

מיניפרויקט בסיסי נתונים-דוח

שם האירגון: סיפריה עירונית

מגישה: מיכל שלומי

מבוא

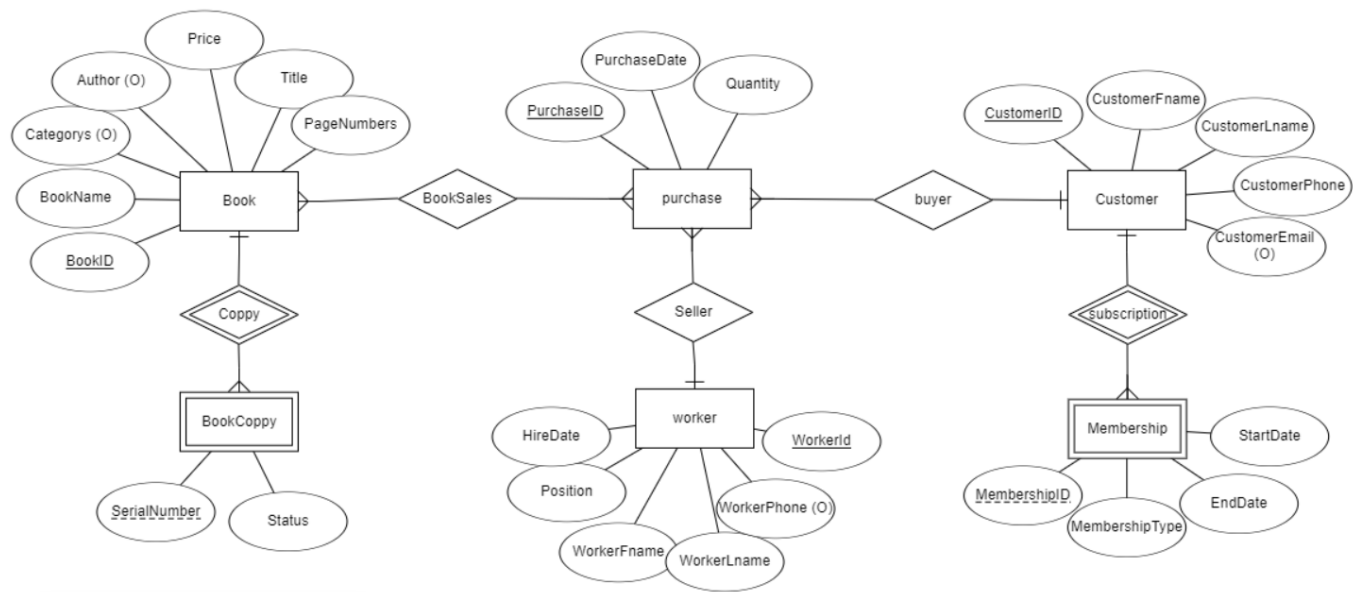
תיאור האירגון:

האירגון נותן שירותים של קניית ספרים חדשים או משומשים במצבים שונים. יש אפשרות לרכוש מנויים שונים כגון מנוי חודשי או מנוי שנתי וכד'. הסיפריה נותנת שירות לכל סוגי האוכלוסיה שבעיר, בין הספרים השונים ניתן למצוא סיפרי ילדים, סיפרי לימוד לתלמידים/סטודנטים, וכן סיפרי העשרה בכל התחומים וסיפרי קריאה בכל תחום.

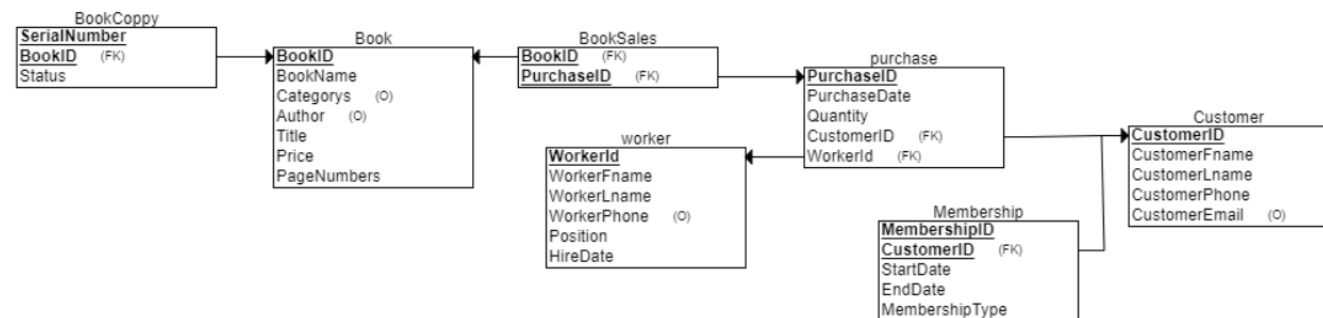
היישיות הכלולות במחלקה שלי:

- ספרים (Book)
- עותקים של ספר (BookCopy)
- לקוחות (Customer)
- מנויים (Membership)
- קניה (Purchase)
- עובדים (Worker)

תרשים ERD:



תרשים DSD:



תיאור הישויות והקשרים ביניהם:

בסיפריה הרבה ספרים שונים, ומכל ספר יש הרבה עותקים. לכל עותק יש מספר סידורי אישי שלו וכן את הקוד של הספר. כל עותק יכול להופיע רק פעם אחת ולא יכול להתקיים ללא הישות ספר. ולכן, עותק ספר הוא ישות חלשה של הישות ספר בקשר של יחיד לרבים.

לכל לקוח יכול להיות מספר מנויים שונים אך כל מנוי יכול להיות שייך ללקוח אחד בלבד ולא יכול להיות מנוי ללא לקוח ולכן מנוי הוא יישות חלשה של לקוח בקשר של יחיד לרבים.

לכל קניה יש עובד אחד שמבצע את המכירה, וכל עובד יכול לבצע הרבה מכירות שונות. ולכן, הקשר של של עובד-קניה זה קשר יחיד לרבים.

בכל קניה ניתן לרכוש מספר ספרים, ובכל ספר (bookID) יכול להימכר בקניות שונות. לכן הקשר של קניה ספר הוא קשר רבים לרבים

כל קניה מתבצעת ע"י לקוח אחד, ולקוח אחד יכול לבצע מס' רכישות שונות. לכן הקשר של לקוח קניה הוא קשר יחיד לרבים.

פקודת DESC:

```
C:\WINDOWS\system32\cmd. x + ~
```

```
SQL> DESCRIBE book;
```

Name	Null?	Type
BOOKID	NOT NULL	NUMBER
BOOKNAME	NOT NULL	VARCHAR2(30)
CATEGORYS		VARCHAR2(30)
AUTHOR		VARCHAR2(15)
TITLE	NOT NULL	VARCHAR2(30)
PRICE	NOT NULL	NUMBER
PAGENUMBERS	NOT NULL	NUMBER

```
SQL> DESCRIBE bookCopy;
```

Name	Null?	Type
SERIALNUMBER	NOT NULL	NUMBER
STATUS	NOT NULL	VARCHAR2(15)
BOOKID	NOT NULL	NUMBER

```
SQL> DESCRIBE booksales;
```

Name	Null?	Type
BOOKID	NOT NULL	NUMBER
PURCHASEID	NOT NULL	NUMBER

```
SQL> DESCRIBE customer;
```

Name	Null?	Type
CUSTOMERID	NOT NULL	NUMBER
CUSTOMERFNAME	NOT NULL	VARCHAR2(15)
CUSTOMERLNAME	NOT NULL	VARCHAR2(15)
CUSTOMERPHONE	NOT NULL	NUMBER
CUSTOMEREMAIL		VARCHAR2(30)

```
SQL> DESCRIBE membership;
```

Name	Null?	Type
MEMBERSHIPID	NOT NULL	NUMBER
STARTDATE	NOT NULL	DATE
ENDDATE	NOT NULL	DATE
MEMBERSHIPTYPE	NOT NULL	VARCHAR2(15)
CUSTOMERID	NOT NULL	NUMBER

```
SQL> DESCRIBE worker;
```

Name	Null?	Type
WORKERID	NOT NULL	NUMBER
WORKERFNAME	NOT NULL	VARCHAR2(15)
WORKERLNAME	NOT NULL	VARCHAR2(15)
WORKERPHONE		NUMBER
POSITION	NOT NULL	VARCHAR2(15)
HIREDATE	NOT NULL	DATE

```
SQL> DESCRIBE purchase;
```

Name	Null?	Type
PURCHASEID	NOT NULL	NUMBER
PURCHASEDATE	NOT NULL	DATE
QUANTITY	NOT NULL	NUMBER
CUSTOMERID	NOT NULL	NUMBER
WORKERID	NOT NULL	NUMBER

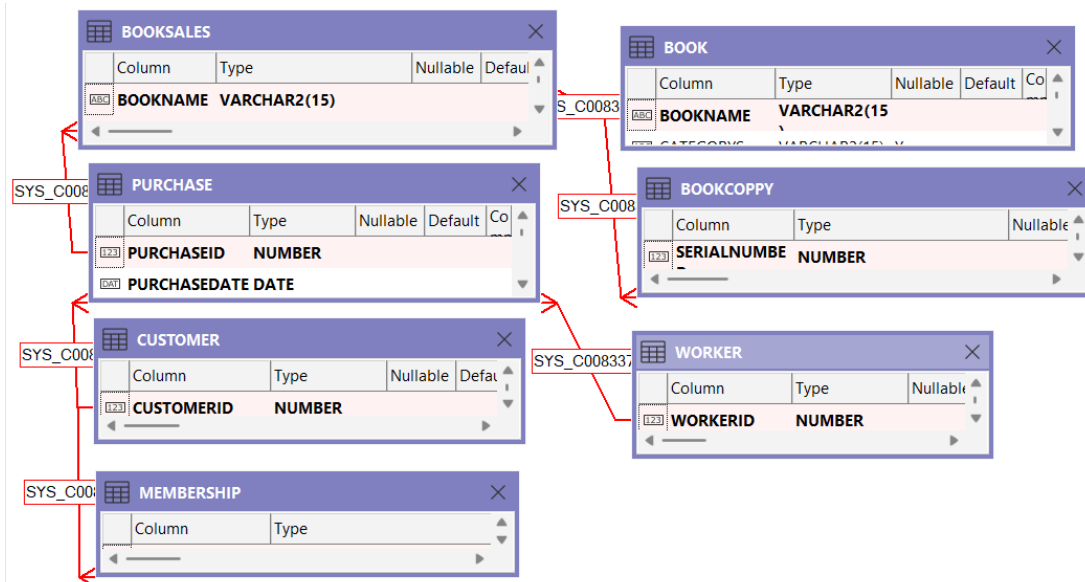
```
SQL> |
```

זה סוף הפקודות create ניתן לראות שאף אחד מהם לא מסומן באדום:

```
CREATE TABLE Membership
(
    MembershipID NUMBER NOT NULL,
    StartDate DATE NOT NULL,
    EndDate DATE NOT NULL,
    MembershipType VARCHAR2(15) NOT NULL,
    CustomerID NUMBER NOT NULL,
    PRIMARY KEY (MembershipID, CustomerID),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

CREATE TABLE BookCopy
(
    SerialNumber NUMBER NOT NULL,
    Status VARCHAR2(15) NOT NULL,
    BookID NUMBER NOT NULL,
    PRIMARY KEY (SerialNumber, BookID),
    FOREIGN KEY (BookID) REFERENCES Book(BookID)
);

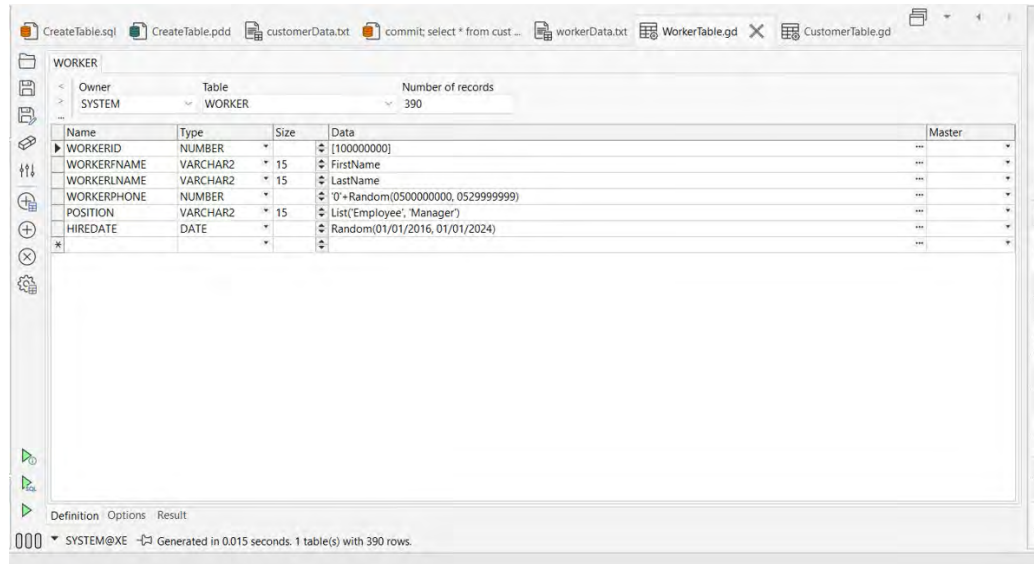
CREATE TABLE BookSales
(
    BookID NUMBER NOT NULL,
    PurchaseID NUMBER NOT NULL,
    PRIMARY KEY (BookID, PurchaseID),
    FOREIGN KEY (BookID) REFERENCES Book(BookID),
    FOREIGN KEY (PurchaseID) REFERENCES purchase(PurchaseID)
);
```



בחרתי 3 שיטות הכנסה:

1. שיטת data generator:

הכנסה לטבלת worker:

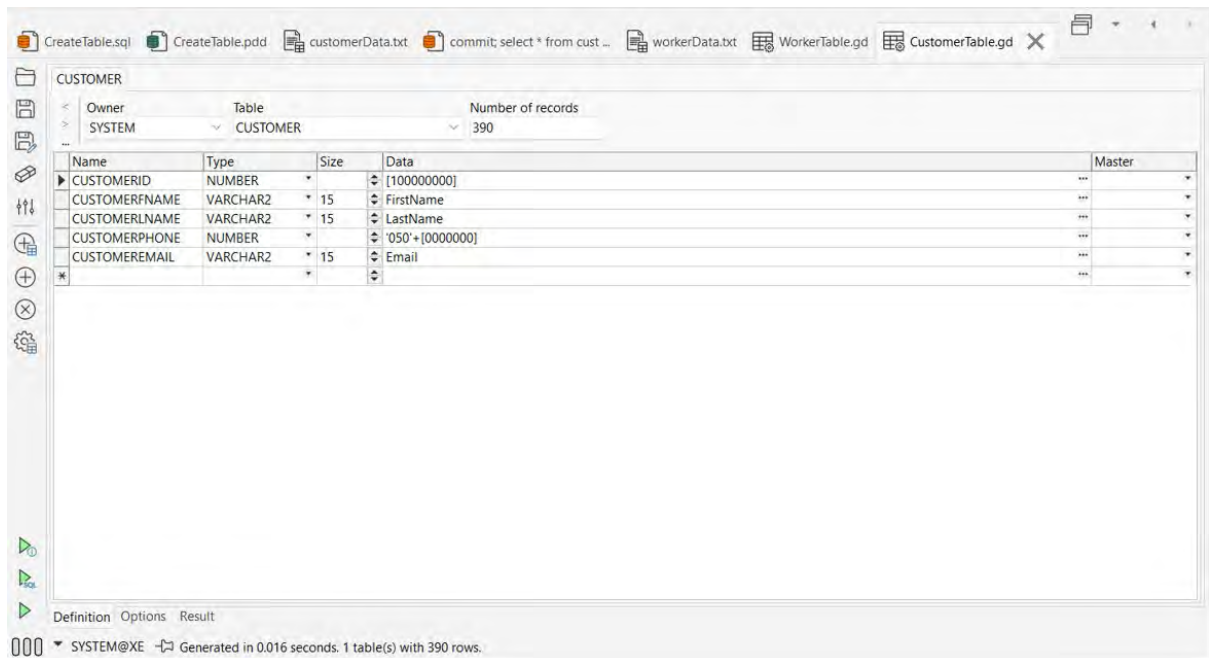


Name	Type	Size	Data	Master
WORKERID	NUMBER	*	[100000000]	
WORKERFNAME	VARCHAR2	15	FirstName	
WORKERLNAME	VARCHAR2	15	LastName	
WORKERPHONE	NUMBER	*	'0'+Random(0500000000, 0529999999)	
POSITION	VARCHAR2	15	List('Employee', 'Manager')	
HIREDATE	DATE	*	Random(01/01/2016, 01/01/2024)	

Definition Options Result

SYSTEM@XE Generated in 0.015 seconds. 1 table(s) with 390 rows.

הכנסה לטבלת customer:



Name	Type	Size	Data	Master
CUSTOMERID	NUMBER	*	[100000000]	
CUSTOMERFNAME	VARCHAR2	15	FirstName	
CUSTOMERLNAME	VARCHAR2	15	LastName	
CUSTOMERPHONE	NUMBER	*	'050'+[0000000]	
CUSTOMEREMAIL	VARCHAR2	15	Email	

Definition Options Result

SYSTEM@XE Generated in 0.016 seconds. 1 table(s) with 390 rows.

2. שיטת TXT:

הכנסה לworker:

CreateTable.sql CreateTable.pdd customerData.txt commit; select * from cust ... workerData.txt WorkerTable.gd CustomerTable.gd

Data from Textfile Data to Oracle

General

Owner: WORKER Table: WORKER

Commit every... 0

☒ Overwrite duplicates ☐ Ignore duplicates ☐ Delete records ☐ Truncate table

Initializing Script: ...

Finalizing Script: ...

Fields

Field1 -> WORKERID (NUMBER)
Field2 -> WORKERFNAME (VARCHAR2)
Field3 -> WORKERLNAME (VARCHAR2)
Field4 -> WORKERPHONE (NUMBER)
Field5 -> POSITION (VARCHAR2)
Field6 -> HIREDATE (DATE)

Field: HIREDATE (DATE)
Fieldtype: Date
Create SQL

SQL function: to_date('#', 'DD/MM/YYYY')
additional Oracle processing, for example: substr(#, 1, 20)

Result Preview

1	2	3	4	5	6
123456789	noam	levi	0504865985	Employee	01/01/2020
123568974	chen	naki	0526486564	Employee	04/07/2020
323564871	shai	gamiel	0584571323	Employee	04/05/2018
323135648	noga	okashi	0544696812	Employee	06/02/2021
315556965	avi	shilomi	0524493683	Manager	01/01/2018

Import Import to Script Close SYSTEM@XE [10:59:28] 5 records imported in 0.031 seconds Help

הכנסה לcustomer:

CreateTable.sql CreateTable.pdd customerData.txt commit; select * from cust ... workerData.txt WorkerTable.gd CustomerTable.gd

Data from Textfile Data to Oracle

General

Owner: CUSTOMER Table: CUSTOMER

Commit every... 0

☒ Overwrite duplicates ☐ Ignore duplicates ☐ Delete records ☐ Truncate table

Initializing Script: ...

Finalizing Script: ...

Fields

Field1 -> CUSTOMERID (NUMBER)
Field2 -> CUSTOMERFNAME (VARCHAR2)
Field3 -> CUSTOMERLNAME (VARCHAR2)
Field4 -> CUSTOMERPHONE (NUMBER)
Field5 -> CUSTOMEREMAIL (VARCHAR2)

Field:
Fieldtype:
Create SQL

SQL function:
additional Oracle processing, for example: substr(#, 1, 20)

Result Preview

1	2	3	4	5
325625541	tov	avni	0586645247	tom@gmail.com
245556898	dov	barak	0524960993	dov@gmail.com
987564231	nir	jank	0507884545	nir@g.com
123456789	John	Doe	0551234567	john.doe@example.com
544125895	Jane	Smith	0557654321	jane.smith@example.com

Import Import to Script Close SYSTEM@XE [10:55:00] 5 records imported in 0.281 seconds (5 duplicates) Help

3. ע"י תוכנית בשפת C

הכנסה לcustomer:

```

31
32 #include <stdio.h>
33
34 int main() {
35     int customerIDs[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
36     const char* customerFNames[10] = {
37         "John", "Jane", "Alice", "Bob", "Charlie",
38         "David", "Eva", "Frank", "Grace", "Hannah"
39     };
40     const char* customerLNames[10] = {
41         "Doe", "Smith", "Johnson", "Brown", "Williams",
42         "Jones", "Garcia", "Miller", "Davis", "Rodriguez"
43     };
44     int customerPhones[10] = {
45         1234567890, 2345678901, 3456789012, 4567890123,
46         5678901234, 6789012345, 7890123456, 8901234567, 9012345678, 1234567899
47     };
48     const char* customerEmails[10] = {
49         "john.doe@example.com", "jane.smith@example.com", "alice.johnson@example.com", "bob.brown@example.com", "charlie.williams@example.com",
50         "david.jones@example.com", "eva.garcia@example.com", "frank.miller@example.com", "grace.davis@example.com", "hannah.rodriguez@example.com"
51     };
52
53     for (int i = 0; i < 10; i++) {
54         printf("insert into Customer(CustomerID, CustomerFname, CustomerLname, CustomerPhone, CustomerEmail) \nvalues(%d, '%s', '%s', %d, '%s')\n",
55             customerIDs[i], customerFNames[i], customerLNames[i], customerPhones[i], customerEmails[i]);
56     }
57     return 0;
58 }
59

```

```

Microsoft Visual Studio Debug
insert into Customer(CustomerID, CustomerFname, CustomerLname, CustomerPhone, CustomerEmail)
values( 1, 'John', 'Doe', 1234567890, 'john.doe@example.com');
insert into Customer(CustomerID, CustomerFname, CustomerLname, CustomerPhone, CustomerEmail)
values( 2, 'Jane', 'Smith', 2345678901, 'jane.smith@example.com');
insert into Customer(CustomerID, CustomerFname, CustomerLname, CustomerPhone, CustomerEmail)
values( 3, 'Alice', 'Johnson', 3456789012, 'alice.johnson@example.com');
insert into Customer(CustomerID, CustomerFname, CustomerLname, CustomerPhone, CustomerEmail)
values( 4, 'Bob', 'Brown', 4567890123, 'bob.brown@example.com');
insert into Customer(CustomerID, CustomerFname, CustomerLname, CustomerPhone, CustomerEmail)
values( 5, 'Charlie', 'Williams', 5678901234, 'charlie.williams@example.com');
insert into Customer(CustomerID, CustomerFname, CustomerLname, CustomerPhone, CustomerEmail)
values( 6, 'David', 'Jones', 6789012345, 'david.jones@example.com');
insert into Customer(CustomerID, CustomerFname, CustomerLname, CustomerPhone, CustomerEmail)
values( 7, 'Eva', 'Garcia', 7890123456, 'eva.garcia@example.com');
insert into Customer(CustomerID, CustomerFname, CustomerLname, CustomerPhone, CustomerEmail)
values( 8, 'Frank', 'Miller', 8901234567, 'frank.miller@example.com');
insert into Customer(CustomerID, CustomerFname, CustomerLname, CustomerPhone, CustomerEmail)
values( 9, 'Grace', 'Davis', 9012345678, 'grace.davis@example.com');
insert into Customer(CustomerID, CustomerFname, CustomerLname, CustomerPhone, CustomerEmail)
values( 10, 'Hannah', 'Rodriguez', 1234567899, 'hannah.rodriguez@example.com');

C:\michal\oracleMichal\insertBookTable\x64\Debug\insertBookTable.exe (process 16688) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

```

:כנסה לworker

```

Microsoft Visual Studio Debug
insert into worker(WorkerId, WorkerFname, WorkerLname, WorkerPhone, Position, HireDate)
values(11, 'Alice', 'Smith', 1234567890, 'Manager', TO_DATE('2020-01-15', 'YYYY-MM-DD'));
insert into worker(WorkerId, WorkerFname, WorkerLname, WorkerPhone, Position, HireDate)
values(12, 'Bob', 'Johnson', 2345678901, 'Engineer', TO_DATE('2019-03-22', 'YYYY-MM-DD'));
insert into worker(WorkerId, WorkerFname, WorkerLname, WorkerPhone, Position, HireDate)
values(13, 'Charlie', 'Williams', 3456789012, 'Technician', TO_DATE('2021-07-30', 'YYYY-MM-DD'));
insert into worker(WorkerId, WorkerFname, WorkerLname, WorkerPhone, Position, HireDate)
values(14, 'David', 'Jones', 4567890123, 'Clerk', TO_DATE('2018-11-12', 'YYYY-MM-DD'));
insert into worker(WorkerId, WorkerFname, WorkerLname, WorkerPhone, Position, HireDate)
values(15, 'Eva', 'Brown', 5678901234, 'Analyst', TO_DATE('2017-05-06', 'YYYY-MM-DD'));
insert into worker(WorkerId, WorkerFname, WorkerLname, WorkerPhone, Position, HireDate)
values(16, 'Frank', 'Davis', 6789012345, 'Supervisor', TO_DATE('2022-09-18', 'YYYY-MM-DD'));
insert into worker(WorkerId, WorkerFname, WorkerLname, WorkerPhone, Position, HireDate)
values(17, 'Grace', 'Miller', 7890123456, 'Operator', TO_DATE('2016-02-25', 'YYYY-MM-DD'));
insert into worker(WorkerId, WorkerFname, WorkerLname, WorkerPhone, Position, HireDate)
values(18, 'Hannah', 'Wilson', 8901234567, 'Coordinator', TO_DATE('2023-04-11', 'YYYY-MM-DD'));
insert into worker(WorkerId, WorkerFname, WorkerLname, WorkerPhone, Position, HireDate)
values(19, 'Ivy', 'Moore', 9012345678, 'Administrator', TO_DATE('2015-08-19', 'YYYY-MM-DD'));
insert into worker(WorkerId, WorkerFname, WorkerLname, WorkerPhone, Position, HireDate)
values(20, 'Jack', 'Taylor', 1234567809, 'Consultant', TO_DATE('2014-12-29', 'YYYY-MM-DD'));

C:\michal\oracleMichal\insertBookTable\x64\Debug\insertBookTable.exe (process 8112) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

```


הרצה של * select על כל אחת מהטבלאות:

The image displays two screenshots of the PL/SQL Developer application, showing the execution of SQL queries and the resulting data.

Top Screenshot:

- SQL Window:** The query is `select * from book;`. The results are displayed in a table with columns: PURCHASEID, PURCHASEDATE, QUANTITY, CUSTOMERID, and WORKERID. The table contains 16 rows of data.
- PL/SQL Clipboard:** The clipboard contains the same query: `select * from book;`.
- Window List:** The list shows the current window is `SQL Window - CreateTable.sql`.

Bottom Screenshot:

- SQL Window:** The query is `select * from membership;`. The results are displayed in a table with columns: MEMBERSHIPID, STARTDATE, ENDDATE, MEMBERSHIPTYPE, and CUSTOMERID. The table contains 16 rows of data.
- PL/SQL Clipboard:** The clipboard contains the same query: `select * from membership;`.
- Window List:** The list shows the current window is `SQL Window - CreateTable.sql`.

SQL Window - CreateTable.sql

```
select * from book;
select * from bookcopy;
select * from booksales;
select * from worker;
select * from customer;
select * from membership;
select * from purchase;
```

	CUSTOMERID	CUSTOMERNAME	CUSTOMEREMAIL
1	325625541	toiv	avni
2	243556898	dov	barak
3	987564231	nir	jank
4	123456789	John	Doe
5	544125895	Jane	Smith
6	263933120	Annette	Nothe
7	421980038	Harris	Redgrave
8	143136513	Saul	Pollack
9	906571901	Lari	Matarazzo
10	428081030	Etta	Pierce
11	766023468	Rolando	Kravitz
12	938819453	Meryl	Parsons
13	945109317	Rose	Judd
14	374349078	Ned	Giraldi
15	301747054	Lindsay	Sweet
16	643749764	Thelma	Fluke

SQL Window - CreateTable.sql

405 rows selected in 0.550 seconds

SQL Window - CreateTable.sql

```
select * from book;
select * from bookcopy;
select * from booksales;
select * from worker;
select * from customer;
select * from membership;
select * from purchase;
```

	WORKERID	WORKERFNAME	WORKERLNAME	WORKERPHONE	POSITION	HIREDATE
1	123456789	noam	levi	504865985	Employee	01/01/2020
2	123568974	chen	naki	526486564	Employee	04/07/2020
3	323564871	shai	garniel	584571323	Employee	04/05/2018
4	323135648	noga	okashi	544696812	Employee	06/02/2021
5	315556965	avi	shlomi	524493683	Manager	01/01/2018
6	226755554	Cesar	Wincott	515565640	Employee	12/06/2017
7	743502274	Hikaru	Stevenson	511360250	Manager	17/09/2023
8	718109763	Emilio	Pollack	500869993	Manager	19/04/2022
9	335673276	Marlon	Moraz	513212401	Employee	11/11/2020
10	901683660	Marianne	Shatner	517220436	Manager	29/01/2019
11	137920011	Andy	Fender	500238138	Employee	02/02/2022
12	397989285	Harold	Caine	524902515	Manager	01/03/2017
13	38138523	Winona	Reiner	500079412	Manager	04/04/2023
14	294357583	Leo	Short	511008998	Employee	04/01/2018
15	519629540	Morgan	Gambon	511537349	Employee	06/04/2019
16	803048636	Fajar	Bany	639538556	Employee	30/13/2020

SQL Window - CreateTable.sql

405 rows selected in 0.570 seconds

SQL Window - CreateTables.sql

```
select * from book;
select * from bookcopy;
select * from booksales;
select * from worker;
select * from customer;
select * from membership;
select * from purchase;
```

	BOOKID	PURCHASEID
1	23	123
2	23	321
3	24	123
4	25	321
5	26	321
6	27	321
7	28	321
8	29	321
9	30	321
10	30	456
11	111111	123
12	111115	123
13	111115	321
14	111115	456
15	111119	123
16	111150	321

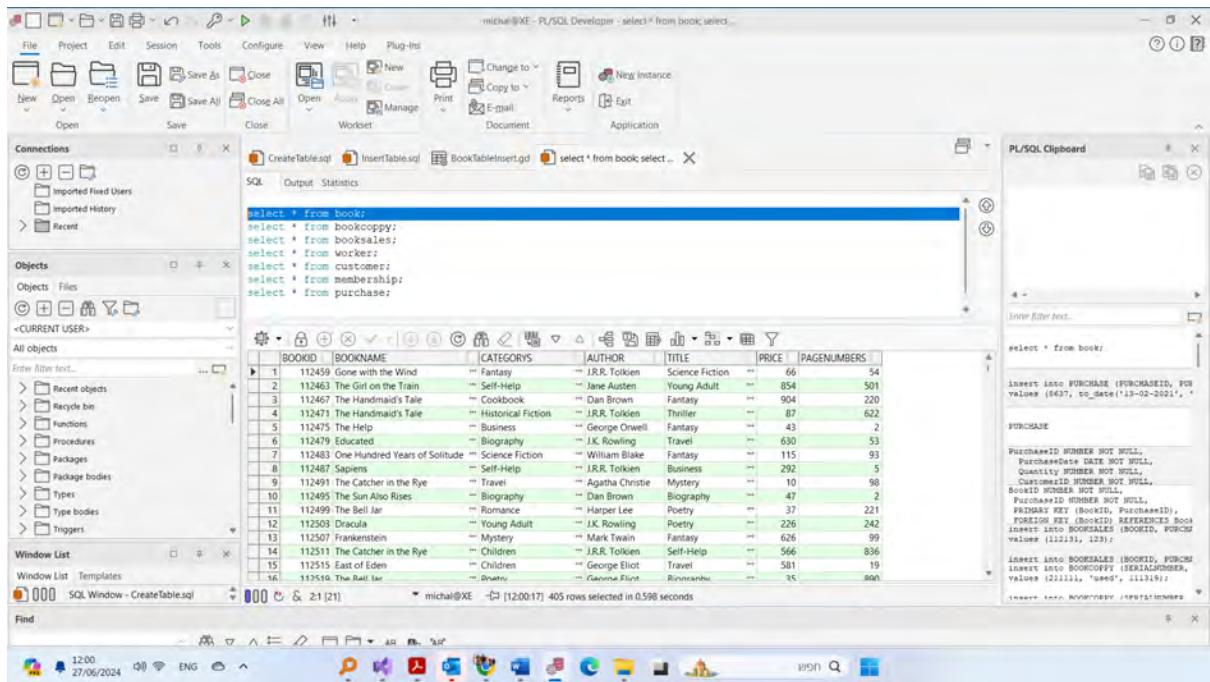
SQL Window - CreateTables.sql 4:1 [26] 409 rows selected in 0.486 seconds

SQL Window - CreateTables.sql

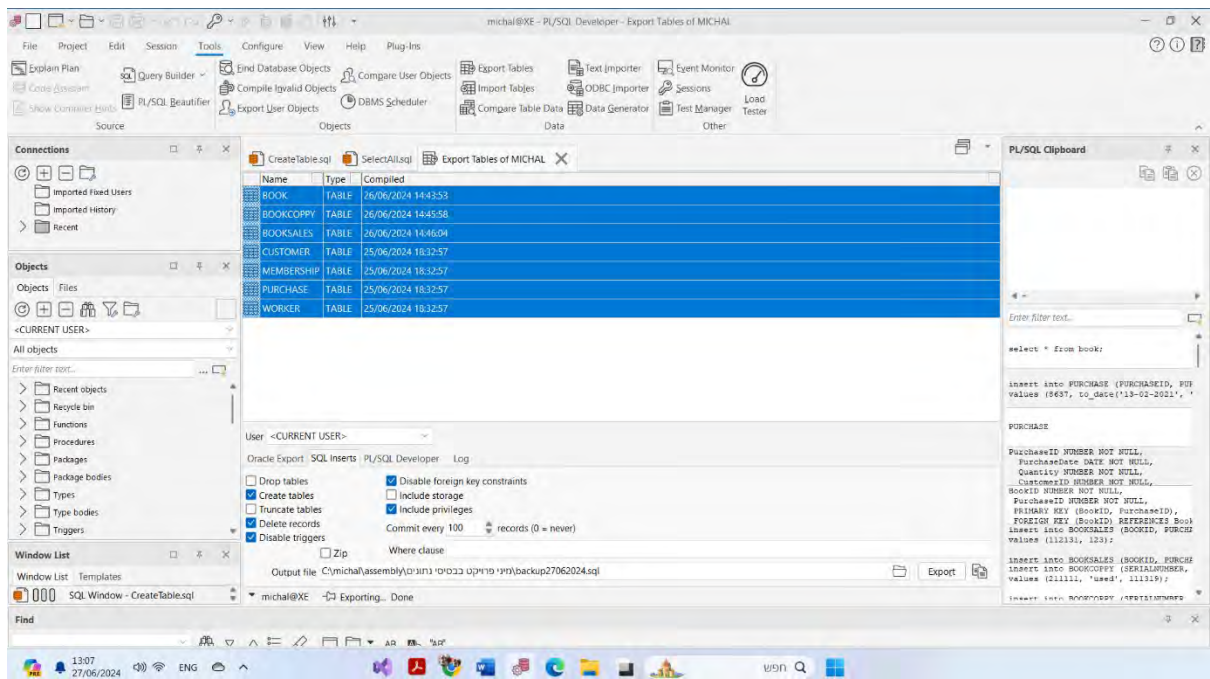
```
select * from book;
select * from bookcopy;
select * from booksales;
select * from worker;
select * from customer;
select * from membership;
select * from purchase;
```

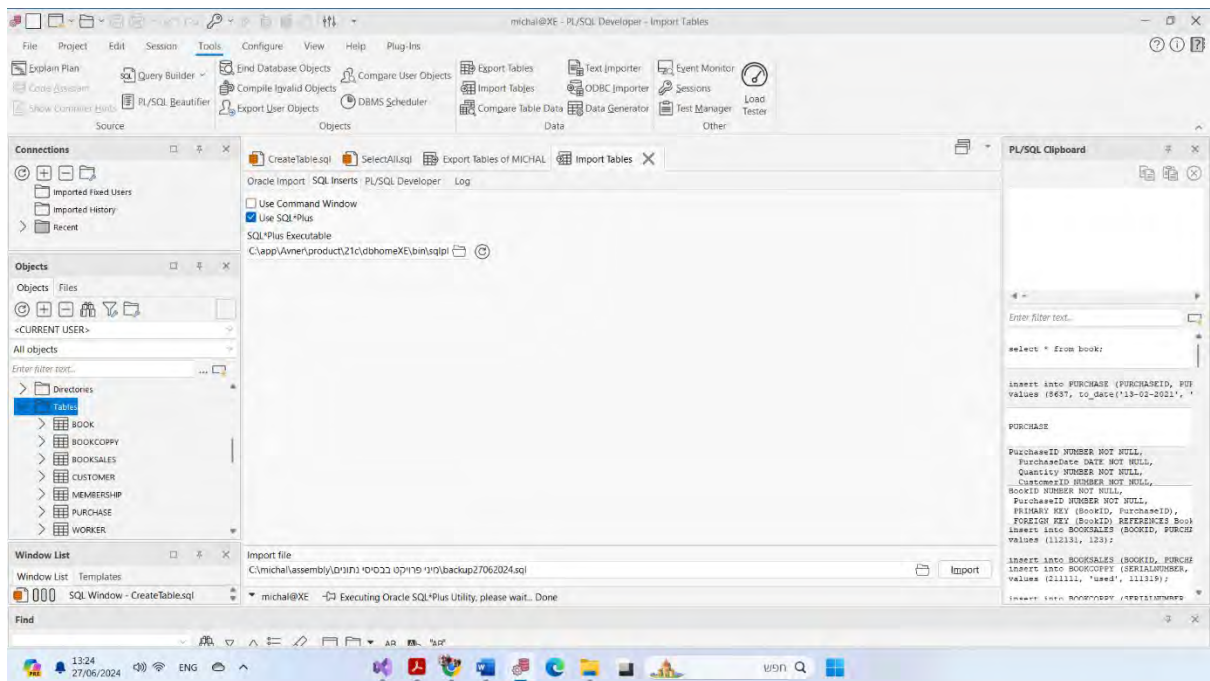
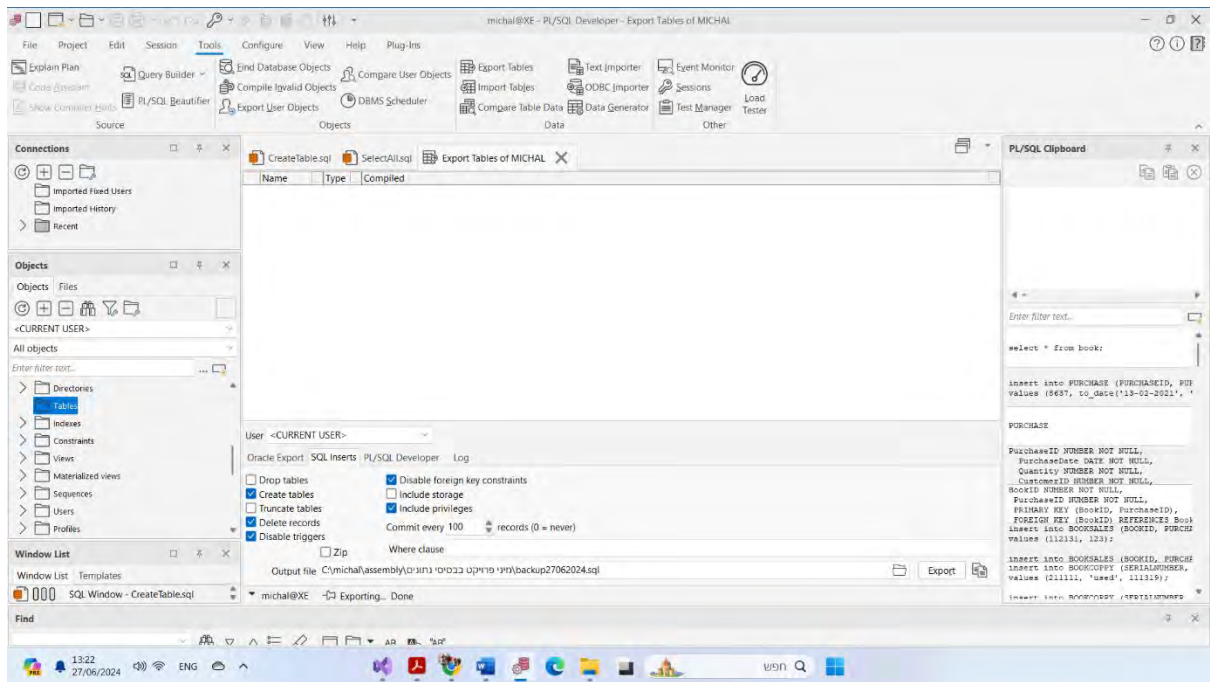
	SERIALNUMBER	STATUS	BOOKID
1	12502	new	112459
2	4780003	used	112467
3	96504	new	112463
4	211111	used	112287
5	211115	used	112539
6	211119	new	111883
7	211123	new	26
8	211127	used	112191
9	211131	new	112067
10	211135	new	112231
11	211139	new	111291
12	211143	used	111499
13	211147	used	112267
14	211151	new	111487
15	211155	new-used	111707
16	211159	new	111473

SQL Window - CreateTables.sql 3:1 [26] 403 rows selected in 0.507 seconds



צילומי מסך של ביצוע גיבוי ושיחזור הטבלאות:





PL/SQL Developer - SelectAll.sql

File Project Edit Session Tools Configure View Help Plug-Ins

Connections: Importing Fixed Users, Importing History, Recent

Objects: Objects, Files, <CURRENT USER>

Window List: SQL Window - CreateTable.sql

SQL Window - CreateTable.sql

```
SQL Output Statistics
select * from books;
select * from bookcopy;
select * from booksales;
select * from worker;
select * from customer;
select * from membership;
select * from purchase;
commit;
```

210 rows selected in 0.504 seconds (more...)

BOOKID	BOOKNAME	CATEGORIES	AUTHOR	TITLE	PRICE	PAGENUMBERS	
1	112459	Gone with the Wind	Fantasy	J.R.R. Tolkien	Science Fiction	66	54
2	112463	The Girl on the Train	Self-Help	Jane Austen	Young Adult	854	501
3	112467	The Handmaid's Tale	Cookbook	Dan Brown	Fantasy	904	220
4	112471	The Handmaid's Tale	Historical Fiction	J.R.R. Tolkien	Thriller	87	622
5	112475	The Help	Business	George Orwell	Fantasy	43	2
6	112479	Educated	Biography	J.K. Rowling	Travel	630	53
7	112483	One Hundred Years of Solitude	Science Fiction	William Blake	Fantasy	115	93
8	112487	Sapiens	Self-Help	J.R.R. Tolkien	Business	292	5
9	112491	The Catcher in the Rye	Travel	Agatha Christie	Mystery	10	98
10	112495	The Sun Also Rises	Biography	Dan Brown	Biography	47	2
11	112499	The Bell Jar	Romance	Harper Lee	Poetry	37	221
12	112503	Dracula	Young Adult	J.K. Rowling	Poetry	226	242
13	112507	Frankenstein	Mystery	Mark Twain	Fantasy	626	99
14	112511	The Catcher in the Rye	Children	J.R.R. Tolkien	Self-Help	566	836
15	112515	East of Eden	Children	George Eliot	Travel	581	19
16	112519	The Ball Jar	Thriller	George Eliot	Romance	35	890

PL/SQL Clipboard

```
select * from books;

insert into PURCHASE (PURCHASEID, PUR
values (6437, to_date('13-02-2021', '

PURCHASE

PurchaseID NUMBER NOT NULL,
PurchaseDate DATE NOT NULL,
Quantity NUMBER NOT NULL,
CustomerID NUMBER NOT NULL,
BookID NUMBER NOT NULL,
PRIMARY KEY (BookID, PurchaseID),
FOREIGN KEY (BookID) REFERENCES BOOK
insert into BOOKSALES (BOOKID, PURCH
values (112131, 123);

insert into BOOKSALES (BOOKID, PURCH
insert into BOOKCOPY (SERIALNUMBER,
values (121111, 'new', 111119);

insert into BOOKCOPY (SERIALNUMBER
```