

Lab 5: Subqueries

MUHAMMAD SAAD HASSAN
P176137 BCS-6B

5.1 SELECT Subqueries:

```
mysql> SELECT TICKET_NO, TICKET_TYPE, TICKET_PRICE FROM TICKET WHERE TICKET_PRICE >= (SELECT AVG(TICKET_PRICE) FROM TICKET);
```

TICKET_NO	TICKET_TYPE	TICKET_PRICE
11001	Adult	24.99
13002	Adult	34.99
67833	Adult	28.67
88567	Child	22.50
88568	Adult	42.10

```
5 rows in set (0.02 sec)
```

```
mysql>
```

5.2 IN Subqueries:

```
mysql>
mysql> SELECT DISTINCT EMP_NUM, EMP_LNAME, EMP_FNAME, PARK_NAME
-> FROM EMPLOYEE NATURAL JOIN HOURS NATURAL JOIN ATTRACTION NATURAL JOIN THEMEPARK
-> WHERE PARK_CODE IN (SELECT THEMEPARK.PARK_CODE FROM THEMEPARK
-> WHERE PARK_NAME LIKE '%Fairy%');
```

EMP_NUM	EMP_LNAME	EMP_FNAME	PARK_NAME
100	Calderdale	Emma	FairyLand
105	Namova	Mirrelle	FairyLand

```
2 rows in set (0.02 sec)
```

```
mysql>
```

5.3 HAVING Subqueries:

```
mysql> SELECT PARK_CODE, SUM(LINE_QTY)
-> FROM SALES_LINE NATURAL JOIN TICKET
-> GROUP BY PARK_CODE
-> HAVING SUM(LINE_QTY) > (SELECT AVG(LINE_QTY) FROM SALES_LINE);
```

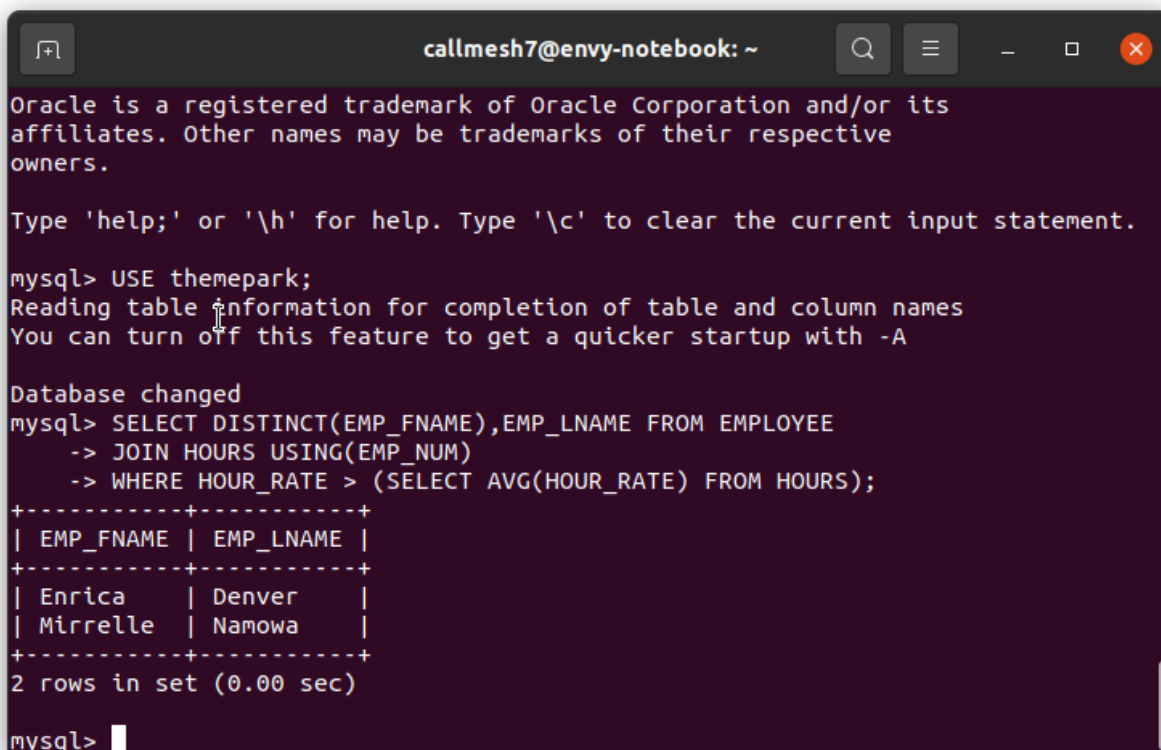
PARK_CODE	SUM(LINE_QTY)
FR1001	14
UK3452	29
ZA1342	18

```
3 rows in set (0.02 sec)
```

Exercises

E 5.1:

Write a query that displays the first name, last name of all employees who earn more than the average hourly rate. Do not display duplicate rows. Your output should match that shown in Figure E-5.1.



```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE themepark;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT DISTINCT(EMP_FNAME),EMP_LNAME FROM EMPLOYEE
-> JOIN HOURS USING(EMP_NUM)
-> WHERE HOUR_RATE > (SELECT AVG(HOUR_RATE) FROM HOURS);
```

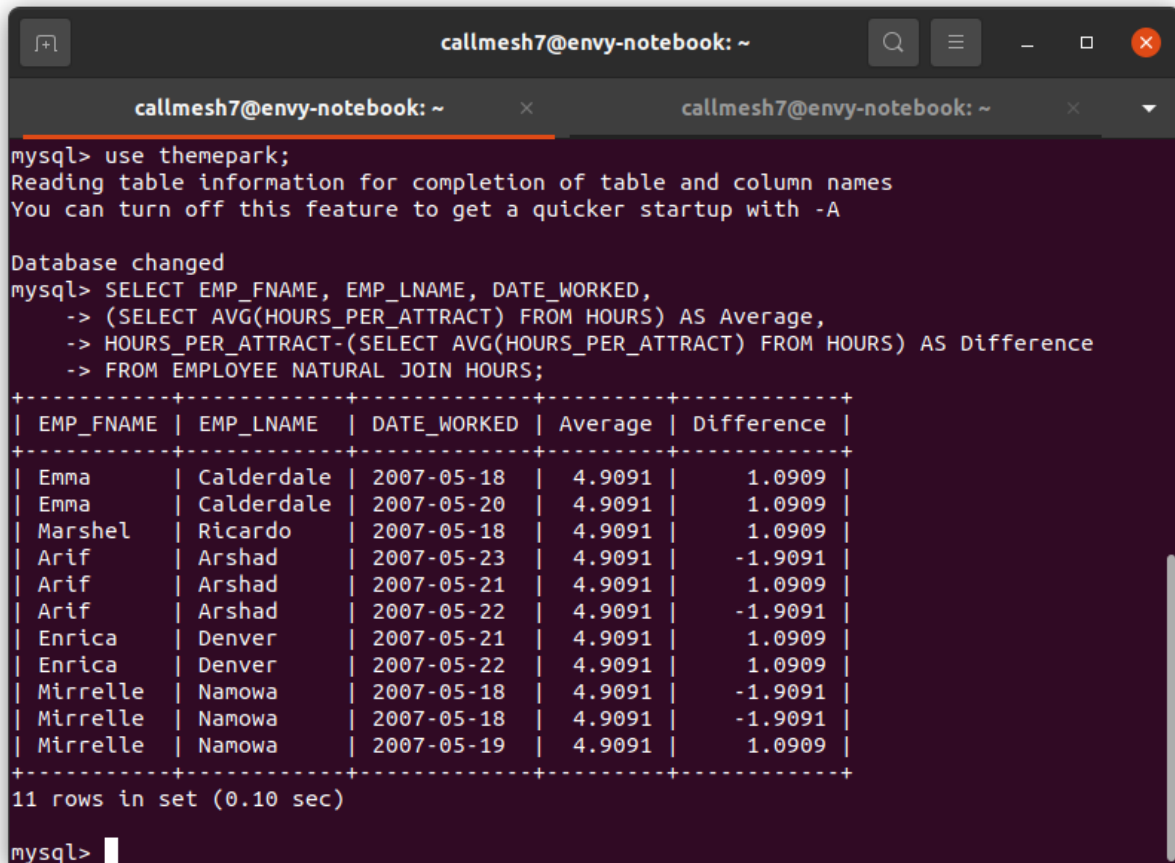
EMP_FNAME	EMP_LNAME
Enrica	Denver
Mirrelle	Namowa

```
2 rows in set (0.00 sec)

mysql>
```

E.5.2:

Write a query to display an employee's first name, last name and date worked which lists the difference between the number of hours an employee has worked on an attraction and the average hours worked on that attraction. Label this column 'DIFFERENCE' and the average hours column 'AVERAGE'.



```
callmesh7@envy-notebook: ~
mysql> use themepark;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT EMP_FNAME, EMP_LNAME, DATE_WORKED,
-> (SELECT AVG(HOURS_PER_ATTRACT) FROM HOURS) AS Average,
-> HOURS_PER_ATTRACT-(SELECT AVG(HOURS_PER_ATTRACT) FROM HOURS) AS Difference
-> FROM EMPLOYEE NATURAL JOIN HOURS;
+-----+-----+-----+-----+-----+
| EMP_FNAME | EMP_LNAME | DATE_WORKED | Average | Difference |
+-----+-----+-----+-----+-----+
| Emma      | Calderdale | 2007-05-18 | 4.9091 | 1.0909 |
| Emma      | Calderdale | 2007-05-20 | 4.9091 | 1.0909 |
| Marshal    | Ricardo    | 2007-05-18 | 4.9091 | 1.0909 |
| Arif       | Arshad     | 2007-05-23 | 4.9091 | -1.9091 |
| Arif       | Arshad     | 2007-05-21 | 4.9091 | 1.0909 |
| Arif       | Arshad     | 2007-05-22 | 4.9091 | -1.9091 |
| Enrica     | Denver     | 2007-05-21 | 4.9091 | 1.0909 |
| Enrica     | Denver     | 2007-05-22 | 4.9091 | 1.0909 |
| Mirrelle   | Namowa     | 2007-05-18 | 4.9091 | -1.9091 |
| Mirrelle   | Namowa     | 2007-05-18 | 4.9091 | -1.9091 |
| Mirrelle   | Namowa     | 2007-05-19 | 4.9091 | 1.0909 |
+-----+-----+-----+-----+-----+
11 rows in set (0.10 sec)

mysql>
```

E 5.3:

Type in and execute the two correlated subqueries in section 5.6 and check your output against that shown in figures 76 and 77.

```
mysql> SELECT TRANSACTION_NO, LINE_NO, LINE_QTY, LINE_PRICE
-> FROM SALES_LINE SL
-> WHERE SL.LINE_QTY > (SELECT AVG(LINE_QTY)
-> FROM SALES_LINE SA
-> WHERE SA. TRANSACTION_NO = SL. TRANSACTION_NO);
```

TRANSACTION_NO	LINE_NO	LINE_QTY	LINE_PRICE
12781	1	2	69.98
12785	3	4	139.96
34534	1	4	168.40
34537	1	2	84.20
34540	1	4	168.40

5 rows in set (0.00 sec)

```
mysql>
```

E 5.4:

Modify the second query you entered in E 5.3 to display all the theme parks where there have been no recorded tickets sales recently.

```
mysql> SELECT PARK_CODE, PARK_NAME, PARK_COUNTRY FROM THEMEPARK
-> WHERE NOT EXISTS (SELECT PARK_CODE FROM SALES WHERE SALES.PARK_CODE = THEMEPARK
.PARK_CODE);
```

PARK_CODE	PARK_NAME	PARK_COUNTRY
NL1202	Efling	NL
SP4533	AdventurePort	SP
SW2323	Labyrinthe	SW
UK2622	MiniLand	UK

4 rows in set (0.10 sec)

```
mysql>
```

E 5.5:

Write a query to find the attract capacity, with a attract capacity less than or equal to the average attract capacity.

```
mysql> SELECT ATTRACT_CAPACITY  
-> FROM ATTRACTION  
-> WHERE ATTRACT_CAPACITY <=(SELECT AVG(ATTRACT_CAPACITY) FROM ATTRACTION);
```

```
+-----+  
| ATTRACT_CAPACITY |  
+-----+  
|          34      |  
|          62      |  
|          24      |  
|          30      |  
|          40      |  
|          34      |  
|          42      |  
+-----+
```

7 rows in set (0.10 sec)

```
mysql> █
```