

MYSQL FIRST TASK

```
CREATE DATABASE Entri_assignment;  
use Entri_assignment;
```

```
CREATE TABLE departments (  
Department_id INT PRIMARY KEY, Department_name VARCHAR (30), Location_id INT  
);
```

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);
```

```
CREATE TABLE employees (
```

MYSQL FIRST TASK

```
Employee_id INT PRIMARY KEY,  
first_name VARCHAR(50),  
last_name VARCHAR(50),  
email VARCHAR(100),  
phone_number VARCHAR(20),  
hire_date DATE,  
job_id VARCHAR(10),  
salary DECIMAL(10, 2),  
commission_pct DECIMAL(5, 2),  
manager_id INT,  
department_id INT,  
FOREIGN KEY (department_id) REFERENCES departments(department_id) -- Assuming  
there's a departments table with department_id as PK  
);
```

```
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, 8, 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);
```

MYSQL FIRST TASK

```
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);
```

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

```
SELECT  
    first_name  
FROM  
    employees  
WHERE first_name LIKE "S%";
```

OUTPUT:

```
first_name  
Steven  
Shelli  
Sigal  
Shanta  
Steven
```

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

```
select first_name ,last_name,salary FROM employees  
where salary = (SELECT max(salary) FROM employees );
```

MYSQL FIRST TASK

OUTPUT:

first_name	last_name	salary
Steven	King	24000

3. Select employee with the second highest salary

```
SELECT
    first_name, last_name, salary
FROM
    employees
ORDER BY salary DESC
LIMIT 5;
```

OUTPUT:

first_name	last_name	salary
Steven	King	24000
Neena	Kochhar	17000
Lex	De Haan	17000
Nancy	Greenberg	12000
Den	Raphaely	11000

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

```
e1.first_name as employee_first_name,
e1.last_name as employee_last_name,
e2.first_name as manager_first_name,
e2.last_name AS manager_last_name,
e1.salary AS employee_salary,
e2.salary AS manager_salary
```

```
FROM
    employees e1
LEFT JOIN
    employees e2
ON
    e1.manager_id = e2.Employee_id;
```

OUTPUT:

employee_first_name	employee_last_name	manager_first_name	manager_last_name	employee_salary	manager_salary
Steven	King	NULL	NULL	24000	NULL
Neena	Kochhar	Steven	King	17000	24000
Lex	De Haan	Steven	King	17000	24000

MYSQL FIRST TASK

Bruce	Ernst	NULL	NULL	6000	NULL
David	Austin	NULL	NULL	4800	NULL
Valli	Pataballa	NULL	NULL	4800	NULL
Diana	Lorentz	NULL	NULL	4200	NULL
Nancy	Greenberg	Neena	Kochhar	12000	17000
Daniel	Faviet	Nancy	Greenberg	9000	12000
John	Chen	Nancy	Greenberg	8200	12000
Ismael	Sciarra	Nancy	Greenberg	7700	12000
Jose Manuel	Urman	NULL	NULL	7800	NULL
Den	Raphaely	Steven	King	11000	24000
Alexander	Khoo	Den	Raphaely	3100	11000
Shelli	Baida	Den	Raphaely	2900	11000
Sigal	Tobias	Den	Raphaely	2800	11000
Guy	Himuro	Den	Raphaely	2600	11000
Karen	Colmenares	Den	Raphaely	2500	11000
Matthew	Weiss	Steven	King	8000	24000
Payam	Kaufling	Steven	King	7900	24000
Shanta	Vollman	Steven	King	6500	24000
Kevin	Mourgos	Steven	King	5800	24000
Julia	Nayer	Matthew	Weiss	3200	8000
Irene	Mikkilineni	Matthew	Weiss	2700	8000
James	Landry	Matthew	Weiss	2400	8000
Steven	Markle	Matthew	Weiss	2200	8000
Mozhe	Atkinson	NULL	NULL	2800	NULL

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

```
select
e1.first_name as employee_first_name,
e1.last_name as employee_last_name,
e2.first_name as manager_first_name,
e2.last_name AS manager_last_name,
e1.salary AS employee_salary,
e2.salary AS manager_salary
```

```
FROM
  employees e1
INNER JOIN
  employees e2
ON
  e1.manager_id = e2.Employee_id;
```

OUTPUT:

MYSQL FIRST TASK

employee_first_ name	employee_last_ name	manager_first_ name	manager_last_ name	employee_s alary	manager_sal ary
Neena	Kochhar	Steven	King	17000	24000
Lex	De Haan	Steven	King	17000	24000
Nancy	Greenberg	Neena	Kochhar	12000	17000
Daniel	Faviet	Nancy	Greenberg	9000	12000
John	Chen	Nancy	Greenberg	8200	12000
Ismael	Sciarra	Nancy	Greenberg	7700	12000
Den	Raphaely	Steven	King	11000	24000
Alexander	Khoo	Den	Raphaely	3100	11000
Shelli	Baida	Den	Raphaely	2900	11000
Sigal	Tobias	Den	Raphaely	2800	11000
Guy	Himuro	Den	Raphaely	2600	11000
Karen	Colmenares	Den	Raphaely	2500	11000
Matthew	Weiss	Steven	King	8000	24000
Payam	Kaufling	Steven	King	7900	24000
Shanta	Vollman	Steven	King	6500	24000
Kevin	Mourgos	Steven	King	5800	24000
Julia	Nayer	Matthew	Weiss	3200	8000
Irene	Mikkilineni	Matthew	Weiss	2700	8000
James	Landry	Matthew	Weiss	2400	8000
Steven	Markle	Matthew	Weiss	2200	8000

6. Create a view for the above query

```
CREATE VIEW hierarchy as
select
e1.first_name as employee_first_name,
e1.last_name as employee_last_name,
e2.first_name as manager_first_name,
e2.last_name AS manager_last_name,
e1.salary AS employee_salary,
e2.salary AS manager_salary
```

```
FROM
  employees e1
INNER JOIN
  employees e2
ON
  e1.manager_id = e2.Employee_id;
```

show full tables;

OUTPUT:

MYSQL FIRST TASK

Tables_in_entri_assignment	Table_type
departments	BASE TABLE
employees	BASE TABLE
hierarchy	VIEW

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

```
SELECT
    manager_first_name,
    manager_last_name,
    COUNT(employee_first_name) AS no_of_employee
FROM
    hierarchy
GROUP BY manager_first_name , manager_last_name
ORDER BY no_of_employee;
```

OUTPUT:

manager_first_name	manager_last_name	no_of_employee
Neena	Kochhar	1
Nancy	Greenberg	3
Matthew	Weiss	4
Den	Raphaely	5
Steven	King	7

#8. Find the count of employees in each department

```
select
d.department_id,
d.department_name,
count(e.first_name) as counts
```

```
FROM
    employees e
LEFT JOIN
    departments d
ON
    e.department_id = d.department_id
```

```
group by
d.department_id,
d.department_name ;
```

MYSQL FIRST TASK

OUTPUT:

department_id	department_name	counts
20	Marketing	2
30	Purchasing	3
60	IT	3
40	Human Resources	3
100	Finance	1
170	Payroll	2
160	Benefits	1
	Shareholder	
150	Services	1
80	Sales	2
70	Public Relations	1
130	Corporate Tax	1
50	Shipping	5
90	Executive	1
110	Accounting	1

9. Get the count of employees hired year wise

```
SELECT
  COUNT(*) AS counts, YEAR(hire_date) AS hire_year
FROM
  employees
GROUP BY YEAR(hire_date)
ORDER BY hire_year;
```

OUTPUT:

counts	hire_year
1	1987
1	1989
1	1991
1	1993
3	1994
2	1995
1	1996
8	1997
4	1998
4	1999
1	2000

MYSQL FIRST TASK

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

```
DELIMITER $$
CREATE PROCEDURE COUNT_EMPLOYEES(IN input_year INT, OUT employee_count
INT)
BEGIN
SET employee_count=(select count(*) from employees
where year(hire_date)=input_year);

END $$
```

DELIMITER ;

```
call COUNT_EMPLOYEES(1999,@count);
select @count;
```

OUTPUT:

```
@count
      4
```

11.Select the employees whose first_name contains “an”

```
SELECT
    first_name
FROM
    employees
WHERE first_name LIKE "%an%";
```

OUTPUT:

```
first_name
Diana
Nancy
Daniel
Jose Manuel
Alexander
Shanta
```

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

MYSQL FIRST TASK

```
SELECT
    first_name,
    CONCAT('(',
        SUBSTRING(phone_number, 1, 3),
        ')-(',
        SUBSTRING(phone_number, 5, 3),
        ')-(',
        SUBSTRING(phone_number, 9, 4),
        ')') AS 'phone number'
FROM
    employees;
```

OUTPUT:

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)

MYSQL FIRST TASK

Mozhe (650)-(124)-(6234)

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

```
SELECT
    first_name, last_name, hire_date
FROM
    employees
WHERE
    hire_date LIKE '1994-08-%'
```

OUTPUT:

first_name	last_name	hire_date
Nancy	Greenberg	8/17/1994
Daniel	Faviet	8/12/1994

14. Find the maximum salary from each department.

```
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:

Department ID	Department	Maximum salary
20	Marketing	24000
30	Purchasing	17000
60	IT	6000
40	Human Resources	7900
100	Finance	12000
170	Payroll	9000
160	Benefits	7700

MYSQL FIRST TASK

	Shareholder	
150	Services	7800
80	Sales	5800
70	Public Relations	2900
130	Corporate Tax	2500
50	Shipping	8000
90	Executive	2400
110	Accounting	2800

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

```
SELECT
    first_name, salary
FROM
    employees
ORDER BY salary
LIMIT 5;
```

OUTPUT:

first_name	salary
Steven	2200
James	2400
Karen	2500
Guy	2600
Irene	2700

16. Find the employees hired in the 80s

```
SELECT
    first_name, last_name, YEAR(hire_date)
FROM
    employees
WHERE
    YEAR(hire_date) LIKE '198%'
```

MYSQL FIRST TASK

OUTPUT:

first_name	last_name	YEAR(hire_date)
Steven	King	1987
Neena	Kochhar	1989

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

```

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

```

OUTPUT:

Empl oyee _id	first _na me	last_ _na me	emai l	phone _num ber	hire _da te	job _id	sal ary	commi ssion_ pct	man ager _id	departme nt_id
					###	AD				
100	Stev en	King	SKIN G	515.12 3.4567	### ##	_PR ES	240 00	NULL	NULL	20
			NKO		###					
101	Nee na	Koc hhar	CHH AR	515.12 3.4568	### ##	AD _VP	170 00	NULL	100	20
					###	IT_				
104	Bruc e	Erns t	BER NST	590.42 3.4568	### ##	PR OG	600 0	NULL	103	60
					###	IT_				
105	Davi d	Aust in	DAU STIN	590.42 3.4569	### ##	PR OG	480 0	NULL	103	60
			Gre NGR		###	FI_				
108	Nan cy	enb erg	EEN BE	515.12 4.4569	### ##	MG R	120 00	NULL	101	100
					###	ST_				
120	Mat the w	Wei ss	MW EISS	650.12 3.1234	### ##	MA N	800 0	NULL	100	50

MYSQL FIRST TASK