LAB NO. 10

Sub-Queries

Muhammad Sohaib

8719

Lab10

Objectives

To perform sub-queries using DML commands.

Instructions

- This is individual Lab work/task.
- Complete this lab work within lab timing.
- Discussion with peers is not allowed.
- You can consult any book, notes & Internet.
- Copy paste from Internet will give you negative marks.
- Lab work is divided into small tasks, complete all tasks sequentially.
- Show solution of each lab task to your Lab Instructor.
- Paste your solution (i.e. code) in given space under each task.
- Also make a zip/rar archive of all lab tasks and upload this file on LMS before leaving lab.
- In-Lab Exercises/Tasks

FACILITIES REQUIRED AND PROCEDURE

Facilities required to do the experiment

SNo	Facilities Required	Quantity
1	System	1
2	Operating System	Windows 8.1
3	Front End	Java
4	Back End	Oracle 11g, MySQL

Procedure for doing the experiment

Step	Detail
1	Subqueries The query within another is known as a subquery. A statement containing sub query is called parent statement. The rows returned by a subquery are used by the parent statement.
2	Types of Subqueries Single-row subquery Queries that return only one row from the inner SELECT statement. Multiple-row subqueries Queries that return more than one row from the inner SELECT statement. The results can be obtained using the operators IN, ANY and ALL. Multiple-column subqueries Queries that return more than one column from the inner

	SELECT statement. Correlated subquery A sub query is evaluated once for the entire parent statement whereas a correlated sub query is evaluated once per row processed by the parent statement.

SQL COMMANDS

Using a Subquery

Example:

```
select eno, ename
where salary >
(select sal
from employee
where ename =’JONES’);
```

Single-Row Subqueries

Example:

```
select ename, job
from emp
where job =
(select job
from emp
where empno =7369);
```

Multiple-Row Subqueries

Example:

```
select ename, sal, deptno
```

```
from emp
where sal in
(select min(sal)
from emp
group by deptno);
```

Multiple-Column Subqueries

Example:

```
select ordid, prodid, qty
from item
where  (prodid, qty) in
(select prodid, qty
from item
where ordid =605);
```

Correlated subquery

Example:

```
select *
from emp x
where x.salary >
(select avg(salary)
from emp
where deptno =x.deptno);
```

Tasks

Q1: Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with ‘M’.

Answer:

Use select from clause.

Use like operator to match job and in select clause to get the result.

SQL> select ename, sal from emp where sal >

(select min(sal) from emp where job like 'A%');

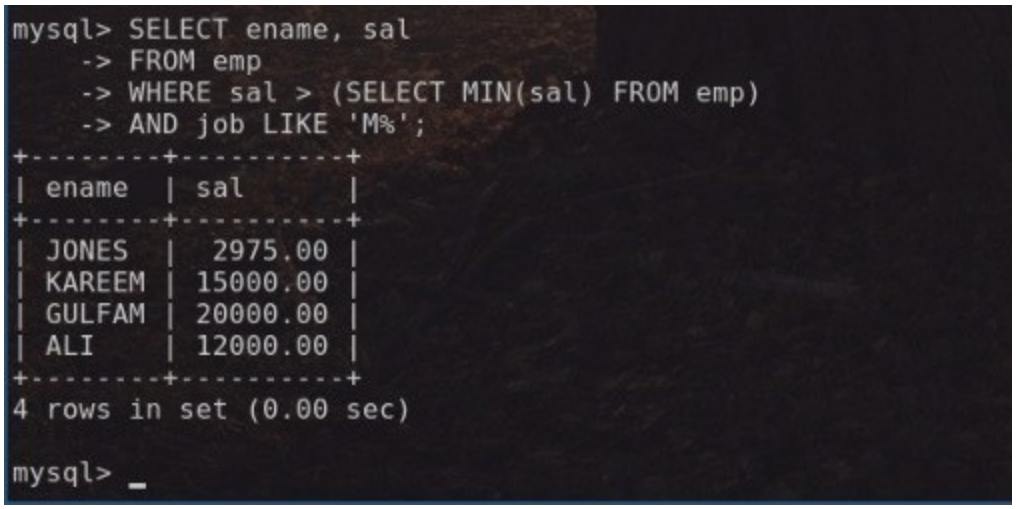
ENAME SAL

---------- ----------

Ali 12000

Gulfam 20000

Kareem 15000



Q2: Issue a query to display the employees whose job title is the same as that of employee 7369 and whose salary is greater than that of employee 7876.

Answer:

```
SQL> select ename, job
        from emp
        where job =
                (select job
                from emp
                where empno = 7369)
        and sal >
                (select sal
                from emp
                where empno =7876):
```

```
ENAME      JOB
----------      --------------------
MILLER     CLERK
```

```
mysql> SELECT ename, job
    -> FROM emp
    -> WHERE job = (SELECT job FROM emp WHERE empno = 7369)
    -> AND sal > (SELECT sal FROM emp WHERE empno = 7876);
+--------+-------+
| ename  | job   |
+--------+-------+
| MILLER | CLERK |
+--------+-------+
1 row in set (0.00 sec)

mysql>
```

Q3: Issue a query to display the employee name, job title and salary for all employees whose salary is equal to the minimum salary.

Answer:

SQL> select ename, job, sal

from emp

where sal =

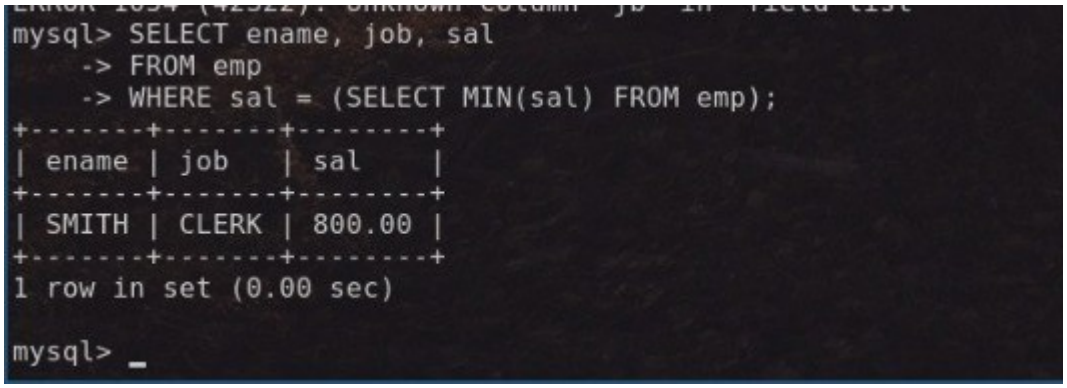
(select min(sal)

from emp);

ENAME JOB SAL

---------- ---------- ----------

SMITH CLERK 800



Q4: Issue a query to display all the that have a minimum salary greater than that of department 20.

Answer:

SQL> select deptno, min(sal)

from emp

group by deptno

havint min(sal) >

(select min(sal)

from emp

where deptno = 20);

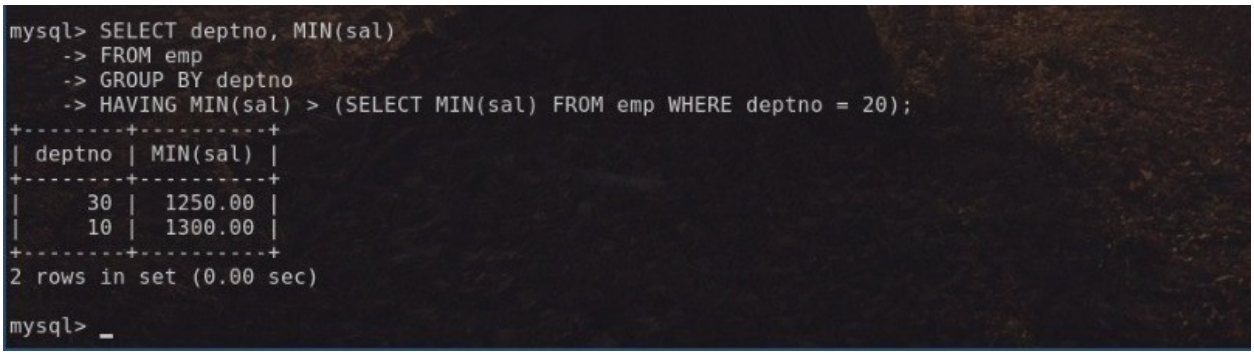
DEPTNO MIN(SAL)

---------- --------------------

10 1300

30 950

Multi-row Subqueries



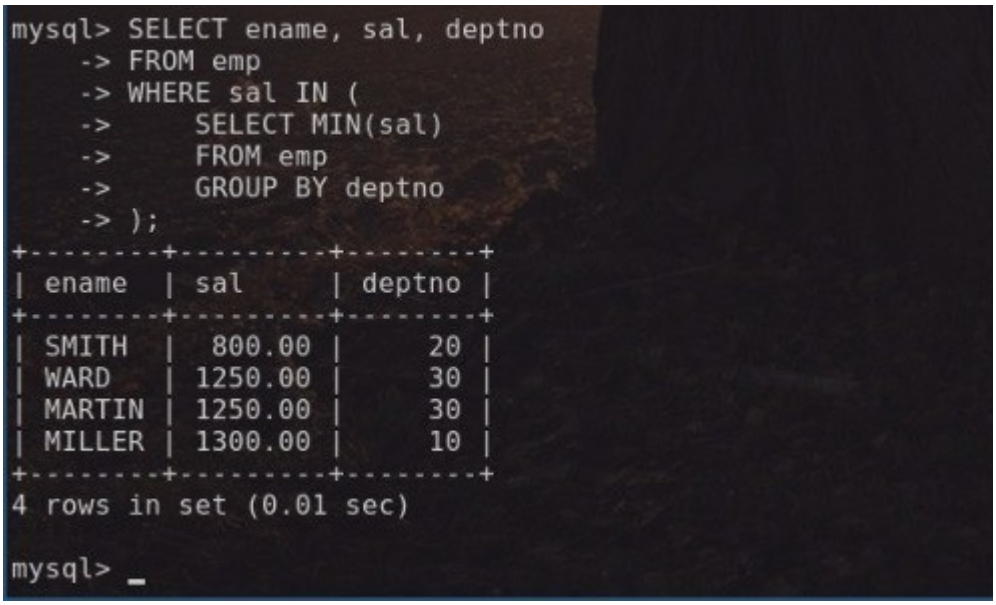
Q5: List the employee name, salary and department No for all employees who earn the same salary as the minimum salary for the department.

Answer:

SQL> select ename, sal, deptno

from emp

where sal in (800, 950, 1300);



```
mysql> SELECT ename, sal, deptno
    -> FROM emp
    -> WHERE sal IN (
    ->     SELECT MIN(sal)
    ->     FROM emp
    ->     GROUP BY deptno
    -> );
+--------+---------+--------+
| ename  | sal     | deptno |
+--------+---------+--------+
| SMITH  |  800.00 |     20 |
| WARD   | 1250.00 |     30 |
| MARTIN | 1250.00 |     30 |
| MILLER | 1300.00 |     10 |
+--------+---------+--------+
4 rows in set (0.01 sec)

mysql> _
```

Q6: Issue a query to display the employee No, name, job title whose salary is less than any clerk and who are not clerk.

Answer:

SQL> select empno, ename, job

from emp

where sal < any

(select sal

from emp

where job = ‘CLERK’)

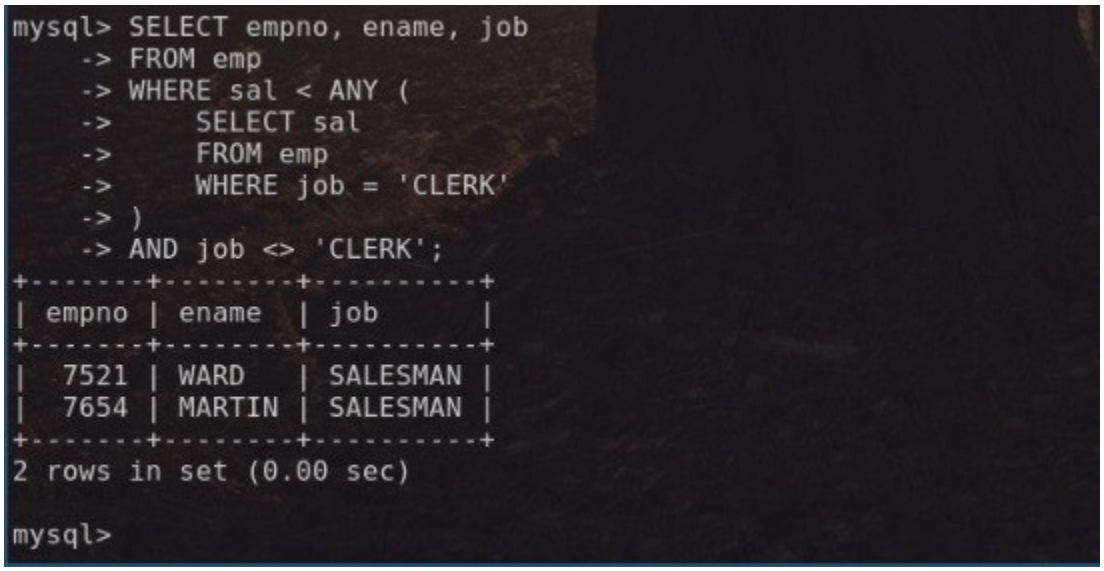
and job <> ‘CLERK’;

EMPNO ENAME JOB

---------- ---------- ----------

7654 MARTIN SALESMAN

7521 WARD SALESMAN



```
mysql> SELECT empno, ename, job
    -> FROM emp
    -> WHERE sal < ANY (
    ->     SELECT sal
    ->     FROM emp
    ->     WHERE job = 'CLERK'
    -> )
    -> AND job <> 'CLERK';
+-------+--------+----------+
| empno | ename  | job      |
+-------+--------+----------+
|  7521 | WARD   | SALESMAN |
|  7654 | MARTIN | SALESMAN |
+-------+--------+----------+
2 rows in set (0.00 sec)

mysql>
```

Q7: Issue a query to display the employee No, name, job title whose salary is greater than the average salaries of all department.

Answer:

SQL> select empno, ename, job

from emp

where sal > all

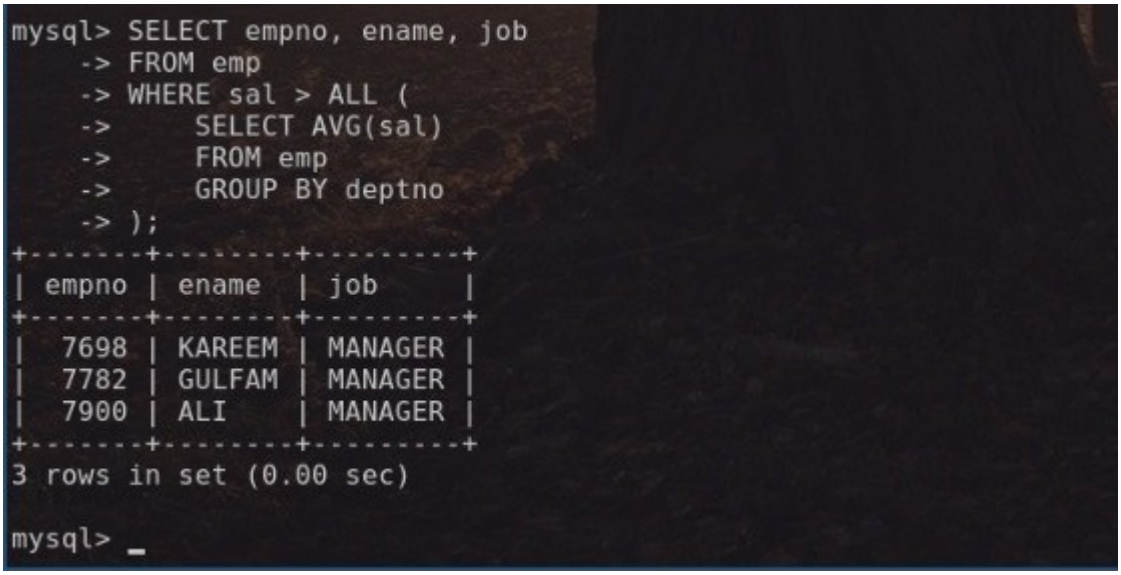
(select avg(sal)

from emp

group by deptno);

EMPNO	ENAME	JOB
7839	KING	PRESIDENT
7566	JONES	MANAGER
7902	FORD	ANALYST
7788	SCOTT	ANALYST

Multiple-column Subqueries



```
mysql> SELECT empno, ename, job
    -> FROM emp
    -> WHERE sal > ALL (
    ->     SELECT AVG(sal)
    ->     FROM emp
    ->     GROUP BY deptno
    -> );
+-------+--------+---------+
| empno | ename  | job     |
+-------+--------+---------+
|  7698 | KAREEM | MANAGER |
|  7782 | GULFAM | MANAGER |
|  7900 | ALI    | MANAGER |
+-------+--------+---------+
3 rows in set (0.00 sec)

mysql> _
```

Q8: Issue a query to display the order number, product number, and quantity of any item in which the product number and quantity match both the product number and quantity of an item in order 605.

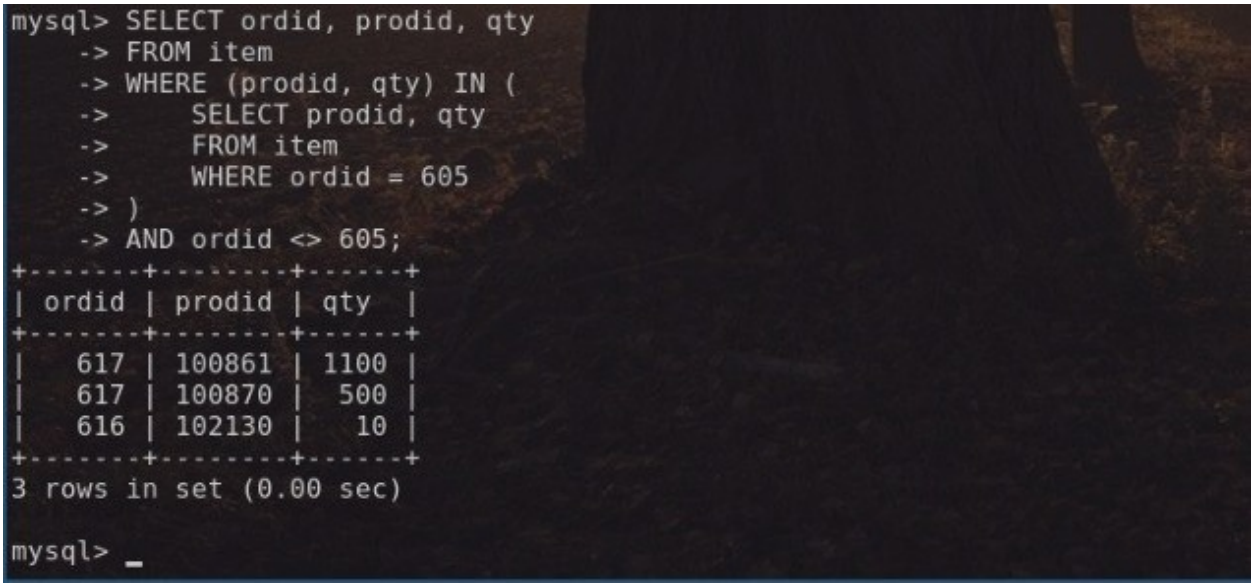
Answer:

```
SQL> select ordid, prodid, qty
from item
where (prodid, qty) in
(select prodid, qty
from item
where ordid = 605)
        and ordid <> 605;
```

```
ORDID       PRODID      QTY
 ---------- ------------------- --------------------
617         100861      100
```

617 100870 500

616 102130 10



Q9: Create a query to display the employee names, salaries, department numbers, and average salaries for all the employees who make more than the average salary in their department.

Answer:

SQL> select a.ename, a.deptno, b.salavg

from emp a,

(select deptno, avg(sal) salavg

from emp

group by deptno) b

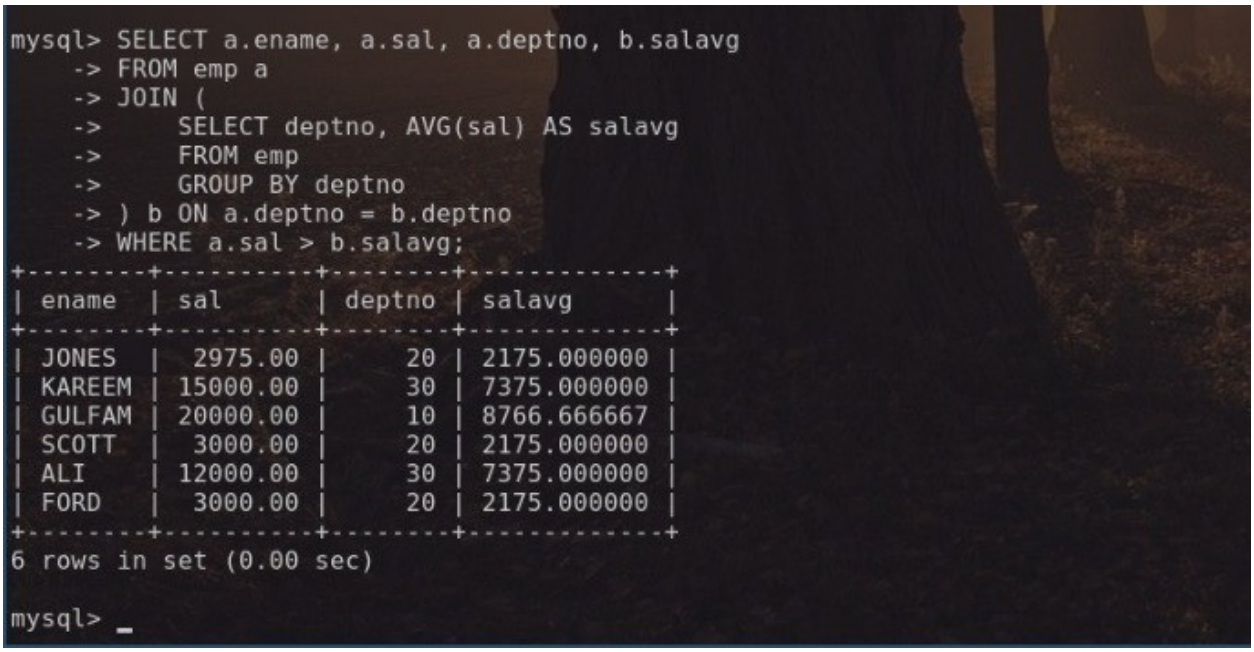
where a.deptno = b.deptno

and a.sal > b.

ENAME	SAL	DEPTNO	SALAVG
KING	5000	10	2916.6667
JONES	2975	20	2175
SCOTT	3000	20	2175

....

6 rows selected.



OUTCOME

Thus the nested queries and join queries were performed and executed successfully.

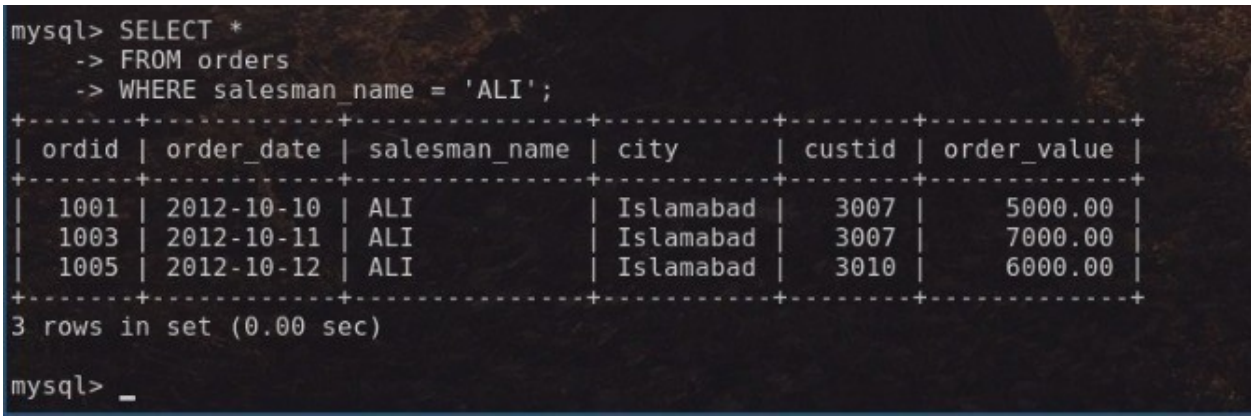
QUESTIONS AND ANSWERS

What is the use of sub queries?

A subquery is a select-from-where expression that is nested with in another query. A common use of subquery is to perform tests for set membership, make set comparisons, and determine set cardinality.

Extra Tasks

Q1. Write a query to display all the orders from the orders table issued by the salesman 'Ali’.



```
mysql> SELECT *
    -> FROM orders
    -> WHERE salesman_name = 'ALI';
+-------+------------+---------------+-----------+--------+-------------+
| ordid | order_date | salesman_name | city      | custid | order_value |
+-------+------------+---------------+-----------+--------+-------------+
|  1001 | 2012-10-10 | ALI           | Islamabad |   3007 |     5000.00 |
|  1003 | 2012-10-11 | ALI           | Islamabad |   3007 |     7000.00 |
|  1005 | 2012-10-12 | ALI           | Islamabad |   3010 |     6000.00 |
+-------+------------+---------------+-----------+--------+-------------+
3 rows in set (0.00 sec)

mysql> _
```

Q2. Write a query to display all the orders for the salesman who belongs to the city Islamabad.

```
mysql> SELECT o.*
    -> FROM orders o
    -> JOIN salesman s ON o.salesman_name = s.salesman_name
    -> WHERE s.city = 'Islamabad';
+-------+------------+---------------+-----------+--------+-------------+
| ordid | order_date | salesman_name | city      | custid | order_value |
+-------+------------+---------------+-----------+--------+-------------+
|  1001 | 2012-10-10 | ALI           | Islamabad |   3007 |     5000.00 |
|  1003 | 2012-10-11 | ALI           | Islamabad |   3007 |     7000.00 |
|  1005 | 2012-10-12 | ALI           | Islamabad |   3010 |     6000.00 |
|  1006 | 2012-10-10 | ALI           | Islamabad |   3011 |     4500.00 |
+-------+------------+---------------+-----------+--------+-------------+
4 rows in set (0.00 sec)

mysql> _
```

Q3. Write a query to find all the orders issued against the salesman who may works for customer whose id is 3007.

```
mysql> SELECT *
    -> FROM orders
    -> WHERE custid = 3007;
+-------+------------+---------------+-----------+--------+-------------+
| ordid | order_date | salesman_name | city      | custid | order_value |
+-------+------------+---------------+-----------+--------+-------------+
|  1001 | 2012-10-10 | ALI           | Islamabad |   3007 |     5000.00 |
|  1003 | 2012-10-11 | ALI           | Islamabad |   3007 |     7000.00 |
+-------+------------+---------------+-----------+--------+-------------+
2 rows in set (0.00 sec)

mysql>
```

Q4. Write a query to display all the orders which values are greater than the average order value for 10th October 2012.

```
mysql> SELECT *
    -> FROM orders
    -> WHERE order_date = '2012-10-10'
    -> AND order_value > (
    ->     SELECT AVG(order_value)
    ->     FROM orders
    ->     WHERE order_date = '2012-10-10'
    -> );
+-------+------------+--------------+-----------+--------+-------------+
| ordid | order_date | salesman_name | city      | custid | order_value |
+-------+------------+--------------+-----------+--------+-------------+
|  1001 | 2012-10-10 | ALI           | Islamabad |   3007 |     5000.00 |
|  1006 | 2012-10-10 | ALI           | Islamabad |   3011 |     4500.00 |
+-------+------------+--------------+-----------+--------+-------------+
2 rows in set (0.00 sec)

mysql> _
```

Q5. Write a query to find all orders attributed to a salesman in Islamabad.

```
mysql> SELECT o.*
    -> FROM orders o
    -> JOIN salesman s ON o.salesman_name = s.salesman_name
    -> WHERE s.city = 'Islamabad';
+-------+------------+--------------+-----------+--------+-------------+
| ordid | order_date | salesman_name | city      | custid | order_value |
+-------+------------+--------------+-----------+--------+-------------+
|  1001 | 2012-10-10 | ALI           | Islamabad |   3007 |     5000.00 |
|  1003 | 2012-10-11 | ALI           | Islamabad |   3007 |     7000.00 |
|  1005 | 2012-10-12 | ALI           | Islamabad |   3010 |     6000.00 |
|  1006 | 2012-10-10 | ALI           | Islamabad |   3011 |     4500.00 |
+-------+------------+--------------+-----------+--------+-------------+
4 rows in set (0.00 sec)

mysql>
```

Q6. Write a query to display the commission of all the salesmen servicing customers in Multan.

```
mysql> SELECT DISTINCT s.salesman_name, s.commission
    -> FROM salesman s
    -> JOIN orders o ON s.salesman_name = o.salesman_name
    -> WHERE o.city = 'Multan';
+--------------+------------+
| salesman_name | commission |
+--------------+------------+
| BILAL         |       0.13 |
| KAMRAN        |       0.11 |
+--------------+------------+
2 rows in set (0.01 sec)

mysql>
```