Q.1 Create a **database** named google **and** table in it named products.

* **create database google;**

**use google;**

**create table products** (

empid int,

name varchar (50),

age int,

country varchar (50),

salary int,

dateofjoining date,

phone bigint

);

**ALTER TABLE products**

**CHANGE COLUMN phone phone bigint;**

# changed datatype from ‘int’ to ‘bigint’

**USE google;**

**SELECT** \* FROM products;

**insert** into products (empid, name, age, country, salary, dateofjoining, phone)

values **( 1, "reshma", 28, "india", 25000, '2023-01-09', 9478349283),**

**( 2, "Ana", 23, "Australia", 30000, '2023-05-01', 9478546283),**

**( 3, "Kenan", 25, "USA", 27000,'2023-02-10', 9778349283),**

**( 4, "Daniel", 30, "Germany", 20000,'2023-05-02', 9878349283),**

**( 5, "Aditya", 23, "india", 29000, '2023-05-06', 8783492833),**

**( 6, "kavya", 24, "USA", 28000, '2024-01-10', 9978349283),**

**( 7, "Mark", 28, "USA", 22000, '2023-01-09', 9978349283);**

Q2. Write a SQL **statement to display specific columns** such as names and salary for all Employees

select name,salary from products;

* **select name,salary from products;**

Q3. Write a SQL query to locate an **employee who lives in ‘India'**. Return employee’s name and country.

* **select name, country**

**from products**

**where country = 'india';**

Q4 . Write a SQL statement to return the name of an **employee whose salary is more than 25000**.

* **select name, salary**

**from products**

**where salary >=25000;**

Q5.Write a SQL statement to return the name of an employee **whose salary is  between 25000 and 30000.**

* **select name, salary**

**from products**

**where salary between 25000 and 30000;**

Q6. Write a SQL query to find the employee **whose ages are higher than or equal to 30**. Order the **result by age in descending.**

* **select name, age**

**from products**

**where age >=25**

**ORDER BY age DESC;**

Q7. Write a query to display the **name of employees in the order of their joining date**.

* **select name, dateofjoining**

**from products**

**order by dateofjoining desc;**

Q8. Write a query to **count the number of employees**.

* **select count(\*) from products;**

Q9.Write a query to display the **name and salary of an employee who either lives in India or Australia.**

* **select \* from products**

**where country in ('india', 'Australia');**

Q10.From the following table, write a SQL query to **display the name as 'NameOfEmployee'** .

* **alter table products**

**rename column name to NameOfEmployee;**

Q11**.** Write a query to delete **the records of Ana.**

* **DELETE from products**

**WHERE nameofemployee = 'reshma';**

Q12. Write a sql query to find the **lowest salary of an employee.**

* **select min(salary)**

**from products;**

Q13. Write a sql query to find the **highest salary of an employee.**

* **select max(salary)**

**from products;**

Q 14. Write a sql query to display those **names of employees**

**which ends with an ‘a’.**

* **select nameofemployee from products**

**where nameofemployee like '%a';**

Q15. Write a SQL query to find the **number of employees with names living in each country**.

* **select count(\*) country, name from customer**

**group by country, name;**

Q16. Write a query to find the **highest salary of employees in each country having salary more than 70000.**

* **select name, salary, country from customer**

**where salary >70000**

**group by name, salary,country;**

Q17.Write a query to **add an email column in the existing employee table**.

* **alter table customer**

**add column email varchar (50);**

alter table customer

**modify** column email varchar (255); # changed varchar **(50) to (255)**

Q18. Write a query to  **update the age of an employee to 55 and salary to 90000 where id is 4.**

* **update customer**

# You use **UPDATE** when you want to change the values of ‘**EXISTING RECORDS’**  in a table based on certain conditions.

**set age = 55, salary = 90000**

**where Empid = 4;**

Q19. Write a query to **rename a column country to address.**

* **alter table customer**

**rename column country to address;**

Q20. Write a query to **delete the age column**.

* **alter table** **customer**

**#** You use **ALTER** when you want to add, modify, or delete columns, constraints, or other structural elements of a table.

**drop column age;**

Q21.Write a query to **extract only the first three records from the given table**.

* **select** \* **from** **customer**

**limit** **3**;

Q22. Write a sql query to **return all of the records from customer table but only matching records from salesperson table using joins.**

* **select \* from customers**

**inner join salesman**

**on customers.id = salesman.id;**

Q 23. Write a SQL query to **find the salesperson and customer who reside in the same city**.

* **SELECT c.custId AS customer\_id, c.cname AS customer\_name, s.id AS salesman\_id, s.name AS salesman\_name, c.city**

**FROM customers c**

**INNER JOIN salesman s ON c.city = s.city;**

Q 24. Write a SQL query to **find sales people who received commissions of more than 11 from the company. Return cname, customer city, name of salesman, commission.**

* **SELECT c.cname AS customer\_name, c.city AS customer\_city, s.name AS salesman\_name, s.commission**

**FROM customers c**

**INNER JOIN salesman s ON c.city = s.city**

**WHERE s.commission > 11;**

Q25.Create a **table named person with fields id, name, city, age, email, country using the following constraints.**

i) The field id should be a primary key

ii) Name field cannot be null

iii) city field cannot be null and for default value use “Delhi”

iv) Age should be greater than 18

v) email should be unique and cannot be null

vi) country field cannot be null and for default use “India”

* **create table person (**

**fieldid int primary key,**

**name varchar (50) not null,**

**city varchar (50) default 'Nashik',**

**age int check (age>=18),**

**email varchar (50) not null unique,**

**country varchar (50) default "india"**

**);**

**select \* from person;**

**insert into person (fieldid, name, city, age, email, country)**

**value (1, "Sanvi", "Pune", 22, "sanvi@gamil.com.com", "india"),**

**(2, "Akash", "Mumbai", 23, "aksah@gamil.com", "USA"),**

**(3, "Tanya", "Delhi", 25, "tanya@gamil.com", "Australia"),**

**(4, "Kenan", "NY", 23, "kenan@gamil.com", "Turkey"),**

**(5, "Kanya", "Pune", 24, "kanya@gamil.com", "germany");**

Q. 26 Create a **stored procedure** named **“practice”** and store the query that you used above to create a table “person”.

* **CREATE DEFINER=`root`@`localhost` PROCEDURE `practice`()**

**BEGIN**

**create table SN (**

**fieldid int primary key,**

**name varchar (50) not null,**

**city varchar (50) default 'Nashik',**

**age int check (age>=18),**

**email varchar (50) not null unique,**

**country varchar (50) default "india"**

**);**

**select \* from SN;**

**insert into person (fieldid, name, city, age, email, country)**

**value (1, "Sanvi", "Pune", 22, "sanvi@gamil.com.com", "india"),**

**(2, "Akash", "Mumbai", 23, "aksah@gamil.com", "USA"),**

**(3, "Tanya", "Delhi", 25, "tanya@gamil.com", "Australia"),**

**(4, "Kenan", "NY", 23, "kenan@gamil.com", "Turkey"),**

**(5, "Kanya", "Pune", 24, "kanya@gamil.com", "germany");**

**END**

**Q27.** Find the **highest salary** of an employee

* **select max(salary)**

**from products;**

Q28**.** Find the **third highest salary** of an employee

* **select distinct salary from qspace**

**order by salary desc**

# skips the first two salaries using offset 2, and then selects the third-highest salary using limit 1.

**limit 1 offset 2;**

**Q29.** Find the **fourth minimum** salary of an employee

* **select distinct salary from qspace**

**order by salary asc**

**limit 1 offset 3;**

Q30.Write a query to display **the  name of the employee who works in the shipping department**

* **select empid**

**from pune**

**where deptname = 'marketing';**

Q31. Write a query to display the names **of the employees who are not in the shipping department.**

* **select empid**

**from pune #** not equal (**<>**)

**where deptname <> 'shipping';**