

דוח שלב 2 צוריה מלכה 204843676 ואליסף כהן 311557227

1. שאליות select פרמטרים:

א. כל הלווים שלוו מעל 50 אלף ומשוייכים לסניפים שממוקמים בישראל (כלומר – ירושלים או תל אביב)

```
-- all the loaners who loaned over 50K from the branches located in israel

SELECT l.CUSTOMERID, c.CUSTOMERNAME, l.LOANAMOUNT, b.BRANCHELOCATION
FROM loans l
LEFT JOIN customers c ON l.CUSTOMERID = c.CUSTOMERID
LEFT JOIN branche b ON c.BRANCHEID = b.BRANCHEID
WHERE l.LOANAMOUNT >= 50000 AND c.BRANCHEID IN (
    SELECT b.BRANCHEID
    FROM branche b
    WHERE b.BRANCHELOCATION LIKE '%jerusalem%' or b.BRANCHELOCATION LIKE '%tel aviv%'
)
order by l.loanamount desc
```

	CUSTOMERID	CUSTOMERNAME	LOANAMOUNT	BRANCHELOCATION
1	100201	Brett	99500	jerusalem
2	100297	Daniel	96839	jerusalem
3	100281	Wendy	95618	jerusalem
4	100280	Cassandra	95179	tel aviv

ב. מיון הסניפים לפי ממוצע ההלוואות בסניף בסדר יורד

```
SELECT
    brancheID,
    brancheLocation,
    avg_loan_amount,
    num_loans,
    opened_date
FROM (
    SELECT
        brancheID,
        brancheLocation,
        avg_loan_amount,
        num_loans,
        opened_date
    FROM (
        SELECT
            c.brancheID AS brancheID,
            b.brancheLocation AS brancheLocation,
            AVG(l.loanAmount) OVER (PARTITION BY c.brancheID) AS avg_loan_amount,
            COUNT(l.loanID) OVER (PARTITION BY c.brancheID) AS num_loans,
            TO_CHAR(b.openedDate, 'YYYY-MM-DD') AS opened_date,
            ROW_NUMBER() OVER (PARTITION BY c.brancheID ORDER BY NULL) AS rn
        FROM
            Customers c
        JOIN
            Loans l ON c.customerID = l.customerID
        JOIN
            Branche b ON c.brancheID = b.brancheID
    )
    WHERE rn = 1
)
ORDER BY
    avg_loan_amount DESC;
```

	BRANCHEID	BRANCHELOCATION	AVG_LOAN_AMOUNT	NUM_LOANS	OPENED_DATE
1	103059	paris	99984	1	2001-05-09
2	103119	madrid	99825	1	2004-05-22
3	103250	new york	99750	1	2000-09-08
4	103201	jerusalem	99500	1	2002-12-14

ג. חמשת הלווים בעלי ההלוואות הכי גבוהות

```
SELECT
  c.customerName,
  SUM(l.loanAmount) AS total_loan_amount,
  b.branchID,
  b.branchLocation
FROM
  Loans l
JOIN
  Customers c ON l.customerID = c.customerID
JOIN
  Branche b ON c.branchID = b.branchID
GROUP BY
  c.customerID, c.customerName, b.branchID, b.branchLocation
ORDER BY
  total_loan_amount DESC
FETCH FIRST 5 ROWS ONLY;
```

	CUSTOMERNAME	TOTAL_LOAN_AMOUNT	BRANCHEID	BRANCHELOCATION
1	Ashley	99984	103069	paris
2	Christian	99825	103119	madrid
3	Nancy	99750	103250	new york
4	Brett	99500	103201	jerusalem
5	Robert	99412	103347	chicago

ד. מספר ההלוואות בסניפי העיר דנור במהלך שנת 2012 בחלוקה לחודשים

```
-- Find the total number of loans per month in 2012 for the cranches in tel aviv

SELECT
  TO_CHAR(l.loanDate, 'YYYY-MM') AS loan_month,
  COUNT(l.loanID) AS num_loans,
  b.branchLocation
FROM
  Loans l
JOIN
  Customers c ON l.customerID = c.customerID
JOIN
  Branche b ON c.branchID = b.branchID
WHERE
  TO_CHAR(l.loanDate, 'YYYY') = '2012' AND b.branchLocation = 'denver'
GROUP BY
  TO_CHAR(l.loanDate, 'YYYY-MM'), b.branchLocation
ORDER BY
  loan_month;
```

	LOAN_MONTH	NUM_LOANS	BRANCHELOCATION
1	2012-01	1	denver
2	2012-02	1	denver
3	2012-03	1	denver
4	2012-04	3	denver
5	2012-06	2	denver

2. שאילות עם פרמטרים:

א. מחזיר את כל העובדים שהם מנהלים או פקידים

```
15 # 1. Query Using a List Parameter
16 roles = ['Manager', 'Officer']
17 cursor.execute("""
18     SELECT workerID, workerName, workerRole, brancheID, teamID
19     FROM Workers
20     WHERE workerRole IN (:role1, :role2)
21     """, role1=roles[0], role2=roles[1])
22 for row in cursor:
23     print(row)
```

Run

C:\Users\tsuri\AppData\Local\Programs\Python\Python39\python.exe C:\Users\tsuri\PycharmProjects\pythonProject6\params_queries.py

Connected to the Oracle database

(105000, 'Sharon', 'Manager', 103000, 104000)

(105001, 'Susan', 'Officer', 103001, 104000)

(105004, 'Alvin', 'Manager', 103004, 104001)

(105005, 'Hector', 'Officer', 103005, 104001)

ב. מחזיר את כל הלקוחות שתאריך ההצטרפות שלהם הוא אחרי 01/01/2007

```
25 # 2. Query Using a Hint
26 join_date = '2007-01-01'
27 cursor.execute("""
28     SELECT customerID, customerName, accountID, joinDate, brancheID
29     FROM Customers
30     WHERE joinDate > TO_DATE(:join_date, 'YYYY-MM-DD')
31     """, join_date=join_date)
32 for row in cursor:
33     print(row)
```

Run

(103018, 'new york', 102018, datetime.datetime(2003, 10, 11, 0, 0))

(103032, 'tel aviv', 102032, datetime.datetime(2003, 10, 17, 0, 0))

(103051, 'chicago', 102051, datetime.datetime(2003, 11, 10, 0, 0))

(103110, 'denver', 102110, datetime.datetime(2003, 11, 16, 0, 0))

ג. כל הלווים שההלוואה שלהם גדולה מ95000 בסדר יורד

```
36 min_loan_amount = 95000
37 cursor.execute("""
38     SELECT loanID, loanAmount, customerID, loanDate
39     FROM Loans
40     WHERE loanAmount > :min_loan_amount
41     ORDER BY loanAmount DESC
42     """, min_loan_amount=min_loan_amount)
43 for row in cursor:
44     print(row)
```

Run

C:\Users\tsuri\AppData\Local\Programs\Python\Python39\python.exe C:\Users\tsuri\PycharmProjects\pythonProject6\params_queries.py

Connected to the Oracle database

(101069, 99984, 100069, datetime.datetime(2015, 3, 2, 0, 0))

(101119, 99825, 100119, datetime.datetime(2011, 1, 12, 0, 0))

(101250, 99750, 100250, datetime.datetime(2011, 5, 11, 0, 0))

(101201, 99500, 100201, datetime.datetime(2014, 1, 4, 0, 0))

ד.

מחזיר את כל הסניפים שתאריך הקמתם הוא בין 01/01/2002 לבין 31/12/2003

```
46 # # 4. Query Using a Date Parameter
47 start_date = '2002-01-01'
48 end_date = '2003-12-31'
49 cursor.execute("""
50     SELECT BrancheID, BrancheLocation, reporterID, openedDate
51     FROM Branche
52     WHERE openedDate BETWEEN TO_DATE(:start_date, 'YYYY-MM-DD') AND TO_DATE(:end_date, 'YYYY-MM-DD')
53     ORDER BY openedDate
54     """, start_date=start_date, end_date=end_date)
55 for row in cursor:
56     print(row)
57
58 # Close the cursor and connection
59 except cx_Oracle.DatabaseError as e:
60     # Handle database connection errors
61     try:
```

Run

C:\Users\tsuri\AppData\Local\Programs\Python\Python39\python.exe C:\Users\tsuri\PycharmProjects\pythonProject\params_queries.py

Connected to the Oracle database

(103136, 'tel aviv', 102136, datetime.datetime(2002, 1, 8, 0, 0))

(103122, 'new york', 102122, datetime.datetime(2002, 1, 13, 0, 0))

(103386, 'new york', 102386, datetime.datetime(2002, 1, 19, 0, 0))

(103038, 'denver', 102038, datetime.datetime(2002, 1, 30, 0, 0))

3. מחיקה:

א. מוחק את כל ההלוואות של לקוחות שבסניף תל אביב ושתאריך ההצטרפות הוא 2012

```
DECLARE
    city_name VARCHAR2(100) := 'tel aviv';
    target_year VARCHAR2(4) := '2005';
BEGIN
    DELETE FROM Loans
    WHERE customerID IN (
        SELECT c.customerID
        FROM Customers c
        JOIN Branche b ON c.brancheID = b.brancheID
        WHERE TO_CHAR(c.joinDate, 'YYYY') = target_year
        AND b.brancheLocation = city_name
    );
```

לפני

```
select *
FROM Loans
WHERE customerID IN (
    SELECT c.customerID
    FROM Customers c
    JOIN Branche b ON c.brancheID = b.brancheID
    WHERE TO_CHAR(c.joinDate, 'YYYY') = '2005'
    AND b.brancheLocation = 'tel aviv'
);
```

	LOANID	LOANAMOUNT	CUSTOMERID	LOANDATE
1	101000	12147	100000	4/7/2015
2	101016	23995	100016	12/3/2015
3	101024	1655	100024	4/24/2015
4	101072	4154	100072	11/26/2010
5	101096	44750	100096	11/6/2010
6	101248	32270	100248	7/19/2013
7	101288	42018	100288	1/9/2015
8	101368	16458	100368	10/20/2011

אחרי

```

select *
FROM Loans
WHERE customerID IN (
  SELECT c.customerID
  FROM Customers c
  JOIN Branches b ON c.branchID = b.branchID
  WHERE TO_CHAR(c.joinDate, 'YYYY') = '2005'
  AND b.branchLocation = 'tel aviv'
);

```

19:1 2601 SYS@XE AS SYSDBA 110:37:42 PM 0 rows selected in 0.024 seconds

ב. מוחק את ההלוואה הכי נמוכה מכל סניף

```

DELETE FROM Loans
WHERE customerID IN (
  SELECT customerID
  FROM (
    SELECT
      c.customerID,
      c.branchID,
      SUM(l.loanAmount) AS total_loan_amount,
      ROW_NUMBER() OVER (PARTITION BY c.branchID ORDER BY SUM(l.loanAmount)) AS rn
    FROM
      Customers c
    JOIN
      Loans l ON c.customerID = l.customerID
    GROUP BY
      c.customerID, c.branchID
  )
  WHERE rn = 1
);

```

לפני:

```


select *
from Loans
WHERE customerID IN (
  SELECT customerID
  FROM (
    SELECT
      c.customerID,
      c.branchID,
      SUM(l.loanAmount) AS total_loan_amount,
      ROW_NUMBER() OVER (PARTITION BY c.branchID ORDER BY SUM(l.loanAmount)) AS rn
    FROM
      Customers c
    JOIN
      Loans l ON c.customerID = l.customerID
    GROUP BY
      c.customerID, c.branchID
  )
  WHERE rn = 1
);

```

	LOANID	LOANAMOUNT	CUSTOMERID	LOANDATE	
1	101000	16164	100000	4/4/2015	...
2	101001	44457	100001	4/19/2013	...
3	101002	79477	100002	7/7/2014	...
4	101003	5294	100003	6/15/2015	...
5	101004	48950	100004	4/5/2015	...

אחרי:

```
select *
from Loans
WHERE customerID IN (
  SELECT customerID
  FROM (
    SELECT
      c.customerID,
      c.brancheID,
      SUM(l.loanAmount) AS total_loan_amount,
      ROW_NUMBER() OVER (PARTITION BY c.brancheID ORDER BY SUM(l.loanAmount)) AS
  FROM
    Customers c
  JOIN
    Loans l ON c.customerID = l.customerID
  GROUP BY
    c.customerID, c.brancheID
)
WHERE rn = 1
```



LOANID	LOANAMOUNT	CUSTOMERID	LOANDATE
--------	------------	------------	----------

41:1 SYS@XE AS SYSDBA [10:55:08 PM] 0 rows selected in 0.037 seconds

Update: 3.

א. מעדכנת את סכום ההלוואה של כל הלווים בברלין להיות על סך 15000

```
DECLARE
  branch_name VARCHAR2(100) := 'berlin';
  new_loan_amount NUMBER := 15000; -- New loan amount to set
BEGIN
  UPDATE Loans
  SET loanAmount = new_loan_amount
  WHERE customerID IN (
    SELECT c.customerID
    FROM Customers c
    JOIN Branche b ON c.brancheID = b.brancheID
    WHERE b.brancheLocation = branch_name
  );
```

לפני:

```

select *
from Loans
WHERE customerID IN (
    SELECT c.customerID
    FROM Customers c
    JOIN Branche b ON c.brancheID = b.brancheID
    WHERE b.brancheLocation = 'berlin'
);

```

	LOANID	LOANAMOUNT	CUSTOMERID	LOANDATE	
1	101004	33788	100004	11/24/2010	...
2	101012	13876	100012	3/24/2013	...
3	101020	66261	100020	12/6/2013	...
4	101028	92834	100028	10/8/2012	...
5	101036	31054	100036	10/8/2015	...

אחרי:

```

select *
from Loans
WHERE customerID IN (
    SELECT c.customerID
    FROM Customers c
    JOIN Branche b ON c.brancheID = b.brancheID
    WHERE b.brancheLocation = 'berlin'
);

```

	LOANID	LOANAMOUNT	CUSTOMERID	LOANDATE	
1	101004	15000	100004	11/24/2010	...
2	101012	15000	100012	3/24/2013	...
3	101020	15000	100020	12/6/2013	...
4	101028	15000	100028	10/8/2012	...
5	101036	15000	100036	10/8/2015	...

ב. עדכון התפקיד של פקיד להיות פקיד בכיר

```

UPDATE Workers
SET workerRole = 'Senior Officer'
WHERE workerID IN (
    SELECT workerID
    FROM Workers
    WHERE workerRole = 'Officer'
    AND brancheID = 103001
);

```

לפני

```

select *
from workers
WHERE workerID IN (
    SELECT workerID
    FROM Workers
    WHERE workerRole = 'Officer'
    AND brancheID = 103001
);

COMMIT;

```

	WORKERID	WORKERNAME	WORKERROLE	BRANCHEID	TEAMID	WORKERAGE
1	105001	Adam	Officer	103001	104000	89
2	105401	Anne	Officer	103001	104100	18
3	105801	Anthony	Officer	103001	104200	46
4	106201	Anthony	Officer	103001	104300	24

אחרי

```

select *
from workers
WHERE workerID IN (
    SELECT workerID
    FROM Workers
    WHERE workerRole = 'Senior Officer'
    AND brancheID = 103001
);

COMMIT;

```

	WORKERID	WORKERNAME	WORKERROLE	BRANCHEID	TEAMID	WORKERAGE
1	105001	Adam	Senior Officer	103001	104000	89
2	105401	Anne	Senior Officer	103001	104100	18
3	105801	Anthony	Senior Officer	103001	104200	46
4	106201	Anthony	Senior Officer	103001	104300	24

4.אילוצים:

א. ערך ברירת מחדל עבור מיקום סניף להיות מודיעין

```
ALTER TABLE Branche  
MODIFY brancheLocation DEFAULT 'Modiin';
```

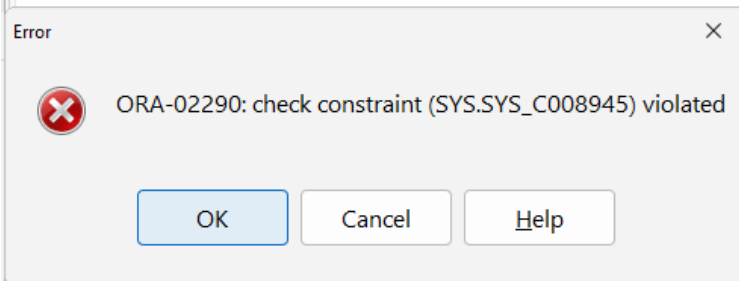
```
insert into SYS.BRANCHE (BRANCHEID, REPORTERID, OPENEDDATE)  
values (103403, 102718, to_date('24-07-2006', 'dd-mm-yyyy'));  
  
SELECT *  
FROM BRANCHE B  
WHERE B.BRANCHELOCATION = 'Modiin'
```

	BRANCHEID	BRANCHELOCATION	REPORTERID	OPENEDDATE
1	103402	Modiin	102718	7/24/2006
2	103403	Modiin	102718	7/24/2006

ב. הגדרת גיל עובד למינימום 18

```
ALTER TABLE Workers  
ADD CHECK (workerAge>=18);
```

```
insert into SYS.WORKERS (WORKERID, WORKERNAME, WORKERROLE, BRANCHEID, TEAMID,WORKERAGE)  
values (107001, 'Taryn', 'Manager', 103401, 104996,17);
```



ג. הגדרת שדה workday אצל teams להיות שדה חובה

```
ALTER TABLE Teams  
MODIFY workDay VARCHAR2(10) NOT NULL;
```

```
insert into SYS.TEAMS (TEAMID)  
values (104996);
```

Error



ORA-01400: cannot insert NULL into ("SYS"."TEAMS"."WORKDAY")

OK

Cancel

Help