**Create a table Employee\_Info with following attributes EmployeeID, EmployeeName, Emergency\_ContactName, PhoneNumber, Address, City, Country;**

CREATE TABLE Employee\_Info

(

EmployeeID int,

EmployeeName varchar(255),

Emergency\_ContactName varchar(255),

PhoneNumber int,

Address varchar(255),

City varchar(255),

Country varchar(255)

);

**Add a column BloodGroup**

ALTER TABLE Employee\_Info

ADD BloodGroup varchar(255);

**Drop column BloodGroup**

ALTER TABLE Employee\_Info

DROP COLUMN BloodGroup ;

**Add a column DOB and modify its datatype from year to date**

ALTER TABLE Employee\_Info

ADD DOB year;

ALTER TABLE Employee\_Info modify DOB date;

DROP Table Employee\_Info;

**Create a table Employee\_Info with following attributes EmployeeID, EmployeeName, Emergency\_ContactName, PhoneNumber, Address, City, Country, with primary key as Employee\_Info; EmployeeID should NOT NULL UNIQUE with EmployeeID >1, country should have default value India. PhoneNumber should be int**

CREATE TABLE Employee\_Info

(

EmployeeID int NOT NULL UNIQUE CHECK (EmployeeID >1),

EmployeeName varchar(255),

Emergency\_ContactName varchar(255),

PhoneNumber int,

Address varchar(255),

City varchar(255),

Country varchar(255) DEFAULT 'India'

);

**Change PhoneNumber to varchar**

ALTER TABLE Employee\_Info MODIFY PhoneNumber VARCHAR(255);

**Insert data into employee\_info**

INSERT INTO Employee\_Info(EmployeeID, EmployeeName, Emergency\_ContactName, PhoneNumber, Address, City, Country)

VALUES (06, 'Sanjana','Jagannath', '9921321141', 'Camel Street House No 12', 'Chennai', 'India');

INSERT INTO Employee\_Info

VALUES ('07', 'Sayantini','Praveen', '9934567654', 'Nice Road 21', 'Pune', 'India');

INSERT INTO Employee\_Info(EmployeeID, EmployeeName, Emergency\_ContactName, PhoneNumber, Address, City, Country)

VALUES (01, 'Preeti','Sayantini', '9921328141', 'PWD Street House No 12', 'Mumbai', 'India');

**Update employee name and city to Ahmedabad where employeeId is 1**

UPDATE Employee\_Info

SET EmployeeName = 'Aahana', City= 'Ahmedabad'

WHERE EmployeeID = 1;

**Delete record with employee id 1**

DELETE FROM Employee\_Info WHERE EmployeeID='1';

INSERT INTO Employee\_Info(EmployeeID, EmployeeName, Emergency\_ContactName, PhoneNumber, Address, City, Country)

VALUES (01, 'Preeti','Sayantini', '9921328141', 'PWD Street House No 12', 'Mumbai', 'India');

INSERT INTO Employee\_Info(EmployeeID, EmployeeName, Emergency\_ContactName, PhoneNumber, Address, City, Country)

VALUES (02, 'Seema','Sana', '9821328141', 'ABC Street House No 12', 'Nagpur', 'India');

INSERT INTO Employee\_Info(EmployeeID, EmployeeName, Emergency\_ContactName, PhoneNumber, Address, City, Country)

VALUES (03, 'Rahul','Som', '9981328141', 'DEF Street House No 12', 'Kanpur', 'India');

INSERT INTO Employee\_Info(EmployeeID, EmployeeName, Emergency\_ContactName, PhoneNumber, Address, City, Country)

VALUES (04, 'Ayush','Mohit', '9821928141', 'GHI Street House No 12', 'Kolkata', 'India');

INSERT INTO Employee\_Info(EmployeeID, EmployeeName, Emergency\_ContactName, PhoneNumber, Address, City, Country)

VALUES (05, 'Sam','Ram', '9821928101', 'JKL Street House No 12', 'Bangalore', 'India');

INSERT INTO Employee\_Info(EmployeeID, EmployeeName, Emergency\_ContactName, PhoneNumber, Address, City, Country)

VALUES (08, 'Pam','Ram', '9821928181', 'MNO Street House No 12', 'Calicut', 'India');

INSERT INTO Employee\_Info(EmployeeID, EmployeeName, Emergency\_ContactName, PhoneNumber, Address, City, Country)

VALUES (09, 'Shalu','Ram', '9021928101', 'PQR Street House No 12', 'Bangalore', 'India');

INSERT INTO Employee\_Info(EmployeeID, EmployeeName, Emergency\_ContactName, PhoneNumber, Address, City, Country)

VALUES (10, 'Krishna','Ram', '9021928901', 'PQR Street House No 12', 'Chennai', 'India');

CREATE TABLE Employee\_Salary

(

EmployeeID int NOT NULL UNIQUE CHECK (EmployeeID >1),

Employee\_Salary int,

PRIMARY KEY(EmployeeID),

FOREIGN KEY (EmployeeID) REFERENCES Employee\_Info(EmployeeID)

);

INSERT INTO Employee\_Salary(EmployeeID, Employee\_Salary)

VALUES (10, 45000);

INSERT INTO Employee\_Salary(EmployeeID, Employee\_Salary)

VALUES (08, 47000);

INSERT INTO Employee\_Salary(EmployeeID, Employee\_Salary)

VALUES (09, 37000);

ALTER TABLE Employee\_Salary CHANGE COLUMN Employee\_Salary Salary int;

CREATE TABLE Technologies

(

TechID int,

EmployeeID int NOT NULL UNIQUE CHECK (EmployeeID >1),

TechName varchar(255),

ProjectStartDate Date,

PRIMARY KEY(TechID),

FOREIGN KEY (EmployeeID) REFERENCES Employee\_Info(EmployeeID)

);

ALTER TABLE Technologies MODIFY ProjectStartDate varchar(255);

INSERT INTO Technologies(TechID, EmployeeID, TechName, ProjectStartDate)

VALUES (1, 10, 'DevOps','04-01-2019');

INSERT INTO Technologies(TechID, EmployeeID, TechName, ProjectStartDate)

VALUES (2, 8, 'Blockchain','06-07-2019');

INSERT INTO Technologies(TechID, EmployeeID, TechName, ProjectStartDate)

VALUES (3, 4, 'Python','01-03-2019');

**Select EmployeeId and EmployeeName from table Employee\_Info**

SELECT EmployeeID, EmployeeName, City FROM Employee\_Info;

**Select unique PhoneNumber from Employee\_Info**

SELECT DISTINCT PhoneNumber FROM Employee\_Info;

**Fetch all records ordered by Emergency\_ContactName asc, EmployeeName desc**

SELECT \* FROM Employee\_Info ORDER BY Emergency\_ContactName ASC, EmployeeName DESC;

**Fetch count of employees belonging to same cities and their City**

SELECT COUNT(EmployeeID), City FROM Employee\_Info GROUP BY City;

**Fetch count of employees belonging to same cities and their City , where count of employees is greater than 2, orderd by count**

SELECT COUNT(EmployeeID), City FROM Employee\_Info GROUP BY City HAVING COUNT(EmployeeID) = 2 ORDER BY COUNT(EmployeeID) DESC;

*SELECT COUNT(EmployeeID), City FROM Employee\_Info GROUP BY City HAVING EmployeeID > 2 ORDER BY COUNT(EmployeeID) DESC; -* ***wont work***

**Select Employees belonging to Mumbai or Hyderabad**

SELECT \* FROM Employee\_Info WHERE City='Mumbai' AND City='Hyderabad';

SELECT \* FROM Employee\_Info WHERE City='Mumbai' OR City='Hyderabad';

**Select Employees not belonging to Mumbai**

SELECT \* FROM Employee\_Info WHERE NOT City='Mumbai';

**Select all records from Employee\_Salary where Salary BETWEEN 40000 AND 50000**

SELECT \* FROM Employee\_Salary WHERE Salary BETWEEN 40000 AND 50000;

**Select EmployeeName where name starts with S**

SELECT \* FROM Employee\_Info WHERE EmployeeName LIKE 'S%';

**Select EmployeeName who live in any of these cities Mumbai', 'Bangalore', 'Hyderabad'**

SELECT \* FROM Employee\_Info WHERE City IN ('Mumbai', 'Bangalore', 'Hyderabad');

**Select EmployeeNames from Employee\_Info whose EmployeeID is greater than all the EmployeeID of people from Bangalore**

SELECT EmployeeName

FROM Employee\_Info

WHERE EmployeeID > ALL (SELECT EmployeeID FROM Employee\_Info WHERE City = 'Bangalore');

**Select EmployeeNames from Employee\_Info whose EmployeeID is greater than any the EmployeeID of people from Bangalore**

SELECT EmployeeName

FROM Employee\_Info

WHERE EmployeeID > ANY (SELECT EmployeeID FROM Employee\_Info WHERE City = 'Bangalore');

**Select EmployeeNames from Employee\_Info only if any employee lives in Hyderabad or return empty**

SELECT EmployeeName

FROM Employee\_Info

WHERE EXISTS (SELECT EmployeeID FROM Employee\_Info WHERE City = 'Hyderabad');

**Select EmployeeNames from Employee\_Info only if any employee lives in Chennai or return empty**

SELECT EmployeeName

FROM Employee\_Info

WHERE EXISTS (SELECT EmployeeID FROM Employee\_Info WHERE City = 'Chennai');

**Fetch maximum Salary as LargestFees from Employee\_Salary**

SELECT MAX(Salary) AS LargestFees

FROM Employee\_Salary;

**Fetch average Salary as AvgFees from Employee\_Salary**

SELECT AVG(Salary) AS AvgFees

FROM Employee\_Salary;

**Fetch sum of Salary as SumFees from Employee\_Salary**

SELECT SUM(Salary) AS SumFees

FROM Employee\_Salary;

**For each city , select employee having highes EmployeeId value, with the result sorted by Corresponding highest id**

Select Max(EmployeeId) as MaxId, city as MaxSalary from Employee\_Info group by city order by Max(EmployeeId) desc;

**Fetch EmployeeId, EmployeeName of Employees who salaries are greater than 42000 using subqueries**

Select EmployeeId, EmployeeName from Employee\_Info where EmployeeId in (Select EmployeeId from Employee\_Salary where Salary > 42000)

**Joins on EmployeeId**

SELECT \*

FROM Technologies t

INNER JOIN Employee\_Info e ON t.EmployeeID = e.EmployeeID;

SELECT \*

FROM Technologies t

LEFT JOIN Employee\_Info e ON t.EmployeeID = e.EmployeeID;

SELECT \*

FROM Technologies t

RIGHT JOIN Employee\_Info e ON t.EmployeeID = e.EmployeeID;

**Full Join**

SELECT \*

FROM Technologies t

LEFT JOIN Employee\_Info e ON t.EmployeeID = e.EmployeeID UNION

SELECT \*

FROM Technologies t

RIGHT JOIN Employee\_Info e ON t.EmployeeID = e.EmployeeID;

**Create a View containing only employees from Chennai**

CREATE OR REPLACE VIEW Chennai\_Employees AS

SELECT \*

FROM Employee\_Info

WHERE City = "Chennai";

Select \* from Chennai\_Employees;

**Create a Procedure which takes the city and gives the count of employees from the city;**

DELIMITER $$

DROP PROCEDURE IF EXISTS CityEmployeeCount$$

CREATE PROCEDURE CityEmployeeCount(city VARCHAR(255), OUT total INT)

BEGIN

SELECT COUNT(EmployeeID) into total from Employee\_Info group by city;

END$$

DELIMITER ;

CALL CityEmployeeCount('Chennai',@total);

SELECT @total;

**Return all cities with more customers than the average number of customers of all cities. For each such city, return the country name, the city name and the number of customers. Order the result by country name ascending**

SELECT country1.country\_name, country2.city\_name, count(\*) as counts from customer country3 inner join city country 2 on country2.id =country3.city\_id inner join country country1 on country1.id = country2.country\_id group by country3.city\_id having count > (select count(customer.id)/count(distinct customer.city\_id) from customer) order by country1.country\_name;