

## **דוח מיני פרויקט בבסיסי נתונים**

### מגישות:

רעיה פלדמר

חיה לקס

מערכת: ספריה

יחידה: השאלת ספרים

## תוכן עניינים:

1	שער
2	תוכן עניינים
3	מבוא
4	תרשים ERD
4	תרשים DSD
5	פקודות create table
7	צילום הפעלת פקודות desc
10	הכנסת נתונים
16	צילום גיבוי
16	צילום שחזור

## **מבוא:**

מערכת השאלת הספרים כוללת: ספרניות, לקוחות עם חשבון להשאלת ספרים, ספרים, משחקים והשאלות.

ישנן 2 סוגי השאלות- השאלת ספרים והשאלת משחקים.  
כל השאלה מורכבת מהלקוח שהשאל, הספרנית שביצעה את ההשאלה, הספר/משחק שהושאל, תאריך ההשאלה, תאריך אחרון להחזרה, תאריך החזרה(במידה והוחזר) ומספר השאלה.

גם עבור הספרנית וגם עבור הלקוחות נשמרים הת"ז, השם, הפלאפון, המייל.  
בנוסף, לספרנית נשמר גם תאריך תחילת העבודה, ושם משתמש וסיסמה עבור כניסה למערכת.

וללקוח נשמרים מספר ספרים מקסימלי להשאלה והאם הוא פעיל (יכול להשאל).

עבור כל ספר נשמרים מספר הספר, הכותרת, מספר הנושא, מספר הוצאה לאור, כותרת ושפה.

ועבור כל עותק של הספר נשמר מספר הספר, מספר עותק, שנת הוצאה, מהדורה וזמינותו.

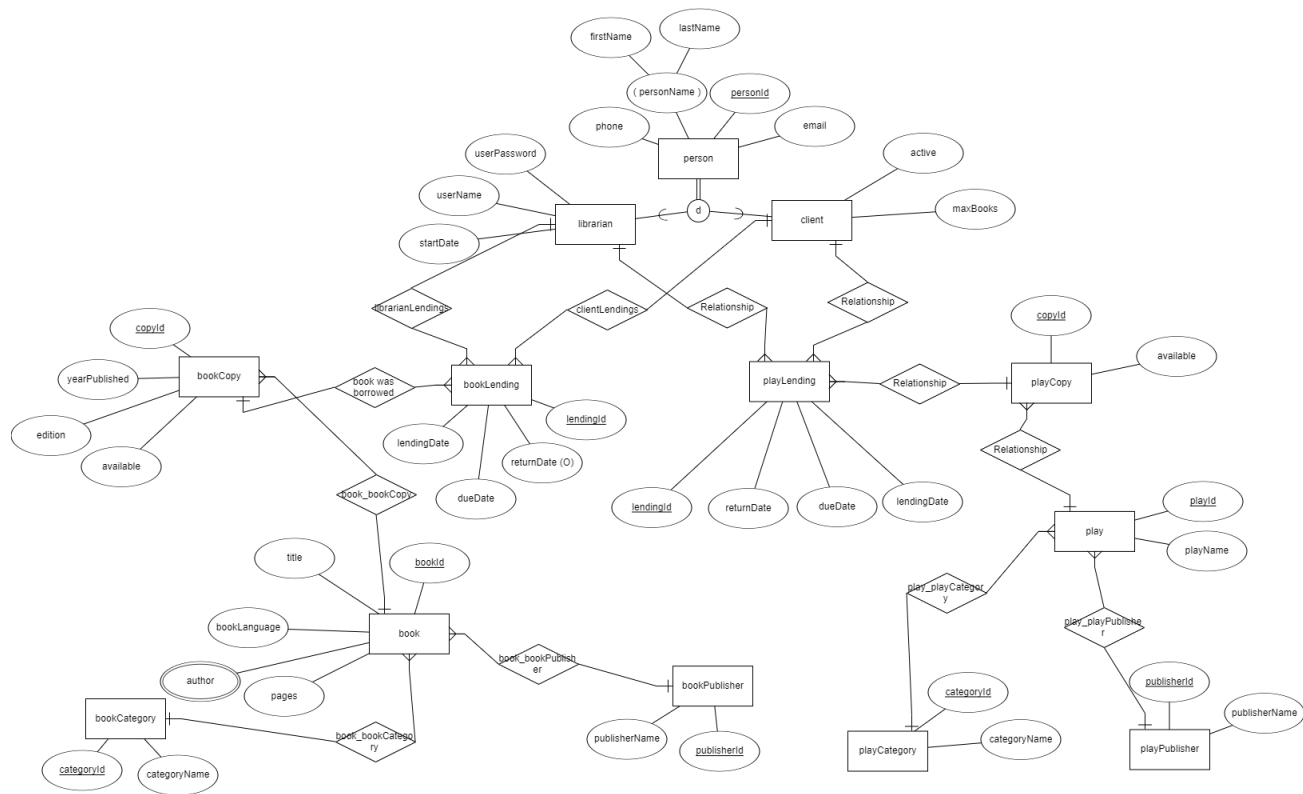
עבור כל משחק נשמרים מספר המשחק, שם המשחק, מספר נושא ומספר חברה.  
ועבור כל עותק של המשחק נשמר מספר המשחק, מספר העותק וזמינותו.

## **החלטות עיצוב:**

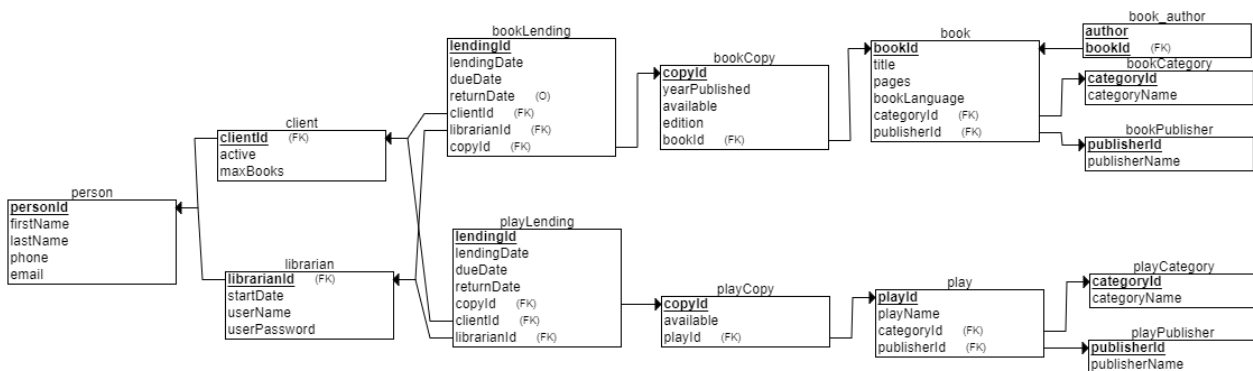
בחרנו שכל השאלה תכיל ספר/משחק אחד ולא את כל הספרים והמשחקים מאותו הזמן ע"מ שיהיה אפשרות להחזיר כל ספר בזמן אחר ולא יהיה צריך להחזיר את כל מה שהושאל ביחד.

על הלקוח נשמר אם הוא פעיל כיוון שגם עבור לקוחות שאינם פעילים אנו רוצים לדעת את היסטוריית ההחלפות שלהם לצרכים סטטיסטיים וכדו'.

## תרשים ERD:

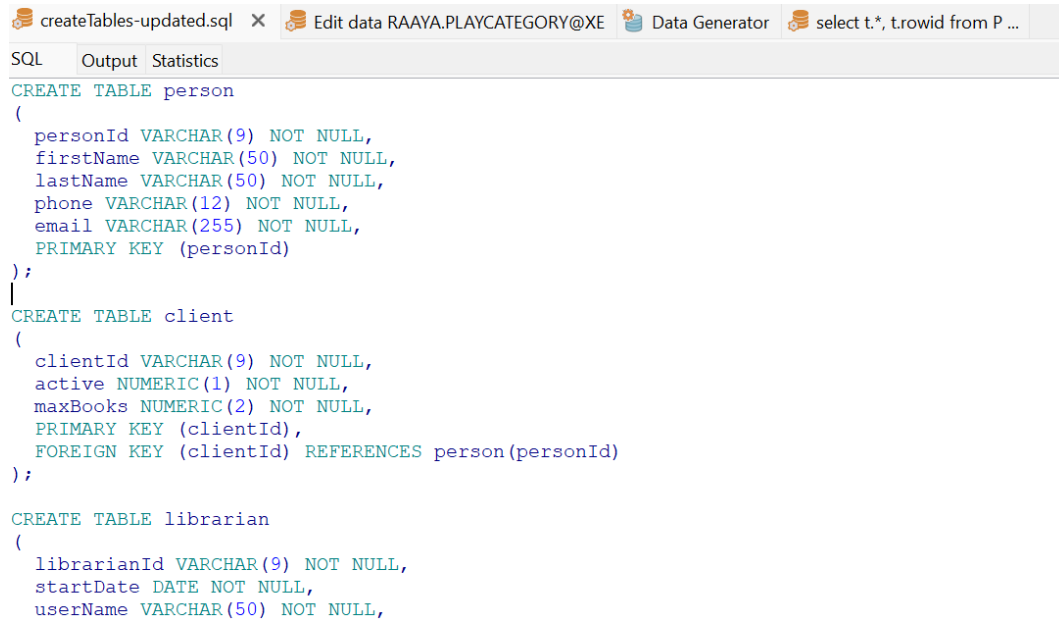


## תרשים DSD



## פקודות createTable:

צילום מסך



```
createTables-updated.sql X Edit data RAAVA.PLAYCATEGORY@XE Data Generator select t.*, t.rowid from P ...
SQL Output Statistics
CREATE TABLE person
(
  personId VARCHAR(9) NOT NULL,
  firstName VARCHAR(50) NOT NULL,
  lastName VARCHAR(50) NOT NULL,
  phone VARCHAR(12) NOT NULL,
  email VARCHAR(255) NOT NULL,
  PRIMARY KEY (personId)
);
CREATE TABLE client
(
  clientId VARCHAR(9) NOT NULL,
  active NUMERIC(1) NOT NULL,
  maxBooks NUMERIC(2) NOT NULL,
  PRIMARY KEY (clientId),
  FOREIGN KEY (clientId) REFERENCES person(personId)
);
CREATE TABLE librarian
(
  librarianId VARCHAR(9) NOT NULL,
  startDate DATE NOT NULL,
  userName VARCHAR(50) NOT NULL,
```

כל פקודות הcreate tablen בעמוד הבא.

```

CREATE TABLE person
(
    personId VARCHAR(9) NOT NULL,
    firstName VARCHAR(50) NOT NULL,
    lastName VARCHAR(50) NOT NULL,
    phone VARCHAR(12) NOT NULL,
    email VARCHAR(255) NOT NULL,
    PRIMARY KEY (personId)
);

CREATE TABLE client
(
    active NUMERIC(1) NOT NULL,
    maxBooks NUMERIC(2) NOT NULL,
    clientId VARCHAR(9) NOT NULL,
    PRIMARY KEY (clientId),
    FOREIGN KEY (clientId) REFERENCES person(personId)
);

CREATE TABLE librarian
(
    startDate DATE NOT NULL,
    userName VARCHAR(50) NOT NULL,
    userPassword VARCHAR(50) NOT NULL,
    librarianId VARCHAR(9) NOT NULL,
    PRIMARY KEY (librarianId),
    FOREIGN KEY (librarianId) REFERENCES person(personId)
);

CREATE TABLE bookCategory
(
    categoryName VARCHAR(50) NOT NULL,
    categoryId NUMERIC(3) NOT NULL,
    PRIMARY KEY (categoryId)
);

CREATE TABLE bookPublisher
(
    publisherName VARCHAR(50) NOT NULL,
    publisherId NUMERIC(4) NOT NULL,
    PRIMARY KEY (publisherId)
);

CREATE TABLE playCategory
(
    categoryId NUMERIC(3) NOT NULL,
    categoryName VARCHAR(50) NOT NULL,
    PRIMARY KEY (categoryId)
);

CREATE TABLE playPublisher
(
    publisherId NUMERIC(4) NOT NULL,
    publisherName VARCHAR(50) NOT NULL,
    PRIMARY KEY (publisherId)
);

CREATE TABLE book
(
    bookId NUMERIC(6) NOT NULL,
    title VARCHAR(50) NOT NULL,
    pages NUMERIC(4) NOT NULL,
    bookLanguage VARCHAR(50) NOT NULL,
    categoryId NUMERIC(3) NOT NULL,
    publisherId NUMERIC(4) NOT NULL,
    PRIMARY KEY (bookId),
    FOREIGN KEY (categoryId) REFERENCES bookCategory(categoryId),
    FOREIGN KEY (publisherId) REFERENCES bookPublisher(publisherId)
);

```

```

CREATE TABLE play
(
    playId NUMERIC(6) NOT NULL,
    playName VARCHAR(50) NOT NULL,
    categoryId NUMERIC(3) NOT NULL,
    publisherId NUMERIC(4) NOT NULL,
    PRIMARY KEY (playId),
    FOREIGN KEY (categoryId) REFERENCES playCategory(categoryId),
    FOREIGN KEY (publisherId) REFERENCES playPublisher(publisherId)
);

CREATE TABLE bookCopy
(
    copyId NUMERIC(8) NOT NULL,
    yearPublished NUMERIC(4) NOT NULL,
    available NUMERIC(1) NOT NULL,
    edition NUMERIC(3) NOT NULL,
    bookId NUMERIC(6) NOT NULL,
    PRIMARY KEY (copyId),
    FOREIGN KEY (bookId) REFERENCES book(bookId)
);

CREATE TABLE playCopy
(
    copyId NUMERIC(8) NOT NULL,
    available NUMERIC(1) NOT NULL,
    playId NUMERIC(6) NOT NULL,
    PRIMARY KEY (copyId),
    FOREIGN KEY (playId) REFERENCES play(playId)
);

CREATE TABLE playLending
(
    lendingId INT NOT NULL,
    lendingDate DATE NOT NULL,
    dueDate DATE NOT NULL,
    returnDate DATE NOT NULL,
    copyId NUMERIC(8) NOT NULL,
    clientId VARCHAR(9) NOT NULL,
    librarianId VARCHAR(9) NOT NULL,
    PRIMARY KEY (lendingId),
    FOREIGN KEY (copyId) REFERENCES playCopy(copyId),
    FOREIGN KEY (clientId) REFERENCES client(clientId),
    FOREIGN KEY (librarianId) REFERENCES librarian(librarianId)
);

CREATE TABLE book_author
(
    author VARCHAR(100) NOT NULL,
    bookId NUMERIC(6) NOT NULL,
    PRIMARY KEY (author, bookId),
    FOREIGN KEY (bookId) REFERENCES book(bookId)
);

CREATE TABLE bookLending
(
    lendingDate DATE NOT NULL,
    returnDate DATE,
    lendingId INT NOT NULL,
    dueDate DATE NOT NULL,
    clientId VARCHAR(9) NOT NULL,
    librarianId VARCHAR(9) NOT NULL,
    copyId NUMERIC(8) NOT NULL,
    PRIMARY KEY (lendingId),
    FOREIGN KEY (clientId) REFERENCES client(clientId),
    FOREIGN KEY (librarianId) REFERENCES librarian(librarianId),
    FOREIGN KEY (copyId) REFERENCES bookCopy(copyId)
);

```

## צילום הפעלת פקודות desc:

	Name	Type	Nullable	Default	Comments
►	PUBLISHERID	NUMBER(4)			
2	PUBLISHERNAME	VARCHAR2(50)			

	Name	Type	Nullable	Default	Comments
►	LENDINGID	INTEGER			
2	LENDINGDATE	DATE			
3	DUE DATE	DATE			
4	RETURN DATE	DATE	Y		
5	COPYID	NUMBER(8)			
6	CLIENTID	VARCHAR2(9)			
7	LIBRARIANID	VARCHAR2(9)			

	Name	Type	Nullable	Default	Comments
►	COPYID	NUMBER(8)			
2	AVAILABLE	NUMBER(1)			
3	PLAYID	NUMBER(6)			

	Name	Type	Nullable	Default	Comments
►	CATEGORYID	NUMBER(3)			
2	CATEGORYNAME	VARCHAR2(50)			

	Name	Type	Nullable	Default	Comments
►	PLAYID	NUMBER(6)			
2	PLAYNAME	VARCHAR2(50)			
3	CATEGORYID	NUMBER(3)			
4	PUBLISHERID	NUMBER(4)			

	Name	Type	Nullable	Default	Comments
►	PERSONID	VARCHAR2(9)			
2	FIRSTNAME	VARCHAR2(50)			
3	LASTNAME	VARCHAR2(50)			
4	PHONE	VARCHAR2(12)			
5	EMAIL	VARCHAR2(255)			

	Name	Type	Nullable	Default	Comments
1	LIBRARIANID	VARCHAR2(9)			
2	STARTDATE	DATE			
3	USERNAME	VARCHAR2(50)			
4	USERPASSWORD	VARCHAR2(50)			

	Name	Type	Nullable	Default	Comments
1	CLIENTID	VARCHAR2(9)			
2	ACTIVE	NUMBER(1)			
3	MAXBOOKS	NUMBER(2)			

	Name	Type	Nullable	Default	Comments
1	PUBLISHERID	NUMBER(4)			
2	PUBLISHERNAME	VARCHAR2(50)			

	Name	Type	Nullable	Default	Comments
1	LENDINGID	INTEGER			
2	LENDINGDATE	DATE			
3	DUEDATE	DATE			
4	RETURNDATE	DATE	Y		
5	CLIENTID	VARCHAR2(9)			
6	LIBRARIANID	VARCHAR2(9)			
7	COPYID	NUMBER(8)			

	Name	Type	Nullable	Default	Comments
1	COPYID	NUMBER(8)			
2	AVAILABLE	NUMBER(1)			
3	YEARPUBLISHED	NUMBER(4)			
4	EDITION	NUMBER(3)			
5	BOOKID	NUMBER(6)			

	Name	Type	Nullable	Default	Comments
1	CATEGORYID	NUMBER(3)			
2	CATEGORYNAME	VARCHAR2(50)			



Columns of BOOK_AUTHOR					
	Name	Type	Nullable	Default	Comments
1	AUTHOR	VARCHAR2(100)			
2	BOOKID	NUMBER(6)			

Columns of BOOK					
	Name	Type	Nullable	Default	Comments
1	BOOKID	NUMBER(6)			
2	TITLE	VARCHAR2(50)			
3	PAGES	NUMBER(4)			
4	BOOKLANGUAGE	VARCHAR2(50)			
5	CATEGORYID	NUMBER(3)			
6	PUBLISHERID	NUMBER(4)			

## הכנסת נתונים (ב3 דרכים שונות):

### הכנסה מדטא גנרטור

book\_Author-

select \* from person; sele ... book (1).sql insertTables.sql dg into bookAuthor.gd

BOOK\_AUTHOR

Owner: LAX Table: BOOK\_AUTHOR Number of records: 700

Name	Type	Size	Data	Master
AUTHOR	VARCHAR2	100	firstName '-' lastName	...
BOOKID	NUMBER	6	list(select bookid from book)	...
*				...

Definition Options Result

bookCopy -

select \* from person; sele ... book (1).sql insertTables.sql dg into bookCopy.gd

BOOKCOPY

Owner: LAX Table: BOOKCOPY Number of records: 1000

Name	Type	Size	Data	Master
COPYID	NUMBER	8	sequence(1)	...
AVAILABLE	NUMBER	1	random(0, 1)	...
YEARPUBLISHED	NUMBER	4	random(1980, 2024)	...
EDITION	NUMBER	3	random(1, 4)	...
BOOKID	NUMBER	6	list(select bookid from book)	...
*				...

Definition Options Result

lax@XE [14:27:57] 1000 records generated in 2.078 seconds

## bookLending-

Export Tables of LAX drop table book\_was\_borrow ... createTables-new.sql Data Generator

BOOKLENDING

Owner: LAX Table: BOOKLENDING Number of records: 500

Name	Type	Size	Data	Master
LENDINGID	NUMBER		Sequence(1)	...
LENDINGDATE	DATE		Random(1-1-2000, 1-1-2024)	...
DUEDATE	DATE		Random(1-1-2010, 29-5-2024)	...
RETURNDATE	DATE			...
CLIENTID	VARCHAR2	9	List(select clientId from client)	...
LIBRARIANID	VARCHAR2	9	List(select librarianId from librarian)	...
COPYID	NUMBER	8	List(select copyId from bookCopy)	...
*				...

Definition Options Result

lax@XE [11:41:14] 500 records generated in 0.219 seconds

## client-

insertTables.sql dg into client.gd select count(\*) from person

CLIENT

Owner: LAX Table: CLIENT Number of records: 500

Name	Type	Size	Data	Master
CLIENTID	VARCHAR2	9	List(select personId from person)	...
ACTIVE	NUMBER	1	Random(0, 1)	...
MAXBOOKS	NUMBER	2	Random(1, 12)	...
*				...

Definition Options Result

lax@XE [5:57:37] 500 records generated in 84.594 seconds

librarian-

insertTables.sql dg into lending.gd dg into book\_was\_borrowed.gd select \* from librarian; s ... dg into librarian.gd

LIBRARIAN

Owner: LAX Table: LIBRARIAN Number of records: 500

Name	Type	Size	Data	Master
LIBRARIANID	VARCHAR2	9	List(select personid from person)	...
STARTDATE	DATE		Random(1/1/2000, 1/5/2024)	...
USERNAME	VARCHAR2	50	FirstName+[111]	...
USERPASSWORD	VARCHAR2	50	[#####]	...
*				...

Definition Options Result

person-

insertTables.sql Data Generator select count(\*) from person

PERSON

Owner: LAX Table: PERSON Number of records: 1000

Name	Type	Size	Data	Master
PERSONID	VARCHAR2	9	[000000000]	...
FIRSTNAME	VARCHAR2	50	FirstName	...
LASTNAME	VARCHAR2	50	LastName	...
PHONE	VARCHAR2	12	'05'[0]'-'[1100000]	...
EMAIL	VARCHAR2	255	email	...
*				...

Definition Options Result

lax@XE [5:46:29] 1000 records generated in 0.094 seconds

## playCopy-

insertTables.sql select \* from play Data Generator X

PLAYCOPY

Owner: LAX Table: PLAYCOPY Number of records: 800

Name	Type	Size	Data	Master
COPYID	NUMBER	8	Sequence(1)	...
AVAILABLE	NUMBER	1	Random(0, 1)	...
PLAYID	NUMBER	6	List(select playId from play)	...
*				...

Definition Options Result

## playLending-

Export Tables of LAX drop table book\_was\_borrow ... createTables-new.sql dg into playLending.gd X dg into book

PLAYLENDING

Owner: LAX Table: PLAYLENDING Number of records: 500

Name	Type	Size	Data	Master
LENDINGID	NUMBER		Sequence(1)	...
LENDINGDATE	DATE		Random(1-1-2000, 1-1-2024)	...
DUE DATE	DATE		Random(1-1-2010, 29-5-2024)	...
RETURNDATE	DATE			...
COPYID	NUMBER	8	List(select copyId from playCopy)	...
CLIENTID	VARCHAR2	9	List(select clientId from client)	...
LIBRARIANID	VARCHAR2	9	List(select librarianId from librarian)	...
*				...

Definition Options Result

lax@XE [11:50:12] 500 records generated in 0.062 seconds

## הכנסה מאקסל

bookCategory-

insertTables.sql Data Generator select \* from bookcategory ODBC Importer X

Data from ODBC Data to Oracle

**General**

Owner: [ ] Table: BOOKCATEGORY

Commit every...: 100

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

Initializing Script: [ ]

Finalizing Script: [ ]

**Fields**

category\_num -> CATEGORYID  
category\_name -> CATEGORYNAME

Field: CATEGORYID  
Fieldtype: Number

Create SQL

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

**Result Preview**

category_num	category_name
1.0	Action

Import Import to Script Close lax@XE [22:35:36] 33 records imported in 16.5 seconds Help

bookPublisher-

insertTables.sql dg into librarian.gd select personId from person ODBC Importer X

Data from ODBC Data to Oracle

**General**

Owner: [ ] Table: BOOKPUBLISHER

Commit every...: 100

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

Initializing Script: [ ]

Finalizing Script: [ ]

**Fields**

publishing-name -> PUBLISHERNAME  
publishing-number -> PUBLISHERID

Field: PUBLISHERID  
Fieldtype: Number

Create SQL

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

**Result Preview**

publishing-name	publishing-number
Macmillan	1.0
Cambridge University Press	2.0

Import Import to Script Close lax@XE [13:09:28] 32 records imported in 0.078 seconds Help

## הכנסה מהאתר Mockaroo

book-

Need some mock data to test your app? Mockaroo lets you generate up to 1,000 rows of realistic test data in CSV, JSON, SQL, and Excel formats.

Need more data? Plans start at just \$60/year. Mockaroo is also available as a [docker image](#) that you can deploy in your own private cloud.

Field Name	Type	Options
bookId	Sequence	start at: 1 step: 1 repeat: 1 restart at: blank: 0 % Σ ×
title	Movie Title	blank: 0 % Σ ×
pages	Number	min: 1 max: 1000 decimals: 0 blank: 0 % Σ ×
bookLanguage	Custom List	Hebrew, Arabic, Russian, Yiddish, English, Spanish, French random blank: 0 % Σ ×
categoryId	Number	min: 1 max: 33 decimals: 0 blank: 0 % Σ ×
publisherId	Number	min: 1 max: 32 decimals: 0 blank: 0 % Σ ×

+ ADD ANOTHER FIELD GENERATE FIELDS USING AI...

# Rows: 500 Format: SQL Table Name: book ☐ include CREATE TABLE

GENERATE DATA PREVIEW SAVE AS... DERIVE FROM EXAMPLE... MORE

play-

Field Name Type Options

playId	Sequence	start at: 1 step: 1 repeat: 1 restart at: blank: 0 % Σ ×
playName	Dataset Column	game names play name sequential blank: 0 % Σ ×
categoryId	Number	min: 1 max: 5 decimals: 0 blank: 0 % Σ ×
publisherId	Number	min: 1 max: 6 decimals: 0 blank: 0 % Σ ×

+ ADD ANOTHER FIELD GENERATE FIELDS USING AI...

# Rows: 400 Format: SQL Table Name: play ☐ include CREATE TABLE

Append Dataset: choose a dataset...

Follow @mockarodev

GENERATE DATA PREVIEW SAVE AS... DERIVE FROM EXAMPLE... MORE

## צילום מסך של גיבוי:

The screenshot shows the Mockaroo website interface for generating data. The browser address bar shows 'mockaroo.com'. The main area has a table with columns: Field Name, Type, and Options.

Field Name	Type	Options
playId	Sequence	start at: 1 step: 1 repeat: 1 restart at: blank: 0 %
playName	Dataset Column	game names play name sequential blank: 0 %
categoryId	Number	min: 1 max: 5 decimals: 0 blank: 0 %
publisherId	Number	min: 1 max: 6 decimals: 0 blank: 0 %

Below the table are buttons: '+ ADD ANOTHER FIELD' and 'GENERATE FIELDS USING AI...'. Further down, there are settings for '# Rows: 400', 'Format: SQL', 'Table Name: play', and a checkbox 'Include CREATE TABLE'. At the bottom, there is a 'Append Dataset:' dropdown menu and a 'Follow @mockarodev' link.

At the very bottom, there is a green bar with buttons: 'GENERATE DATA', 'PREVIEW', 'SAVE AS...', 'DERIVE FROM EXAMPLE...', and 'MORE'.

## צילום מסך של שחזור:

The screenshot shows the Oracle SQL\*Plus Import utility window. The title bar includes tabs for 'insertTables.sql', 'dropTables.sql', and 'Import Tables'. The main window has tabs for 'Oracle Import', 'SQL Inserts', 'PL/SQL Developer', and 'Log'. The 'Oracle Import' tab is active.

Options:

- ☐ Use Command Window
- ☒ Use SQL\*Plus

SQL\*Plus Executable:

C:\app\Chaya\product\18.0.0\dbhomeXE\bin\sqlplus.exe

Import file:

C:\Users\זה\משתמש\Documents\chaya\מניי פרויקט בבסיסי נתונים\backup\backup 29-05-2024.sql

Buttons: Import

Status bar: lax@XE Executing Oracle SQL\*Plus Utility, please wait... Done