```
-- **MySQL Workbench CODE by Abhilash Mishra** --

-- Creating database employee
CREATE DATABASE employee;
USE employee;

-- Creating project table
CREATE TABLE project_table(
project_id VARCHAR(8),
project_name VARCHAR(50),
domain VARCHAR(30),
start_date DATE,
closure_date DATE,
dev_qtr VARCHAR(3),
status_ VARCHAR(20),
PRIMARY KEY(project_id));

-- Creating employee table
CREATE TABLE emp_table(
emp_id VARCHAR(5),
first_name VARCHAR(20),
last_name VARCHAR(20),
gender VARCHAR(1),
role_ VARCHAR(50),
dept VARCHAR(30),
exp INT,
country VARCHAR(40),
continent VARCHAR(40),
salary INT,
emp_rating INT,
manager_id VARCHAR(5),
project_id VARCHAR(8),
PRIMARY KEY (emp_id),
FOREIGN KEY (project_id) REFERENCES project_table(project_id));

-- Creatin data science team table
CREATE TABLE data_science_team_table(
emp_id VARCHAR(5),
first_name VARCHAR(20),
last_name VARCHAR(20),
gender VARCHAR(1),
role_ VARCHAR(50),
dept VARCHAR(30),
exp INT,
country VARCHAR(40),
continent VARCHAR(40),
FOREIGN KEY (emp_id) REFERENCES emp_table(emp_id));


-- Importing values in project table
INSERT INTO project_table VALUES
('P103','Drug Discovery','Healthcare','2021-04-06','2021-06-
20','Q1','Done'),
```

```
('P105','Fraud Detection','Finance','2021-04-11','2021-06-
25','Q1','Done'),
('P109','Market Basket Analysis','Retail','2021-04-12','2021-06-
30','Q1','Delayed'),
('P204','Supply Chain Management','Automative','2021-07-15','2021-09-
28','Q2','WIP'),
('P302','Early Detection of Lung Cancer','Healthcare','2021-10-08','2021-
12-18','Q3','YTS'),
('P406','Customer Sentiment Analysis','Retail','2021-07-09','2021-09-
24','Q2','WIP');

-- Importing values in the employee details table
INSERT INTO emp_table VALUES
('E001','Arthur','Black','M','President','All',20,'USA','North
America',16500,5,NULL,NULL),
('E005','Eric','Hoffman','M','Lead Data
Scientist','Finance',11,'USA','North America',8500,3,'E103','P105'),
('E010','William','Butler','M','Lead Data
Scientist','Automative',12,'France','Europe',9000,2,'E428','P204'),
('E052','Dianna','Wilson','F','Senior Data
Scientist','Healthcare',6,'Canada','North America',5500,5,'E083','P103'),
('E057','Dorothy','Wilson','F','Senior Data
Scientist','Healthcare',9,'USA','North America',7700,1,'E083','P302'),
('E083','Patrick','Voltz','M','Manager','Healthcare',15,'USA','North
America',9500,5,'E001',NULL),
('E103','Emily','Grove','F','Manager','Finance',14,'Canada','North
America',10500,4,'E001',NULL),
('E204','Karene','Nowak','F','Senior Data
Scientist','Automative',8,'Germany','Europe',7500,5,'E428','P204'),
('E245','Nian','Zhen','M','Senior Data
Scientist','Retail',6,'China','Asia',6500,2,'E583','P109'),
('E260','Roy','Collins','M','Senior Data
Scientist','Retail',7,'India','Asia',7000,3,'E583',NULL),
('E403','Steve','Hoffman','M','Associate Data
Scientist','Finance',4,'USA','North America',5000,3,'E103','P105'),
('E428','Pete','Allen','M','Manager','Automative',14,'Germany','Europe',1
1000,4,'E001',NULL),
('E478','David','Smith','M','Associate Data
Scientist','Retail',3,'Colombia','South America',4000,4,'E583','P109'),
('E505','Chad','Wilson','M','Associate Data
Scientist','Healthcare',5,'Canada','North America',5000,2,'E083','P103'),
('E532','Claire','Brennan','F','Associate Data
Scientist','Automative',3,'Germany','Europe',4300,1,'E428','P204'),
('E583','Janet','Hale','F','Manager','Retail',14,'Colombia','South
America',10000,2,'E001',NULL),
('E612','Tracy','Norris','F','Manager','Retail',13,'India','Asia',8500,4,
'E001',NULL),
('E620','Katrina','Allen','F','Junior Data
Scientist','Retail',2,'India','Asia',3000,1,'E612','P406'),
('E640','Jenifer','Jhones','F','Junior Data
Scientist','Retail',1,'Colombia','South America',2800,4,'E612','P406');

-- Importing values in Data Science Team table
INSERT INTO data_science_team_table VALUES
```

```
('E005','Eric','Hoffman','M','Lead Data
Scientist','Finance',11,'USA','North America'),
('E010','William','Butler','M','Lead Data
Scientist','Automative',12,'France','Europe'),
('E052','Dianna','Wilson','F','Senior Data
Scientist','Healthcare',6,'Canada','North America'),
('E057','Dorothy','Wilson','F','Senior Data
Scientist','Healthcare',9,'USA','North America'),
('E204','Karene','Nowak','F','Senior Data
Scientist','Automative',8,'Germany','Europe'),
('E245','Nian','Zhen','M','Senior Data
Scientist','Retail',6,'China','Asia'),
('E260','Roy','Collins','M','Senior Data
Scientist','Retail',7,'India','Asia'),
('E403','Steve','Hoffman','M','Associate Data
Scientist','Finance',4,'USA','North America'),
('E478','David','Smith','M','Associate Data
Scientist','Retail',3,'Colombia','South America'),
('E505','Chad','Wilson','M','Associate Data
Scientist','Healthcare',5,'Canada','North America'),
('E532','Claire','Brennan','F','Associate Data
Scientist','Automative',3,'Germany','Europe'),
('E620','Katrina','Allen','F','Junior Data
Scientist','Retail',2,'India','Asia'),
('E640','Jenifer','Jhones','F','Junior Data
Scientist','Retail',1,'Colombia','South America');

-- Writing 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 emp_table.emp_id,CONCAT(emp_table.first_name,"
",emp_table.last_name) AS NAME,emp_table.gender,emp_table.dept,

project_table.project_name,project_table.start_date,project_table.closure
_date,project_table.dev_qtr,project_table.status_
       FROM emp_table
       LEFT JOIN project_table
       ON emp_table.project_id=project_table.project_id;

-- Writing a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER,
DEPARTMENT, and EMP_RATING if the EMP_RATING is:
-- less than two
-- greater than four
-- between two and four

SELECT emp_id,first_name,last_name,gender,dept,emp_rating FROM emp_table
WHERE (emp_rating<2 OR emp_rating>4 OR (emp_rating BETWEEN 2 AND 4));

-- Writing a query to concatenate the FIRST_NAME and the LAST_NAME of
employees in the Finance department from the employee table
-- and then giving the resultant column alias as NAME.

SELECT CONCAT(first_name," ",last_name) AS NAME,dept FROM emp_table
WHERE dept='Finance';
```

```
-- Writing a query to list only those employees who have someone
reporting to them. Also, showing the number of reporters (including the
President).
SELECT CONCAT(first_name,' ',last_name) AS Managers, NULL AS
Total_reporters FROM emp_table
WHERE emp_id IN(SELECT DISTINCT manager_id FROM emp_table)
UNION
SELECT NULL AS Managers, COUNT(manager_id) AS Total_Reporters FROM
emp_table
ORDER BY Total_Reporters DESC;

-- Writing a query to list down all the employees from the healthcare and
finance departments using union.
-- Taking data from the employee record table.
SELECT CONCAT(first_name," ",last_name) AS NAME FROM emp_table
WHERE dept='Healthcare'
UNION
SELECT CONCAT(first_name," ",last_name) FROM emp_table
WHERE dept='Finance';

-- Write a query to list down employee details such as EMP_ID,
FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, and EMP_RATING grouped by dept.
-- Also include the respective employee rating along with the max emp
rating for the department.

SELECT emp_id,first_name,last_name,role_,dept,emp_rating AS
Individual_employee_rating,MAX(emp_rating) OVER(PARTITION BY dept) AS
Maximum_employee_rating_per_department
FROM emp_table
ORDER BY dept;

-- Writing a query to calculate the minimum and the maximum salary of the
employees in each role. Taking data from the employee record table.
SELECT role_,MIN(salary) AS Minimum_Salary_per_role,MAX(salary) AS
Maximum_Salary_per_role
FROM emp_table
GROUP BY role_;

-- Writing a query to assign ranks to each employee based on their
experience. Taking data from the employee record table.
SELECT CONCAT(first_name," ",last_name) AS NAME,exp AS
Experience ,ROW_NUMBER() OVER(ORDER BY EXP DESC) AS Rank_ FROM emp_table;

-- Writing a query to create a view that displays employees in various
countries whose salary is more than six thousand.
-- Taking data from the employee record table.

CREATE VIEW emp_salary_greater_than_6000 AS
SELECT CONCAT(first_name," ",last_name) AS NAME,country,salary FROM
emp_table
WHERE salary>6000;

SELECT * FROM emp_salary_greater_than_6000;
```

```
-- Write a nested query to find employees with experience of more than
ten years. Taking data from the employee record table.
SELECT CONCAT(first_name," ",last_name) AS NAME, exp FROM emp_table
WHERE emp_id IN(
                        SELECT emp_id FROM emp_table
                WHERE exp>10
                );
-- Writing a query to create a stored procedure to retrieve the details
of the employees whose experience is more than three years.
-- Taking data from the employee record table.

DELIMITER &&

CREATE PROCEDURE Employees_with_mid_experience()
BEGIN
SELECT * FROM emp_table
WHERE exp>3;
END
&&

DELIMITER ;
CALL Employees_with_mid_experience();

-- Writing 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 organizationâ€™s set standard.
-- The standard being:
-- For an employee with experience less than or equal to 2 years assign
'JUNIOR DATA SCIENTIST',
-- For an employee with the experience of 2 to 5 years assign 'ASSOCIATE
DATA SCIENTIST',
-- For an employee with the experience of 5 to 10 years assign 'SENIOR
DATA SCIENTIST',
-- For an employee with the experience of 10 to 12 years assign 'LEAD
DATA SCIENTIST',
-- For an employee with the experience of 12 to 16 years assign
'MANAGER'.

DELIMITER //
CREATE FUNCTION job_profile_check(experience INT)
RETURNS VARCHAR(40)
DETERMINISTIC
BEGIN
DECLARE designation VARCHAR(40);
IF experience<=2 THEN
SET designation='JUNIOR DATA SCIENTIST';
ELSEIF experience>2 AND experience<=5 THEN
SET designation='ASSOCIATE DATA SCIENTIST';
ELSEIF experience>5 AND experience<=10 THEN
SET designation='SENIOR DATA SCIENTIST';
ELSEIF experience>10 AND experience<=12 THEN
SET designation='LEAD DATA SCIENTIST';
```

```
ELSEIF experience>12 AND experience<=16 THEN
SET designation='MANAGER';
END IF;
RETURN designation;
END //

DELIMITER ;

SELECT CONCAT(first_name," ",last_name) AS NAME, exp, role_ AS
Current_role,job_profile_check(exp)AS Organization_standard_role
FROM data_science_team_table;

-- Create an index to improve the cost and performance of the query to
find the employee
--  whose FIRST_NAME is â€˜Ericâ€™ in the employee table after checking
the execution plan.

SELECT * FROM emp_table
WHERE first_name='Eric';
-- query cost before optimaztion=2.15

CREATE INDEX idx_name ON emp_table(first_name);

SELECT * FROM emp_table
WHERE first_name='Eric';

-- query cost after optimization=0.35

-- Writing a query to calculate the bonus for all the employees, based on
their ratings and salaries
-- (Using the formula: 5% of salary * employee rating).

SELECT CONCAT(first_name," ",last_name) AS NAME,
salary,ROUND((0.05*salary*emp_rating),2) AS Bonus
FROM emp_table;

-- Writing a query to calculate the average salary distribution based on
the continent and country.
-- Taking data from the employee record table.

SELECT continent,country,ROUND(AVG(salary),2) AS Average_Salary
FROM emp_table
GROUP BY continent,country
ORDER BY continent,country;
```