

מגישות : נועה לנדמן 213877160

אביגיל טננבאום 213865520

תוכן עינים:

עמוד 2 : פרוט והצגת טבלאות הפרוייקט

עמוד 5 : הכנסת נתונים

עמוד 10 : שאילתות

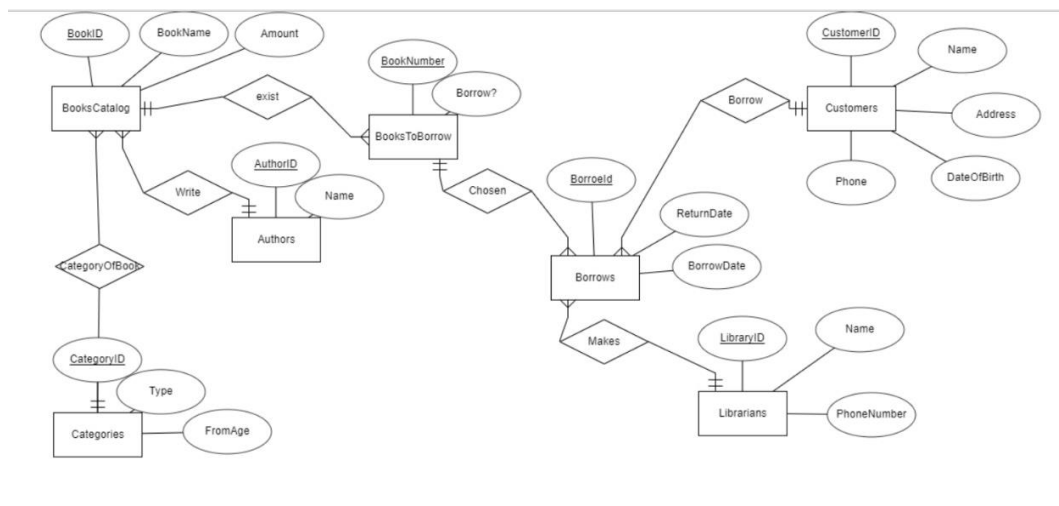
עמוד 21: פונקציות ופורצדורות

עמוד 27: שלב 4 - אינטגרציה

תרגיל 1 – מיניפ בסנת:

ספריה:

:ERD



פירוט:

Customers: רשימת מנויים של הספריה.

לכל מנוי יש תעודת זהות בתור מפתח ראשי, שם המנוי, כתובת, תאריך לידה, מספר פלאפון.

Librarians: רשימת הספרנים שעובדים בספריה.

לכל ספרן יש את תעודת הזהות שלו בתור מפתח ראשי, שם, ומספר פלאפון.

BooksToBorrow: רשימת הספרים שניתנים להשאלה בספריה.

לכל ספר יש את מספר הספר בתור מפתח ראשי(כאשר יש כמה עותקים מאותו ספר, כל עותק יקבל מספר אחר) וכן האם הספר מושאל (0) או זמין (1)

Borrows: רשימת השאלות שהתבצעו בספריה.

לכל השאלה יש מספר השאלה בתור מפתח ראשי, תאריך השאלה ותאריך החזרה של הספר (שבועיים לאחר תאריך ההשאלה).

כמו כן יש קשר של יחיד לרבים לישויות הבאות: רשימת מנויים, ספרנים, ספרים להשאלה כך שלכל השאלה יש בדיוק לקוח אחד, ספרן אחד וספר אחד.

כמות הספרים שניתן לשאול אינה מוגבלת אך כל ספר ישמר בתור השאלה נפרדת.

BooksCatalog: קטלוג הספרים של הספרייה.

עבור כל ספר ישמר מזהה הספר בתור מפתח ראשי, שם הספר, וכמות העותקים שלו.

הקשר exist: קשר יחיד לרבים כך שכל ספר מרשימת הספרים שניתנים להשאלה מופיע בדיוק פעם אחת בקטלוג הספרים.

Authors: רשימת הסופרים.

לכל סופר יש תעודת זהות בתור מפתח ראשי ושם.

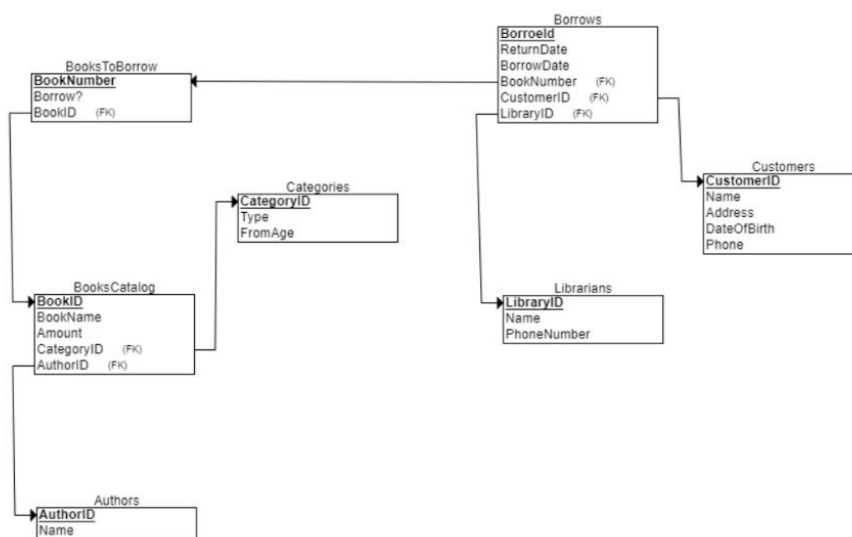
הקשר write: קשר של יחיד לרבים כך שלכל ספר יש סופר אחד בדיוק.

Categories: רשימת קטגוריות של ספרים.

לכל קטגוריה יש מספר מזהה קטגוריה בתור מפתח ראשי, סוג הקטגוריה ומאיזה גיל ניתן להשאיל ספרים ששייכים לקטגוריה זו.

הקשר categoryOfBooks: קשר של יחיד לרבים כך שכל ספר בקטלוג שייך לקטגוריה אחת.

תרשים DSD



CreateTable

```
CREATE TABLE Customers
(
    CustomerID NUMERIC(5) NOT NULL,
    cName VARCHAR(15),
    Address VARCHAR(15),
    DateOfBirth DATE NOT NULL,
    Phone VARCHAR(10),
    PRIMARY KEY (CustomerID)
);
```

```
CREATE TABLE Categories
(
    CategoryID NUMERIC(5) NOT NULL,
    cType VARCHAR(15),
```

```

    FromAge INT NOT NULL,
    PRIMARY KEY (CategoryID)
);

CREATE TABLE Librarians
(
    LibraryID NUMERIC(5) NOT NULL,
    lName VARCHAR(15),
    PhoneNumber VARCHAR(10),
    PRIMARY KEY (LibraryID)
);

CREATE TABLE Authors
(
    AuthorID NUMERIC(5) NOT NULL,
    aName VARCHAR(15),
    PRIMARY KEY (AuthorID)
);

CREATE TABLE BooksCatalog
(
    BookID NUMERIC(5) NOT NULL,
    BookName VARCHAR(15),
    Amount INT,
    CategoryID NUMERIC(5),
    AuthorID NUMERIC(5),
    PRIMARY KEY (BookID),
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID),
    FOREIGN KEY (AuthorID) REFERENCES Authors(AuthorID)
);

CREATE TABLE BooksToBorrow
(
    BookNumber NUMERIC(5) NOT NULL,
    isBorrow CHAR(1),
    BookID NUMERIC(5) NOT NULL,
    PRIMARY KEY (BookNumber),
    FOREIGN KEY (BookID) REFERENCES BooksCatalog(BookID)
);

CREATE TABLE Borrows
(
    BorroeId NUMERIC(5) NOT NULL,
    ReturnDate DATE,
    BorrowDate DATE,
    BookNumber NUMERIC(5) NOT NULL,
    CustomerID NUMERIC(5) NOT NULL,
    LibraryID NUMERIC(5) NOT NULL,
    PRIMARY KEY (BorroeId),
    FOREIGN KEY (BookNumber) REFERENCES BooksToBorrow(BookNumber),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (LibraryID) REFERENCES Librarians(LibraryID)
);

```

:desc תורת

```
Command Window - New
Dialog Editor
Connected as SYSTEM@XE

SQL> DESC Customers
Name      Type      Nullable Default Comments
-----
CUSTOMERID NUMBER(5)
CNAME     VARCHAR2(15) Y
ADDRESS   VARCHAR2(15) Y
DATEOFBIRTH DATE
PHONE     VARCHAR2(10) Y

SQL> DESC Categories
Name      Type      Nullable Default Comments
-----
CATEGORYID NUMBER(5)
CTYPE     VARCHAR2(15) Y
FROMAGE   INTEGER

SQL> DESC Librarians
Name      Type      Nullable Default Comments
-----
LIBRARYID NUMBER(5)
LNAME     VARCHAR2(15) Y
PHONENUMBER VARCHAR2(10) Y

SQL>
```

```
Command Window - New
Dialog Editor
LNAME     VARCHAR2(15) Y
PHONENUMBER VARCHAR2(10) Y

SQL> DESC BooksCatalog
Name      Type      Nullable Default Comments
-----
BOOKID    NUMBER(5)
BOOKNAME  VARCHAR2(15) Y
AMOUNT    INTEGER Y
CATEGORYID NUMBER(5) Y
AUTHORID  NUMBER(5) Y

SQL> DESC BooksToBorrow
Name      Type      Nullable Default Comments
-----
BOOKNUMBER NUMBER(5)
ISBORROW  CHAR(1) Y
BOOKID    NUMBER(5)

SQL> Desc Authors
Name      Type      Nullable Default Comments
-----
AUTHORID  NUMBER(5)
NAME      VARCHAR2(15) Y

SQL>
```

```
Command Window - New
Dialog Editor
Connected to Oracle Database 11g Express Edition Release 11.2.0.2.0
Connected as Avigail@XE

SQL> DESC Borrows
Name      Type      Nullable Default Comments
-----
BORROEID  NUMBER(5)
RETURNDATE DATE Y
BORROWDATE DATE Y
BOOKNUMBER NUMBER(5)
CUSTOMERID NUMBER(5)
LIBRARYID NUMBER(5)

SQL>
```

צילום מסך של הכנסת נתונים ע"י excel:

ODBC Importer

Data from ODBC | Data to Oracle

Connection

User / System DSN: Excel Files

User Name:

Password:

Connect Disconnect

Table / Query

☒ Import Table ☐ Import Query Result

גיליון Excel: 'גיליון1' 'Excel.xlsx'

View Data

Result Preview

CustomerID	cName	Address	DateOfBirth	Phone
10.0	John Doe	123 Main St	1990-05-15 00:00:00	555-1234
11.0	Alice Smith	456 Oak Ave	1985-08-21 00:00:00	555-5678
12.0	Bob Johnson	789 Elm St	1978-12-10 00:00:00	555-9876
13.0	Emily Brown	321 Pine Rd	1995-03-28 00:00:00	555-4321
14.0	Sarah Lee	654 Cedar St	1982-09-17 00:00:00	555-8765
15.0	Michael Wong	987 Birch Ln	1973-06-03 00:00:00	555-3456
16.0	Jessica Chen	741 Maple Dr	1988-11-29 00:00:00	555-6543
17.0	David Kim	852 Walnut St	1992-04-12 00:00:00	555-7890
18.0	Laura Davis	369 Cherry Rd	1980-07-24 00:00:00	555-2109
19.0	Kevin White	159 Oakwood Ave	1975-01-05 00:00:00	555-9630

Import Import to Script Close SYSTEM@XE Help

SQL Window - Edit data of table CUSTOMERS@XE

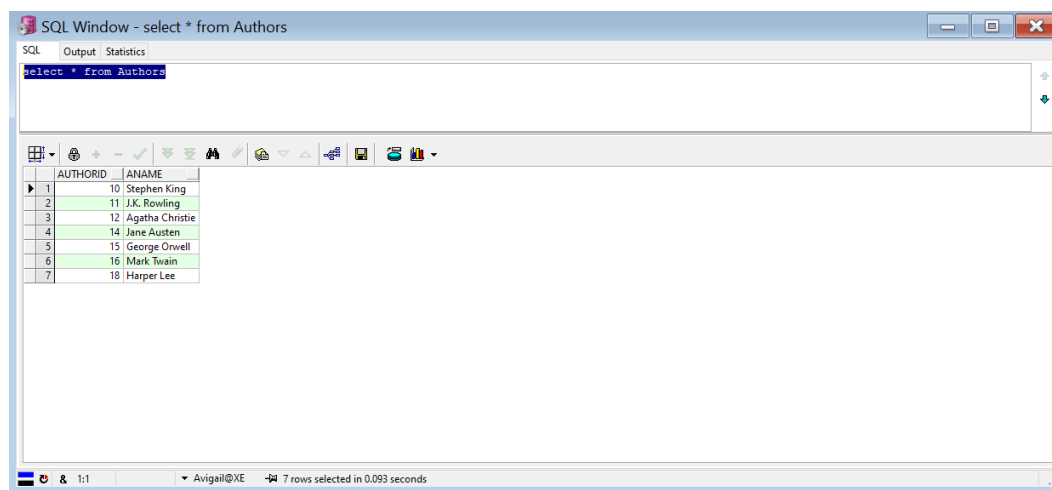
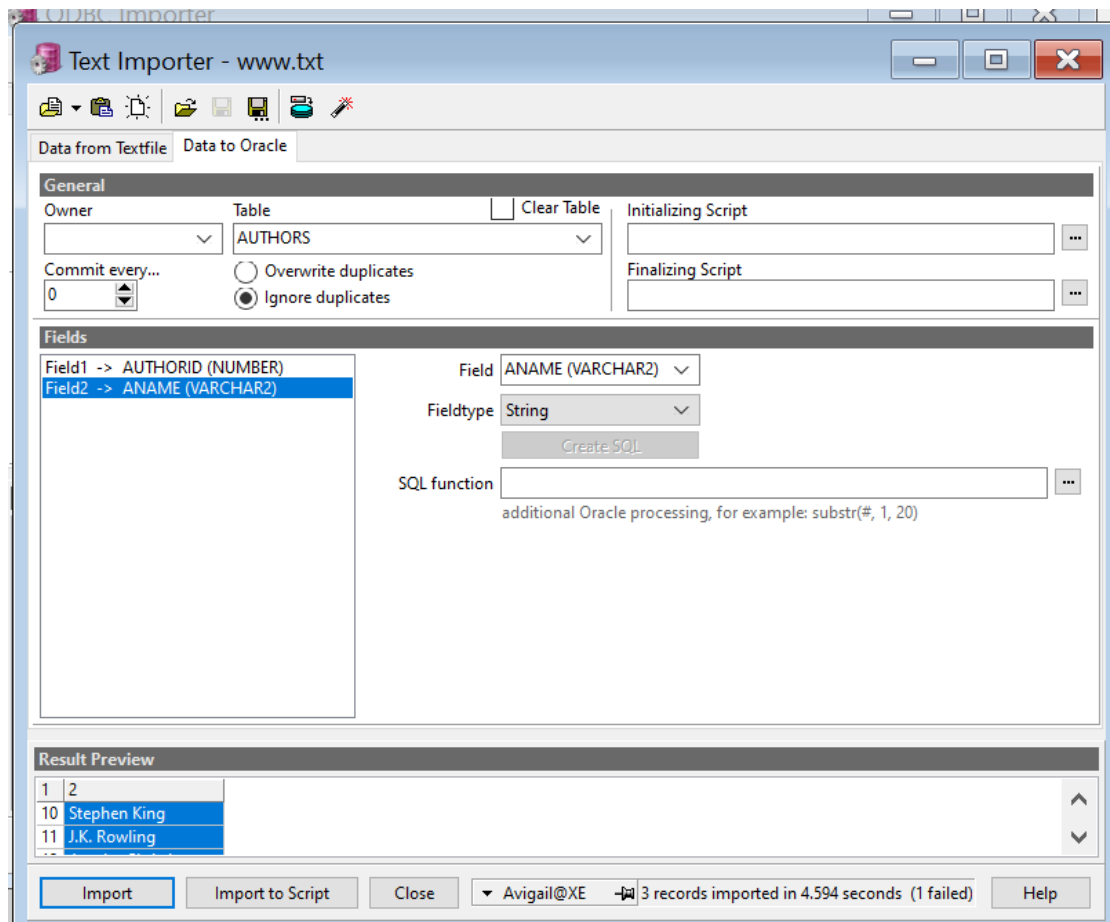
SQL Output Statistics

```
select t.*, t.rowid from CUSTOMERS t
```

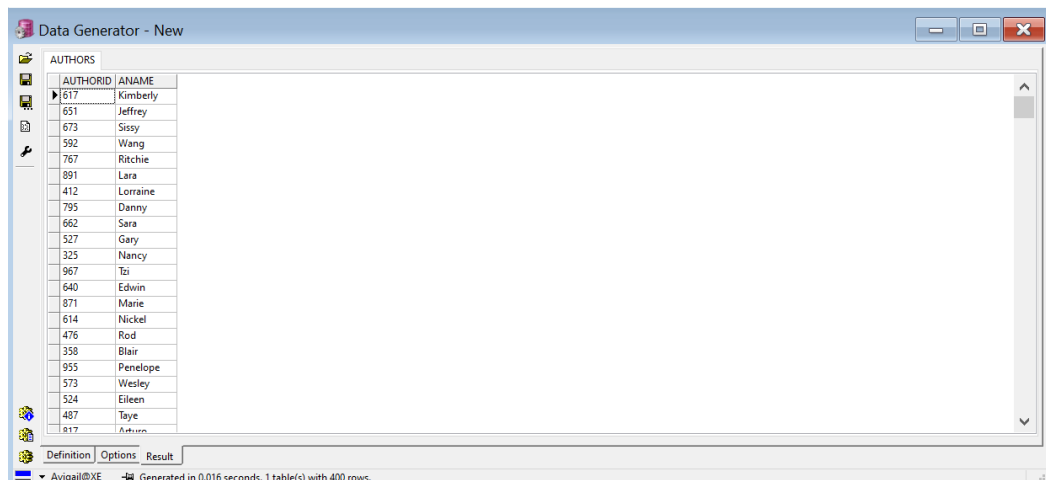
	CUSTOMERID	CNAME	ADDRESS	DATEOFBIRTH	PHONE	ROWID
1	123	Avigail	ruzin	07/12/2003	0583258717	AAAEspAABAAAK-pAAA
2	156	Noa	bublik	07/12/2003	0583546678	AAAEspAABAAAK-pAAB
3	334	Chaya	yyy	11/09/2001	0523987227	AAAEspAABAAAK-pAAC
4	1	John Doe	123 Elm St	07/12/2003	5551234567	AAAEspAABAAAK-pAAD
5	2	Jane Smith	456 Oak St	07/12/2003	5559876543	AAAEspAABAAAK-pAAE
6	10	John Doe	123 Main St	15/05/1990	555-1234	AAAEspAABAAAK-pAAF
7	11	Alice Smith	456 Oak Ave	21/08/1985	555-5678	AAAEspAABAAAK-pAAG
8	12	Bob Johnson	789 Elm St	10/12/1978	555-9876	AAAEspAABAAAK-pAAH
9	13	Emily Brown	321 Pine Rd	28/03/1995	555-4321	AAAEspAABAAAK-pAAI
10	14	Sarah Lee	654 Cedar St	17/09/1982	555-8765	AAAEspAABAAAK-pAAJ
11	15	Michael Wong	987 Birch Ln	03/06/1973	555-3456	AAAEspAABAAAK-pAAK
12	16	Jessica Chen	741 Maple Dr	29/11/1988	555-6543	AAAEspAABAAAK-pAAL
13	17	David Kim	852 Walnut St	12/04/1992	555-7890	AAAEspAABAAAK-pAAM
14	18	Laura Davis	369 Cherry Rd	24/07/1980	555-2109	AAAEspAABAAAK-pAAN
15	19	Kevin White	159 Oakwood Ave	05/01/1975	555-9630	AAAEspAABAAAK-pAAO

SYSTEM@XE 15 rows selected in 0.234 seconds

הכנסת נתונים בקובץ txt:

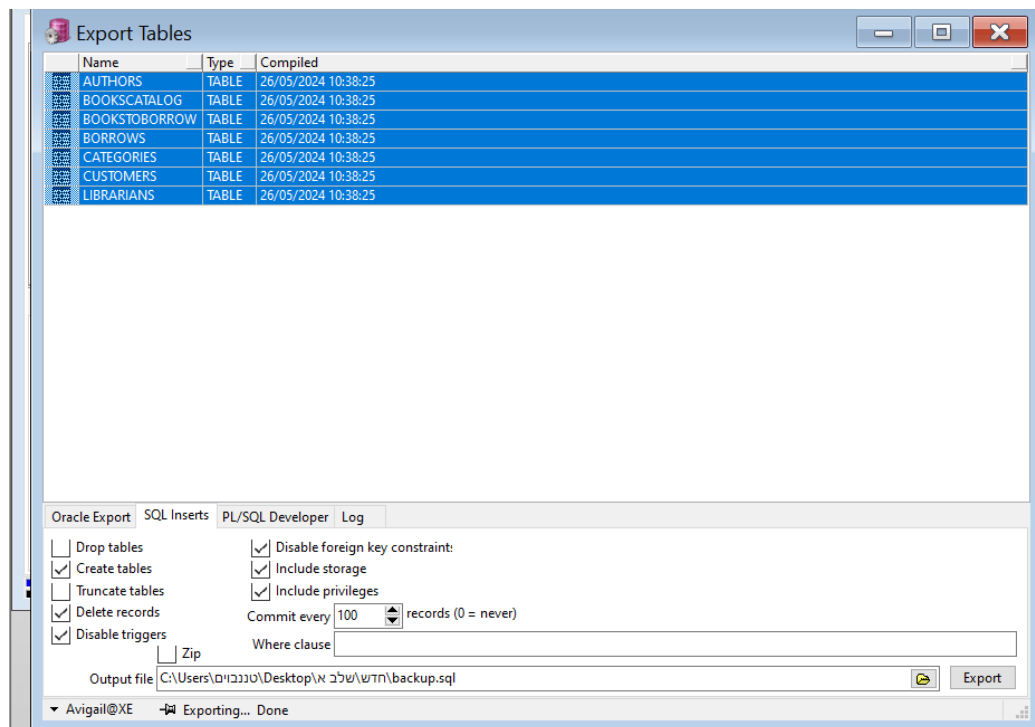


הכנסת נתונים באמצעות DataGenareator:



הערה: עשינו מדגם של הכנסה בכל שיטה ולבסוף הכנסנו לכל טבלה 400 שורות באמצעות ה data generator

גיבוי:

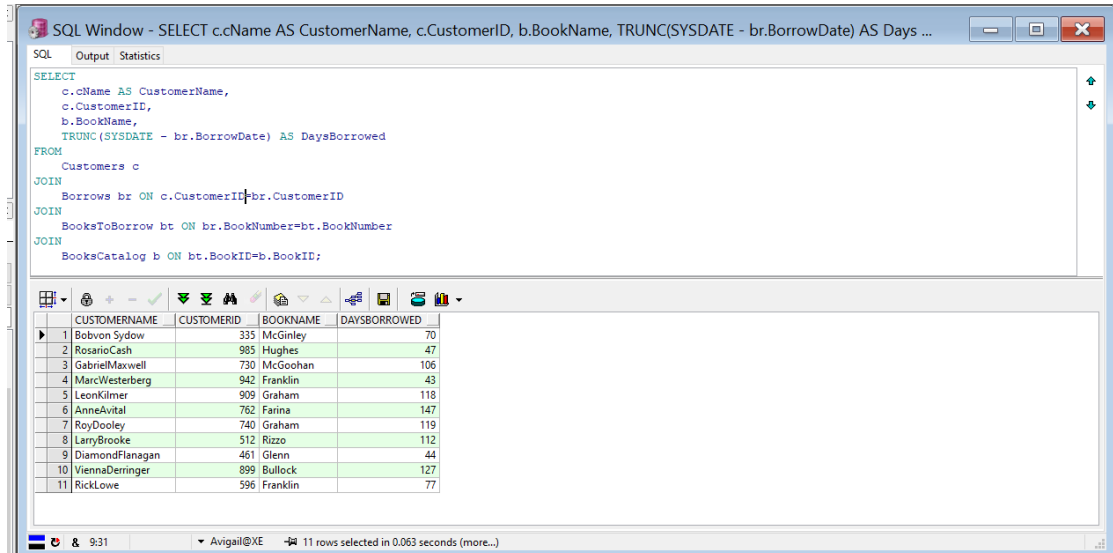




שלב 2:

שאלות:

שאלת select 1:



SQL Window - SELECT c.cName AS CustomerName, c.CustomerID, b.BookName, TRUNC(SYSDATE - br.BorrowDate) AS Days ...

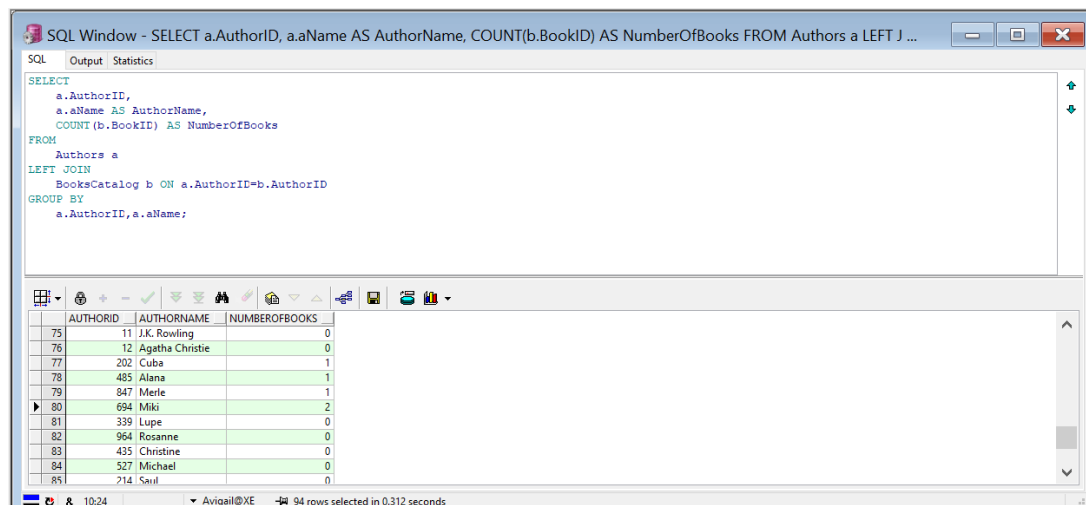
```
SELECT
  c.cName AS CustomerName,
  c.CustomerID,
  b.BookName,
  TRUNC(SYSDATE - br.BorrowDate) AS DaysBorrowed
FROM
  Customers c
JOIN
  Borrows br ON c.CustomerID=br.CustomerID
JOIN
  BooksToBorrow bt ON br.BookNumber=bt.BookNumber
JOIN
  BooksCatalog b ON bt.BookID=b.BookID;
```

	CUSTOMERNAME	CUSTOMERID	BOOKNAME	DAYSBORROWED
1	Bobvon Sydow	335	McGinley	70
2	RosarioCash	985	Hughes	47
3	GabrielMaxwell	730	McGoochan	106
4	MarcWesterberg	942	Franklin	43
5	LeonKilmer	909	Graham	118
6	AnneAvital	762	Farina	147
7	RoyDooley	740	Graham	119
8	LarryBrooke	512	Rizzo	112
9	DiamondFlanagan	461	Glenn	44
10	ViennaDerringer	899	Bullock	127
11	RickLowe	596	Franklin	77

Avigail@XE 11 rows selected in 0.063 seconds (more...)

השאלתה מחזירה לכל לקוח את שמו, תעודת הזהות שלו, שם הספר שהוא שאל וכמה ימים הספר נמצא אצלו.

שאלת select 2:



SQL Window - SELECT a.AuthorID, a.aName AS AuthorName, COUNT(b.BookID) AS NumberOfBooks FROM Authors a LEFT J ...

```
SELECT
  a.AuthorID,
  a.aName AS AuthorName,
  COUNT(b.BookID) AS NumberOfBooks
FROM
  Authors a
LEFT JOIN
  BooksCatalog b ON a.AuthorID=b.AuthorID
GROUP BY
  a.AuthorID, a.aName;
```

	AUTHORID	AUTHORNAME	NUMBEROFBOOKS
75	11	J.K. Rowling	0
76	12	Agatha Christie	0
77	202	Cuba	1
78	485	Alana	1
79	847	Merle	1
80	694	Miki	2
81	339	Lupe	0
82	964	Rosanne	0
83	435	Christine	0
84	527	Michael	0
85	714	Saul	0

Avigail@XE 94 rows selected in 0.312 seconds

השאלתה מחזירה לכל סופר את שמו, תעודת הזהות שלו ואת מספר הספרים שהוא כתב.

שאלת select 3:

SQL Window - SELECT l.LibraryID, l.lName AS LibrarianName, l.PhoneNumber, CASE WHEN COUNT(br.Borroeld) > 2 TH ...

```

SELECT
  l.LibraryID,
  l.lName AS LibrarianName,
  l.PhoneNumber,
  CASE
    WHEN COUNT(br.Borroeld) > 2 THEN 'Yes'
    ELSE 'No'
  END AS IsEligibleForBonus
FROM
  Librarians l
LEFT JOIN
  Borrows br ON l.LibraryID=br.LibraryID
GROUP BY
  l.LibraryID, l.lName, l.PhoneNumber;

```

	LIBRARYID	LIBRARIANNAME	PHONENUMBER	ISELIGIBLEFORBONUS
1	599	MartinMcDonnell	1604542899	No
2	284	JasonFiennes	544577504	Yes
3	711	DanWinwood	2041222785	No
4	817	RickieBello	3374694696	No
5	246	DenisBergen	2399781773	No
6	454	KimMantegna	1960550408	No
7	500	SonnyHutch	4219606693	No
8	493	JulianaO'Conno	990392438	No
9	545	JoseDavis	514570588	No
10	997	NenehPenders	4163013106	No

Avigail@XE 14:26 10 rows selected in 0.094 seconds (more...)

שאלתה מחזירה את תעודת הזהות של הספרן , שמו , מספר הטלפון שלו והאם זכאי לבונוס או לא.

ספרן יהיה זכאי לבונוס במקרה שביצע יותר משתי השאלות.

שאלתת 4 select :

SQL Window - query4.sql

```

WITH BorrowCounts AS (
  SELECT
    c.CustomerID,
    c.cName,
    COUNT(b.Borroeld) AS BorrowCount
  FROM
    Customers c
  JOIN
    Borrows b ON c.CustomerID = b.CustomerID
  GROUP BY
    c.CustomerID, c.cName
),
MaxBorrowCount AS (
  SELECT
    MAX(BorrowCount) AS MaxCount
  FROM
    BorrowCounts
)
SELECT
  bc.CustomerID,
  bc.cName,
  bc.BorrowCount
FROM
  BorrowCounts bc
JOIN
  MaxBorrowCount mbc ON bc.BorrowCount = mbc.MaxCount;

```

	CUSTOMERID	CNAME	BORROWCOUNT
1	740	RoyDoolak	5
2	800	EugeneAdkina	5
3	726	TommyLeviA	5

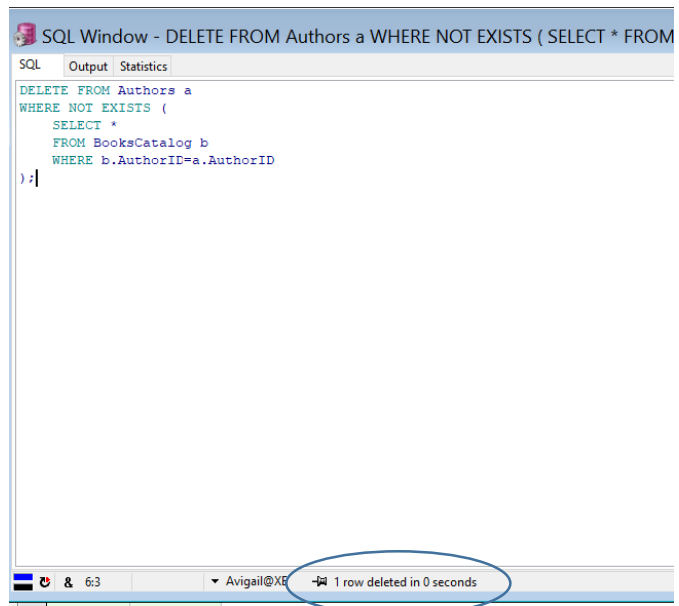
Avigail@XE 26:57 3 rows selected in 0.125 seconds

השאלתה מחזירה את שם ומספר הזהות של הלקוח ששאל הכי הרבה ספרים וכן כמה ספרים הוא שאל.

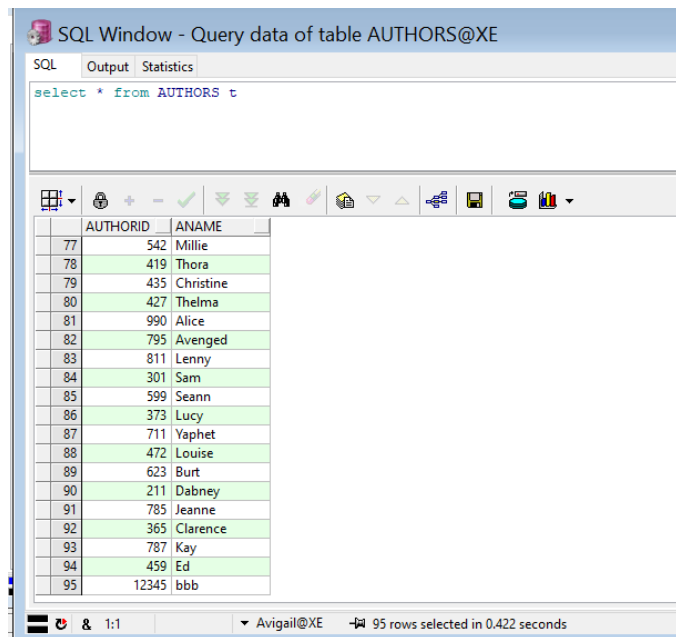
שאלתות delete:

שאלתה 1:

השאלתה מוחקת את כל הסופרים שלא כתבו אף ספר.



לפני: (הוספנו את השורה האחרונה כך שלסופר אין שום ספר שהוא כתב)

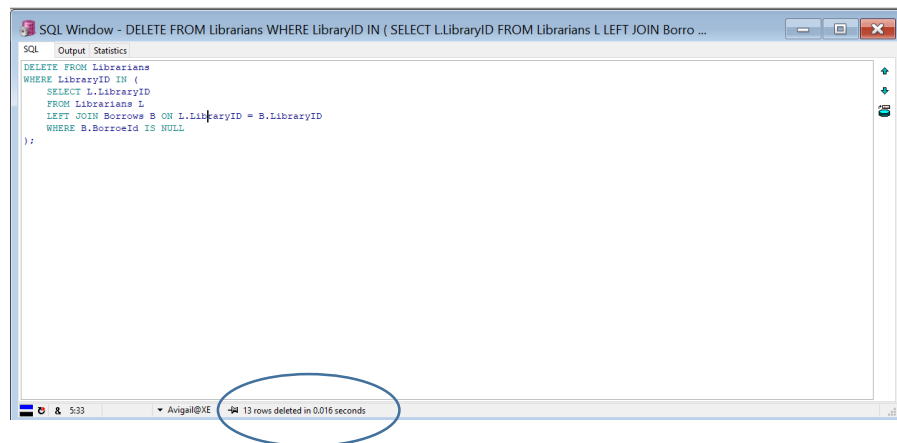


אחרי:

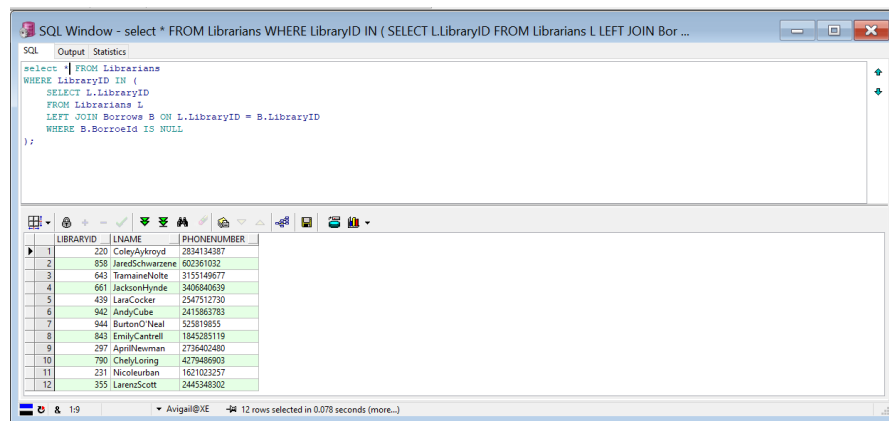


שאלתה 2:

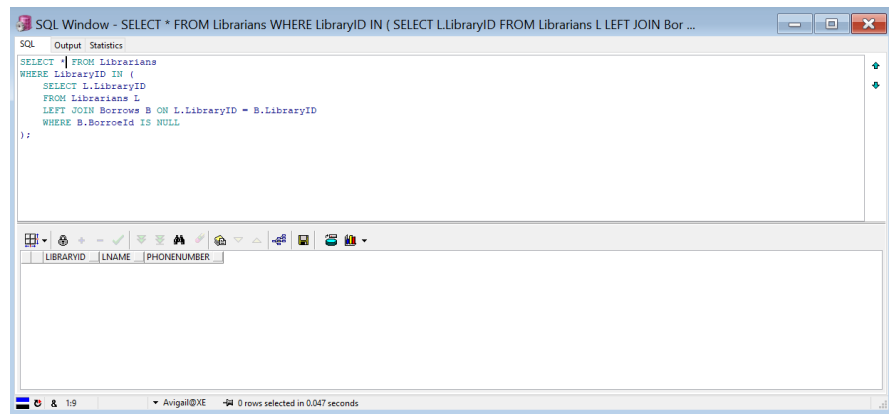
השאלתה מוחקת את כל הספרנים שלא ביצעו אף השאלה



לפני:



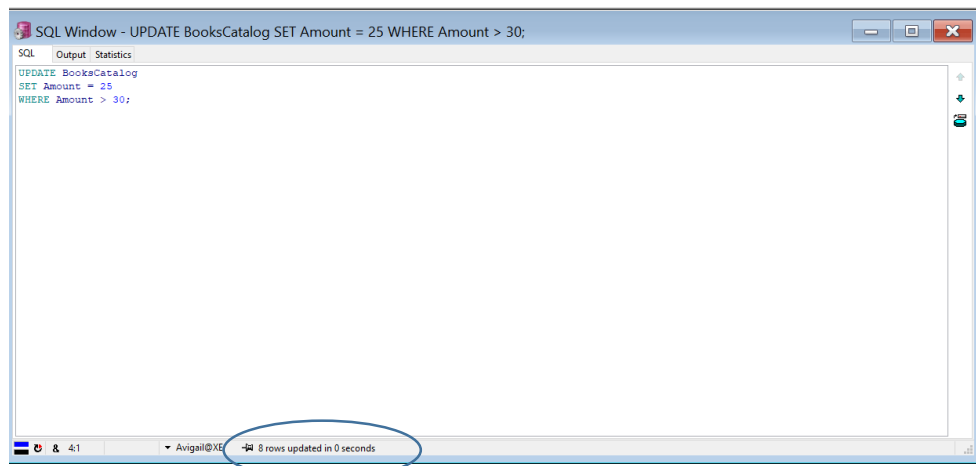
אחרי:



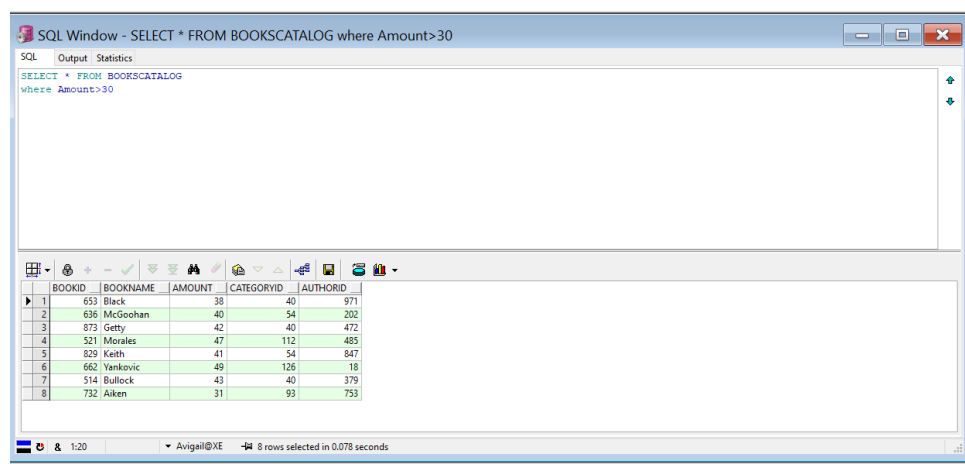
שאלות update:

שאלתה 1:

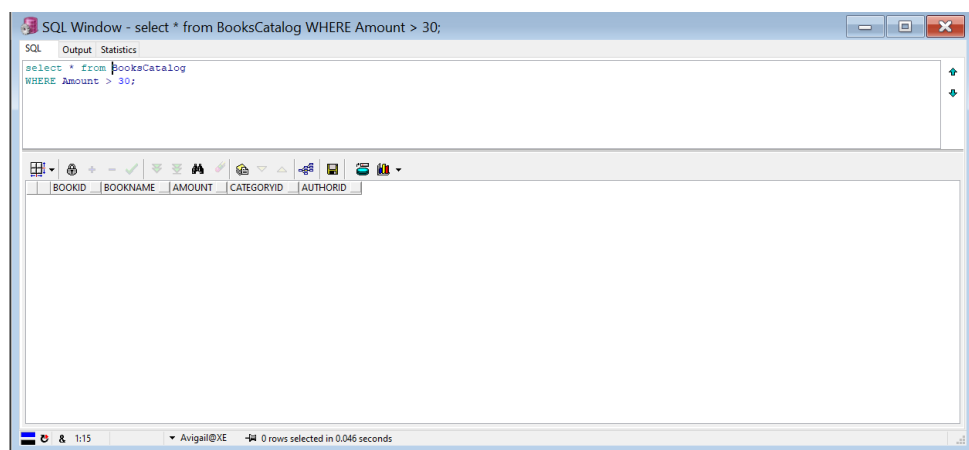
השאלתה מעדכנת את מלאי כל הספרים שהמלאי שלהם גדול מ30 להיות 25.



לפני:

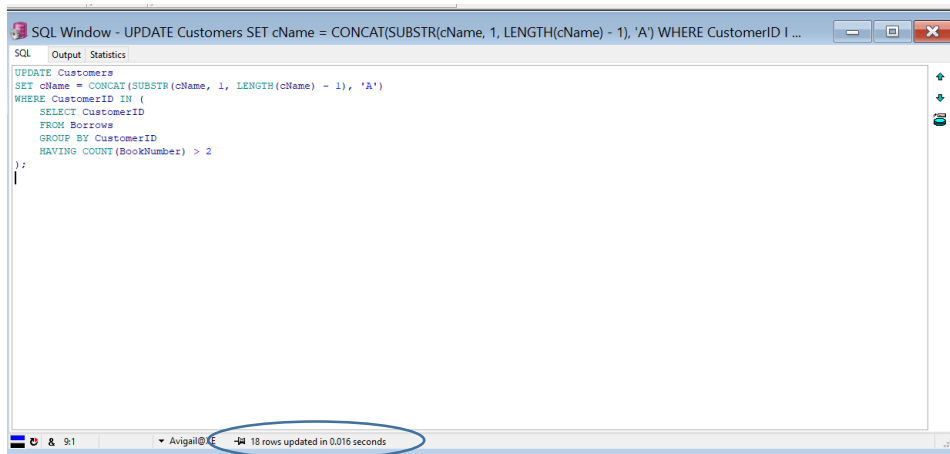


אחרי:

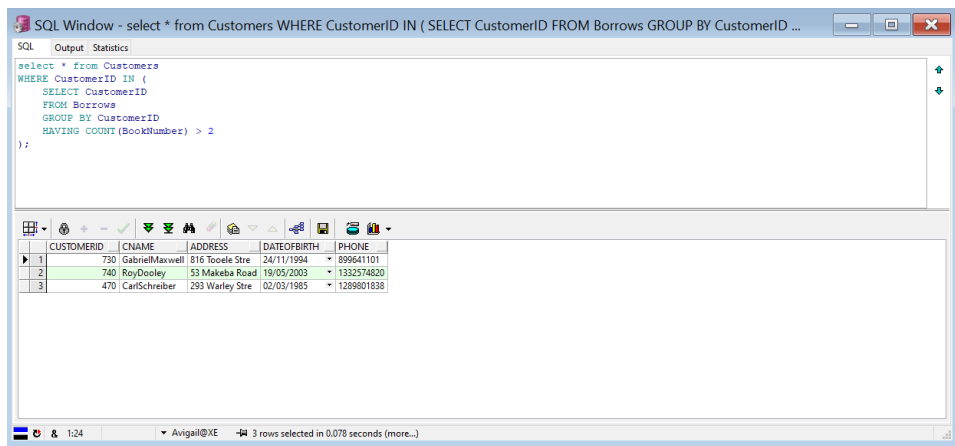


שאלתה 2:

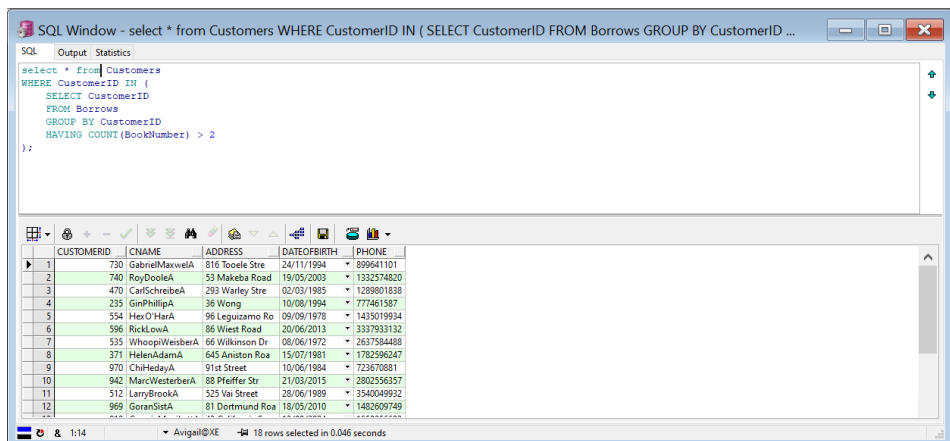
השאלתה מעדכנת את שמות כל הלקוחות ששאלו יותר משני ספרים להסתיים באות A במקום האות האחרונה שלהם.



לפני: (רואים כאן רק חלק)



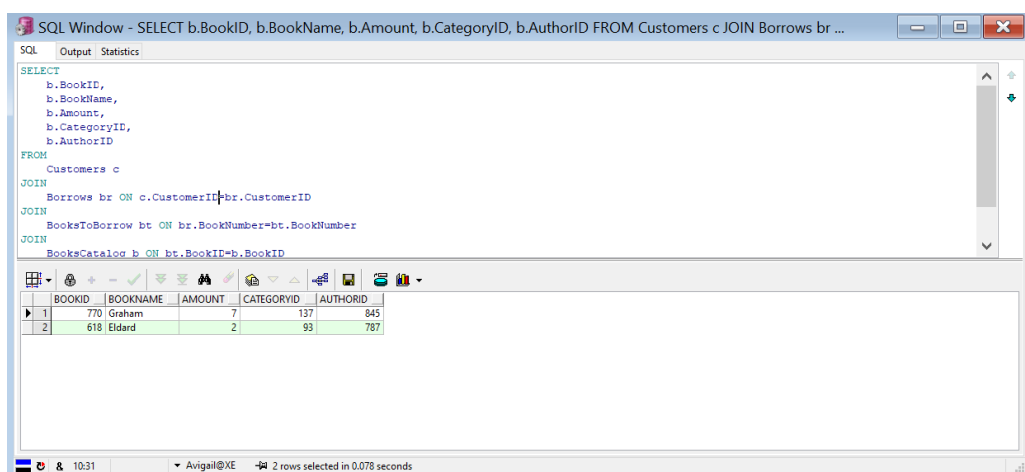
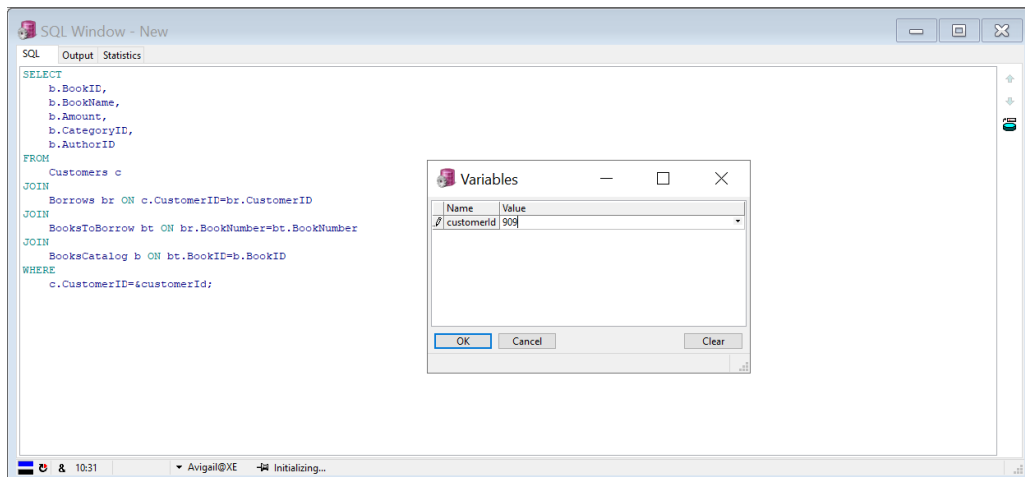
אחרי:



שאלות עם פרמטרים:

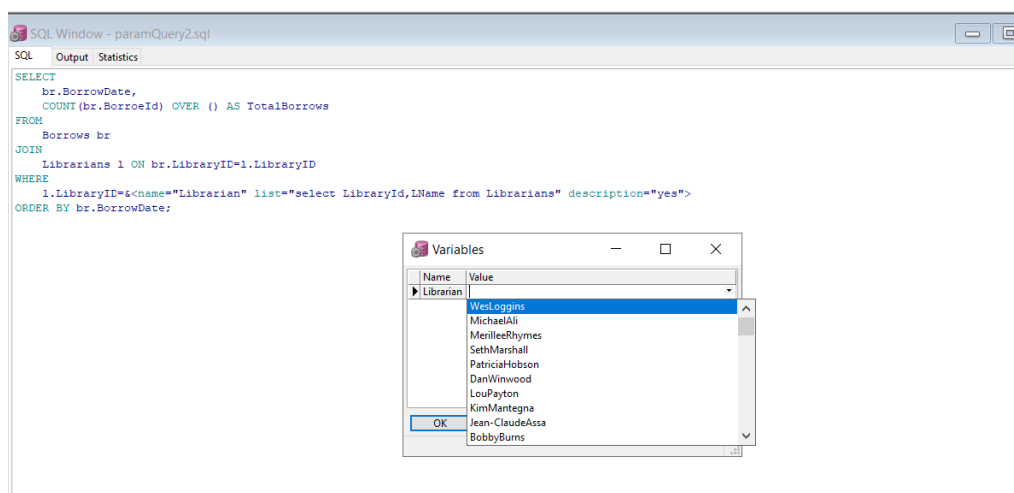
שאלתה 1:

השאלתה מקבלת כפרמטר תעודת זהות של לקוח ומחזירה את פרטי כל הספרים שהוא שאל.



שאלתה 2:

השאלתה מקבלת כפרמטר מספר זהות של ספרן(מתוך רשימה של שמות הספרנים) ומחזירה את מספר ההשאלות שהוא ביצע ואת התאריכים שלהם.



SQL Window - paramQuery2.sql

```

SELECT
  br.BorrowDate,
  COUNT(br.BorrowId) OVER () AS TotalBorrows
FROM
  Borrows br
JOIN
  Librarians l ON br.LibraryID=l.LibraryID
WHERE
  l.LibraryID=4<name="Librarian" list="select LibraryId, lName from Librarians" description="yes">
ORDER BY br.BorrowDate;

```

	BORROWDATE	TOTALBORROWS
1	01/02/2024	2
2	13/04/2024	2

שאלתה 3:

השאלתה מקבלת כפרמטר תאריך מסוים ומחזירה את שמות כל הספרים שהושאלו בתאריך זה ואת מספר הספרים שהושאלו בתאריך זה.

SQL Window - New

```

SELECT
  b.BookName,
  COUNT(*) AS BorrowCount
FROM
  Borrows br
JOIN
  BooksToBorrow bt ON br.BookNumber=bt.BookNumber
JOIN
  BooksCatalog b ON bt.BookID=b.BookID
WHERE
  br.BorrowDate=4<name="myDate" type="date" hint="please enter a date less than today">
GROUP BY b.BookName;

```

Variables

Name	Value
myDate	27/2/2024

please enter a date less than today

SQL Window - SELECT b.BookName, COUNT(*) AS BorrowCount FROM Borrows br JOIN BooksToBorrow bt ON br.BookNumber=bt.BookNumber JOIN BooksCatalog b ON bt.BookID=b.BookID WHERE br.BorrowDate=4<name="myDate" type="date" hint="please enter a date less than today"> GROUP BY b.BookName;

```

SELECT
  b.BookName,
  COUNT(*) AS BorrowCount
FROM
  Borrows br
JOIN
  BooksToBorrow bt ON br.BookNumber=bt.BookNumber
JOIN
  BooksCatalog b ON bt.BookID=b.BookID
WHERE
  br.BorrowDate=4<name="myDate" type="date" hint="please enter a date less than today">
GROUP BY b.BookName;

```

	BOOKNAME	BORROWCOUNT
1	Farina	1

1 row selected in 0.063 seconds

שאלתה 4:

השאלתה מקבלת כפרמטר שם של ספר שנבחר מתוך הרשימה של ספרים הקיימים ומחזירה את רשימת כל הלקוחות ששאלו את הספר הזה בחודש האחרון.

The first screenshot shows a SQL query in a window titled "SQL Window - paramQuery4.sql". The query is as follows:

```
SELECT
  c.CustomerID,
  c.cName,
  br.BorrowDate
FROM
  Customers c
JOIN
  Borrows br ON c.CustomerID=br.CustomerID
JOIN
  BooksToBorrow bt ON br.BookNumber=bt.BookNumber
JOIN
  BooksCatalog b ON bt.BookID=b.BookID
WHERE
  b.BookName=(:cName=:bookName) list="select BookName from BooksCatalog" type="string">
  AND br.BorrowDate>=ADD_MONTHS(SYSDATE,-1)
ORDER BY br.BorrowDate DESC;
```

The second screenshot shows the same SQL query window with a "Variables" dialog box open. The dialog has a "Name" column with the value "bookName" and a "Value" column with a list of book names: Page, Brooks, Black, Hauer, Hayek, Cane, D'Franco, McCookhan, Getty, and Morales. The "bookName" variable is selected.

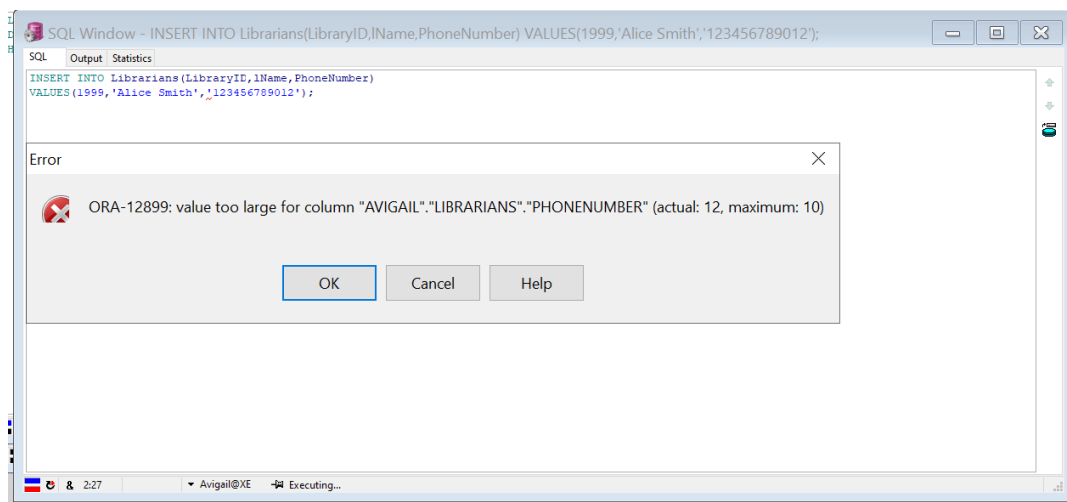
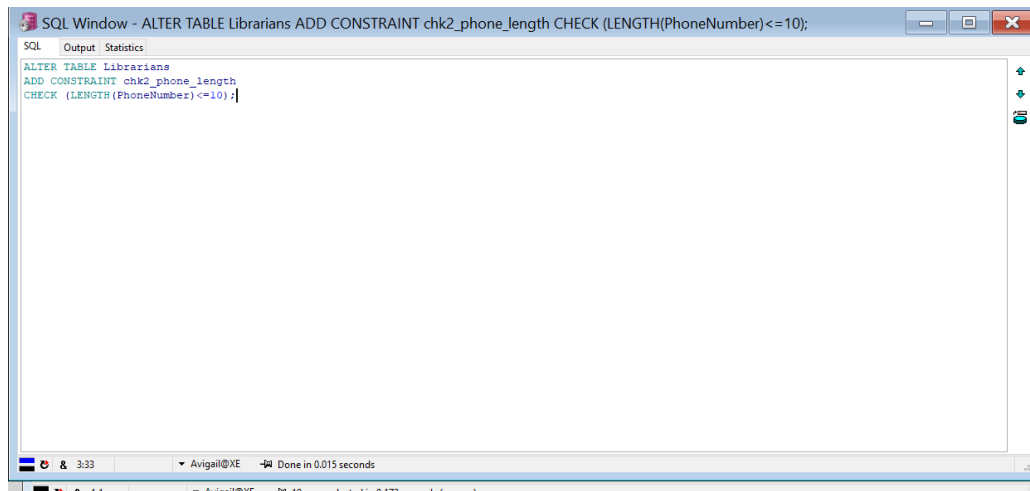
The third screenshot shows the same SQL query window with the query results displayed in a table. The table has three columns: CUSTOMERID, CNAME, and BORROWDATE. The results are as follows:

CUSTOMERID	CNAME	BORROWDATE
819	ConneMeniketa	20/05/2024
238	JimmyRamirez	19/05/2024
595	NemuesBall	09/05/2024

אילוצים:

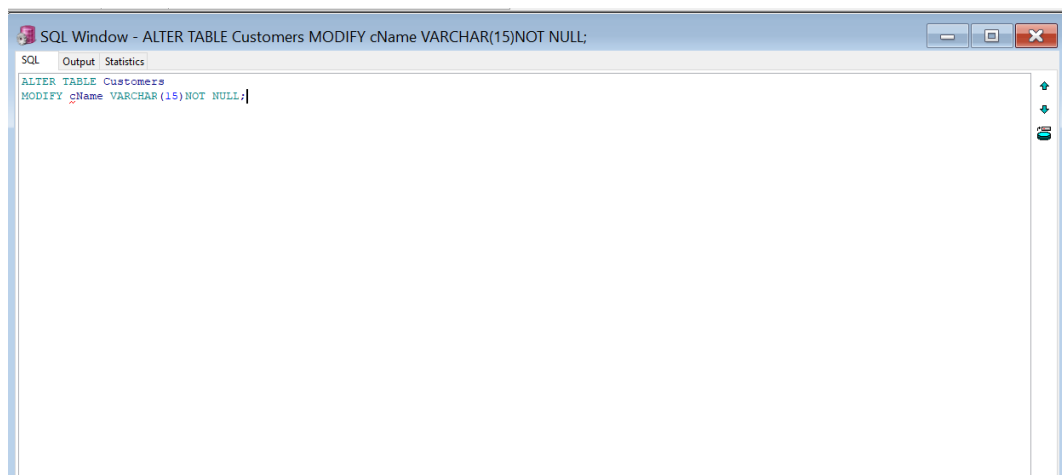
אילוץ 1:

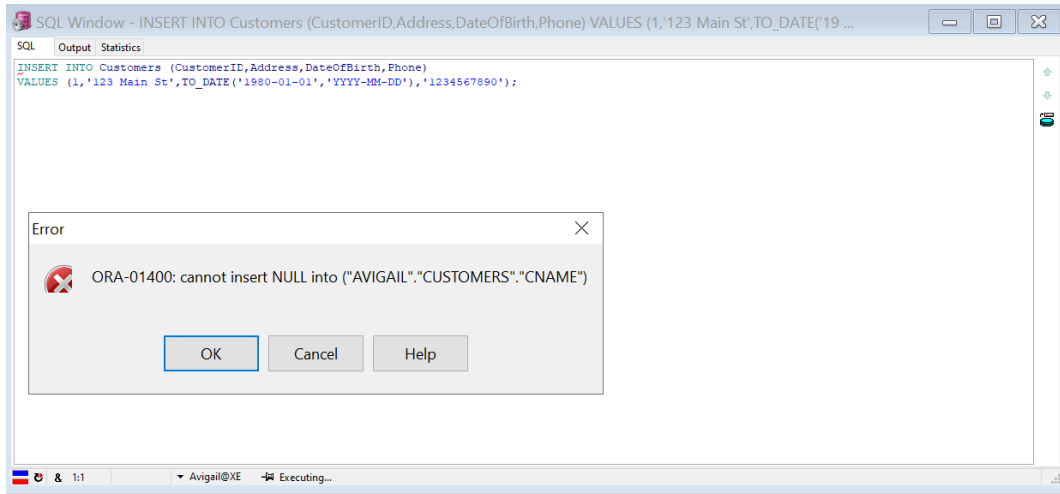
הגבלת אורך מספר הטלפון להיות עד 10 תווים.



אילוח 2:

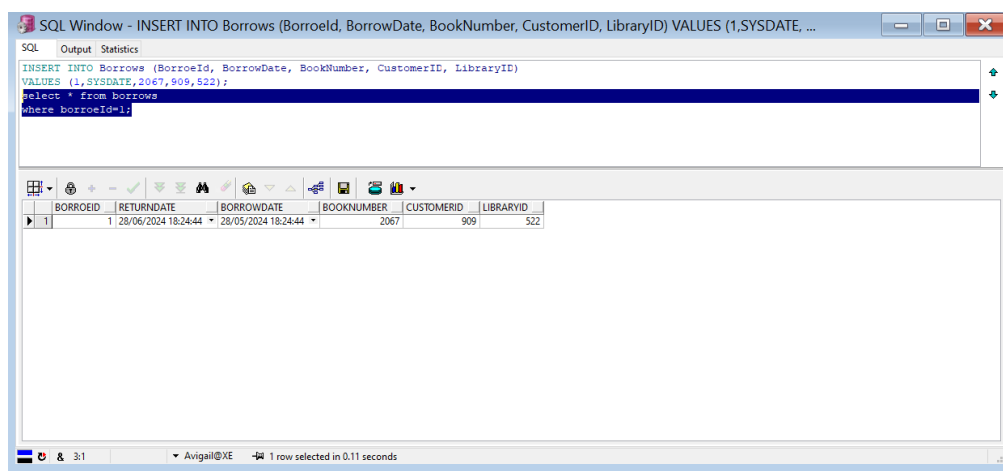
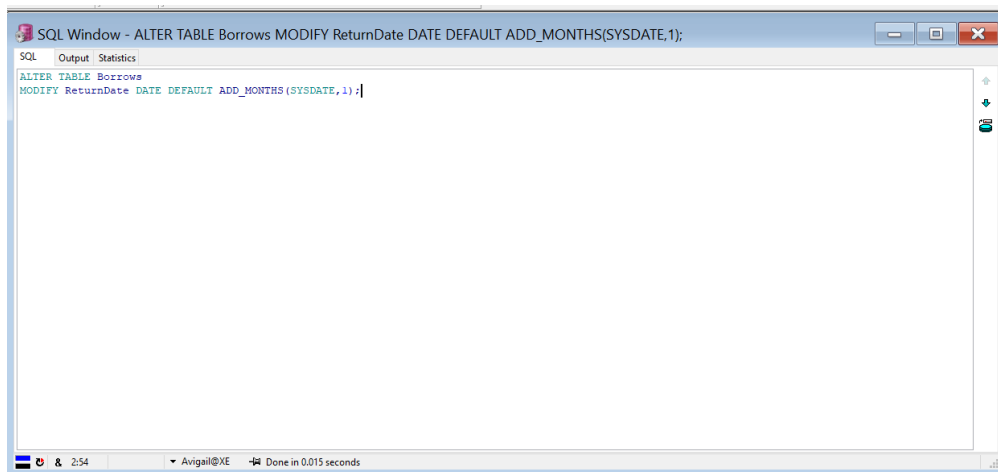
שדה שם הלקוח לא יכול להיות null.





אילון 3:

הגדרת ברירת המחדל של תאריך החזרה להיות חודש מהתאריך הנוכחי.

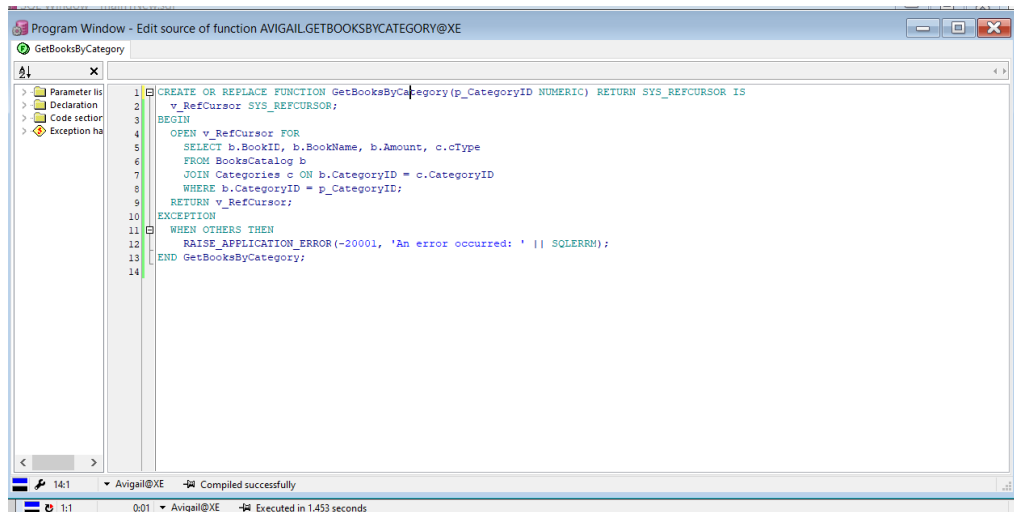


שלב ג – פונקציות ופרוצדורות:

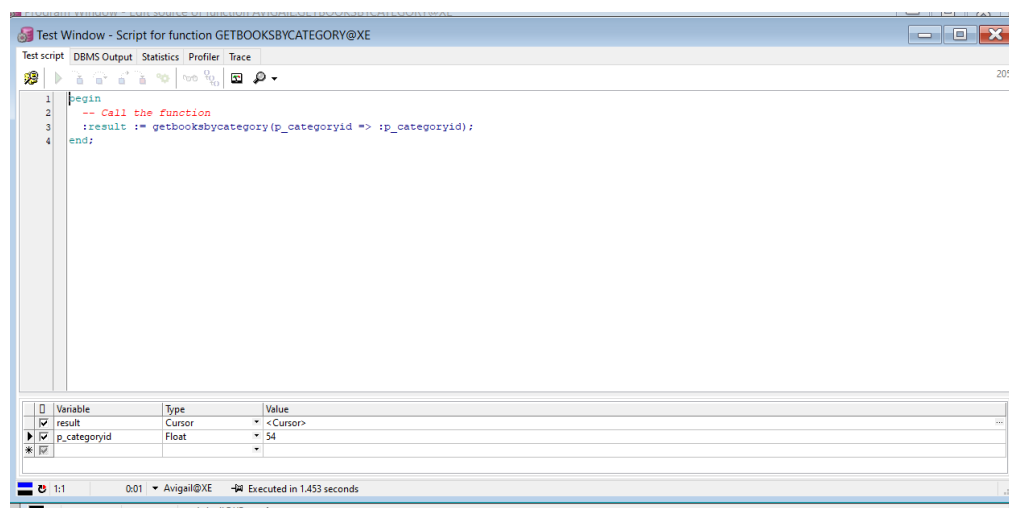
תוכנית 1:

הפונקציה:

הפונקציה מחזירה refcursor עם רשימת כל הספרים השייכים לקטגוריה מסויימת



```
1 CREATE OR REPLACE FUNCTION GetBooksByCategory(p_CategoryID NUMERIC) RETURN SYS_REFCURSOR IS
2   v_RefCursor SYS_REFCURSOR;
3 BEGIN
4   OPEN v_RefCursor FOR
5     SELECT b.BookID, b.BookName, b.Amount, c.cType
6     FROM BooksCatalog b
7     JOIN Categories c ON b.CategoryID = c.CategoryID
8     WHERE b.CategoryID = p_CategoryID;
9   RETURN v_RefCursor;
10 EXCEPTION
11 WHEN OTHERS THEN
12   RAISE_APPLICATION_ERROR(-20001, 'An error occurred: ' || SQLERRM);
13 END GetBooksByCategory;
```



```
1 begin
2   -- Call the function
3   :result := getbooksbycategory(p_categoryid => :p_categoryid);
4 end;
```

Variable	Type	Value
result	Cursor	<Cursor>
p_categoryid	Float	54

הפרוצדורה 1 :

היא מקבלת מזהה ספר וכמות לעדכון ומעדכנת את כמות הספרים של אותו ספר במאגר

Program Window - Edit source of procedure AVIGAILUPDATEBOOKAMOUNT@XE

```

1 CREATE OR REPLACE PROCEDURE UpdateBookAmount(
2   p_BookID NUMERIC,
3   p_AmountChange INT
4 ) IS
5   v_CurrentAmount INT;
6 BEGIN
7   -- Get the current amount of the book
8   SELECT Amount INTO v_CurrentAmount
9   FROM BooksCatalog
10  WHERE BookID = p_BookID;
11
12  -- Update the amount
13  v_CurrentAmount := v_CurrentAmount + p_AmountChange;
14
15  -- Update the BooksCatalog table
16  UPDATE BooksCatalog
17  SET Amount = v_CurrentAmount
18  WHERE BookID = p_BookID;
19
20  COMMIT;
21 EXCEPTION
22  WHEN OTHERS THEN
23    ROLLBACK;
24    RAISE_APPLICATION_ERROR(-20002, 'An error occurred: ' || SQLERRM);
25  END UpdateBookAmount;
26

```

Avigail@XE Compiled successfully

SQL Window - Query data of table BOOKSCATALOG@XE

SQL Output Statistics

```
select * from BOOKSCATALOG t
```

	BOOKID	BOOKNAME	AMOUNT	CATEGORYID	AUTHORID
1	864	Hughes	34	112	459
2	539	Page	34	93	902
3	638	Brooks	34	126	365
4	653	Black	34	40	971
5	718	Hauer	34	112	638
6	776	Hayek	34	40	628
7	984	Glenn	34	126	307
8	466	DiFranco	34	40	538
9	636	McGoohan	34	54	202
10	873	Getty	34	40	472
11	521	Morales	34	112	485
12	407	Franklin	34	54	753
13	419	McGinley	34	112	211
14	488	Garcia	34	137	645
15	914	Rizzo	34	112	782
16	994	Farina	34	137	933
17	829	Keith	34	54	847
18	662	Yankovic	34	126	18
19	887	Biel	34	126	801
20	739	Hackman	34	112	773
21	514	Bullock	34	40	379

Avigail@XE 21 rows selected in 0.125 seconds (more...)

אחרי הרצה:

SQL Window - Query data of table BOOKSCATALOG@XE

SQL Output Statistics

```
select * from BOOKSCATALOG t
```

	BOOKID	BOOKNAME	AMOUNT	CATEGORYID	AUTHORID
1	864	Hughes	34	112	459
2	539	Page	36	93	902
3	638	Brooks	34	126	365
4	653	Black	34	40	971
5	718	Hauer	34	112	638
6	776	Hayek	34	40	628
7	984	Glenn	34	126	307
8	466	DiFranco	34	40	538
9	636	McGoohan	34	54	202
10	873	Getty	34	40	472
11	521	Morales	34	112	485
12	407	Franklin	34	54	753
13	419	McGinley	34	112	211
14	488	Garcia	34	137	645
15	914	Rizzo	34	112	782
16	994	Farina	34	137	933
17	829	Keith	34	54	847
18	662	Yankovic	34	126	18
19	887	Biel	34	126	801
20	739	Hackman	34	112	773
21	514	Bullock	34	40	379

Avigail@XE 21 rows selected in 0.171 seconds (more...)

פרוצדורה:

מקבלת מספר ספר ובודקת אם הכמות בקטלוג גדולה יותר ממספר הספרים שמופיעים בספרים להשאלה ואם כן מוסיפה את הכמות הנדרשת כספרים הניתנים להשאלה

```

1  BEGIN
2  SELECT Amount INTO v_Amount
3  FROM BooksCatalog
4  WHERE BookID = p_BookID;
5
6  SELECT COUNT(*), COALESCE(SUM(BookNumber), 0) INTO v_BorrowedCount, v_MaxBookNumber
7  FROM BooksToBorrow
8  WHERE BookID = p_BookID;
9
10 v_ToAdd := v_Amount - v_BorrowedCount;
11
12 IF v_ToAdd > 0 THEN
13   FOR i IN 1..v_ToAdd LOOP
14     v_MaxBookNumber := v_MaxBookNumber + 1;
15     INSERT INTO BooksToBorrow (BookNumber, IsBorrow, BookID)
16     VALUES (v_MaxBookNumber, 'N', p_BookID);
17   END LOOP;
18 END IF;
19
20 COMMIT;
21
22 EXCEPTION
23 WHEN OTHERS THEN
24   ROLLBACK;
25   RAISE;
26 END UpdateBooksToBorrow;

```

התוכנית:

התוכנית מקבלת כקלט מהמשתמש מזהה קטגוריה וכמות לעדכון קורא לפונקציה ועובר על הספרים שהיא מחזירה ולכל ספר מעדכן את הכמות על ידי קריאה לפורצדורה 1 ומוסיפה ספרים זמינים להשאלה על ידי קריאה לפורצדורה 2.

```

SQL Window - main2.sql
SQL Output Statistics
v_BookName VARCHAR(15);
v_Amount INT;
v_Type VARCHAR(15);
v_CategoryID NUMBER(5) := &cname="categoryID" type=integer;
v_AmountChange INT := &cname="amount" type=integer;
BEGIN
  -- Call the function to get books by category
  v_RefCursor := GetBooksByCategory(p_CategoryID => v_CategoryID);
  LOOP
    FETCH v_RefCursor INTO v_BookID, v_BookName, v_Amount, v_Type;
    EXIT WHEN v_RefCursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('BookID: ' || v_BookID || ', BookName: ' || v_BookName || ', Amount: ' || v_Amount || ', Category: ' || v_Type);
    -- Call the procedure to update book amount
    UpdateBookAmount(p_BookID => v_BookID, p_AmountChange => v_AmountChange);
    -- Call the procedure to update booksToBorrow
    UpdateBooksToBorrow(p_BookID => v_BookID);
  END LOOP;
  CLOSE v_RefCursor;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred: ' || SQLERRM);
END;

```

תוכנית 2:

פורצדורה:

מסמנת ספר כמוחזר ומעדכנת את תאריך ההחזרה על פי מספר השאלה ותאריך החזרה שמקבלת

Program Window - Edit source of procedure AVIGAILMARKBOOKASRETURNED@XE

```
MarkBookAsReturned
Code section Loop Code section Statement
> Parameter list
> Declaration
> Code section
> Exception handler
5 CURSOR c_BooksToReturn IS
6 SELECT BookNumber
7 FROM Borrows
8 WHERE BorrowID = p_BorrowID;
9
10 v_BookNumber NUMERIC(5);
11 BEGIN
12 OPEN c_BooksToReturn;
13 LOOP
14 FETCH c_BooksToReturn INTO v_BookNumber;
15 EXIT WHEN c_BooksToReturn%NOTFOUND;
16
17 -- Update the BooksToBorrow table
18 UPDATE BooksToBorrow
19 SET isBorrow = 'N'
20 WHERE BookNumber = v_BookNumber;
21
22 -- Update the Borrows table
23 UPDATE Borrows
24 SET ReturnDate = p_ReturnDate
25 WHERE BorrowID = p_BorrowID;
26 END LOOP;
27 CLOSE c_BooksToReturn;
28 COMMIT;
29 EXCEPTION
30 WHEN OTHERS THEN
31 ROLLBACK;
32 RAISE;
33 END MarkBookAsReturned;
34
35
```

14:45 Avigail@XE Compiled successfully

לפני הרצה:

SQL Window - select * from BORROWS t where bookNumber=2067

SQL Output Statistics

select * from BORROWS t where bookNumber=2067

	BORROEID	RETURNDATE	BORROWDATE	BOOKNUMBER	CUSTOMERID	LIBRARYID
1	99	27/06/2024	11/02/2024	2067	730	284
2	308	30/05/2024	05/04/2024	2067	361	459
3	1	23/06/2024 14:46:13	28/05/2024 18:24:44	2067	909	522
4	277	23/07/2024	23/06/2024	2067	909	522

1:1 Avigail@XE borroweid, number(5), mandatory

18 2045 n 776

	BOOKNUMBER	BORROW	BOOKID
1	206	Y	636
2	3238	y	662
3	3012	Y	636
4	2767	Y	654
5	4125	y	539
6	1565	n	718
7	1043	y	407
8	3107	n	638
9	3516	y	914
10	1394	n	407
11	4834	n	776
12	1586	n	770
13	3522	n	662
14	3954	n	654
15	1380	n	636
16	342	y	653
17	4378	n	757
18	2045	n	776
19	3149	n	407
20	2391	n	662

Avigail@XE 1:1 20 rows selected in 0.125 seconds (more...)

Avigail@XE 1:1 borrowid, number(5), mandatory

```

Test Window - Script for procedure MARKBOOKASRETURNED@XE
1 begin
2   -- Call the procedure
3   markbookasreturned(p_borrowid => ip_borrowid,
4                     p_returndate => ip_returndate);
5 end;

```

Variable	Type	Value
p_borrowid	Float	993
p_returndate	Date	01/07/2006

Avigail@XE 0:03 Executed in 3.297 seconds

אחרי הרצה:

	BOOKNUMBER	BORROW	BOOKID
1	206	N	636
2	3238	y	662
3	3012	Y	636
4	2767	Y	654
5	4125	y	539
6	1565	n	718
7	1043	y	407
8	3107	n	638
9	3516	y	914
10	1394	n	407
11	4834	n	776
12	1586	n	770
13	3522	n	662
14	3954	n	654
15	1380	n	636
16	342	y	653
17	4378	n	757
18	2045	n	776
19	3149	n	407
20	2391	n	662

Avigail@XE 1:1 20 rows selected in 0.156 seconds (more...)

SQL Window - select * from BORROWS t where bookNumber=2067

SQL Output Statistics

```
select * from BORROWS t where bookNumber=2067
```

	BORROEID	RETURNDATE	BORROWDATE	BOOKNUMBER	CUSTOMERID	LIBRARYID
1	1	01/07/2024	11/02/2024	2067	730	284
2	308	30/08/2024	05/04/2024	2067	361	459
3	1	23/06/2024 14:46:13	28/05/2024 18:24:44	2067	909	522
4	277	23/07/2024	23/06/2024	2067	909	522

Avigail@XE 4 rows selected in 0.062 seconds

פונקציה:

מחשבת עמלות איחור עבור ספר מושאל היא מחזירה את הקנס בהתבסס על סמך הימים שהספר איחר לפי מספא השאלה שהיא מקבלת:

Program Window - Edit source of function AVIGAIL.CALCULATELATEFEES@XE

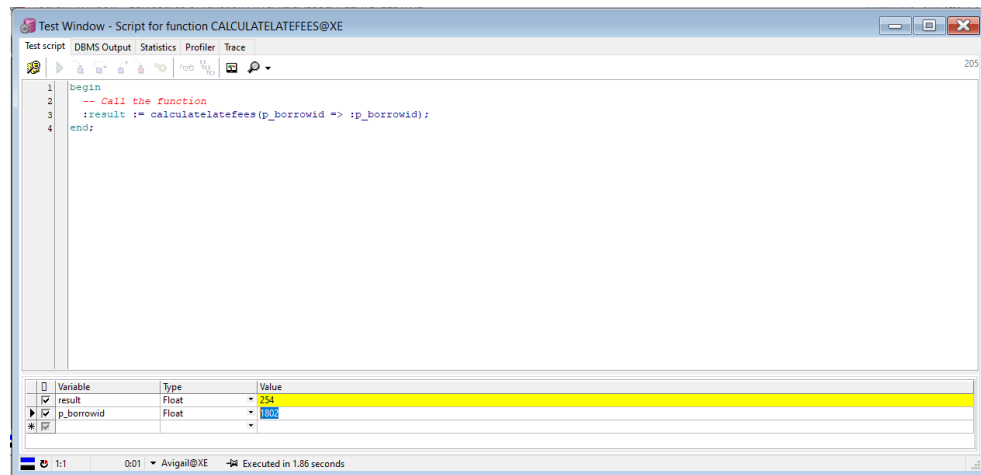
```

1 create or replace function CalculateLateFees(p_BorrowID in NUMERIC) return number is
2 v_ReturnDate DATE;
3 v_BorrowDate DATE;
4 v_FeePerDay NUMBER := 1;
5 v_LateFee NUMBER := 0;
6 v_DaysLate NUMBER;
7 BEGIN
8 SELECT BorrowDate, ReturnDate INTO v_BorrowDate, v_ReturnDate
9 FROM Borrows WHERE BorroweId = p_BorrowID;
10
11 IF v_ReturnDate IS NOT NULL THEN
12 v_DaysLate := v_ReturnDate - v_BorrowDate;
13 IF v_DaysLate > 30 THEN
14 v_LateFee := (v_DaysLate-30)*v_FeePerDay;
15 END IF;
16 END IF;
17
18 RETURN v_LateFee;
19 EXCEPTION
20 WHEN NO_DATA_FOUND THEN
21 RETURN 0;
22 WHEN OTHERS THEN
23 RETURN 0;
24 END CalculateLateFees;

```

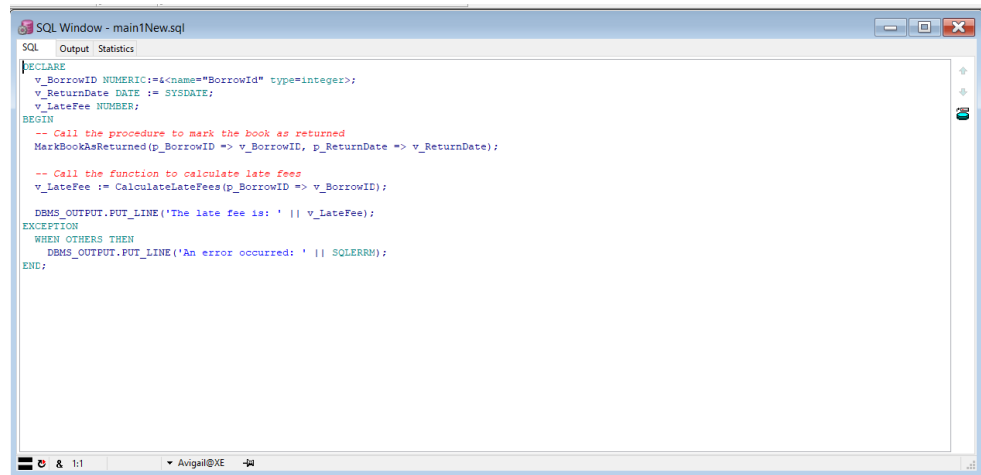
Avigail@XE 4.28

אחרי הרצה:



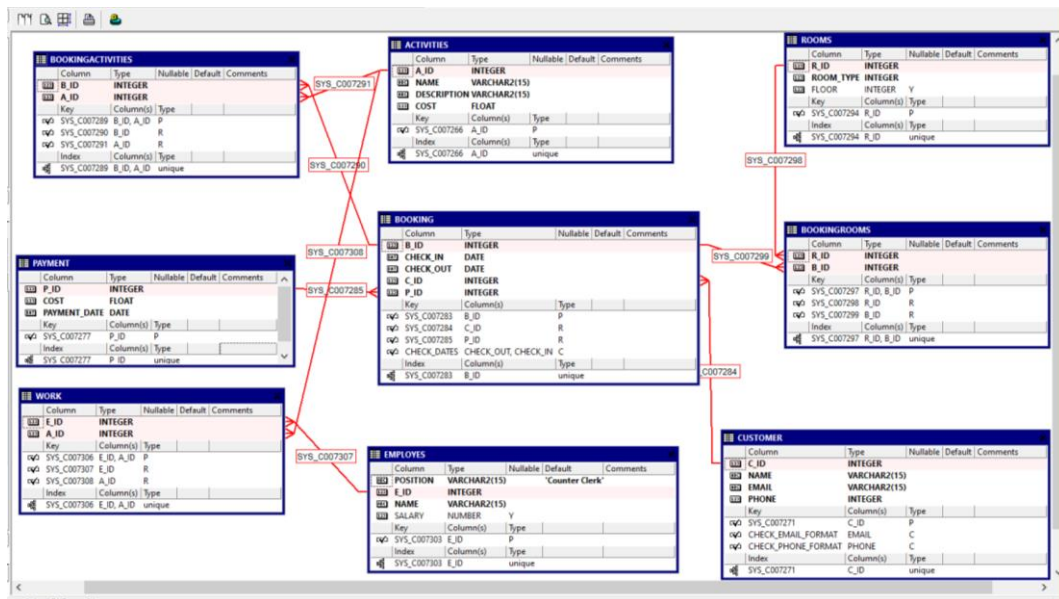
התוכנית:

מקבלת מהמשתמש מספר השאלה מעדכנת את תאריך ההחזרה והספרים להיות זמינים וכן מדפיסה אם וכמה קנס על הלקוח לשלם.

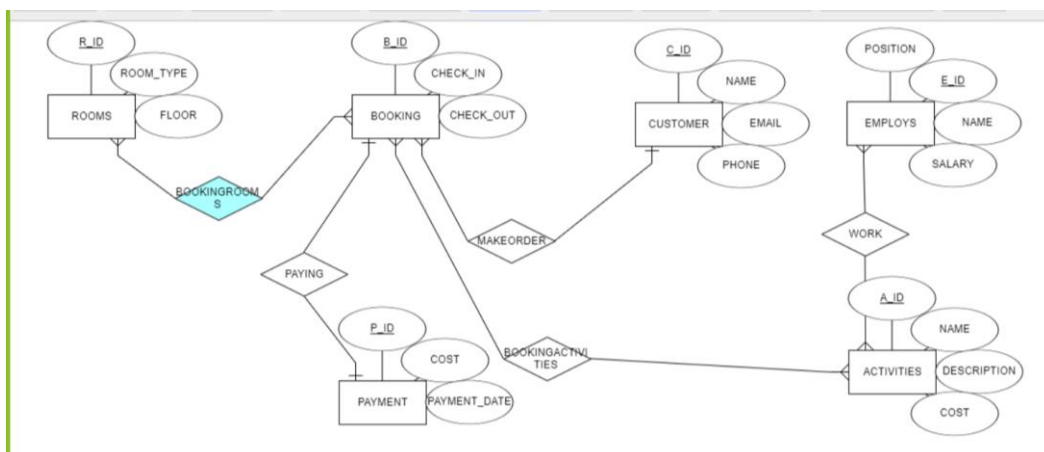


שלב 4:

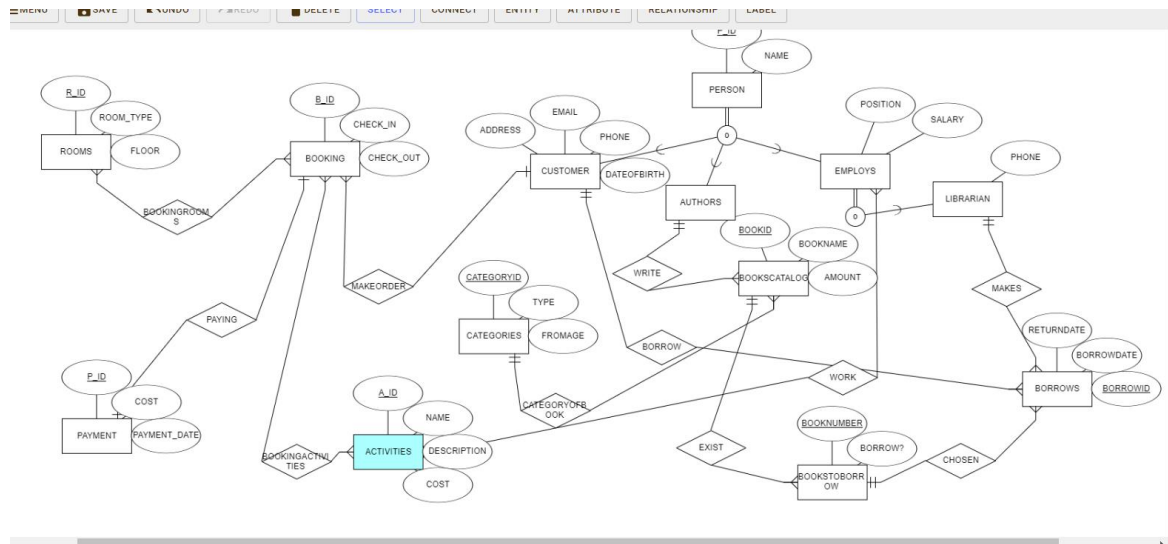
שיחזרנו את קובץ הגיבוי שקיבלנו , פתחנו דיאגרמה ב sqlplus גררנו לשם את הטבלאות החדשות וקיבלנו את ה dsd הבא:



תרשים ERD:



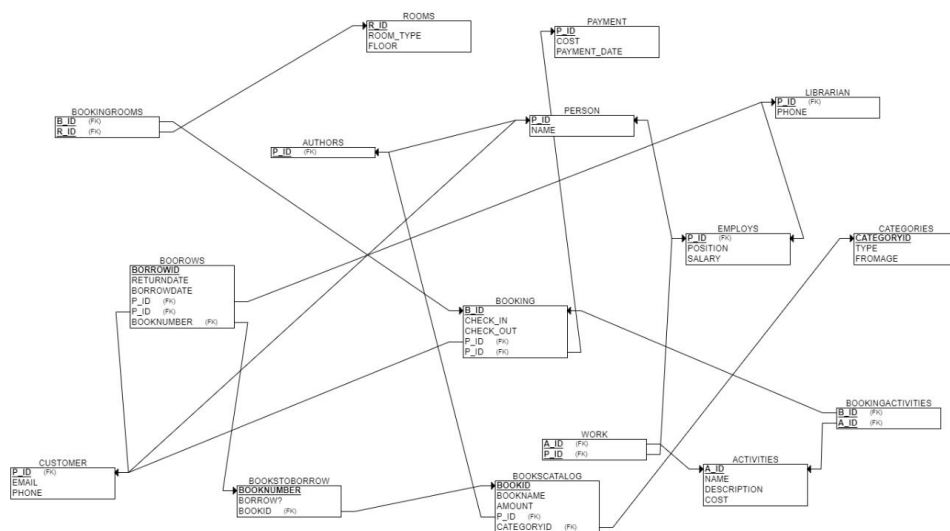
INTEGRATION ERD



החלטות:

1. גם erd שלנו וגם במה שקיבלנו הייתה ישות CUSTOMER ולכן איחדנו אותם לישות אחת תוך השלמת הפרטים החסרים.
2. הגדרנו ישות חדשה PERSON כך שCUSTOMER AUTHOR EMPLOYEE ירשו ממנה.
3. הגדרנו את LIBRARIANS לרשת מ EMPLOYEE.

IntegrationDSD:



התאמת הנתונים:

איחדנו את הטבלה customer שלהם לתוך הטבלה customers שלנו.
הוספנו עמודה EMAIL בטבלה שלנו והוספנו את העמודות החופפות לתוך הטבלה שלנו:

```

Output Statistics
alter table customers
add EMAIL varchar(15) null;

INSERT INTO CUSTOMERS (CUSTOMERID, CNAME, PHONE, EMAIL)
SELECT C_ID, name, PHONE, EMAIL
FROM CUSTOMER
WHERE CUSTOMER.C_ID NOT IN (SELECT CUSTOMERID FROM CUSTOMERS);

```

שינינו את המפתח זר בbooking שיצביע על customers שלנו אחרי האיחוד ומחקנו את הטבלה customer שלהם:

```

ALTER TABLE booking
DROP CONSTRAINT SYS_C007284;

ALTER TABLE booking
ADD CONSTRAINT fk_customer_new
FOREIGN KEY (customerId)
REFERENCES customers(customerId);

drop table customer

```

הוספנו את כל הנתונים של הישויות ששורות מ person אל הטבלה PERSON:

```

CREATE TABLE PERSON (
    p_id INT NOT NULL,
    p_name VARCHAR(15),
    PRIMARY KEY (p_id)
);

INSERT INTO PERSON (p_id, p_name)
SELECT customerID, cname
FROM CUSTOMERS

INSERT INTO PERSON (p_id, p_name)
SELECT AUTHORID, ANAME
FROM AUTHORS
WHERE AUTHORS.AUTHORID NOT IN (SELECT p_id FROM PERSON);

INSERT INTO EMPLOYES (E_ID, name, POSITION)
SELECT LIBRARYID, LNAME, 'LIBRARIAN'
FROM LIBRARIANS
WHERE LIBRARYID NOT IN (SELECT E_ID FROM EMPLOYES);

INSERT INTO PERSON (p_id, p_name)
SELECT E_ID, name
FROM EMPLOYES
WHERE EMPLOYES.E_ID NOT IN (SELECT p_id FROM PERSON);

```

שינינו את כל הטבלאות ששורות מ person שה id שלהם יצביע על ה pid של person ומחקנו את הname כך שיהיה שמור רק ב person .

בנוסף מחקנו את הקשרים של הטבלאות שהצביעו על הטבלאות ששורות מ person והוספנו מחדש לאחר שינוי המפתח.

```
SQL Output Statistics

ALTER TABLE customers ADD(p_id INT);

UPDATE customers c set c.p_id=(select p.p_id from person p where p.p_id=c.customerId);
commit;

ALTER TABLE customers ADD constraints fk_pid foreign key(p_id) references person(p_id);

ALTER TABLE booking drop constraint FK_CUSTOMER_NEW;

ALTER TABLE borrows drop constraint SYS_C007064;

ALTER TABLE customers drop primary key;

ALTER TABLE customers ADD primary key(p_id);

ALTER TABLE customers drop column customerId;

ALTER TABLE customers rename column p_id to customerId;

ALTER TABLE booking ADD constraints fk_booking_id foreign key(customerId) references customers(customerId);

ALTER TABLE borrows ADD constraints fk_borrows_id foreign key(customerId) references customers(customerId);

ALTER TABLE customers drop column cName;

ALTER TABLE authors ADD(p_id INT);

UPDATE authors a set a.p_id=(select p.p_id from person p where p.p_id=a.authorId);
commit;

ALTER TABLE authors ADD constraints fk_pid1 foreign key(p_id) references person(p_id);

ALTER TABLE bookscatalog drop constraint SYS_C007053;

ALTER TABLE authors drop primary key;

ALTER TABLE authors ADD primary key(p_id);
```

```
SQL Window - alter table customers add EMAIL varchar(15) null; INSERT INTO CUSTOMERS (CUSTOMERID, CNAME, PHON ...
SQL Output Statistics

ALTER TABLE bookscatalog drop constraint SYS_C007053;

ALTER TABLE authors drop primary key;

ALTER TABLE authors ADD primary key(p_id);

ALTER TABLE authors drop column authorId;

ALTER TABLE authors rename column p_id to authorId;

ALTER TABLE Bookscatalog ADD constraints fk_booksCatalogId foreign key(authorId) references authors(authorId);

ALTER TABLE authors drop column aName;

ALTER TABLE work drop constraint SYS_C007307;

ALTER TABLE employees drop primary key;

ALTER TABLE employees ADD constraints fk_pid2 foreign key(e_id) references person(p_id);

ALTER TABLE employees ADD primary key(e_id);

ALTER TABLE employees drop column name;

ALTER TABLE work ADD constraints fk_workId foreign key(e_id) references employees(e_id);

ALTER TABLE borrows drop constraint SYS_C007065;

ALTER TABLE librarians drop primary key;

ALTER TABLE librarians ADD constraint fk_pid3 foreign key(libraryId) references employees(e_id);

ALTER TABLE librarians ADD primary key(libraryId);

ALTER TABLE librarians drop column lName;

ALTER TABLE borrows ADD constraint fk_borrowId foreign key(libraryId) references librarians(libraryId);
```

מבטים:

מבט 1:

מביא עבור כל לקוח כמה ספרים הוא שאל סה"כ

```
SQL Window - create view customerBorrowCount as select c.customerId,count(b.borrowId) as borrowCount
SQL Output Statistics

create view customerBorrowCount as
select c.customerId,count(b.borrowId) as borrowCount
from customers c
join borrows b on c.customerId=b.customerId
group by c.customerId;
```

SQL Output Statistics		
select * from CUSTOMERBORROWCOUNT 1		
	CUSTOMERID	BORROWCOUNT
1	335	2
2	730	3
3	504	2
4	718	1
5	745	1
6	985	1
7	762	2
8	740	5
9	679	2
10	450	2
11	807	2
12	536	2
13	903	2
14	470	3
15	235	4
16	554	3
17	642	1
18	366	1
19	461	1
20	899	2
21	596	3

שאלתה 1:

שאלתה שמחשבת ממוצע ספרים שכל לקוח שאל בחודש האחרון

SQL Output Statistics		
<pre>SELECT round (AVG (borrow_count)) AS AverageBooksBorrowedLastMonth FROM (SELECT b.customerId, COUNT(b.BORROEID) AS borrow_count FROM BORROWS b WHERE b.BORROWDATE >= add_months(SYSDATE,-1) GROUP BY b.customerId)</pre>		
	AVERAGEBOOKSBORROWEDLASTMONTH	
1	2	

שאלתה 2:

שאלתה שמציגה את הלקוח ששאל הכי הרבה ספרים יחד עם הכמות שהוא שאל.

SQL Output Statistics		
<pre>WITH MaxBooks AS (SELECT MAX (borrow.BorrowCount) AS MaxBorrowCount FROM CustomerBorrowCount borrow) SELECT borrow.customerId, borrow.BorrowCount FROM CustomerBorrowCount borrow JOIN MaxBooks max_books ON borrow.BorrowCount = max_books.MaxBorrowCount;</pre>		
	CUSTOMERID	BORROWCOUNT
1	726	5

מבט 2:

המבט מציג את הנתונים הבאים: תז לקוח, שם לקוח, מספר הפעילויות שהלקוח השתתף בהם והעלות הכוללת שהלקוח השתמש בהם.

```
SQL Window - Create new customer activity summary view
SQL Output Statistics
JOIN
MaxBooks max_books ON borrow.BorrowCount = max_books.MaxBorrowCount;

CREATE VIEW CustomerActivitySummary AS
SELECT
c.customerId,
p.P_NAME,
COUNT(ba.A_ID) AS TotalActivities,
SUM(a.COST) AS TotalActivityCost
FROM
CUSTOMERS c
JOIN
PERSON p ON c.customerId = p.P_ID
JOIN
BOOKING b ON c.customerId = b.customerId
JOIN
BOOKINGACTIVITIES ba ON b.B_ID = ba.B_ID
JOIN
ACTIVITIES a ON ba.A_ID = a.A_ID
GROUP BY
c.customerId, p.P_NAME;
```

SQL Window - Query data of view CUSTOMERACTIVITYSUMMARY@XE

SQL Output Statistics

select * from CUSTOMERACTIVITYSUMMARY t

	CUSTOMERID	P_NAME	TOTALACTIVITIES	TOTALACTIVITYCOST
1	256	AaronVorn	1	65
2	226	CelliDavis	1	65
3	235	GinPhillipA	1	65
4	762	AnneKvital	1	65
5	279	Rutger-Manning	1	65
6	244	Kenny-Dalton	1	65
7	134	Suzi-Crystal	1	65
8	816	Curtis-Van Der	1	65
9	296	Dwight-Bell	1	65
10	937	Danni-Steiger	1	65
11	955	Jann-Robards	1	65
12	318	Roddy-Quinlan	2	165
13	287	Vanessa-Ball	1	65
14	325	Anna-Morton	1	65
15	641	Victoria-Bell	1	65
16	288	Owen-Lunch	1	65
17	93	Liquid-Twilley	1	65
18	151	Art-Nivola	1	65
19	370	Dylan-Winter	1	65
20	654	Garry-Neeson	1	65

Avigail@XE 20 rows selected in 0.234 seconds (more)

שאלתה 1:

השאלתה מוצאת את הלקוחות שהזמינו חדר במלון אבל לא השתתפו באף פעילות.

```
SELECT
c.customerId,
p.P_NAME
FROM
CustomerActivitySummary c
JOIN
PERSON p ON c.customerId = p.P_ID
WHERE
TotalActivities = 0;
```

שאלתה 2:

השאלתה מוצאת את הלקוחות שיש להם הוצאות על פעילויות גבוהות מהממוצע הכללי של הוצאות על פעילויות לכל הלקוחות.

SQL Output Statistics

```
FROM
  CustomerActivitySummary c
JOIN
  PERSON p ON c.customerId = p.P_ID
WHERE
  TotalActivities = 0;

WITH AvgActivityCost AS (
  SELECT
    AVG(TotalActivityCost) AS AvgCost
  FROM
    CustomerActivitySummary
)

SELECT
  c.customerId,
  p.P_NAME,
  c.TotalActivityCost
FROM
  CustomerActivitySummary c
JOIN
  PERSON p ON c.customerId = p.P_ID
JOIN
  AvgActivityCost a ON c.TotalActivityCost > a.AvgCost;
```

Grid view icons: Table, Grid, Print, Save, Undo, Redo, Find, Filter, Sort, Zoom, Refresh, Help, etc.

	CUSTOMERID	P_NAME	TOTALACTIVITYCOST
1	366	BurtonDukakis	85
2	747	HankOrlando	100
3	200	TerenceAbraham	135