create database employee;

CREATE TABLE `employee`.`data\_science\_team` (

`EMP\_ID` VARCHAR(45) NOT NULL,

`FIRST\_NAME` VARCHAR(45) NOT NULL,

`LAST\_NAME` VARCHAR(45) NOT NULL,

`GENDER` VARCHAR(45) NOT NULL,

`ROLE` VARCHAR(45) NOT NULL,

`DEPT` VARCHAR(45) NOT NULL,

`EXP` VARCHAR(45) NOT NULL,

`COUNTRY` VARCHAR(45) NOT NULL,

`CONTINENT` VARCHAR(45) NOT NULL,

PRIMARY KEY (`DEPT`));

CREATE TABLE `employee`.`proj\_table` (

`PROJ\_ID` VARCHAR(45) NOT NULL,

`PROJ\_NAME` VARCHAR(45) NOT NULL,

`DOMAIN` VARCHAR(45) NOT NULL,

`START\_DATE` DATE NOT NULL,

`CLOSURE\_DATE` DATE NOT NULL,

`DEV\_QTR` VARCHAR(45) NOT NULL,

`STATUS` VARCHAR(45) NOT NULL,

PRIMARY KEY (`PROJ\_ID`));

CREATE TABLE `employee`.`emp\_record\_table` (

`EMP\_ID` VARCHAR(45) NOT NULL,

`FIRST\_NAME` VARCHAR(45) NOT NULL,

`LAST\_NAME` VARCHAR(45) NOT NULL,

`GENDER` CHAR(1) NOT NULL,

`ROLE` VARCHAR(45) NOT NULL,

`DEPT` VARCHAR(45) NOT NULL,

`EXP` VARCHAR(45) NOT NULL,

`COUNTRY` VARCHAR(45) NOT NULL,

`CONTINENT` VARCHAR(45) NOT NULL,

`SALARY` INT NOT NULL,

`EMP\_RATING` INT NOT NULL,

`MANAGER\_ID` VARCHAR(45) NOT NULL,

`PROJ\_ID` VARCHAR(45) NOT NULL,

PRIMARY KEY (`EMP\_ID`),

INDEX `PROJ\_ID\_IDX` (`PROJ\_ID` ASC) INVISIBLE,

INDEX `DEPT\_IDX` (`DEPT` ASC) VISIBLE,

CONSTRAINT `PROJ\_ID`

FOREIGN KEY (`PROJ\_ID`)

REFERENCES `employee`.`proj\_table` (`PROJ\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `DEPT`

FOREIGN KEY (`DEPT`)

REFERENCES `employee`.`data\_science\_team` (`DEPT`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

use employee;

select \* from emp\_record\_table

# 1) Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department

select ï»¿EMP\_ID, FIRST\_NAME,LAST\_NAME,GENDER,DEPT FROM emp\_record\_table order by ï»¿EMP\_ID asc;

alter table emp\_record\_table change ï»¿EMP\_ID EMP\_ID varchar(45);

select EMP\_ID, FIRST\_NAME,LAST\_NAME,GENDER,DEPT FROM emp\_record\_table order by EMP\_ID asc;

# Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPARTMENT, and EMP\_RATING if the EMP\_RATING is less than 2:

select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING FROM emp\_record\_table where EMP\_RATING < 2;

# Write a query to fetch EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPARTMENT, and EMP\_RATING if the EMP\_RATING is greater than 2:

select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING FROM emp\_record\_table where EMP\_RATING > 2;

#Between 2 & 4

select EMP\_ID, FIRST\_NAME, LAST\_NAME, GENDER, DEPT, EMP\_RATING FROM emp\_record\_table where EMP\_RATING > 2 and EMP\_RATING < 4;

# Write a query to concatenate the FIRST\_NAME and the LAST\_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME

select concat(FIRST\_NAME, " ", LAST\_NAME) as NAME FROM emp\_record\_table where DEPT = 'FINANCE';

#4) Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President)

alter table emp\_record\_table change `MANAGER ID` MANAGER\_ID varchar(45);

SELECT MANAGER\_ID, count(EMP\_ID)

FROM emp\_record\_table

WHERE MANAGER\_ID IS NOT NULL

GROUP BY MANAGER\_ID ORDER BY MANAGER\_ID ASC;

#5) Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table

SELECT \* FROM emp\_record\_table WHERE DEPT = 'HEALTHCARE'

UNION

SELECT \* FROM emp\_record\_table WHERE DEPT = 'FINANCE';

#6) Write a query to list down employee details such as EMP\_ID, FIRST\_NAME, LAST\_NAME, ROLE, DEPARTMENT, and EMP\_RATING grouped by dept

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, `ROLE`, DEPT, EMP\_RATING, AVG(EMP\_RATING)

FROM emp\_record\_table GROUP BY DEPT;

SELECT MAX(EMP\_ID) as EMP\_ID, MAX(FIRST\_NAME) as FIRST\_NAME, MAX(LAST\_NAME) as LAST\_NAME,

MAX(`ROLE`) as `ROLE`, DEPT, AVG(EMP\_RATING)

FROM emp\_record\_table

GROUP BY DEPT

LIMIT 0, 1000;

#7) Write a query to calculate the minimum and the maximum salary of the employees in each role

SELECT role, min(EMP\_RATING), max(EMP\_RATING) FROM emp\_record\_table GROUP BY `ROLE`;

#8) Write a query to assign ranks to each employee based on their experience

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, EXP, rank() OVER (ORDER BY exp DESC) AS 'Rank'

FROM emp\_record\_table;

#9) Write a query to create a view that displays employees in various countries whose salary is more than six thousand

CREATE VIEW Test AS SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, COUNTRY, SALARY

FROM emp\_record\_table WHERE SALARY > 6000;

#10) Write a nested query to find employees with experience of more than ten years

SELECT \* FROM Test;

SELECT \* FROM (SELECT \* FROM emp\_record\_table WHERE exp>10) AS tab;

#Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years

DELIMITER //

CREATE PROCEDURE 3PlusExp()

BEGIN

SELECT \* FROM emp\_record\_table WHERE EXP>3;

END //

delimiter ;

Call 3PlusExp();

#Write a query using stored functions in the project table to check whether the job profile assigned to each employee in the data science team matches the organization’s set standard

delimiter $$

CREATE FUNCTION check\_job\_role(EXP integer)

RETURNS VARCHAR(40)

DETERMINISTIC

BEGIN

DECLARE chck VARCHAR(40);

if EXP < 2 THEN SET chck = "JUNIOR DATA SCIENTIST";

elseif EXP >=2 AND EXP < 5 THEN SET chck = "ASSOCIATE DATA SCIENTIST";

elseif EXP >=5 AND EXP < 10 THEN SET chck = "SENIOR DATA SCIENTIST";

elseif EXP >= 10 AND EXP < 12 THEN SET chck = "LEAD DATA SCIENTIST";

elseif EXP >= 12 THEN SET chck = "MANAGER";

end if; RETURN (chck);

END $$

delimiter ;

#Checking Data Science TEAM

alter table data\_science\_team change `ï»¿EMP\_ID` EMP\_ID varchar(45);

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, `ROLE`, check\_job\_role(EXP)

FROM data\_science\_team WHERE `ROLE` != check\_job\_role(EXP);

#13) Create an index to improve the cost and performance of the query to find the employee whose FIRST\_NAME is ‘Eric’ in the employee table after checking the execution plan

CREATE INDEX FirstName ON emp\_record\_table (FIRST\_NAME(10));

#14) Write a query to calculate the bonus for all the employees, based on their ratings and salaries (Use the formula: 5% of salary \* employee rating)

SELECT EMP\_ID, FIRST\_NAME, LAST\_NAME, SALARY, EMP\_RATING, (0.05 \* SALARY \* EMP\_RATING) AS comm

FROM emp\_record\_table

LIMIT 0, 1000

#15) Write a query to calculate the average salary distribution based on the continent and country

SELECT COUNTRY, AVG(SALARY) FROM emp\_record\_table GROUP BY COUNTRY;

SELECT continent, AVG(salary) FROM emp\_record\_table GROUP BY continent;