DQL - Tasks database TechCompany (HR management)

```
CREATE DATABASE TechCompany;

USE TechCompany;
```

Table Countries

```
CREATE TABLE Countries (
        Country_ID CHAR(2) NOT NULL,
    Country_Name VARCHAR(40),
    Region_ID INT,
    CONSTRAINT PK_Countries PRIMARY KEY (Country_ID)
    );

INSERT INTO Countries
VALUES ("US", "United States", 2),
        ("IT", "Italy", 1),
    ("BR", "Brazil", 3),
    ("UK", "United Kingdom", 4);

SELECT *
FROM Countries
```

Country_ID	Country_Name	Region_ID
BR	Brazil	3
IT	Italy	1
UK	United Kingdom	4
US	United States	2
NULL	NULL	NULL

Table Locations

```
CREATE TABLE Locations (
        Location_ID INT NOT NULL,
    Street_Address VARCHAR(40),
    Postal_Code VARCHAR(12),
    City VARCHAR(30) NOT NULL,
    State_Province VARCHAR(25),
    Country_ID CHAR(2),
    CONSTRAINT PK_Locations PRIMARY KEY (Location_ID),
    CONSTRAINT FK_Locations_Countries FOREIGN KEY (Country_ID)
        REFERENCES Countries (Country_ID)
    );

INSERT INTO Locations
VALUES (10, "123 Main Street", "NW12345", "New York", "New York", "US"),
        (20, "456 Elm Avenue", "SW67890", "Los Angeles", "California", "US"),
    (30, "234 Rua das Abobrinhas", "179-8787", "Niteroi", "Rio de Janeiro", "BR"),
    (40, "223 Oxford Street", "SW1A 0AA", "London", "Greater London", "UK");
```

```
SELECT *
FROM Locations
```

Location_ID	Street_Address	Postal_Code	City	State_Province	Country_ID
10	123 Main Street	NW12345	New York	New York	US
20	456 Elm Avenue	SW67890	Los Angeles	California	US
30	234 Rua das Abobrinhas	179-8787	Niteroi	Rio de Janeiro	BR
40	223 Oxford Street	SW1A 0AA	London	Greater London	UK
NULL	NULL	NULL	NULL	NULL	NULL

Table Departments

```
CREATE TABLE Departments (
        Department_ID INT NOT NULL,
    Department_Name VARCHAR(30),
    Manager_ID INT,
    Location_ID INT,
    CONSTRAINT PK_Departments PRIMARY KEY (Department_ID),
    CONSTRAINT FK_Departments_Locations FOREIGN KEY (Location_ID)
        REFERENCES Locations (Location_ID)
    );

INSERT INTO Departments
VALUES (1, "Finance", 1, 10),
        (2, "HR", 2, 20),
    (3, "IT", 3, 30),
    (4, "Sales", 3, 40),
    (5, "Marketing", 2, 40);

SELECT *
FROM Departments
```

Department_ID	Department_Na...	Manager_ID	Location_ID
1	Finance	1	10
2	HR	2	20
3	IT	3	30
4	Sales	3	40
5	Marketing	2	40
NULL	NULL	NULL	NULL

Table Jobs

```
CREATE TABLE Jobs (
        Job_ID VARCHAR(10) NOT NULL,
    Job_Title VARCHAR(35) NOT NULL,
    Min_Salary INT,
    Max_Salary INT,
    CONSTRAINT PK_Jobs PRIMARY KEY (Job_ID)
    );

INSERT INTO Jobs
VALUES
    ('J001-IT', 'Software Engineer', 50000, 100000),
```

```
        ('J002-MK', 'Marketing Specialist', 40000, 80000),
        ('J003-FN', 'Accountant', 45000, 90000),
        ('J004-SL', 'Project Manager', 60000, 120000),
        ('J005-MK', 'Graphic Designer', 35000, 70000),
        ('J006-SL', 'Sales Representative', 30000, 60000),
        ('J007-HR', 'Human Resources Manager', 55000, 110000),
        ('J008-FN', 'Financial Analyst', 48000, 96000),
        ('J009-SL', 'Operations Manager', 60000, 120000),
        ('J010-SL', 'Customer Service Representative', 25000, 50000);

SELECT *
FROM Jobs
```

Job_ID	Job_Title	Min_Salary	Max_Salary
J001-IT	Software Engineer	50000	100000
J002-MK	Marketing Specialist	40000	80000
J003-FN	Accountant	45000	90000
J004-SL	Project Manager	60000	120000
J005-MK	Graphic Designer	35000	70000
J006-SL	Sales Representative	30000	60000
J007-HR	Human Resources Manager	55000	110000
J008-FN	Financial Analyst	48000	96000
J009-SL	Operations Manager	60000	120000
J010-SL	Customer Service Representative	25000	50000
NULL	NULL	NULL	NULL

Table Employee

```
CREATE TABLE Employee (
        Employee_ID INT NOT NULL,
    First_Name VARCHAR(20),
    Last_Name VARCHAR(25) NOT NULL,
    Email VARCHAR(20) NOT NULL,
    Phone_Number VARCHAR(20),
    Hire_Date DATE NOT NULL,
    Job_ID VARCHAR(10) NOT NULL,
    Salary DOUBLE,
    Commission_PCT DOUBLE,
    Manager_ID INT,
    Department_ID INT NOT NULL,
    CONSTRAINT PK_Employee PRIMARY KEY (Employee_ID),
    CONSTRAINT FK_Employee_Departments FOREIGN KEY (Department_ID)
        REFERENCES Departments (Department_ID),
        CONSTRAINT FK_Employee_Jobs FOREIGN KEY (Job_ID)
        REFERENCES Jobs (Job_ID)
    );


INSERT INTO Employee
VALUES
        (1, 'Meryl', 'Streep', 'meryl.s@ex.com', '123-456-7890', '2020-01-01', 'J001-IT',
51000.00, NULL, NULL, 1),
        (2, 'Cate', 'Blanchett', 'cate.b@ex.com', '987-654-3210', '2019-05-15', 'J002-MK',
63000.00, 0.05, 1, 2),
```

```
        (3, 'Julianne', 'Moore', 'julianne.m@ex.com', '555-555-5555', '2022-03-10', 'J003-FN',
75000.00, NULL, 1, 1),
        (4, 'Jennifer', 'Lawrence', 'jennifer.l@ex.com', '111-222-3333', '2021-09-30', 'J004-
SL', 65000.00, NULL, 2, 2),
        (5, 'Emma', 'Stone', 'emma.s@ex.com', '444-444-4444', '2023-01-05', 'J004-SL',
80000.00, 0.08, 3, 1),
        (6, 'Natalie', 'Portman', 'natalie.p@ex.com', '777-888-9999', '2022-07-20', 'J002-MK',
60000.00, 0.03, 3, 3),
        (7, 'Frances', 'McDormand', 'frances.m@ex.com', '666-666-6666', '2023-03-01', 'J002-
MK', 70000.00, NULL, 2, 2),
        (8, 'Anne', 'Hathaway', 'anne.h@ex.com', '999-999-9999', '2021-11-11', 'J004-SL',
105000.00, 0.02, 3, 3),
        (9, 'Viola', 'Davis', 'viola.d@ex.com', '222-333-4444', '2023-12-04', 'J002-MK',
50400.00, 150.90, 3, 5),
        (10, 'Lila', 'Love', 'lila.l@ex.com', '07406080799', '2020-06-10', 'J004-SL', 65500.00,
500.50, 2, 4);

SELECT *
FROM Employee
```

Employee_ID	First_Name	Last_Name	Email	Phone_Number	Hire_Date	Job_ID	Salary	Commission_PCT	Manager_ID	Department_ID
1	Meryl	Streep	meryl.s@ex.com	123-456-7890	2020-01-01	J001-IT	51000	NULL	NULL	1
2	Cate	Blanchett	cate.b@ex.com	987-654-3210	2019-05-15	J002-MK	63000	0.05	1	2
3	Julianne	Moore	julianne.m@ex.com	555-555-5555	2022-03-10	J003-FN	75000	NULL	1	1
4	Jennifer	Lawrence	jennifer.l@ex.com	111-222-3333	2021-09-30	J004-SL	65000	NULL	2	2
5	Emma	Stone	emma.s@ex.com	444-444-4444	2023-01-05	J004-SL	80000	0.08	3	1
6	Natalie	Portman	natalie.p@ex.com	777-888-9999	2022-07-20	J002-MK	60000	0.03	3	3
7	Frances	McDormand	frances.m@ex.com	666-666-6666	2023-03-01	J002-MK	70000	NULL	2	2
8	Anne	Hathaway	anne.h@ex.com	999-999-9999	2021-11-11	J004-SL	105000	0.02	3	3
9	Viola	Davis	viola.d@ex.com	222-333-4444	2023-12-04	J002-MK	50400	150.9	3	5
10	Lila	Love	lila.l@ex.com	07406080799	2020-06-10	J004-SL	65500	500.5	2	4
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table Job_History

```
CREATE TABLE Job_History (
        Employee_ID INT NOT NULL,
    Start_Date DATE NOT NULL,
    End_Date DATE NOT NULL,
    Job_ID VARCHAR(10) NOT NULL,
    Department_ID INT,
    CONSTRAINT FK_JH_Employee FOREIGN KEY (Employee_ID)
                REFERENCES Employee (Employee_ID),
    CONSTRAINT FK_JH_Departments FOREIGN KEY (Department_ID)
        REFERENCES Departments (Department_ID),
        CONSTRAINT FK_JH_Jobs FOREIGN KEY (Job_ID)
        REFERENCES Jobs (Job_ID),
        CONSTRAINT PK_Job_History PRIMARY KEY (Employee_ID, Job_ID, Department_ID)
    );

INSERT INTO Job_History
VALUES (1, '2020-01-01', '2005-03-03', 'J001-IT', 1),
        (4, '2021-09-30', '2022-09-30', 'J002-MK', 5),
    (6, '2022-07-20', '2022-12-09', 'J002-MK', 3),
    (8, '2021-11-11', '2022-08-01', 'J004-SL', 3);

SELECT *
FROM Job_History;
```

Employee_ID	Start_Date	End_Date	Job_ID	Department_ID
1	2020-01-01	2005-03-03	J001-IT	1
4	2021-09-30	2022-09-30	J002-MK	5
6	2022-07-20	2022-12-09	J002-MK	3
8	2021-11-11	2022-08-01	J004-SL	3
NULL	NULL	NULL	NULL	NULL

Tasks

1. SELECT fields and use Alias

The HR department wants a query to display the last name, job code, hire date, and employee number for each employee, with employee number appearing first. Provide an alias STARTDATE for the HIRE_DATE column.

```
SELECT
Employee_ID,
Last_Name,
Job_ID,
Hire_Date "STARTDATE"
FROM Employee;
```

Employee_ID	Last_Name	Job_ID	STARTDATE
1	Streep	J001-IT	2020-01-01
2	Blanchett	J002-MK	2019-05-15
3	Moore	J003-FN	2022-03-10
4	Lawrence	J004-SL	2021-09-30
5	Stone	J004-SL	2023-01-05
6	Portman	J002-MK	2022-07-20
7	McDormand	J002-MK	2023-03-01
8	Hathaway	J004-SL	2021-11-11
9	Davis	J002-MK	2023-12-04
10	Love	J004-SL	2020-06-10
NULL	NULL	NULL	NULL

2. Renaming with Alias

The HR department wants more descriptive column headings for its report on employees. Name the column headings Emp#, Employee, Job, and Hire Date, respectively.

```
SELECT
Last_Name "Employee",
Job_ID "Job",
Hire_Date "Hire Date"
FROM Employee;
```

Employee	Job	Hire Date
Streep	J001-IT	2020-01-01
Blanchett	J002-MK	2019-05-15
Moore	J003-FN	2022-03-10
Lawrence	J004-SL	2021-09-30
Stone	J004-SL	2023-01-05
Portman	J002-MK	2022-07-20
McDormand	J002-MK	2023-03-01
Hathaway	J004-SL	2021-11-11
Davis	J002-MK	2023-12-04
Love	J004-SL	2020-06-10

3. Concatenating

The HR department has requested a report of all employees and their job IDs. Display the last name concatenated with the job ID (separated by a comma and space) and name the column Employee and Title.

```
SELECT
CONCAT ("The employee " , Last_Name , " works as " , Job_ID) "Employee and Title"
FROM Employee;
```

Employee and Title
The employee Streep works as J001-IT
The employee Blanchett works as J002-MK
The employee Moore works as J003-FN
The employee Lawrence works as J004-SL
The employee Stone works as J004-SL
The employee Portman works as J002-MK
The employee McDormand works as J002-MK
The employee Hathaway works as J004-SL
The employee Davis works as J002-MK
The employee Love works as J004-SL

4. Display all data

To familiarize yourself with the data in the EMPLOYEE table, create a query to display all the data from that table. Separate each column output by a comma. Name the column title THE_OUTPUT.

```
SELECT
CONCAT (Employee_ID , ", " , First_Name , ", " ,
    Last_Name , ", " , Email , ", " , Phone_Number , ", " ,
    Hire_Date , ", " , Job_ID , ", " ,
    Salary , ", " , Commission_PCT , ", " ,
    Manager_ID , ", " , Department_ID) "THE_OUTPUT"
FROM Employee;
```

THE_OUTPUT
NULL
2, Cate, Blanchett, cate.b@ex.com, 987-654-3210, 2019-05-15, J002-MK, 63000, 0.05, 1, 2
NULL
NULL
5, Emma, Stone, emma.s@ex.com, 444-444-4444, 2023-01-05, J004-SL, 80000, 0.08, 3, 1
6, Natalie, Portman, natalie.p@ex.com, 777-888-9999, 2022-07-20, J002-MK, 60000, 0.03, 3, 3
NULL
8, Anne, Hathaway, anne.h@ex.com, 999-999-9999, 2021-11-11, J004-SL, 105000, 0.02, 3, 3
9, Viola, Davis, viola.d@ex.com, 222-333-4444, 2023-12-04, J002-MK, 50400, 150.9, 3, 5
10, Lila, Love, lila.l@ex.com, 07406080799, 2020-06-10, J004-SL, 65500, 500.5, 2, 4

5. Display data using >

Due to budget issues, the HR department needs a report that displays the last name and salary of employees who earn more than £21,000.

```
SELECT Last_Name, Salary
FROM Employee
WHERE Salary > 21000;
```

Last_Name	Salary
Streep	51000
Blanchett	63000
Moore	75000
Lawrence	65000
Stone	80000
Portman	60000
McDormand	70000
Hathaway	105000
Davis	50400
Love	65500

6. Display Last_Name and Department_ID

Create a report that displays the last name and department ID for employee number 6.

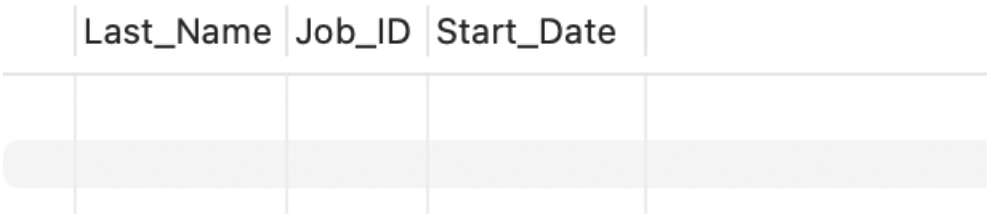
```
SELECT Last_Name, Department_ID
FROM Employee
WHERE Employee_ID = 6;
```

Last_Name	Department_ID
Portman	3

7. Display data using AND

Create a report to display the last name, job ID, and start date for the employees with the last names of Blanchett and Moore.

```
SELECT Last_Name, Job_ID, Hire_Date 'Start_Date'
FROM Employee
WHERE Last_Name = 'Blanchett' AND Last_Name = 'Moore';
```



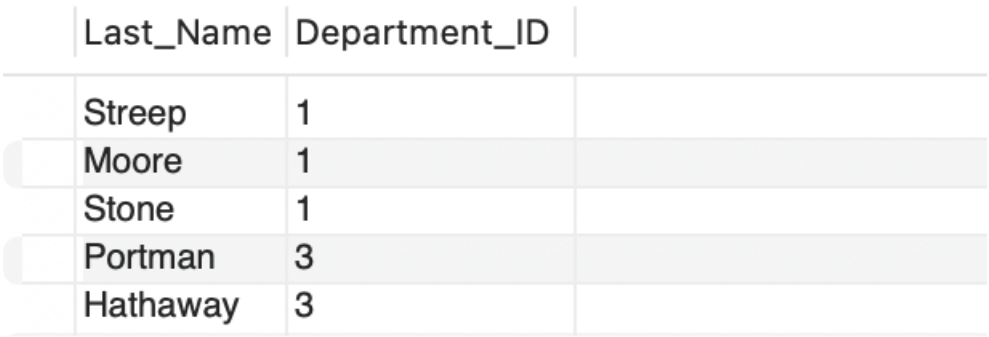
Last_Name	Job_ID	Start_Date

Note: There's none employees with both surnames in this company.

8. Display using IN

Display the last name and department number of all employees in departments 1 or 3.

```
SELECT Last_Name, Department_ID
FROM Employee
WHERE Department_ID = 1 OR Department_ID = 3;
```



Last_Name	Department_ID
Streep	1
Moore	1
Stone	1
Portman	3
Hathaway	3

9. Display using BETWEEN

Display the last name and salary of employees who earn between £40,000 and £80,000 and are in department 1 or 3. Label the columns Employee and Monthly Salary, respectively.

```
SELECT Last_Name, Salary
FROM Employee
WHERE Salary BETWEEN 40000.00 AND 80000.00;
```

Last_Name	Salary
Streep	51000
Blanchett	63000
Moore	75000
Lawrence	65000
Stone	80000
Portman	60000
McDormand	70000
Davis	50400
Love	65500

10. Select with BETWEEN and AND

The HR department needs a report that displays the last name and hire date for all employees who were hired in 2020.

```
SELECT Last_Name, Hire_Date
FROM Employee
WHERE Hire_Date BETWEEN '2020-01-01' AND '2020-12-31';
```

Last_Name	Hire_Date
Streep	2020-01-01
Love	2020-06-10

11. Select with IS NUL

Create a report to display the last name and job title of all employees who do not have a manager.

```
SELECT Last_Name, Job_ID
FROM Employee
WHERE Manager_ID IS NULL;
```

```
Last_Name  |  Job_ID
Streep     |  J001-IT
```

12. Select with IS NOT NULL

Create a report to display the last name, salary, and commission of all employees who earn commissions.

```
SELECT Last_Name, Salary, Commission_PCT
FROM Employee
WHERE Commission_PCT IS NOT NULL;
```

Last_Name	Salary	Commission_PCT
Blanchett	63000	0.05
Stone	80000	0.08
Portman	60000	0.03
Hathaway	105000	0.02
Davis	50400	150.9
Love	65500	500.5

13. Select with COUNT

Write a query display total number of employees working in "IT" Department.

```
SELECT COUNT(*)
FROM Employee
WHERE Department_ID = 3;
```

```
COUNT(*)
2
```

14. Select with MIN

Display the minimum salary among employees working for the IT department.

```
SELECT min(Salary)
FROM Employee
WHERE Department_ID = 3;
```

```
min(Salary)
60000
```

15. Select with COUNT

Display the number of positions (Job_ID) in each department

```
SELECT Department_ID, Job_ID, COUNT(*)
FROM Employee
GROUP BY Department_ID, Job_ID;
```

Department_ID	Job_ID	COUNT(*)
1	J001-IT	1
2	J002-MK	2
1	J003-FN	1
2	J004-SL	1
1	J004-SL	1
3	J002-MK	1
3	J004-SL	1
5	J002-MK	1
4	J004-SL	1

16. Select with AVG

Find the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively.

```
SELECT max(Salary) "Maximum",min(Salary) "Minimum",sum(Salary) "Sum", AVG(Salary) "Average"
FROM Employee
```

Maximum	Minimum	Sum	Average
105000	50400	684900	68490

17. Select with AVG and GROUP BY

Modify the above query to display the minimum, maximum, sum, and average salary for each job type.

```
SELECT Department_ID "Department", max(Salary) "Maximum",min(Salary) "Minimum",sum(Salary)
"Sum", AVG(Salary) "Average"
FROM Employee
GROUP BY Department_ID;
```

Department	Maximum	Minimum	Sum	Average
1	80000	51000	206000	68666.66666666667
2	70000	63000	198000	66000
3	105000	60000	165000	82500
4	65500	65500	65500	65500
5	50400	50400	50400	50400

18. Select GROUP BY

Write a query to display the number of people with the same job.

```
SELECT Job_ID, COUNT(*) "Employees with the same job"
FROM Employee
GROUP BY Job_ID;
```

Job_ID	Employees with the same job
J001-IT	1
J002-MK	4
J003-FN	1
J004-SL	4

19. Select with DISTINCT

Determine the number of managers without listing them.

```
SELECT COUNT(DISTINCT Manager_ID) "Num of Managers"
FROM Employee
WHERE manager_id is not null;
```

```
Num of Managers
3
```

20. Select with max-min

Find the difference between the highest and lowest salaries. Label the column "Difference".

```
SELECT (max(Salary) - min(Salary)) as Difference
FROM Employee;
```

```
Difference
54600
```

21. Select with LIMIT

Display info for the 2 employees with higher salaries

```
SELECT * FROM Employee
ORDER BY Salary DESC
LIMIT 2;
```

Employee_ID	First_Name	Last_Name	Email	Phone_Number	Hire_Date	Job_ID	Salary	Commission_PCT	Manager_ID	Department_ID
8	Anne	Hathaway	anne.h@ex.com	999-999-9999	2021-11-11	J004-SL	105000	0.02	3	3
5	Emma	Stone	emma.s@ex.com	444-444-4444	2023-01-05	J004-SL	80000	0.08	3	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

22. Select with JOIN and ON

Write a query for the HR department to produce the addresses of all the departments. Use the LOCATIONS and COUNTRIES tables. Show the location ID, street address, city, state or province, and country in the output.

```
SELECT Department_Name, D.Location_ID, Street_Address, City, State_Province, C.Country_Name
FROM Departments D
INNER JOIN Locations L
ON D.Location_ID = L.Location_ID
```

```
INNER JOIN Countries C
ON L.Country_ID = C.Country_ID;
```

Department_Name	Location_ID	Street_Address	City	State_Province	Country_Name
Finance	10	123 Main Street	New York	New York	United States
HR	20	456 Elm Avenue	Los Angeles	California	United States
IT	30	234 Rua das Abobrinhas	Niteroi	Rio de Janeiro	Brazil
Sales	40	223 Oxford Street	London	Greater London	United Kingdom
Marketing	40	223 Oxford Street	London	Greater London	United Kingdom

23. Select with two tables

Write a query to display the last name, department number, and department name for all employees.

```
SELECT Last_Name, E.Department_ID, D.Department_Name
FROM Employee E
INNER JOIN Departments D
ON E.Department_ID = D.Department_ID
ORDER BY Department_Name;
```

Last_Name	Department_ID	Department_Name
Streep	1	Finance
Moore	1	Finance
Stone	1	Finance
Blanchett	2	HR
Lawrence	2	HR
McDormand	2	HR
Portman	3	IT
Hathaway	3	IT
Davis	5	Marketing
Love	4	Sales

24. Select with multiple tables

The HR department needs a report of employees in Toronto. Display the last name, job, department number, and department name for all employees who work in London.

```
SELECT Last_Name, Job_ID, D.Department_ID, D.Department_Name
FROM Employee E
INNER JOIN Departments D
on E.Department_ID = D.Department_ID
INNER JOIN Locations L
on D.Location_ID = L.Location_ID
WHERE L.City = "London";
```

Last_Name	Job_ID	Department_ID	Department_Name
Love	J004-SL	4	Sales
Davis	J002-MK	5	Marketing

25. Select with SubQuery

Write a query that displays the employee number and last name of all employees who work **in a department** with any employee whose last name contains a *e*.

```
SELECT Employee_ID, Last_Name
FROM Employee
WHERE Department_ID = 1
        AND Department_ID IN (
    SELECT Department_ID
    FROM Employee
    WHERE Last_Name LIKE "%e%"
    );
-- option without specifying the department
SELECT Employee_ID, Last_Name
FROM Employee
WHERE Department_ID IN (
    SELECT Department_ID
    FROM Employee
    WHERE Last_Name LIKE "%e%"
    );
```

Employee_ID	Last_Name
1	Streep
3	Moore
5	Stone
NULL	NULL

Result without specifying the department:

Employee_ID	Last_Name
1	Streep
3	Moore
5	Stone
2	Blanchett
4	Lawrence
7	McDormand
10	Love
NULL	NULL

26. Select with SubQuery

The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 10.

```
SELECT Last_Name, Department_ID, Job_ID
FROM Employee
WHERE Department_ID IN (
        SELECT Department_ID
    FROM Departments
    WHERE Location_ID = 10
    );
```

Last_Name	Department_ID	Job_ID
Streep	1	J001-IT
Moore	1	J003-FN
Stone	1	J004-SL

27. Select employee by manager id

Create a report for HR that displays the last name and salary of every employee **who reports** to Manager 2.

```
SELECT Last_Name, Salary
FROM Employee
WHERE Manager_ID = 2;
```

Last_Name	Salary
Lawrence	65000
McDormand	70000
Love	65500

28. Select with SubQuery and ORDER BY

Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary.

```
SELECT Employee_ID, Last_Name, Salary
FROM employee
WHERE Salary > (
        SELECT AVG(Salary)
    FROM Employee
    )
        ORDER BY Salary ASC;
```

Employee_ID	Last_Name	Salary
7	McDormand	70000
3	Moore	75000
5	Stone	80000
8	Hathaway	105000
NULL	NULL	NULL

29. Create VIEW

The staff in the HR department wants to hide some of the data in the EMPLOYEES table. They want a view called EMPLOYEES_VU based on the employee numbers, employee names, and department numbers from the EMPLOYEES table. They want the heading for the employee name to be EMPLOYEE.

```
CREATE VIEW EMPLOYEE_VU
AS
SELECT Employee_ID, First_Name "Employee" , Department_ID
FROM Employee;
```

Employee_ID	Employee	Department_ID
1	Meryl	1
2	Cate	2
3	Julianne	1
4	Jennifer	2
5	Emma	1
6	Natalie	3
7	Frances	2
8	Anne	3

30. Select with VIEW

Using your EMPLOYEES_VU view, write a query for the HR department to display all employee names and department numbers.

```
SELECT Employee, Department_ID
FROM EMPLOYEE_VU;
```

Employee	Department_ID
Meryl	1
Cate	2
Julianne	1
Jennifer	2
Emma	1
Natalie	3
Frances	2
Anne	3
Viola	5
Lila	4

31. Create VIEW filtering by department

Department 1 needs access to its employee data. Create a view named DEPT1 that contains the employee numbers, employee last names, and department numbers for all employees in department 1. You have been asked to label the view columns EMPNO, EMPLOYEE, and DEPTNO.

Syntax 1:

```
CREATE VIEW DEPT1
AS
SELECT Employee_ID "EmpNo", Last_Name "Employee", Department_ID "DeptNo"
FROM Employee
WHERE Department_ID = 1;

SELECT * FROM DEPT1;
```

Syntax 2:

```
CREATE VIEW DEPT1 (EmpNo, Employee, DeptNo)
AS
SELECT Employee_ID, Last_Name, Department_ID
FROM Employee
WHERE Department_ID = 1;

SELECT * FROM DEPT1;
```

The same result:

EmpNo	Employee	DeptNo
1	Streep	1
3	Moore	1
5	Stone	1