**Databases Notes – mySql**

**In this notes we will learn about all the things of mysql**

**So Lets get started :-**

**Query For Databases**

Question 1) **Query to show database ?**

* Show databases;

**Question 2)Query to create database?**

* CREATE DATABASE <database\_name>

**Question 3) Query to drop database ?**

* DROP DATABASE <database\_name>

**Question 4) Query to Use database ?**

* USE DATABASE <database\_name>

**TABLES AND DATA TYPES**

**Question 1) Query to create a table?**

**🡪create** **table** <table\_name>

(

<column\_name> <column\_DataType>,

<column\_name> <column\_DataType>

)

**Question 2 ) Query to show all tables ?**

**🡪show** tables

**Question 3) Query to Describe tables?**

* DESC <table\_name>

**Question 4)Dropping Tables**

* **drop** <table\_name>

**INSERTING THE DATA INTO TABLES**

**Question 5) Inserting the data into tables**

**🡪insert** **into** dogs(name,age)

**values** (**'roxy'**,10)

**Question 6)Inserting multiple data in single insert call ?(We can insert multiple values in the single insert statement)?**

**🡪insert** **into** dogs (name,age)

**values** (**'bruno'**,10),(**'marsh'**,20),(**'shadow'**,30)

**Question 7) What is not null and null values ?**

* By default when we create the columns their **NULL** properties is automatically set to **yes which means that while inserting the new values we can add the empty field for that column .**
* **On** the other hand when we need that for every entry particular columns needs to be fulfilled then we need to add **not Null clause while defining the columns**

**Below is the query to create columns whose values is not null**

**create** **table** people2(

first\_name **VARCHAR**(128) **not** **null**,

last\_name **VARCHAR**(128) **not** **null**,

age **int** **not** **null**

)

We need to all **not null clause**

**Question 8) What is the Default Value?**

**🡪**Sometimes there are condition arises where we the column values are mandatory , but their may be cases are their where the values for that particular column may would not be their in the dataset due to polluted data , so in that **case we use default values**

**Where the sql automatically fills with the default value in that column if in the insert statements the value for that particular column is not present.**

* Default values may be **standalone values or it could be a functions also .**

**Question 9)Query to insert the default values ?**

**🡪**-- Using of the default values

**create** **table** people3 (

first\_name **varchar**(128) **not** **null** **default** **'unknown'**,

last\_name **varchar**(128) **not** **null** **default** **'unknown'**,

age **int** **not** **null** **default** 0

)

**Question 10) What are the primary key constraint?**

* The primary keys are the one the most important things in the sql tables it created to make each row of data as single **unique entity or indentifier**  which depicts one data .
* **The Primary key should be distinguishable and hence its unique in the table ,**
* **Only single column can be selected as the primary keys .**
* **Primary Keys cannot be null.**

**Question 11) What is unique key constraint ?**

* As the primary keys are the sole indentifier of the data for the table . the unique keys are applied to non primary column where we need **constraint that the column should not have a duplicate values , like the order number and cart number can be unique values , addhar can be values**
* **A table can have multiple unique keys or columns .**

**Question 12) Query to create the Primary Key ?**

**🡪 create** **table** people4 (

people\_id **int** **primary** **key** **auto\_increment**,

first\_name **varchar**(128) **not** **null** **default** **'unknown'**,

last\_name **varchar** (128) **not** **null** **default** **'unknown'**,

age **int** **not** **null** **default** 0

)

**Question 13) What is the AUTO\_INCREMENT Constraints ?**

* When we define the column name with primary key constraints , so with every insert statement we need to add the primary key .
* If the column is just set to create the unique indentity and that column is not present in our data . We can put the constraint to **column with AUTO\_INCREMENT Constraint ,**  what this will do is that it will automatically puts the incremented id for the primary key column .

**CRUD BASICS**

In the below sections we will learn about the crud basics operations and advances to further ladders

**Question 14) Query to select all the entries from the tables ?**

**🡪 select** \* **from** <table\_name>

**Question 15) Query to select only the selected columns only from the table?**

**🡪 select** <column\_name>,<column\_name> **from** <table\_name>.

**We** can also select the specific column from the table rather than selecting the whole all columns

**Question 14) What is WHERE Clause and how to make selection using the where clause ?**

* Where clause works like the filter to filter out the specific data where the condition mets .
  + **select** <column\_name>,<column\_name> **from** <table\_name> **where** <column\_name>=<**any** value>

**Question 15) What are aliases and how to use the aliases ?**

* Aliasing is an exercise where while selecting the data we want to get the data and columns name to changed to some meaningful or required representative names . For that scenario we use the aliasing.
* **Aliasing Just means to provide the temporary pet name to the columns.**
* To alias the column while selecting we use **, the AS keyword.**

**Question 16) Query syntax for the aliasing concepts ?**

-- Aliasing Query syntax

**select** <actual\_column\_name> **as** <aliased\_column\_name> **from** <table\_name>

**Question 17 ) Query syntax for the UPDATE Statements?**

* Update statement is used to update the data pre – present in the table , its also widely used scenario in the databases.
* -- Update Query Syntax
* **update** <table\_name> **set** <column\_name>=<updated\_value> **where** <any\_syntax>

**UPDATE AND SET are two keywords which are required to update the data present in the table.**

**Question 18 ) Query Syntax for the Delete statements ?**

* Delete statements is used to delete the rows from the tables and its important to use **where conditions if we want to delete only certain amount of rows.**

**Below are the syntax to use the delete query**

-- delete query syntax to delete particular rows

**delete** **from** <table\_name> **where** <column\_name>=<some\_value>

-- query syntax to delete all rows (never use this )

**delete** **from** <table\_name>

**Refining Selections - TBD**

**Question 19) Distinct Keyword in sql ?**

* The distinct keyword is used to get the unique items from the columns .

Below is the syntax for the distinct keyword.

-- syntax for the distinct key

**select** **distinct** <column\_name> **from** <table\_name>;

**MOST MOST IMPORTANT ( JOINS and Relationships IN SQL )**

**Question 1) What are the relationships in sql ?**

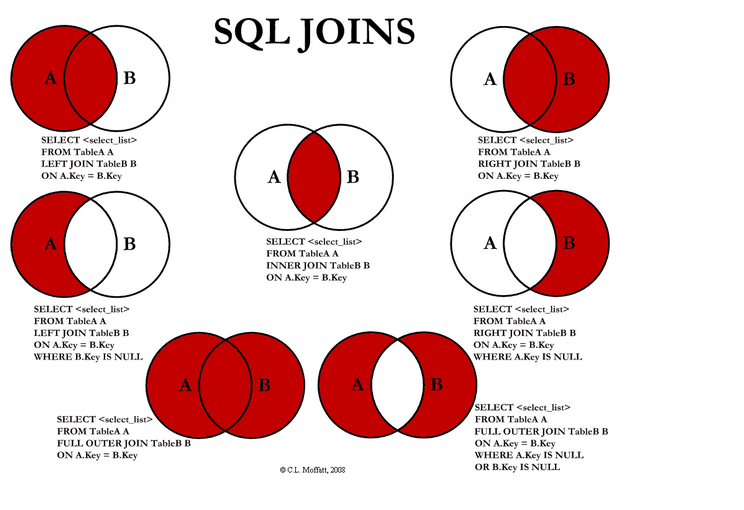
* The relationships are essential part of any sql as In sql we cant create everything in the single table , actually it defeats the rule of the single responsibility principle. And may even may the querying the database very heavy and latency may be increased.
* So we create the relationships between the tables . so each table stores the unit sensible data

**Question 2) What are the kinds of the relationships in sql ?**

* **We are the types of relationships between the tables :-**
  + **One to One Relationship –** This type of relationship exist one table has only one entry in other tables like **data customer relation with aadhar card data ( not have wide use cases)**
  + **One to many Relationship -**  This type of relationship exits one table has many entries in the second table means **one entity could have many entities in other tables (Most Use cases Occurs in this type of the relationships)**
    - **Like :-** orders data , cart data , purchase history , transactions data and many more usecases
  + **Many to many Relationship –** This type of the relationships where many enitities have many relations into other tables (**Second mose usecases occurs in this type of relationships)**
    - **Like one authors have many books and one books can have many authors (So In relation to both tables have many to many relations)**

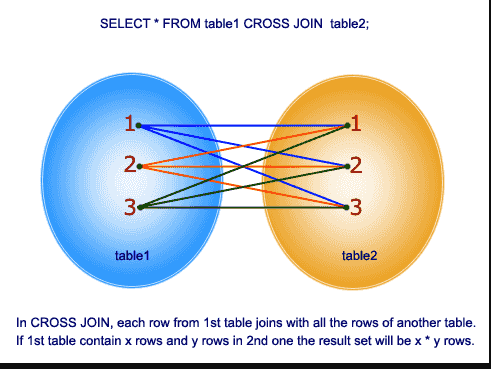
**Question 3) What are the joins ?**

* Now as we have learned that there are relations between the tables , so get the relavent and connected data from the tables in the single shot .
* We have joins which helps to access the data in the single shot .
* Below are the joins we will learn in this topic
  + **Cross Join**
  + **Left Join**
  + **Right Join**
  + **OuterJoin**

****

**Question4) What is cross join ?**

* The cross join is the join where it returns the cartesian products of the rows from the tables .
* Means each rows from table 1 will be joined with the each rows of the table 2
* Its is very less used join in the sql because , it just did not provide us with some analytical insights of joins.

****

**Question 6) Syntax of the Cross Joins ?**

**🡪** -- CrossJoins query syntax

**select** \* **from** customers,orders;

**Question 7) What is the Inner joins ?**

* The inner join is the join where we get the intersection of the two tables , which means the common values which are present in both the tables .
* The inner joins are essential to get the data where we need only those data entitites which are present in both tables .
* For **example provide the data for only those customers which are have orders in the orders tables .**

**Question 8) Provide the Query Syntax for the Inner Join ? (Very Very Important )**

**🡪** -- Inner Joins Query Syntax

-- Inner Join query to get all the customers and orders tables data

**select** \* **from** customers

**inner** **join** orders **on** orders.customer\_id = customers.customer\_id

-- Inner join query syntax to get specified customers and orders tables data

**select** first\_name , last\_name ,email,order\_date,amount **from** customers

**inner** **join** orders **on** orders.customer\_id = customers.customer\_id

Main Key Syntax Points we need to learn

* JOIN – This keyword is required to tell which type of the join we need to make on which tables
* ON – this keyword is to determine on which columns is used as based for the join note columns should be present in both the columns.

**Questions9 ) Provide the query syntax of the inner Join with the Group by syntax?**

* Consider a scenario where we need to get all the total order amount of only those customers from customer table which have maded the order.
* So in these case we can **GROUP BY Clause.**
* -- Question I want total order amount of the customers
* **select** first\_name , last\_name , **SUM**(orders.amount) **as** *order\_total* **from** customers
* **inner** **join** orders **on** orders.customer\_id = customers.customer\_id
* **group** **by** first\_name ,last\_name
* **order** **by** *order\_total* **desc** ;

**Understand the query**

* **We are taking all the required the parameters from both customers and orders tables selective search**
* **We maded the inner join because we wanted only the intersection values between the two tables**
* **Then maded grouping using the firstName and lastname by which we can use the Aggregate sum function**
* **Then we maded sorted in the ascending order.**