

DATA BASE LAB TASK 05

AITZAZ TAHIR CH 19P0012 SIR MUHAMMAD USMAN



**EXERCISE:**

**Task 5.1 Enter and execute the above query and compare your output with that shown in**

**Figure 71.**

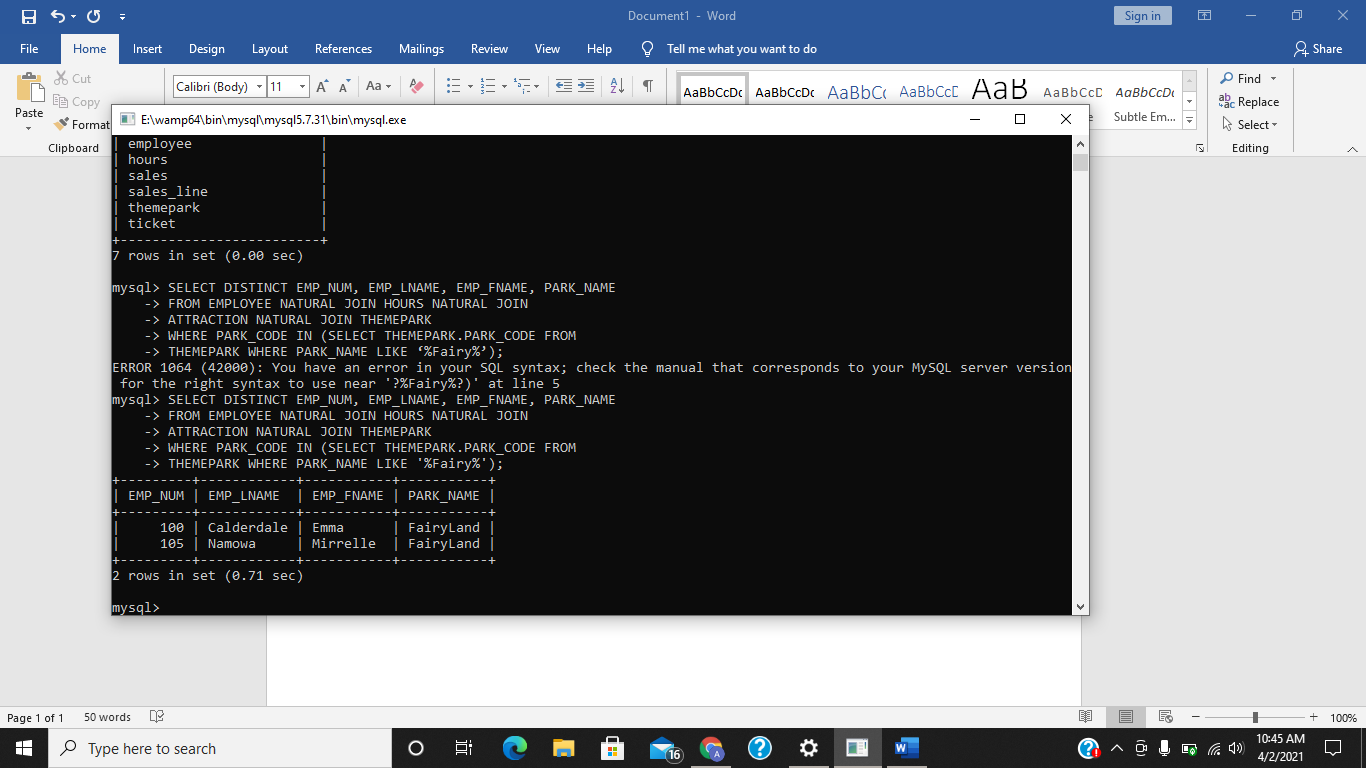
SELECT DISTINCT EMP\_NUM, EMP\_LNAME, EMP\_FNAME, PARK\_NAME

-> 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%');



**Task 5.2 Using the query above as a guide, write a new query to display the first and last**

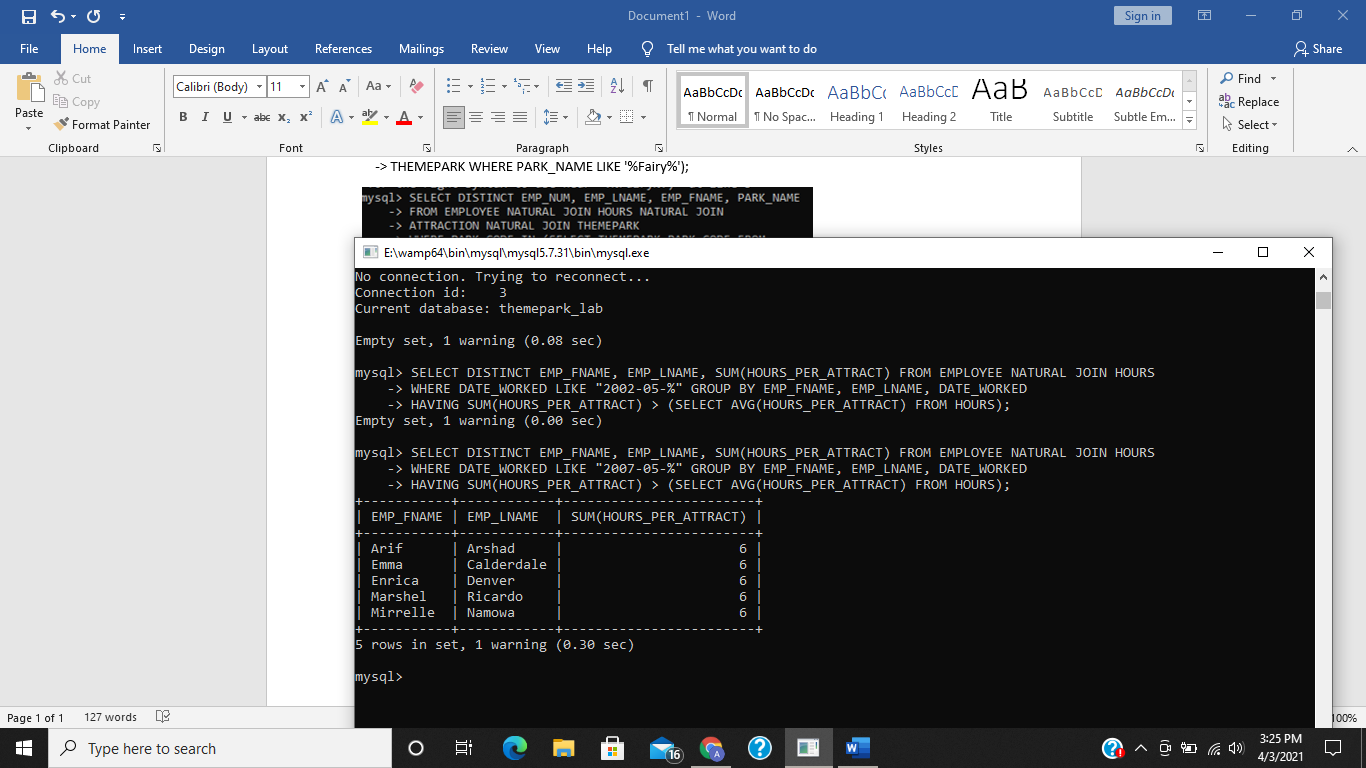
**names of all employees who have worked in total greater that the average number of**

**hours in total during May 2007. Your output should match that shown in Figure 73.**

SELECT DISTINCT EMP\_FNAME, EMP\_LNAME, SUM(HOURS\_PER\_ATTRACT) FROM EMPLOYEE NATURAL JOIN HOURS

-> WHERE DATE\_WORKED LIKE "2007-05-%" GROUP BY EMP\_FNAME, EMP\_LNAME, DATE\_WORKED

-> HAVING SUM(HOURS\_PER\_ATTRACT) > (SELECT AVG(HOURS\_PER\_ATTRACT) FROM HOURS);

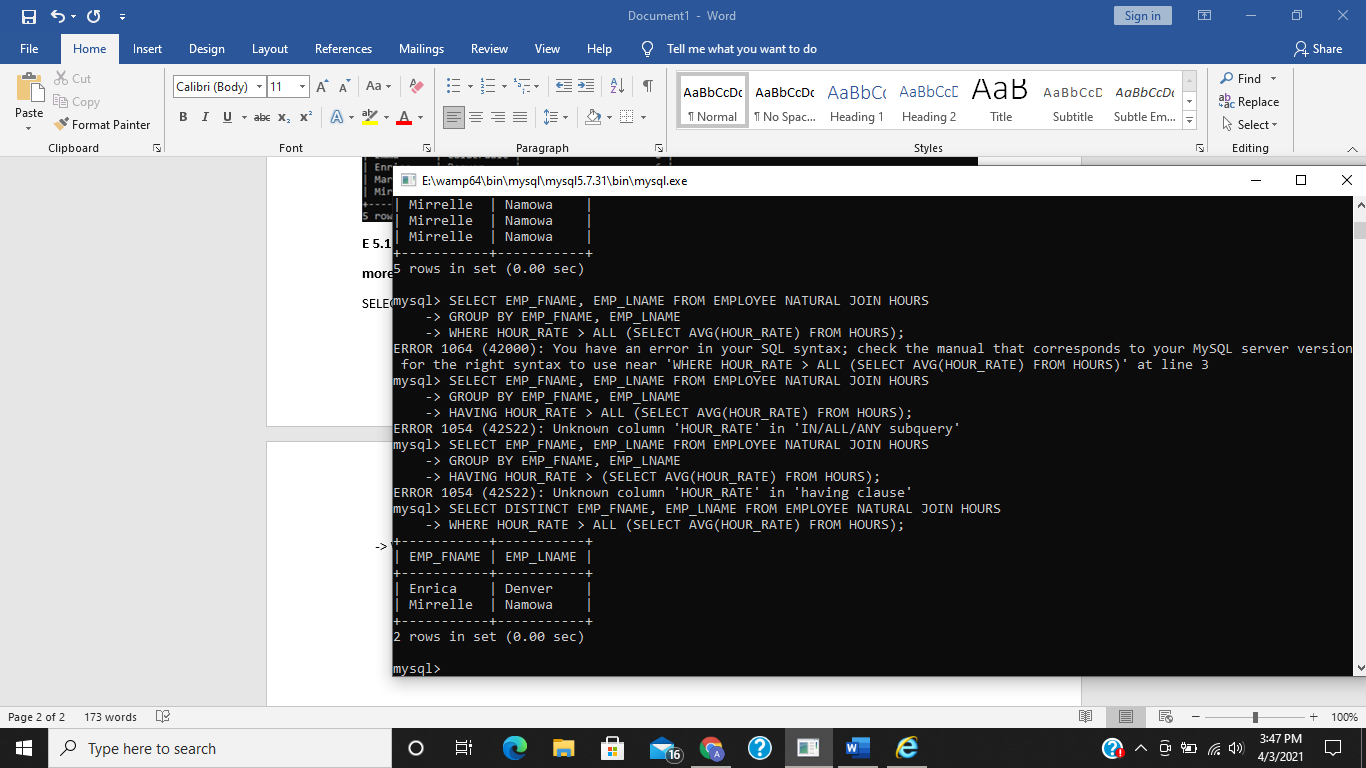


**E 5.1 Write a query that displays the first name, last name of all employees who earn**

**more than the average hourly rate. Do not display duplicate rows.**

SELECT DISTINCT EMP\_FNAME, EMP\_LNAME FROM EMPLOYEE NATURAL JOIN HOURS

-> WHERE HOUR\_RATE > ALL (SELECT AVG(HOUR\_RATE) FROM HOURS);



**E.5.2 Write a query to display an employee’s first name, last name and date worked**

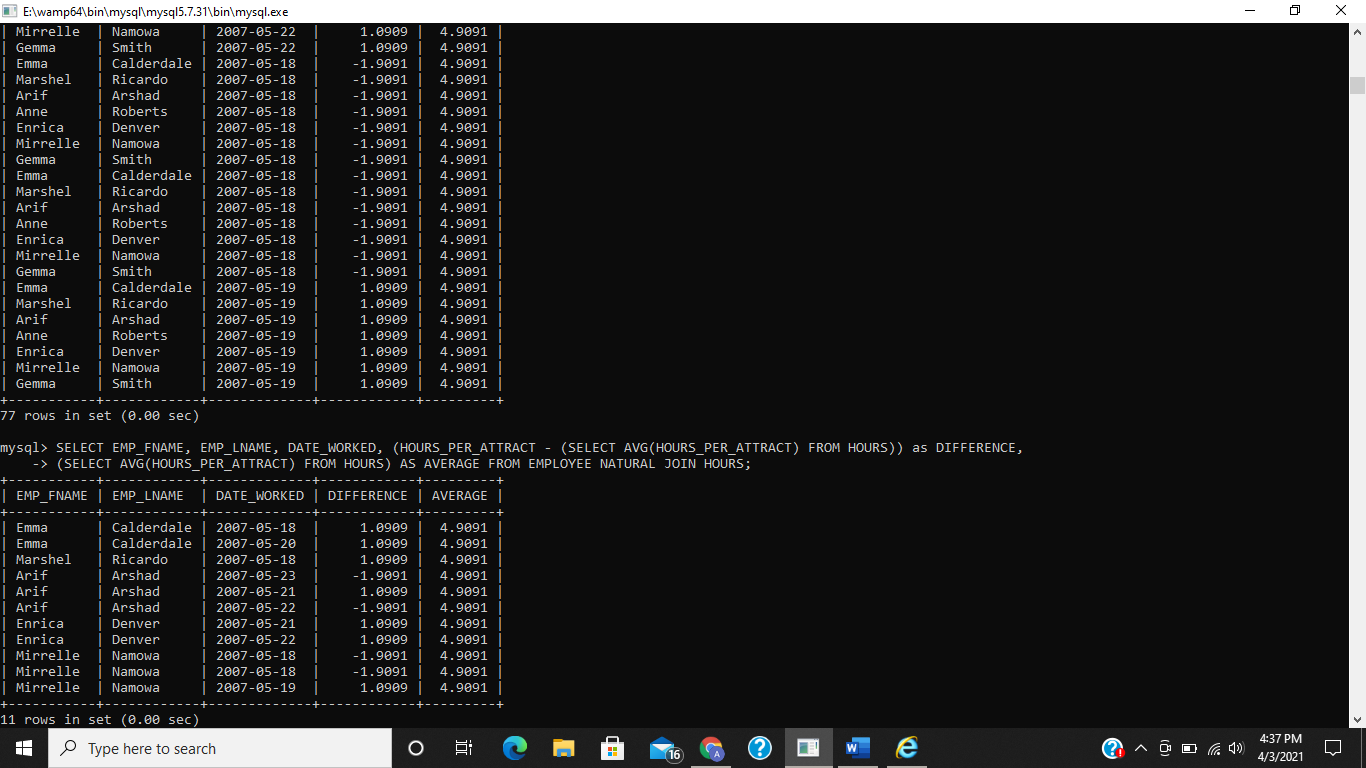
**which lists the difference between the number of hours an employee has worked on an**

**attraction and the average hours worked on that attraction. Label this column**

**‘DIFFERENCE’ and the average hours column ‘AVERAGE’.**

SELECT EMP\_FNAME, EMP\_LNAME, DATE\_WORKED, (HOURS\_PER\_ATTRACT - (SELECT AVG(HOURS\_PER\_ATTRACT) FROM HOURS)) as DIFFERENCE,

-> (SELECT AVG(HOURS\_PER\_ATTRACT) FROM HOURS) AS AVERAGE FROM EMPLOYEE NATURAL JOIN HOURS;



**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.**

**QUERY NO 1:**

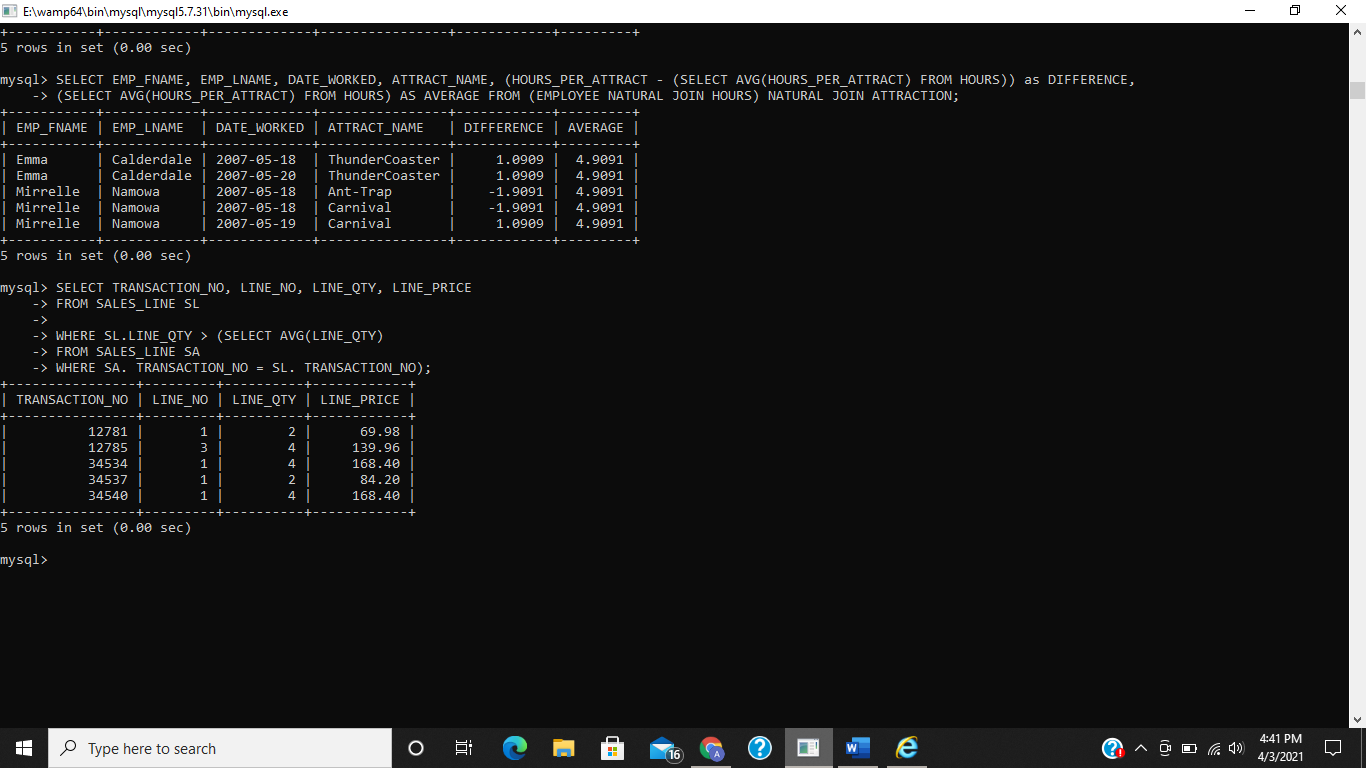
SELECT TRANSACTION\_NO, LINE\_NO, LINE\_QTY, LINE\_PRICE

FROM SALES\_LINE SL

WHERE SL.LINE\_QTY > (SELECT AVG(LINE\_QTY)

FROM SALES\_LINE SA

WHERE SA. TRANSACTION\_NO = SL. TRANSACTION\_NO);



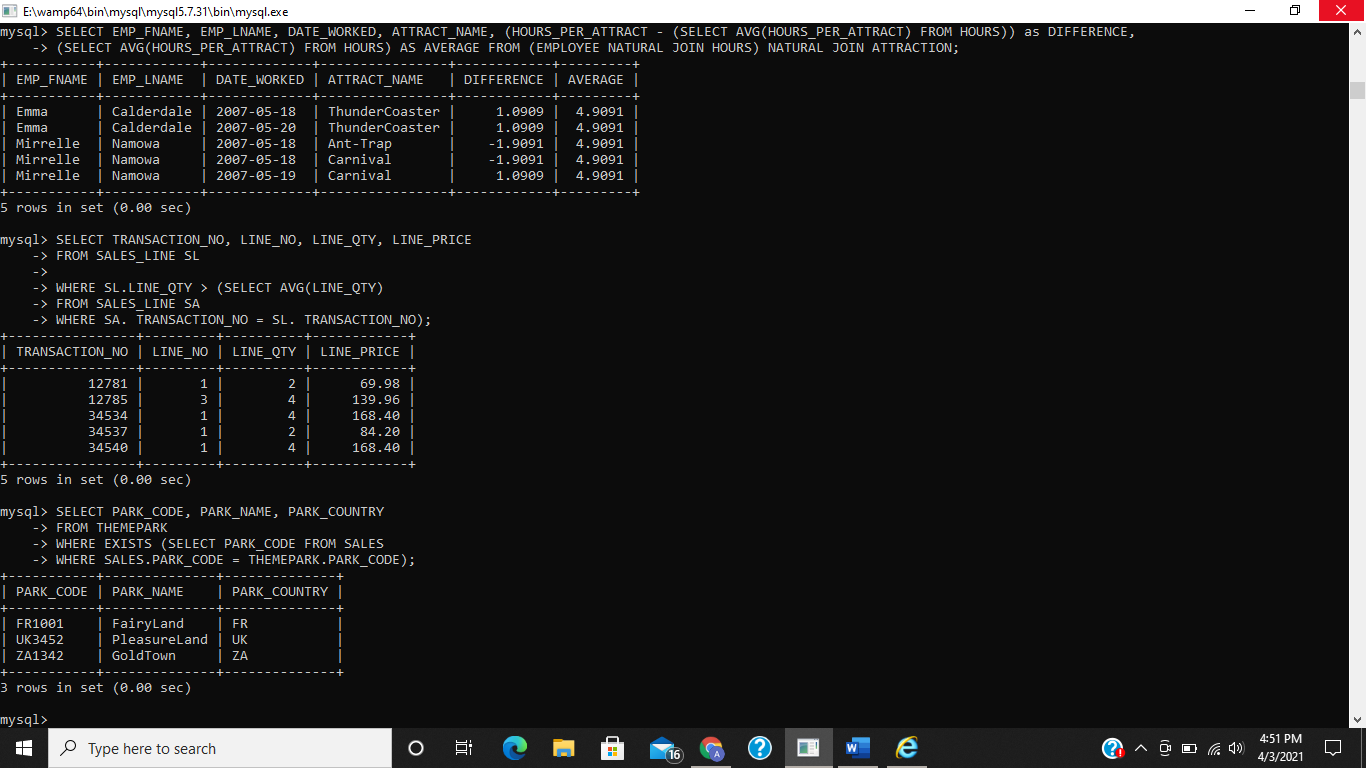
**QUERY NO 2:**

SELECT PARK\_CODE, PARK\_NAME, PARK\_COUNTRY

FROM THEMEPARK

WHERE EXISTS (SELECT PARK\_CODE FROM SALES

WHERE SALES.PARK\_CODE = THEMEPARK.PARK\_CODE);



**E 5.4 Modify the second query you entered in E 5.3 to display all the theme parks**

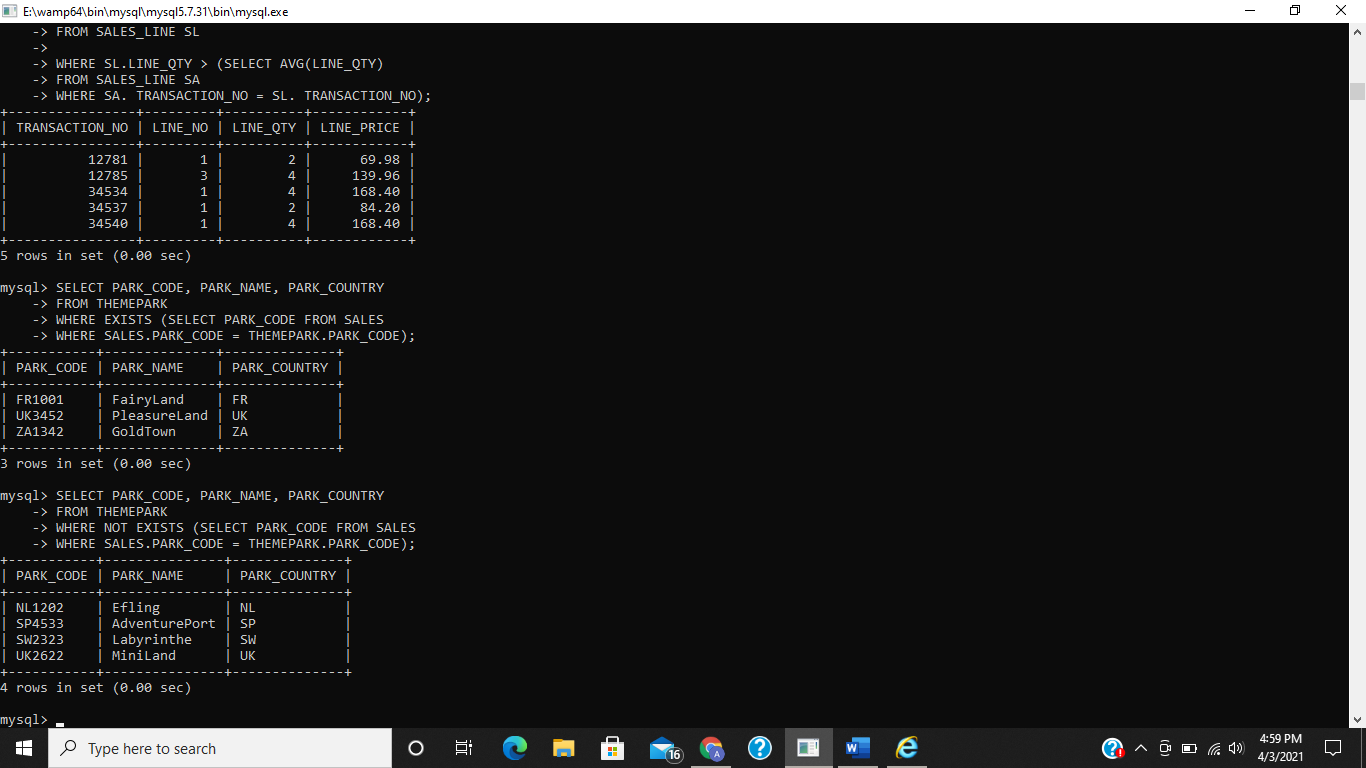
**where there have been no recorded tickets sales recently.**

SELECT PARK\_CODE, PARK\_NAME, PARK\_COUNTRY

-> FROM THEMEPARK

-> WHERE NOT EXISTS (SELECT PARK\_CODE FROM SALES

-> WHERE SALES.PARK\_CODE = THEMEPARK.PARK\_CODE);



**E 5.5 Write a query to find the attract capacity, with a attract capacity less than or equal**

**to the average attract capacity.**

SELECT ATTRACT\_NAME, ATTRACT\_CAPACITY FROM ATTRACTION

-> WHERE ATTRACT\_CAPACITY < (SELECT AVG(ATTRACT\_CAPACITY) FROM ATTRACTION)

-> OR ATTRACT\_CAPACITY = (SELECT AVG(ATTRACT\_CAPACITY) FROM ATTRACTION);

