

ScienceQtech Employee Performance Mapping.

Course-end Project 1

Description

ScienceQtech is a startup that works in the Data Science field. ScienceQtech has worked on fraud detection, market basket, self-driving cars, supply chain, algorithmic early detection of lung cancer, customer sentiment, and the drug discovery field. With the annual appraisal cycle around the corner, the HR department has asked you (Junior Database Administrator) to generate reports on employee details, their performance, and on the project that the employees have undertaken, to analyze the employee database and extract specific data based on different requirements.

Objective:

To facilitate a better understanding, managers have provided ratings for each employee which will help the HR department to finalize the employee performance mapping. As a DBA, you should find the maximum salary of the employees and ensure that all jobs are meeting the organization's profile standard. You also need to calculate bonuses to find extra cost for expenses. This will raise the overall performance of the organization by ensuring that all required employees receive training.

- Note: You must download the dataset from the course resource section in LMS and create a table to perform the above objective.

Dataset description:

emp_record_table: It contains the information of all the employees.

- EMP_ID – ID of the employee
- FIRST_NAME – First name of the employee
- LAST_NAME – Last name of the employee
- GENDER – Gender of the employee
- ROLE – Post of the employee
- DEPT – Field of the employee
- EXP – Years of experience the employee has
- COUNTRY – Country in which the employee is presently living

- CONTINENT – Continent in which the country is
- SALARY – Salary of the employee
- EMP_RATING – Performance rating of the employee
- MANAGER_ID – The manager under which the employee is assigned
- PROJ_ID – The project on which the employee is working or has worked on

Proj_table: It contains information about the projects.

- PROJECT_ID – ID for the project*
- PROJ_Name – Name of the project*
- DOMAIN – Field of the project*
- START_DATE – Day the project began*
- CLOSURE_DATE – Day the project was or will be completed*
- DEV_QTR – Quarter in which the project was scheduled*
- STATUS – Status of the project currently

Data_science_team: It contains information about all the employees in the Data Science team.

- EMP_ID – ID of the employee*
- FIRST_NAME – First name of the employee*
- LAST_NAME – Last name of the employee*
- GENDER – Gender of the employee*
- ROLE – Post of the employee*
- DEPT – Field of the employee*
- EXP – Years of experience the employee has*
- COUNTRY – Country in which the employee is presently living*
- CONTINENT – Continent in which the country is

The task to be performed:

1. Create a database named employee, then import data_science_team.csv proj_table.csv and emp_record_table.csv into the employee database from the given resources.
2. Create an ER diagram for the given employee database.
3. 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.
4. Write 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

5. 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.

6. Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President).
7. Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.
8. 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.
9. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.
10. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.
11. Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table.
12. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.
13. Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years. Take data from the employee record table.
14. 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.

Note: 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'.

15. 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.
16. 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).
17. Write a query to calculate the average salary distribution based on the continent and country. Take data from the employee record table.

-- Dropping Database

```
DROP DATABASE IF EXISTS employee;
```

The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following script:

```
-- Dropping Database
DROP DATABASE IF EXISTS employee;

-- 1. Create a database named employee, then import data_science_team.csv proj_table.csv
-- and emp_record_table.csv into the employee database from the given resources.

-- CREATE DATABASE
CREATE DATABASE IF NOT EXISTS employee;

-- USE DATABASE
USE employee;

-- CREATE TABLES
-- emp_record_table
CREATE TABLE IF NOT EXISTS emp_record_table(
    EMP_ID VARCHAR(6) NOT NULL,
    FIRST_NAME VARCHAR(30) NOT NULL,
    LAST_NAME VARCHAR(30) NOT NULL,
    GENDER VARCHAR(4) NOT NULL,
    ROLE VARCHAR(40) NOT NULL,
    DEPT VARCHAR(40) NOT NULL,
    EXP INT NOT NULL,
    COUNTRY VARCHAR(20) NOT NULL,
    CONTINENT VARCHAR(20) NOT NULL,
    SALARY INT NOT NULL,
    EMP_RATING INT NOT NULL,
    MANAGER_ID VARCHAR(6) ,
    PROJ_ID VARCHAR(6)
);
```

The Output pane shows a warning message: "1 0:15:21 DROP DATABASE IF EXISTS... 0 rows affected, 1 warning(s); 1008 Can't drop database 'employee'; database doesn't exist".

-- 1. Create a database named employee, then import data_science_team.csv proj_table.csv and emp_record_table.csv into the employee database from the given resources.

-- CREATE DATABASE

```
CREATE DATABASE IF NOT EXISTS employee;
```

-- USE DATABASE

```
USE employee;
```

-- CREATE TABLES

-- emp_record_table

```
CREATE TABLE IF NOT EXISTS emp_record_table(
    EMP_ID VARCHAR(6) NOT NULL,
    FIRST_NAME VARCHAR(30) NOT NULL,
    LAST_NAME VARCHAR(30) NOT NULL,
    GENDER VARCHAR(4) NOT NULL,
    ROLE VARCHAR(40) NOT NULL,
    DEPT VARCHAR(40) NOT NULL,
    EXP INT NOT NULL,
    COUNTRY VARCHAR(20) NOT NULL,
    CONTINENT VARCHAR(20) NOT NULL,
    SALARY INT NOT NULL,
    EMP_RATING INT NOT NULL,
    MANAGER_ID VARCHAR(6) ,
    PROJ_ID VARCHAR(6)
);
```

```
-- proj_table
```

```
CREATE TABLE IF NOT EXISTS proj_table(
    PROJ_ID VARCHAR(6),
    PROJ_Name VARCHAR(40) NOT NULL,
    DOMAIN VARCHAR(30) NOT NULL,
    START_DATE VARCHAR(20) NOT NULL,
    CLOSURE_DATE VARCHAR(20) NOT NULL,
    DEV_QTR VARCHAR(2) NOT NULL,
    STATUS VARCHAR(20) NOT NULL
);
```

```
-- Data_science_team
```

```
CREATE TABLE IF NOT EXISTS data_science_team(
    EMP_ID VARCHAR(6) NOT NULL,
    FIRST_NAME VARCHAR(30) NOT NULL,
    LAST_NAME VARCHAR(30) NOT NULL,
    GENDER VARCHAR(4) NOT NULL,
    ROLE VARCHAR(40) NOT NULL,
    DEPT VARCHAR(20) NOT NULL,
    EXP INT NOT NULL,
    COUNTRY VARCHAR(20) NOT NULL,
    CONTINENT VARCHAR(20) NOT NULL
);
```

```
-- LOAD DATA -- proj_table
```

```
LOAD DATA LOCAL INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/proj_table.csv'
INTO TABLE proj_table
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

```
-- Data_science_team
```

```
LOAD DATA LOCAL INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/data_science_team.csv'
INTO TABLE data_science_team
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

```
-- emp_record_table
```

```
LOAD DATA LOCAL INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/emp_record_table.csv'
```

```

INTO TABLE emp_record_table
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

```

The screenshot shows the MySQL Workbench interface with two tabs open: 'emp_record_table' and 'proj_table'. The left pane displays the 'Schemas' tree, which includes the 'employee' schema containing tables like 'emp_record_table', 'data_science_team', and 'proj_table'. The right pane shows the SQL editor with the following code:

```

CREATE TABLE IF NOT EXISTS `emp_record_table`(
    `EMP_ID` VARCHAR(20) NOT NULL,
    `FIRST_NAME` VARCHAR(30) NOT NULL,
    `LAST_NAME` VARCHAR(30) NOT NULL,
    `GENDER` VARCHAR(4) NOT NULL,
    `ROLE` VARCHAR(40) NOT NULL,
    `DEPT` VARCHAR(40) NOT NULL,
    `EXP` INT NOT NULL,
    `COUNTRY` VARCHAR(20) NOT NULL,
    `CONTINENT` VARCHAR(20) NOT NULL,
    `SALARY` INT NOT NULL,
    `EMP_RATING` INT NOT NULL,
    `MANAGER_ID` VARCHAR(5),
    `PROJ_ID` VARCHAR(5)
) ENGINE=InnoDB;

-- proj_table
CREATE TABLE IF NOT EXISTS `proj_table`(
    `PROJ_ID` VARCHAR(10) NOT NULL,
    `DNAME` VARCHAR(10) NOT NULL,
    `START_DATE` VARCHAR(20) NOT NULL,
    `CLOSURE_DATE` VARCHAR(20) NOT NULL,
    `DEV_QTR` VARCHAR(2) NOT NULL,
    `STATUS` VARCHAR(20) NOT NULL
) ENGINE=InnoDB;

```

The 'Output' tab shows the execution results with 177 rows affected.

The screenshot shows the MySQL Workbench interface with the 'Table Data Import' dialog open over the 'emp_record_table' tab. The 'Configure Import Settings' section shows the file format as 'csv' and encoding as 'utf-8'. The 'Columns' section maps columns from the CSV file to the database table:

Source Column	Dest Column
EMP_ID	EMP_ID
FIRST_NAME	FIRST_NAME
LAST_NAME	LAST_NAME
GENDER	GENDER
ROLE	ROLE
DEPT	DEPT

The 'Data Preview' section shows the first few rows of the CSV file:

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT
E005	Eric	Hoffman	M	LEAD DATA	FINANCE	11	USA	NORTH AM.
E009	Robert	Miller	M	LEAD DATA	AUTOMOTIVE	12	FRANCE	EUROPE
E052	Diana	Wilson	F	SENIOR DA	HEALTHCARE	6	CANADA	NORTH AM.
E057	Dorothy	Wilson	F	SENIOR DA	HEALTHCARE	9	USA	NORTH AM.
E204	Karen	Nowak	F	SENIOR DA	AUTOMOTIVE	8	GERMANY	EUROPE

The 'Output' tab shows the import process with 12 rows affected.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the Navigator and Schemas (employee). The main area has a query editor with the following content:

```
-- Uncomment next line if the file loads automatically else load CSV files manually
-- 2. Create an ER diagram for the given employee database.
-- 3. 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.
-- 4. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:
-- less than two
-- SELECT
-- emp_record_table.EMP_ID,
-- emp_record_table.FIRST_NAME,
-- emp_record_table.LAST_NAME,
-- emp_record_table.GENDER,
-- emp_record_table.DEPT
```

Below the query editor, the Object Info and Session tabs are visible. The History tab shows the following log:

Time	Action	Message	Duration / Fetch
7 01:16:32	LOAD DATA LOCAL INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Downloads/data_science_jean.csv' INTO TABLE data_science_jean	Error Code: 3945. Loading local data is disabled; this must be enabled on both the client and server sides	0.000 sec
8 01:16:32	LOAD DATA LOCAL INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Downloads/emp_record_table.csv' INTO TABLE emp_record_table	Error Code: 3945. Loading local data is disabled; this must be enabled on both the client and server sides	0.000 sec
9 01:19:21	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
10 01:19:21	SHOW DATABASES	OK	0.000 sec
11 01:19:26	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
12 01:19:26	SHOW COLUMNS FROM `employee`.`data_science_jean`	OK	0.000 sec
13 01:19:37	PREPARE stmt1 FROM INSERT INTO `employee`.`data_science_jean`(`EMP_ID`, `FIRST_NAME`, `LAST_NAME`, `GENDER`)	OK	0.000 sec
14 01:19:37	DEALLOCATE PREPARE stmt1	OK	0.000 sec
15 01:19:44	SELECT * FROM employee.emp_record_table LIMIT 0,1000	0 rows returned	0.000 sec / 0.000 sec

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** Local instance MySQL80
- Tables:** employee (with sub-tables: data_science_team, record_table, proj_table), views, stored procedures, functions, and schema sakila.
- Data Import Dialog:** A modal window titled "Configure Import Settings" is open, showing the detected file format as CSV and columns mapping: EMP_ID to EMP_ID, FIRST_NAME to FIRST_NAME, LAST_NAME to LAST_NAME, GENDER to GENDER, ROLE to ROLE, and DEPT to DEPT.
- Table Data Import:** A preview table titled "emp_record_table" displays data for employees E001 through E002.
- History Log:** A scrollable list of recent queries, mostly SELECT statements, with execution times.

MySQL Workbench

Local instance MySQL 8.0

File Edit View Query Database Server Tools Scripting Help

Navigator:

SCHEMAS

- Filter objects
- employee
 - Tables
 - emp_table
 - data_science_team
 - emp_record_table
 - proj_table
 - Views
 - Stored Procedures
 - Functions
 - sakila
 - testdb
 - school_ranking_analysis
 - sql_basics
 - tpc
 - us_crime
 - world

Administration Schemas Information

Table: proj_table

Columns:

PROJ_ID	varchar(6)
PROJ_NAME	varchar(20)
DOMAIN	varchar(10)
START_DATE	date
CLOSE_DATE	date
DEV_QTR	varchar(2)
STATUS	varchar(20)

Object Info Session

Query Completed

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT
E005	Eric	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA
E010	Willen	Buller	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE
E015	Diane	Miller	F	SENIOR DATA SCIENTIST	HEALTHCARE	10	CANADA	NORTH AMERICA
E037	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA
E204	Karen	Novak	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE
E405	Man	Zhen	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA
E410	Roy	Callahan	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA
E478	Dave	Smith	M	ASSOCIATE DATA SCIENTIST	RETAIL	3	COLOMBIA	SOUTH AMERICA
E205	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA
E332	Claire	Allen	F	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	2	GERMANY	EUROPE
E620	Katrina	Alien	F	JUNIOR DATA SCIENTIST	RETAIL	1	INDIA	ASIA
E640	Jenifer	Jones	F	JUNIOR DATA SCIENTIST	RETAIL	1	COLOMBIA	SOUTH AMERICA

Action Output

Time	Action	Message	Duration / Fetch
21 01:20:41	DEALLOCATE PREPARE stnt	OK	0.000 sec
22 01:20:40	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
23 01:20:40	SHOW DATABASES	OK	0.000 sec
24 01:20:42	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
25 01:20:42	SHOW COLUMNS FROM employee.`proj_table`	OK	0.000 sec
26 01:20:46	PREPARE stnt FROM INSERT INTO `employee`.`proj_table`(`PROJ_ID`,`PROJ_Name`,`DOMAIN`,`START_DATE`,`CLOSE_DATE`,`DEV_QTR`,`STATUS`)	OK	0.000 sec
27 01:20:46	DEALLOCATE PREPARE stnt	OK	0.000 sec
28 01:20:58	SELECT * FROM employee.data_science_team LIMIT 0, 1000	13 rows returned	0.000 sec / 0.000 sec
29 01:21:00	SELECT * FROM employee.data_science_team LIMIT 0, 1000	13 rows returned	0.000 sec / 0.000 sec

MySQL Workbench Local instance MySQL80

Schema: data_science_team

Tables:

- empLOYEE (EMP_ID, FIRST_NAME, LAST_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, EMP_RATING, MANAGER_ID, PROJ_ID)
- proj_TABLE (PROJ_ID, PROJ_Name, DOMAIN, START_DATE, CLOSURE_DATE, DEV_QTR, STATUS)
- emp_RECORD_TABLE (EMP_ID, FIRST_NAME, LAST_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, EMP_RATING, MANAGER_ID, PROJ_ID)
- data_science_team (EMP_ID, FIRST_NAME, LAST_NAME, GENDER, ROLE, DEPT, EXP, COUNTRY, CONTINENT, SALARY, EMP_RATING, MANAGER_ID, PROJ_ID)

Result Grid (emp_record_table)

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID
E001	Arthur	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3	E013	P205
E005	David	Smith	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	INDIA	ASIA	7500	2	E013	P205
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2	E028	P204
E052	Diana	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	9500	5	E083	P203
E057	Walter	Wolff	F	SENIOR DATA SCIENTIST	HEALTHCARE	7	USA	NORTH AMERICA	9200	3	E083	P302
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5	E001	None
E103	Emily	Grove	F	MANAGER	FINANCE	14	GERMANY	NORTH AMERICA	10500	4	E001	None
E204	Karen	Newcek	F	MANAGER	DATA SCIENTIST	10	RETAIL	ASIA	7500	5	E028	P204
E215	Zane	Shaw	M	SENIOR DATA SCIENTIST	DATA SCIENTIST	8	CHINA	ASIA	6500	2	E083	P209
E390	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	7000	3	E083	NA
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3	E013	P205
E408	Allen	Wong	M	MANAGER	DATA SCIENTIST	10	CHINA	ASIA	8000	4	E001	None
E478	David	Smith	M	ASSOCIATE DATA SCIENTIST	RETAIL	3	COLOMBIA	SOUTH AMERICA	4000	4	E083	P209
E505	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2	E083	P103
E532	Claire	Brennan	F	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	3	GERMANY	EUROPE	4500	1	E028	P204
E533	Mark	Hall	M	MANAGER	DATA SCIENTIST	10	COLOMBIA	SOUTH AMERICA	9000	2	E001	None
E612	Tracy	Norris	F	MANAGER	RETAIL	13	INDIA	ASIA	8500	4	E001	None
E620	Katrina	Allen	F	JUNIOR DATA SCIENTIST	RETAIL	2	INDIA	ASIA	3000	1	E012	P406
E640	Jenifer	Jones	F	JUNIOR DATA SCIENTIST	RETAIL	1	COLOMBIA	SOUTH AMERICA	2800	4	E012	P406

Result Grid (proj_table)

PROJ_ID	PROJ_Name	DOMAIN	START_DATE	CLOSURE_DATE	DEV_QTR	STATUS
P103	Drug Discovery	HEALTHCARE	04-06-2021	6/20/2021	Q1	DONE
P105	AI for Diagnosis	HEALTHCARE	05-06-2021	7/20/2021	Q1	DONE
P109	Market Basket Analysis	RETAIL	04-12-2021	6/20/2021	Q1	DELAYED
P204	Supply Chain Management	AUTOMOTIVE	07/15/2021	9/20/2021	Q2	WIP
P302	Early Detection of Lung Cancer	HEALTHCARE	10-08-2021	12/28/2021	Q3	YTS
P400	Customer Sentiment Analysis	AI/ML	07-09-2021	9/24/2021	Q2	WIP

Result Grid (emp_RECORD_TABLE)

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID
E001	Arthur	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3	E013	P205
E005	David	Smith	M	ASSOCIATE DATA SCIENTIST	DATA SCIENTIST	8	CHINA	ASIA	6500	2	E083	P209
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2	E028	P204
E052	Diana	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	9500	5	E083	P203
E057	Walter	Wolff	F	SENIOR DATA SCIENTIST	HEALTHCARE	7	USA	NORTH AMERICA	9200	3	E083	P302
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5	E001	None
E103	Emily	Grove	F	MANAGER	FINANCE	14	GERMANY	NORTH AMERICA	10500	4	E001	None
E204	Karen	Newcek	F	MANAGER	DATA SCIENTIST	10	RETAIL	ASIA	7500	5	E028	P204
E215	Zane	Shaw	M	SENIOR DATA SCIENTIST	DATA SCIENTIST	8	CHINA	ASIA	6500	2	E083	P209
E390	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	7000	3	E083	NA
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3	E013	P205
E408	Allen	Wong	M	MANAGER	DATA SCIENTIST	10	CHINA	ASIA	8000	4	E001	None
E478	David	Smith	M	ASSOCIATE DATA SCIENTIST	RETAIL	3	COLOMBIA	SOUTH AMERICA	4000	4	E083	P209
E505	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2	E083	P103
E532	Claire	Brennan	F	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	3	GERMANY	EUROPE	4500	1	E028	P204
E533	Mark	Hall	M	MANAGER	DATA SCIENTIST	10	COLOMBIA	SOUTH AMERICA	9000	2	E001	None
E612	Tracy	Norris	F	MANAGER	RETAIL	13	INDIA	ASIA	8500	4	E001	None
E620	Katrina	Allen	F	JUNIOR DATA SCIENTIST	RETAIL	2	INDIA	ASIA	3000	1	E012	P406
E640	Jenifer	Jones	F	JUNIOR DATA SCIENTIST	RETAIL	1	COLOMBIA	SOUTH AMERICA	2800	4	E012	P406

Result Grid (data_science_team)

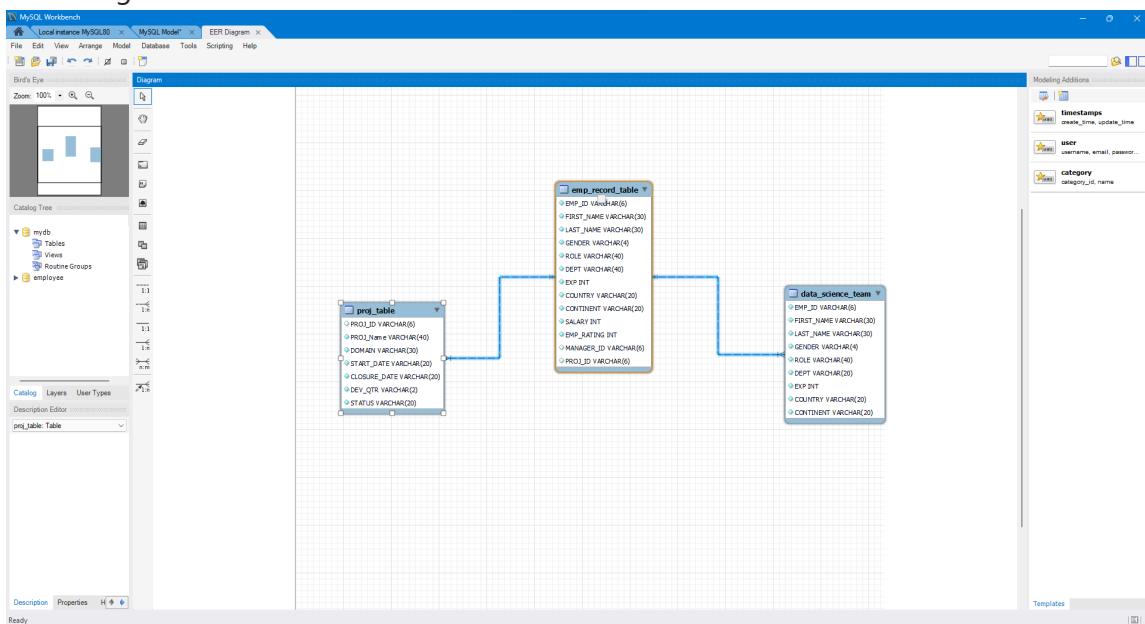
EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID
E001	Arthur	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3	E013	P205
E005	David	Smith	M	ASSOCIATE DATA SCIENTIST	DATA SCIENTIST	8	CHINA	ASIA	6500	2	E083	P209
E010	William	Butler	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2	E028	P204
E052	Diana	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	9500	5	E083	P203
E057	Walter	Wolff	F	SENIOR DATA SCIENTIST	HEALTHCARE	7	USA	NORTH AMERICA	9200	3	E083	P302
E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5	E001	None
E103	Emily	Grove	F	MANAGER	FINANCE	14	GERMANY	NORTH AMERICA	10500	4	E001	None
E204	Karen	Newcek	F	MANAGER	DATA SCIENTIST	10	RETAIL	ASIA	7500	5	E028	P204
E215	Zane	Shaw	M	SENIOR DATA SCIENTIST	DATA SCIENTIST	8	CHINA	ASIA	6500	2	E083	P209
E390	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	7000	3	E083	NA
E403	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3	E013	P205
E408	Allen	Wong	M	MANAGER	DATA SCIENTIST	10	CHINA	ASIA	8000	4	E001	None
E478	David	Smith	M	ASSOCIATE DATA SCIENTIST	RETAIL	3	COLOMBIA	SOUTH AMERICA	4000	4	E083	P209
E505	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2	E083	P103
E532	Claire	Brennan	F	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	3	GERMANY	EUROPE	4500	1	E028	P204
E533	Mark	Hall	M	MANAGER	DATA SCIENTIST	10	COLOMBIA	SOUTH AMERICA	9000	2	E001	None
E612	Tracy	Norris	F	MANAGER	RETAIL	13	INDIA	ASIA	8500	4	E001	None
E620	Katrina	Allen	F	JUNIOR DATA SCIENTIST	RETAIL	2	INDIA	ASIA	3000	1	E012	P406
E640	Jenifer	Jones	F	JUNIOR DATA SCIENTIST	RETAIL	1	COLOMBIA	SOUTH AMERICA	2800	4	E012	P406

Result Grid (proj_table)

PROJ_ID	PROJ_Name	DOMAIN	START_DATE	CLOSURE_DATE	DEV_QTR	STATUS
P103	Drug Discovery	HEALTHCARE	04-06-2021	6/20/2021	Q1	DONE
P105	AI for Diagnosis	HEALTHCARE	05-06-2021	7/20/2021	Q1	DONE
P109	Market Basket Analysis	RETAIL	04-12-2021	6/20/2021	Q1	DELAYED
P204	Supply Chain Management	AUTOMOTIVE	07/15/2021	9/20/2021	Q2	WIP
P302	Early Detection of Lung Cancer	HEALTHCARE	10-08-2021	12/28/2021	Q3	YTS
P400	Customer Sentiment Analysis	AI/ML	07-09-2021	9/24/2021	Q2	WIP

-- 2. Create an ER diagram for the given employee database.

-- ER Diagram



-- 3. 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
    emp_record_table.EMP_ID,
    emp_record_table.FIRST_NAME,
    emp_record_table.LAST_NAME,
    emp_record_table.GENDER,
    emp_record_table.DEPT
FROM
    emp_record_table
ORDER BY emp_record_table.DEPT;

```

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL query for selecting employee details by department.
- Results Grid:** Displays the results of the query, showing columns: EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPT. The data includes rows such as E001 (Arthur, Black, M, AUTOMOTIVE), E002 (Bernard, Bush, M, AUTOMOTIVE), E004 (Karen, Nowak, F, AUTOMOTIVE), E010 (Pete, Allen, M, AUTOMOTIVE), E012 (Clare, Brennan, F, AUTOMOTIVE), E005 (Eric, Hoffman, M, FINANCE), E003 (Emily, Grove, F, FINANCE), E007 (Steve, Hoffmann, M, FINANCE), E008 (Diane, Williams, F, FINANCE), E009 (Dorothy, Wilson, F, HEALTHCARE), E005 (Patrick, Voltz, M, HEALTHCARE), E005 (Chad, Wilson, M, HEALTHCARE), E014 (Alan, Dunn, M, RETAIL), E006 (Ray, Collins, M, RETAIL), E010 (David, Smith, M, RETAIL), E011 (Helen, Price, F, RETAIL), E012 (Tracy, Norris, F, RETAIL), E020 (Fabrice, Allen, F, RETAIL), and E040 (Jennifer, Jones, F, RETAIL).
- Output:** Shows the execution log with actions like SELECT, LOAD DATA, and the final query results.

-- 4. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:

-- less than two

```

SELECT
    emp_record_table.EMP_ID,
    emp_record_table.FIRST_NAME,
    emp_record_table.LAST_NAME,
    emp_record_table.GENDER,
    emp_record_table.DEPT,
    emp_record_table.EMP_RATING
FROM
    emp_record_table
WHERE
    emp_record_table.EMP_RATING < 2
ORDER BY emp_record_table.EMP_RATING;

```

-- greater than four

```
SELECT
    emp_record_table.EMP_ID,
    emp_record_table.FIRST_NAME,
    emp_record_table.LAST_NAME,
    emp_record_table.GENDER,
    emp_record_table.DEPT,
    emp_record_table.EMP_RATING

FROM
    emp_record_table
WHERE
    emp_record_table.EMP_RATING > 4
ORDER BY emp_record_table.EMP_RATING;
```

-- between two and four

```
SELECT
    emp_record_table.EMP_ID,
    emp_record_table.FIRST_NAME,
    emp_record_table.LAST_NAME,
    emp_record_table.GENDER,
    emp_record_table.DEPT,
    emp_record_table.EMP_RATING

FROM
    emp_record_table
WHERE
    emp_record_table.EMP_RATING BETWEEN 2 AND 4
ORDER BY emp_record_table.EMP_RATING;
```

-- 4. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:

```

229   ORDER BY emp_record_table.DEPARTMENT
230
231   -- less than two
232
233 •   SELECT
234     emp_record_table.EMP_ID,
235     emp_record_table.FIRST_NAME,
236     emp_record_table.LAST_NAME,
237     emp_record_table.GENDER,
238     emp_record_table.DEPARTMENT,
239     emp_record_table.EMP_RATING
240
241   FROM
242     emp_record_table
243
244   WHERE
245     emp_record_table.EMP_RATING < 2
246
247   ORDER BY emp_record_table.EMP_RATING
248
249   -- greater than four

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPARTMENT	EMP_RATING
E327	Dorothy	Wick	F	HEALTHCARE	1
E332	Clare	Brennan	F	AUTOMOTIVE	1
E620	Katrina	Allen	F	RETAIL	1

Output

```

SELECT
    emp_record_table.EMP_ID,
    emp_record_table.FIRST_NAME,
    emp_record_table.LAST_NAME,
    emp_record_table.GENDER,
    emp_record_table.DEPARTMENT,
    emp_record_table.EMP_RATING
FROM
    emp_record_table
WHERE
    emp_record_table.EMP_RATING < 2
ORDER BY emp_record_table.EMP_RATING
LIMIT 0, 1000

```

-- 4. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:

```

245   emp_record_table.EMP_RATING < 2
246
247   ORDER BY emp_record_table.EMP_RATING
248
249 •   SELECT
250     emp_record_table.EMP_ID,
251     emp_record_table.FIRST_NAME,
252     emp_record_table.LAST_NAME,
253     emp_record_table.GENDER,
254     emp_record_table.DEPARTMENT,
255     emp_record_table.EMP_RATING
256
257   FROM
258     emp_record_table
259
260   WHERE
261     emp_record_table.EMP_RATING > 4
262
263   ORDER BY emp_record_table.EMP_RATING
264
265   -- between two and four

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E001	Arthur	Stek	M	ALL	5
E052	Deanna	Wilson	F	HEALTHCARE	5
E083	Pehrak	Voltz	M	HEALTHCARE	5
E284	Karen	Nawak	F	AUTOMOTIVE	5

Output

```

SELECT
    emp_record_table.EMP_ID,
    emp_record_table.FIRST_NAME,
    emp_record_table.LAST_NAME,
    emp_record_table.GENDER,
    emp_record_table.DEPARTMENT,
    emp_record_table.EMP_RATING
FROM
    emp_record_table
WHERE
    emp_record_table.EMP_RATING < 2
ORDER BY emp_record_table.EMP_RATING
LIMIT 0, 1000

```

-- 4. Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:

```

245   emp_record_table.EMP_RATING < 2
246
247   ORDER BY emp_record_table.EMP_RATING
248
249 •   SELECT
250     emp_record_table.EMP_ID,
251     emp_record_table.FIRST_NAME,
252     emp_record_table.LAST_NAME,
253     emp_record_table.GENDER,
254     emp_record_table.DEPARTMENT,
255     emp_record_table.EMP_RATING
256
257   FROM
258     emp_record_table
259
260   WHERE
261     emp_record_table.EMP_RATING > 4
262
263   ORDER BY emp_record_table.EMP_RATING
264
265   -- between two and four

```

-- 5. 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.

```

277
278 •   SELECT
279     CONCAT(emp_record_table.FIRST_NAME,
280           ' '
281           emp_record_table.LAST_NAME)
282
283
284
285
286
287
288

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
E030	William	Butler	M	AUTOMOTIVE	2
E245	Mark	Zhen	M	RETAIL	2
E327	David	Wick	M	HEALTHCARE	2
E883	Sherif	Hale	F	RETAIL	2
E005	Eric	Hoffman	M	FINANCE	3
E260	Roy	Collins	M	RETAIL	3
E284	John	Miller	M	FINANCE	3
E103	Emily	Grove	F	FINANCE	4
E438	Pete	Allen	M	AUTOMOTIVE	4
E420	David	Smith	M	RETAIL	4
E212	Andy	Norris	F	RETAIL	4
E640	Jenifer	Jones	F	RETAIL	4

Output

```

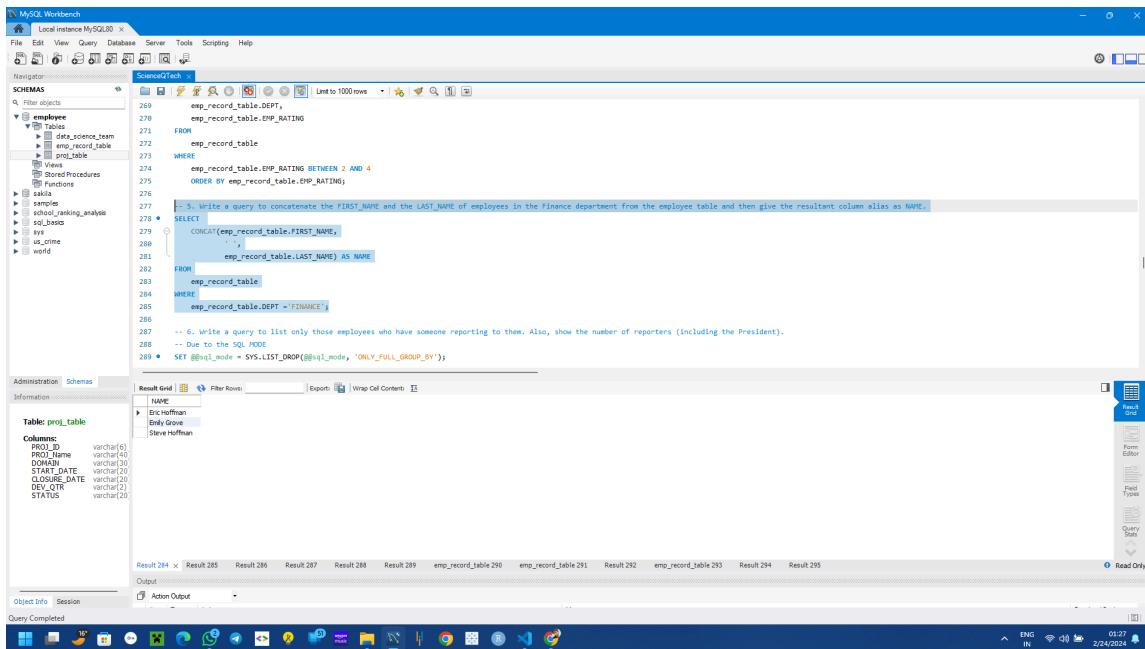
SELECT
    CONCAT(emp_record_table.FIRST_NAME,
           ' '
           emp_record_table.LAST_NAME)
FROM
    emp_record_table
WHERE
    emp_record_table.DEPARTMENT = 'FINANCE'
ORDER BY emp_record_table.EMP_RATING
LIMIT 0, 1000

```

-- 5. 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(emp_record_table.FIRST_NAME,
           ' ',
           emp_record_table.LAST_NAME) AS NAME
FROM
    emp_record_table
WHERE
    emp_record_table.DEPT = 'FINANCE';
```



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
269     emp_record_table.DEPT,
270   emp_record_table.EMP_RATING
271   FROM
272   emp_record_table
273   WHERE
274     emp_record_table.EMP_RATING BETWEEN 2 AND 4
275   ORDER BY emp_record_table.EMP_RATING;
```

Below the editor, the results grid displays the following data:

NAME
Eric Hoffman
Emily Grove
Steve Hoffman

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

-- Due to the SQL MODE

```
SET @@sql_mode = SYS.LIST_DROP(@@sql_mode, 'ONLY_FULL_GROUP_BY');
SELECT
    emp_record_table.FIRST_NAME,
    emp_record_table.LAST_NAME,
    COUNT(emp_record_table.MANAGER_ID) AS NUMBER_OF_REPORTERS
FROM
    emp_record_table
GROUP BY emp_record_table.MANAGER_ID
HAVING COUNT(emp_record_table.MANAGER_ID) > 0
ORDER BY emp_record_table.MANAGER_ID;
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas, employee, emp_record_table, sys.
- Query Editor:** Contains a complex SQL script for reporting on employee data across multiple departments. The script includes subqueries, joins, and aggregate functions like COUNT(). It also uses system variables and comments for documentation.
- Results Grid:** Shows the output of the last query (Result 301) in a tabular format. The columns are FIRST_NAME, LAST_NAME, and NUMBER_OF_REPORTERS. The data includes rows for Patrick, Valerie, Debbie, Eric, William, Buffer, Karen, and Zhen.
- Bottom Navigation:** Result 305, Result 286, Result 287, Result 288, Result 289, emp_record_table 290, emp_record_table 291, Result 292, emp_record_table 293, Result 294, Result 295.
- Status Bar:** Query Completed.

-- 7. 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
    emp_record_table.DEPT = 'HEALTHCARE'
UNION
SELECT *
FROM
    emp_record_table
WHERE
    emp_record_table.DEPT = 'FINANCE'
ORDER BY EMP_ID;
```

MySQL Workbench

Local instance MySQL 8.0 x

File Edit View Query Database Server Tools Scripting Help

Navigator:

SCHEMAS

- employee
 - Tables
 - emp
 - emp_science_team
 - emp_record_table
 - proj_table
 - Views
 - Stored Procedures
 - Functions
 - Triggers
 - sample
 - school_ranking_analysis
 - temp
 - sys
 - us_crime
 - world

SCHEMAS ScinedTech v1

296 GROUP BY emp_record_table.MANAGER_ID
297 HAVING COUNT(emp_record_table.MANAGER_ID) > 0
298
299 ORDER BY emp_record_table.MANAGER_ID;
300
301 -- 7. Write a query to list down all the employees from the healthcare and finance departments using union. Take data from the employee record table.
302
303 SELECT *
304 FROM
305 emp_record_table
306 WHERE
307 emp_record_table.DEPT = 'HEALTHCARE'
308 UNION
309 SELECT *
310 FROM
311 emp_record_table
312 WHERE
313 emp_record_table.DEPT = 'FINANCE'
314
315 -- 8. Write a query to list down employee details such as EMP_ID, FIRST_NAME, LAST_NAME, GENDER, ROLE, DEPARTMENT, and EMP_RATING grouped by dept. Also include the respective employee rating along with the max emp rating for the department
316 SELECT *,
317 MAX(emp_record_table.EMP_RATING)

Administration Schemas Information

Table: proj_table

Columns:

PROJ_ID	PROJ_Name	DEVMGR_ID	START_DATE	CLOSE_DATE	DEV_QTR	STATUS					
E005	Eric Hoffmann	M	LEAD DATA SCIENTIST	FINANCE	11	USA	NORTH AMERICA	8500	3	E003	P005
E006	Diane Moore	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7500	3	E003	P006
E007	Dorothy Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7500	1	E003	P002
E008	Patrick Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5	E001	P008
E009	Emily Grove	F	DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3	E003	P010
E010	Steve Hoffmann	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3	E003	P010
E005	Omid Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2	E003	P003

Result Grid Filter Rows Export Wrap Cell Context: E5

Result 286 x Result 287 Result 288 Result 289 emp_record_table 290 emp_record_table 291 Result 292 emp_record_table 293 Result 294 Result 295

Output Action Output

Query Completed

END IN 03:27 2/24/2024

-- 8. 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 *,  
       MAX(emp_record_table.EMP_RATING)  
FROM  
       emp_record_table  
GROUP BY emp_record_table.DEPT  
ORDER BY emp_record_table.DEPT;
```

The screenshot shows the MySQL Workbench interface. The query editor contains a multi-line SQL script. The results grid displays data from the 'proj_table' for rows E001 through E245. The results output tab shows the execution status: 'Query Completed'.

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID	MAX(emp_record_table.EMP_RATING)
E001	Arthur	Black	M	PRESIDENT		20	USA	NORTH AMERICA	16500	5			5
E002	William	Burn	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2	E058	P204	5
E003	Ben	Hoffman	M	LEAD DATA SCIENTIST	FINANCE	14	USA	NORTH AMERICA	10000	3	E103	P205	4
E004	Diana	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500	5	E083	P203	5
E005	John	Smith	M	SENIOR DATA SCIENTIST	RETAIL	6	CHINA	ASIA	6500	2	E583	P109	4
E245	Nan	Zhen	M	SENIOR DATA SCIENTIST	RETAIL								

-- 9. Write a query to calculate the minimum and the maximum salary of the employees in each role. Take data from the employee record table.

```
SELECT  
       emp_record_table.ROLE,  
       MIN(emp_record_table.SALARY) AS MINIMUM_SALARY,  
       MAX(emp_record_table.SALARY) AS MAXIMUM_SALARY  
FROM  
       emp_record_table  
GROUP BY emp_record_table.ROLE  
ORDER BY emp_record_table.ROLE;
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas, employee, emp_record_table, data_science_team, emp_record_table, emp_record_table, Views, Triggers, Stored Procedures, Functions, sakila, sample, dbms_ranking_analysis, sql_base, sys, utl_smse, world.
- Query Editor:** The code area contains SQL queries for problem 9 and 10. Problem 9 calculates the minimum and maximum salary for each role. Problem 10 assigns ranks based on experience. The code uses aliases like MINIMUM_SALARY and MAXIMUM_SALARY.
- Results Grid:** A table titled "ASSOCIATE DATA SCIENTIST" displays salary data for various roles. The columns are ROLE, MINIMUM_SALARY, and MAXIMUM_SALARY.
- Bottom Navigation:** Object Info, Session, Output, Action Output, Read Only.

ROLE	MINIMUM_SALARY	MAXIMUM_SALARY
ASSOCIATE DATA SCIENTIST	4800	4800
JUNIOR DATA SCIENTIST	2800	3000
LEAD DATA SCIENTIST	8500	9000
MANAGER	8500	11000
PROJECT MANAGER	10500	10500
SENIOR DATA SCIENTIST	5500	7700

-- 10. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

```
SELECT
    emp_record_table.EMP_ID,
    concat(emp_record_table.FIRST_NAME,
           ' ',
           emp_record_table.LAST_NAME) as NAME,
    emp_record_table.DEPT,
    emp_record_table.EXP,
    rank() OVER(ORDER BY emp_record_table.EXP DESC) as EMP_EXP_RANK
FROM
    emp_record_table;
```

The screenshot shows the MySQL Workbench interface with a query editor containing a complex SQL script. The script includes multiple SELECT statements for ranking employees by experience and creating a view for employees in various countries. The results grid displays the ranked employee data.

```
329 GROUP BY emp_record_table.ROLE
330 ORDER BY emp_record_table.ROLE;
331
332 -- 10. Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.
333
334 SELECT
335     emp_record_table.EMP_ID,
336     concat(emp_record_table.FIRST_NAME,
337            ' ',
338            emp_record_table.LAST_NAME) AS NAME,
339     emp_record_table.DEPT,
340     emp_record_table.EXP,
341     rank() OVER(ORDER BY emp_record_table.EXP DESC) AS EXP_RANK
342
343 FROM
344     emp_record_table
345
346 -- 11. Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table.
347
348 CREATE VIEW
349     empl_salary_view AS
```

EMP_ID	NAME	DEPT	EXP	EXP_RANK
E001	Arthur Black	ALL	20	2
E003	Patrick Voltz	HEALTHCARE	15	2
E103	Emily Grove	FINANCE	14	3
E403	Henry Parker	AUTOMOTIVE	12	4
E503	Jerry Hale	RETAIL	14	3
E512	Tracy Norris	RETAIL	13	6
E010	William Butler	AUTOMOTIVE	12	7
E200	Elizabeth Parker	FINANCE	11	8
E057	Dorothy Wilson	HEALTHCARE	9	9
E204	Karlene Howell	AUTOMOTIVE	8	10
E206	Roy Collier	RETAIL	11	11
E300	Donald Wilson	HEALTHCARE	6	12
E454	Nan Zheng	RETAIL	6	12
E505	Chad Wilson	HEALTHCARE	5	14
E507	Steve Parker	RETAIL	15	15
E708	David Smith	RETAIL	3	16
E532	Claire Brennan	AUTOMOTIVE	3	16
E620	Katrina Allen	RETAIL	2	18
E645	Sasha Jackson	ALL	11	19

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

```

CREATE VIEW
    emp_salary_view AS
SELECT
    emp_record_table.EMP_ID,
    emp_record_table.FIRST_NAME,
    emp_record_table.LAST_NAME,
    emp_record_table.SALARY,
    emp_record_table.COUNTRY
FROM
    emp_record_table
WHERE
    emp_record_table.SALARY > 6000
ORDER BY emp_record_table.SALARY;

```

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

employee

Views

emp_salary_view

Information

View: emp_salary_view

Columns:

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	COUNTRY
E345	Man	Zhen	6000	CHINA
E204	Roy	Collins	7000	INDIA
E204	Karen	Nowak	7500	GERMANY
E204	Dorothy	Willis	7700	USA
E005	Eric	Hoffmann	8000	USA
E612	Tracy	Norris	8500	INDIA
E010	William	Butler	9000	FRANCE
E204	Michael	Voss	9500	USA
E383	Janet	Hale	10000	COLOMBIA
E103	Emily	Grove	10500	CANADA
E428	Pete	Allen	11000	GERMANY
E001	Arthur	Blank	11500	USA

Object Info Session Output

Query Completed

01:33 2/24/2024

-- 12. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

```

SELECT
    emp_record_table.EMP_ID,
    emp_record_table.FIRST_NAME,
    emp_record_table.LAST_NAME,
    emp_record_table.EXP
FROM
    emp_record_table
WHERE
    emp_record_table.EXP > 10
ORDER BY emp_record_table.EXP;

```

```

-- 12. Write a nested query to find employees with experience of more than ten years. Take data from the employee record table.

SELECT *
FROM emp_record_table
WHERE emp_record_table.EXP > 10
ORDER BY emp_record_table.EXP;

-- 13. Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years. Take data from the employee record table.

SELECT *
FROM emp_record_table
WHERE emp_record_table.EXP > 3
ORDER BY emp_record_table.EXP;

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	EXP
E005	Brian	Hoffman	12
E009	William	Butler	12
E012	Troy	Norris	13
E003	Emily	Grove	14
E428	Pete	Allen	14
E009	John	Hall	14
E003	Pavlik	Voltz	15
E001	Arthur	Black	20

-- 13. Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years. Take data from the employee record table.

```

SELECT *
FROM
    emp_record_table
WHERE
    emp_record_table.EXP > 3
ORDER BY emp_record_table.EXP;

```

```

-- 13. Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years. Take data from the employee record table.

SELECT *
FROM emp_record_table
WHERE emp_record_table.EXP > 3
ORDER BY emp_record_table.EXP;

-- 14. 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.

-- Note: The standard being:
-- * For an employee with experience less than or equal to 2 years assign 'JUNIOR DATA SCIENTIST',
-- ...

```

Result Grid

EMP_ID	FIRST_NAME	LAST_NAME	EXP
E005	Brian	Hoffman	12
E009	William	Butler	12
E012	Troy	Norris	13
E003	Emily	Grove	14
E428	Pete	Allen	14
E009	John	Hall	14
E003	Pavlik	Voltz	15
E001	Arthur	Black	20

-- 14. 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.

-- Note: 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
    check_role(exp int)
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 AND exp <= 16 THEN
    SET chck = "MANAGER";
end if;
RETURN(chck);
END //
delimiter ;
-- checking Data Science Team
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    ROLE,
    EXP,
    check_role(exp)
FROM
    data_science_team
WHERE data_science_team.ROLE != check_role(exp);
```

The screenshot shows the MySQL Workbench interface with the following details:

MySQL Workbench - Local instance MySQL 8.0.22

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS (Schemas), emp_employee (Tables), emp_salary_view (Views)

SCHEMAS: emp_employee, emp_salary_view, emp_salary_view

emp_employee: emp_id, first_name, last_name, gender, role, dept, exp, country, continent, salary, emp_rating, manager_id, proj_id

emp_salary_view: RETURNS VARCHAR(48) DETERMINISTIC BEGIN DECLARE check VARCHAR(50); IF EXP > 2 THEN SET check = "JUNIOR DATA SCIENTIST"; ELSEIF EXP > 2 AND EXP <= 5 THEN SET check = "ASSOCIATE DATA SCIENTIST"; ELSEIF EXP > 5 AND EXP <= 10 THEN SET check = "SENIOR DATA SCIENTIST"; ELSEIF EXP > 10 AND EXP <= 12 THEN SET check = "LEAD DATA SCIENTIST"; ELSEIF EXP > 12 AND EXP <= 15 THEN SET check = "MANAGER"; END IF; RETURN(check); END // delimiter ; -- checking Data Science Team SELECT * FROM emp_employee;

Administration Schemas Information

Result Grid: emp_salary_view

EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID
E001	Steve	Hoffman	M	ASSOCIATE DATA SCIENTIST	FINANCE	4	USA	NORTH AMERICA	5000	3	E003	P103
E002	Chad	Wilson	M	ASSOCIATE DATA SCIENTIST	HEALTHCARE	5	CANADA	NORTH AMERICA	5000	2	E003	P103
E003	Donna	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	6	CANADA	NORTH AMERICA	5500	5	E003	P103
E004	Mark	Holland	M	SENIOR DATA SCIENTIST	RETAIL	7	CHINA	ASIA	7000	2	E003	P109
E005	Roy	Collins	M	SENIOR DATA SCIENTIST	RETAIL	7	INDIA	ASIA	7000	3	E003	NA
E006	Karen	Newkirk	F	SENIOR DATA SCIENTIST	AUTOMOTIVE	8	GERMANY	EUROPE	7500	5	E008	P204
E007	Dorothy	Wilson	F	SENIOR DATA SCIENTIST	HEALTHCARE	9	USA	NORTH AMERICA	7700	1	E003	P302
E008	Eric	Alonso	M	LEAD DATA SCIENTIST	FINANCE	10	INDIA	ASIA	8000	3	E003	P105
E009	Tracy	Bulmer	M	LEAD DATA SCIENTIST	AUTOMOTIVE	12	FRANCE	EUROPE	9000	2	E003	P204
E010	Willam	Bulmer	M	LEAD DATA SCIENTIST	FINANCE	12	INDIA	ASIA	9000	2	E003	P105
E011	Tracy	Norris	F	MANAGER	RETAIL	13	INDIA	ASIA	8500	4	E001	0008
E012	Emily	Grove	F	MANAGER	FINANCE	14	INDIA	ASIA	8500	4	E001	0008
E013	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	11000	4	E001	0008
E014	Jens	Hahn	F	MANAGER	RETAIL	14	COLOMBIA	SOUTH AMERICA	10000	2	E001	0008
E015	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500	5	E001	0008
E016	Arthur	Beck	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	16000	5	E003	0008

Object Info Session emp_salary_view.emp_salary_view

Query Completed

ENG IN 2/24/2024 01:37

-- 15. 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.

```
SELECT *
FROM
    emp_record_table
WHERE
    emp_record_table.FIRST_NAME = 'Eric';
```

-- 16. 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_record_table.EMP_ID,  
    emp_record_table.FIRST_NAME,
```

```

emp_record_table.DEPT,
emp_record_table.SALARY,
emp_record_table.EMP_RATING,
(emp_record_table.SALARY * 0.05 * emp_record_table.EMP_RATING)

AS BONUS
FROM
    emp_record_table
ORDER BY EMP_ID DESC;

```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** employee, data_science_team, emp_record_table, emp_salary_view, sakila, samples, sql_ranking_analysis, sys, utz_zone, world.
- Query Editor:** The code for question 16 is visible, followed by the code for question 17.
- Result Grid:** The results of the query for question 17 are displayed in a grid format.
- Table Headers:** The columns are EMP_ID, FIRST_NAME, DEPT, SALARY, EMP_RATING, BONUS.
- Data Rows:** The grid contains approximately 20 rows of employee data.

EMP_ID	FIRST_NAME	DEPT	SALARY	EMP_RATING	BONUS
E640	Jenifer	RETAIL	2800	4	560.00
E590	Mike	RETAIL	3000	3	600.00
E612	Tracy	RETAIL	6500	4	1700.00
E583	Janet	RETAIL	4000	2	1000.00
E532	Corey	AUTOMOTIVE	4000	1	215.00
E531	Orn	HEALTHCARE	5000	3	1500.00
E478	David	RETAIL	4000	4	800.00
E428	Pete	AUTOMOTIVE	11000	4	2200.00
E429	Grace	FINANCE	7500	3	1500.00
E260	Roy	RETAIL	7000	3	1550.00
E245	Nan	RETAIL	6500	2	650.00
E252	Karen	AUTOMOTIVE	7900	5	1875.00
E103	Patricia	FINANCE	12000	5	3000.00
E083	Parisa	HEALTHCARE	9500	5	2375.00
E057	Dorothy	HEALTHCARE	7700	1	385.00
E023	Wendy	HEALTHCARE	9500	1	1375.00
E010	Vilma	AUTOMOTIVE	6000	2	900.00
E005	Eric	FINANCE	8500	3	1275.00
E001	Arthur	ALL	3500	5	4125.00

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

```

SELECT
    emp_record_table.COUNTRY,
    emp_record_table.CONTINENT,
    AVG(emp_record_table.SALARY) AS AVERAGE_SALARY
FROM
    emp_record_table
GROUP BY emp_record_table.COUNTRY, emp_record_table.CONTINENT
ORDER BY emp_record_table.CONTINENT, emp_record_table.COUNTRY;

```

MySQL Workbench Local Instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Schemas Filter objects

SCHEMAS

- empLOYEE
- empLOYEE_SALARY
- empLOYEE_RECORD
- empLOYEE_VIEW
- FUNCTIONS
- PROCEDURES
- TRIGGERS
- USERS
- VIEW

Navigator

SearchTech x emp_salary_view

427 FROM emp_record_table

428 WHERE emp_record_table.FIRST_NAME = 'eric';

-- 16. 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).

432 SELECT

433 emp_record_table.EMP_ID,

434 emp_record_table.FIRST_NAME,

435 emp_record_table.DEPT,

436 emp_record_table.SALARY,

437 emp_record_table.EMP_RATING,

438 (emp_record_table.SALARY * 0.05 * emp_record_table.EMP_RATING) AS BONUS

439

440 FROM emp_record_table

441 ORDER BY EMP_ID DESC;

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

444 SELECT

445 ●

Result Grid | Filter Rows: Export: Wrap Cell Contents: Search Grid | Filter Rows: Export: Wrap Cell Contents: Object Info Session Result 321 x

Administration Schemas Information

View: emp_salary_view

Columns:

EMP_ID	FIRST_NAME	LAST_NAME	SALARY	COUNTRY
10000000	vinod	reddy	6000.0000	INDIA
10000001	anil	reddy	6166.6667	INDIA
10000002	praveen	reddy	9000.0000	INDIA
10000003	praveen	reddy	9000.0000	INDIA
10000004	praveen	reddy	9000.0000	INDIA
10000005	praveen	reddy	9000.0000	INDIA
10000006	praveen	reddy	9000.0000	INDIA
10000007	praveen	reddy	9000.0000	INDIA
10000008	praveen	reddy	9000.0000	INDIA
10000009	praveen	reddy	9000.0000	INDIA
10000010	praveen	reddy	9000.0000	INDIA
10000011	praveen	reddy	9000.0000	INDIA
10000012	praveen	reddy	9000.0000	INDIA
10000013	praveen	reddy	9000.0000	INDIA
10000014	praveen	reddy	9000.0000	INDIA
10000015	praveen	reddy	9000.0000	INDIA
10000016	praveen	reddy	9000.0000	INDIA
10000017	praveen	reddy	9000.0000	INDIA
10000018	praveen	reddy	9000.0000	INDIA
10000019	praveen	reddy	9000.0000	INDIA
10000020	praveen	reddy	9000.0000	INDIA
10000021	praveen	reddy	9000.0000	INDIA
10000022	praveen	reddy	9000.0000	INDIA
10000023	praveen	reddy	9000.0000	INDIA
10000024	praveen	reddy	9000.0000	INDIA
10000025	praveen	reddy	9000.0000	INDIA
10000026	praveen	reddy	9000.0000	INDIA
10000027	praveen	reddy	9000.0000	INDIA
10000028	praveen	reddy	9000.0000	INDIA
10000029	praveen	reddy	9000.0000	INDIA
10000030	praveen	reddy	9000.0000	INDIA
10000031	praveen	reddy	9000.0000	INDIA
10000032	praveen	reddy	9000.0000	INDIA
10000033	praveen	reddy	9000.0000	INDIA
10000034	praveen	reddy	9000.0000	INDIA
10000035	praveen	reddy	9000.0000	INDIA
10000036	praveen	reddy	9000.0000	INDIA
10000037	praveen	reddy	9000.0000	INDIA
10000038	praveen	reddy	9000.0000	INDIA
10000039	praveen	reddy	9000.0000	INDIA
10000040	praveen	reddy	9000.0000	INDIA
10000041	praveen	reddy	9000.0000	INDIA
10000042	praveen	reddy	9000.0000	INDIA
10000043	praveen	reddy	9000.0000	INDIA
10000044	praveen	reddy	9000.0000	INDIA
10000045	praveen	reddy	9000.0000	INDIA
10000046	praveen	reddy	9000.0000	INDIA
10000047	praveen	reddy	9000.0000	INDIA
10000048	praveen	reddy	9000.0000	INDIA
10000049	praveen	reddy	9000.0000	INDIA
10000050	praveen	reddy	9000.0000	INDIA
10000051	praveen	reddy	9000.0000	INDIA
10000052	praveen	reddy	9000.0000	INDIA
10000053	praveen	reddy	9000.0000	INDIA
10000054	praveen	reddy	9000.0000	INDIA
10000055	praveen	reddy	9000.0000	INDIA
10000056	praveen	reddy	9000.0000	INDIA
10000057	praveen	reddy	9000.0000	INDIA
10000058	praveen	reddy	9000.0000	INDIA
10000059	praveen	reddy	9000.0000	INDIA
10000060	praveen	reddy	9000.0000	INDIA
10000061	praveen	reddy	9000.0000	INDIA
10000062	praveen	reddy	9000.0000	INDIA
10000063	praveen	reddy	9000.0000	INDIA
10000064	praveen	reddy	9000.0000	INDIA
10000065	praveen	reddy	9000.0000	INDIA
10000066	praveen	reddy	9000.0000	INDIA
10000067	praveen	reddy	9000.0000	INDIA
10000068	praveen	reddy	9000.0000	INDIA
10000069	praveen	reddy	9000.0000	INDIA
10000070	praveen	reddy	9000.0000	INDIA
10000071	praveen	reddy	9000.0000	INDIA
10000072	praveen	reddy	9000.0000	INDIA
10000073	praveen	reddy	9000.0000	INDIA
10000074	praveen	reddy	9000.0000	INDIA
10000075	praveen	reddy	9000.0000	INDIA
10000076	praveen	reddy	9000.0000	INDIA
10000077	praveen	reddy	9000.0000	INDIA
10000078	praveen	reddy	9000.0000	INDIA
10000079	praveen	reddy	9000.0000	INDIA
10000080	praveen	reddy	9000.0000	INDIA
10000081	praveen	reddy	9000.0000	INDIA
10000082	praveen	reddy	9000.0000	INDIA
10000083	praveen	reddy	9000.0000	INDIA
10000084	praveen	reddy	9000.0000	INDIA
10000085	praveen	reddy	9000.0000	INDIA
10000086	praveen	reddy	9000.0000	INDIA
10000087	praveen	reddy	9000.0000	INDIA
10000088	praveen	reddy	9000.0000	INDIA
10000089	praveen	reddy	9000.0000	INDIA
10000090	praveen	reddy	9000.0000	INDIA
10000091	praveen	reddy	9000.0000	INDIA
10000092	praveen	reddy	9000.0000	INDIA
10000093	praveen	reddy	9000.0000	INDIA
10000094	praveen	reddy	9000.0000	INDIA
10000095	praveen	reddy	9000.0000	INDIA
10000096	praveen	reddy	9000.0000	INDIA
10000097	praveen	reddy	9000.0000	INDIA
10000098	praveen	reddy	9000.0000	INDIA
10000099	praveen	reddy	9000.0000	INDIA
10000100	praveen	reddy	9000.0000	INDIA
10000101	praveen	reddy	9000.0000	INDIA
10000102	praveen	reddy	9000.0000	INDIA
10000103	praveen	reddy	9000.0000	INDIA
10000104	praveen	reddy	9000.0000	INDIA
10000105	praveen	reddy	9000.0000	INDIA
10000106	praveen	reddy	9000.0000	INDIA
10000107	praveen	reddy	9000.0000	INDIA
10000108	praveen	reddy	9000.0000	INDIA
10000109	praveen	reddy	9000.0000	INDIA
10000110	praveen	reddy	9000.0000	INDIA
10000111	praveen	reddy	9000.0000	INDIA
10000112	praveen	reddy	9000.0000	INDIA
10000113	praveen	reddy	9000.0000	INDIA
10000114	praveen	reddy	9000.0000	INDIA
10000115	praveen	reddy	9000.0000	INDIA
10000116	praveen	reddy	9000.0000	INDIA
10000117	praveen	reddy	9000.0000	INDIA
10000118	praveen	reddy	9000.0000	INDIA
10000119	praveen	reddy	9000.0000	INDIA
10000120	praveen	reddy	9000.0000	INDIA
10000121	praveen	reddy	9000.0000	INDIA
10000122	praveen	reddy	9000.0000	INDIA
10000123	praveen	reddy	9000.0000	INDIA
10000124	praveen	reddy	9000.0000	INDIA
10000125	praveen	reddy	9000.0000	INDIA
10000126	praveen	reddy	9000.0000	INDIA
10000127	praveen	reddy	9000.0000	INDIA
10000128	praveen	reddy	9000.0000	INDIA
10000129	praveen	reddy	9000.0000	INDIA
10000130	praveen	reddy	9000.0000	INDIA
10000131	praveen	reddy	9000.0000	INDIA
10000132	praveen	reddy	9000.0000	INDIA
10000133	praveen	reddy	9000.0000	INDIA
10000134	praveen	reddy	9000.0000	INDIA
10000135	praveen	reddy	9000.0000	INDIA
10000136	praveen	reddy	9000.0000	INDIA
10000137	praveen	reddy	9000.0000	INDIA
10000138	praveen	reddy	9000.0000	INDIA
10000139	praveen	reddy	9000.0000	INDIA
10000140	praveen	reddy	9000.0000	INDIA
10000141	praveen	reddy	9000.0000	INDIA
10000142	praveen	reddy	9000.0000	INDIA
10000143	praveen	reddy	9000.0000	INDIA
10000144	praveen	reddy	9000.0000	INDIA
10000145	praveen	reddy	9000.0000	INDIA
10000146	praveen	reddy	9000.0000	INDIA
10000147	praveen	reddy	9000.0000	INDIA
10000148	praveen	reddy	9000.0000	INDIA
10000149	praveen	reddy	9000.0000	INDIA
10000150	praveen	reddy	9000.0000	INDIA
10000151	praveen	reddy	9000.0000	INDIA
10000152	praveen	reddy	9000.0000	INDIA
10000153	praveen	reddy	9000.0000	INDIA
10000154	praveen	reddy	9000.0000	INDIA
10000155	praveen	reddy	9000.0000	INDIA
10000156	praveen	reddy	9000.0000	INDIA
10000157	praveen	reddy	9000.0000	INDIA
10000158	praveen	reddy	9000.0000	INDIA
10000159	praveen	reddy	9000.0000	INDIA
10000160	praveen	reddy	9000.0000	INDIA
10000161	praveen	reddy	9000.0000	INDIA
10000162	praveen	reddy	9000.0000	INDIA
10000163	praveen	reddy	9000.0000	INDIA
10000164	praveen	reddy	9000.0000	INDIA
10000165	praveen	reddy	9000.0000	INDIA
10000166	praveen	reddy	9000.0000	INDIA
10000167	praveen	reddy	9000.0000	INDIA
10000168	praveen	reddy	9000.0000	INDIA
10000169	praveen	reddy	9000.0000	INDIA
10000170	praveen	reddy	9000.0000	INDIA
10000171	praveen	reddy	9000.0000	INDIA
10000172	praveen	reddy	9000.0000	INDIA
10000173	praveen	reddy	9000.0000	INDIA
10000174	praveen	reddy	9000.0000	INDIA
10000175	praveen	reddy	9000.0000	INDIA
10000176	praveen	reddy	9000.0000	INDIA
10000177	praveen	reddy	9000.0000	INDIA
10000178	praveen	reddy	9000.0000	INDIA
10000179	praveen	reddy	9000.0000	INDIA
10000180	praveen	reddy	9000.0000	INDIA
10000181	praveen	reddy	9000.0000	INDIA
10000182	praveen	reddy	9000.0000	INDIA
10000183	praveen	reddy	9000.0000	INDIA
10000184	praveen	reddy	9000.0000	INDIA
10000185	praveen	reddy	9000.0000	INDIA
10000186	praveen	reddy	9000.0000	INDIA
10000187	praveen	reddy	9000.0000	INDIA
10000188	praveen	reddy	9000.0000	INDIA
10000189	praveen	reddy	9000.0000	INDIA
10000190	praveen	reddy	9000.0000	INDIA
10000191	praveen	reddy	9000.0000	INDIA
10000192	praveen	reddy	9000.0000	INDIA
10000193	praveen	reddy	9000.0000	INDIA
10000194	praveen	reddy	9000.0000	INDIA
10000195	praveen	reddy	9000.0000	INDIA
10000196	praveen	reddy	9000.0000	INDIA
10000197	praveen	reddy	9000.0000	INDIA
10000198	praveen	reddy	9000.0000	INDIA
10000199	praveen	reddy	9000.0000	INDIA
10000200	praveen	reddy	9000.0000	