SQL Commands

The **SQL** commands are mainly categorized into four categories as:

1. **DDL** – Data Definition Language
2. **DML** – Data Manipulation Language
3. **DQL** – Data Query Language
4. **TCL** – Transaction Control Language

1. DDL(Data Definition Language) :

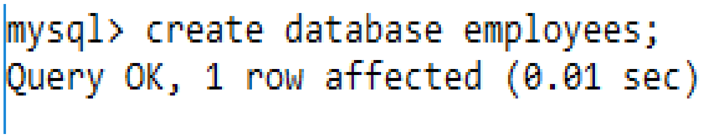
DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

Examples of DDL commands:

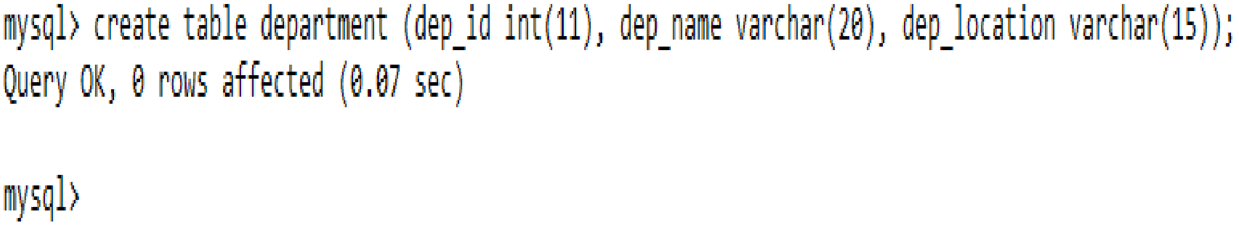
(a) CREATE – is used to create the database or its objects (like table, index, function, views, store procedure and triggers).

Example: `CREATE database employees;`

```
          CREATE TABLE department (dep_id int(11), dep_name
varchar(20),      dep_location varchar(15));
```

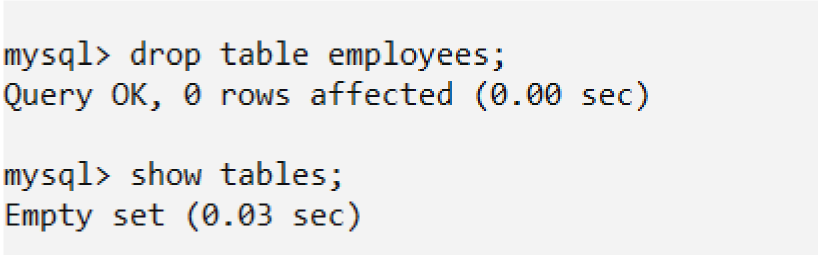


```
mysql> create database employees;
Query OK, 1 row affected (0.01 sec)
```



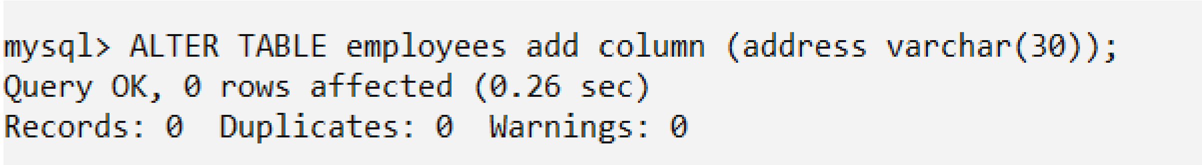
```
mysql> create table department (dep_id int(11), dep_name varchar(20), dep_location varchar(15));
Query OK, 0 rows affected (0.07 sec)

mysql>
```

(b) DROP – is used to delete objects from the database.
Example: `DROP table employees;`



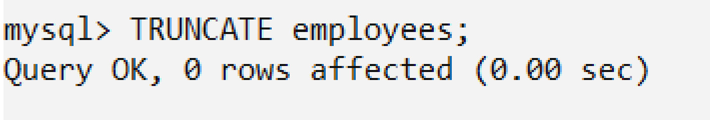
```
mysql> drop table employees;
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
Empty set (0.03 sec)
```

(c) ALTER - is used to alter the structure of the database.
Example: `ALTER TABLE employees add column (address varchar(30));`



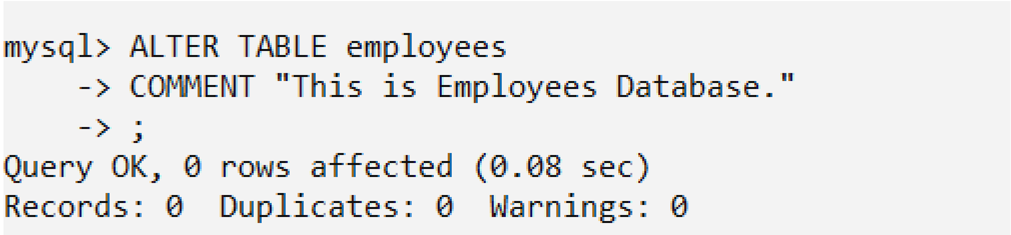
```
mysql> ALTER TABLE employees add column (address varchar(30));
Query OK, 0 rows affected (0.26 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

(d) TRUNCATE–is used to remove all records from a table, including all spaces allocated for the records are removed.
Example: `TRUNCATE employees;`



```
mysql> TRUNCATE employees;
Query OK, 0 rows affected (0.00 sec)
```

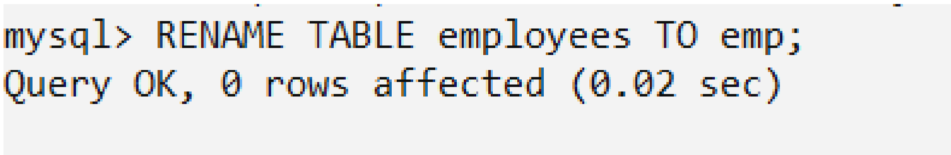
(e) COMMENT –is used to add comments to the data dictionary.
Example:
```
ALTER TABLE employees
          COMMENT “This is Employees Database.”;
```



```
mysql> ALTER TABLE employees
    -> COMMENT "This is Employees Database."
    -> ;
Query OK, 0 rows affected (0.08 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

(f) RENAME –is used to rename an object existing in the database.

Example: `RENAME TABLE employees TO emp;`



```
mysql> RENAME TABLE employees TO emp;
Query OK, 0 rows affected (0.02 sec)
```

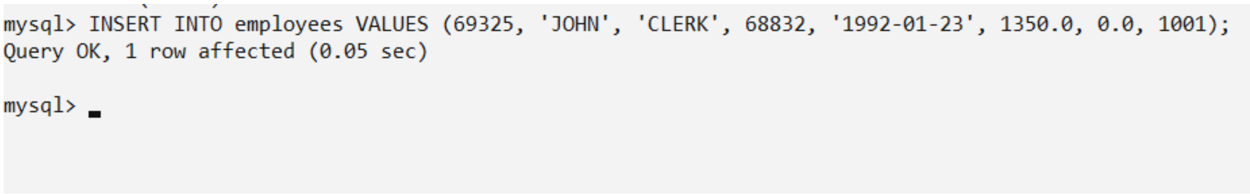
2. DML (Data Manipulation Language) :

The SQL commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements.

Examples of DML:

(a) INSERT – is used to insert data into a table.

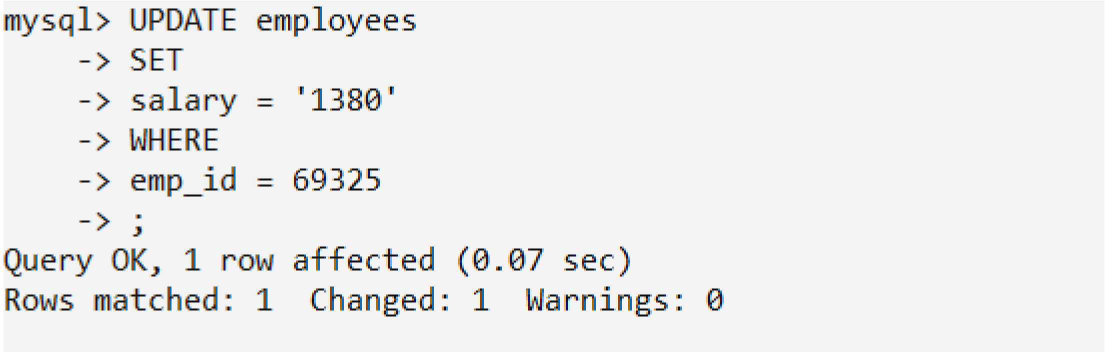
Example: `INSERT INTO employees VALUES (69325, 'JOHN', 'CLERK', 68832, '1992- 01-23', 1350.0, 0.0, 1001);`



```
mysql> INSERT INTO employees VALUES (69325, 'JOHN', 'CLERK', 68832, '1992-01-23', 1350.0, 0.0, 1001);
Query OK, 1 row affected (0.05 sec)

mysql>
```

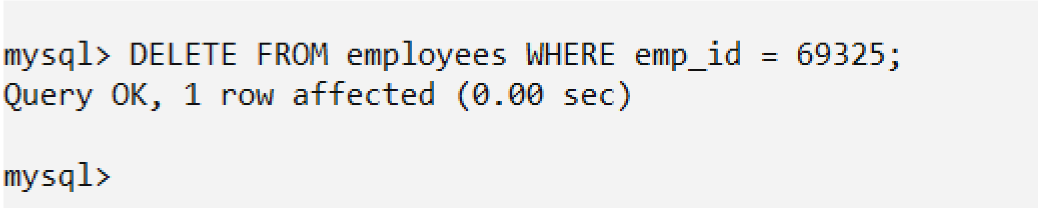
(b) UPDATE – is used to update existing data within a table.

Example:
```
UPDATE employees
          SET salary = '1380'  WHERE emp_id = 69325;
```



```
mysql> UPDATE employees
    -> SET
    -> salary = '1380'
    -> WHERE
    -> emp_id = 69325
    -> ;
Query OK, 1 row affected (0.07 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

(c) DELETE – is used to delete records from a database table.
Example: `DELETE FROM employees WHERE emp_id = 69325;`

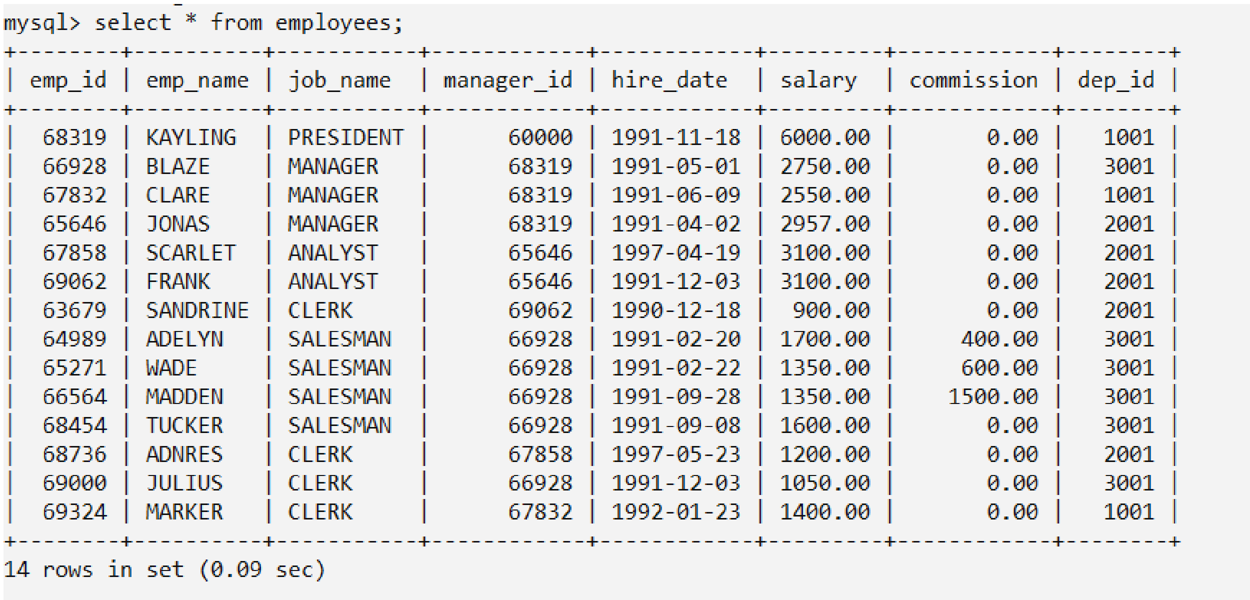


```
mysql> DELETE FROM employees WHERE emp_id = 69325;
Query OK, 1 row affected (0.00 sec)

mysql>
```

3. DQL (Data Query Language) :

DML statements are used for performing queries on the data within schema objects. The purpose of DQL Command is to get some schema relation based on the query passed to it.

Example of DQL:
(a) SELECT – is used to retrieve data from the database.
Example: `SELECT * FROM employees;`



```
mysql> select * from employees;
+--------+----------+-----------+------------+------------+---------+------------+--------+
| emp_id | emp_name | job_name  | manager_id | hire_date  | salary  | commission | dep_id |
+--------+----------+-----------+------------+------------+---------+------------+--------+
|  68319 | KAYLING  | PRESIDENT |      60000 | 1991-11-18 | 6000.00 |       0.00 |   1001 |
|  66928 | BLAZE    | MANAGER   |      68319 | 1991-05-01 | 2750.00 |       0.00 |   3001 |
|  67832 | CLARE    | MANAGER   |      68319 | 1991-06-09 | 2550.00 |       0.00 |   1001 |
|  65646 | JONAS    | MANAGER   |      68319 | 1991-04-02 | 2957.00 |       0.00 |   2001 |
|  67858 | SCARLET  | ANALYST   |      65646 | 1997-04-19 | 3100.00 |       0.00 |   2001 |
|  69062 | FRANK    | ANALYST   |      65646 | 1991-12-03 | 3100.00 |       0.00 |   2001 |
|  63679 | SANDRINE | CLERK     |      69062 | 1990-12-18 |  900.00 |       0.00 |   2001 |
|  64989 | ADELYN   | SALESMAN  |      66928 | 1991-02-20 | 1700.00 |     400.00 |   3001 |
|  65271 | WADE     | SALESMAN  |      66928 | 1991-02-22 | 1350.00 |     600.00 |   3001 |
|  66564 | MADDEN   | SALESMAN  |      66928 | 1991-09-28 | 1350.00 |    1500.00 |   3001 |
|  68454 | TUCKER   | SALESMAN  |      66928 | 1991-09-08 | 1600.00 |       0.00 |   3001 |
|  68736 | ADNRES   | CLERK     |      67858 | 1997-05-23 | 1200.00 |       0.00 |   2001 |
|  69000 | JULIUS   | CLERK     |      66928 | 1991-12-03 | 1050.00 |       0.00 |   3001 |
|  69324 | MARKER   | CLERK     |      67832 | 1992-01-23 | 1400.00 |       0.00 |   1001 |
+--------+----------+-----------+------------+------------+---------+------------+--------+
14 rows in set (0.09 sec)
```

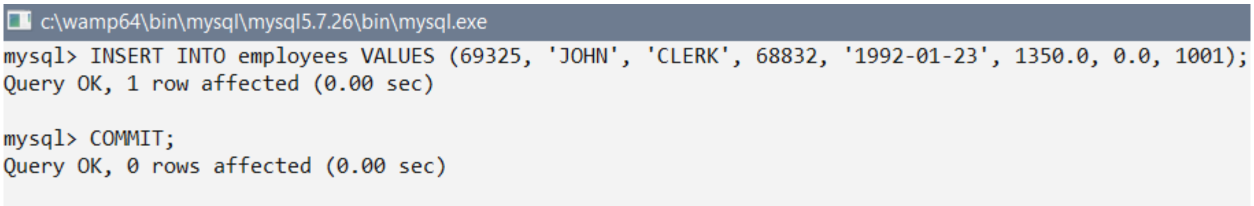
4. TCL (Transaction Control Language) :

TCL commands deals with the transaction within the database.

Examples of TCL commands:

(a) COMMIT– commits a Transaction.

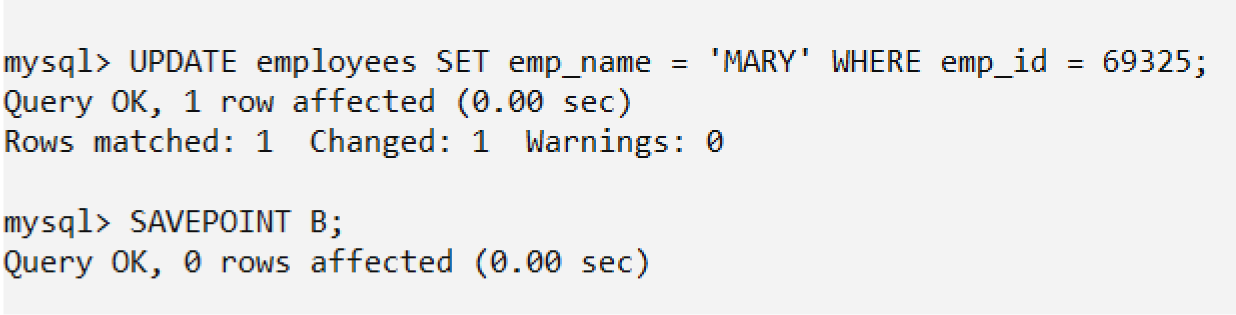
Example: `COMMIT;`



```
mysql> INSERT INTO employees VALUES (69325, 'JOHN', 'CLERK', 68832, '1992-01-23', 1350.0, 0.0, 1001);
Query OK, 1 row affected (0.00 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
```

(b) SAVEPOINT–sets a save point within a transaction.

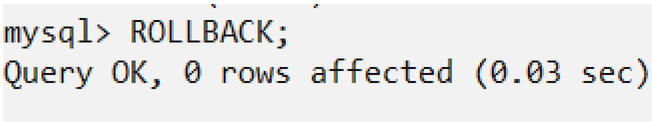
Example : `SAVEPOINT <savepoint_name>;`



```
mysql> UPDATE employees SET emp_name = 'MARY' WHERE emp_id = 69325;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SAVEPOINT B;
Query OK, 0 rows affected (0.00 sec)
```

(c) ROLLBACK– rollbacks a transaction in case of any error occurs.

Example: `ROLLBACK TO` <savepoint_name>;

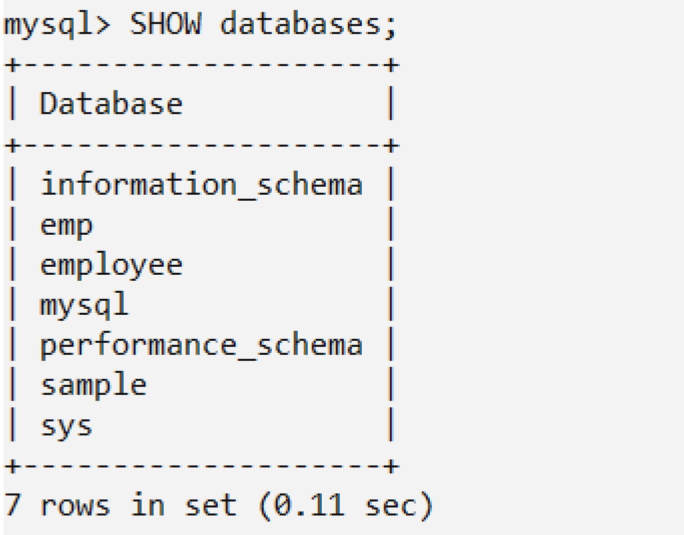


```
mysql> ROLLBACK;
Query OK, 0 rows affected (0.03 sec)
```

Some other Basic Commands

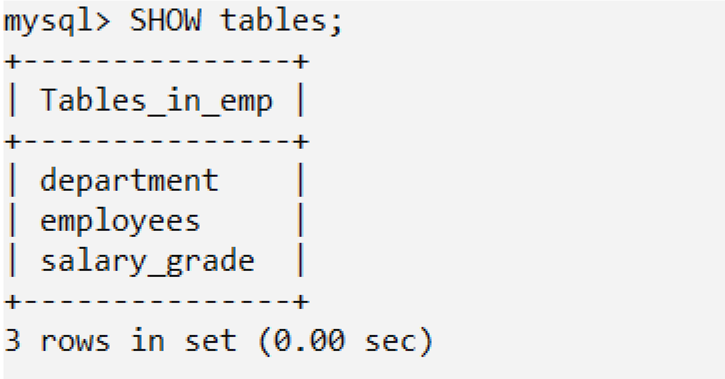
1. SHOW : It is used to view the contents of databases.

Example : `SHOW databases;`



```
mysql> SHOW databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| emp                |
| employee           |
| mysql              |
| performance_schema |
| sample             |
| sys                |
+--------------------+
7 rows in set (0.11 sec)
```

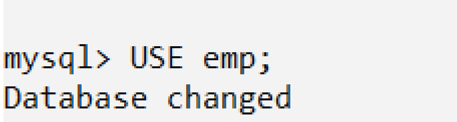
`SHOW tables;`



```
mysql> SHOW tables;
+---------------+
| Tables_in_emp |
+---------------+
| department    |
| employees     |
| salary_grade  |
+---------------+
3 rows in set (0.00 sec)
```

2. USE : It is used to use the desired database.

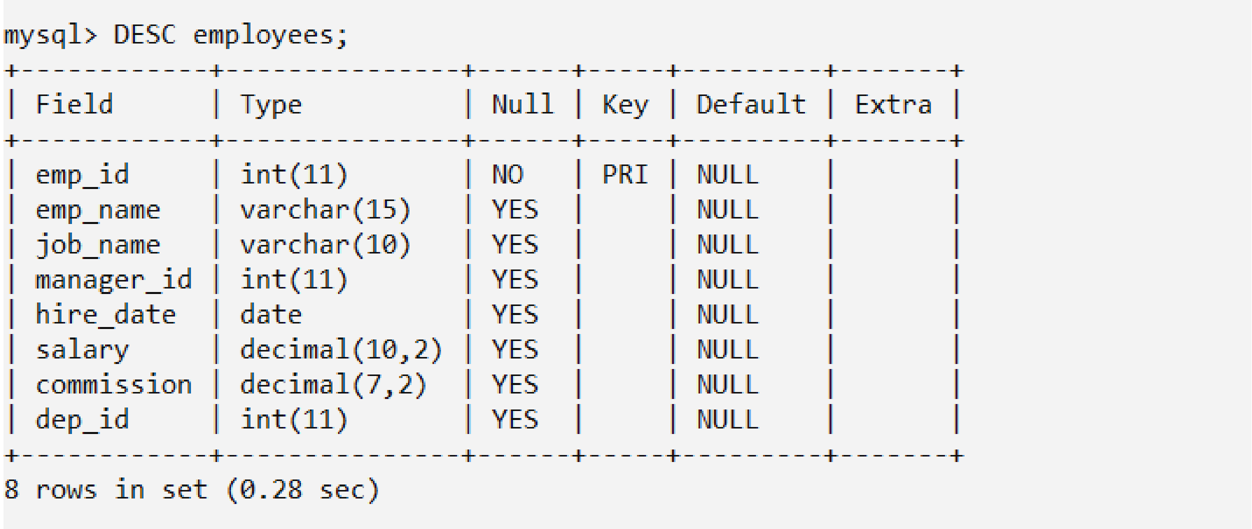
Example : `USE emp;`



```
mysql> USE emp;
Database changed
```

3. DESC : It is used to view the description of any table.

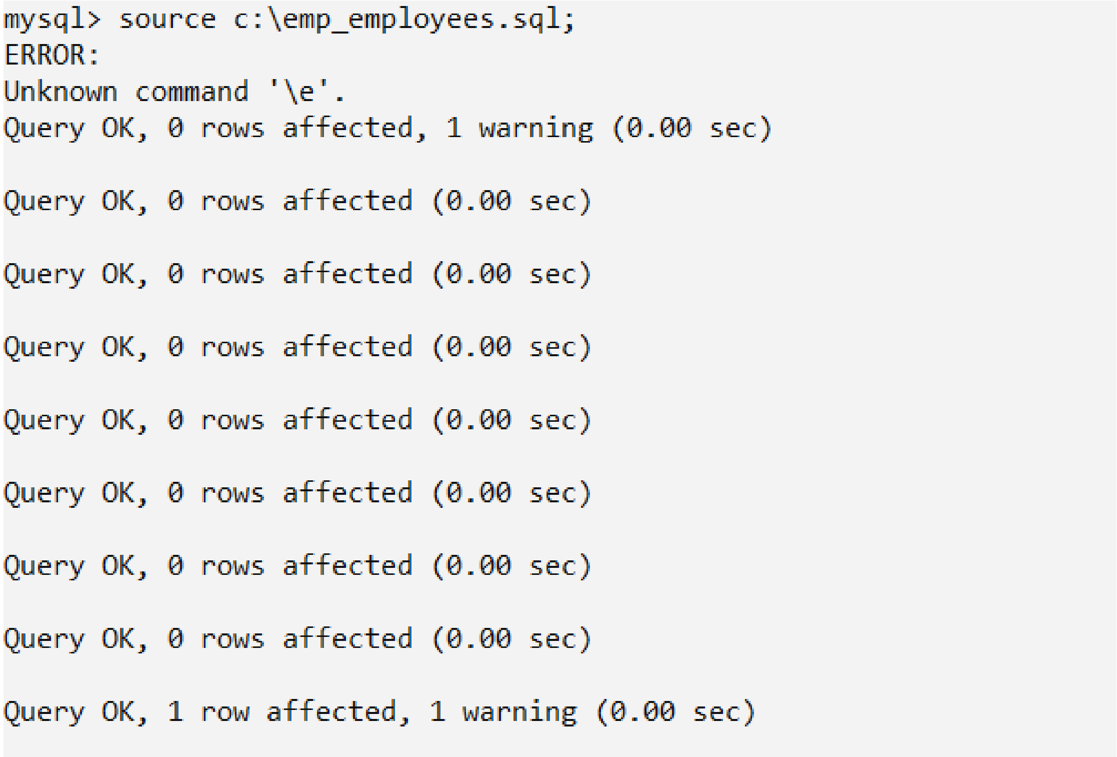
Examole : `DESC employees;`



```
mysql> DESC employees;
+------------+---------------+------+-----+---------+-------+
| Field      | Type          | Null | Key | Default | Extra |
+------------+---------------+------+-----+---------+-------+
| emp_id     | int(11)       | NO   | PRI | NULL    |       |
| emp_name   | varchar(15)   | YES  |     | NULL    |       |
| job_name   | varchar(10)   | YES  |     | NULL    |       |
| manager_id | int(11)       | YES  |     | NULL    |       |
| hire_date  | date          | YES  |     | NULL    |       |
| salary     | decimal(10,2) | YES  |     | NULL    |       |
| commission | decimal(7,2)  | YES  |     | NULL    |       |
| dep_id     | int(11)       | YES  |     | NULL    |       |
+------------+---------------+------+-----+---------+-------+
8 rows in set (0.28 sec)
```

4. SOURCE : It is used to import any database.

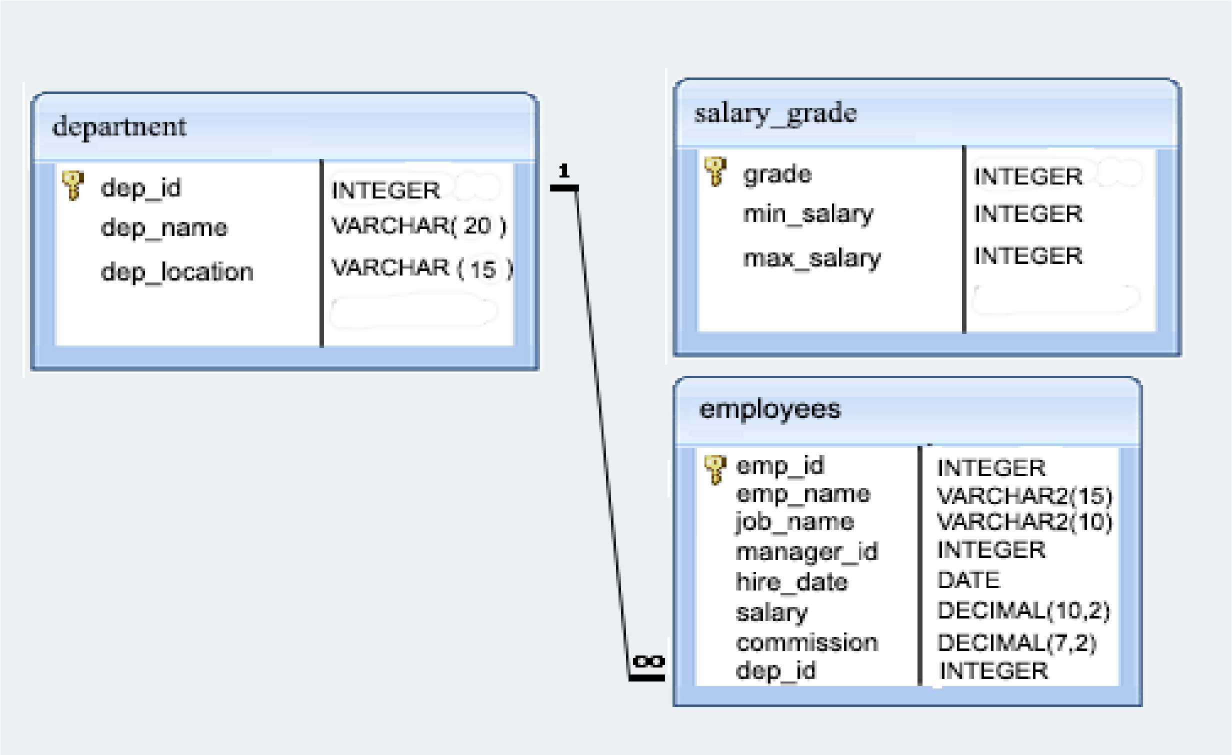
Example : `SOURCE c:\emp_employees.sql;`



```
mysql> source c:\emp_employees.sql;
ERROR:
Unknown command '\e'.
Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 1 row affected, 1 warning (0.00 sec)
```

Employee Database

Structure of Employee Database



department

dep_id	INTEGER
dep_name	VARCHAR(20)
dep_location	VARCHAR (15)

salary_grade

grade	INTEGER
min_salary	INTEGER
max_salary	INTEGER

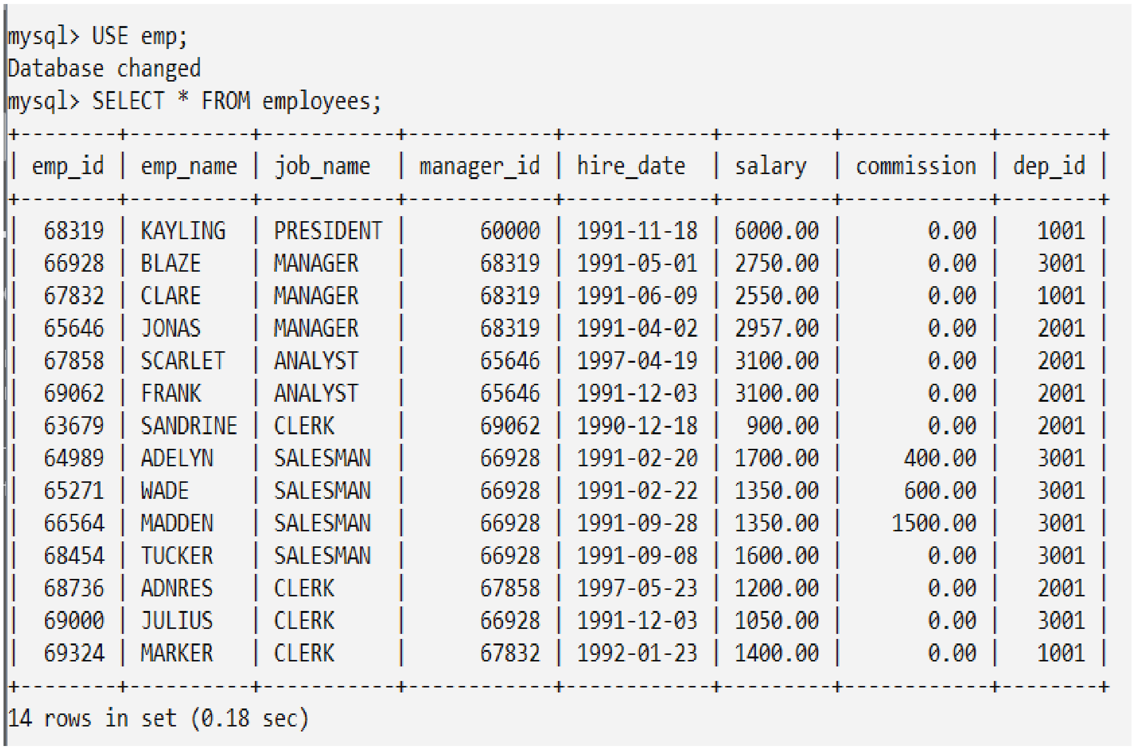
employees

emp_id	INTEGER
emp_name	VARCHAR2(15)
job_name	VARCHAR2(10)
manager_id	INTEGER
hire_date	DATE
salary	DECIMAL(10,2)
commission	DECIMAL(7,2)
dep_id	INTEGER

Task - 1

Write a query in SQL to display all the information of the employees.

Command: `SELECT * FROM employees;`

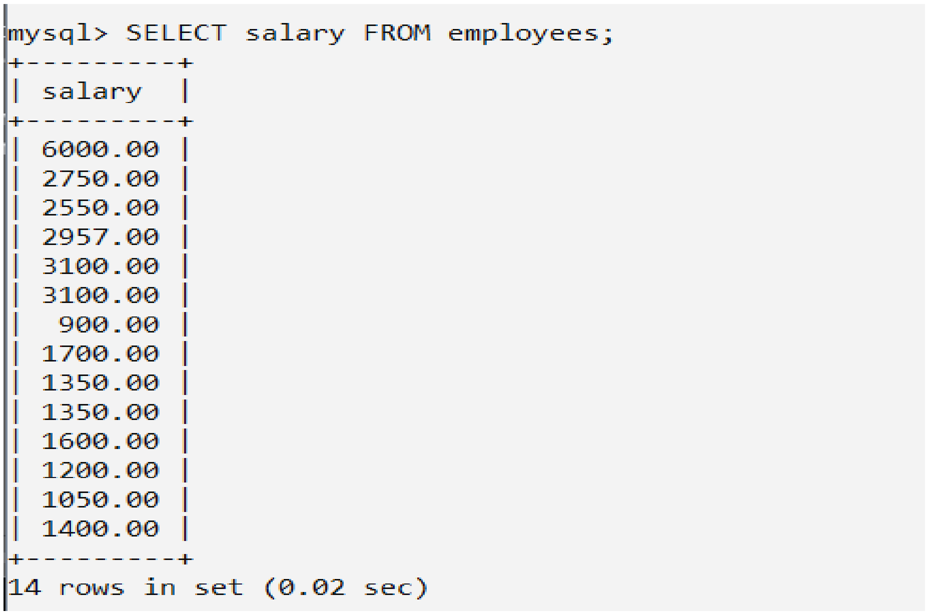


```
mysql> USE emp;
Database changed
mysql> SELECT * FROM employees;
+--------+----------+-----------+------------+------------+---------+------------+--------+
| emp_id | emp_name | job_name  | manager_id | hire_date  | salary  | commission | dep_id |
+--------+----------+-----------+------------+------------+---------+------------+--------+
|  68319 | KAYLING  | PRESIDENT |      60000 | 1991-11-18 | 6000.00 |       0.00 |   1001 |
|  66928 | BLAZE    | MANAGER   |      68319 | 1991-05-01 | 2750.00 |       0.00 |   3001 |
|  67832 | CLARE    | MANAGER   |      68319 | 1991-06-09 | 2550.00 |       0.00 |   1001 |
|  65646 | JONAS    | MANAGER   |      68319 | 1991-04-02 | 2957.00 |       0.00 |   2001 |
|  67858 | SCARLET  | ANALYST   |      65646 | 1997-04-19 | 3100.00 |       0.00 |   2001 |
|  69062 | FRANK    | ANALYST   |      65646 | 1991-12-03 | 3100.00 |       0.00 |   2001 |
|  63679 | SANDRINE | CLERK     |      69062 | 1990-12-18 |  900.00 |       0.00 |   2001 |
|  64989 | ADELYN   | SALESMAN  |      66928 | 1991-02-20 | 1700.00 |     400.00 |   3001 |
|  65271 | WADE     | SALESMAN  |      66928 | 1991-02-22 | 1350.00 |     600.00 |   3001 |
|  66564 | MADDEN   | SALESMAN  |      66928 | 1991-09-28 | 1350.00 |    1500.00 |   3001 |
|  68454 | TUCKER   | SALESMAN  |      66928 | 1991-09-08 | 1600.00 |       0.00 |   3001 |
|  68736 | ADNRES   | CLERK     |      67858 | 1997-05-23 | 1200.00 |       0.00 |   2001 |
|  69000 | JULIUS   | CLERK     |      66928 | 1991-12-03 | 1050.00 |       0.00 |   3001 |
|  69324 | MARKER   | CLERK     |      67832 | 1992-01-23 | 1400.00 |       0.00 |   1001 |
+--------+----------+-----------+------------+------------+---------+------------+--------+
14 rows in set (0.18 sec)
```

Task - 2

Write a query in SQL to find the salaries of all employees.

Command: `SELECT salary FROM employees;`

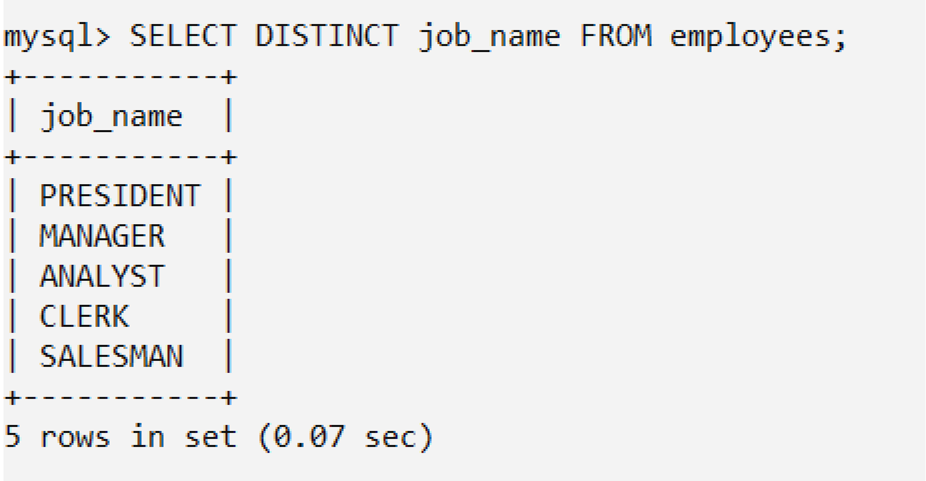


```
mysql> SELECT salary FROM employees;
+---------+
| salary  |
+---------+
| 6000.00 |
| 2750.00 |
| 2550.00 |
| 2957.00 |
| 3100.00 |
| 3100.00 |
|  900.00 |
| 1700.00 |
| 1350.00 |
| 1350.00 |
| 1600.00 |
| 1200.00 |
| 1050.00 |
| 1400.00 |
+---------+
14 rows in set (0.02 sec)
```

Task - 3

Write a query in SQL to display the unique designations for the employees.

Command: `SELECT DISTINCT job_name FROM employees;`

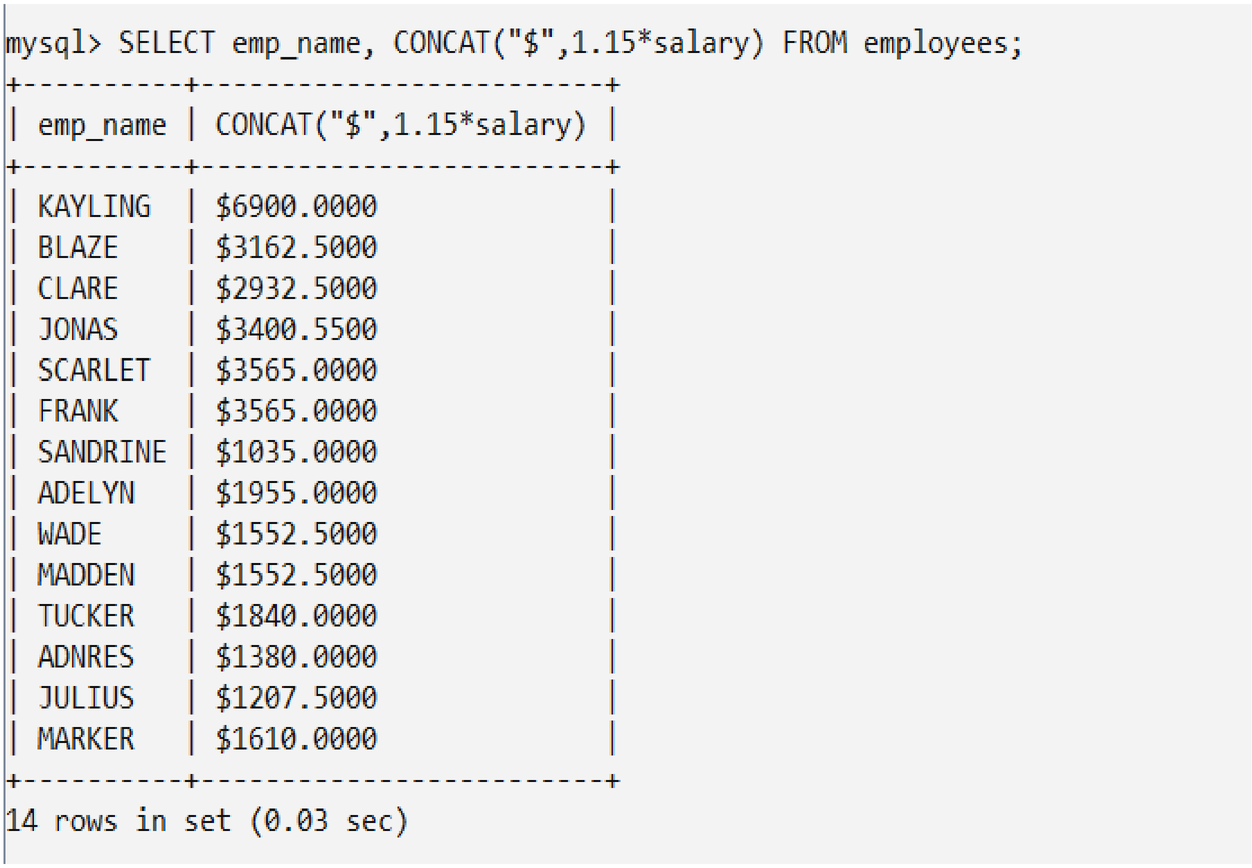


```
mysql> SELECT DISTINCT job_name FROM employees;
+-----------+
| job_name  |
+-----------+
| PRESIDENT |
| MANAGER   |
| ANALYST   |
| CLERK     |
| SALESMAN  |
+-----------+
5 rows in set (0.07 sec)
```

Task - 4

Write a query in SQL to list the emp_name and salary is increased by 15% and expressed as no.of Dollars.

Command: `SELECT emp_name, CONCAT(”$”,1.15*salary) FROM employees;`

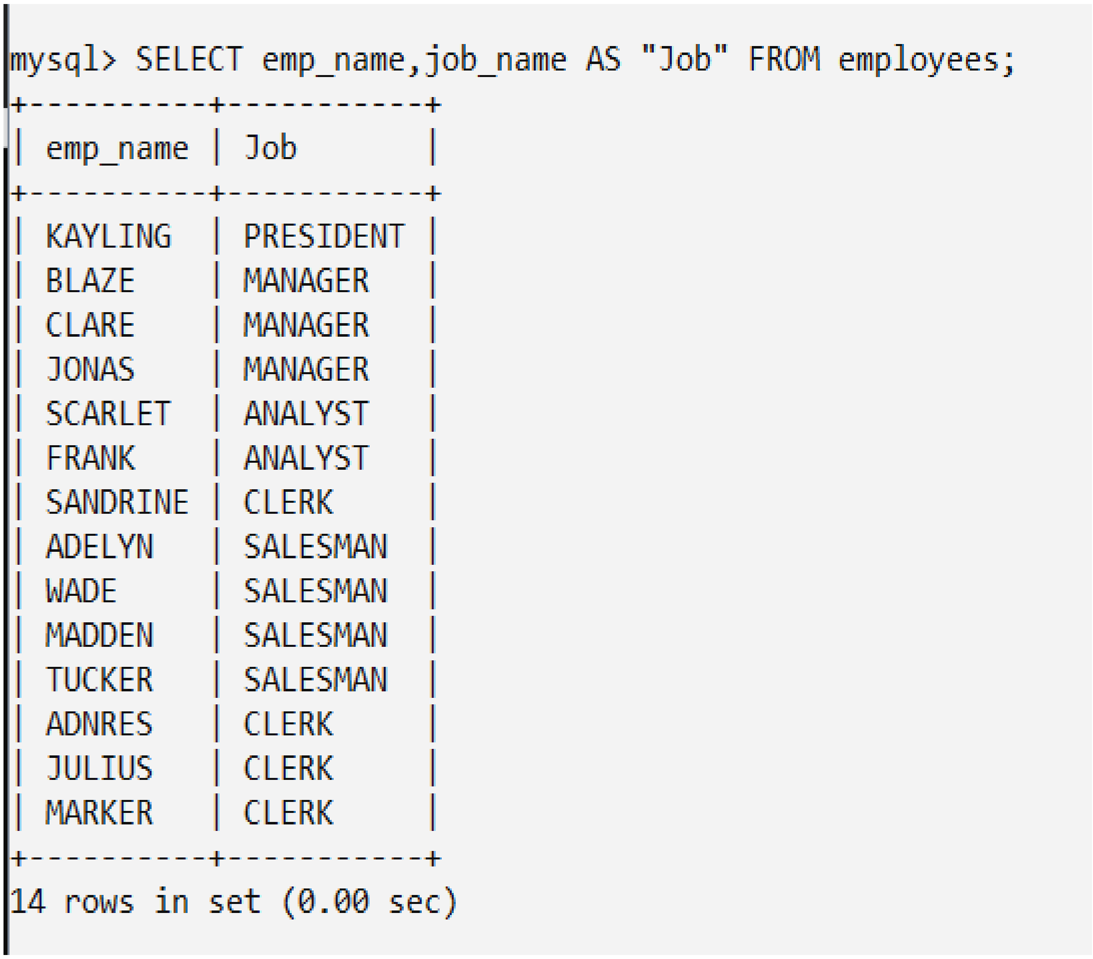


```
mysql> SELECT emp_name, CONCAT("$",1.15*salary) FROM employees;
+----------+-------------------------+
| emp_name | CONCAT("$",1.15*salary) |
+----------+-------------------------+
| KAYLING  | $6900.0000              |
| BLAZE    | $3162.5000              |
| CLARE    | $2932.5000              |
| JONAS    | $3400.5500              |
| SCARLET  | $3565.0000              |
| FRANK    | $3565.0000              |
| SANDRINE | $1035.0000              |
| ADELYN   | $1955.0000              |
| WADE     | $1552.5000              |
| MADDEN   | $1552.5000              |
| TUCKER   | $1840.0000              |
| ADNRES   | $1380.0000              |
| JULIUS   | $1207.5000              |
| MARKER   | $1610.0000              |
+----------+-------------------------+
14 rows in set (0.03 sec)
```

Task - 5

Write a query in SQL to produce the output of employees name and job name as a format of "Employee & Job".

Command: `SELECT emp_name,job_name AS “Job” FROM employees;`

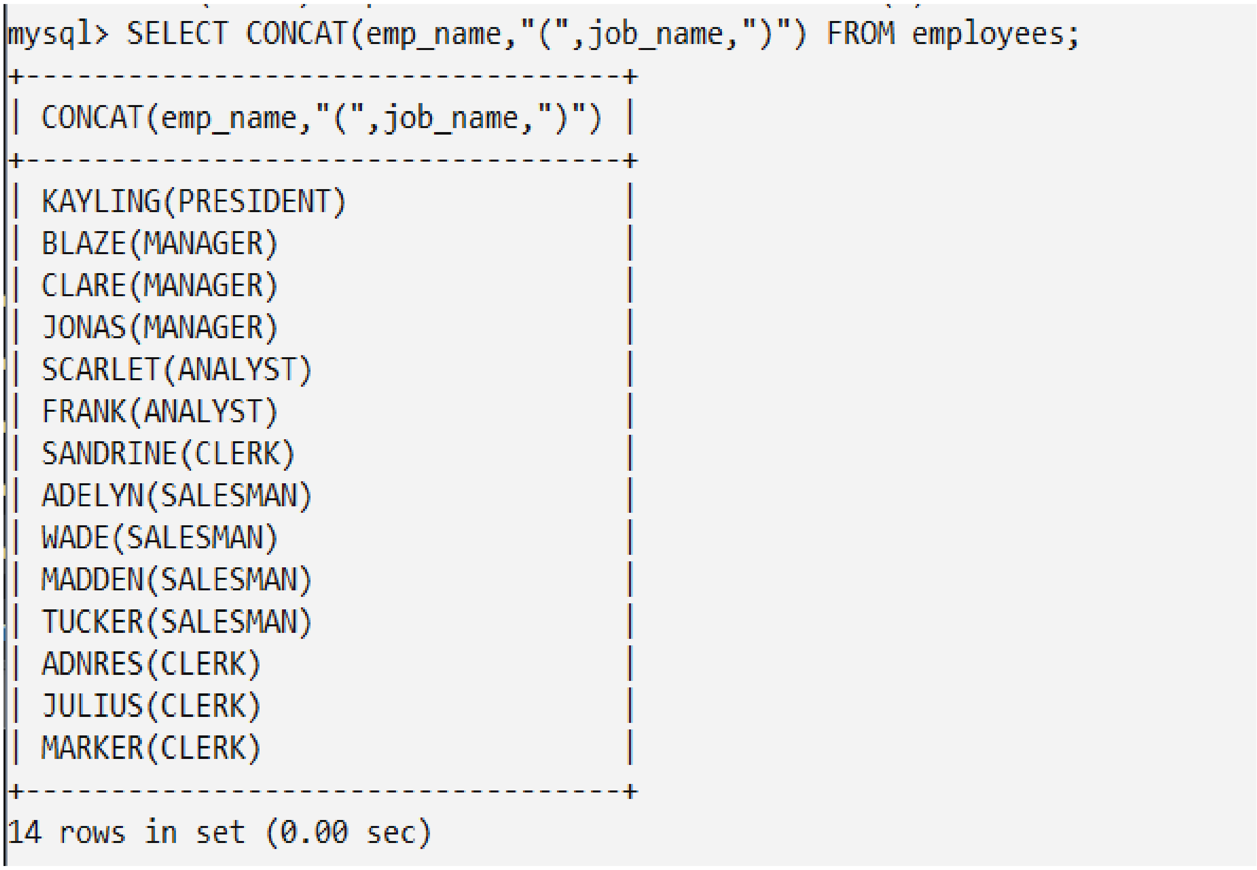


```
mysql> SELECT emp_name,job_name AS "Job" FROM employees;
+----------+-----------+
| emp_name | Job       |
+----------+-----------+
| KAYLING  | PRESIDENT |
| BLAZE    | MANAGER   |
| CLARE    | MANAGER   |
| JONAS    | MANAGER   |
| SCARLET  | ANALYST   |
| FRANK    | ANALYST   |
| SANDRINE | CLERK     |
| ADELYN   | SALESMAN  |
| WADE     | SALESMAN  |
| MADDEN   | SALESMAN  |
| TUCKER   | SALESMAN  |
| ADNRES   | CLERK     |
| JULIUS   | CLERK     |
| MARKER   | CLERK     |
+----------+-----------+
14 rows in set (0.00 sec)
```

Task - 6

Write a query in SQL to produce the output of employees as follows Employee JONAS(manager).

Command: `SELECT CONCAT(emp_name,”(”,job_name,”)”) FROM employees;`

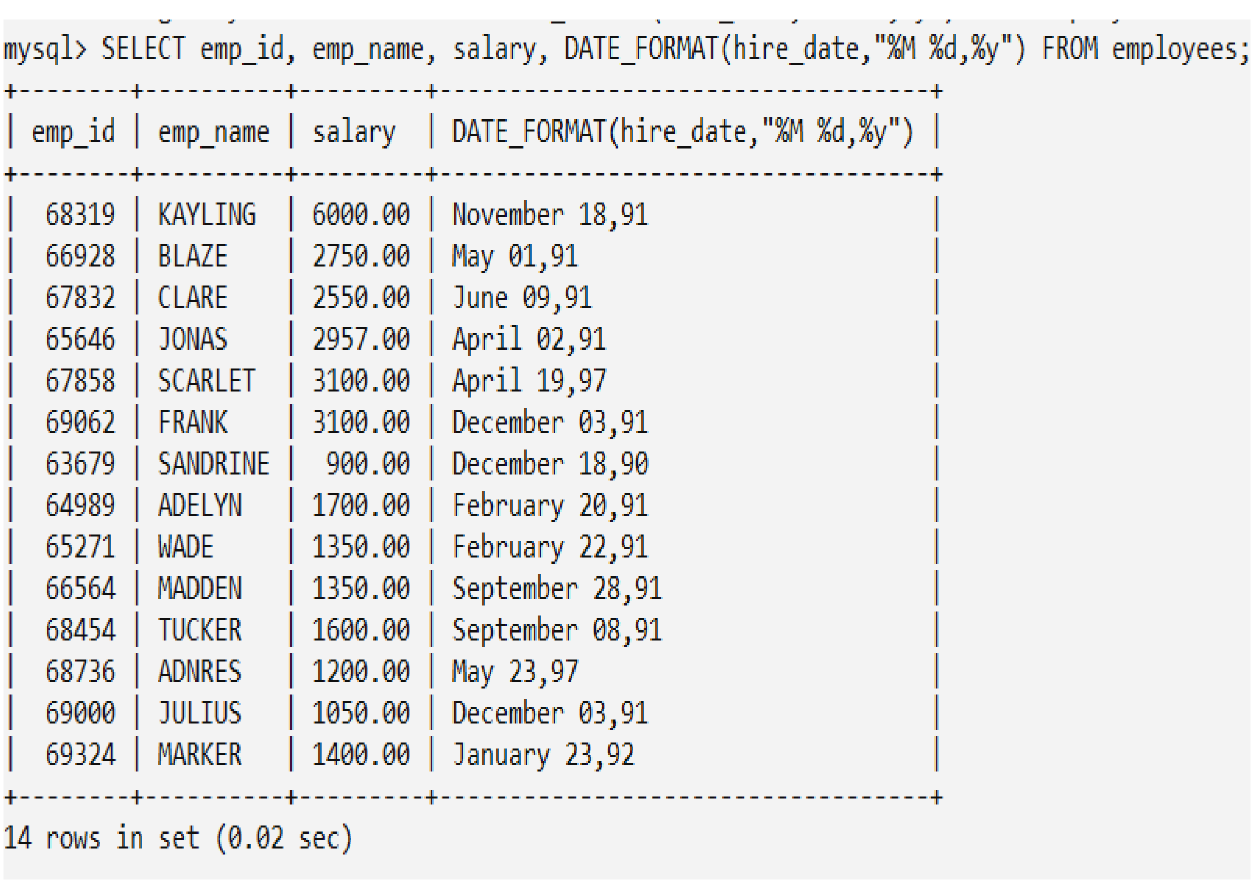


```
mysql> SELECT CONCAT(emp_name,"(",job_name,")") FROM employees;
+------------------------------------+
| CONCAT(emp_name,"(",job_name,")") |
+------------------------------------+
| KAYLING(PRESIDENT)                 |
| BLAZE(MANAGER)                     |
| CLARE(MANAGER)                     |
| JONAS(MANAGER)                     |
| SCARLET(ANALYST)                   |
| FRANK(ANALYST)                     |
| SANDRINE(CLERK)                    |
| ADELYN(SALESMAN)                   |
| WADE(SALESMAN)                     |
| MADDEN(SALESMAN)                   |
| TUCKER(SALESMAN)                   |
| ADNRES(CLERK)                      |
| JULIUS(CLERK)                      |
| MARKER(CLERK)                      |
+------------------------------------+
14 rows in set (0.00 sec)
```

Task - 7

Write a query in SQL to list the employees with Hire date in the format like February 22, 1991.

Command: `SELECT  emp_id, emp_name, salary, DATE_FORMAT(hire_date,”%M %d,%y”) FROM employees;`

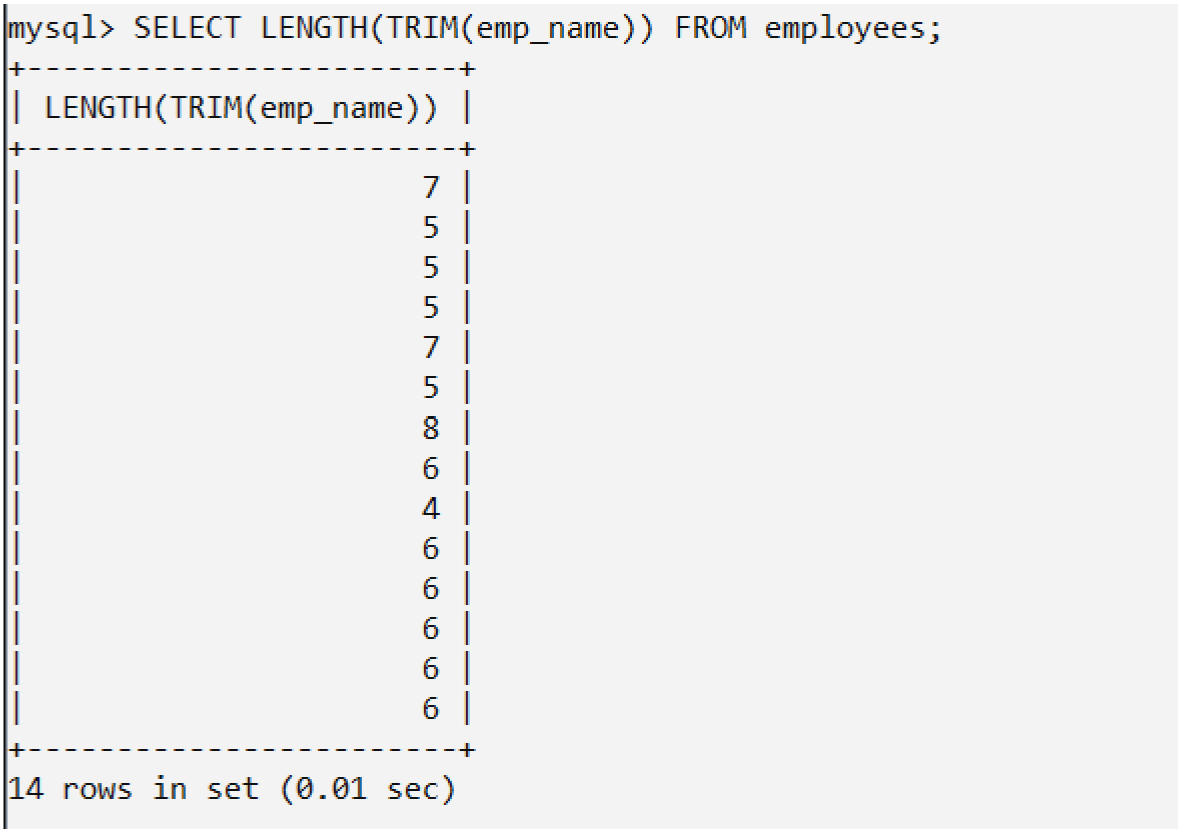


```
mysql> SELECT emp_id, emp_name, salary, DATE_FORMAT(hire_date,"%M %d,%y") FROM employees;
+--------+----------+---------+-----------------------------------+
| emp_id | emp_name | salary  | DATE_FORMAT(hire_date,"%M %d,%y") |
+--------+----------+---------+-----------------------------------+
|  68319 | KAYLING  | 6000.00 | November 18,91                    |
|  66928 | BLAZE    | 2750.00 | May 01,91                         |
|  67832 | CLARE    | 2550.00 | June 09,91                        |
|  65646 | JONAS    | 2957.00 | April 02,91                       |
|  67858 | SCARLET  | 3100.00 | April 19,97                       |
|  69062 | FRANK    | 3100.00 | December 03,91                    |
|  63679 | SANDRINE |  900.00 | December 18,90                    |
|  64989 | ADELYN   | 1700.00 | February 20,91                    |
|  65271 | WADE     | 1350.00 | February 22,91                    |
|  66564 | MADDEN   | 1350.00 | September 28,91                   |
|  68454 | TUCKER   | 1600.00 | September 08,91                   |
|  68736 | ADNRES   | 1200.00 | May 23,97                         |
|  69000 | JULIUS   | 1050.00 | December 03,91                    |
|  69324 | MARKER   | 1400.00 | January 23,92                     |
+--------+----------+---------+-----------------------------------+
14 rows in set (0.02 sec)
```

Task - 8

Write a query in SQL to count the no. of characters without considering the spaces for each name.

Command: `SELECT LENGTH(TRIM(emp_name)) FROM employees;`

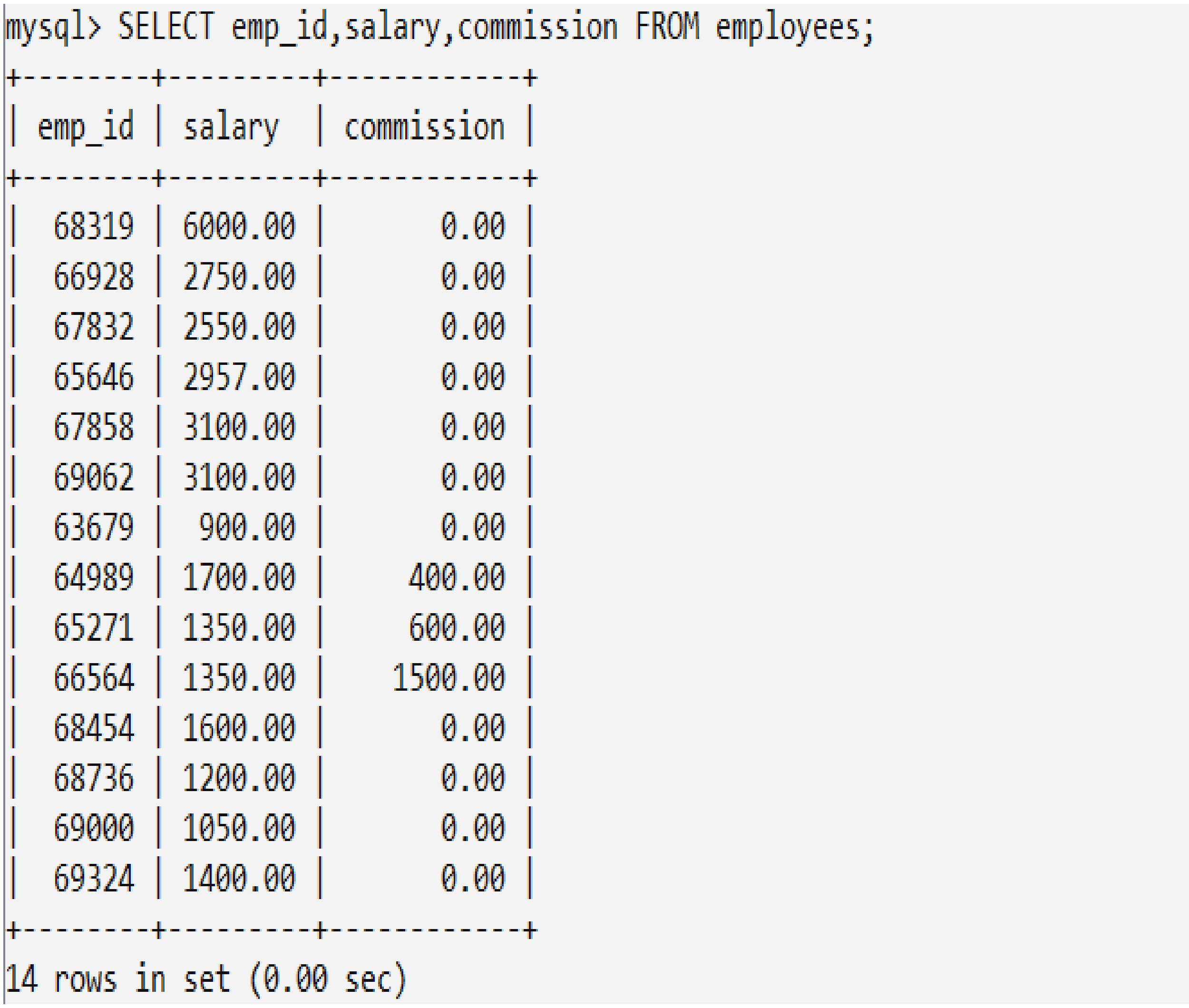


```
mysql> SELECT LENGTH(TRIM(emp_name)) FROM employees;
+------------------------+
| LENGTH(TRIM(emp_name)) |
+------------------------+
|                      7 |
|                      5 |
|                      5 |
|                      5 |
|                      7 |
|                      5 |
|                      8 |
|                      6 |
|                      4 |
|                      6 |
|                      6 |
|                      6 |
|                      6 |
|                      6 |
+------------------------+
14 rows in set (0.01 sec)
```

Task - 9

Write a query in SQL to list the emp_id,salary, and commission of all the employees.

Command: `SELECT emp_id,salary,commission FROM employees;`

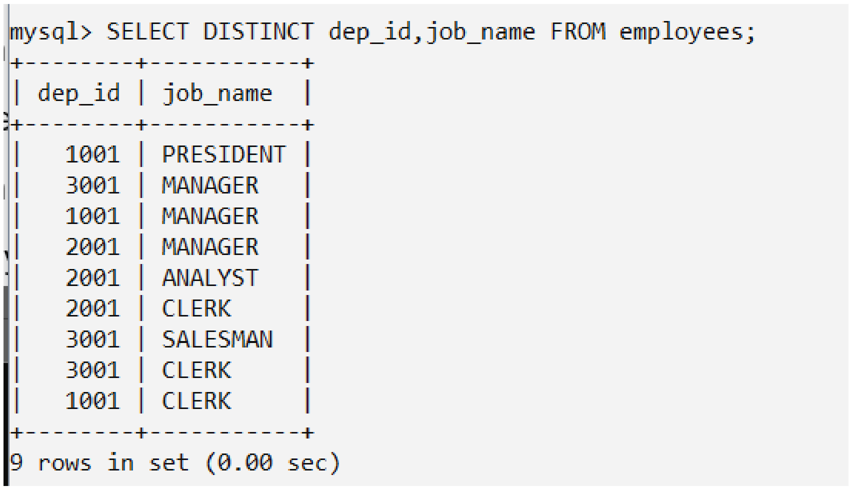


```
mysql> SELECT emp_id,salary,commission FROM employees;
+--------+---------+------------+
| emp_id | salary  | commission |
+--------+---------+------------+
|  68319 | 6000.00 |       0.00 |
|  66928 | 2750.00 |       0.00 |
|  67832 | 2550.00 |       0.00 |
|  65646 | 2957.00 |       0.00 |
|  67858 | 3100.00 |       0.00 |
|  69062 | 3100.00 |       0.00 |
|  63679 |  900.00 |       0.00 |
|  64989 | 1700.00 |     400.00 |
|  65271 | 1350.00 |     600.00 |
|  66564 | 1350.00 |    1500.00 |
|  68454 | 1600.00 |       0.00 |
|  68736 | 1200.00 |       0.00 |
|  69000 | 1050.00 |       0.00 |
|  69324 | 1400.00 |       0.00 |
+--------+---------+------------+
14 rows in set (0.00 sec)
```

Task - 10

Write a query in SQL to display the unique department with jobs.

Command: `SELECT DISTINCT dep_id,job_name FROM employees;`

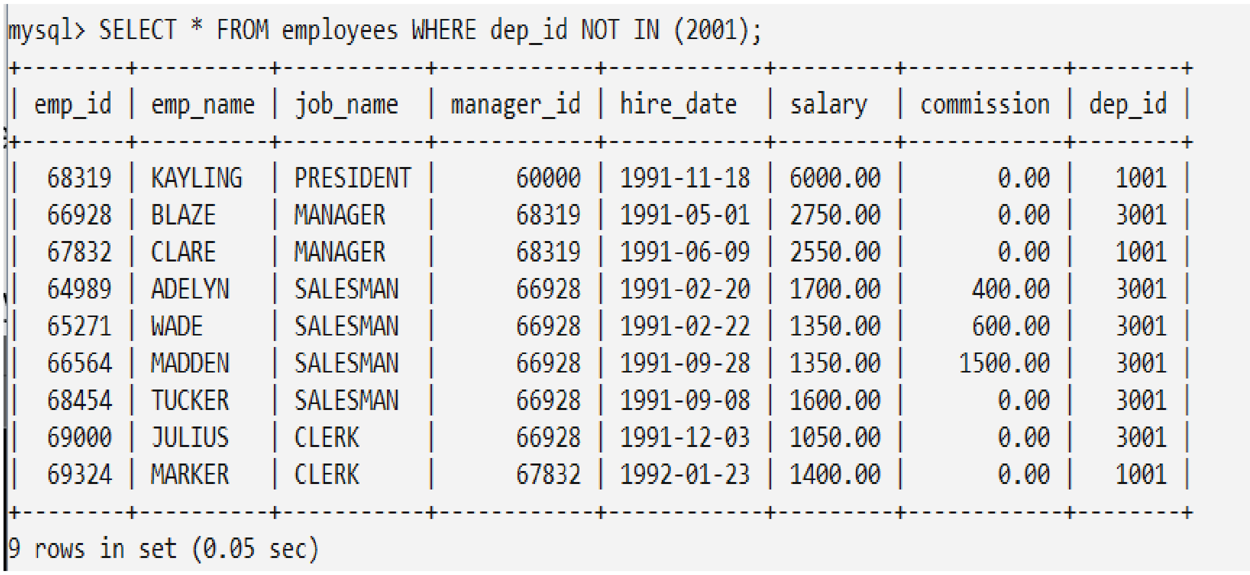


```
mysql> SELECT DISTINCT dep_id,job_name FROM employees;
+--------+-----------+
| dep_id | job_name  |
+--------+-----------+
|   1001 | PRESIDENT |
|   3001 | MANAGER   |
|   1001 | MANAGER   |
|   2001 | MANAGER   |
|   2001 | ANALYST   |
|   2001 | CLERK     |
|   3001 | SALESMAN  |
|   3001 | CLERK     |
|   1001 | CLERK     |
+--------+-----------+
9 rows in set (0.00 sec)
```

Task - 11

Write a query in SQL to list the employees who does not belong to department 2001.

Command : `SELECT * FROM employees WHERE dep_id NOT IN (2001);`

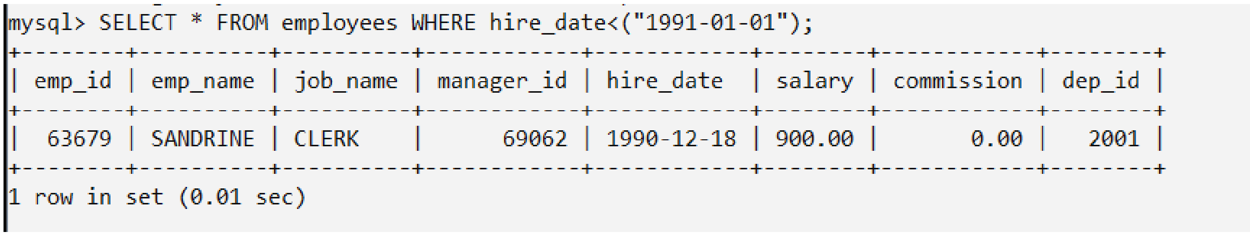


```
mysql> SELECT * FROM employees WHERE dep_id NOT IN (2001);
+--------+----------+-----------+------------+------------+---------+------------+--------+
| emp_id | emp_name | job_name  | manager_id | hire_date  | salary  | commission | dep_id |
+--------+----------+-----------+------------+------------+---------+------------+--------+
|  68319 | KAYLING  | PRESIDENT |      60000 | 1991-11-18 | 6000.00 |       0.00 |   1001 |
|  66928 | BLAZE    | MANAGER   |      68319 | 1991-05-01 | 2750.00 |       0.00 |   3001 |
|  67832 | CLARE    | MANAGER   |      68319 | 1991-06-09 | 2550.00 |       0.00 |   1001 |
|  64989 | ADELYN   | SALESMAN  |      66928 | 1991-02-20 | 1700.00 |     400.00 |   3001 |
|  65271 | WADE     | SALESMAN  |      66928 | 1991-02-22 | 1350.00 |     600.00 |   3001 |
|  66564 | MADDEN   | SALESMAN  |      66928 | 1991-09-28 | 1350.00 |    1500.00 |   3001 |
|  68454 | TUCKER   | SALESMAN  |      66928 | 1991-09-08 | 1600.00 |       0.00 |   3001 |
|  69000 | JULIUS   | CLERK     |      66928 | 1991-12-03 | 1050.00 |       0.00 |   3001 |
|  69324 | MARKER   | CLERK     |      67832 | 1992-01-23 | 1400.00 |       0.00 |   1001 |
+--------+----------+-----------+------------+------------+---------+------------+--------+
9 rows in set (0.05 sec)
```

Task - 12

Write a query in SQL to list the employees who joined before 1991.

Command : `SELECT * FROM employees WHERE hire_date<("1991-01-01");`

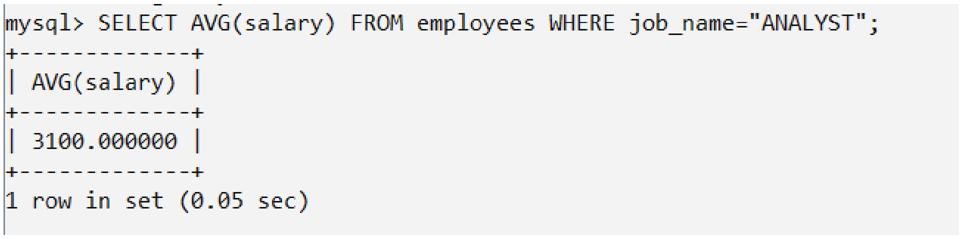


```
mysql> SELECT * FROM employees WHERE hire_date<("1991-01-01");
+--------+----------+----------+------------+------------+--------+------------+--------+
| emp_id | emp_name | job_name | manager_id | hire_date  | salary | commission | dep_id |
+--------+----------+----------+------------+------------+--------+------------+--------+
|  63679 | SANDRINE | CLERK    |      69062 | 1990-12-18 | 900.00 |       0.00 |   2001 |
+--------+----------+----------+------------+------------+--------+------------+--------+
1 row in set (0.01 sec)
```

Task - 13

Write a query in SQL to display the average salaries of all the employees who works as ANALYST.

Command : `SELECT AVG(salary) FROM employees WHERE job_name="ANALYST";`

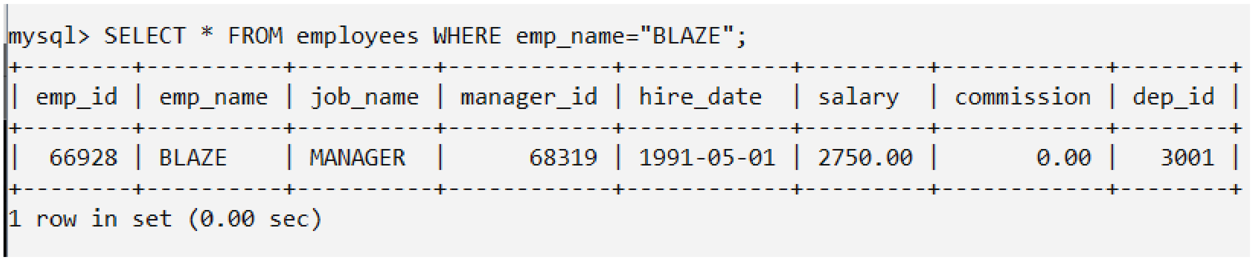


```
mysql> SELECT AVG(salary) FROM employees WHERE job_name="ANALYST";
+-------------+
| AVG(salary) |
+-------------+
| 3100.000000 |
+-------------+
1 row in set (0.05 sec)
```

Task - 14

Write a query in SQL to display the details of the employee BLAZE.

Command : `SELECT * FROM employees WHERE emp_name=”BLAZE”;`

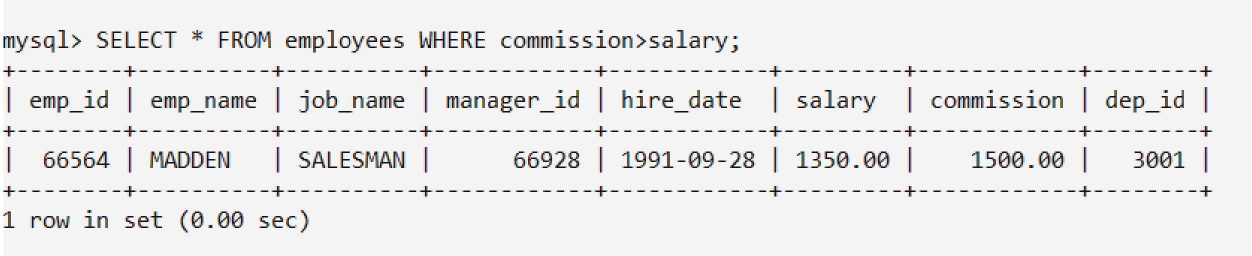


```
mysql> SELECT * FROM employees WHERE emp_name="BLAZE";
+--------+----------+----------+------------+------------+---------+------------+--------+
| emp_id | emp_name | job_name | manager_id | hire_date  | salary  | commission | dep_id |
+--------+----------+----------+------------+------------+---------+------------+--------+
|  66928 | BLAZE    | MANAGER  |      68319 | 1991-05-01 | 2750.00 |       0.00 |   3001 |
+--------+----------+----------+------------+------------+---------+------------+--------+
1 row in set (0.00 sec)
```

Task - 15

Write a query in SQL to display all the details of the employees whose commission is more than their salary.

Command : `SELECT * FROM employees WHERE commission>salary;`

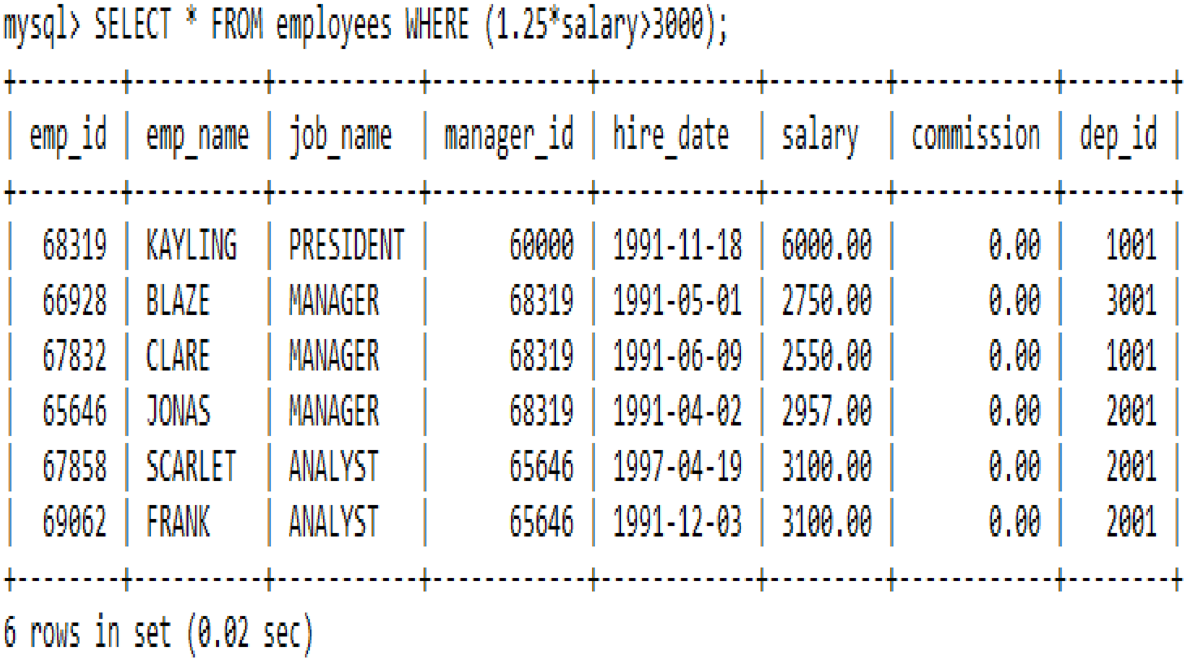


```
mysql> SELECT * FROM employees WHERE commission>salary;
+--------+----------+----------+------------+------------+---------+------------+--------+
| emp_id | emp_name | job_name | manager_id | hire_date  | salary  | commission | dep_id |
+--------+----------+----------+------------+------------+---------+------------+--------+
|  66564 | MADDEN   | SALESMAN |      66928 | 1991-09-28 | 1350.00 |    1500.00 |   3001 |
+--------+----------+----------+------------+------------+---------+------------+--------+
1 row in set (0.00 sec)
```

Task - 16

Write a query in SQL to list the employees whose salary is more than 3000 after giving 25% increment.

Command : `SELECT * FROM employees WHERE (1.25*salary>3000);`

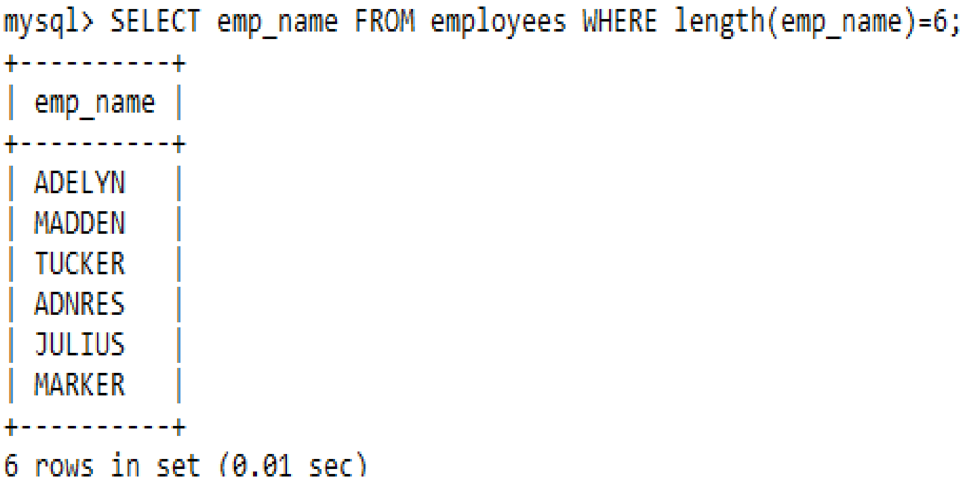


```
mysql> SELECT * FROM employees WHERE (1.25*salary>3000);
+--------+----------+-----------+------------+------------+---------+------------+--------+
| emp_id | emp_name | job_name  | manager_id | hire_date  | salary  | commission | dep_id |
+--------+----------+-----------+------------+------------+---------+------------+--------+
|  68319 | KAYLING  | PRESIDENT |      60000 | 1991-11-18 | 6000.00 |       0.00 |   1001 |
|  66928 | BLAZE    | MANAGER   |      68319 | 1991-05-01 | 2750.00 |       0.00 |   3001 |
|  67832 | CLARE    | MANAGER   |      68319 | 1991-06-09 | 2550.00 |       0.00 |   1001 |
|  65646 | JONAS    | MANAGER   |      68319 | 1991-04-02 | 2957.00 |       0.00 |   2001 |
|  67858 | SCARLET  | ANALYST   |      65646 | 1997-04-19 | 3100.00 |       0.00 |   2001 |
|  69062 | FRANK    | ANALYST   |      65646 | 1991-12-03 | 3100.00 |       0.00 |   2001 |
+--------+----------+-----------+------------+------------+---------+------------+--------+
6 rows in set (0.02 sec)
```

Task - 17

Write a query in SQL to list the name of the employees, those having six characters to their name.

Command : `SELECT emp_name FROM employees WHERE length(emp_name)=6;`

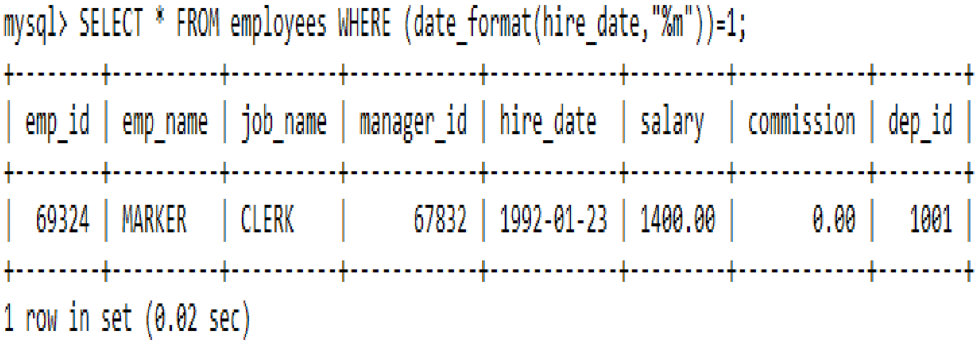


```
mysql> SELECT emp_name FROM employees WHERE length(emp_name)=6;
+----------+
| emp_name |
+----------+
| ADELYN   |
| MADDEN   |
| TUCKER   |
| ADNRES   |
| JULIUS   |
| MARKER   |
+----------+
6 rows in set (0.01 sec)
```

Task - 18

Write a query in SQL to list the employees who joined in the month January.

Command : `SELECT * FROM employees WHERE (date_format(hire_date,”%m”))=1;`

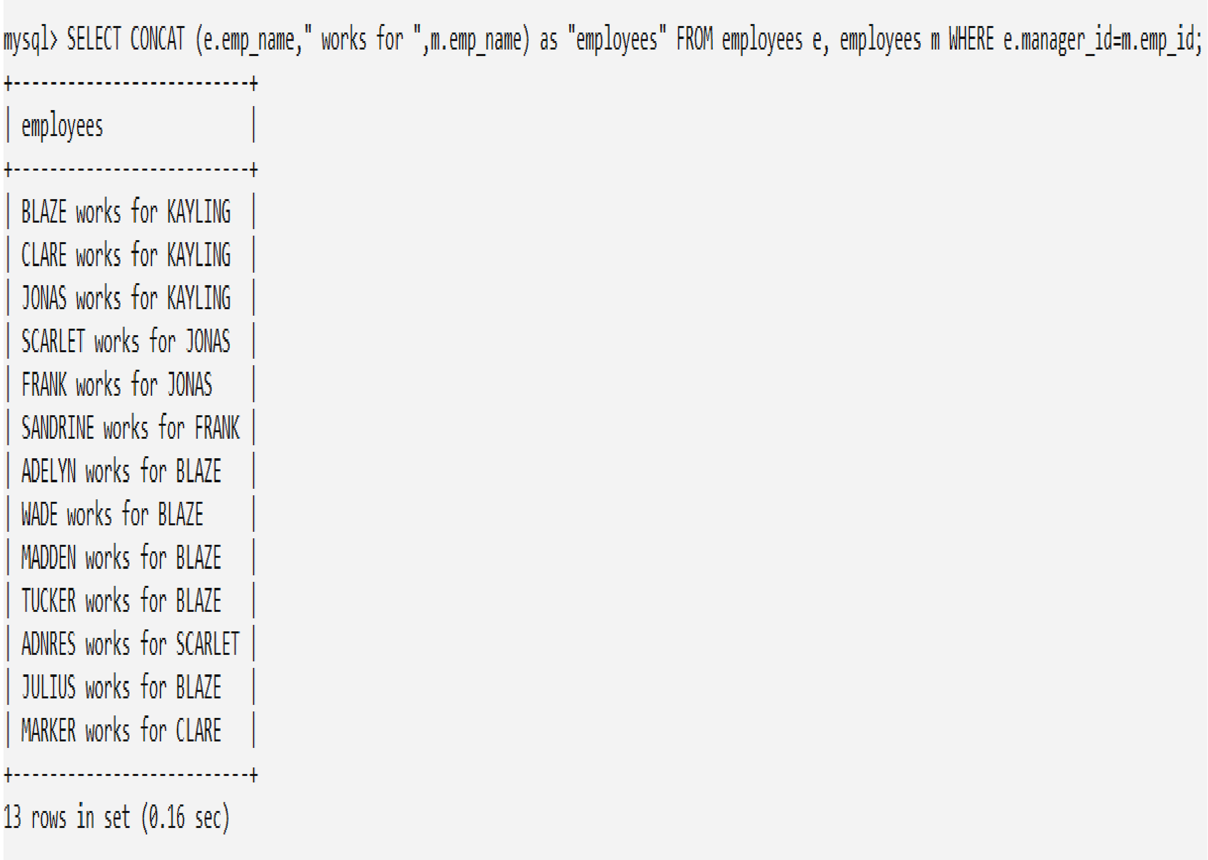


```
mysql> SELECT * FROM employees WHERE (date_format(hire_date,"%m"))=1;
+--------+----------+----------+------------+------------+---------+------------+--------+
| emp_id | emp_name | job_name | manager_id | hire_date  | salary  | commission | dep_id |
+--------+----------+----------+------------+------------+---------+------------+--------+
|  69324 | MARKER   | CLERK    |      67832 | 1992-01-23 | 1400.00 |       0.00 |   1001 |
+--------+----------+----------+------------+------------+---------+------------+--------+
1 row in set (0.02 sec)
```

Task - 19

Write a query in SQL to list the name of employees and their manager separated by the string 'works for'.

Command : `SELECT CONCAT (e.emp_name,” works for ”,m.emp_name) as “employees” FROM employees e, employees m WHERE e.manager_id=m.emp_id;`

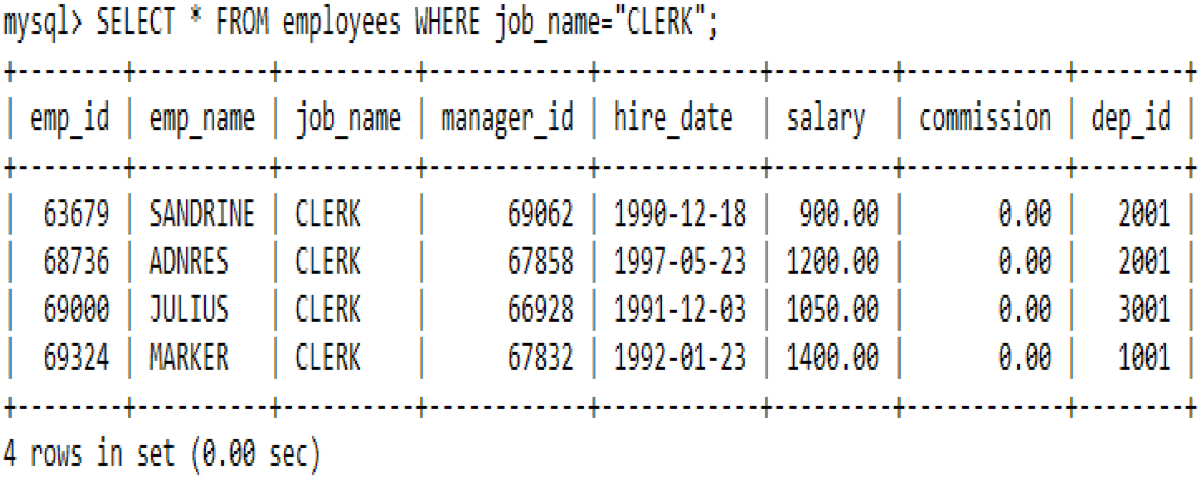


```
mysql> SELECT CONCAT (e.emp_name," works for ",m.emp_name) as "employees" FROM employees e, employees m WHERE e.manager_id=m.emp_id;
+--------------------------+
| employees                |
+--------------------------+
| BLAZE works for KAYLING  |
| CLARE works for KAYLING  |
| JONAS works for KAYLING  |
| SCARLET works for JONAS  |
| FRANK works for JONAS    |
| SANDRINE works for FRANK |
| ADELYN works for BLAZE   |
| WADE works for BLAZE     |
| MADDEN works for BLAZE   |
| TUCKER works for BLAZE   |
| ADNRES works for SCARLET |
| JULIUS works for BLAZE   |
| MARKER works for CLARE   |
+--------------------------+
13 rows in set (0.16 sec)
```

Task - 20

Write a query in SQL to list all the employees whose designation is CLERK.

Command : `SELECT * FROM employees WHERE job_name=”CLERK”;`



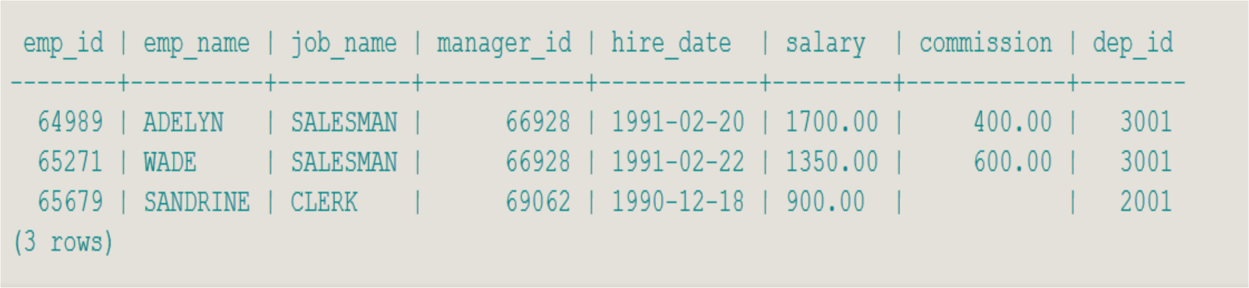
```
mysql> SELECT * FROM employees WHERE job_name="CLERK";
+--------+----------+----------+------------+------------+---------+------------+--------+
| emp_id | emp_name | job_name | manager_id | hire_date  | salary  | commission | dep_id |
+--------+----------+----------+------------+------------+---------+------------+--------+
|  63679 | SANDRINE | CLERK    |      69062 | 1990-12-18 |  900.00 |       0.00 |   2001 |
|  68736 | ADNRES   | CLERK    |      67858 | 1997-05-23 | 1200.00 |       0.00 |   2001 |
|  69000 | JULIUS   | CLERK    |      66928 | 1991-12-03 | 1050.00 |       0.00 |   3001 |
|  69324 | MARKER   | CLERK    |      67832 | 1992-01-23 | 1400.00 |       0.00 |   1001 |
+--------+----------+----------+------------+------------+---------+------------+--------+
4 rows in set (0.00 sec)
```

Task – 21

Write a query in SQL to list the employees whose experience is more than 27 years.

Command : `SELECT *`

```
 FROM employees
 WHERE EXTRACT(YEAR
        FROM age(CURRENT_DATE, hire_date)) > 27;
```



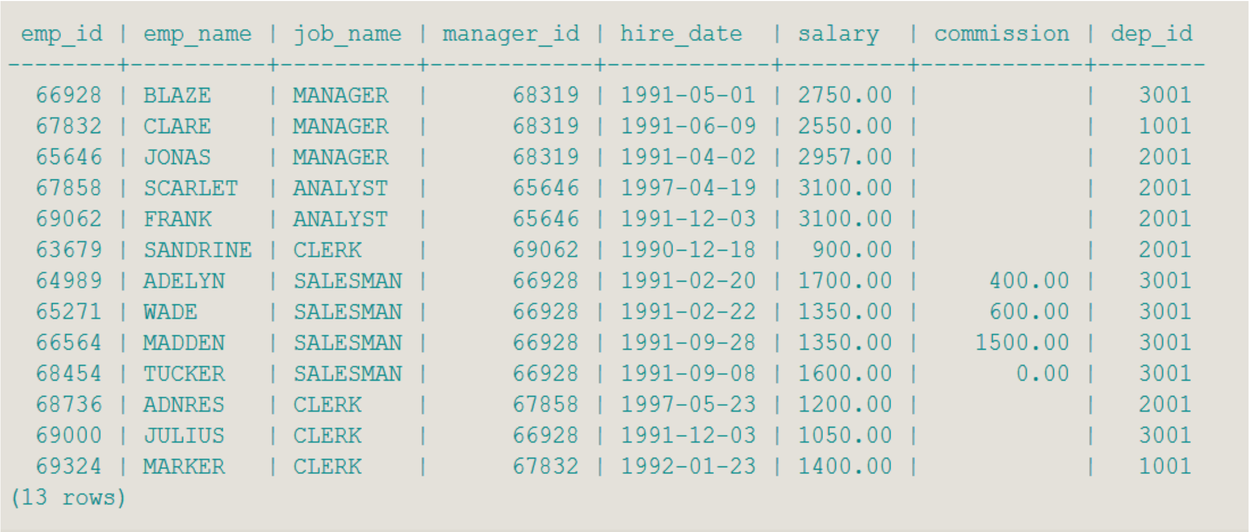
```
 emp_id | emp_name | job_name | manager_id | hire_date  | salary  | commission | dep_id
--------+----------+----------+------------+------------+---------+------------+--------
  64989 | ADELYN   | SALESMAN |      66928 | 1991-02-20 | 1700.00 |     400.00 |   3001
  65271 | WADE     | SALESMAN |      66928 | 1991-02-22 | 1350.00 |     600.00 |   3001
  65679 | SANDRINE | CLERK    |      69062 | 1990-12-18 | 900.00  |            |   2001
(3 rows)
```

Task - 22

Write a query in SQL to list the employees whose salaries are less than 3500.

Command :

```
SELECT *
 FROM employees
 WHERE salary <3500;
```



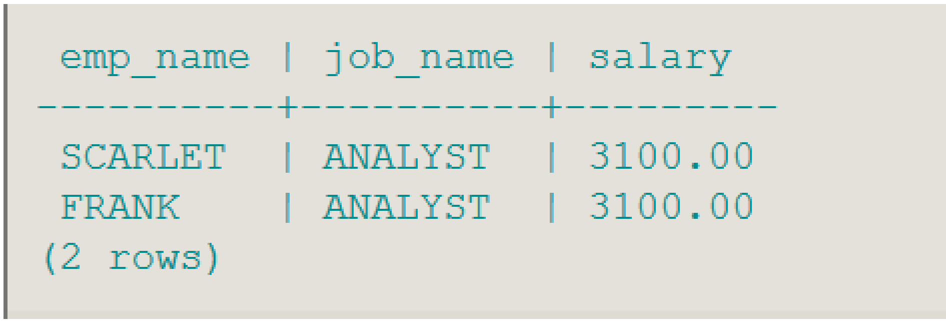
```
 emp_id | emp_name | job_name | manager_id | hire_date  | salary  | commission | dep_id
--------+----------+----------+------------+------------+---------+------------+--------
  66928 | BLAZE    | MANAGER  |      68319 | 1991-05-01 | 2750.00 |            |   3001
  67832 | CLARE    | MANAGER  |      68319 | 1991-06-09 | 2550.00 |            |   1001
  65646 | JONAS    | MANAGER  |      68319 | 1991-04-02 | 2957.00 |            |   2001
  67858 | SCARLET  | ANALYST  |      65646 | 1997-04-19 | 3100.00 |            |   2001
  69062 | FRANK    | ANALYST  |      65646 | 1991-12-03 | 3100.00 |            |   2001
  63679 | SANDRINE | CLERK    |      69062 | 1990-12-18 |  900.00 |            |   2001
  64989 | ADELYN   | SALESMAN |      66928 | 1991-02-20 | 1700.00 |     400.00 |   3001
  65271 | WADE     | SALESMAN |      66928 | 1991-02-22 | 1350.00 |     600.00 |   3001
  66564 | MADDEN   | SALESMAN |      66928 | 1991-09-28 | 1350.00 |    1500.00 |   3001
  68454 | TUCKER   | SALESMAN |      66928 | 1991-09-08 | 1600.00 |       0.00 |   3001
  68736 | ADNRES   | CLERK    |      67858 | 1997-05-23 | 1200.00 |            |   2001
  69000 | JULIUS   | CLERK    |      66928 | 1991-12-03 | 1050.00 |            |   3001
  69324 | MARKER   | CLERK    |      67832 | 1992-01-23 | 1400.00 |            |   1001
(13 rows)
```

Task - 23

Write a query in SQL to list the name, job_name, and salary of any employee whose designation is ANALYST.

Command :

```
SELECT emp_name,
        job_name,
        salary
    FROM employees
    WHERE job_name = 'ANALYST';
```



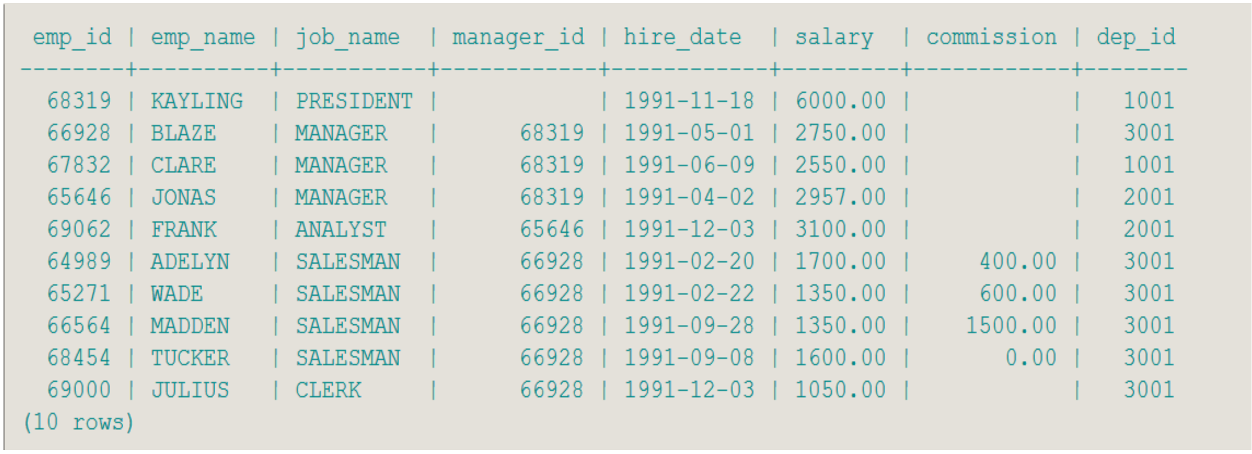
```
 emp_name | job_name | salary
----------+----------+----------
 SCARLET  | ANALYST  | 3100.00
 FRANK    | ANALYST  | 3100.00
(2 rows)
```

Task - 24

Write a query in SQL to list the employees who have joined in the year 1991.

Command : SELECT *

```
FROM employees
WHERE to_char(hire_date,'YYYY') = '1991';
```



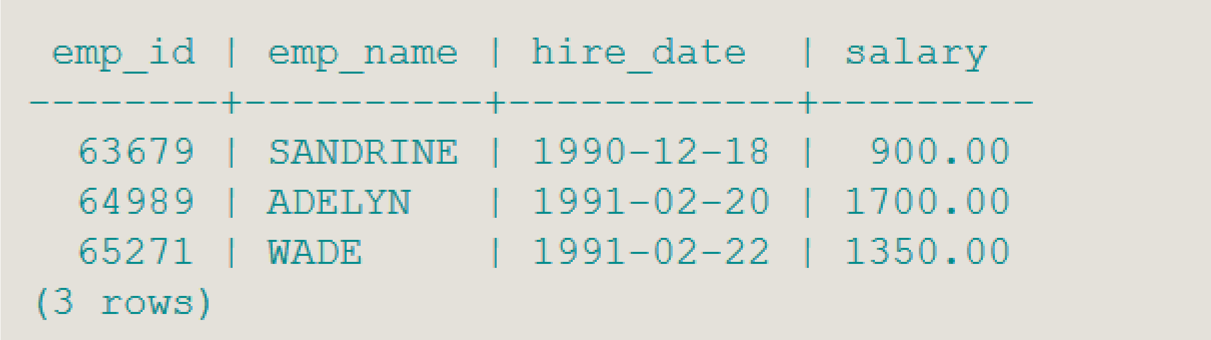
```
 emp_id | emp_name | job_name  | manager_id | hire_date  | salary  | commission | dep_id
--------+----------+-----------+------------+------------+---------+------------+--------
  68319 | KAYLING  | PRESIDENT |            | 1991-11-18 | 6000.00 |            |   1001
  66928 | BLAZE    | MANAGER   |      68319 | 1991-05-01 | 2750.00 |            |   3001
  67832 | CLARE    | MANAGER   |      68319 | 1991-06-09 | 2550.00 |            |   1001
  65646 | JONAS    | MANAGER   |      68319 | 1991-04-02 | 2957.00 |            |   2001
  69062 | FRANK    | ANALYST   |      65646 | 1991-12-03 | 3100.00 |            |   2001
  64989 | ADELYN   | SALESMAN  |      66928 | 1991-02-20 | 1700.00 |     400.00 |   3001
  65271 | WADE     | SALESMAN  |      66928 | 1991-02-22 | 1350.00 |     600.00 |   3001
  66564 | MADDEN   | SALESMAN  |      66928 | 1991-09-28 | 1350.00 |    1500.00 |   3001
  68454 | TUCKER   | SALESMAN  |      66928 | 1991-09-08 | 1600.00 |       0.00 |   3001
  69000 | JULIUS   | CLERK     |      66928 | 1991-12-03 | 1050.00 |            |   3001
(10 rows)
```

Task - 25

Write a query in SQL to list the name, id, hire_date, and salary of all the employees joined before 1 apr 91.

Command :

```
SELECT e.emp_id,
   e.emp_name,
   e.hire_date,
         e.salary
   FROM employees e
   WHERE hire_date <'1991-04-01';
```



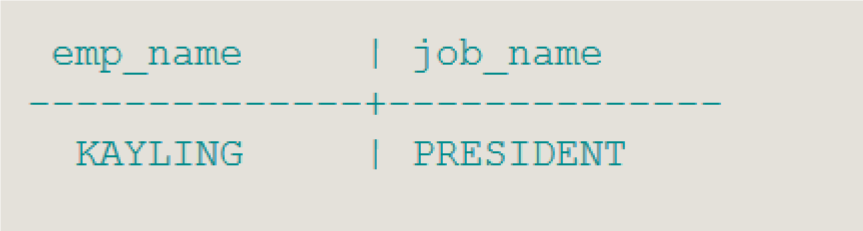
```
 emp_id | emp_name | hire_date   | salary
--------+----------+-------------+---------
  63679 | SANDRINE | 1990-12-18 |   900.00
  64989 | ADELYN   | 1991-02-20 | 1700.00
  65271 | WADE     | 1991-02-22 | 1350.00
(3 rows)
```

Task - 26

Write a query in SQL to list the employee name, and job_name who are not working under a manager.

Command :

```
SELECT e.emp_name,
        e.job_name
  FROM employees e
  WHERE manager_id IS NULL;
```



```
 emp_name      | job_name
---------------+---------------
  KAYLING      | PRESIDENT
```

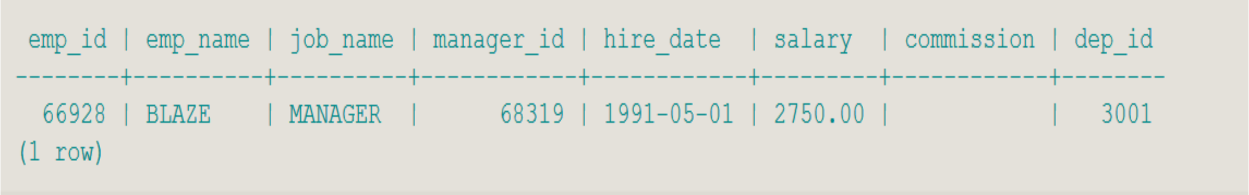
Task - 27

Write a query in SQL to list all the employees joined on 1st may 91.

Command :

```
SELECT *
  FROM employees
```

```
            WHERE hire_date = '1991-05-01';
```



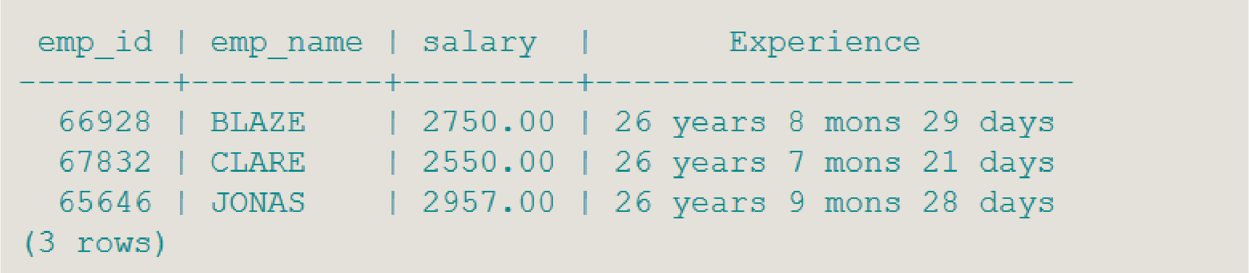
```
 emp_id | emp_name | job_name | manager_id | hire_date  | salary  | commission | dep_id
--------+----------+----------+------------+------------+---------+------------+--------
  66928 | BLAZE    | MANAGER  |      68319 | 1991-05-01 | 2750.00 |            |   3001
(1 row)
```

Task - 28

Write a query in SQL to list the id, name, salry, and experiences of all the employees working for the manger 68319.

Command :

```
SELECT emp_id,
       emp_name,
       salary,
       age(CURRENT_DATE, hire_date) "Experience"
FROM employees
WHERE manager_id=68319;
```



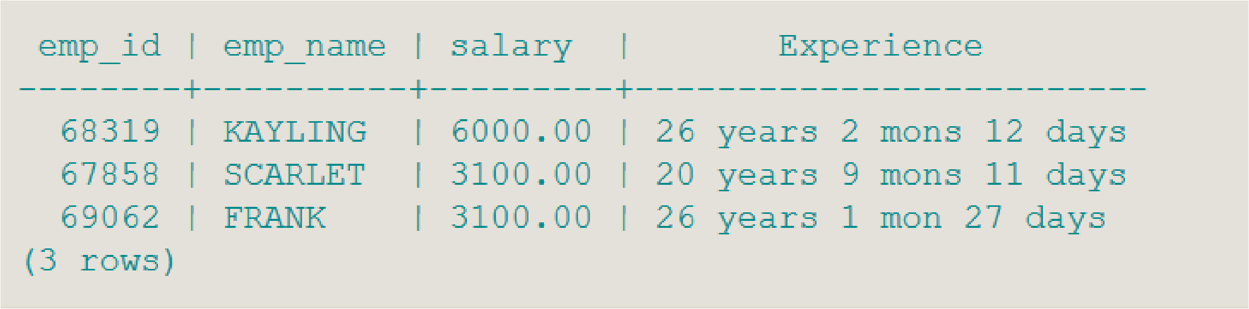
```
 emp_id | emp_name | salary  |        Experience
--------+----------+---------+-------------------------
  66928 | BLAZE    | 2750.00 | 26 years 8 mons 29 days
  67832 | CLARE    | 2550.00 | 26 years 7 mons 21 days
  65646 | JONAS    | 2957.00 | 26 years 9 mons 28 days
(3 rows)
```

Task - 29

Write a query in SQL to list the id, name, salary, and experience of all the employees who earn more than 100 as daily salary.

Command :

```
SELECT emp_id,
          emp_name,
          salary,
          age(CURRENT_DATE, hire_date) "Experience"
      FROM employees
      WHERE (salary/30)>100;
```



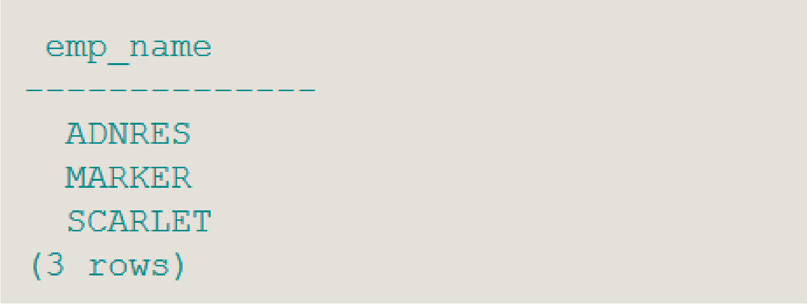
```
 emp_id | emp_name | salary  |        Experience
--------+----------+---------+-------------------------
  68319 | KAYLING  | 6000.00 | 26 years 2 mons 12 days
  67858 | SCARLET  | 3100.00 | 20 years 9 mons 11 days
  69062 | FRANK    | 3100.00 | 26 years 1 mon 27 days
(3 rows)
```

Task - 30

Write a query in SQL to list the employees who are retiring after 31-Dec-99 after completion of 8 years of service period.

Command : SELECT emp_name

```
 FROM employees
 WHERE hire_date + interval '96 months' > '1999-12-
31';
```



```
 emp_name
---------------
  ADNRES
  MARKER
  SCARLET
(3 rows)
```