Symbiosis Skills and Professional University

Skill Journal

Name: Yograj Anant Gadekar **Date:**10-12-2021
PRN:
School: School of Data Science
Course: Data Associate (Data Science)
Module Name: Python for Data Analysis / Managing with Data / Analyzing Data from Disparate Sources **(tick any one)**

1. **Skill Activity Number**– 8

2. **Title-** Joins, Clauses and Sub queries in MySQL

3. **Skills / Competencies to be acquired:** MYSQL

4. **Duration**
   2 days

5. **What is the purpose of the activity?**
   Purpose of this activity joins brings different groups of data together on the basis of relationships.

6. **Steps Performed in this activity?**
   1) Create Database and tables.
   2) Insert values in tables.
   3) Perform Inner join clause
   4) Perform Left join clause
   5) Perform Right join clause
   6) Perform Cross join clause
   7)Perform Natural join clause
   8) Perform a subquery

7. **What resources / materials / equipment / tools did you use for this activity?**
   MYSQL, MS-word

8. **What skills did you acquire?**
   MYSQL

9. **Time taken to complete this activity?**
   1 to 2 hours

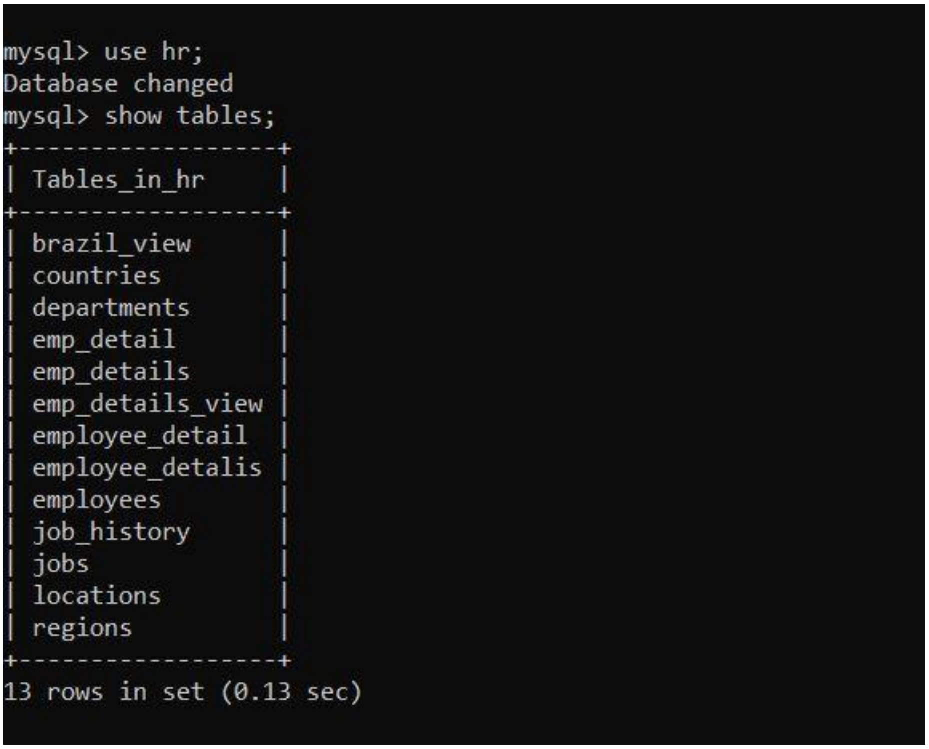
Joins, Clauses and Sub queries in MySQL

JOIN clauses are used to return the rows of two or more queries using two or more tables that shares a meaningful relationship based on a common set of values.

Types of MySQL Joins :

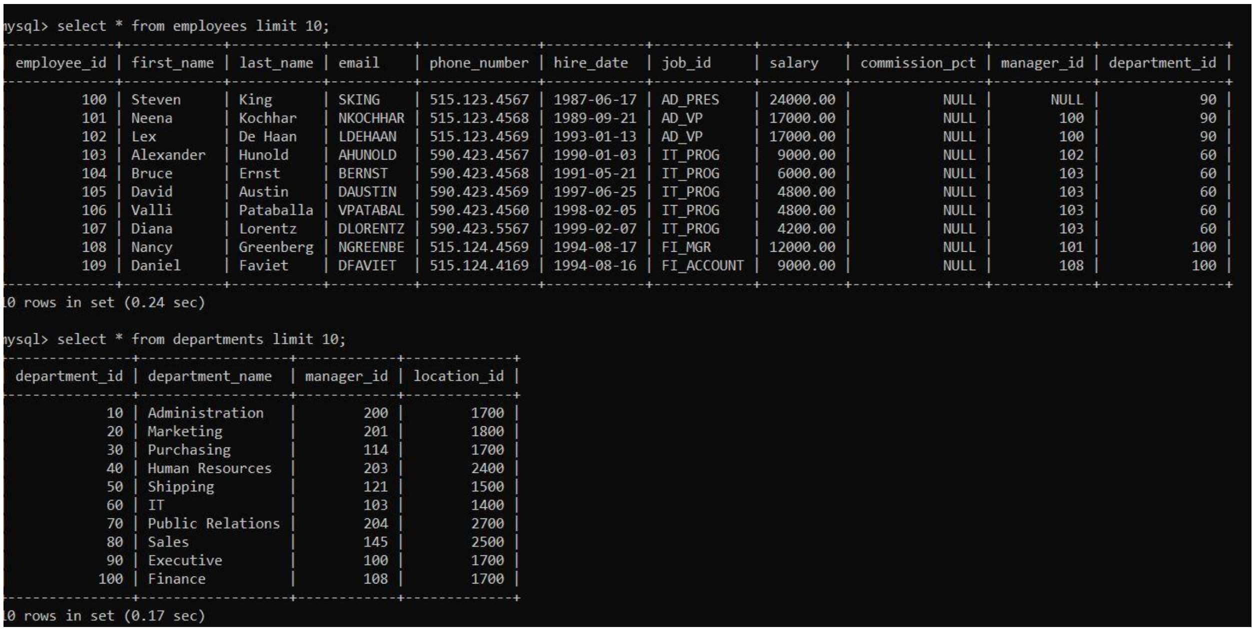
1. INNER JOIN
2. LEFT JOIN
3. RIGHT JOIN
4. CROSS JOIN
5. NATURAL JOIN

use database;



```
mysql> use hr;
Database changed
mysql> show tables;
+------------------+
| Tables_in_hr     |
+------------------+
| brazil_view      |
| countries        |
| departments      |
| emp_detail       |
| emp_details      |
| emp_details_view |
| employee_detail  |
| employee_detalis |
| employees        |
| job_history      |
| jobs             |
| locations        |
| regions          |
+------------------+
13 rows in set (0.13 sec)
```

Show tables data using select



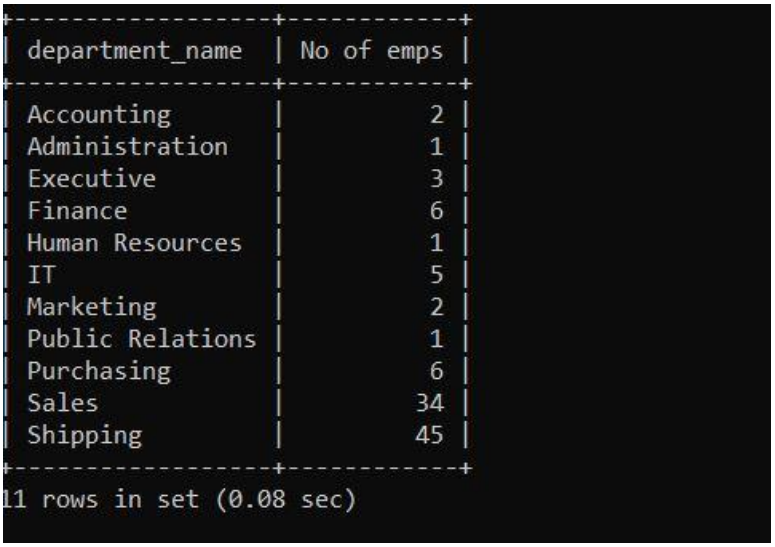
```
ysql> select * from employees limit 10;
+-------------+------------+-----------+----------+--------------+------------+------------+----------+----------------+------------+---------------+
| employee_id | first_name | last_name | email    | phone_number | hire_date  | job_id     | salary   | commission_pct | manager_id | department_id |
+-------------+------------+-----------+----------+--------------+------------+------------+----------+----------------+------------+---------------+
|         100 | Steven     | King      | SKING    | 515.123.4567 | 1987-06-17 | AD_PRES    | 24000.00 |           NULL |       NULL |            90 |
|         101 | Neena      | Kochhar   | NKOCHHAR | 515.123.4568 | 1989-09-21 | AD_VP      | 17000.00 |           NULL |        100 |            90 |
|         102 | Lex        | De Haan   | LDEHAAN  | 515.123.4569 | 1993-01-13 | AD_VP      | 17000.00 |           NULL |        100 |            90 |
|         103 | Alexander  | Hunold    | AHUNOLD  | 590.423.4567 | 1990-01-03 | IT_PROG    |  9000.00 |           NULL |        102 |            60 |
|         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 |            60 |
|         107 | Diana      | Lorentz   | DLORENTZ | 590.423.5567 | 1999-02-07 | IT_PROG    |  4200.00 |           NULL |        103 |            60 |
|         108 | Nancy      | Greenberg | NGREENBE | 515.124.4569 | 1994-08-17 | FI_MGR     | 12000.00 |           NULL |        101 |           100 |
|         109 | Daniel     | Faviet    | DFAVIET  | 515.124.4169 | 1994-08-16 | FI_ACCOUNT |  9000.00 |           NULL |        108 |           100 |
+-------------+------------+-----------+----------+--------------+------------+------------+----------+----------------+------------+---------------+
10 rows in set (0.24 sec)

ysql> select * from departments limit 10;
+---------------+------------------+------------+-------------+
| department_id | department_name  | manager_id | location_id |
+---------------+------------------+------------+-------------+
|            10 | Administration   |        200 |        1700 |
|            20 | Marketing        |        201 |        1800 |
|            30 | Purchasing       |        114 |        1700 |
|            40 | Human Resources  |        203 |        2400 |
|            50 | Shipping         |        121 |        1500 |
|            60 | IT               |        103 |        1400 |
|            70 | Public Relations |        204 |        2700 |
|            80 | Sales            |        145 |        2500 |
|            90 | Executive        |        100 |        1700 |
|           100 | Finance          |        108 |        1700 |
+---------------+------------------+------------+-------------+
10 rows in set (0.17 sec)
```

1) **INNER JOIN:**
   It returns the common records that have matching values in both tables.

   Question. Query to get the department name and number of employees in the department

   Query:
   select department_name,count(*) as 'No of emps' from departments as d inner join employees as e on e.department_id=d.department_id group by d.department_id,department_name order by department_name;

Output:



```
+------------------+------------+
| department_name  | No of emps |
+------------------+------------+
| Accounting       |          2 |
| Administration   |          1 |
| Executive        |          3 |
| Finance          |          6 |
| Human Resources  |          1 |
| IT               |          5 |
| Marketing        |          2 |
| Public Relations |          1 |
| Purchasing       |          6 |
| Sales            |         34 |
| Shipping         |         45 |
+------------------+------------+
11 rows in set (0.08 sec)
```

2) **LEFT JOIN:**

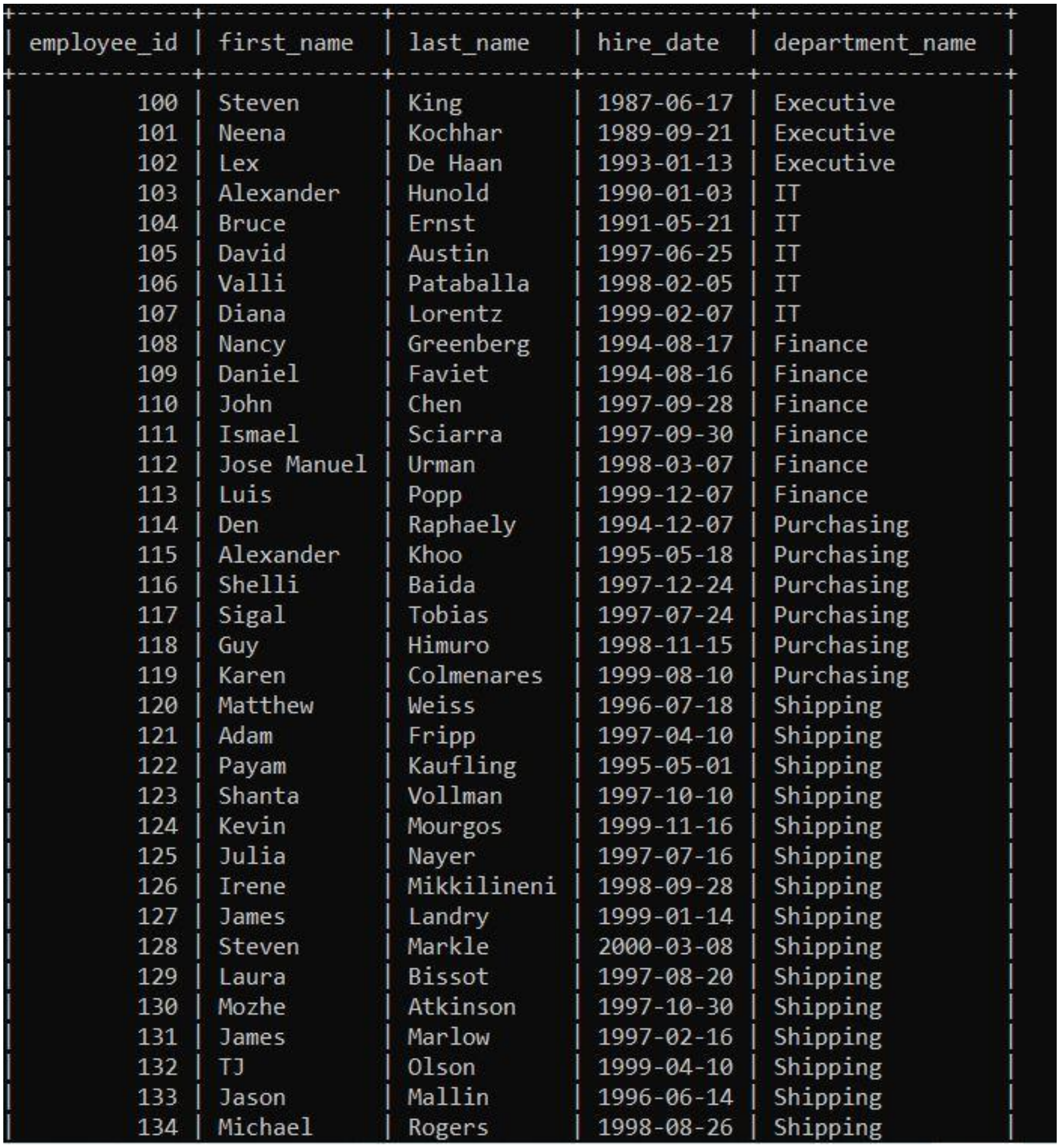
It returns all records from left table and match records from right table.

Question1. The following statement retrieves employee's id, name, hiring date and their department name by joining the *employees* and *departments* tables together using the common *dept_id* field. It also includes those employees who are not assigned to a department.

Query:

select e.employee_id,e.first_name,e.last_name,e.hire_date,d.department_name from employees as e left join departments as d on e.department_id=d.department_id order by employee_id;

Output:



```
+-------------+-------------+-------------+------------+------------------+
| employee_id | first_name  | last_name   | hire_date  | department_name  |
+-------------+-------------+-------------+------------+------------------+
|         100 | Steven      | King        | 1987-06-17 | Executive        |
|         101 | Neena       | Kochhar     | 1989-09-21 | Executive        |
|         102 | Lex         | De Haan     | 1993-01-13 | Executive        |
|         103 | Alexander   | Hunold      | 1990-01-03 | IT               |
|         104 | Bruce       | Ernst       | 1991-05-21 | IT               |
|         105 | David       | Austin      | 1997-06-25 | IT               |
|         106 | Valli       | Pataballa   | 1998-02-05 | IT               |
|         107 | Diana       | Lorentz     | 1999-02-07 | IT               |
|         108 | Nancy       | Greenberg   | 1994-08-17 | Finance          |
|         109 | Daniel      | Faviet      | 1994-08-16 | Finance          |
|         110 | John        | Chen        | 1997-09-28 | Finance          |
|         111 | Ismael      | Sciarra     | 1997-09-30 | Finance          |
|         112 | Jose Manuel | Urman       | 1998-03-07 | Finance          |
|         113 | Luis        | Popp        | 1999-12-07 | Finance          |
|         114 | Den         | Raphaely    | 1994-12-07 | Purchasing       |
|         115 | Alexander   | Khoo        | 1995-05-18 | Purchasing       |
|         116 | Shelli      | Baida       | 1997-12-24 | Purchasing       |
|         117 | Sigal       | Tobias      | 1997-07-24 | Purchasing       |
|         118 | Guy         | Himuro      | 1998-11-15 | Purchasing       |
|         119 | Karen       | Colmenares  | 1999-08-10 | Purchasing       |
|         120 | Matthew     | Weiss       | 1996-07-18 | Shipping         |
|         121 | Adam        | Fripp       | 1997-04-10 | Shipping         |
|         122 | Payam       | Kaufling    | 1995-05-01 | Shipping         |
|         123 | Shanta      | Vollman     | 1997-10-10 | Shipping         |
|         124 | Kevin       | Mourgos     | 1999-11-16 | Shipping         |
|         125 | Julia       | Nayer       | 1997-07-16 | Shipping         |
|         126 | Irene       | Mikkilineni | 1998-09-28 | Shipping         |
|         127 | James       | Landry      | 1999-01-14 | Shipping         |
|         128 | Steven      | Markle      | 2000-03-08 | Shipping         |
|         129 | Laura       | Bissot      | 1997-08-20 | Shipping         |
|         130 | Mozhe       | Atkinson    | 1997-10-30 | Shipping         |
|         131 | James       | Marlow      | 1997-02-16 | Shipping         |
|         132 | TJ          | Olson       | 1999-04-10 | Shipping         |
|         133 | Jason       | Mallin      | 1996-06-14 | Shipping         |
|         134 | Michael     | Rogers      | 1998-08-26 | Shipping         |
```

3) RIGHT JOIN:

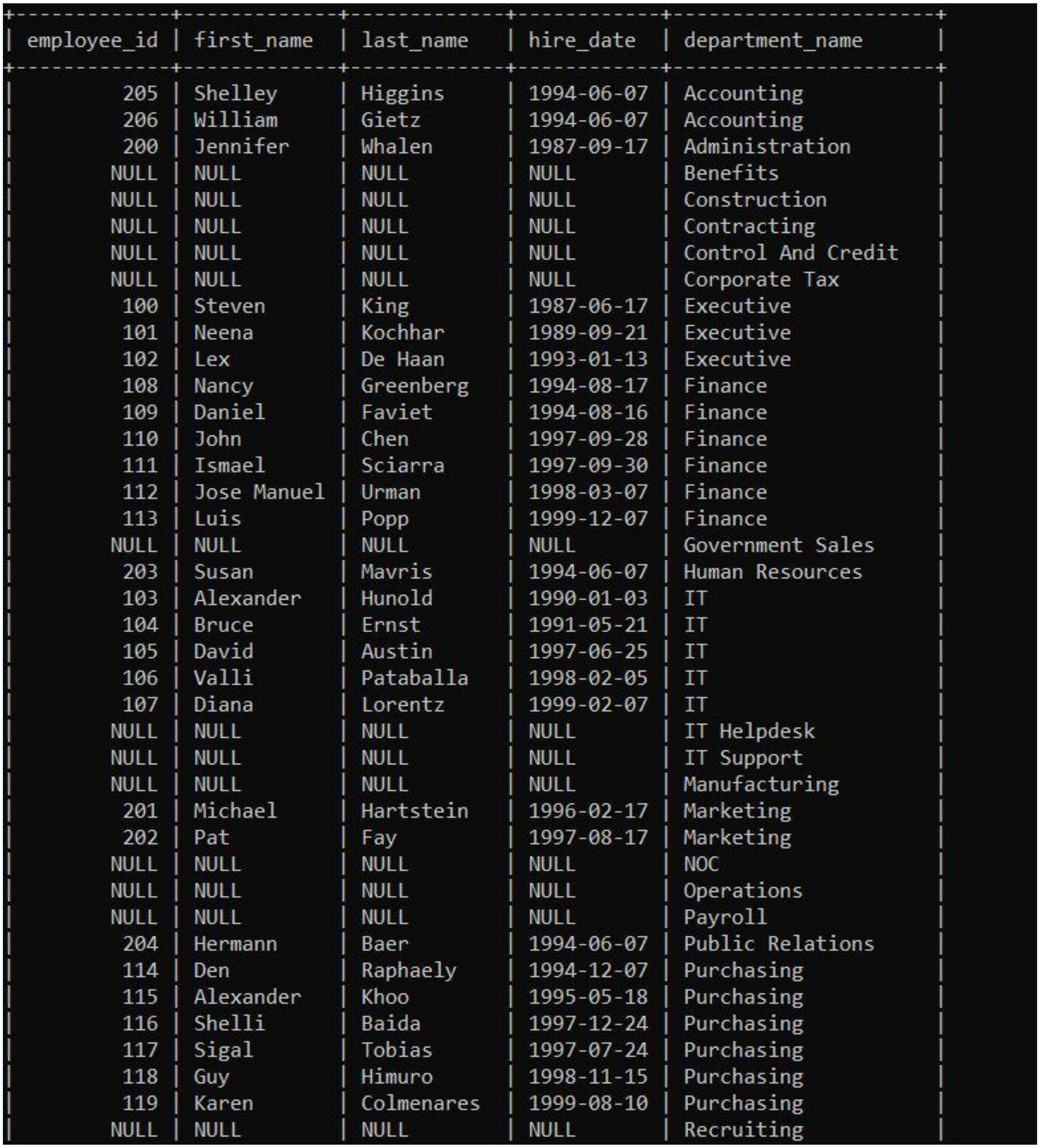
It returns all records from right table and match records from left table.

Question1. The following statement retrieves all the available departments as well as the id, name, hiring date of the employees who belongs to that department by joining the *employees* and *departments* tables together using the common *dept_id* field.

Query:

Select e.employee_id,e.first_name,e.last_name,e.hire_date,d.department_name from employees as e right join departments as d on e.department_id=d.department_id order by department_name;

Output:



```
+-------------+-------------+-------------+------------+----------------------+
| employee_id | first_name  | last_name   | hire_date  | department_name      |
+-------------+-------------+-------------+------------+----------------------+
|         205 | Shelley     | Higgins     | 1994-06-07 | Accounting           |
|         206 | William     | Gietz       | 1994-06-07 | Accounting           |
|         200 | Jennifer    | Whalen      | 1987-09-17 | Administration       |
|        NULL | NULL        | NULL        | NULL       | Benefits             |
|        NULL | NULL        | NULL        | NULL       | Construction         |
|        NULL | NULL        | NULL        | NULL       | Contracting          |
|        NULL | NULL        | NULL        | NULL       | Control And Credit   |
|        NULL | NULL        | NULL        | NULL       | Corporate Tax        |
|         100 | Steven      | King        | 1987-06-17 | Executive            |
|         101 | Neena       | Kochhar     | 1989-09-21 | Executive            |
|         102 | Lex         | De Haan     | 1993-01-13 | Executive            |
|         108 | Nancy       | Greenberg   | 1994-08-17 | Finance              |
|         109 | Daniel      | Faviet      | 1994-08-16 | Finance              |
|         110 | John        | Chen        | 1997-09-28 | Finance              |
|         111 | Ismael      | Sciarra     | 1997-09-30 | Finance              |
|         112 | Jose Manuel | Urman       | 1998-03-07 | Finance              |
|         113 | Luis        | Popp        | 1999-12-07 | Finance              |
|        NULL | NULL        | NULL        | NULL       | Government Sales     |
|         203 | Susan       | Mavris      | 1994-06-07 | Human Resources      |
|         103 | Alexander   | Hunold      | 1990-01-03 | IT                   |
|         104 | Bruce       | Ernst       | 1991-05-21 | IT                   |
|         105 | David       | Austin      | 1997-06-25 | IT                   |
|         106 | Valli       | Pataballa   | 1998-02-05 | IT                   |
|         107 | Diana       | Lorentz     | 1999-02-07 | IT                   |
|        NULL | NULL        | NULL        | NULL       | IT Helpdesk          |
|        NULL | NULL        | NULL        | NULL       | IT Support           |
|        NULL | NULL        | NULL        | NULL       | Manufacturing        |
|         201 | Michael     | Hartstein   | 1996-02-17 | Marketing            |
|         202 | Pat         | Fay         | 1997-08-17 | Marketing            |
|        NULL | NULL        | NULL        | NULL       | NOC                  |
|        NULL | NULL        | NULL        | NULL       | Operations           |
|        NULL | NULL        | NULL        | NULL       | Payroll              |
|         204 | Hermann     | Baer        | 1994-06-07 | Public Relations     |
|         114 | Den         | Raphaely    | 1994-12-07 | Purchasing           |
|         115 | Alexander   | Khoo        | 1995-05-18 | Purchasing           |
|         116 | Shelli      | Baida       | 1997-12-24 | Purchasing           |
|         117 | Sigal       | Tobias      | 1997-07-24 | Purchasing           |
|         118 | Guy         | Himuro      | 1998-11-15 | Purchasing           |
|         119 | Karen       | Colmenares  | 1999-08-10 | Purchasing           |
|        NULL | NULL        | NULL        | NULL       | Recruiting           |
```

4) **CROSS JOIN:**

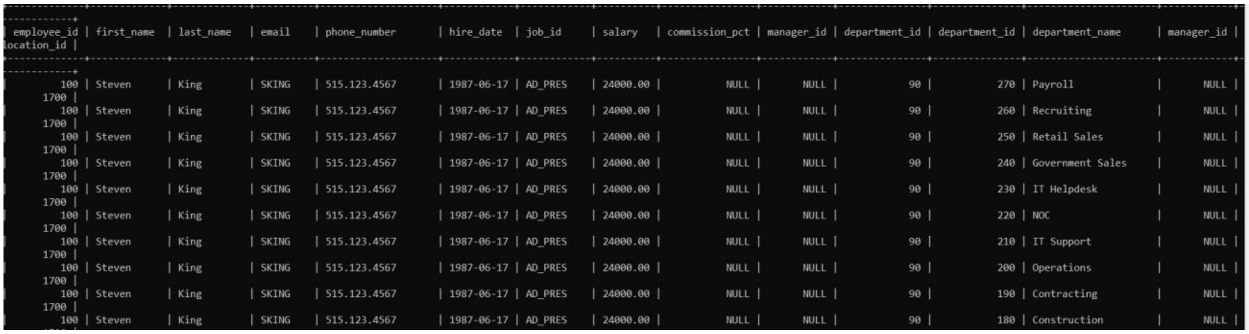
It combines all rows from both tables and return cartesian product of tables.

Question1. Following statement retrieve the cartesian product of employees and departments table.

Query:

select * from employees cross join departments;

Output:



```
+-------------+------------+-----------+-------+--------------+------------+---------+----------+----------------+------------+---------------+---------------+-------------------+------------+
-------------+
| employee_id | first_name | last_name | email | phone_number | hire_date  | job_id  | salary   | commission_pct | manager_id | department_id | department_id | department_name   | manager_id |
location_id |
+-------------+------------+-----------+-------+--------------+------------+---------+----------+----------------+------------+---------------+---------------+-------------------+------------+
-------------+
|         100 | Steven     | King      | SKING | 515.123.4567 | 1987-06-17 | AD_PRES | 24000.00 |           NULL |       NULL |            90 |           270 | Payroll           |       NULL |
       1700 |
|         100 | Steven     | King      | SKING | 515.123.4567 | 1987-06-17 | AD_PRES | 24000.00 |           NULL |       NULL |            90 |           260 | Recruiting        |       NULL |
       1700 |
|         100 | Steven     | King      | SKING | 515.123.4567 | 1987-06-17 | AD_PRES | 24000.00 |           NULL |       NULL |            90 |           250 | Retail Sales      |       NULL |
       1700 |
|         100 | Steven     | King      | SKING | 515.123.4567 | 1987-06-17 | AD_PRES | 24000.00 |           NULL |       NULL |            90 |           240 | Government Sales  |       NULL |
       1700 |
|         100 | Steven     | King      | SKING | 515.123.4567 | 1987-06-17 | AD_PRES | 24000.00 |           NULL |       NULL |            90 |           230 | IT Helpdesk       |       NULL |
       1700 |
|         100 | Steven     | King      | SKING | 515.123.4567 | 1987-06-17 | AD_PRES | 24000.00 |           NULL |       NULL |            90 |           220 | NOC               |       NULL |
       1700 |
|         100 | Steven     | King      | SKING | 515.123.4567 | 1987-06-17 | AD_PRES | 24000.00 |           NULL |       NULL |            90 |           210 | IT Support        |       NULL |
       1700 |
|         100 | Steven     | King      | SKING | 515.123.4567 | 1987-06-17 | AD_PRES | 24000.00 |           NULL |       NULL |            90 |           200 | Operations        |       NULL |
       1700 |
|         100 | Steven     | King      | SKING | 515.123.4567 | 1987-06-17 | AD_PRES | 24000.00 |           NULL |       NULL |            90 |           190 | Contracting       |       NULL |
       1700 |
|         100 | Steven     | King      | SKING | 515.123.4567 | 1987-06-17 | AD_PRES | 24000.00 |           NULL |       NULL |            90 |           180 | Construction      |       NULL |
```

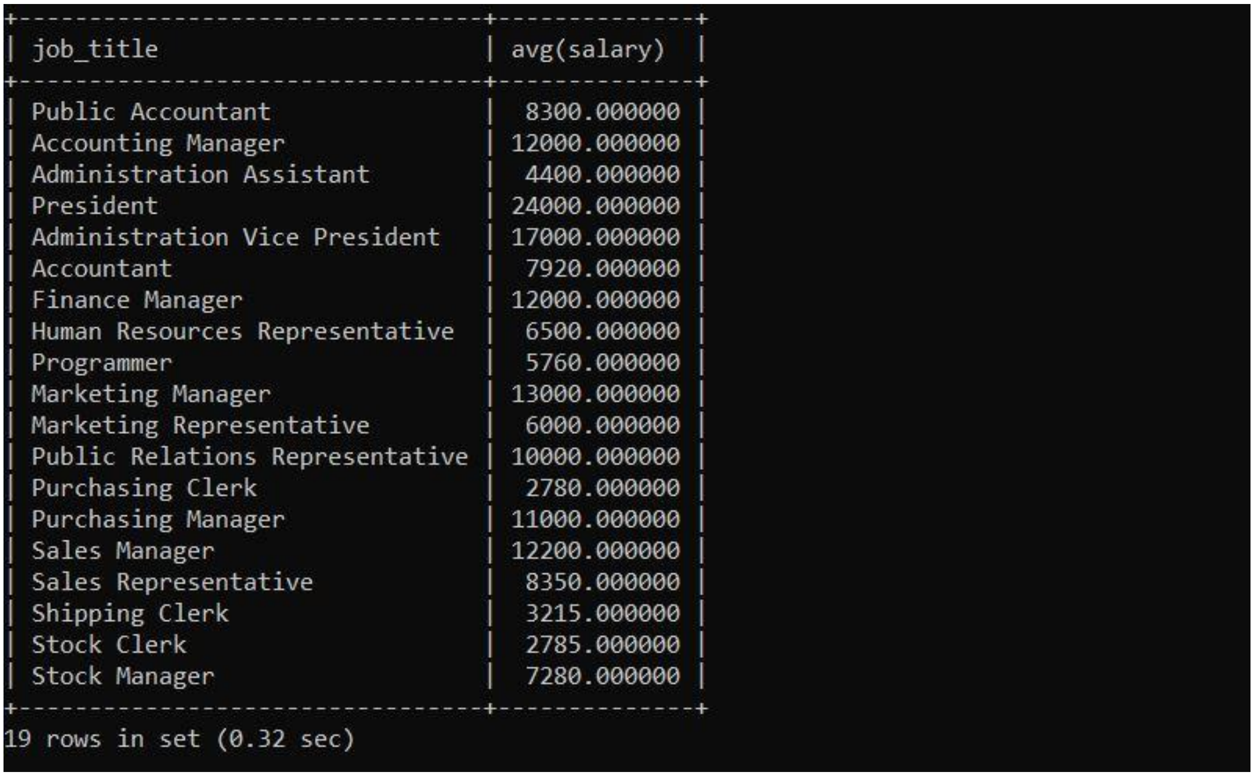
5) **NATURAL JOIN:**

Question1. Query to display the job title and average salary of employees**.**

Query:

select job_title,avg(salary)from employees natural join jobs group by job_title;

Output:



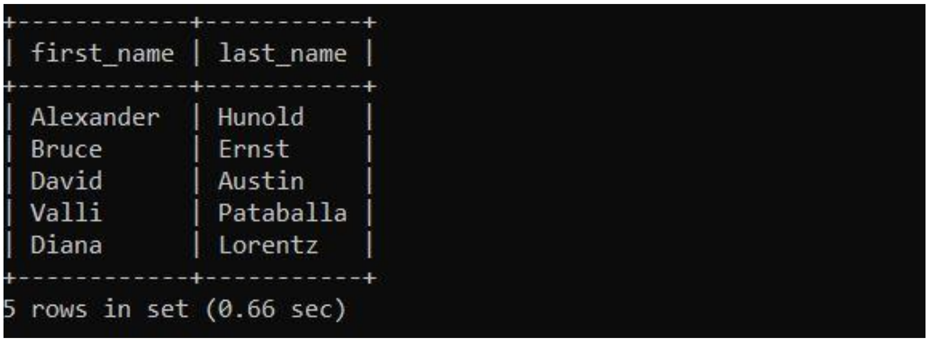
```
+----------------------------------+--------------+
| job_title                        | avg(salary)  |
+----------------------------------+--------------+
| Public Accountant                |  8300.000000 |
| Accounting Manager               | 12000.000000 |
| Administration Assistant         |  4400.000000 |
| President                        | 24000.000000 |
| Administration Vice President    | 17000.000000 |
| Accountant                       |  7920.000000 |
| Finance Manager                  | 12000.000000 |
| Human Resources Representative   |  6500.000000 |
| Programmer                       |  5760.000000 |
| Marketing Manager                | 13000.000000 |
| Marketing Representative         |  6000.000000 |
| Public Relations Representative  | 10000.000000 |
| Purchasing Clerk                 |  2780.000000 |
| Purchasing Manager               | 11000.000000 |
| Sales Manager                    | 12200.000000 |
| Sales Representative             |  8350.000000 |
| Shipping Clerk                   |  3215.000000 |
| Stock Clerk                      |  2785.000000 |
| Stock Manager                    |  7280.000000 |
+----------------------------------+--------------+
19 rows in set (0.32 sec)
```

Sub queries in MySQL

A MySQL subquery is a query nested within another query such as SELECT,INSERT,UPDATE or DELETE. Also, a subquery can be nested within another subquery.

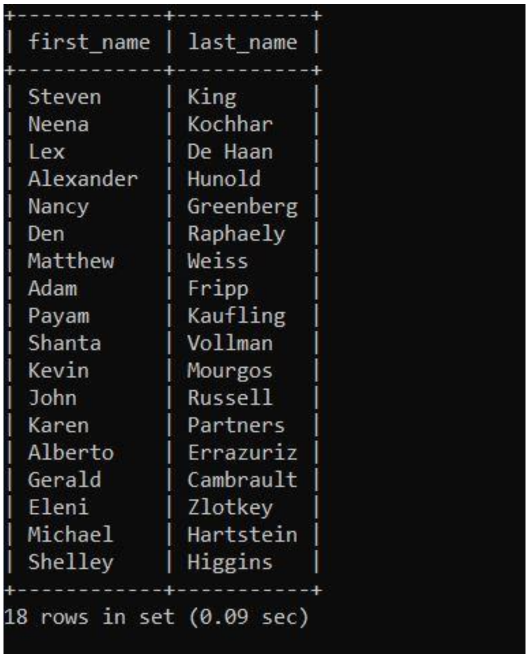
It returns either single value or row set.

Q1. Write a query to find the name (first_name, last_name) of all employees who works in the IT department.
Query: select first_name,last_name from employees where department_id in(select department_id from departments where department_name='IT');
Output:



```
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Alexander  | Hunold    |
| Bruce      | Ernst     |
| David      | Austin    |
| Valli      | Pataballa |
| Diana      | Lorentz   |
+------------+-----------+
5 rows in set (0.66 sec)
```

Q2. Write a query to find the name (first_name, last_name) of the employees who are managers.
Query: select first_name,last_name from employees where(employee_id in(select manager_id from employees));
Output:



```
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Steven     | King      |
| Neena      | Kochhar   |
| Lex        | De Haan   |
| Alexander  | Hunold    |
| Nancy      | Greenberg |
| Den        | Raphaely  |
| Matthew    | Weiss     |
| Adam       | Fripp     |
| Payam      | Kaufling  |
| Shanta     | Vollman   |
| Kevin      | Mourgos   |
| John       | Russell   |
| Karen      | Partners  |
| Alberto    | Errazuriz |
| Gerald     | Cambrault |
| Eleni      | Zlotkey   |
| Michael    | Hartstein |
| Shelley    | Higgins   |
+------------+-----------+
18 rows in set (0.09 sec)
```