

## SQL Exercises : Hr database(Solution)

### Basic SELECT statement [19 exercises]

1. Write a query to display the names (first\_name, last\_name) using alias name "First Name", "Last Name"

**SELECT FIRST\_NAME, LAST\_NAME FROM employees;**

2. Write a query to get unique department ID from employee table.

**SELECT DISTINCT DEPARTMENT\_ID FROM employees;**

3. Write a query to get all employee details from the employee table order by first name, descending.

**SELECT \* FROM employees ORDER BY FIRST\_NAME DESC;**

4. Write a query to get the names (first\_name, last\_name), salary, PF of all the employees (PF is calculated as 15% of salary).

**SELECT FIRST\_NAME, LAST\_NAME, SALARY, SALARY \* 0.15 AS PF FROM employees;**

5. Write a query to get the employee ID, names (first\_name, last\_name), salary in ascending order of salary.

**SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY FROM employees ORDER BY SALARY;**

6. Write a query to get the total salaries payable to employees.

**SELECT SUM(SALARY) AS total\_salary FROM employees;**

7. Write a query to get the maximum and minimum salary from employees table.

**SELECT MAX(SALARY) AS max\_salary FROM employees;**

**SELECT MIN(SALARY) AS min\_salary FROM employees;**

8. Write a query to get the average salary and number of employees in the employees table.

**SELECT AVG(SALARY) AS average\_salary FROM employees;**

9. Write a query to get the number of employees working with the company.

**SELECT COUNT(\*) FROM employees;**

10. Write a query to get the number of jobs available in the employees table.

**SELECT COUNT(DISTINCT JOB\_ID) FROM employees;**

11. Write a query get all first name from employees table in upper case.

**SELECT UPPER(FIRST\_NAME) FROM employees;**

12. Write a query to get the first 3 characters of first name from employees table.

**SELECT SUBSTRING(FIRST\_NAME, 1, 3) FROM employees;**

13. Write a query to calculate  $171 \times 214 + 625$ .

**SELECT 171\*214+625 AS result;**

14. Write a query to get the names (for example Ellen Abel, Sundar Ande etc.) of all the employees from employees table.

**SELECT CONCAT(FIRST\_NAME, ' ', LAST\_NAME) AS employee\_name FROM employees;**

15. Write a query to get first name from employees table after removing white spaces from both side.

**SELECT TRIM(FIRST\_NAME) FROM employees;**

16. Write a query to get the length of the employee names (first\_name, last\_name) from employees table.

**SELECT FIRST\_NAME, LAST\_NAME, LENGTH(FIRST\_NAME) + LENGTH(LAST\_NAME) AS length\_number FROM employees;**

17. Write a query to check if the first\_name fields of the employees table contains numbers.

**SELECT \* FROM employees WHERE FIRST\_NAME REGEXP '[0-9]';**

18. Write a query to select first 10 records from a table.

**SELECT EMPLOYEE\_ID, FIRST\_NAME FROM employees LIMIT 10;**

19. Write a query to get monthly salary (round 2 decimal places) of each and every employee  
Note : Assume the salary field provides the 'annual salary' information.

**SELECT FIRST\_NAME, LAST\_NAME, ROUND(SALARY/12,2) AS monthly\_salary FROM employees;**

## **MySQL Restricting and Sorting data: [11 exercises]**

1. Write a query to display the name (first\_name, last\_name) and salary for all employees whose salary is not in the range \$10,000 through \$15,000.

**SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM employees WHERE SALARY NOT BETWEEN 10000 AND 15000;**

2. Write a query to display the name (first\_name, last\_name) and department ID of all employees in departments 30 or 100 in ascending order.

**SELECT FIRST\_NAME, LAST\_NAME, DEPARTMENT\_ID FROM employees WHERE DEPARTMENT\_ID IN (30,100) ORDER BY DEPARTMENT\_ID ASC;**

3. Write a query to display the name (first\_name, last\_name) and salary for all employees whose salary is not in the range \$10,000 through \$15,000 and are in department 30 or 100.  
**SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT\_ID FROM employees WHERE SALARY NOT BETWEEN 10000 AND 15000 AND DEPARTMENT\_ID IN (30,100);**
4. Write a query to display the name (first\_name, last\_name) and hire date for all employees who were hired in 1987.  
**SELECT FIRST\_NAME, LAST\_NAME, HIRE\_DATE FROM employees WHERE YEAR(HIRE\_DATE) LIKE '1987%';**
5. Write a query to display the first\_name of all employees who have both "b" and "c" in their first name.  
**SELECT FIRST\_NAME FROM employees WHERE FIRST\_NAME LIKE '%b%' AND FIRST\_NAME LIKE '%c%';**
6. Write a query to display the last name, job, and salary for all employees whose job is that of a Programmer or a Shipping Clerk, and whose salary is not equal to \$4,500, \$10,000, or \$15,000.  
**SELECT LAST\_NAME, JOB\_ID, SALARY FROM employees WHERE JOB\_ID IN ('IT\_PROG','SH\_CLERK') AND SALARY NOT IN (4500,10000,15000);**
7. Write a query to display the last name of employees whose names have exactly 6 characters.  
**SELECT LAST\_NAME FROM employees WHERE LAST\_NAME LIKE '\_\_\_\_\_';**
8. Write a query to display the last name of employees having 'e' as the third character.  
**SELECT LAST\_NAME FROM employees WHERE LAST\_NAME LIKE '\_\_e%';**
9. Write a query to display the jobs/designations available in the employees table.  
**SELECT DISTINCT JOB\_ID FROM employees;**
10. Write a query to display the name (first\_name, last\_name), salary and PF (15% of salary) of all employees.  
**SELECT FIRST\_NAME, LAST\_NAME, SALARY, SALARY\*0.15 AS PF FROM employees;**
11. Write a query to select all record from employees where last name in 'BLAKE', 'SCOTT', 'KING' and 'FORD'.  
**SELECT \* FROM employees WHERE LAST\_NAME IN ('JONES', 'BLAKE', 'SCOTT', 'KING', 'FORD');**

## **Aggregate Functions and Group by [14 exercises]**

1. Write a query to list the number of jobs available in the employees table.  
**SELECT COUNT (DISTINCT JOB\_ID) AS job\_available FROM employees;**
2. Write a query to get the total salaries payable to employees.  
**SELECT SUM(SALARY) AS total\_salary FROM employees;**
3. Write a query to get the minimum salary from employees table.  
**SELECT MIN(SALARY) AS min\_salary FROM employees;**
4. Write a query to get the maximum salary of an employee working as a Programmer.  
**SELECT MAX(SALARY) FROM employees WHERE JOB\_ID='IT\_PROG';**
5. Write a query to get the average salary and number of employees working the department 90.  
**SELECT AVG(SALARY), COUNT (\*) FROM employees WHERE DEPARTMENT\_ID=90;**
6. Write a query to get the highest, lowest, sum, and average salary of all employees.  
**SELECT ROUND(MAX(SALARY),0) AS maximum, ROUND(MIN(SALARY),0) AS minimum, ROUND(SUM(SALARY),0) AS sum, ROUND(AVG(SALARY),0) AS average FROM employees;**
7. Write a query to get the number of employees with the same job.  
**SELECT JOB\_ID, COUNT (\*) FROM employees GROUP BY JOB\_ID;**
8. Write a query to get the difference between the highest and lowest salaries.  
**SELECT MAX(SALARY) - MIN(SALARY) AS difference FROM employees;**
9. Write a query to find the manager ID and the salary of the lowest-paid employee for that manager.  
**SELECT MANAGER\_ID, MIN(SALARY) FROM employees WHERE MANAGER\_ID IS NOT NULL GROUP BY MANAGER\_ID ORDER BY MIN(SALARY) DESC;**
10. Write a query to get the department ID and the total salary payable in each department.  
**SELECT DEPARTMENT\_ID, SUM(SALARY) FROM employees GROUP BY DEPARTMENT\_ID;**
11. Write a query to get the average salary for each job ID excluding programmer.  
**SELECT JOB\_ID, ROUND(AVG(SALARY),0) FROM employees WHERE JOB\_ID <> 'IT\_PROG' GROUP BY JOB\_ID;**
12. Write a query to get the total salary, maximum, minimum, average salary of employees (job ID wise), for department ID 90 only.  
**SELECT JOB\_ID, SUM(SALARY), MAX(SALARY), MIN(SALARY), AVG(SALARY) FROM employees WHERE DEPARTMENT\_ID='90' GROUP BY JOB\_ID;**
13. Write a query to get the job ID and maximum salary of the employees where maximum salary is greater than or equal to \$4000.

**SELECT JOB\_ID, MAX(SALARY) FROM employees GROUP BY JOB\_ID HAVING  
MAX(SALARY)>=4000;**

- 14.** Write a query to get the average salary for all departments employing more than 10 employees.

**SELECT DEPARTMENT\_ID, AVG(SALARY), COUNT (\*) FROM employees GROUP BY  
DEPARTMENT\_ID HAVING COUNT (\*) >= 10;**

## **Date and Time functions [21 exercises]**

- 1.** Write a query to display the first day of the month (in datetime format) three months before the current month.

Sample current date : 2014-09-03

Expected result : 2014-06-01

**SELECT DATE\_FORMAT (DATE\_SUB ('2014-09-03', INTERVAL 3 MONTH), '%Y-%m-01') AS  
first\_day\_three\_months\_ago;**

**SELECT DATE\_FORMAT (ADDDATE (NOW (), INTERVAL-3 MONTH),'01/%M/%Y') AS  
preDate;**

- 2.** Write a query to display the last day of the month (in datetime format) three months before the current month.

**SELECT LAST\_DAY (DATE\_SUB ('2014-09-03', INTERVAL 3 MONTH)) AS  
last\_day\_three\_months\_ago;**

**SELECT LAST\_DAY(ADDDATE(NOW(), INTERVAL-3 MONTH)) AS lastDate;**

- 3.** Write a query to get the distinct Mondays from hire\_date in employees tables.

**SELECT DISTINCT hire\_date FROM employees WHERE DAYOFWEEK (hire\_date) = 2;**

**SELECT DISTINCT(HIRE\_DATE) FROM employees WHERE DAYNAME(HIRE\_DATE) =  
'Monday';**

- 4.** Write a query to get the first day of the current year.

**SELECT DATE\_FORMAT (CURRENT\_DATE, '%Y-01-01') AS first\_day\_of\_current\_year;**

**SELECT DATE\_FORMAT(NOW(), '01-01-%Y') AS Firstday;**

- 5.** Write a query to get the last day of the current year.

**SELECT DATE\_FORMAT(LAST\_DAY(CONCAT(YEAR(CURRENT\_DATE), '-12-01')),  
'%Y-%m- %d') AS last\_day\_of\_current\_year;**

**SELECT DATE\_FORMAT(NOW(), '31-12-%Y') AS lastDayOfYear;**

6. Write a query to calculate the age in year.

**SELECT DATEDIFF(NOW(), '2000-05-15') DIV 365 AS Current\_Age;**

7. Write a query to get the current date in the following format.

Sample date : 2014-09-04

Output : September 4, 2014

**SELECT DATE\_FORMAT (NOW (), '%M %e %Y') AS my\_date;**

8. Write a query to get the current date in Thursday September 2014 format.

Thursday September 2014

**SELECT DATE\_FORMAT(NOW(), '%W %M %Y') AS my\_date;**

9. Write a query to extract the year from the current date.

**SELECT YEAR (NOW ()) AS current\_year;**

10. Write a query to get the DATE value from a given day (number in N).

Sample days: 730677

Output : 2000-07-11

**SELECT FROM\_DAYS (730677) AS date;**

11. Write a query to get the first name and hire date from employees table where hire date between '1987-06-01' and '1987-07-30'

**SELECT FIRST\_NAME, HIRE\_DATE FROM employees WHERE HIRE\_DATE BETWEEN '1987-06-01' AND '1987-07-30';**

12. Write a query to display the current date in the following format.

Sample output: Thursday 4th September 2014 00:00:00

**SELECT DATE\_FORMAT (NOW (), '%W %D %M %Y %T') AS Curr\_date\_time;**

13. Write a query to display the current date in the following format.

Sample output: 05/09/2014

**SELECT DATE\_FORMAT (NOW (), '%d/%m/%Y') AS Curr\_date\_time;**

14. Write a query to display the current date in the following format.

Sample output: 12:00 AM Sep 5, 2014

**SELECT DATE\_FORMAT (NOW (), '%l : %i %p %b %e, %Y') AS Curr\_date\_time;**

15. Write a query to get the firstname, lastname who joined in the month of June.

**SELECT FIRST\_NAME, LAST\_NAME, HIRE\_DATE FROM employees WHERE MONTH(HIRE\_DATE) = 6;**

16. Write a query to get the years in which more than 10 employees joined.

**SELECT YEAR(HIRE\_DATE), COUNT(\*) FROM employees GROUP BY YEAR(HIRE\_DATE) HAVING COUNT(\*) > 10;**

17. Write a query to get first name of employees who joined in 1987.  
**SELECT FIRST\_NAME, HIRE\_DATE FROM employees WHERE YEAR(HIRE\_DATE) = 1987;**
  
18. Write a query to get department name, manager name, and salary of the manager for all managers whose experience is more than 5 years.  
**SELECT \* FROM (SELECT D.DEPARTMENT\_NAME, CONCAT(E.FIRST\_NAME, ' ', E.LAST\_NAME) AS MANAGER\_NAME, DATEDIFF(NOW(), HIRE\_DATE) DIV 365 AS YEAR\_OF\_EXPERIENCE FROM departments AS D, employees AS E WHERE D.MANAGER\_ID = E.EMPLOYEE\_ID) AS SRC WHERE YEAR\_OF\_EXPERIENCE > 5;**
  
19. Write a query to get employee ID, last name, and date of first salary of the employees.  
**SELECT EMPLOYEE\_ID, LAST\_NAME, HIRE\_DATE, LAST\_DAY(HIRE\_DATE) FROM employees;**
  
20. Write a query to get first name, hire date and experience of the employees.  
**SELECT FIRST\_NAME, HIRE\_DATE, NOW(), DATEDIFF(NOW(), HIRE\_DATE) DIV 365 AS YEAR\_OF\_EXPERIENCE FROM employees;**
  
21. Write a query to get the department ID, year, and number of employees joined.  
**SELECT DEPARTMENT\_ID, YEAR(HIRE\_DATE) AS JOININGYEAR, COUNT(\*) FROM employees GROUP BY DEPARTMENT\_ID, YEAR(HIRE\_DATE);**

## MySQL string [17 exercises]

1. Write a [MySQL](#) query to get the job\_id and related employee's id.  
 Partial output of the query :  
**SELECT job\_id, GROUP\_CONCAT (employee\_id, ' ') AS 'Employees ID' FROM employees GROUP BY job\_id;**
  
2. Write a [MySQL](#) query to update the portion of the phone\_number in the employees table, within the phone number the substring '124' will be replaced by '999'.  
**UPDATE employees SET phone\_number = REPLACE (phone\_number, '124', '999') WHERE phone\_number LIKE '%124%';**
  
3. Write a [MySQL](#) query to get the details of the employees where the length of the first name greater than or equal to 8.  
**SELECT \* FROM employees WHERE LENGTH(FIRST\_NAME) >= 8;**
  
4. Write a [MySQL](#) query to display leading zeros before maximum and minimum salary.  
**SELECT job\_id, LPAD (max\_salary, 7, '0') AS ' Max Salary', LPAD (min\_salary, 7, '0') AS ' Min Salary' FROM jobs;**
  
5. Write a MySQL query to append '@example.com' to email field.

**UPDATE employees SET email = CONCAT (email, '@gmail.com');**

6. Write a [MySQL](#) query to get the employee id, first name and hire month.  
**SELECT employee\_id, first\_name, MID (hire\_date, 6, 2) as hire\_month FROM employees;**
7. Write a [MySQL](#) query to find all employees where first names are in upper case.  
**SELECT \* FROM `employees` WHERE first\_name = BINARY UPPER (first\_name);**
8. Write a [MySQL](#) query to get the employee id, email id (discard the last three characters).  
**SELECT employee\_id, REVERSE (SUBSTR (REVERSE (email), 4)) as Email\_ID FROM employees;**
9. Write a [MySQL](#) query to extract the last 4 character of phone numbers.  
**SELECT RIGHT (phone\_number, 4) as 'Ph.No.' FROM employees;**
10. Write a [MySQL](#) query to get the last word of the street address.  
**SELECT location\_id, street\_address, SUBSTRING\_INDEX (REPLACE (REPLACE (REPLACE (street\_address, ',', ''), ' '); ' '), '(', ' '), -1) AS 'Last--word-of-street\_address' FROM locations;**
11. Write a MySQL query to get the locations that have minimum street length.  
**SELECT \* FROM `locations` WHERE LENGTH (street\_address) <= (SELECT MIN (LENGTH (street\_address)) FROM locations);**
12. Write a MySQL query to display the first word from those job titles which contains more than one words.  
**SELECT job\_title, SUBSTR (job\_title, 1, INSTR (job\_title, ' ') - 1) FROM jobs;**
13. Write a MySQL query to display the first name and last name for employees where first occurrence of last name contain character 'c' after 2nd position.  
**SELECT FIRST\_NAME, LAST\_NAME FROM employees WHERE INSTR (last\_name, 'C') > 2;**
14. Write a [MySQL](#) query that displays the first name and the length of the first name for all employees whose name starts with the letters 'A', 'J' or 'M'. Give each column an appropriate label. Sort the results by the employees' first names.  
**SELECT first\_name AS "Name", LENGTH (first\_name) AS "Length" FROM employees WHERE first\_name LIKE 'J%' OR first\_name LIKE 'M%' OR first\_name LIKE 'A%' ORDER BY first\_name;**
15. Write a [MySQL](#) query to display the first name and salary for all employees. Format the salary to be 10 characters long, left-padded with the \$ symbol. Label the column SALARY.  
**SELECT first\_name, LPAD (salary, 10, '\$') AS SALARY FROM employees;**



16. Write a [MySQL](#) query to display the first eight characters of the employees' first names and indicates the amounts of their salaries with '\$' sign. Each '\$' sign signifies a thousand dollars. Sort the data in descending order of salary.

```
SELECT LEFT (first_name, 8), REPEAT ('$ ', FLOOR (salary/1000)) AS 'SALARY ($)',  
salary FROM employees ORDER BY salary DESC;
```

17. Write a [MySQL](#) query to display the employees with their code, first name, last name and hire date who hired either on seventh day of any month or seventh month in any year.

```
SELECT employee_id, first_name, last_name, hire_date FROM employees WHERE  
POSITION ('07' IN DATE_FORMAT (hire_date, '%d %m %Y')) >0;
```

## **MySQL Subquery [22 exercises]**

1. Write a MySQL query to find the name (first\_name, last\_name) and the salary of the employees who have a higher salary than the employee whose last\_name='Bull'.  
**SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM employees WHERE SALARY > (SELECT salary FROM employees WHERE last\_name = 'Bull');**
2. Write a MySQL query to find the name (first\_name, last\_name) of all employees who works in the IT department.  
**SELECT FIRST\_NAME, LAST\_NAME FROM employees WHERE department\_id IN (SELECT department\_id FROM departments WHERE department\_name='IT');**
3. Write a MySQL query to find the name (first\_name, last\_name) of the employees who have a manager and worked in a USA based department.  
**SELECT first\_name, last\_name FROM employees WHERE manager\_id in (SELECT employee\_id FROM employees WHERE department\_id IN (SELECT department\_id FROM departments WHERE location\_id IN (SELECT location\_id FROM locations WHERE country\_id='US')));**
4. Write a MySQL query to find the name (first\_name, last\_name) of the employees who are managers.  
**SELECT FIRST\_NAME, LAST\_NAME FROM employees WHERE (employee\_id IN (SELECT manager\_id FROM employees));**
5. Write a MySQL query to find the name (first\_name, last\_name), and salary of the employees whose salary is greater than the average salary.  
**SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);**

6. Write a MySQL query to find the name (first\_name, last\_name), and salary of the employees whose salary is equal to the minimum salary for their job grade.  
**SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM employees WHERE employees.salary = (SELECT min\_salary FROM jobs WHERE employees.job\_id = jobs.job\_id);**
7. Write a MySQL query to find the name (first\_name, last\_name), and salary of the employees who earns more than the average salary and works in any of the IT departments.  
**SELECT first\_name, last\_name, salary FROM employees WHERE department\_id IN (SELECT department\_id FROM departments WHERE department\_name LIKE 'IT%') AND salary > (SELECT avg(salary) FROM employees);**
8. Write a MySQL query to find the name (first\_name, last\_name), and salary of the employees who earns more than the earning of Mr. Bell.  
**SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM employees WHERE SALARY > (SELECT SALARY FROM employees WHERE LAST\_NAME = 'Bell') ORDER BY FIRST\_NAME;**
9. Write a MySQL query to find the name (first\_name, last\_name), and salary of the employees who earn the same salary as the minimum salary for all departments.  
**SELECT \* FROM employees WHERE SALARY = (SELECT MIN(SALARY) FROM employees);**
10. Write a MySQL query to find the name (first\_name, last\_name), and salary of the employees whose salary is greater than the average salary of each department.  
**SELECT first\_name, last\_name, salary FROM employees WHERE salary > (SELECT AVG (salary) FROM employees AS e2 WHERE e2.department\_id = employees.department\_id);**
11. Write a MySQL query to find the name (first\_name, last\_name) and salary of the employees who earn a salary that is higher than the salary of all the Shipping Clerk (JOB\_ID = 'SH\_CLERK'). Sort the results of the salary of the lowest to highest.  
**SELECT first\_name, last\_name, job\_id, salary FROM employees WHERE salary > ALL (SELECT salary FROM employees WHERE job\_id = 'SH\_CLERK') ORDER BY salary;**
12. Write a MySQL query to find the name (first\_name, last\_name) of the employees who are not supervisors.  
**SELECT first\_name, last\_name FROM employees WHERE employee\_id NOT IN (SELECT DISTINCT JOB\_TITLE FROM jobs WHERE JOB\_TITLE IS NOT NULL);**
13. Write a MySQL query to display the employee ID, first name, last name, and department names of all employees.  
**SELECT employee\_id, first\_name, last\_name, (SELECT department\_name FROM departments d WHERE e.department\_id = d.department\_id) department FROM employees e ORDER BY department;**
14. Write a MySQL query to display the employee ID, first name, last name, and department names of all employees.  
**SELECT employee\_id, first\_name FROM employees AS A WHERE salary > (SELECT AVG (salary) FROM employees WHERE department\_id = A.department\_id);**

15. Write a MySQL query to fetch even numbered records from employees table.  
**SELECT first\_name, last\_name FROM (SELECT first\_name, last\_name, ROW\_NUMBER ()  
OVER (ORDER BY employee\_id) AS row\_num FROM employees) AS subquery WHERE  
row\_num % 2 = 0;**
16. Write a MySQL query to find the 5th maximum salary in the employees table.  
**SELECT DISTINCT salary  
FROM employees e1  
WHERE 5 =(SELECT COUNT(DISTINCT salary)  
FROM employees e2  
WHERE e2.salary >= e1.salary);**
17. Write a MySQL query to find the 4th minimum salary in the employees table.  
**SELECT DISTINCT salary  
FROM employees e1  
WHERE 4 =(SELECT COUNT(DISTINCT salary)  
FROM employees e2  
WHERE e2.salary <= e1.salary);**
18. Write a MySQL query to select last 10 records from a table.  
**SELECT \* FROM (SELECT \* FROM employees ORDER BY employee\_id DESC LIMIT 10)  
sub ORDER BY employee\_id ASC;**
19. Write a MySQL query to list the department ID and name of all the departments where  
no employee is working.  
**SELECT \* FROM departments WHERE department\_id NOT IN (SELECT department\_id  
FROM employees);**
20. Write a MySQL query to get 3 maximum salaries.  
**SELECT DISTINCT salary FROM employees a  
WHERE 3 >= (SELECT COUNT(DISTINCT salary) FROM employees b WHERE b.salary >=  
a.salary) ORDER BY a.salary DESC;**
21. Write a MySQL query to get 3 minimum salaries.  
**SELECT DISTINCT salary FROM employees a  
WHERE 3 >= (SELECT COUNT(DISTINCT salary) FROM employees b WHERE b.salary <=  
a.salary) ORDER BY a.salary DESC;**
22. Write a MySQL query to get nth max salaries of employees.  
**SELECT \* FROM employees emp1  
WHERE (1) = (SELECT COUNT(DISTINCT(emp2.salary)) FROM employees emp2 WHERE  
emp2.salary > emp1.salary);**

## MySQL Joins [13 exercises]

1. Write a MySQL query to find the addresses (location\_id, street\_address, city, state\_province, country\_name) of all the departments.  
Hint : Use NATURAL JOIN.  
**SELECT location\_id, street\_address, city, state\_province, country\_name FROM locations NATURAL JOIN countries;**
2. Write a MySQL query to find the name (first\_name, last name), department ID and name of all the employees.  
**SELECT first\_name, last\_name, E.DEPARTMENT\_ID, D.DEPARTMENT\_NAME FROM employees AS E INNER JOIN departments AS D ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID;**
3. Write a [MySQL](#) query to find the name (first\_name, last\_name), job, department ID and name of the employees who works in London.  
**SELECT first\_name, last\_name, E.DEPARTMENT\_ID, D.DEPARTMENT\_NAME, J.JOB\_TITLE FROM employees AS E INNER JOIN departments AS D ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID INNER JOIN jobs AS J ON J.JOB\_ID = E.JOB\_ID INNER JOIN locations AS L ON L.LOCATION\_ID = D.LOCATION\_ID WHERE L.CITY = 'LONDON';**
4. Write a [MySQL](#) query to find the employee id, name (last\_name) along with their manager\_id and name (last\_name).  
**SELECT E.EMPLOYEE\_ID, E.LAST\_NAME, M.EMPLOYEE\_ID, M.LAST\_NAME FROM employees AS E INNER JOIN employees AS M ON E.MANAGER\_ID = M.EMPLOYEE\_ID;**
5. Write a [MySQL](#) query to find the name (first\_name, last\_name) and hire date of the employees who was hired after 'Jones'.  
**SELECT E1.FIRST\_NAME, E1.LAST\_NAME, E1.HIRE\_DATE FROM employees AS E1 INNER JOIN employees AS E2 ON E2.LAST\_NAME = 'Jones' WHERE E1.HIRE\_DATE > E2.HIRE\_DATE;**
6. Write a [MySQL](#) query to get the department name and number of employees in the department.  
**SELECT D.DEPARTMENT\_ID, D.DEPARTMENT\_NAME, COUNT(\*) AS employeeCount FROM departments AS D INNER JOIN employees AS E ON D.DEPARTMENT\_ID = E.DEPARTMENT\_ID GROUP BY D.DEPARTMENT\_ID, D.DEPARTMENT\_NAME;**
7. Write a [MySQL](#) query to find the employee ID, job title, number of days between ending date and starting date for all jobs in department 90.  
**SELECT E.EMPLOYEE\_ID, E.FIRST\_NAME, (JH.END\_DATE - JH.START\_DATE) AS DAYS FROM employees AS E INNER JOIN jobs AS J ON E.JOB\_ID = J.JOB\_ID INNER JOIN job\_history AS JH ON E.EMPLOYEE\_ID = JH.EMPLOYEE\_ID AND J.JOB\_ID = JH.JOB\_ID WHERE JH.department\_id=90;**

8. Write a [MySQL](#) query to display the department ID and name and first name of manager.

```
SELECT D.DEPARTMENT_ID, D.DEPARTMENT_NAME, E.FIRST_NAME AS  
MANAGER_NAME FROM departments AS D INNER JOIN employees AS E ON  
D.MANAGER_ID = E.EMPLOYEE_ID;
```

9. Write a [MySQL](#) query to display the department name, manager name, and city.
- ```
SELECT D.DEPARTMENT_ID, D.DEPARTMENT_NAME, E.FIRST_NAME AS  
MANAGER_NAME, L.CITY FROM departments AS D INNER JOIN employees AS E ON  
D.MANAGER_ID = E.EMPLOYEE_ID INNER JOIN locations AS L ON D.LOCATION_ID =  
L.LOCATION_ID;
```

10. Write a [MySQL](#) query to display the job title and average salary of employees.
- ```
SELECT J.JOB_TITLE, AVG(E.SALARY) AS AVGSALARY FROM employees AS E INNER  
JOIN jobs AS J ON E.JOB_ID = J.JOB_ID GROUP BY J.JOB_TITLE;
```

11. Write a [MySQL](#) query to display job title, employee name, and the difference between salary of the employee and minimum salary for the job.
- ```
SELECT J.JOB_TITLE, E.FIRST_NAME, E.SALARY - J.MIN_SALARY AS DIFFSALARY  
FROM employees AS E INNER JOIN jobs AS J ON E.JOB_ID = J.JOB_ID;
```

12. Write a [MySQL](#) query to display the job history that were done by any employee who is currently drawing more than 10000 of salary.
- ```
SELECT first_name, last_name, hire_date, salary, (DATEDIFF(now(), hire_date))/365  
Experience FROM departments d JOIN employees e ON (d.manager_id =  
e.employee_id) WHERE (DATEDIFF(now(), hire_date))/365>15;
```

13. Write a [MySQL](#) query to display department name, name (first\_name, last\_name), hire date, salary of the manager for all managers whose experience is more than 15 years.
- ```
SELECT first_name, last_name, hire_date, salary, (DATEDIFF(now(), hire_date))/365  
Experience FROM departments d JOIN employees e ON (d.manager_id =  
e.employee_id) WHERE (DATEDIFF(now(), hire_date))/365>15;
```

