------+
30 rows in set (0.00 sec)
```

SELECT c.course_name, c.fees FROM courses AS c;

```
+-------------------------------+-------+
| course_name                   | fees  |
+-------------------------------+-------+
| MySQL Database                |  4500 |
| PHP Development               |  1500 |
| Java Complete Tutorial        |  7500 |
| Swing GUI In Depth            | 15000 |
| Computer Vision Using Python  | 25000 |
| AI In Real World Using Python | 45000 |
| 2D Games Using PyGame         | 18000 |
| GIMP Photo Editing            |  5000 |
| HTML                          |  1000 |
| CSS3                          |  1500 |
| Amazon Cloud AWS              | 75000 |
| Hadoop Big Data               | 95000 |
| Natural Language Processing   | 45999 |
| The C Ninja                   |  3599 |
| Java Database Connectivity    |  6599 |
| Dynamic Website Development   |  8599 |
| Android App Development       | 17999 |
| IOS Developer                 | 25000 |
```

```
| Algorithms In Depth           |  9999 |
| Data Structure In Depth       | 15000 |
| JQuery Ninja                  |  6500 |
| Twitter Bootstrap             | 14999 |
| Codeigniter                   |  7599 |
| Struts Framework              |  7500 |
| CakePHP                       | 60000 |
| Machine Learning              | 45000 |
| Computer Networks             | 12599 |
| C++ STL Library Tutorial      | 25000 |
| Kali Linux                    |  4500 |
| SQLite Tutorial               |  6500 |
+-------------------------------+-------+
30 rows in set (0.00 sec)
```

SELECT m.id AS MYID , m.sid AS MYSID FROM marks m;

```
+------+-------+
| MYID | MYSID |
+------+-------+
|    1 |    38 |
|    2 |    51 |
|    3 |    29 |
|    4 |    13 |
|    5 |    34 |
|    6 |    36 |
|    7 |    39 |
|    8 |    37 |
|    9 |    32 |
|   10 |    12 |
|   11 |    27 |
|   12 |    34 |
|   13 |     1 |
|   14 |     3 |
|   15 |    27 |
|   16 |     6 |
|   17 |     6 |
|   18 |    47 |
|   19 |    24 |
|   20 |    14 |
|   21 |    18 |
|   22 |    32 |
|   23 |    25 |
|   24 |    10 |
|   25 |    60 |
+------+-------+
25 rows in set (0.00 sec)
```

SELECT m.id AS MYID , m.sid AS MYSID FROM marks AS m;

```
+------+-------+
| MYID | MYSID |
+------+-------+
|    1 |    38 |
|    2 |    51 |
|    3 |    29 |
|    4 |    13 |
|    5 |    34 |
|    6 |    36 |
|    7 |    39 |
|    8 |    37 |
|    9 |    32 |
|   10 |    12 |
```

```
|   11 |    27 |
|   12 |    34 |
|   13 |     1 |
|   14 |     3 |
|   15 |    27 |
|   16 |     6 |
|   17 |     6 |
|   18 |    47 |
|   19 |    24 |
|   20 |    14 |
|   21 |    18 |
|   22 |    32 |
|   23 |    25 |
|   24 |    10 |
|   25 |    60 |
+------+-------+
```

/*---------- Look for update operations ----------*/

update marks SET score = 55 Where id = 9;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

select * from marks;

```
+----+-------------------------------+-------+-----+
| id | course_name                   | score | sid |
+----+-------------------------------+-------+-----+
|  1 | AI In Real World Using Python |    14 |  38 |
|  2 | AI In Real World Using Python |    99 |  51 |
|  3 | GIMP Photo Editing            |    51 |  29 |
|  4 | SQLite Tutorial               |    57 |  13 |
|  5 | Swing GUI In Depth            |    31 |  34 |
|  6 | CSS3                          |    83 |  36 |
|  7 | Codeigniter                   |    22 |  39 |
|  8 | 2D Games Using PyGame         |    60 |  37 |
|  9 | Data Structure In Depth       |    55 |  32 |
| 10 | Amazon Cloud AWS              |    76 |  12 |
| 11 | Amazon Cloud AWS              |    86 |  27 |
| 12 | Computer Networks             |     2 |  34 |
| 13 | Swing GUI In Depth            |    52 |   1 |
| 14 | Dynamic Website Development   |    52 |   3 |
| 15 | CakePHP                       |     2 |  27 |
| 16 | Natural Language Processing   |    55 |   6 |
| 17 | HTML                          |    68 |   6 |
| 18 | Computer Vision Using Python  |    72 |  47 |
| 19 | Computer Vision Using Python  |    58 |  24 |
| 20 | Data Structure In Depth       |    70 |  14 |
| 21 | Kali Linux                    |    76 |  18 |
| 22 | Java Complete Tutorial        |    67 |  32 |
| 23 | Java Database Connectivity    |     8 |  25 |
| 24 | The C Ninja                   |    39 |  10 |
| 25 | SQLite Tutorial               |    69 |  60 |
+----+-------------------------------+-------+-----+
25 rows in set (0.00 sec)
```

update marks SET score = 600 , course_name = "Advance DataStructure" Where id = 9;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

select * from marks;

```
+----+--------------------------------+-------+-----+
| id | course_name                    | score | sid |
+----+--------------------------------+-------+-----+
|  1 | AI In Real World Using Python |    14 |  38 |
|  2 | AI In Real World Using Python |    99 |  51 |
|  3 | GIMP Photo Editing             |    51 |  29 |
|  4 | SQLite Tutorial                |    57 |  13 |
|  5 | Swing GUI In Depth             |    31 |  34 |
|  6 | CSS3                           |    83 |  36 |
|  7 | Codeigniter                    |    22 |  39 |
|  8 | 2D Games Using PyGame          |    60 |  37 |
|  9 | Advance DataStructure          |   600 |  32 |
| 10 | Amazon Cloud AWS               |    76 |  12 |
| 11 | Amazon Cloud AWS               |    86 |  27 |
| 12 | Computer Networks              |     2 |  34 |
| 13 | Swing GUI In Depth             |    52 |   1 |
| 14 | Dynamic Website Development    |    52 |   3 |
| 15 | CakePHP                        |     2 |  27 |
| 16 | Natural Language Processing    |    55 |   6 |
| 17 | HTML                           |    68 |   6 |
| 18 | Computer Vision Using Python   |    72 |  47 |
| 19 | Computer Vision Using Python   |    58 |  24 |
| 20 | Data Structure In Depth        |    70 |  14 |
| 21 | Kali Linux                     |    76 |  18 |
| 22 | Java Complete Tutorial         |    67 |  32 |
| 23 | Java Database Connectivity     |     8 |  25 |
| 24 | The C Ninja                    |    39 |  10 |
| 25 | SQLite Tutorial                |    69 |  60 |
+----+--------------------------------+-------+-----+
25 rows in set (0.00 sec)
```

-- for update multiple column value
update marks AS c SET c.score = 55 , c.course_name = "Advance Data Structure" Where id = 9;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

select * from marks;

```
+----+--------------------------------+-------+-----+
| id | course_name                    | score | sid |
+----+--------------------------------+-------+-----+
|  1 | AI In Real World Using Python |    14 |  38 |
|  2 | AI In Real World Using Python |    99 |  51 |
|  3 | GIMP Photo Editing             |    51 |  29 |
|  4 | SQLite Tutorial                |    57 |  13 |
|  5 | Swing GUI In Depth             |    31 |  34 |
|  6 | CSS3                           |    83 |  36 |
|  7 | Codeigniter                    |    22 |  39 |
|  8 | 2D Games Using PyGame          |    60 |  37 |
|  9 | Advance Data Structure         |    55 |  32 |
| 10 | Amazon Cloud AWS               |    76 |  12 |
| 11 | Amazon Cloud AWS               |    86 |  27 |
| 12 | Computer Networks              |     2 |  34 |
| 13 | Swing GUI In Depth             |    52 |   1 |
| 14 | Dynamic Website Development    |    52 |   3 |
| 15 | CakePHP                        |     2 |  27 |
| 16 | Natural Language Processing    |    55 |   6 |
| 17 | HTML                           |    68 |   6 |
| 18 | Computer Vision Using Python   |    72 |  47 |
| 19 | Computer Vision Using Python   |    58 |  24 |
```

```
| 20 | Data Structure In Depth       |    70 |  14 |
| 21 | Kali Linux                    |    76 |  18 |
| 22 | Java Complete Tutorial        |    67 |  32 |
| 23 | Java Database Connectivity    |     8 |  25 |
| 24 | The C Ninja                   |    39 |  10 |
| 25 | SQLite Tutorial               |    69 |  60 |
+----+-------------------------------+-------+-----+
25 rows in set (0.00 sec)
```

UPDATE marks SET course_name = "java from scarch" WHERE sid = 38;

```
+----+-------------------------------+-------+-----+
| id | course_name                   | score | sid |
+----+-------------------------------+-------+-----+
|  1 | java from scarch              |    14 |  38 |
|  2 | AI In Real World Using Python |    99 |  51 |
|  3 | GIMP Photo Editing            |    51 |  29 |
|  4 | SQLite Tutorial               |    57 |  13 |
|  5 | Swing GUI In Depth            |    31 |  34 |
|  6 | CSS3                          |    83 |  36 |
|  7 | Codeigniter                   |    22 |  39 |
|  8 | 2D Games Using PyGame         |    60 |  37 |
|  9 | Data Structure In Depth       |    31 |  32 |
| 10 | Amazon Cloud AWS              |    76 |  12 |
| 11 | Amazon Cloud AWS              |    86 |  27 |
| 12 | Computer Networks             |     2 |  34 |
| 13 | Swing GUI In Depth            |    52 |   1 |
| 14 | Dynamic Website Development   |    52 |   3 |
| 15 | CakePHP                       |     2 |  27 |
| 16 | Natural Language Processing   |    55 |   6 |
| 17 | HTML                          |    68 |   6 |
| 18 | Computer Vision Using Python  |    72 |  47 |
| 19 | Computer Vision Using Python  |    58 |  24 |
| 20 | Data Structure In Depth       |    70 |  14 |
| 21 | Kali Linux                    |    76 |  18 |
| 22 | Java Complete Tutorial        |    67 |  32 |
| 23 | Java Database Connectivity    |     8 |  25 |
| 24 | The C Ninja                   |    39 |  10 |
| 25 | SQLite Tutorial               |    69 |  60 |
+----+-------------------------------+-------+-----+
```

/*---------- Look for DELETE operations ----------*/

SELECT * FROM courses;

```
+----+-------------------------------+-----------------+-------+
| id | course_name                   | instructor_name | fees  |
+----+-------------------------------+-----------------+-------+
|  1 | MySQL Database                | Sandeep Ganguly |  4500 |
|  2 | PHP Development               | Dolly Singh     |  1500 |
|  3 | Java Complete Tutorial        | Ramesh Yadav    |  7500 |
|  4 | Swing GUI In Depth            | Guddu Sharma    | 15000 |
|  5 | Computer Vision Using Python  | Narendra Murthy | 25000 |
|  6 | AI In Real World Using Python | Satya Kundu     | 45000 |
|  7 | 2D Games Using PyGame         | Sandeep Ganguly | 18000 |
|  8 | GIMP Photo Editing            | Rachna Mishra   |  5000 |
|  9 | HTML                          | Chatur Singh    |  1000 |
| 10 | CSS3                          | pinky singh     |  1500 |
| 11 | Amazon Cloud AWS              | Ruchi Singhania | 75000 |
| 12 | Hadoop Big Data               | Ankita Ganguly  | 95000 |
| 13 | Natural Language Processing   | Sandeep Ganguly | 45999 |
| 14 | The C Ninja                   | Pradeep Gurung  |  3599 |
| 15 | Java Database Connectivity    | Ratan Tata      |  6599 |
```

```
| 16 | Dynamic Website Development   | Girish Patel    |  8599 |
| 17 | Android App Development       | Rishi Khanna    | 17999 |
| 18 | IOS Developer                 | Umesh Verma     | 25000 |
| 19 | Algorithms In Depth           | Arjun Thapa     |  9999 |
| 20 | Data Structure In Depth       | Ashok Kalia     | 15000 |
| 21 | JQuery Ninja                  | James Guido     |  6500 |
| 22 | Twitter Bootstrap             | Mitali Ghosh    | 14999 |
| 23 | Codeigniter                   | Pawan Kumar     |  7599 |
| 24 | Struts Framework              | Umesh Verma     |  7500 |
| 25 | CakePHP                       | Parvez Khan     | 60000 |
| 26 | Machine Learning              | Faisal Qureshi  | 45000 |
| 27 | Computer Networks             | Saleem Khan     | 12599 |
| 28 | C++ STL Library Tutorial      | Kareem Sheikh   | 25000 |
| 29 | Kali Linux                    | Jitan Majhi     |  4500 |
| 30 | SQLite Tutorial               | Nitish Kumar    |  6500 |
+----+-------------------------------+-----------------+-------+
30 rows in set (0.00 sec)
```

DELETE FROM courses WHERE id = 1;
Query OK, 1 row affected (0.01 sec)

SELECT * FROM courses;

```
+----+-------------------------------+-----------------+-------+
| id | course_name                   | instructor_name | fees  |
+----+-------------------------------+-----------------+-------+
|  2 | PHP Development               | Dolly Singh     |  1500 |
|  3 | Java Complete Tutorial        | Ramesh Yadav    |  7500 |
|  4 | Swing GUI In Depth            | Guddu Sharma    | 15000 |
|  5 | Computer Vision Using Python  | Narendra Murthy | 25000 |
|  6 | AI In Real World Using Python | Satya Kundu     | 45000 |
|  7 | 2D Games Using PyGame         | Sandeep Ganguly | 18000 |
|  8 | GIMP Photo Editing            | Rachna Mishra   |  5000 |
|  9 | HTML                          | Chatur Singh    |  1000 |
| 10 | CSS3                          | pinky singh     |  1500 |
| 11 | Amazon Cloud AWS              | Ruchi Singhania | 75000 |
| 12 | Hadoop Big Data               | Ankita Ganguly  | 95000 |
| 13 | Natural Language Processing   | Sandeep Ganguly | 45999 |
| 14 | The C Ninja                   | Pradeep Gurung  |  3599 |
| 15 | Java Database Connectivity    | Ratan Tata      |  6599 |
| 16 | Dynamic Website Development   | Girish Patel    |  8599 |
| 17 | Android App Development       | Rishi Khanna    | 17999 |
| 18 | IOS Developer                 | Umesh Verma     | 25000 |
| 19 | Algorithms In Depth           | Arjun Thapa     |  9999 |
| 20 | Data Structure In Depth       | Ashok Kalia     | 15000 |
| 21 | JQuery Ninja                  | James Guido     |  6500 |
| 22 | Twitter Bootstrap             | Mitali Ghosh    | 14999 |
| 23 | Codeigniter                   | Pawan Kumar     |  7599 |
| 24 | Struts Framework              | Umesh Verma     |  7500 |
| 25 | CakePHP                       | Parvez Khan     | 60000 |
| 26 | Machine Learning              | Faisal Qureshi  | 45000 |
| 27 | Computer Networks             | Saleem Khan     | 12599 |
| 28 | C++ STL Library Tutorial      | Kareem Sheikh   | 25000 |
| 29 | Kali Linux                    | Jitan Majhi     |  4500 |
| 30 | SQLite Tutorial               | Nitish Kumar    |  6500 |
+----+-------------------------------+-----------------+-------+
29 rows in set (0.00 sec)
```

-- NOTE - WHERE CLAUSE IS NECESSARY IF ITS NOT PROVIDED THEN IT DELETE ALL DATA FROM TABLE;
DELETE FROM courses;

Query OK, 29 rows affected (0.01 sec)

SELECT * FROM courses;
Empty set (0.00 sec)

/*---------- ALTER OPERATION ----------*/

SELECT * FROM marks;

id	course_name	score	sid
1	java from scarch	14	38
2	AI In Real World Using Python	99	51
3	GIMP Photo Editing	51	29
4	SQLite Tutorial	57	13
5	Swing GUI In Depth	31	34
6	CSS3	83	36
7	Codeigniter	22	39
8	2D Games Using PyGame	60	37
9	Data Structure In Depth	31	32
10	Amazon Cloud AWS	76	12
11	Amazon Cloud AWS	86	27
12	Computer Networks	2	34
13	Swing GUI In Depth	52	1
14	Dynamic Website Development	52	3
15	CakePHP	2	27
16	Natural Language Processing	55	6
17	HTML	68	6
18	Computer Vision Using Python	72	47
19	Computer Vision Using Python	58	24
20	Data Structure In Depth	70	14
21	Kali Linux	76	18
22	Java Complete Tutorial	67	32
23	Java Database Connectivity	8	25
24	The C Ninja	39	10
25	SQLite Tutorial	69	60

25 rows in set (0.00 sec)

ALTER TABLE marks ADD grade varchar(1) NOT NULL;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [gangulytech]> SELECT * FROM marks;

id	course_name	score	sid	grade
1	java from scarch	14	38	
2	AI In Real World Using Python	99	51	
3	GIMP Photo Editing	51	29	
4	SQLite Tutorial	57	13	
5	Swing GUI In Depth	31	34	
6	CSS3	83	36	
7	Codeigniter	22	39	
8	2D Games Using PyGame	60	37	
9	Data Structure In Depth	31	32	
10	Amazon Cloud AWS	76	12	
11	Amazon Cloud AWS	86	27	
12	Computer Networks	2	34	
13	Swing GUI In Depth	52	1	

```
| 14 | Dynamic Website Development   |    52 |   3 |       |
| 15 | CakePHP                       |     2 |  27 |       |
| 16 | Natural Language Processing   |    55 |   6 |       |
| 17 | HTML                          |    68 |   6 |       |
| 18 | Computer Vision Using Python  |    72 |  47 |       |
| 19 | Computer Vision Using Python  |    58 |  24 |       |
| 20 | Data Structure In Depth       |    70 |  14 |       |
| 21 | Kali Linux                    |    76 |  18 |       |
| 22 | Java Complete Tutorial        |    67 |  32 |       |
| 23 | Java Database Connectivity    |     8 |  25 |       |
| 24 | The C Ninja                   |    39 |  10 |       |
| 25 | SQLite Tutorial               |    69 |  60 |       |
+----+-------------------------------+-------+-----+-------+
25 rows in set (0.00 sec)
```

ALTER TABLE marks MODIFY grade TEXT;
Query OK, 25 rows affected (0.07 sec)
Records: 25 Duplicates: 0 Warnings: 0

MariaDB [gangulytech]> SELECT * FROM marks;

```
+----+-------------------------------+-------+-----+-------+
| id | course_name                   | score | sid | grade |
+----+-------------------------------+-------+-----+-------+
|  1 | java from scarch              |    14 |  38 |       |
|  2 | AI In Real World Using Python |    99 |  51 |       |
|  3 | GIMP Photo Editing            |    51 |  29 |       |
|  4 | SQLite Tutorial               |    57 |  13 |       |
|  5 | Swing GUI In Depth            |    31 |  34 |       |
|  6 | CSS3                          |    83 |  36 |       |
|  7 | Codeigniter                   |    22 |  39 |       |
|  8 | 2D Games Using PyGame         |    60 |  37 |       |
|  9 | Data Structure In Depth       |    31 |  32 |       |
| 10 | Amazon Cloud AWS              |    76 |  12 |       |
| 11 | Amazon Cloud AWS              |    86 |  27 |       |
| 12 | Computer Networks             |     2 |  34 |       |
| 13 | Swing GUI In Depth            |    52 |   1 |       |
| 14 | Dynamic Website Development   |    52 |   3 |       |
| 15 | CakePHP                       |     2 |  27 |       |
| 16 | Natural Language Processing   |    55 |   6 |       |
| 17 | HTML                          |    68 |   6 |       |
| 18 | Computer Vision Using Python  |    72 |  47 |       |
| 19 | Computer Vision Using Python  |    58 |  24 |       |
| 20 | Data Structure In Depth       |    70 |  14 |       |
| 21 | Kali Linux                    |    76 |  18 |       |
| 22 | Java Complete Tutorial        |    67 |  32 |       |
| 23 | Java Database Connectivity    |     8 |  25 |       |
| 24 | The C Ninja                   |    39 |  10 |       |
| 25 | SQLite Tutorial               |    69 |  60 |       |
+----+-------------------------------+-------+-----+-------+
25 rows in set (0.00 sec)
```

ALTER TABLE marks DROP grade;
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [gangulytech]> SELECT * FROM marks;

```
+----+-------------------------------+-------+-----+
| id | course_name                   | score | sid |
+----+-------------------------------+-------+-----+
```

```
|  1 | java from scarch              |    14 |  38 |
|  2 | AI In Real World Using Python |    99 |  51 |
|  3 | GIMP Photo Editing            |    51 |  29 |
|  4 | SQLite Tutorial               |    57 |  13 |
|  5 | Swing GUI In Depth            |    31 |  34 |
|  6 | CSS3                          |    83 |  36 |
|  7 | Codeigniter                   |    22 |  39 |
|  8 | 2D Games Using PyGame         |    60 |  37 |
|  9 | Data Structure In Depth       |    31 |  32 |
| 10 | Amazon Cloud AWS              |    76 |  12 |
| 11 | Amazon Cloud AWS              |    86 |  27 |
| 12 | Computer Networks             |     2 |  34 |
| 13 | Swing GUI In Depth            |    52 |   1 |
| 14 | Dynamic Website Development   |    52 |   3 |
| 15 | CakePHP                       |     2 |  27 |
| 16 | Natural Language Processing   |    55 |   6 |
| 17 | HTML                          |    68 |   6 |
| 18 | Computer Vision Using Python  |    72 |  47 |
| 19 | Computer Vision Using Python  |    58 |  24 |
| 20 | Data Structure In Depth       |    70 |  14 |
| 21 | Kali Linux                    |    76 |  18 |
| 22 | Java Complete Tutorial        |    67 |  32 |
| 23 | Java Database Connectivity    |     8 |  25 |
| 24 | The C Ninja                   |    39 |  10 |
| 25 | SQLite Tutorial               |    69 |  60 |
+----+-------------------------------+-------+-----+
25 rows in set (0.00 sec)
```

/*---------- JOIN OPERATION ----------*/

-- NOTE - AND YOU CAN ALSO JOIN THREE AND MORE NOT JUST A TWO TABLES.
-- By Normal Way you can also JOIN THE MORE TABLES And this fully qualified query is use in JOIN OPERATION.

```
SELECT students.id, marks.sid, students.name, marks.course_name, marks.score FROM marks, students
WHERE students.id = marks.sid;
```

```
+----+-----+------------------+-------------------------------+-------+
| id | sid | name             | course_name                   | score |
+----+-----+------------------+-------------------------------+-------+
| 38 |  38 | nana patekar     | AI In Real World Using Python |    14 |
| 51 |  51 | sujeet thapa     | AI In Real World Using Python |    99 |
| 29 |  29 | sbhubendu sarkar | GIMP Photo Editing            |    51 |
| 13 |  13 | shiv patel       | SQLite Tutorial               |    57 |
| 34 |  34 | shweta ghara     | Swing GUI In Depth            |    31 |
| 36 |  36 | sumita ganguly   | CSS3                          |    83 |
| 39 |  39 | nitin gadkari    | Codeigniter                   |    22 |
| 37 |  37 | sumit thakrey    | 2D Games Using PyGame         |    60 |
| 32 |  32 | rupoma biswas    | Advance Data Structure        |    55 |
| 12 |  12 | Arun Bhatia      | Amazon Cloud AWS              |    76 |
| 27 |  27 | shubham das      | Amazon Cloud AWS              |    86 |
| 34 |  34 | shweta ghara     | Computer Networks             |     2 |
|  1 |   1 | sandeep ganguly  | Swing GUI In Depth            |    52 |
|  3 |   3 | divyanshu shukla | Dynamic Website Development   |    52 |
| 27 |  27 | shubham das      | CakePHP                       |     2 |
|  6 |   6 | siddhartha singh | Natural Language Processing   |    55 |
|  6 |   6 | siddhartha singh | HTML                          |    68 |
| 47 |  47 | asif sheikh      | Computer Vision Using Python  |    72 |
| 24 |  24 | mamta banerjee   | Computer Vision Using Python  |    58 |
| 14 |  14 | aman ali         | Data Structure In Depth       |    70 |
| 18 |  18 | Ankur sharma     | Kali Linux                    |    76 |
```

```
| 32 |  32 | rupoma biswas    | Java Complete Tutorial        |    67 |
| 25 |  25 | dolly ganguly    | Java Database Connectivity    |     8 |
| 10 |  10 | ayushi sharma    | The C Ninja                   |    39 |
| 60 |  60 | sudhir chaudhary | SQLite Tutorial               |    69 |
+----+-----+------------------+-------------------------------+-------+
25 rows in set (0.01 sec)
```

-- now do it by Join Operations.

SELECT students.id, marks.sid, students.name, marks.course_name, marks.score FROM marks INNER JOIN students ON students.id = marks.sid;

```
+----+-----+------------------+-------------------------------+-------+
| id | sid | name             | course_name                   | score |
+----+-----+------------------+-------------------------------+-------+
| 38 |  38 | nana patekar     | AI In Real World Using Python |    14 |
| 51 |  51 | sujeet thapa     | AI In Real World Using Python |    99 |
| 29 |  29 | sbhubendu sarkar | GIMP Photo Editing            |    51 |
| 13 |  13 | shiv patel       | SQLite Tutorial               |    57 |
| 34 |  34 | shweta ghara     | Swing GUI In Depth            |    31 |
| 36 |  36 | sumita ganguly   | CSS3                          |    83 |
| 39 |  39 | nitin gadkari    | Codeigniter                   |    22 |
| 37 |  37 | sumit thakrey    | 2D Games Using PyGame         |    60 |
| 32 |  32 | rupoma biswas    | Advance Data Structure        |    55 |
| 12 |  12 | Arun Bhatia      | Amazon Cloud AWS              |    76 |
| 27 |  27 | shubham das      | Amazon Cloud AWS              |    86 |
| 34 |  34 | shweta ghara     | Computer Networks             |     2 |
|  1 |   1 | sandeep ganguly  | Swing GUI In Depth            |    52 |
|  3 |   3 | divyanshu shukla | Dynamic Website Development   |    52 |
| 27 |  27 | shubham das      | CakePHP                       |     2 |
|  6 |   6 | siddhartha singh | Natural Language Processing   |    55 |
|  6 |   6 | siddhartha singh | HTML                          |    68 |
| 47 |  47 | asif sheikh      | Computer Vision Using Python  |    72 |
| 24 |  24 | mamta banerjee   | Computer Vision Using Python  |    58 |
| 14 |  14 | aman ali         | Data Structure In Depth       |    70 |
| 18 |  18 | Ankur sharma     | Kali Linux                    |    76 |
| 32 |  32 | rupoma biswas    | Java Complete Tutorial        |    67 |
| 25 |  25 | dolly ganguly    | Java Database Connectivity    |     8 |
| 10 |  10 | ayushi sharma    | The C Ninja                   |    39 |
| 60 |  60 | sudhir chaudhary | SQLite Tutorial               |    69 |
+----+-----+------------------+-------------------------------+-------+
25 rows in set (0.00 sec)
```

-- LEFT OUTER JOIN

SELECT courses.id, marks.sid, courses.course_name, marks.course_name, marks.score FROM marks LEFT OUTER JOIN courses ON courses.id = marks.sid;

```
+------+-----+-------------------------------+----------------------------
--+-------+
| id   | sid | course_name                   | course_name
| score |
+------+-----+-------------------------------+----------------------------
--+-------+
| NULL |  38 | NULL                          | AI In Real World Using
Python |    14 |
| NULL |  51 | NULL                          | AI In Real World Using
Python |    99 |
|   29 |  29 | Kali Linux                    | GIMP Photo Editing
|    51 |
|   13 |  13 | Natural Language Processing   | SQLite Tutorial
|    57 |
| NULL |  34 | NULL                          | Swing GUI In Depth
|    31 |
```

```
| NULL |  36 | NULL                          | CSS3
|    83 |
| NULL |  39 | NULL                          | Codeigniter
|    22 |
| NULL |  37 | NULL                          | 2D Games Using PyGame
|    60 |
| NULL |  32 | NULL                          | Advance Data Structure
|    55 |
|   12 |  12 | Hadoop Big Data               | Amazon Cloud AWS
|    76 |
|   27 |  27 | Computer Networks             | Amazon Cloud AWS
|    86 |
| NULL |  34 | NULL                          | Computer Networks
|     2 |
|    1 |   1 | MySQL Database                | Swing GUI In Depth
|    52 |
|    3 |   3 | Java Complete Tutorial        | Dynamic Website Development
|    52 |
|   27 |  27 | Computer Networks             | CakePHP
|     2 |
|    6 |   6 | AI In Real World Using Python | Natural Language Processing
|    55 |
|    6 |   6 | AI In Real World Using Python | HTML
|    68 |
| NULL |  47 | NULL                          | Computer Vision Using Python
|    72 |
|   24 |  24 | Struts Framework              | Computer Vision Using Python
|    58 |
|   14 |  14 | The C Ninja                   | Data Structure In Depth
|    70 |
|   18 |  18 | IOS Developer                 | Kali Linux
|    76 |
| NULL |  32 | NULL                          | Java Complete Tutorial
|    67 |
|   25 |  25 | CakePHP                       | Java Database Connectivity
|     8 |
|   10 |  10 | CSS3                          | The C Ninja
|    39 |
| NULL |  60 | NULL                          | SQLite Tutorial
|    69 |
+------+-----+-------------------------------+----------------------------
--+-------+
25 rows in set (0.00 sec)
```

//RIGHT OUTER JOIN

```
SELECT courses.id, marks.sid, courses.course_name, marks.course_name, marks.score
FROM courses RIGHT OUTER JOIN marks ON courses.id = marks.sid;
+------+-----+-------------------------------+----------------------------
--+-------+
| id   | sid | course_name                   | course_name
| score |
+------+-----+-------------------------------+----------------------------
--+-------+
| NULL |  38 | NULL                          | AI In Real World Using
Python |    14 |
| NULL |  51 | NULL                          | AI In Real World Using
Python |    99 |
|   29 |  29 | Kali Linux                    | GIMP Photo Editing
|    51 |
|   13 |  13 | Natural Language Processing   | SQLite Tutorial
|    57 |
```

```
| NULL |  34 | NULL                           | Swing GUI In Depth
|    31 |
| NULL |  36 | NULL                           | CSS3
|    83 |
| NULL |  39 | NULL                           | Codeigniter
|    22 |
| NULL |  37 | NULL                           | 2D Games Using PyGame
|    60 |
| NULL |  32 | NULL                           | Advance Data Structure
|    55 |
|   12 |  12 | Hadoop Big Data                | Amazon Cloud AWS
|    76 |
|   27 |  27 | Computer Networks              | Amazon Cloud AWS
|    86 |
| NULL |  34 | NULL                           | Computer Networks
|     2 |
|    1 |   1 | MySQL Database                 | Swing GUI In Depth
|    52 |
|    3 |   3 | Java Complete Tutorial         | Dynamic Website Development
|    52 |
|   27 |  27 | Computer Networks              | CakePHP
|     2 |
|    6 |   6 | AI In Real World Using Python | Natural Language Processing
|    55 |
|    6 |   6 | AI In Real World Using Python | HTML
|    68 |
| NULL |  47 | NULL                           | Computer Vision Using Python
|    72 |
|   24 |  24 | Struts Framework               | Computer Vision Using Python
|    58 |
|   14 |  14 | The C Ninja                    | Data Structure In Depth
|    70 |
|   18 |  18 | IOS Developer                  | Kali Linux
|    76 |
| NULL |  32 | NULL                           | Java Complete Tutorial
|    67 |
|   25 |  25 | CakePHP                        | Java Database Connectivity
|     8 |
|   10 |  10 | CSS3                           | The C Ninja
|    39 |
| NULL |  60 | NULL                           | SQLite Tutorial
|    69 |
+------+-----+--------------------------------+----------------------------
--+-------+
25 rows in set (0.00 sec)
```

/*---------- GROUP BY ----------*/

SELECT * FROM students GROUP BY state;

```
+----+----------------+----------+-------+---------+
| id | name           | city     | state | pincode |
+----+----------------+----------+-------+---------+
| 13 | shiv patel     | surat    | GJ    |  197988 |
| 16 | deepak yadav   | gurugram | HR    |  634419 |
| 17 | manjul saini   | dhanbad  | JH    |  860186 |
| 10 | ayushi sharma  | jammu    | JK    |  859431 |
|  8 | pawan kumar    | banglore | KA    |    2612 |
| 11 | shameem beg    | mumbai   | MH    |  776793 |
| 49 | guddu thomas   | imphal   | MN    |  920423 |
| 43 | disha chandok  | ludhiana | PB    |  296055 |
| 14 | aman ali       | ajmer    | RJ    |   72920 |
```

```
|  2 | piyush chandel  | nainital | UK    |  549386 |
|  1 | sandeep ganguly | kanpur   | UP    |  601988 |
|  9 | umesh verma     | kolkata  | WB    |  173453 |
+----+-----------------+----------+-------+---------+
12 rows in set (0.00 sec)
```

-- it is basically noramlly used with aggregate function
-- it is used to find no of user from group
SELECT state,COUNT(state) AS NoOfStudents FROM students GROUP BY state;

```
+-------+--------------+
| state | NoOfStudents |
+-------+--------------+
| GJ    |            4 |
| HR    |            3 |
| JH    |            3 |
| JK    |            4 |
| KA    |            2 |
| MH    |            9 |
| MN    |            4 |
| PB    |            4 |
| RJ    |            2 |
| UK    |            1 |
| UP    |            9 |
| WB    |           15 |
+-------+--------------+
12 rows in set (0.00 sec)
```

-- "HAVING CLAUSE NOTE:- you can't use where clause with group by so having is used."
SELECT state,COUNT(state) AS NoOfStudents FROM students GROUP BY state HAVING count(state) > 5;

```
+-------+--------------+
| state | NoOfStudents |
+-------+--------------+
| MH    |            9 |
| UP    |            9 |
| WB    |           15 |
+-------+--------------+
3 rows in set (0.00 sec)
```

/* ---------- CREATE A VIEW ---------- */
CREATE VIEW StudCourse AS SELECT course_name,instructor_name,fees FROM courses ASEC WHERE fees >= 5000 LIMIT 5;
Query OK, 0 rows affected (0.01 sec)

select * from StudCourse;

```
+------------------------------+-----------------+-------+
| course_name                  | instructor_name | fees  |
+------------------------------+-----------------+-------+
| Java Complete Tutorial       | Ramesh Yadav    |  7500 |
| Swing GUI In Depth           | Guddu Sharma    | 15000 |
| Computer Vision Using Python | Narendra Murthy | 25000 |
| AI In Real World Using Python | Satya Kundu     | 45000 |
| 2D Games Using PyGame        | Sandeep Ganguly | 18000 |
+------------------------------+-----------------+-------+
5 rows in set (0.00 sec)
```

```
 SHOW TABLES;
+----------------------+
| Tables_in_gangulytech |
+----------------------+
```

```
| courses               |
| enrolls               |
| marks                 |
| studcourse            |
| students              |
+-----------------------+
5 rows in set (0.00 sec)
```

//even you make changes in main table its reflected in view also
update courses SET fees = 6500 Where instructor_name = "Ramesh Yadav";
Rows matched: 1 Changed: 1 Warnings: 0

SELECT * FROM courses ASEC LIMIT 5;

```
+----+------------------------------+-----------------+-------+
| id | course_name                  | instructor_name | fees  |
+----+------------------------------+-----------------+-------+
|  1 | MySQL Database               | Sandeep Ganguly |  4500 |
|  2 | PHP Development              | Dolly Singh     |  1500 |
|  3 | Java Complete Tutorial       | Ramesh Yadav    |  6500 |
|  4 | Swing GUI In Depth           | Guddu Sharma    | 15000 |
|  5 | Computer Vision Using Python | Narendra Murthy | 25000 |
+----+------------------------------+-----------------+-------+
5 rows in set (0.00 sec)
```

//now look for view its auto updates its contents. and you can see the main difference.
SELECT * FROM studcourse;

```
+------------------------------+-----------------+-------+
| course_name                  | instructor_name | fees  |
+------------------------------+-----------------+-------+
| Java Complete Tutorial       | Ramesh Yadav    |  6500 |
| Swing GUI In Depth           | Guddu Sharma    | 15000 |
| Computer Vision Using Python | Narendra Murthy | 25000 |
| AI In Real World Using Python | Satya Kundu     | 45000 |
| 2D Games Using PyGame        | Sandeep Ganguly | 18000 |
+------------------------------+-----------------+-------+
5 rows in set (0.00 sec)
```

TRANSACTION

A Transaction is a sequence of one or more SQL statements that together form a logical unit of work
Every statemnet that forms a transaction executes certain portions of an operation.
All the statemnets must complete successfully to conclude the operation
A Transaction must pass the ACID test
If any of the tasks fail, the transaction fails. Therefore, a transaction has only two results: success or failure.

Example
First CREATE TABLE and also need to store in INNODB Storage for TRANSACTION Purpose

```
CREATE TABLE users (
        id INT NOT NULL AUTO_INCREMENT,
        name VARCHAR(15) NOT NULL,
        isActive BOOLEAN NOT NULL,
        message TEXT NOT NULL,
        gender ENUM( 'Male', 'Female' ) NOT NULL,
        createdOn DATETIME NOT NULL,
        balance DOUBLE NOT NULL,
        PRIMARY KEY(id)
)ENGINE=INNODB;
```

LETS INSERT SOME DATA INTO IT

```
INSERT INTO users (name, isActive, message, gender, createdOn, balance)
VALUE ('Sandeep Ganguly', 1, 'Hello from sandeep!', 'Male', NOW(), 75000.45),
("Mayur Kadam",0,"Hello World","Male",NOW(),22135.0),
("MIghty Kadam",0,"Hello Prison","Male",NOW(),2222135.0);
```

LETS TRY TRANSACTION COMMAND ON IT

```
START TRANSACTION
DELETE from users;
SELECT * from users;
ROLLBACK;
```

the above query will delete all column value later due to rollback it will undo it

NOW LOOK FOR COMMIT

```
START TRANSACTION
DELETE from users;
SELECT * from users;
COMMIT;
```

NOW CHNAGES has been saved it sucessfully updated into database...
You Can Exec Multiple query in to as per the defination and you can also exec single also...

SQL Functions

SQL functions are the set of built-in functions to perform a calculation over data that are stored in the table. Let us have a look at the list of most useful SQL functions.

1. SQL Count – returns the count of rows in a database table.
2. SQL Max – returns the maximum value from a database table
3. SQL Min – returns the minimum value from a database table
4. SQL Avg – provides the average of a certain table column value
5. SQL Sum – provides the sun of a certain table column value
6. SQL sqrt – returns the square root of a number.
7. SQL rand – used to generate a random number using SQL command.
8. SQL concat – used for concatenating strings in a SQL command.
9. SQL Ucase – converts a field to upper case.
10. SQL Lcase – converts a field to lower case.

INDEX Statement

CREATE INDEX Statement

The CREATE INDEX statement is used to create indexes in tables.
Indexes are used to retrieve data from the database very fast.
The users cannot see the indexes, they are just used to speed up searches/queries.

CREATE INDEX Syntax
Creates an index on a table. Duplicate values are allowed:
CREATE INDEX index_name
ON table_name (column1, column2, ...);

CREATE UNIQUE INDEX Syntax
Creates a unique index on a table. Duplicate values are not allowed:
CREATE UNIQUE INDEX index_name
ON table_name (column1, column2, ...);
Note: The syntax for creating indexes varies among different databases.
Therefore: Check the syntax for creating indexes in your database.

CREATE INDEX Example
The SQL statement below creates an index named "idx_lastname" on the "LastName" column in the "Persons" table:
CREATE INDEX idx_lastname
ON Persons (LastName);

If you want to create an index on a combination of columns, you can list the column names within the parentheses, separated by commas:
CREATE INDEX idx_pname
ON Persons (LastName, FirstName);

DROP INDEX Statement
The DROP INDEX statement is used to delete an index in a table.

The DROP INDEX statement is used to delete an index in a table.
MS Access:
DROP INDEX index_name ON table_name;

SQL Server:
DROP INDEX table_name.index_name;

DB2/Oracle:
DROP INDEX index_name;

MySQL:
ALTER TABLE table_name
DROP INDEX index_name;

Example

SELECT * FROM marks;

```
+----+-------------------------------+-------+-----+
| id | course_name                   | score | sid |
+----+-------------------------------+-------+-----+
|  1 | AI In Real World Using Python |    14 |  38 |
|  2 | AI In Real World Using Python |    99 |  51 |
|  3 | GIMP Photo Editing            |    51 |  29 |
|  4 | SQLite Tutorial               |    57 |  13 |
|  5 | Swing GUI In Depth            |    31 |  34 |
|  6 | CSS3                          |    83 |  36 |
|  7 | Codeigniter                   |    22 |  39 |
|  8 | 2D Games Using PyGame         |    60 |  37 |
|  9 | Data Structure In Depth       |    31 |  32 |
| 10 | Amazon Cloud AWS              |    76 |  12 |
| 11 | Amazon Cloud AWS              |    86 |  27 |
| 12 | Computer Networks             |     2 |  34 |
| 13 | Swing GUI In Depth            |    52 |   1 |
| 14 | Dynamic Website Development   |    52 |   3 |
| 15 | CakePHP                       |     2 |  27 |
| 16 | Natural Language Processing   |    55 |   6 |
| 17 | HTML                          |    68 |   6 |
| 18 | Computer Vision Using Python  |    72 |  47 |
| 19 | Computer Vision Using Python  |    58 |  24 |
| 20 | Data Structure In Depth       |    70 |  14 |
| 21 | Kali Linux                    |    76 |  18 |
| 22 | Java Complete Tutorial        |    67 |  32 |
| 23 | Java Database Connectivity    |     8 |  25 |
| 24 | The C Ninja                   |    39 |  10 |
| 25 | SQLite Tutorial               |    69 |  60 |
+----+-------------------------------+-------+-----+
25 rows in set (0.01 sec)
```

-- CREATE INDEX Example
CREATE INDEX idx_score ON marks(score);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

-- View INDEX Example
SHOW INDEX FROM marks;

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

```
| Table | Non_unique | Key_name  | Seq_in_index | Column_name | Collation |
Cardinality | Sub_part | Packed | Null | Index_type | Comment |
Index_comment |
+-------+------------+-----------+--------------+-------------+-----------
+-------------+----------+--------+------+------------+---------+----------
-----+
| marks |          0 | PRIMARY   |            1 | id          | A         |
25 |     NULL | NULL   |      | BTREE      |         |               |
| marks |          1 | idx_score |            1 | score       | A         |
25 |     NULL | NULL   |      | BTREE      |         |               |
+-------+------------+-----------+--------------+-------------+-----------
+-------------+----------+--------+------+------------+---------+----------
-----+
2 rows in set (0.00 sec)
```

-- DROP INDEX Example
DROP INDEX idx_score ON marks;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

-- AFTER THAT IT only shows primary key index
-- AND primary key implicitly contain indexes
SHOW INDEX FROM marks;

```
+-------+------------+----------+--------------+-------------+-----------+-
------------+----------+--------+------+------------+---------+------------
---+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation |
Cardinality | Sub_part | Packed | Null | Index_type | Comment |
Index_comment |
+-------+------------+----------+--------------+-------------+-----------+-
------------+----------+--------+------+------------+---------+------------
---+
| marks |          0 | PRIMARY  |            1 | id          | A         |
25 |     NULL | NULL   |      | BTREE      |         |               |
+-------+------------+----------+--------------+-------------+-----------+-
------------+----------+--------+------+------------+---------+------------
---+
1 row in set (0.00 sec)
```

SQL Constraints

SQL Constraints are rules used to limit the type of data that can go into a table,
to maintain the accuracy and integrity of the data inside table.

Constraints can be divided into the following two types,
-- **Column level constraints**: Limits only column data.
-- **Table level constraints**: Limits whole table data.

Following are the most used constraints that can be applied to a table.
-- NOT NULL
-- UNIQUE
-- PRIMARY KEY
-- FOREIGN KEY
-- CHECK
-- DEFAULT

NOT NULL Constraint

NOT NULL constraint restricts a column from having a NULL value.
Once NOT NULL constraint is applied to a column,
you cannot pass a null value to that column. It enforces a column to contain a proper value.
One important point to note about this constraint is that it cannot be defined at table level.

Example using NOT NULL constraint
CREATE TABLE Student(s_id int NOT NULL, Name varchar(60), Age int);

UNIQUE Constraint
UNIQUE constraint ensures that a field or column will only have unique values. A UNIQUE constraint field will not have duplicate data.
This constraint can be applied at column level or table level.

Using UNIQUE constraint when creating a Table (Table Level)
Here we have a simple CREATE query to create a table, which will have a column s_id with unique values.
CREATE TABLE Student(s_id int NOT NULL UNIQUE, Name varchar(60), Age int);
The above query will declare that the s_id field of Student table will only have unique values and wont take NULL value.

Using UNIQUE constraint after Table is created (Column Level)
ALTER TABLE Student ADD UNIQUE(s_id);
The above query specifies that s_id field of Student table will only have unique value.

Primary Key Constraint
Primary key constraint uniquely identifies each record in a database. A Primary Key must contain unique value and it must not contain null value.
Usually Primary Key is used to index the data inside the table.

Using PRIMARY KEY constraint at Table Level
CREATE table Student (s_id int PRIMARY KEY, Name varchar(60) NOT NULL, Age int);
The above command will creates a PRIMARY KEY on the s_id.

Using PRIMARY KEY constraint at Column Level
ALTER table Student ADD PRIMARY KEY (s_id);
The above command will creates a PRIMARY KEY on the s_id.

Foreign Key Constraint
FOREIGN KEY is used to relate two tables.
FOREIGN KEY constraint is also used to restrict actions that would destroy links between tables.

Using FOREIGN KEY constraint at Table Level
```
CREATE table Order_Detail(
   order_id int PRIMARY KEY,
   order_name varchar(60) NOT NULL,
   c_id int FOREIGN KEY REFERENCES Customer_Detail(c_id)
);
```
In this query, c_id in table Order_Detail is made as foriegn key, which is a reference of c_id column in Customer_Detail table.

Using FOREIGN KEY constraint at Column Level
ALTER table Order_Detail ADD FOREIGN KEY (c_id) REFERENCES Customer_Detail(c_id);

CHECK Constraint

CHECK constraint is used to restrict the value of a column between a range. It performs check on the values, before storing them into the database. Its like condition checking before saving data into a column.

Using CHECK constraint at Table Level

```
CREATE table Student(
   s_id int NOT NULL CHECK(s_id > 0),
   Name varchar(60) NOT NULL,
   Age int
);
```

The above query will restrict the s_id value to be greater than zero.

Using CHECK constraint at Column Level

```
ALTER table Student ADD CHECK(s_id > 0);
```

DEFAULT Constraint

The DEFAULT constraint is used to provide a default value for a column.
The default value will be added to all new records IF no other value is specified.

SQL DEFAULT on CREATE TABLE

The following SQL sets a DEFAULT value for the "City" column when the "Persons" table is created:

My SQL / SQL Server / Oracle / MS Access:

```
CREATE TABLE Persons (
   ID int NOT NULL,
   LastName varchar(255) NOT NULL,
   FirstName varchar(255),
   Age int,
   City varchar(255) DEFAULT 'Sandnes'
);
```

The DEFAULT constraint can also be used to insert system values, by using functions like GETDATE():

```
CREATE TABLE Orders (
   ID int NOT NULL,
   OrderNumber int NOT NULL,
   OrderDate date DEFAULT GETDATE()
);
```

SQL DEFAULT on ALTER TABLE

To create a DEFAULT constraint on the "City" column when the table is already created, use the following SQL:

MySQL:

```
ALTER TABLE Persons
ALTER City SET DEFAULT 'Sandnes';
```

DROP a DEFAULT Constraint

To drop a DEFAULT constraint, use the following SQL:

MySQL:

```
ALTER TABLE Persons
ALTER City DROP DEFAULT;
```

SQL Stored Procedures

A stored procedure is a prepared SQL code that you can save,
so the code can be reused over and over again.

So if you have an SQL query that you write over and over again,
save it as a stored procedure,
and then just call it to execute it.

You can also pass parameters to a stored procedure,
so that the stored procedure can act based on the parameter value(s) that is passed.

-- Stored Procedure Syntax
delimiter //
CREATE PROCEDURE procedure_name()
BEGIN sql_statement; END//

-- Execute a Stored Procedure
EXEC procedure_name;

Demo Database

SELECT * FROM marks;

id	course_name	score	sid
1	AI In Real World Using Python	14	38
2	AI In Real World Using Python	99	51
3	GIMP Photo Editing	51	29
4	SQLite Tutorial	57	13
5	Swing GUI In Depth	31	34
6	CSS3	83	36
7	Codeigniter	22	39
8	2D Games Using PyGame	60	37
9	Data Structure In Depth	31	32
10	Amazon Cloud AWS	76	12
11	Amazon Cloud AWS	86	27
12	Computer Networks	2	34
13	Swing GUI In Depth	52	1
14	Dynamic Website Development	52	3
15	CakePHP	2	27
16	Natural Language Processing	55	6
17	HTML	68	6
18	Computer Vision Using Python	72	47
19	Computer Vision Using Python	58	24
20	Data Structure In Depth	70	14
21	Kali Linux	76	18
22	Java Complete Tutorial	67	32
23	Java Database Connectivity	8	25
24	The C Ninja	39	10
25	SQLite Tutorial	69	60

25 rows in set (0.00 sec)

Create Store Procedure
delimiter //
create procedure foobar()
begin select * from marks; end//
Query OK, 0 rows affected (0.00 sec)

delimiter ;

Set the delimiter back and look at the procedure:
SHOW PROCEDURE status;

```
+-------------+--------+-----------+---------------+--------------------
+---------------------+--------------+---------+----------------------+---
-------------------+--------------------+
| Db          | Name   | Type      | Definer        | Modified            |
Created             | Security_type | Comment | character_set_client |
collation_connection | Database Collation |
+-------------+--------+-----------+---------------+--------------------
+---------------------+--------------+---------+----------------------+---
-------------------+--------------------+
| gangulytech | foobar | PROCEDURE | root@localhost | 2019-06-04 11:53:46 |
2019-06-04 11:53:46 | DEFINER       |         | cp850                |
cp850_general_ci     | latin1_swedish_ci  |
+-------------+--------+-----------+---------------+--------------------
+---------------------+--------------+---------+----------------------+---
-------------------+--------------------+
1 row in set (0.00 sec)
```

Another Way

select * from employee;

```
+--------+----------+---------+--------+
| emp_id | emp_name | dept_id | salary |
+--------+----------+---------+--------+
|    103 | Jack     |       1 |   1400 |
|    104 | John     |       2 |   1450 |
|    108 | Alan     |       3 |   1150 |
|    107 | Ram      |    NULL |    600 |
+--------+----------+---------+--------+
4 rows in set (0.22 sec)
```

mysql> DELIMITER //
mysql> create procedure usp_totalEmployeeByDeparment(IN id INT)
   -> begin
   -> select count(*) as total from employee where dept_id = id;
   -> end//
Query OK, 0 rows affected (0.00 sec)

mysql> DELIMITER ;

mysql> call usp_totalEmployeeByDeparment(2);

```
+-------+
| total |
+-------+
|     1 |
+-------+
1 row in set (0.06 sec)
```

Creating and Calling MySQL stored procedure with IN and OUT parameters.

```
mysql> DELIMITER //
mysql> create procedure usp_GetEmployeeName(IN id INT, OUT name VARCHAR(20))
    -> begin
    -> select emp_name into name from employee where emp_id = id;
    -> end//
Query OK, 0 rows affected (0.52 sec)

mysql> DELIMITER ;

mysql> call usp_GetEmployeeName(103, @name);
Query OK, 1 row affected (0.05 sec)


mysql> select @name;
+-------+
| @name |
+-------+
| Jack  |
+-------+
1 row in set (0.00 sec)

# VIEW ALL PROCEDURE
mysql> SHOW PROCEDURE STATUS;

# DROP PROCEDURE
mysql> DROP PROCEDURE usp_totalEmployeeByDeparment;

mysql> DROP PROCEDURE IF EXISTS usp_totalEmployeeByDeparment;
```

TRIGGER STATEMNET

A mysql trigger is a set of SQL statemnets stored in the database.
A mysql trigger is special type of stored procedure.
A mysql trigger is executed or fired whenever an event associated with a table occus example insert, update or delete.
it is special because it is not called manually like procedure it called automatically. whenever associated event are occured.

Syntax:

create trigger [trigger_name]
[before | after]
{insert | update | delete}
on [table_name]
[for each row]
[trigger_body]

Explanation of syntax:
create trigger [trigger_name]: Creates or replaces an existing trigger with the trigger_name.
[before | after]: This specifies when the trigger will be executed.

{insert | update | delete}: This specifies the DML operation.

on [table_name]: This specifies the name of the table associated with the trigger.

[for each row]: This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected.

[trigger_body]: This provides the operation to be performed as trigger is fired.

Example:

SELECT * FROM marks;

id	course_name	score	sid
1	AI In Real World Using Python	14	38
2	AI In Real World Using Python	99	51
3	GIMP Photo Editing	51	29
4	SQLite Tutorial	57	13
5	Swing GUI In Depth	31	34
6	CSS3	83	36
7	Codeigniter	22	39
8	2D Games Using PyGame	60	37
9	Data Structure In Depth	31	32
10	Amazon Cloud AWS	76	12
11	Amazon Cloud AWS	86	27
12	Computer Networks	2	34
13	Swing GUI In Depth	52	1
14	Dynamic Website Development	52	3
15	CakePHP	2	27
16	Natural Language Processing	55	6
17	HTML	68	6
18	Computer Vision Using Python	72	47

```
| 19 | Computer Vision Using Python  |    58 |  24 |
| 20 | Data Structure In Depth       |    70 |  14 |
| 21 | Kali Linux                    |    76 |  18 |
| 22 | Java Complete Tutorial        |    67 |  32 |
| 23 | Java Database Connectivity    |     8 |  25 |
| 24 | The C Ninja                   |    39 |  10 |
| 25 | SQLite Tutorial               |    69 |  60 |
+----+-------------------------------+-------+-----+
25 rows in set (0.00 sec)
```

```
DELIMITER //
CREATE TRIGGER Stud_marks
BEFORE
INSERT ON marks
FOR EACH ROW IF NEW.score < 10 THEN SET NEW.score = 5;
END IF;//
Query OK, 0 rows affected (0.01 sec)
```

```
DELIMITER ;
SELECT * FROM marks;
```

```
+----+-------------------------------+-------+-----+
| id | course_name                   | score | sid |
+----+-------------------------------+-------+-----+
|  1 | AI In Real World Using Python |    14 |  38 |
|  2 | AI In Real World Using Python |    99 |  51 |
|  3 | GIMP Photo Editing            |    51 |  29 |
|  4 | SQLite Tutorial               |    57 |  13 |
|  5 | Swing GUI In Depth            |    31 |  34 |
|  6 | CSS3                          |    83 |  36 |
|  7 | Codeigniter                   |    22 |  39 |
|  8 | 2D Games Using PyGame         |    60 |  37 |
|  9 | Data Structure In Depth       |    31 |  32 |
| 10 | Amazon Cloud AWS              |    76 |  12 |
| 11 | Amazon Cloud AWS              |    86 |  27 |
| 12 | Computer Networks             |     2 |  34 |
| 13 | Swing GUI In Depth            |    52 |   1 |
| 14 | Dynamic Website Development   |    52 |   3 |
| 15 | CakePHP                       |     2 |  27 |
| 16 | Natural Language Processing   |    55 |   6 |
| 17 | HTML                          |    68 |   6 |
| 18 | Computer Vision Using Python  |    72 |  47 |
| 19 | Computer Vision Using Python  |    58 |  24 |
| 20 | Data Structure In Depth       |    70 |  14 |
| 21 | Kali Linux                    |    76 |  18 |
| 22 | Java Complete Tutorial        |    67 |  32 |
| 23 | Java Database Connectivity    |     8 |  25 |
| 24 | The C Ninja                   |    39 |  10 |
| 25 | SQLite Tutorial               |    69 |  60 |
+----+-------------------------------+-------+-----+
25 rows in set (0.00 sec)
```

```
INSERT INTO marks VALUES(26,"HIBERNATE",9,11);
Query OK, 1 row affected (0.01 sec)
```

```
SELECT * FROM marks;
```

```
+----+-------------------------------+-------+-----+
| id | course_name                   | score | sid |
+----+-------------------------------+-------+-----+
```

```
|  1 | AI In Real World Using Python |    14 |  38 |
|  2 | AI In Real World Using Python |    99 |  51 |
|  3 | GIMP Photo Editing            |    51 |  29 |
|  4 | SQLite Tutorial               |    57 |  13 |
|  5 | Swing GUI In Depth            |    31 |  34 |
|  6 | CSS3                          |    83 |  36 |
|  7 | Codeigniter                   |    22 |  39 |
|  8 | 2D Games Using PyGame         |    60 |  37 |
|  9 | Data Structure In Depth       |    31 |  32 |
| 10 | Amazon Cloud AWS              |    76 |  12 |
| 11 | Amazon Cloud AWS              |    86 |  27 |
| 12 | Computer Networks             |     2 |  34 |
| 13 | Swing GUI In Depth            |    52 |   1 |
| 14 | Dynamic Website Development   |    52 |   3 |
| 15 | CakePHP                       |     2 |  27 |
| 16 | Natural Language Processing   |    55 |   6 |
| 17 | HTML                          |    68 |   6 |
| 18 | Computer Vision Using Python  |    72 |  47 |
| 19 | Computer Vision Using Python  |    58 |  24 |
| 20 | Data Structure In Depth       |    70 |  14 |
| 21 | Kali Linux                    |    76 |  18 |
| 22 | Java Complete Tutorial        |    67 |  32 |
| 23 | Java Database Connectivity    |     8 |  25 |
| 24 | The C Ninja                   |    39 |  10 |
| 25 | SQLite Tutorial               |    69 |  60 |
| 26 | HIBERNATE                     |     5 |  11 |
+----+-------------------------------+-------+-----+
26 rows in set (0.00 sec)
```

/*---------- Another One Example ----------*/

SELECT * FROM courses;

```
+----+-------------------------------+-----------------+-------+
| id | course_name                   | instructor_name | fees  |
+----+-------------------------------+-----------------+-------+
|  1 | MySQL Database                | Sandeep Ganguly |  4500 |
|  2 | PHP Development               | Dolly Singh     |  1500 |
|  3 | Java Complete Tutorial        | Ramesh Yadav    |  7500 |
|  4 | Swing GUI In Depth            | Guddu Sharma    | 15000 |
|  5 | Computer Vision Using Python  | Narendra Murthy | 25000 |
|  6 | AI In Real World Using Python | Satya Kundu     | 45000 |
|  7 | 2D Games Using PyGame         | Sandeep Ganguly | 18000 |
|  8 | GIMP Photo Editing            | Rachna Mishra   |  5000 |
|  9 | HTML                          | Chatur Singh    |  1000 |
| 10 | CSS3                          | pinky singh     |  1500 |
| 11 | Amazon Cloud AWS              | Ruchi Singhania | 75000 |
| 12 | Hadoop Big Data               | Ankita Ganguly  | 95000 |
| 13 | Natural Language Processing   | Sandeep Ganguly | 45999 |
| 14 | The C Ninja                   | Pradeep Gurung  |  3599 |
| 15 | Java Database Connectivity    | Ratan Tata      |  6599 |
| 16 | Dynamic Website Development   | Girish Patel    |  8599 |
| 17 | Android App Development       | Rishi Khanna    | 17999 |
| 18 | IOS Developer                 | Umesh Verma     | 25000 |
| 19 | Algorithms In Depth           | Arjun Thapa     |  9999 |
| 20 | Data Structure In Depth       | Ashok Kalia     | 15000 |
| 21 | JQuery Ninja                  | James Guido     |  6500 |
| 22 | Twitter Bootstrap             | Mitali Ghosh    | 14999 |
| 23 | Codeigniter                   | Pawan Kumar     |  7599 |
| 24 | Struts Framework              | Umesh Verma     |  7500 |
| 25 | CakePHP                       | Parvez Khan     | 60000 |
| 26 | Machine Learning              | Faisal Qureshi  | 45000 |
```

```
| 27 | Computer Networks              | Saleem Khan     | 12599 |
| 28 | C++ STL Library Tutorial       | Kareem Sheikh   | 25000 |
| 29 | Kali Linux                     | Jitan Majhi     |  4500 |
| 30 | SQLite Tutorial                | Nitish Kumar    |  6500 |
+----+--------------------------------+-----------------+-------+
30 rows in set (0.00 sec)
```

SELECT * FROM marks;

```
+----+--------------------------------+-------+-----+
| id | course_name                    | score | sid |
+----+--------------------------------+-------+-----+
|  1 | AI In Real World Using Python |    14 |  38 |
|  2 | AI In Real World Using Python |    99 |  51 |
|  3 | GIMP Photo Editing             |    51 |  29 |
|  4 | SQLite Tutorial                |    57 |  13 |
|  5 | Swing GUI In Depth             |    31 |  34 |
|  6 | CSS3                           |    83 |  36 |
|  7 | Codeigniter                    |    22 |  39 |
|  8 | 2D Games Using PyGame          |    60 |  37 |
|  9 | Data Structure In Depth        |    31 |  32 |
| 10 | Amazon Cloud AWS               |    76 |  12 |
| 11 | Amazon Cloud AWS               |    86 |  27 |
| 12 | Computer Networks              |     2 |  34 |
| 13 | Swing GUI In Depth             |    52 |   1 |
| 14 | Dynamic Website Development    |    52 |   3 |
| 15 | CakePHP                        |     2 |  27 |
| 16 | Natural Language Processing    |    55 |   6 |
| 17 | HTML                           |    68 |   6 |
| 18 | Computer Vision Using Python   |    72 |  47 |
| 19 | Computer Vision Using Python   |    58 |  24 |
| 20 | Data Structure In Depth        |    70 |  14 |
| 21 | Kali Linux                     |    76 |  18 |
| 22 | Java Complete Tutorial         |    67 |  32 |
| 23 | Java Database Connectivity     |     8 |  25 |
| 24 | The C Ninja                    |    39 |  10 |
| 25 | SQLite Tutorial                |    69 |  60 |
| 26 | HIBERNATE                      |     5 |  11 |
+----+--------------------------------+-------+-----+
26 rows in set (0.00 sec)
```

```
DILIMITER //
CREATE TRIGGER Stud_Count
AFTER
INSERT
ON marks
FOR EACH ROW
BEGIN
INSERT INTO courses VALUES(31,'Hibernate','Mayur Kadam',1000000);
END //
Query OK, 0 rows affected (0.01 sec)

DELIMITER ;

INSERT INTO marks VALUES(27,"SPRING",9,12);
Query OK, 1 row affected (0.00 sec)
```

SELECT * FROM marks;

id	course_name	score	sid
1	AI In Real World Using Python	14	38
2	AI In Real World Using Python	99	51
3	GIMP Photo Editing	51	29
4	SQLite Tutorial	57	13
5	Swing GUI In Depth	31	34
6	CSS3	83	36
7	Codeigniter	22	39
8	2D Games Using PyGame	60	37
9	Data Structure In Depth	31	32
10	Amazon Cloud AWS	76	12
11	Amazon Cloud AWS	86	27
12	Computer Networks	2	34
13	Swing GUI In Depth	52	1
14	Dynamic Website Development	52	3
15	CakePHP	2	27
16	Natural Language Processing	55	6
17	HTML	68	6
18	Computer Vision Using Python	72	47
19	Computer Vision Using Python	58	24
20	Data Structure In Depth	70	14
21	Kali Linux	76	18
22	Java Complete Tutorial	67	32
23	Java Database Connectivity	8	25
24	The C Ninja	39	10
25	SQLite Tutorial	69	60
26	HIBERNATE	5	11
27	SPRING	5	12

27 rows in set (0.00 sec)

// course table is automatically get updated

SELECT * FROM courses;

id	course_name	instructor_name	fees
1	MySQL Database	Sandeep Ganguly	4500
2	PHP Development	Dolly Singh	1500
3	Java Complete Tutorial	Ramesh Yadav	7500
4	Swing GUI In Depth	Guddu Sharma	15000
5	Computer Vision Using Python	Narendra Murthy	25000
6	AI In Real World Using Python	Satya Kundu	45000
7	2D Games Using PyGame	Sandeep Ganguly	18000
8	GIMP Photo Editing	Rachna Mishra	5000
9	HTML	Chatur Singh	1000
10	CSS3	pinky singh	1500
11	Amazon Cloud AWS	Ruchi Singhania	75000
12	Hadoop Big Data	Ankita Ganguly	95000
13	Natural Language Processing	Sandeep Ganguly	45999
14	The C Ninja	Pradeep Gurung	3599
15	Java Database Connectivity	Ratan Tata	6599
16	Dynamic Website Development	Girish Patel	8599
17	Android App Development	Rishi Khanna	17999
18	IOS Developer	Umesh Verma	25000
19	Algorithms In Depth	Arjun Thapa	9999
20	Data Structure In Depth	Ashok Kalia	15000
21	JQuery Ninja	James Guido	6500

```
| 22 | Twitter Bootstrap             | Mitali Ghosh    |   14999 |
| 23 | Codeigniter                   | Pawan Kumar     |    7599 |
| 24 | Struts Framework              | Umesh Verma     |    7500 |
| 25 | CakePHP                       | Parvez Khan     |   60000 |
| 26 | Machine Learning              | Faisal Qureshi  |   45000 |
| 27 | Computer Networks             | Saleem Khan     |   12599 |
| 28 | C++ STL Library Tutorial      | Kareem Sheikh   |   25000 |
| 29 | Kali Linux                    | Jitan Majhi     |    4500 |
| 30 | SQLite Tutorial               | Nitish Kumar    |    6500 |
| 31 | Hibernate                     | Mayur Kadam     | 1000000 |
+----+-------------------------------+-----------------+---------+
31 rows in set (0.00 sec)
```

list out trigger is system by

SHOW TRIGGERS;

DROP TRIGGER IF EXISTS Stud_Count;
Query OK, 0 rows affected (0.01 sec)