Create a Database name entri_assignment

```
CREATE DATABASE entri_assignment; # Create a Database name entri_assignment
USE entri_assignment; # Use database entri_assignment
```

Create a Table with name departments Department_id (pk) Department_name Location_id+

```
CREATE TABLE departments
(
Department_id INT NOT NULL PRIMARY KEY,
Department_name VARCHAR(30),
Location_id INT
);
```

Create a Table with name employees

Employee_id (pk) ,first_name,last_name ,email,phone_number,hire_date,
job_id, salary, commission_pct, manager_id, department_id (fk reference)

```
CREATE TABLE employees
(
Employee_id INT NOT NULL PRIMARY KEY,
first_name VARCHAR(30) NOT NULL,
last_name VARCHAR(30) NOT NULL,
email VARCHAR(100),
phone_number VARCHAR(12),
hire_date DATE,
job_id VARCHAR(30),
salary DECIMAL(10,2),
commission_pct DECIMAL(10,2),
manager_id INT,
department_id INT NOT NULL,
FOREIGN KEY (department_id) REFERENCES departments(Department_id)
);
```

```
# Insert into Departments table
INSERT INTO departments VALUES ( 20, 'Marketing', 180);
INSERT INTO departments VALUES ( 30, 'Purchasing', 1700);
INSERT INTO departments VALUES ( 40,'Human Resources', 2400);
INSERT INTO departments VALUES ( 50, 'Shipping', 1500);
INSERT INTO departments VALUES ( 60, 'IT', 1400);
INSERT INTO departments VALUES ( 70, 'Public Relations', 2700);
INSERT INTO departments VALUES ( 80, 'Sales', 2500 );
INSERT INTO departments VALUES ( 90, 'Executive', 1700);
INSERT INTO departments VALUES ( 100, 'Finance', 1700);
INSERT INTO departments VALUES ( 110, 'Accounting', 1700);
INSERT INTO departments VALUES ( 120, 'Treasury', 1700);
INSERT INTO departments VALUES ( 130, 'Corporate Tax', 1700 );
INSERT INTO departments VALUES ( 140, 'Control And Credit', 1700);
INSERT INTO departments VALUES ( 150, 'Shareholder Services', 1700);
INSERT INTO departments VALUES ( 160, 'Benefits', 1700);
INSERT INTO departments VALUES ( 170, 'Payroll', 1700);
```

TABLE departments :

```
mysql> select * from departments;
+---------------+----------------------+-------------+
| Department_id | Department_name      | Location_id |
+---------------+----------------------+-------------+
|            20 | Marketing            |         180 |
|            30 | Purchasing           |        1700 |
|            40 | Human Resources      |        2400 |
|            50 | Shipping             |        1500 |
|            60 | IT                   |        1400 |
|            70 | Public Relations     |        2700 |
|            80 | Sales                |        2500 |
|            90 | Executive            |        1700 |
|           100 | Finance              |        1700 |
|           110 | Accounting           |        1700 |
|           120 | Treasury             |        1700 |
|           130 | Corporate Tax        |        1700 |
|           140 | Control And Credit   |        1700 |
|           150 | Shareholder Services |        1700 |
|           160 | Benefits             |        1700 |
|           170 | Payroll              |        1700 |
+---------------+----------------------+-------------+
16 rows in set (0.00 sec)
```

```
# Insertinto employees table
INSERT INTO employees VALUES (100, 'Steven', 'King', 'SKING', '515.123.4567', '1987-06-17' , 'AD_PRES', 24000 , NULL, NULL, 20);
INSERT INTO employees VALUES (101, 'Neena' , 'Kochhar' , 'NKOCHHAR' , '515.123.4568' , '1989-11-21' , 'AD_VP' , 17000 , NULL , 100 , 20);
INSERT INTO employees VALUES (102 , 'Lex' , 'De Haan' , 'LDEHAAN' , '515.123.4569' , '1993-09-12' , 'AD_VP' , 17000 , NULL , 100 , 30);
INSERT INTO employees VALUES (104 , 'Bruce' , 'Ernst' , 'BERNST' , '590.423.4568' , '1991-05-21', 'IT_PROG' , 6000 , NULL , 103 , 60);
INSERT INTO employees VALUES (105 , 'David' , 'Austin' , 'DAUSTIN' , '590.423.4569' , '1997-06-25', 'IT_PROG' , 4800 , NULL , 103 , 60);
INSERT INTO employees VALUES (106 , 'Valli' , 'Pataballa' , 'VPATABAL' , '590.423.4560' , '1998-02-05', 'IT_PROG' , 4800 , NULL , 103 , 40);
INSERT INTO employees VALUES (107 , 'Diana' , 'Lorentz' , 'DLORENTZ' , '590.423.5567' , '1999-02-09', 'IT_PROG' , 4200 , NULL , 103 , 40);
INSERT INTO employees VALUES (108 , 'Nancy' , 'Greenberg' , 'NGREENBE' , '515.124.4569' , '1994-08-17', 'FI_MGR' , 12000 , NULL , 101 , 100);
INSERT INTO employees VALUES (109 , 'Daniel' , 'Faviet' , 'DFAVIET' , '515.124.4169' , '1994-08-12', 'FI_ACCOUNT' , 9000 , NULL , 108 , 170);
INSERT INTO employees VALUES (110 , 'John' , 'Chen' , 'JCHEN' , '515.124.4269' , '1997-04-09', 'FI_ACCOUNT' , 8200 , NULL , 108 , 170);
INSERT INTO employees VALUES (111 , 'Ismael' , 'Sciarra' , 'ISCIARRA' , '515.124.4369' , '1997-02-01', 'FI_ACCOUNT' , 7700 , NULL, 108, 160);
INSERT INTO employees VALUES (112 , 'Jose Manuel' , 'Urman' , 'JMURMAN' , '515.124.4469' , '1998-06-03', 'FI_ACCOUNT' , 7800 , NULL, 108, 150);
INSERT INTO employees VALUES (114 , 'Den' , 'Raphaely' , 'DRAPHEAL' , '515.127.4561' , '1994-11-08', 'PU_MAN' , 11000 , NULL , 100 , 30);
INSERT INTO employees VALUES (115 , 'Alexander' , 'Khoo' , 'AKHOO' , '515.127.4562' , '1995-05-12', 'PU_CLERK' , 3100 , NULL , 114 , 80);
INSERT INTO employees VALUES (116 , 'Shelli' , 'Baida' , 'SBAIDA' , '515.127.4563' ,'1997-12-13', 'PU_CLERK' , 2900 , NULL , 114 , 70);
INSERT INTO employees VALUES (117 , 'Sigal' , 'Tobias' , 'STOBIAS' , '515.127.4564' , '1997-09-10', 'PU_CLERK' , 2800 , NULL , 114 , 30);
INSERT INTO employees VALUES (118 , 'Guy' , 'Himuro' , 'GHIMURO' , '515.127.4565' , '1998-01-02', 'PU_CLERK' , 2600 , NULL , 114 , 60);
INSERT INTO employees VALUES (119 , 'Karen' , 'Colmenares' , 'KCOLMENA' , '515.127.4566' , '1999-04-08', 'PU_CLERK' , 2500 , NULL , 114 , 130);
INSERT INTO employees VALUES (120 , 'Matthew' , 'Weiss' , 'MWEISS' , '650.123.1234' ,'1996-07-18', 'ST_MAN' , 8000 , NULL , 100 , 50);
INSERT INTO employees VALUES (122 , 'Payam' , 'Kaufling' , 'PKAUFLIN' , '650.123.3234' ,'1995-05-01', 'ST_MAN' , 7900 , NULL , 100 , 40);
INSERT INTO employees VALUES (123 , 'Shanta' , 'Vollman' , 'SVOLLMAN' , '650.123.4234' , '1997-10-12', 'ST_MAN' , 6500 , NULL , 100 , 50);
INSERT INTO employees VALUES (124, 'Kevin' , 'Mourgos' , 'KMOURGOS' , '650.123.5234' , '1999-11-12', 'ST_MAN' , 5800 , NULL , 100 , 80);
INSERT INTO employees VALUES (125, 'Julia' , 'Nayer' , 'JNAYER' , '650.124.1214' , '1997-07-02', 'ST_CLERK' , 3200 , NULL , 120 , 50);
INSERT INTO employees VALUES (126, 'Irene' , 'Mikkilineni' , 'IMIKKILI' , '650.124.1224' , '1998-11-12', 'ST_CLERK' , 2700 , NULL , 120 , 50);
INSERT INTO employees VALUES (127, 'James' , 'Landry' , 'JLANDRY' , '650.124.1334' , '1999-01-02' , 'ST_CLERK' , 2400 , NULL , 120 , 90);
INSERT INTO employees VALUES (128, 'Steven' , 'Markle' , 'SMARKLE' , '650.124.1434' , '2000-03-04' , 'ST_CLERK' , 2200 , NULL , 120 , 50);
INSERT INTO employees VALUES (130, 'Mozhe' , 'Atkinson' , 'MATKINSO' , '650.124.6234' , '1997-10-12' , 'ST_CLERK' , 2800 , NULL , 121 , 110);
```

TABLE employees:

```
mysql> select * from employees
    -> ;
+-------------+------------+-------------+----------+--------------+------------+------------+---------+----------------+------------+---------------+
| Employee_id | first_name | last_name   | email    | phone_number | hire_date  | job_id     | salary  | commission_pct | manager_id | department_id |
+-------------+------------+-------------+----------+--------------+------------+------------+---------+----------------+------------+---------------+
|         100 | Steven     | King        | SKING    | 515.123.4567 | 1987-06-17 | AD_PRES    | 24000.00 |           NULL |       NULL |            20 |
|         101 | Neena      | Kochhar     | NKOCHHAR | 515.123.4568 | 1989-11-21 | AD_VP      | 17000.00 |           NULL |        100 |            20 |
|         102 | Lex        | De Haan     | LDEHAAN  | 515.123.4569 | 1993-09-12 | AD_VP      | 17000.00 |           NULL |        100 |            30 |
|         104 | Bruce      | Ernst       | BERNST   | 590.423.4568 | 1991-05-21 | IT_PROG    |  6000.00 |           NULL |        103 |            60 |
|         105 | David      | Austin      | DAUSTIN  | 590.423.4569 | 1997-06-25 | IT_PROG    |  4800.00 |           NULL |        103 |            60 |
|         106 | Valli      | Pataballa   | VPATABAL | 590.423.4560 | 1998-02-05 | IT_PROG    |  4800.00 |           NULL |        103 |            40 |
|         107 | Diana      | Lorentz     | DLORENTZ | 590.423.5567 | 1999-02-09 | IT_PROG    |  4200.00 |           NULL |        103 |            40 |
|         108 | Nancy      | Greenberg   | NGREENBE | 515.124.4569 | 1994-08-17 | FI_MGR     | 12000.00 |           NULL |        101 |           100 |
|         109 | Daniel     | Faviet      | DFAVIET  | 515.124.4169 | 1994-08-12 | FI_ACCOUNT |  9000.00 |           NULL |        108 |           170 |
|         110 | John       | Chen        | JCHEN    | 515.124.4269 | 1997-04-09 | FI_ACCOUNT |  8200.00 |           NULL |        108 |           170 |
|         111 | Ismael     | Sciarra     | ISCIARRA | 515.124.4369 | 1997-02-01 | FI_ACCOUNT |  7700.00 |           NULL |        108 |           160 |
|         112 | Jose Manuel | Urman      | JMURMAN  | 515.124.4469 | 1998-06-03 | FI_ACCOUNT |  7800.00 |           NULL |        108 |           150 |
|         114 | Den        | Raphaely    | DRAPHEAL | 515.127.4561 | 1994-11-08 | PU_MAN     | 11000.00 |           NULL |        100 |            30 |
|         115 | Alexander  | Khoo        | AKHOO    | 515.127.4562 | 1995-05-12 | PU_CLERK   |  3100.00 |           NULL |        114 |            80 |
|         116 | Shelli     | Baida       | SBAIDA   | 515.127.4563 | 1997-12-13 | PU_CLERK   |  2900.00 |           NULL |        114 |            70 |
|         117 | Sigal      | Tobias      | STOBIAS  | 515.127.4564 | 1997-09-10 | PU_CLERK   |  2800.00 |           NULL |        114 |            30 |
|         118 | Guy        | Himuro      | GHIMURO  | 515.127.4565 | 1998-01-02 | PU_CLERK   |  2600.00 |           NULL |        114 |            60 |
|         119 | Karen      | Colmenares  | KCOLMENA | 515.127.4566 | 1999-04-08 | PU_CLERK   |  2500.00 |           NULL |        114 |           130 |
|         120 | Matthew    | Weiss       | MWEISS   | 650.123.1234 | 1996-07-18 | ST_MAN     |  8000.00 |           NULL |        100 |            50 |
|         122 | Payam      | Kaufling    | PKAUFLIN | 650.123.3234 | 1995-05-01 | ST_MAN     |  7900.00 |           NULL |        100 |            40 |
|         123 | Shanta     | Vollman     | SVOLLMAN | 650.123.4234 | 1997-10-12 | ST_MAN     |  6500.00 |           NULL |        100 |            50 |
|         124 | Kevin      | Mourgos     | KMOURGOS | 650.123.5234 | 1999-11-12 | ST_MAN     |  5800.00 |           NULL |        100 |            80 |
|         125 | Julia      | Nayer       | JNAYER   | 650.124.1214 | 1997-07-02 | ST_CLERK   |  3200.00 |           NULL |        120 |            50 |
|         126 | Irene      | Mikkilineni | IMIKKILI | 650.124.1224 | 1998-11-12 | ST_CLERK   |  2700.00 |           NULL |        120 |            50 |
|         127 | James      | Landry      | JLANDRY  | 650.124.1334 | 1999-01-02 | ST_CLERK   |  2400.00 |           NULL |        120 |            90 |
|         128 | Steven     | Markle      | SMARKLE  | 650.124.1434 | 2000-03-04 | ST_CLERK   |  2200.00 |           NULL |        120 |            50 |
|         130 | Mozhe      | Atkinson    | MATKINSO | 650.124.6234 | 1997-10-12 | ST_CLERK   |  2800.00 |           NULL |        121 |           110 |
+-------------+------------+-------------+----------+--------------+------------+------------+---------+----------------+------------+---------------+
27 rows in set (0.00 sec)
```

Solve SQL Exercises

1. Select employees first name, last name, job_id and salary whose first name starts with alphabet S

A:

```
SELECT

        first_name, last_name, job_id, salary
FROM
        employees
WHERE
        first_name LIKE 'S%';  # starts with the letter 'S' LIKE is used for pattern matching, % symbol represents any sequence of characters.
```

OUTPUT :

```
80
81
82      # 1. Select employees first name, last name, job_id and salary whose first name starts with alphabet S
83      SELECT
84          first_name, last_name, job_id, salary
85      FROM
86          employees
87      WHERE
88          first_name LIKE 'S%';  # starts with the letter 'S' LIKE is used for pattern matching, % symbol represents any sequence of characters.
89
90
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	job_id	salary
Steven	King	AD_PRES	24000.00
Shelli	Baida	PU_CLERK	2900.00
Sigal	Tobias	PU_CLERK	2800.00
Shanta	Vollman	ST_MAN	6500.00
Steven	Markle	ST_CLERK	2200.00

2. Write a query to select employee with the highest salary (using an inner query)

A:

```
SELECT
        *                                                   # '*' symbol represents select all
FROM
        employees                                           # employees table
WHERE                                                       # filters the rows returned by the main (outer) query where salary = maxium
        salary = ( SELECT MAX(salary) FROM employees );     # SELECT MAX(salary) function returns the maximum value of the salary column.
```

OUTPUT:

```
90
91
92      # 2. Write a query to select employee with the highest salary  (using an inner query)
93  •   SELECT
94          *                                               # '*' symbol represents select all
95      FROM
96          employees                                       # employees table
97      WHERE                                               # filters the rows returned by the main (outer) query where salary = maxium
98          salary = ( SELECT MAX(salary) FROM employees ); # SELECT MAX(salary) function returns the maximum value of the salary column.
99
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24000.00	NULL	NULL	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. Select employee with the second highest salary

A:

```
SELECT
        *
FROM
        employees
ORDER BY
        salary DESC                    # DESC sort the result in descending order
        LIMIT 1                        # LIMIT determines how many rows to return after the offset.
        OFFSET 1;                      # OFFSET skips the specified number of rows from the beginning.
```

OUTPUT:

```
101
102     # 3. Select employee with the second highest salary
103 •   SELECT
104         *
105     FROM
106         employees
107     ORDER BY
108         salary DESC                        # DESC sort the result in descending order
109         LIMIT 1                            # LIMIT determines how many rows to return after the offset.
110         OFFSET 1;                          # OFFSET skips the specified number of rows from the beginning.
111
112
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

Employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-11-21	AD_VP	17000.00	NULL	100	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

4. Write a query to select employees and their corresponding managers and their salaries

A:

```
SELECT
	e.Employee_id AS EmployeeID,
	e.first_name AS EmployeeFirstName,
	e.last_name AS EmployeeLastName,
	e.salary AS EmployeeSalary,
	(SELECT m.Employee_id
		FROM employees m
		WHERE m.Employee_id = e.manager_id) AS ManagerID,  # get manager's ID

	(SELECT m.first_name
		FROM employees m
		WHERE m.Employee_id = e.manager_id) AS ManagerFirstName, # get manager's first name

	 (SELECT m.last_name
		FROM employees m
		WHERE m.Employee_id = e.manager_id) AS ManagerLastName, # get manager's last name

	(SELECT m.salary
		FROM employees m
		WHERE m.Employee_id = e.manager_id) AS ManagerSalary # get manager's salary
FROM
	employees e
ORDER BY
	e.Employee_id;
```

OUTPUT:

```
113
114     # 4. Write a query to select employees and their corresponding managers and their salaries
115 •   SELECT
116         e.Employee_id AS EmployeeID,
117         e.first_name AS EmployeeFirstName,
118         e.last_name AS EmployeeLastName,
119         e.salary AS EmployeeSalary,
120
121         (SELECT m.Employee_id  # select the Employee_id from the employees table with the alias m.
122             FROM employees m   # m.Employee_id = e.manager_id retrieves the ID of the manager whose Employee_id matches the manager_id of the employee in the current row of the main query.
123             WHERE m.Employee_id = e.manager_id) AS ManagerID,  # get manager's ID
124
125         (SELECT m.first_name
126             FROM employees m
127             WHERE m.Employee_id = e.manager_id) AS ManagerFirstName, # get manager's first name
128
129         (SELECT m.last_name
130             FROM employees m
131             WHERE m.Employee_id = e.manager_id) AS ManagerLastName, # get manager's last name
132
133         (SELECT m.salary
134             FROM employees m
135             WHERE m.Employee_id = e.manager_id) AS ManagerSalary # get manager's salary
136     FROM
137         employees e
138     ORDER BY
139         e.Employee_id;
140
141
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmployeeID	EmployeeFirstName	EmployeeLastName	EmployeeSalary	ManagerID	ManagerFirstName	ManagerLastName	ManagerSalary
100	Steven	King	24000.00	NULL	NULL	NULL	NULL
101	Neena	Kochhar	17000.00	100	Steven	King	24000.00
102	Lex	De Haan	17000.00	100	Steven	King	24000.00
104	Bruce	Ernst	6000.00	NULL	NULL	NULL	NULL
105	David	Austin	4800.00	NULL	NULL	NULL	NULL
106	Valli	Pataballa	4800.00	NULL	NULL	NULL	NULL
107	Diana	Lorentz	4200.00	NULL	NULL	NULL	NULL
108	Nancy	Greenberg	12000.00	101	Neena	Kochhar	17000.00
109	Daniel	Faviet	9000.00	108	Nancy	Greenberg	12000.00
110	John	Chen	8200.00	108	Nancy	Greenberg	12000.00
111	Ismael	Sciarra	7700.00	108	Nancy	Greenberg	12000.00
112	Jose Manuel	Urman	7800.00	108	Nancy	Greenberg	12000.00
114	Den	Raphaely	11000.00	100	Steven	King	24000.00
115	Alexander	Khoo	3100.00	114	Den	Raphaely	11000.00
116	Shelli	Baida	2900.00	114	Den	Raphaely	11000.00
117	Sigal	Tobias	2800.00	114	Den	Raphaely	11000.00
118	Guy	Himuro	2600.00	114	Den	Raphaely	11000.00
119	Karen	Colmenares	2500.00	114	Den	Raphaely	11000.00
120	Matthew	Weiss	8000.00	100	Steven	King	24000.00
122	Payam	Kaufling	7900.00	100	Steven	King	24000.00
123	Shanta	Vollman	6500.00	100	Steven	King	24000.00
124	Kevin	Mourgos	5800.00	100	Steven	King	24000.00
125	Julia	Nayer	3200.00	120	Matthew	Weiss	8000.00
126	Irene	Mikkilineni	2700.00	120	Matthew	Weiss	8000.00
127	James	Landry	2400.00	120	Matthew	Weiss	8000.00
128	Steven	Markle	2200.00	120	Matthew	Weiss	8000.00
130	Mozhe	Atkinson	2800.00	NULL	NULL	NULL	NULL

5. Write a query to select employees and their corresponding managers and their salaries (SELF Join)

A:

```
SELECT
        e.Employee_id AS EmployeeID,
        e.first_name AS EmployeeFirstName,
        e.last_name AS EmployeeLastName,
        e.salary AS EmployeeSalary,
        m.Employee_id AS ManagerID,
        m.first_name AS ManagerFirstName,
        m.last_name AS ManagerLastName,
        m.salary AS ManagerSalary
FROM
        employees e                                  # Specify the main table and its alias
LEFT JOIN
        employees m ON e.manager_id = m.Employee_id  # left join to link each employee to their manager using the manager_id
ORDER BY
         e.Employee_id;                              # Then order the results by the employee's ID
```

OUTPUT:

```
144    # 5. Write a query to select employees and their corresponding managers and their salaries (SELF Join)
145    SELECT
146        e.Employee_id AS EmployeeID,
147        e.first_name AS EmployeeFirstName,
148        e.last_name AS EmployeeLastName,
149        e.salary AS EmployeeSalary,
150        m.Employee_id AS ManagerID,
151        m.first_name AS ManagerFirstName,
152        m.last_name AS ManagerLastName,
153        m.salary AS ManagerSalary
154    FROM
155        employees e                                  # Specify the main table and its alias
156    LEFT JOIN
157        employees m ON e.manager_id = m.Employee_id # left join to link each employee to their manager using the manager_id
158    ORDER BY
159        e.Employee_id;                               # Then order the results by the employee's ID
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmployeeID	EmployeeFirstName	EmployeeLastName	EmployeeSalary	ManagerID	ManagerFirstName	ManagerLastName	ManagerSalary
100	Steven	King	24000.00	NULL	NULL	NULL	NULL
101	Neena	Kochhar	17000.00	100	Steven	King	24000.00
102	Lex	De Haan	17000.00	100	Steven	King	24000.00
104	Bruce	Ernst	6000.00	NULL	NULL	NULL	NULL
105	David	Austin	4800.00	NULL	NULL	NULL	NULL
106	Valli	Pataballa	4800.00	NULL	NULL	NULL	NULL
107	Diana	Lorentz	4200.00	NULL	NULL	NULL	NULL
108	Nancy	Greenberg	12000.00	101	Neena	Kochhar	17000.00
109	Daniel	Faviet	9000.00	108	Nancy	Greenberg	12000.00
110	John	Chen	8200.00	108	Nancy	Greenberg	12000.00
111	Ismael	Sciarra	7700.00	108	Nancy	Greenberg	12000.00
112	Jose Manuel	Urman	7800.00	108	Nancy	Greenberg	12000.00
114	Den	Raphaely	11000.00	100	Steven	King	24000.00
115	Alexander	Khoo	3100.00	114	Den	Raphaely	11000.00
116	Shelli	Baida	2900.00	114	Den	Raphaely	11000.00
117	Sigal	Tobias	2800.00	114	Den	Raphaely	11000.00
118	Guy	Himuro	2600.00	114	Den	Raphaely	11000.00
119	Karen	Colmenares	2500.00	114	Den	Raphaely	11000.00
120	Matthew	Weiss	8000.00	100	Steven	King	24000.00
122	Payam	Kaufling	7900.00	100	Steven	King	24000.00
123	Shanta	Vollman	6500.00	100	Steven	King	24000.00
124	Kevin	Mourgos	5800.00	100	Steven	King	24000.00
125	Julia	Nayer	3200.00	120	Matthew	Weiss	8000.00
126	Irene	Mikkilineni	2700.00	120	Matthew	Weiss	8000.00
127	James	Landry	2400.00	120	Matthew	Weiss	8000.00
128	Steven	Markle	2200.00	120	Matthew	Weiss	8000.00
130	Mozhe	Atkinson	2800.00	NULL	NULL	NULL	NULL

6. Create a view for the above query

A:

```
CREATE VIEW Employee_and_Managers AS
        SELECT
                e.Employee_id AS EmployeeID,
                e.first_name AS EmployeeFirstName,
                e.last_name AS EmployeeLastName,
                e.salary AS EmployeeSalary,
                m.Employee_id AS ManagerID,
                m.first_name AS ManagerFirstName,
                m.last_name AS ManagerLastName,
                m.salary AS ManagerSalary
        FROM
                employees e                                     # Specify the main table and its alias
        LEFT JOIN
                employees m ON e.manager_id = m.Employee_id     # left join to link each employee to their manager using the manager_id
        ORDER BY
                e.Employee_id;                                  # Then order the results by the employee's ID

# To View the created VIEW 'Employee_and_Managers'
SELECT
        *
FROM
        Employee_and_Managers;
```

OUTPUT:

```
161
162
163     # 6. Create a view for the above query
164 •   CREATE VIEW Employee_and_Managers AS
165         SELECT
166             e.Employee_id AS EmployeeID,
167             e.first_name AS EmployeeFirstName,
168             e.last_name AS EmployeeLastName,
169             e.salary AS EmployeeSalary,
170             m.Employee_id AS ManagerID,
171             m.first_name AS ManagerFirstName,
172             m.last_name AS ManagerLastName,
173             m.salary AS ManagerSalary
174         FROM
175             employees e                                 # Specify the main table and its alias
176         LEFT JOIN
177             employees m ON e.manager_id = m.Employee_id # left join to link each employee to their manager using the manager_id
178         ORDER BY
179             e.Employee_id;                              # Then order the results by the employee's ID
180
181
182     # To View the created VIEW 'Employee_and_Managers'
183 •   SELECT
184         *
185     FROM
186         Employee_and_Managers;
187
188
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmployeeID	EmployeeFirstName	EmployeeLastName	EmployeeSalary	ManagerID	ManagerFirstName	ManagerLastName	ManagerSalary
100	Steven	King	24000.00	NULL	NULL	NULL	NULL
101	Neena	Kochhar	17000.00	100	Steven	King	24000.00
102	Lex	De Haan	17000.00	100	Steven	King	24000.00
104	Bruce	Ernst	6000.00	NULL	NULL	NULL	NULL
105	David	Austin	4800.00	NULL	NULL	NULL	NULL
106	Valli	Pataballa	4800.00	NULL	NULL	NULL	NULL
107	Diana	Lorentz	4200.00	NULL	NULL	NULL	NULL
108	Nancy	Greenberg	12000.00	101	Neena	Kochhar	17000.00
109	Daniel	Faviet	9000.00	108	Nancy	Greenberg	12000.00
110	John	Chen	8200.00	108	Nancy	Greenberg	12000.00
111	Ismael	Sciarra	7700.00	108	Nancy	Greenberg	12000.00
112	Jose Manuel	Urman	7800.00	108	Nancy	Greenberg	12000.00
114	Den	Raphaely	11000.00	100	Steven	King	24000.00

Employee_and_Managers 3 ×

Result Grid | Form Editor | Field Types | Read Only

```
182     # To View the created VIEW 'Employee_and_Managers'
183 •   SELECT
184         *
185     FROM
186         Employee_and_Managers;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmployeeID	EmployeeFirstName	EmployeeLastName	EmployeeSalary	ManagerID	ManagerFirstName	ManagerLastName	ManagerSalary
100	Steven	King	24000.00	NULL	NULL	NULL	NULL
101	Neena	Kochhar	17000.00	100	Steven	King	24000.00
102	Lex	De Haan	17000.00	100	Steven	King	24000.00
104	Bruce	Ernst	6000.00	NULL	NULL	NULL	NULL
105	David	Austin	4800.00	NULL	NULL	NULL	NULL
106	Valli	Pataballa	4800.00	NULL	NULL	NULL	NULL
107	Diana	Lorentz	4200.00	NULL	NULL	NULL	NULL
108	Nancy	Greenberg	12000.00	101	Neena	Kochhar	17000.00
109	Daniel	Faviet	9000.00	108	Nancy	Greenberg	12000.00
110	John	Chen	8200.00	108	Nancy	Greenberg	12000.00
111	Ismael	Sciarra	7700.00	108	Nancy	Greenberg	12000.00
112	Jose Manuel	Urman	7800.00	108	Nancy	Greenberg	12000.00
114	Den	Raphaely	11000.00	100	Steven	King	24000.00
115	Alexander	Khoo	3100.00	114	Den	Raphaely	11000.00
116	Shelli	Baida	2900.00	114	Den	Raphaely	11000.00
117	Sigal	Tobias	2800.00	114	Den	Raphaely	11000.00
118	Guy	Himuro	2600.00	114	Den	Raphaely	11000.00
119	Karen	Colmenares	2500.00	114	Den	Raphaely	11000.00
120	Matthew	Weiss	8000.00	100	Steven	King	24000.00
122	Payam	Kaufling	7900.00	100	Steven	King	24000.00
123	Shanta	Vollman	6500.00	100	Steven	King	24000.00
124	Kevin	Mourgos	5800.00	100	Steven	King	24000.00
125	Julia	Nayer	3200.00	120	Matthew	Weiss	8000.00
126	Irene	Mikkilineni	2700.00	120	Matthew	Weiss	8000.00
127	James	Landry	2400.00	120	Matthew	Weiss	8000.00
128	Steven	Markle	2200.00	120	Matthew	Weiss	8000.00
130	Mozhe	Atkinson	2800.00	NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types | Query Stats | Execution Plan

Employee_and_Managers 3 | Read Only

7. Write a query to show the count of employees under each manager in descending order (from view)

A:

```
SELECT
        ManagerId, ManagerFirstName, ManagerLastName, COUNT(EmployeeID) AS Employees
FROM
        employee_and_managers
WHERE
        ManagerID IS NOT NULL      #THE GIVEN SAMPLE DATABASE CONTAINS EMPLOYEES HAVING NO MANAGERS SO I REMOVED IT TO BEAUTIFY
GROUP BY
        ManagerId
ORDER BY
        Employees DESC;
```

OUTPUT:

```
189
190     # 7. Write a query to show the count of employees under each manager in descending order (from view)
191 •   SELECT
192         ManagerId, ManagerFirstName, ManagerLastName, COUNT(EmployeeID) AS Employees
193     FROM
194         employee_and_managers
195     WHERE
196         ManagerID IS NOT NULL      #THE GIVEN SAMPLE DATABASE CONTAINS EMPLOYEES HAVING NO MANAGERS SO I REMOVED IT TO BEAUTIFY
197     GROUP BY
198         ManagerId
199     ORDER BY
200         Employees DESC;
201
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ManagerID	ManagerFirstName	ManagerLastName	Employees
100	Steven	King	7
114	Den	Raphaely	5
108	Nancy	Greenberg	4
120	Matthew	Weiss	4
101	Neena	Kochhar	1

8. Find the count of employees in each department

A:

```
SELECT
	Department_Name AS Department,
	COUNT(employees.Employee_id) AS 'Number of Employees'
FROM
	departments
LEFT JOIN
	employees
ON
	departments.Department_id = employees.department_id
GROUP BY
	Department_name;
```

OUTPUT:

```
203
204     # 8. Find the count of employees in each department
205 •   SELECT
206         Department_Name AS Department,                          # The column is selected from the departments table and is aliased as Department
207         COUNT(employees.Employee_id) AS 'Number of Employees'   # Counts the number of employees in each department and aliased as Number of Employees
208     FROM                                                        # Specifies the departments table as the main table from which to select data.
209         departments
210     LEFT JOIN                                                   # LEFT JOIN ensures that all records from the departments table are included in the result,
211         employees                                               # even if there are no matching records in the employees table.
212     ON                                                          # This condition makes the join performed where the Department_id from
213         departments.Department_id = employees.department_id     # the departments table matches the department_id from the employees table.
214     GROUP BY                                                    # groups the rows by Department_Name,
215         Department_name;                                        # so the COUNT function calculates the number of employees in each department group.
216
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Department	Number of Employees
Marketing	2
Purchasing	3
Human Resources	3
Shipping	5
IT	3
Public Relations	1
Sales	2
Executive	1
Finance	1
Accounting	1
Treasury	0
Corporate Tax	1
Control And Credit	0
Shareholder Serv...	1
Benefits	1
Payroll	2

9. Get the count of employees hired year wise

A:

```
SELECT
        YEAR(hire_date ) AS 'year',
        COUNT(Employee_id) AS 'employees hired'
FROM
        employees
GROUP BY
        YEAR(hire_date );
```

OUTPUT:

```
218
219     # 9. Get the count of employees hired year wise
220 •   SELECT
221         YEAR(hire_date ) AS 'year',                 # extracts the year from the hire_date column using the YEAR function and aliased as 'year'
222         COUNT(Employee_id) AS 'employees hired'     # count the number of Employee_id values in each group and aliased as 'employees hired'
223     FROM
224         employees                                   # from employees table as the input data
225     GROUP BY                                        # group the rows by the extracted year,
226         YEAR(hire_date );                           # so the COUNT function calculate number of employees hired for each year.
227
228
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	employees hired
1987	1
1989	1
1993	1
1991	1
1997	8
1998	4
1999	4
1994	3
1995	2
1996	1
2000	1

Result Grid | Form Editor | Field Types | Query Stats | Execution Plan

10 . create a stored procedure to get the “ Get the count of employees hired in the input year”(IN year , OUT count)

A:

```
DELIMITER //
CREATE PROCEDURE get_employee_count_by_year(IN input_year INT, OUT count INT)
BEGIN
	SELECT COUNT(*)
	INTO count
	FROM employees
	WHERE YEAR(hire_date) = input_year;
END //
DELIMITER ;

CALL get_employee_count_by_year(1994, @employee_count);
SELECT
	@employee_count AS 'The count of employees hired in 1994';
```

OUTPUT:

```
229
230    # 10 . create a stored procedure to get the “ Get the count of employees hired in the input year”(IN year , OUT count)
231    DELIMITER //                                                    # change delimiter to //
232    CREATE PROCEDURE get_employee_count_by_year(IN input_year INT, OUT count INT)   # define procedure with input parameter 'input_year', output parameter 'employee_count'
233    BEGIN                                                           # BEGIN ... END: is the body of the procedure
234        SELECT COUNT(*)                                             # run the query to count the number of employees hired in the given year
235        INTO count                                                  # and stores the result into count variable.
236        FROM employees
237        WHERE YEAR(hire_date) = input_year;
238    END //
239    DELIMITER ;                                                    # change delimiter back to ;
240
241    CALL get_employee_count_by_year(1994, @employee_count);         # call the procedure
242    SELECT                                                          # @employee_count is session variable to hold the output of the stored procedure.
243        @employee_count AS 'The count of employees hired in 1994';  # output aliased to 'The count of employees hired in 1994'
244
245
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

The count of employees hired in 1994
3

Result Grid

Result 14 ×

Read Only

11.Select the employees whose first_name contains “an”

A:

```
SELECT
        *                                       # select all
FROM
        employees                               # from employees table
WHERE                                           # where first_name contains “an”
        first_name LIKE '%an%';                 # %an%' Filters the rows where the first_name column contains the substring "an"
```

OUTPUT:

```
246
247     # 11.Select the employees whose first_name contains “an”
248     SELECT
249         *                                   # select all
250     FROM
251         employees                           # from employees table
252     WHERE                                   # where first_name contains “an”
253         first_name LIKE '%an%';             # %an%' Filters the rows where the first_name column contains the substring "an"
254
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
107	Diana	Lorentz	DLORENTZ	590.423.5567	1999-02-09	IT_PROG	4200.00	NULL	103	40
108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17	FI_MGR	12000.00	NULL	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	1994-08-12	FI_ACCOUNT	9000.00	NULL	108	170
112	Jose Manuel	Urman	JMURMAN	515.124.4469	1998-06-03	FI_ACCOUNT	7800.00	NULL	108	150
115	Alexander	Khoo	AKHOO	515.127.4562	1995-05-12	PU_CLERK	3100.00	NULL	114	80
123	Shanta	Vollman	SVOLLMAN	650.123.4234	1997-10-12	ST_MAN	6500.00	NULL	100	50
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

12. Select employee first name and the corresponding phone number in the format (_ _ _)-(_ _ _)-(_ _ _ _)

A:

```
# phone_number format on employees table 515.123.4567
SELECT
	# Selects the first_name coulum
	first_name,
	# SUBSTRING(data, start position, length)
	CONCAT( '(',  SUBSTRING(phone_number,1,3),  ')-(',  SUBSTRING(phone_number,5,3),  ')-(',  SUBSTRING(phone_number,9,4),  ')') AS 'phonenumber'
FROM
	employees;
```

OUTPUT:

```
256
257     # 12.  Select employee first name and the corresponding phone number in the format (_ _ _)-(_ _ _)-(_ _ _ _)
258     # phone_number format on employees table 515.123.4567
259     SELECT
260         # Selects the first_name coulum
261         first_name,
262         # SUBSTRING(data, start position, length)
263         CONCAT( '(',  SUBSTRING(phone_number,1,3),  ')-(',  SUBSTRING(phone_number,5,3),  ')-(',  SUBSTRING(phone_number,9,4),  ')')  AS 'phone number'
264     FROM
265         employees;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	phone number
Steven	(515)-(123)-(4567)
Neena	(515)-(123)-(4568)
Lex	(515)-(123)-(4569)
Bruce	(590)-(423)-(4568)
David	(590)-(423)-(4569)
Valli	(590)-(423)-(4560)
Diana	(590)-(423)-(5567)
Nancy	(515)-(124)-(4569)
Daniel	(515)-(124)-(4169)
John	(515)-(124)-(4269)
Ismael	(515)-(124)-(4369)
Jose Manuel	(515)-(124)-(4469)
Den	(515)-(127)-(4561)
Alexander	(515)-(127)-(4562)
Shelli	(515)-(127)-(4563)
Sigal	(515)-(127)-(4564)
Guy	(515)-(127)-(4565)
Karen	(515)-(127)-(4566)
Matthew	(650)-(123)-(1234)
Payam	(650)-(123)-(3234)
Shanta	(650)-(123)-(4234)
Kevin	(650)-(123)-(5234)
Julia	(650)-(124)-(1214)
Irene	(650)-(124)-(1224)
James	(650)-(124)-(1334)
Steven	(650)-(124)-(1434)
Mozhe	(650)-(124)-(6234)

13. Find the employees who joined in August, 1994.

A:

```
SELECT
        *                                                   # select all from employees table
FROM
        employees
WHERE                                                       # where both matched (YEAR(hire_date) = 1994 AND MONTH(hire_date) = 8;)
        YEAR(hire_date) = 1994 AND MONTH(hire_date) = 8;
```

OUTPUT:

```
269     # 13. Find the employees who joined in August, 1994.
270     SELECT
271         *                                               # select all from employees table
272     FROM
273         employees
274     WHERE                                               # where both matched (YEAR(hire_date) = 1994 AND MONTH(hire_date) = 8;)
275         YEAR(hire_date) = 1994 AND MONTH(hire_date) = 8;
276
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17	FI_MGR	12000.00	NULL	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	1994-08-12	FI_ACCOUNT	9000.00	NULL	108	170
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

14. Find the maximum salary from each department.

A:

```
SELECT
	e.department_id AS 'Department ID', d.Department_name AS 'Department', MAX(salary) AS 'Maximum salary'
FROM
	employees e
JOIN
	departments d ON e.department_id = d.Department_id
GROUP BY
	e.department_id;
```

OUTPUT:

```
278
279     # 14. Find the maximum salary from each department.
280     SELECT
281         e.department_id AS 'Department ID', d.Department_name AS 'Department', MAX(salary) AS 'Maximum salary'
282     FROM
283         employees e
284     JOIN                                                # join employees table (e) and the departments table (d).
285         departments d ON e.department_id = d.Department_id   # ON where the department_id in employees matches the Department_id in departments table
286     GROUP BY
287         e.department_id;
288
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Department ID	Department	Maximum salary
20	Marketing	24000.00
30	Purchasing	17000.00
60	IT	6000.00
40	Human Resources	7900.00
100	Finance	12000.00
170	Payroll	9000.00
160	Benefits	7700.00
150	Shareholder Services	7800.00
80	Sales	5800.00
70	Public Relations	2900.00
130	Corporate Tax	2500.00
50	Shipping	8000.00
90	Executive	2400.00
110	Accounting	2800.00

15.Write a SQL query to display the 5 least earning employees

A:

```
SELECT
        *
FROM
        employees
ORDER BY
        salary LIMIT 5;
```

OUTPUT:

```
290
291     # 15. Write a SQL query to display the 5 least earning employees
292     SELECT
293         *
294     FROM
295         employees
296     ORDER BY
297         salary LIMIT 5;
298
299
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

Employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
128	Steven	Markle	SMARKLE	650.124.1434	2000-03-04	ST_CLERK	2200.00	NULL	120	50
127	James	Landry	JLANDRY	650.124.1334	1999-01-02	ST_CLERK	2400.00	NULL	120	90
119	Karen	Colmenares	KCOLMENA	515.127.4566	1999-04-08	PU_CLERK	2500.00	NULL	114	130
118	Guy	Himuro	GHIMURO	515.127.4565	1998-01-02	PU_CLERK	2600.00	NULL	114	60
126	Irene	Mikkilineni	IMIKKILI	650.124.1224	1998-11-12	ST_CLERK	2700.00	NULL	120	50
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

16. Find the employees hired in the 80s

A:

```
SELECT
        *
FROM
        employees
WHERE
        YEAR(hire_date) >= 1980 AND YEAR(hire_date) < 1990;
```

OUTPUT:

```
300
301     # 16. Find the employees hired in the 80s
302     SELECT
303         *
304     FROM
305         employees
306     WHERE
307         YEAR(hire_date) >= 1980 AND YEAR(hire_date) < 1990;
308
309
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24000.00	NULL	NULL	20
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-11-21	AD_VP	17000.00	NULL	100	20
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

17. Find the employees who joined the company after 15th of the month

A:

```
SELECT
        *
FROM
        employees
WHERE
        DAY(hire_date) > 15;
```

OUTPUT:

```
310
311     # 17. Find the employees who joined the company after 15th of the month
312     SELECT
313         *
314     FROM
315         employees
316     WHERE
317        DAY(hire_date) > 15;
318
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24000.00	NULL	NULL	20
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-11-21	AD_VP	17000.00	NULL	100	20
104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6000.00	NULL	103	60
105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4800.00	NULL	103	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17	FI_MGR	12000.00	NULL	101	100
120	Matthew	Weiss	MWEISS	650.123.1234	1996-07-18	ST_MAN	8000.00	NULL	100	50
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL