Note : To solve below queries use “sales” database

1. Write a query that counts the number of salespeople registering orders for eachday. (If a salesperson has more than one order on a given day, he or she should be counted only once.).

mysql> SELECT count(distinct (snum)), odate FROM orders GROUP BY odate;

```
+------------------------+------------+
| count(distinct (snum)) | odate      |
+------------------------+------------+
|                      4 | 1990-10-03 |
|                      3 | 1990-10-04 |
+------------------------+------------+
```

2. Write a query on the Customers table that will find the highest rating in each city. Put the output in this form: For the city (city), the highest rating is : (rating).

mysql> SELECT city, MAX(rating) as RATING FROM customers GROUP BY city;

```
+----------+--------+
| city     | RATING |
+----------+--------+
| London   |    100 |
| Rome     |    200 |
| San Jose |    300 |
| Berlin   |    300 |
+----------+--------+
```

3 Write a query that totals the orders for each day and places the results in descending order.

mysql> SELECT sum(amt), odate FROM orders GROUP BY odate ORDER BY sum(amt) DESC;

```
+----------+------------+
| sum(amt) | odate      |
+----------+------------+
| 16713.81 | 1990-10-04 |
|  8944.59 | 1990-10-03 |
+----------+------------+
```

4. Write a query that selects the total amount in orders for each salesperson for whom this total is greater than the average amount of the order in the table. (Note Use the average amount value directly →2565.84)

mysql> SELECT SUM(amt), snum from orders GROUP BY snum HAVING sum(amt)>2565.84;

```
+----------+------+
| SUM(amt) | snum |
+----------+------+
| 15382.07 | 1001 |
|  5546.15 | 1002 |
+----------+------+
```

5. Write a query that selects the highest rating in each city.

mysql> SELECT city, MAX(rating) as RATING FROM customers GROUP BY city;

```
+----------+--------+
| city     | RATING |
+----------+--------+
| London   |    100 |
| Rome     |    200 |
| San Jose |    300 |
| Berlin   |    300 |
+----------+--------+
```

6. Largest order taken by each salesperson with order value more than Rs.3000.

mysql> SELECT MAX(amt), snum FROM orders GROUP BY snum HAVING MAX(amt)>3000;

```
+----------+------+
| MAX(amt) | snum |
+----------+------+
|  9891.88 | 1001 |
|  5160.45 | 1002 |
+----------+------+
```

7. Select each customer smallest order.

mysql> SELECT cnum, MIN(amt) FROM orders GROUP BY cnum;

```
+------+----------+
| cnum | MIN(amt) |
+------+----------+
| 2008 |    18.69 |
| 2001 |   767.19 |
| 2007 |  1900.10 |
| 2003 |  5160.45 |
| 2002 |  1713.23 |
| 2004 |    75.75 |
| 2006 |  4723.00 |
+------+----------+
```

Note : To solve below queries use “hr” database

1. Display manager ID and number of employees managed by the manager.

mysql> SELECT manager_id, count(email) FROM employees GROUP BY MANAGER_ID;

```
+------------+--------------+
| manager_id | count(email) |
+------------+--------------+
|          0 |            1 |
|        100 |           14 |
|        101 |            5 |
|        102 |            1 |
|        103 |            4 |
|        108 |            5 |
|        114 |            5 |
|        120 |            8 |
|        121 |            8 |
|        122 |            8 |
|        123 |            8 |
|        124 |            8 |
|        145 |            6 |
|        146 |            6 |
|        147 |            6 |
|        148 |            6 |
|        149 |            6 |
|        201 |            1 |
|        205 |            1 |
+------------+--------------+
```

2. Display the country ID and number of cities we have in the country.

mysql> SELECT COUNTRY_ID, COUNT(CITY) from locations GROUP BY COUNTRY_ID;

```
+------------+-------------+
| COUNTRY_ID | COUNT(CITY) |
+------------+-------------+
| "          |           1 |
| AU         |           1 |
| BR         |           1 |
| CA         |           2 |
| CH         |           2 |
| CN         |           1 |
| DE         |           1 |
| IN         |           1 |
| IT         |           2 |
| JP         |           2 |
| NL         |           1 |
| Ox         |           1 |
| SG         |           1 |
| UK         |           2 |
| US         |           4 |
+------------+-------------+
```

3. Display average salary of employees in each department who have commission percentage.

mysql> SELECT AVG(SALARY), DEPARTMENT_ID FROM employees WHERE COMMISSION_PCT!=0.00 GROUP BY DEPARTMENT_ID;

```
+-------------+---------------+
| AVG(SALARY) | DEPARTMENT_ID |
+-------------+---------------+
```

```
| 7000.000000 |            0 |
| 8955.882353 |           80 |
+-------------+--------------+
```

4. Display job ID, number of employees, sum of salary, and difference between highest salary and lowest salary of the employees of the job.

mysql> SELECT JOB_ID, COUNT(JOB_ID), sum(MIN_SALARY), sum(MAX_SALARY)-sum(MIN_SALARY) from JOBS GROUP BY JOB_ID;

```
+------------+---------------+----------------+---------------------------------+
| JOB_ID     | COUNT(JOB_ID) | sum(MIN_SALARY) | sum(MAX_SALARY)-sum(MIN_SALARY) |
+------------+---------------+----------------+---------------------------------+
| AC_ACCOUNT |             1 |           4200 |                            4800 |
| AC_MGR     |             1 |           8200 |                            7800 |
| AD_ASST    |             1 |           3000 |                            3000 |
| AD_PRES    |             1 |          20000 |                           20000 |
| AD_VP      |             1 |          15000 |                           15000 |
| FI_ACCOUNT |             1 |           4200 |                            4800 |
| FI_MGR     |             1 |           8200 |                            7800 |
| HR_REP     |             1 |           4000 |                            5000 |
| IT_PROG    |             1 |           4000 |                            6000 |
| MK_MAN     |             1 |           9000 |                            6000 |
| MK_REP     |             1 |           4000 |                            5000 |
| PR_REP     |             1 |           4500 |                            6000 |
| PU_CLERK   |             1 |           2500 |                            3000 |
| PU_MAN     |             1 |           8000 |                            7000 |
| SA_MAN     |             1 |          10000 |                           10000 |
| SA_REP     |             1 |           6000 |                            6000 |
| SH_CLERK   |             1 |           2500 |                            3000 |
| ST_CLERK   |             1 |           2000 |                            3000 |
| ST_MAN     |             1 |           5500 |                            3000 |
+------------+---------------+----------------+---------------------------------+
```

5. Display job ID for jobs with average salary more than 10000.

mysql> SELECT JOB_ID,AVG(SALARY) FROM employees GROUP BY JOB_ID HAVING AVG(SALARY)>10000;

```
+---------+--------------+
| JOB_ID  | AVG(SALARY)  |
+---------+--------------+
| AC_MGR  | 12000.000000 |
| AD_PRES | 24000.000000 |
| AD_VP   | 17000.000000 |
| FI_MGR  | 12000.000000 |
| MK_MAN  | 13000.000000 |
| PU_MAN  | 11000.000000 |
| SA_MAN  | 12200.000000 |
+---------+--------------+
```

6. Display years in which more than 10 employees joined.

mysql> SELECT YEAR(HIRE_DATE), COUNT(YEAR(HIRE_DATE)) FROM employees GROUP BY YEAR(HIRE_DATE) HAVING COUNT(YEAR(HIRE_DATE))>10;

```
+-----------------+------------------------+
| YEAR(HIRE_DATE) | COUNT(YEAR(HIRE_DATE)) |
+-----------------+------------------------+
|            1987 |                    107 |
+-----------------+------------------------+
```

7. Display departments in which more than five employees have commission percentage.

mysql> SELECT DEPARTMENT_ID,COUNT(COMMISSION_PCT) FROM employees GROUP BY DEPARTMENT_ID HAVING COUNT(COMMISSION_PCT)>5;

```
+---------------+-----------------------+
| DEPARTMENT_ID | COUNT(COMMISSION_PCT) |
+---------------+-----------------------+
|            30 |                     6 |
|            50 |                    45 |
|            80 |                    34 |
|           100 |                     6 |
+---------------+-----------------------+
```

8. Display employee ID for employees who did more than one job in the past.

mysql> SELECT EMPLOYEE_ID, count(EMPLOYEE_ID) FROM job_history GROUP BY EMPLOYEE_ID HAVING count(EMPLOYEE_ID)>1;

```
+-------------+--------------------+
| EMPLOYEE_ID | count(EMPLOYEE_ID) |
+-------------+--------------------+
|         101 |                  2 |
|         176 |                  2 |
|         200 |                  2 |
+-------------+--------------------+
```

9. Display job ID of jobs that were done by more than 3 employees for more than 100 days.

mysql> select job_id, count(employee_id),
-> sum(timestampdiff(day,start_date,end_date)) as days
-> from job_history group by job_id
-> having days>100 and count(employee_id)>3;

```
Empty set (0.04 sec)
```

10. Display department ID, year, and Number of employees joined.

mysql> select department_id, year(hire_date) as year,
-> count(employee_id) as count
-> from employees group by department_id,year;

```
+---------------+------+-------+
| department_id | year | count |
+---------------+------+-------+
|            90 | 1987 |     3 |
|            60 | 1987 |     5 |
|           100 | 1987 |     6 |
|            30 | 1987 |     6 |
|            50 | 1987 |    45 |
|            80 | 1987 |    34 |
|             0 | 1987 |     1 |
|            10 | 1987 |     1 |
|            20 | 1987 |     2 |
|            40 | 1987 |     1 |
|            70 | 1987 |     1 |
|           110 | 1987 |     2 |
+---------------+------+-------+
```

11. Display how many employees joined in each month of the current year.

mysql> SELECT count(EMPLOYEE_ID),MONTH(HIRE_DATE) from EMPLOYEES GROUP BY MONTH(HIRE_DATE);

```
+--------------------+------------------+
| count(EMPLOYEE_ID) | MONTH(HIRE_DATE) |
+--------------------+------------------+
|                 14 |                6 |
|                 31 |                7 |
|                 31 |                8 |
|                 30 |                9 |
|                  1 |               10 |
+--------------------+------------------+
```

12. Display details of departments in which the maximum salary is more than 10000.

mysql> SELECT DEPARTMENT_ID,MAX(SALARY) FROM employees GROUP BY DEPARTMENT_ID HAVING MAX(SALARY)>10000;

```
+---------------+-------------+
| DEPARTMENT_ID | MAX(SALARY) |
+---------------+-------------+
|            20 |    13000.00 |
|            30 |    11000.00 |
|            80 |    14000.00 |
|            90 |    24000.00 |
|           100 |    12000.00 |
|           110 |    12000.00 |
+---------------+-------------+
```