

בז"ד לך!"

שם קורס: מיני פרויקט - בסיסי נתונים  
מספר קורס: 150225.3.5783  
מרצה: אליעזר גנסבורגר  
המגיישים: דוד ברגר 341441053, אריאל צוקרמן 209948553

# יחסים ציבור לחברה המשווקת משקאות קלים

אפשר לראות את כל קבצי SQL ותרשיים בקישור:

<https://github.com/David-YY-Berger/Database-Project-5783>

## שלב 1: הקמת בסיס הנתונים

### תיאור החברה ומטרת הפרויקט:

החברה משווקת משקאות שונים לציבור הרחב. עובdot החברה מורכבת מהרבה צוותים, ובפרויקט זהה אנחנו נתכנים ניהול לחלק מבסיס הנתונים שיכיל את כל המידע הקשור ליח"צ.

מרכז בסיס הנתונים שלנו הוא **פריטות** שדרכים החברה משווקת את עצמה לציבור הרחב. פרסום יכול להיות אמר, או פרסום מבוססת תמונות, כל אחד עם הפרטים מסוימים. בסיס הנתונים שלנו מכיל את הפרטים הרלוונטיים של **עובד בחברה** שמייצרים את הפרסומות ("Writer"), לדוגמה - גובה המשכורת שלהם, ושות ניסיון. והוא גם מכיל את הפרטים של **פלטפורמות** שבהם החברה מוציאה פרסום. פלטפורמה יכולה להיות אתר, או עיתון. קיימת בכל פלטפורמה דרך למדוד את חשיבות הפלטפורמה (עיתון - מספר עותקים שמדפסים כל היום, ואיתר - מספר הגולשים בכל יום). וכמוון קיימים בסיסי נתונים טבלאות **הmarkerות** בין שלושת היחסיות (AppearsOn ו-WrittenBy).

הירושה שבה השתמשנו תומכת ביצירת שני בנימ שקשורים לאוטו אבא. לדוגמה - "תכן רשומה אחת של פלטפורמה ושני בנימ (גם אתר וגם עיתון) שקשורים אותה פלטפורמה. זה כדי לאפשר מצב שיש חברה אחת (לדוגמה ערוץ 7) שיש להם גם עיתון וגם אתר.

## תיאור הישיות הקיימות במערכת:

### 1. מחבר (Writer)

הטבלה מכילה את כל עובדי החברה שמייצרים פרסומות (או בכתב או בגלופות).  
הטבלה מכילה:

- מס' זיהוי (writerId) - [מפתח - Primary Key]
- שם (writerName)
- שנות ניסיון (numYearsOfExperience)
- משכורת לפי שעה (PayPerHour)

### 2. פרסום (Advertisement)

הטבלה מכילה את כל הפרסומות הקיימות במערכת. כל רשומה של הבנים (PictureAd ו Article) (PictureAd Article) תסתמך על רשומה כאן בטבלת פרסום.  
הטבלה מכילה:

- מס' זיהוי (adId) - [מפתח - Primary Key]
- מס' שעות הנדרש לכתיבה (hoursToWrite)
- שם הפרסומת (adName)

### 3. פרסום מסווג תמונה (PictureAd)

הישות יורשת מפרסומת, ומסתמכת על המספר זיהוי הנמצא בטבלת פרסום.  
הטבלה מכילה:

- מס' זיהוי (adId) - [מפתח זר - Foreign Key]
- אורך/גובה של הפרסומת בס"מ (length)
- רוחב של הפרסומת בס"מ (width)

### 4. מאמר (Article)

הישות יורשת מפרסומת, ומסתמכת על המספר זיהוי הנמצא בטבלת פרסום.  
הטבלה מכילה:

- מס' זיהוי (adId) - [מפתח זר - Foreign Key]
- מספר המילים שיש במאמר (numWords)

### 5. פלטפורמה (Platform)

הטבלה מכילה את כל הפלטפורמות הקיימות במערכת. כל רשומה של הבנים (Newspaper ו Website) (Newspaper Website) תסתמך על רשומה כאן בטבלת פלטפורמה.  
הטבלה מכילה:

- שם הפלטפורמה (platformName) - [מפתח - Primary Key]

### 6. אתר (Website)

הישות יורשת מפלטפורמה, ומסתמכת על שם הפלטפורמה (בטבלת פלטפורמה) בתור מפתח.  
הטבלה מכילה:

- שם הפלטפורמה (platformName) - [מפתח זר - Foreign Key]
- כתובת האתר (url)
- מספר הצופים/מבקרים בכל יום (numViewersDaily)

. 7. **עיתון** (Newspaper).

הישות יורשת מפלטפורמה, ומסתמכת על שם הפלטפורמה (בטבלת פלטפורמה) בתור מפתח. הטבלה מכילה:

- שם הפלטפורמה (platformName) - [מפתח זר -]
- מספר עותקים שמדפסים כל יום (numCopiesPrinted)

**תאור הקשרים בין הישויות:**

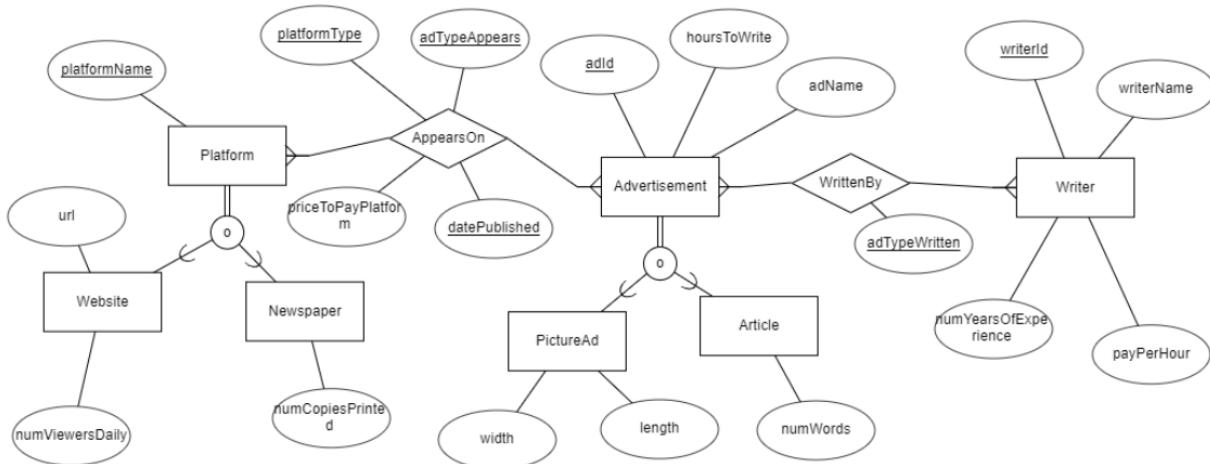
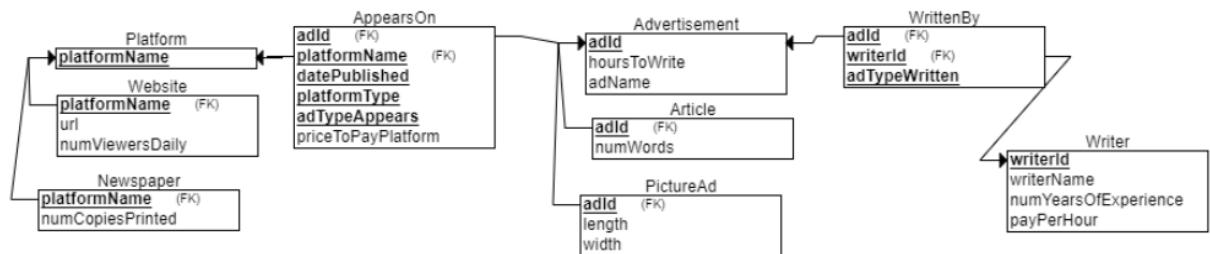
1. **הקשר מופיע ב-** (AppearsOn) הוא M:M:

- יתכן הרבה פרסומות קשורות להרבה פלטפורמות (כגון אותה פרסום שמוינעה בכמה פלטפורמות שונות, וכן כמה פרסומות שונות שמוינעות באותה פלטפורמה).  
בנוסף לשני המפתחות (platformName, adid, -[Primary Key - datePublished] מבדיל בין הופעות שונות ביום אחד  
  - תאריך פרסום (datePublished) - [מפתח -]
  - סוג הפלטפורמה בה הפרסומת מופיע (platformType) [מפתח -]
  - יכול להיות תמונה או עיתון או אתר אינטרנט.
  - סוג הפרסומת שמוינעה (adType) - [Primary Key -]
  - יכול להיות תמונה או כתבה.
  - המחיר להציג את הפרסומת בפלטפורמה (priceToPayPlatform)

2. **הקשר נכתב ע"י** (WrittenBy) הוא M:M:

יתכן הרבה פרסומות נכתבים ע"י הרבה מחברים (כגון: שני מחברים שכתו מאמר ביחד, וכן מחבר שכותב כמה מאמרים)  
יש שני המפתחות (pid, adid, -[writerId] לקשר.  
ובנוסף:

- סוג הפרסומת אותה הוא כותב (adTypeWritten) - [Primary Key -]
- יכול להיות תמונה או אתר אינטרנט.

תרשים ERDתרשים DSD

יצירת טבלאות:

## מידע התחלתי

Writer					
writerId	1-100	ודרתית			
payperHour	30-150	Rndomly			
numYearsOf Experience	3-15	Rndomly			

Advertisement			Platform		
AdId		PlatformName	Newspaper	Website	
1-50	Only pictureAd	Besheva	v	v	
51-150	Picture AND Article	Israel_hayom	X	v	
151-200	Only article	Mishpacha	v	X	
		Yediot_acharonot	v	v	
		Yated_neeman	v	v	

	WrittenBy			
	amount:	adTypeWritten	AdId	Writer id
	200	PictureType or ArticleType (random)	51-150 random	1-100 random
	50	pictureType	1-50 random	1-100 random
	50	articleType	151-200	1-100

			random	random
	300	ס"כ		

AppearsOn						
amount:	platformType	Date	adTypeAppears	adid		
100	newspapers	Random(1.1.2000, 1.1.2023)	PictureType or ArticleType (random)	51-150 (random)		
100	websites		PictureType or ArticleType (random)	51-150 (random)		
100	Website or newspaper (random)		pictureType	1-50		
100	Website or newspaper (random)		articleType	151-200		
400					ס"כ	

### הכנסת נתונים בשיטות שונות:

\*כל הקבצים נמצאים בקישור גיטהאב בתחילת המסמך \*\*

#### (1) הכנסת נתונים ע"י PL/SQL Developer Data Generator בתוכנה

הכנסת הנתונים בטבלה writtenby נעשתה ב-3 שלבים

1. נתונים 1-50: פרטומות של תמונות
2. נתונים 51-150: תמונות ומאמרים משולב
3. נתונים 151-200: פרטומות של מאמרים

The screenshot shows the PL/SQL Developer Data Generator interface. The top bar has the title 'WRITTENBY'. Below it, there's a configuration panel with fields: 'Owner' set to 'SYSTEM', 'Table' set to 'WRITTENBY', and 'Number of records' set to '200'. Below this is a data grid with four columns: 'Name', 'Type', 'Size', and 'Data'. It contains three rows of data:

Name	Type	Size	Data
ADTYPEWRITTEN	VARCHAR2	30	List('PictureType', 'ArticleType')
ADID	NUMBER		Random(51, 150)
WRITERID	NUMBER		Random(1, 100)

This screenshot shows the same interface as above, but with a different configuration. The 'Number of records' field is now set to '50'. The data grid includes an additional column 'Master' at the far right. The first row has a checkmark in the 'Master' column. The second row has a checkmark in the 'Master' column. The third row has an asterisk (\*) in the 'Master' column.

Name	Type	Size	Data	Master
ADTYPEWRITTEN	VARCHAR2	30	'PictureType'	...
✓ ADID	NUMBER		Random(1, 50)	...
WRITERID	NUMBER		Random(1, 100)	...
*				...

WRITTENBY				
<	Owner	Table	Number of records	
>	SYSTEM	WRITTENBY	50	
...				
	Name	Type	Size	Data
	ADTYPEWRITTEN	VARCHAR2	30	'ArticleType'
	ADID	NUMBER		Random(151, 200)
	WRITERID	NUMBER		Random(1, 100)
✓				
*				

הכנסת הנתונים בטבלה `appearson` נעשתה ב-3 שלבים

4. נתונים 1-50: פרסוםות של תמונות שפורסמו בעיתון או באתר
5. נתונים 51-150: פרסוםות של תמונות ומאמרים משולב שפורסמו בעיתון וכן תמונות ומאמרים משולב שפורסמו באתר
6. נתונים 151-200: פרסוםות של מאמרים שפורסמו בעיתון או באתר

APPEARON				
<	Owner	Table	Number of records	
>	SYSTEM	APPEARON	10..20	
...				
	Name	Type	Size	Data
	DATEPUBLISHED	DATE		Random(1.1.2000, 1.1.2023)
	PRICETOPAYPLATFORM	NUMBER		Random(500, 2000)
	PLATFROMTYPE	VARCHAR2	20	List('NewspaperType', 'WebsiteType')
	ADTYPEAPPEARS	VARCHAR2	20	List('ArticleType')
	ADID	NUMBER		Random(151, 200)
✓	PLATFROMNAME	VARCHAR2	30	List(select platformname from website union select platformname from newspaper)
*				

APPEARON				
<	Owner	Table	Number of records	
>	SYSTEM	APPEARON	10..20	
...				
	Name	Type	Size	Data
	DATEPUBLISHED	DATE		Random(1.1.2000, 1.1.2023)
	PRICETOPAYPLATFORM	NUMBER		Random(500, 2000)
	PLATFROMTYPE	VARCHAR2	20	List('NewspaperType', 'WebsiteType')
✓	ADTYPEAPPEARS	VARCHAR2	20	List('PictureType')
	ADID	NUMBER		Random(1, 50)
*	PLATFROMNAME	VARCHAR2	30	List(select platformname from website union select platformname from newspaper)
*				

APPEARSON				
< Owner	Table	Number of records		
> SYSTEM	APPEARSON	10..20		
...				
Name	Type	Size	Data	
DATEPUBLISHED	DATE		Random(1.1.2000, 1.1.2023)	
PRICETOPAYPLATFORM	NUMBER		Random(500, 2000)	
PLATFORMTYPE	VARCHAR2	20	List('NewspaperType')	
ADTYPEAPPEARS	VARCHAR2	20	List('PictureType', 'ArticleType')	
ADID	NUMBER		Random(51,1 50)	
✓ PLATFORMNAME	VARCHAR2	30	List(select platformname from newspaper)	
*				

APPEARSON				
< Owner	Table	Number of records		
> SYSTEM	APPEARSON	10..20		
...				
Name	Type	Size	Data	
DATEPUBLISHED	DATE		Random(1.1.2000, 1.1.2023)	
PRICETOPAYPLATFORM	NUMBER		Random(500, 2000)	
PLATFORMTYPE	VARCHAR2	20	List('WebsiteType')	
ADTYPEAPPEARS	VARCHAR2	20	List('PictureType', 'ArticleType')	
ADID	NUMBER		Random(51,1 50)	
✓ PLATFORMNAME	VARCHAR2	30	List(select platformname from website)	
*				

:SQL - create, insert - קבצי (2)

The screenshot shows a SQL editor interface with the following details:

- Tab Bar:** Contains two open tabs: "CreateTables.sql" and "ViewAllTables.sql". Below the tabs are three buttons: "SQL", "Output", and "Statistics".
- CreateTables.sql Content:** Displays the following SQL code for creating four tables: Advertisement, PictureAd, Article, and Platform.

```
CREATE TABLE Advertisement
(
    adId INT NOT NULL,
    hoursToWrite INT NOT NULL,
    adName VARCHAR(40) NOT NULL,
    PRIMARY KEY (adId)
);

CREATE TABLE PictureAd
(
    length INT NOT NULL,
    width INT NOT NULL,
    adId INT NOT NULL,
    PRIMARY KEY (adId),
    FOREIGN KEY (adId) REFERENCES Advertisement(adId)
);

CREATE TABLE Article
(
    numWords INT NOT NULL,
    adId INT NOT NULL,
    PRIMARY KEY (adId),
    FOREIGN KEY (adId) REFERENCES Advertisement(adId)
);

CREATE TABLE Platform
(
    platformName VARCHAR(30) NOT NULL,
    PRIMARY KEY (platformName)
);
```

The screenshot shows a SQL editor interface with three tabs at the top: "CreateTables.sql", "ViewAllTables.sql", and the currently selected "SQL". The SQL tab contains the following code:

```
CREATE TABLE Writer
(
    writerId INT NOT NULL,
    writerName VARCHAR(30) NOT NULL,
    numYearsOfExperience INT NOT NULL,
    payPerHour INT NOT NULL,
    PRIMARY KEY (writerId)
);

CREATE TABLE AppearsOn
(
    datePublished DATE NOT NULL,
    priceToPayPlatform INT NOT NULL,
    platformType VARCHAR(20) NOT NULL,
    adTypeAppears VARCHAR(20) NOT NULL,
    adId INT NOT NULL,
    platformName VARCHAR(30) NOT NULL,
    PRIMARY KEY (datePublished, platformType, adTypeAppears, adId, platformName),
    FOREIGN KEY (adId) REFERENCES Advertisement(adId),
    FOREIGN KEY (platformName) REFERENCES Platform(platformName)
);

CREATE TABLE WrittenBy
(
    adTypeWritten VARCHAR(20) NOT NULL,
    adId INT NOT NULL,
    writerId INT NOT NULL,
    PRIMARY KEY (adTypeWritten, adId, writerId),
    FOREIGN KEY (adId) REFERENCES Advertisement(adId),
    FOREIGN KEY (writerId) REFERENCES Writer(writerId)
);
```

```
insert into SYSTEM.PLATFORM (PLATFORMNAME)
values ('Besheva');

insert into SYSTEM.PLATFORM (PLATFORMNAME)
values ('Yated_neeman');

insert into SYSTEM.PLATFORM (PLATFORMNAME)
values ('Israel_Hayom');

insert into SYSTEM.PLATFORM (PLATFORMNAME)
values ('Mishpacha');

insert into SYSTEM.PLATFORM (PLATFORMNAME)
values ('Yediot_acharonot');

commit;
```

(3) הכנסת נתונים ע"י קובץ CVS (המיוצר ב-<https://www.mockaroo.com>)

פעולה PL/SQL Developer IMPORT

**Data from Textfile** **Data to Oracle**

**File Data**

```
writerId,writerName,numYearsOfExperience,payPerHour
1,Clarisse Richemont,14,94
2,Nicolea Yakebowitch,8,82
3,Kass Prestedge,4,40
4,Padraig Arrigucci,7,148
5,Mehetabel Cathro,12,99
6,Kerry Skacel,8,66
7,Adrea Greenhead,15,46
```

**Configuration**

**General**

- Fieldcount: 4
- Quote character: "
- Comment line:
- Import lines: 1 ..
- End at line-end
- Name in header
- Skip empty lines

Field Start

- Relative position
- Absolute position
- Character

Field End

- Length
- Character

**Filter**

**Result Preview**

writerId	writerName	numYearsOfExperience	payPerHour
1	Clarisse Richemont	14	94
2	Nicolea Yakebowitch	8	82
3	Kass Prestedge	4	40
4	Padraig Arrigucci	7	148
5	Mhetabel Cathro	12	99
6	Kerry Skacel	8	66

וידוי אקלוס נתוניים:

ע"פ השאלה:

ניתן לראות שכל היטבלאות מלאות:

SQL   Output   Statistics

```
-- Use the tabs below to see each one!

select * from Writer;
select * from WrittenBy;
select * from advertisement;
select * from article;
select * from picturead;
select * from appearson;
select * from platform;
select * from newspaper;
select * from website;
```

Select writer   Select writtenby   Select advertisement   Select article   Select picturead   Select appearson   Select platform   Select newspaper   Select website

The screenshot shows a database interface with a SQL editor at the top containing sample queries. Below is a table titled 'Select writer' with the following data:

WRITERID	WRITERNAME	NUMYEARSOFEXPERIENCE	PAYPERHOUR
87	Jesse Winans	...	115
88	Chi Adams	...	116
89	Elias Young	...	98
90	Derrick Watley	...	103
91	Bob Ticotin	...	58
92	Sophie Kinney	...	138
93	Elizabeth Spacek	...	93
94	Cole Del Toro	...	66
95	Russell Hayek	...	120
96	Rosario Baranski	...	101
97	Sissy Turner	...	30
98	Jesse Tomei	...	37
99	Illeana Paquin	...	91
100	Hank Levert	...	58

## שלב 2: טיפול נתונים - שאלות

שאלות בסיסות:

לפני השינוי:

The screenshot shows a SQL Server Management Studio interface. At the top, there are two tabs: 'ViewAllTables.sql' and 'select \* from advertisement ...'. Below them is a toolbar with icons for copy, lock, insert, delete, and others. The main area contains a query window with the following SQL code:

```
select * from advertisement
where adid = '13';
```

Below the query window is a results grid with the following data:

	ADID	HOURSTOWRITE	ADNAME
▶	1	13	32 Fanta1 ...

אחרי השינוי:

The screenshot shows a SQL Server Management Studio interface. At the top, there are two tabs: 'ViewAllTables.sql' and 'UPDATE advertisement SET a ...'. Below them is a toolbar with icons for copy, lock, insert, delete, and others. The main area contains a query window with the following SQL code:

```
UPDATE advertisement
SET adName = 'KolHatorahKulo'
WHERE adid = 13;
select *
from advertisement
where adid = '13';
```

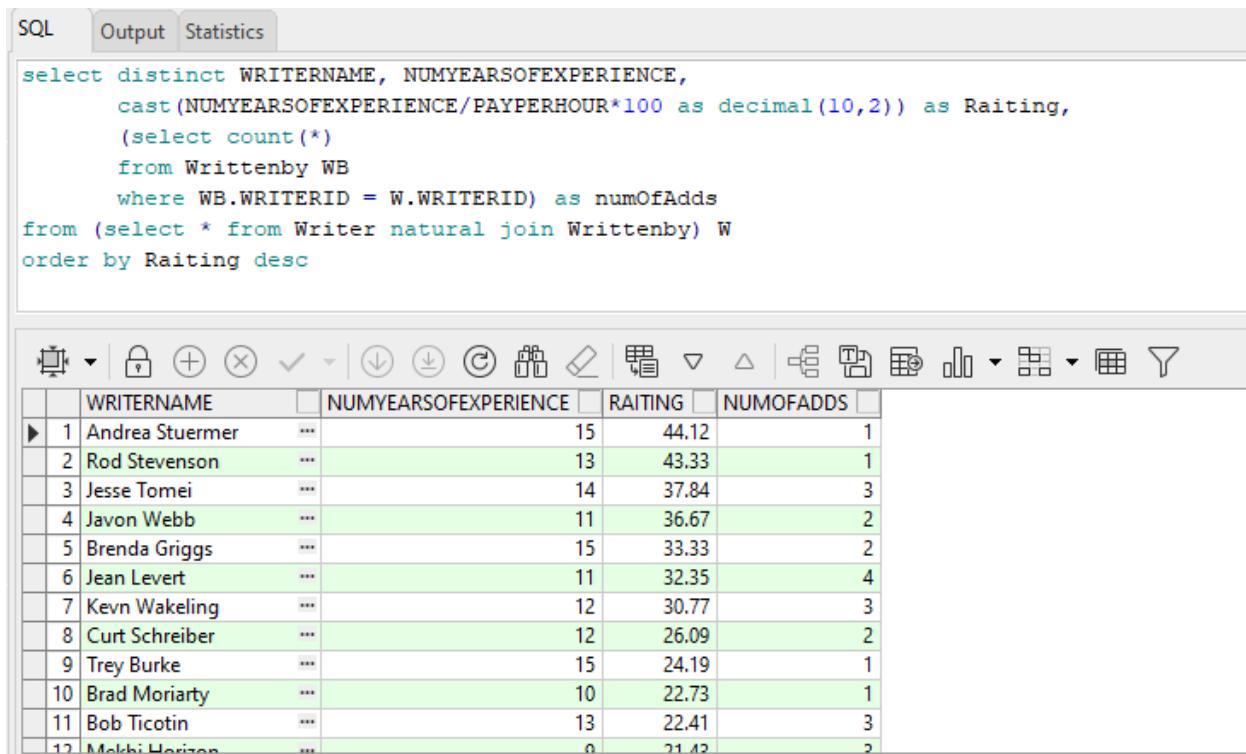
Below the query window is a results grid with the following data:

	ADID	HOURSTOWRITE	ADNAME
▶	1	13	32 KolHatorahKulo ...

**(1) המחברים שהci שווה להעסיק**

שאילתא מחייב טבלה של שמות המחברים, שנות ניסיון, ושדה נוסף בשם "RATING" של כל המחברים, ממוינים (בסדר יורד) לפי "RATING".  
 (השדה RATING הוא YearsOfExperience/PayPerHour מוכפל ב-100 ומוצג כמספר ברור).

משמעות העסקית: השאלה מציגה רשימה ממוינת של המחברים שכאורה hei טובים, שעולים hei פחות.



The screenshot shows the SQL tab of the SSMS interface. The query window contains the following SQL code:

```

select distinct WRITERNAME, NUMYEARSOFEXPERIENCE,
       cast(NUMYEARSOFEXPERIENCE/PAYPERHOUR*100 as decimal(10,2)) as Raiting,
       (select count(*)
        from Writtenby WB
       where WB.WRITERID = W.WRITERID) as numOfAdds
  from (select * from Writer natural join Writtenby) W
 order by Raiting desc
  
```

The results grid displays 12 rows of data:

	WRITERNAME	NUMYEARSOFEXPERIENCE	RAITING	NUMOFAdds
1	Andrea Stuermer	15	44.12	1
2	Rod Stevenson	13	43.33	1
3	Jesse Tomei	14	37.84	3
4	Javon Webb	11	36.67	2
5	Brenda Griggs	15	33.33	2
6	Jean Levert	11	32.35	4
7	Kevn Wakeling	12	30.77	3
8	Curt Schreiber	12	26.09	2
9	Trey Burke	15	24.19	1
10	Brad Moriarty	10	22.73	1
11	Bob Ticotin	13	22.41	3
12	Melchi Horizon	0	21.12	2

**(2) הפרסומות כי טובות לפרסום**

השאילתה מחזירה טבלה של שמות הפרסומות, ושעות שלקח לכתוב אותם, וشنות ניסיון של המחבר של כל הפרסומות שנכתבו בפועל ע"י מחבר, וממיינים (בסדר יורד) לפי השעות המושקעות בהם

משמעות העסקית: השאלה מציגה רשימה ממוחשבת של פרסומות שכבר השקיעו בהן כי הרבה זמן ע"י המחברים המנוסים ביותר.

	ADNAME	NUMYEARSOFEXPERIENCE	HOURSTOWRITE
1	Fanta1	15	37
2	Kinley2	15	30
3	Fanta1	15	30
4	Fanta2	15	29
5	Kinley2	15	26
6	Kinley2	15	26
7	Fanta2	15	24
8	Sprite2	15	23
9	Coca_cola1	15	22
10	Fanta2	15	20
11	Fanta2	15	20
12	Fanta2	15	17
13	Sprite1	15	16

## (3) העיתונים שבהם הכי יעיל להשתמש

השאילתת מציגה עבור כל עיתון (ובכל עיתון עבור כל סוג פרסום - תמונה או כתבה) את מספר הפריטות שפורסמו בו, את ממוצע המחיר של כל פרסום, כמהו עותקים של פרסום כל שקל בודד, ממוצע המחיר של פרסום, מספר העותקים של פרסום. התוצאה מוצגת בסדר ממויין לפי מספר פרסום עבור שקל מהגבוה לנמוך.

משמעות עסקית: השאילתת מציגה את העיתונים הכי "חזקים" (שמקבלים הכי הרבה עותקים לפי שקל).

```
CREATE VIEW platANDapp AS
SELECT
    PLATFORMNAME, adtypeappears, platformType, COUNT(*) AS totalAppearances, AVG(AP.PRICETOPAYPLATFORM) AS AvgPricePerAd
FROM
    platform P NATURAL JOIN Appearson AP
GROUP BY
    adtypeappears, platformType, platformName
ORDER BY
    AvgPricePerAd DESC, totalAppearances DESC;

SELECT
    platANDapp.PLATFORMNAME, platANDapp.adtypeappears, platANDapp.totalAppearances,
    CAST(n.numcopiesprinted / platANDapp.AvgPricePerAd AS DECIMAL(10,2)) AS CopyPerShekel,
    platANDapp.AvgPricePerAd, n.numcopiesprinted
FROM
    platANDapp, newspaper n
WHERE
    platANDapp.PLATFORMNAME = n.platformname AND (platANDapp.platformType = 'NewspaperType')
ORDER BY
    CopyPerShekel DESC;
```

	PLATFORMNAME	ADTYPEAPPEARS	TOTALAPPEARANCES	COPYPERSHEKEL	AVGPRICEPERAD	NUMCOPIESPRINTED
1	Besheva	PictureType	41	3.98	1190.68292682927	4736
2	Besheva	ArticleType	30	3.94	1203.4	4736
3	Mishpacha	PictureType	10	3.07	1249.8	3836
4	Mishpacha	ArticleType	10	2.77	1386	3836
5	Yediot_acharonot	PictureType	31	1.74	1257.1935483871	2185
6	Yediot_acharonot	ArticleType	33	1.64	1328.666666666667	2185
7	Yated_neeman	ArticleType	29	1.51	1261.06896551724	1898
8	Yated_neeman	PictureType	26	1.32	1433.23076923077	1898

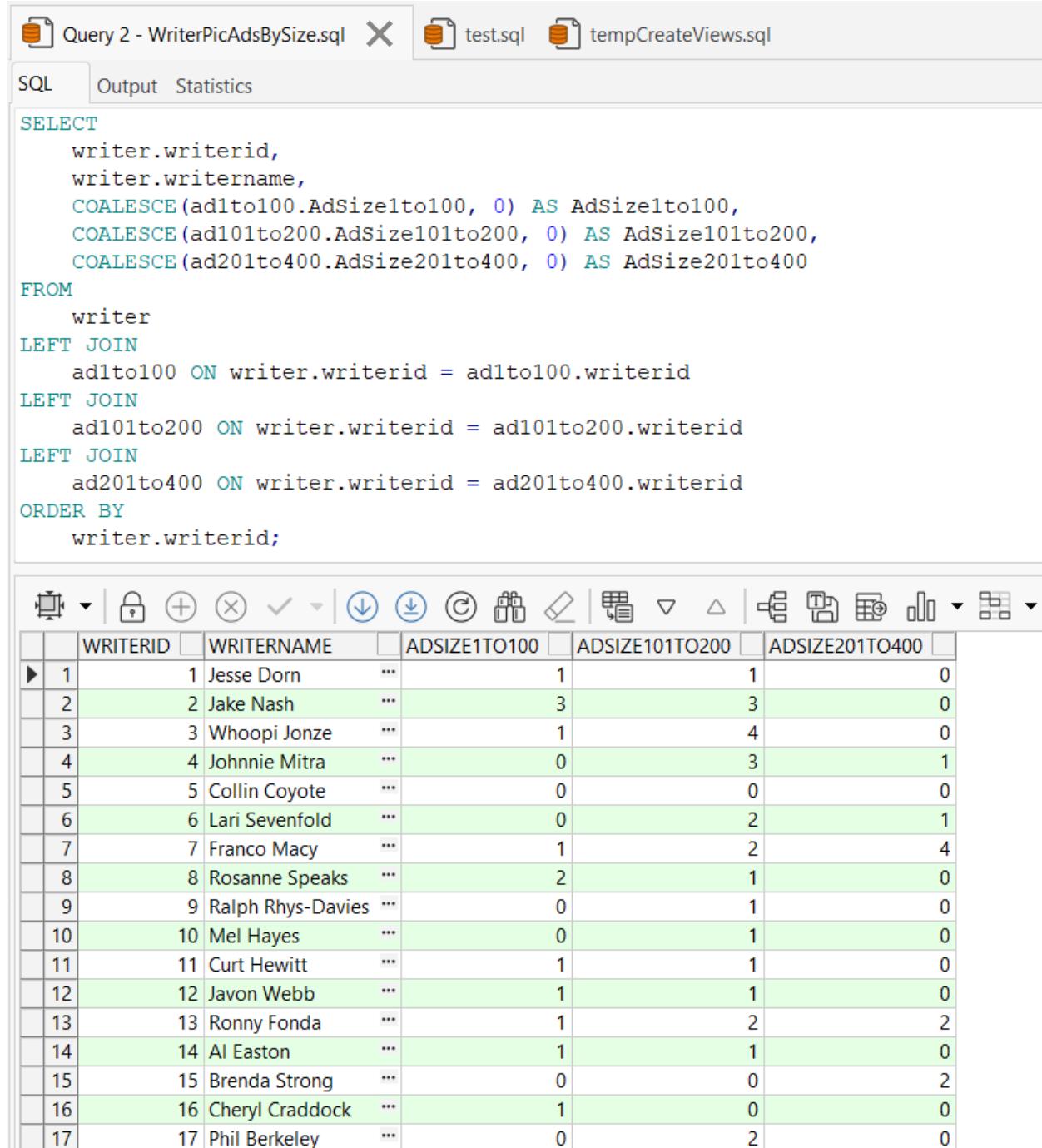
**(4) פירוט ניסיון המחברים לפי גודל פרסוםת תמונה**

השאילתה מציגה טבלה של מחברים ושמותיהם, ושלוש עמודות של מספרי פרסוםת תמונה שכתבו (בשלוש קטגוריות לפי גודל התמונה - 1 (1-100, 2 (101-200, 3 (201-400)).

המשמעות העסקי: לראות את הנטיון של כל כתבים בכל אחד מסוגי הגודלים של הפרסומות.

```
create view adlto100 as
select writerid, count(adid) as AdSize1to100
from (writtenBy natural join (advertisement natural join Picturead))
where (length * width) <= 100 AND (length * width) > 0
group by writerid
order by writerid;
;
create view ad101to200 as
select writerid, count(adid) as AdSize101to200
from (writtenBy natural join (advertisement natural join Picturead))
where (length * width) <= 200 AND (length * width) > 101
group by writerid
order by writerid
;
create view ad201to400 as
select writerid, count(adid) as AdSize201to400
from (writtenBy natural join (advertisement natural join Picturead))
where (length * width) <= 400 AND (length * width) > 201
group by writerid
order by writerid
;
```

הפעולה **COALESCE** היא גורמת לכך שככל המיקומות בהם-Amor להופיע בטבלה NULL איז במקומות שאותה שורה שיש בה NULL לא תופיע כלל אז היא כן תופיע ויופיע 0 במקום NULL וכן תופיע בטבלה הסופית.



```

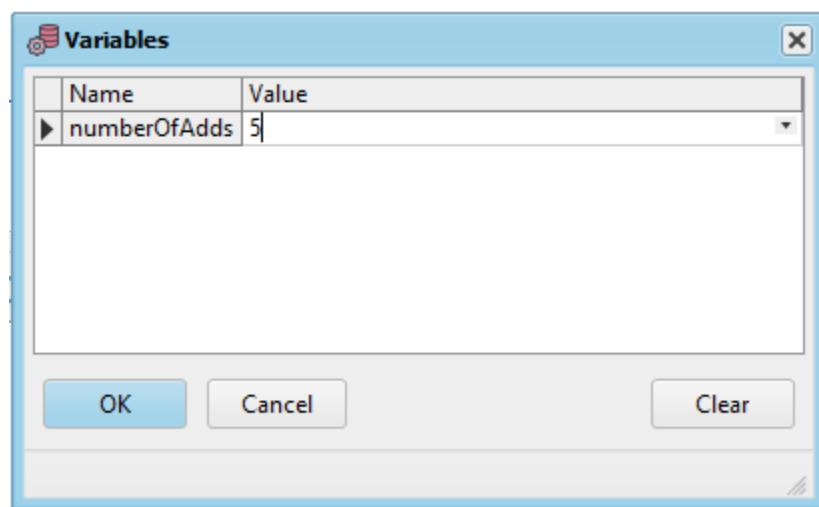
SELECT
    writer.writerid,
    writer.writername,
    COALESCE(ad1to100.AdSize1to100, 0) AS AdSize1to100,
    COALESCE(ad101to200.AdSize101to200, 0) AS AdSize101to200,
    COALESCE(ad201to400.AdSize201to400, 0) AS AdSize201to400
FROM
    writer
LEFT JOIN
    ad1to100 ON writer.writerid = ad1to100.writerid
LEFT JOIN
    ad101to200 ON writer.writerid = ad101to200.writerid
LEFT JOIN
    ad201to400 ON writer.writerid = ad201to400.writerid
ORDER BY
    writer.writerid;

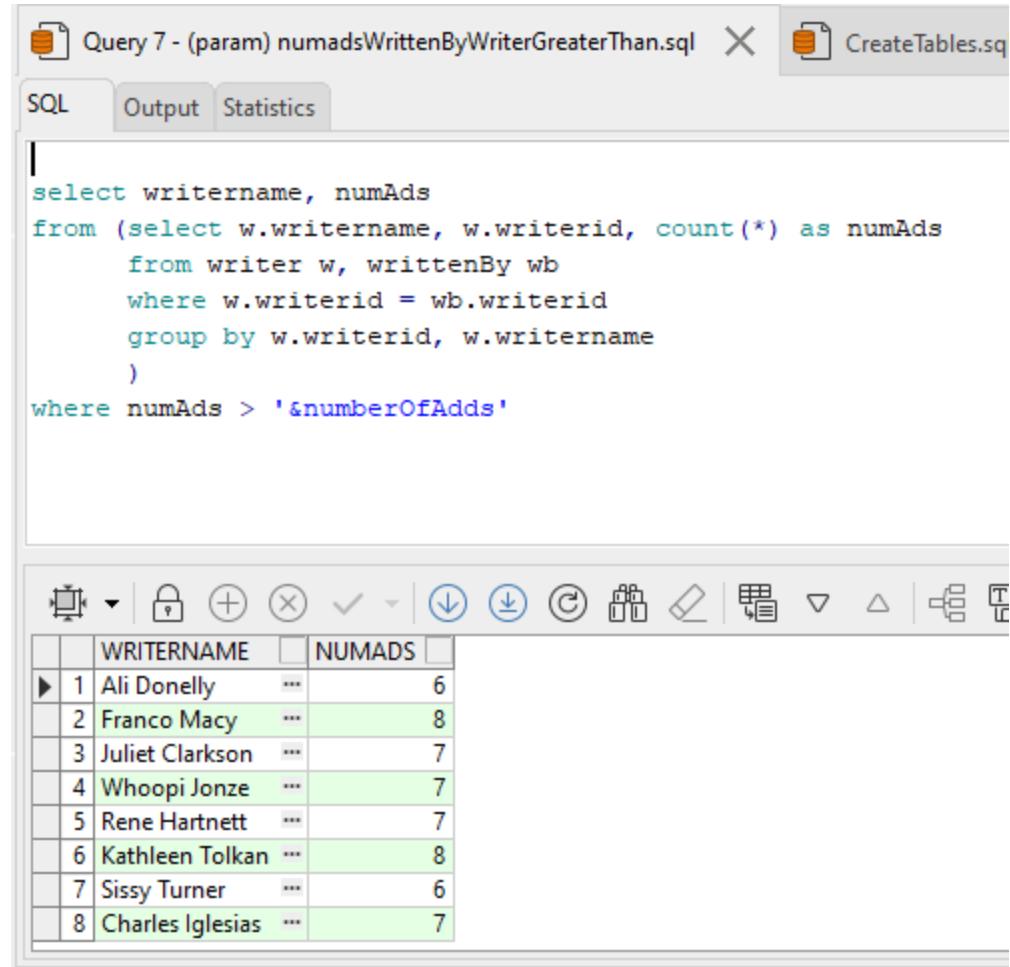
```

	WRITERID	WRITERNAME	ADSIZ1TO100	ADSIZ101TO200	ADSIZ201TO400	
►	1	Jesse Dorn	...	1	1	0
	2	Jake Nash	...	3	3	0
	3	Whoopi Jonze	...	1	4	0
	4	Johnnie Mitra	...	0	3	1
	5	Collin Coyote	...	0	0	0
	6	Lari Sevenfold	...	0	2	1
	7	Franco Macy	...	1	2	4
	8	Rosanne Speaks	...	2	1	0
	9	Ralph Rhys-Davies	...	0	1	0
	10	Mel Hayes	...	0	1	0
	11	Curt Hewitt	...	1	1	0
	12	Javon Webb	...	1	1	0
	13	Ronny Fonda	...	1	2	2
	14	Al Easton	...	1	1	0
	15	Brenda Strong	...	0	0	2
	16	Cheryl Craddock	...	1	0	0
	17	Phil Berkeley	...	0	2	0

**שאלות עם פרמטרים:**

(1) שמות ומספר הפריטות של כתבים שכתבו יותר מ X פריטות:





The screenshot shows the Oracle SQL Developer interface. At the top, there are two tabs: "Query 7 - (param) numadsWrittenByWriterGreaterThan.sql" and "CreateTables.sql". Below the tabs, there are three buttons: "SQL", "Output", and "Statistics". The "SQL" button is selected. In the main area, there is a code editor containing the following SQL query:

```

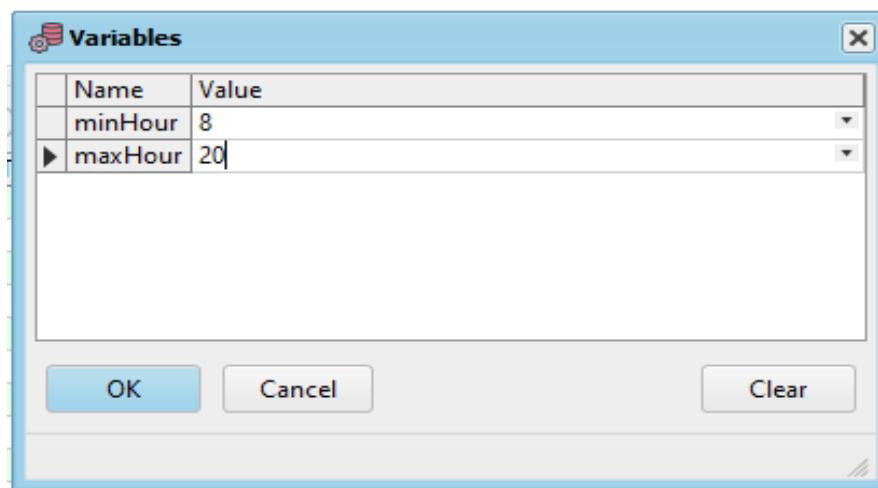
select writername, numAds
from (select w.writername, w.writerid, count(*) as numAds
      from writer w, writtenBy wb
     where w.writerid = wb.writerid
   group by w.writerid, w.writername
)
where numAds > '&numberOfAds'

```

Below the code editor is a toolbar with various icons for database operations. Underneath the toolbar is a result grid displaying the output of the query. The grid has two columns: "WRITERNAME" and "NUMADS". The data is as follows:

	WRITERNAME	NUMADS
1	Ali Donelly	6
2	Franco Macy	8
3	Juliet Clarkson	7
4	Whoopi Jonze	7
5	Rene Hartnett	7
6	Kathleen Tolkan	8
7	Sissy Turner	6
8	Charles Iglesias	7

(2) שמות הפרסומות ומספר הפרסום וזמן כתיבת הפרסום (בשעות) של פרסומות שנכתבו בין X לY  
שעות:



The screenshot shows the SQL Server Management Studio interface. At the top, there are three tabs: 'DropAllTables.sql', 'Query 8 - (param) addsWrittenBetweenNumHours', and 'Query 9 - (param)'. Below the tabs, there are three buttons: 'SQL', 'Output', and 'Statistics'. The 'SQL' button is selected.

```
select a.adid, a.adname, a.hourstowrite
from advertisement a
where a.hourstowrite>'&minHour' and a.hourstowrite<'&maxHour'
```

The results grid displays the following data:

	ADID	ADNAME	HOURSTOWRITE
44	122	Fanta2	17
45	123	Fanta1	19
46	127	Fanta2	17
47	128	Shweps2	18
48	130	Shweps2	11
49	137	Shweps1	12
50	138	Sprite1	12
51	139	Sprite2	12
52	150	Kinley1	17
53	151	Sprite1	19
54	154	Fanta1	12
55	157	Coca_cola1	14
56	161	Sprite2	14

## ייצוג הנתונים בצורות שונות

אנחנו מראים שלוש דרכים שונים להציג נתונים עם גרפים, כל גרפף עם שאלתה מתאימים שלו.

עבור שאלתא 5:

שומרה את הפרסומות עם המחיר שלהם לפי תאריך:

```
select a.datepublished, a.pricetopayplatform, a.platformtype
from appearson a
order by a.datepublished
```

	DATEPUBLISHED	PRICETOPAYPLATFORM	PLATFORMTYPE	
► 1	01/01/2000	...	949	NewspaperType ...
2	21/01/2000	...	1562	NewspaperType ...
3	08/04/2000	...	1630	Website ...
4	08/04/2000	...	1274	NewspaperType ...

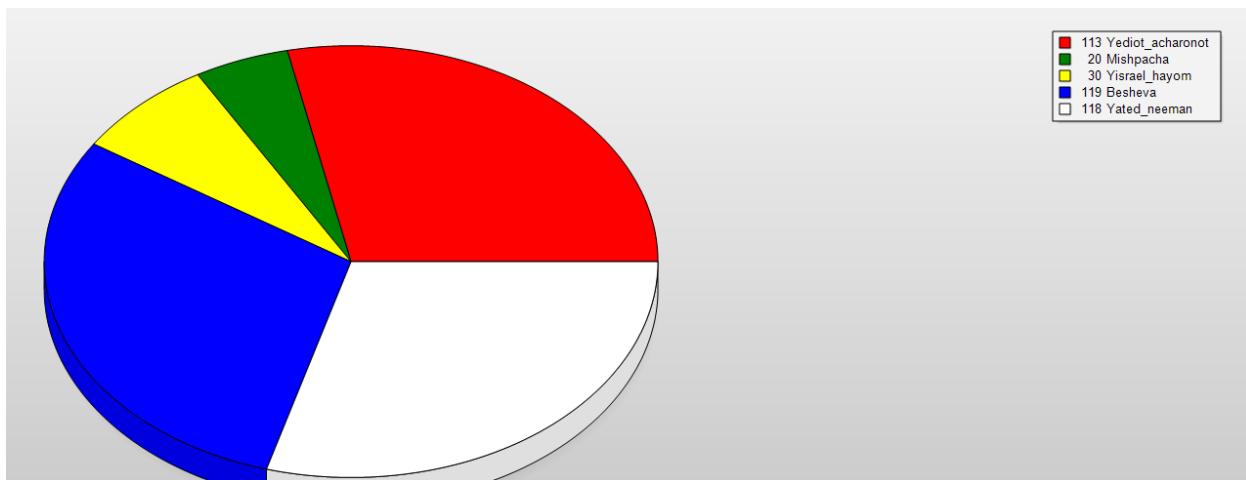
ראאים בקלות לפי הגרף שהיתה ירידת מחירים בחודש 7 ...



עבור שאלתא 6:  
המיראה את הפלטפורמה, ומספר הפרסומות הכלל:

```
select platform.platformname, count(*)
from appearsOn, platform
where appearsOn.platformname = platform.platformname
group by platform.platformname|
```

	PLATFORMNAME	COUNT(*)
► 1	Yediot_acharonot	113
2	Mishpacha	20
3	Yisrael_hayom	30
4	Besheva	119
5	Yated_neeman	118



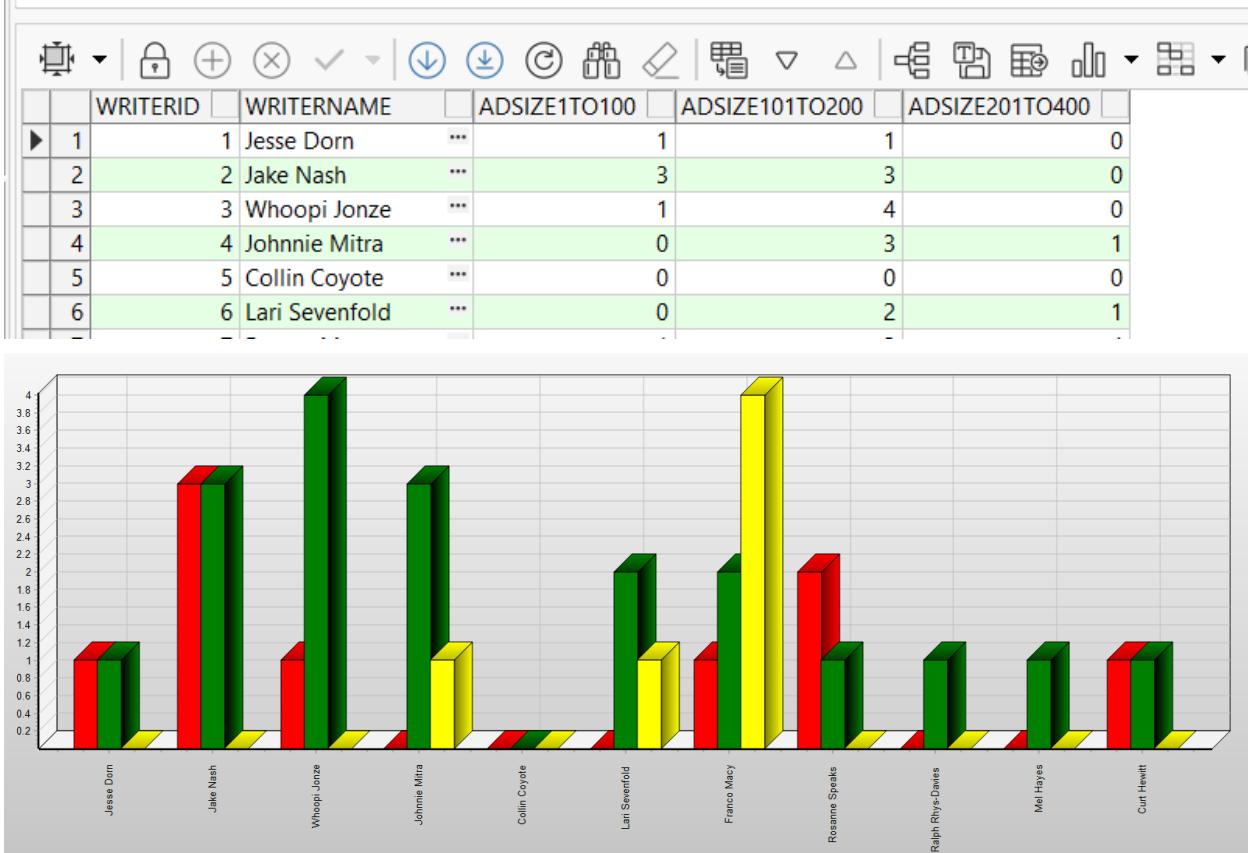
שאילטה 4:

ככה אפשר לראות בקלות את המחברים הći מנועים לפי תחום

```

SELECT
    writer.writerid,
    writer.writername,
    COALESCE(ad1to100.AdSize1to100, 0) AS AdSize1to100,
    COALESCE(ad101to200.AdSize101to200, 0) AS AdSize101to200,
    COALESCE(ad201to400.AdSize201to400, 0) AS AdSize201to400
FROM
    writer
LEFT JOIN
    ad1to100 ON writer.writerid = ad1to100.writerid
LEFT JOIN
    ad101to200 ON writer.writerid = ad101to200.writerid
LEFT JOIN
    ad201to400 ON writer.writerid = ad201to400.writerid
ORDER BY
    writer.writerid;

```



## פונקציות בשפת PL/SQL:

(1) פונקציה שגדילה את השכר של כתבים לפי הוותק שלהם, הפונקציה מקבלת את מספר שנות הוותק שהם תנאי סף לקבלת הعلاה בשכר ואת הסכום בו להעלות את השכרת ומעדכנת את הטבלה של הכותבים בהתאם, הפונקציה מחזירה את מספר הכותבים שקיבלו הعلاה בשכר.

```
CREATE OR REPLACE FUNCTION increase_salary(p_experience NUMBER, p_increase_amount NUMBER) RETURN NUMBER IS
  v_total_updated NUMBER := 0;
BEGIN
  FOR writer_rec IN (
    SELECT w.writerid, w.payperhour
    FROM writer w
    WHERE w.numyearsofexperience > p_experience
  ) LOOP
    UPDATE writer w
    SET w.payperhour = writer_rec.payperhour + p_increase_amount
    WHERE w.writerid = writer_rec.writerid;

    v_total_updated := v_total_updated + 1;
  END LOOP;

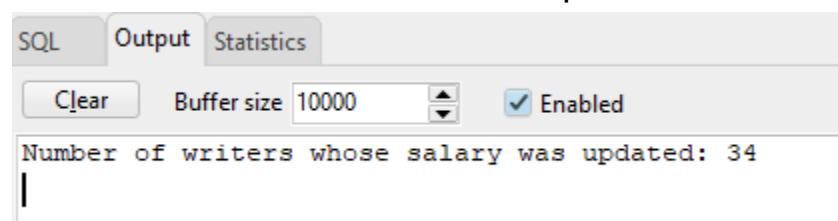
  COMMIT;

  RETURN v_total_updated;
END;
/
```

הקריאה לפונקציה היא כך, בדוגמה מעלים את השכר ב1000 לכל מי שיש וותק של מעל 10 שנה:

```
DECLARE
  v_updated_count NUMBER;
BEGIN
  v_updated_count := increase_salary(10, 1000); -- Example: Increase salary for writers with more than 10 years of
  DBMS_OUTPUT.PUT_LINE('Number of writers whose salary was updated: ' || v_updated_count);
END;
/
```

זה הפלט של הפונקציה:



הטבלה writer לפני הרצת הפונקציה:

	WRITERID	WRITERNAME	NUMYEARSOFEXPERIENCE	PAYPERHOUR
▶	1	Jesse Dorn	4	30
	2	Jake Nash	8	127
	3	Whoopi Jonze	5	48
	4	Johnnie Mitra	8	130
	5	Collin Coyote	7	129
	6	Lari Sevenfold	7	127
	7	Franco Macy	4	99
	8	Rosanne Speaks	12	88
	9	Ralph Rhys-Davies	5	104
	10	Mel Hayes	4	108
	11	Curt Hewitt	11	68
	12	Javon Webb	11	30
	13	Ronny Fonda	7	145
	14	Al Easton	7	39
	15	Brenda Strong	9	73
	16	Cheryl Craddock	5	59
	17	Phil Berkeley	12	117
	18	Keith Coward	15	122

הטבלה writer אחרי הרצת הפונקציה:

	WRITERID	WRITERNAME	NUMYEARSOFEXPERIENCE	PAYPERHOUR
► 1	1	Jesse Dorn	4	30
	2	Jake Nash	8	127
	3	Whoopi Jonze	5	48
	4	Johnnie Mitra	8	130
	5	Collin Coyote	7	129
	6	Lari Sevenfold	7	127
	7	Franco Macy	4	99
	8	Rosanne Speaks	12	1088
	9	Ralph Rhys-Davies	5	104
	10	Mel Hayes	4	108
	11	Curt Hewitt	11	1068
	12	Javon Webb	11	1030
	13	Ronny Fonda	7	145
	14	Al Easton	7	39
	15	Brenda Strong	9	73
	16	Cheryl Craddock	5	59
	17	Phil Berkeley	12	1117
	18	Keith Coward	15	1122

(2) פונקציה שמקבלת כפרמטר שם של פלטפורמה ו-2 תאריכים, ומחזירה את הסכום שהרוויחה בין 2 התאריכים האלה.

```

CREATE OR REPLACE FUNCTION get_price_sum_between_dates(
    p_platform_name IN VARCHAR2,
    p_start_date     IN DATE,
    p_end_date       IN DATE
) RETURN NUMBER IS
    v_total_price NUMBER := 0;
BEGIN
    FOR price_rec IN (
        SELECT a.pricetopayplatform
        FROM appearson a
        WHERE a.platformname = p_platform_name
            AND a.datepublished BETWEEN p_start_date AND p_end_date
    ) LOOP
        v_total_price := v_total_price + price_rec.pricetopayplatform;
    END LOOP;

    RETURN v_total_price;
END;
/

```

התוכנית שקוראת לפונקציה היא:

```

DECLARE
    v_total_price NUMBER;
BEGIN
    v_total_price := get_price_sum_between_dates('Yated_neeman', DATE '2004-01-01', DATE '2007-02-01');

    DBMS_OUTPUT.PUT_LINE('Total price between dates: ' || v_total_price);
END;
/

```

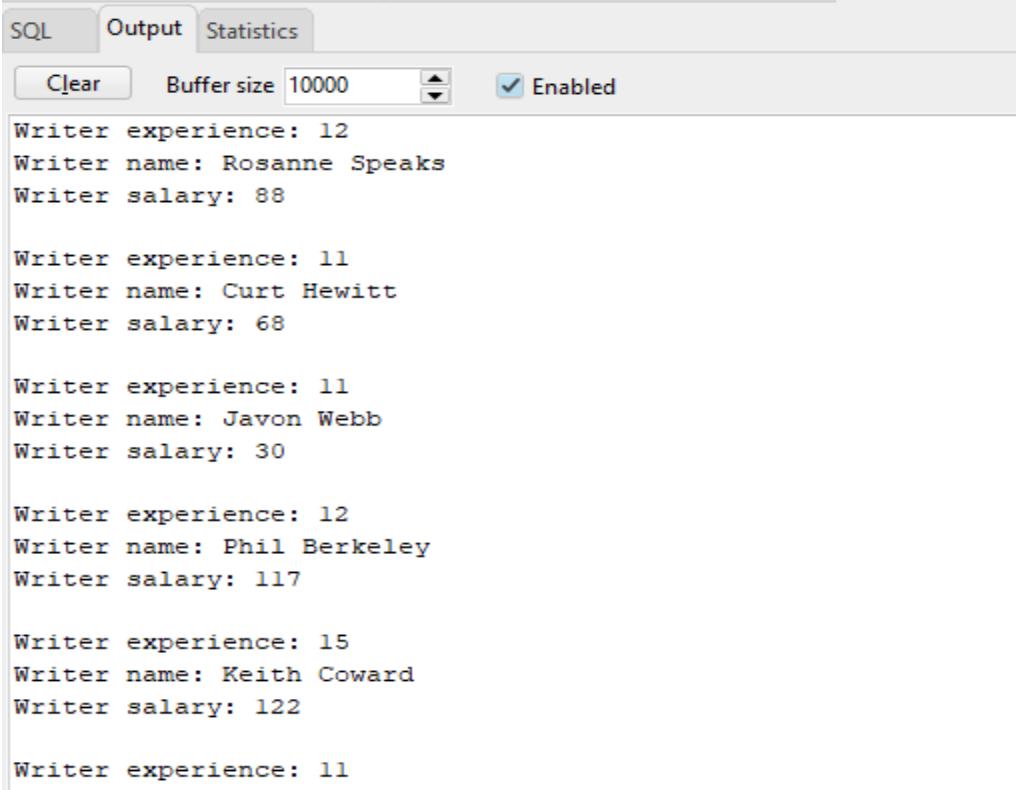
## שימוש ב :cursor

```

DECLARE
    CURSOR writer_cur (p_experience NUMBER) IS
        SELECT writername, numyearsofexperience, payperhour, writerid
        FROM writer
        WHERE numyearsofexperience >= p_experience;
BEGIN
    FOR r_writer_rec IN writer_cur(10)
    LOOP
        DBMS_OUTPUT.put_line('Writer experience: ' || r_writer_rec.numyearsofexperience);
        DBMS_OUTPUT.put_line('Writer name: ' || r_writer_rec.writername);
        DBMS_OUTPUT.put_line('Writer salary: ' || r_writer_rec.payperhour);
        DBMS_OUTPUT.new_line;
    END LOOP;
END;

```

התוצאה היא:



The screenshot shows the Oracle SQL Developer interface with the 'Output' tab selected. The buffer size is set to 10000 and is enabled. The output window displays the results of the cursor query, listing five writers with their experience, names, and salaries.

```

Writer experience: 12
Writer name: Rosanne Speaks
Writer salary: 88

Writer experience: 11
Writer name: Curt Hewitt
Writer salary: 68

Writer experience: 11
Writer name: Javon Webb
Writer salary: 30

Writer experience: 12
Writer name: Phil Berkeley
Writer salary: 117

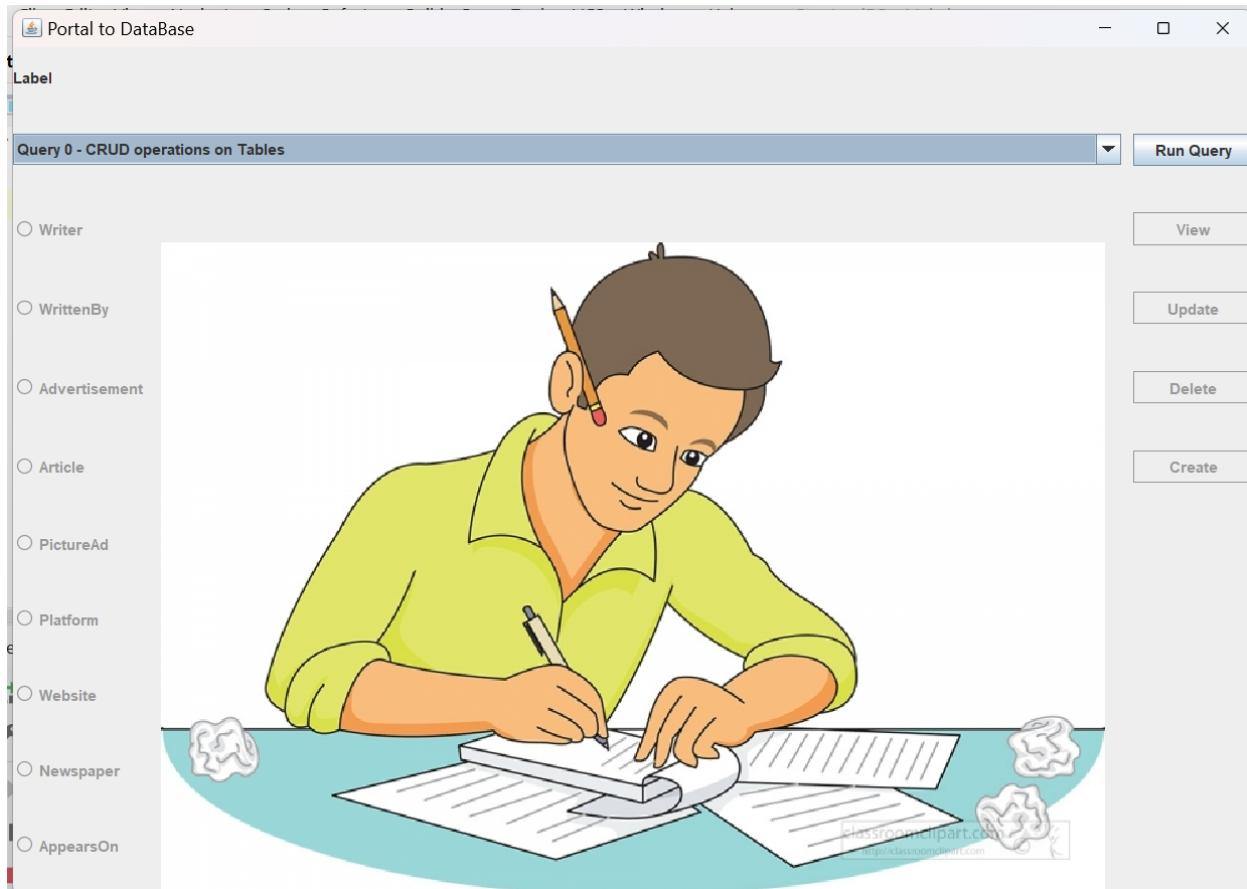
Writer experience: 15
Writer name: Keith Coward
Writer salary: 122

Writer experience: 11

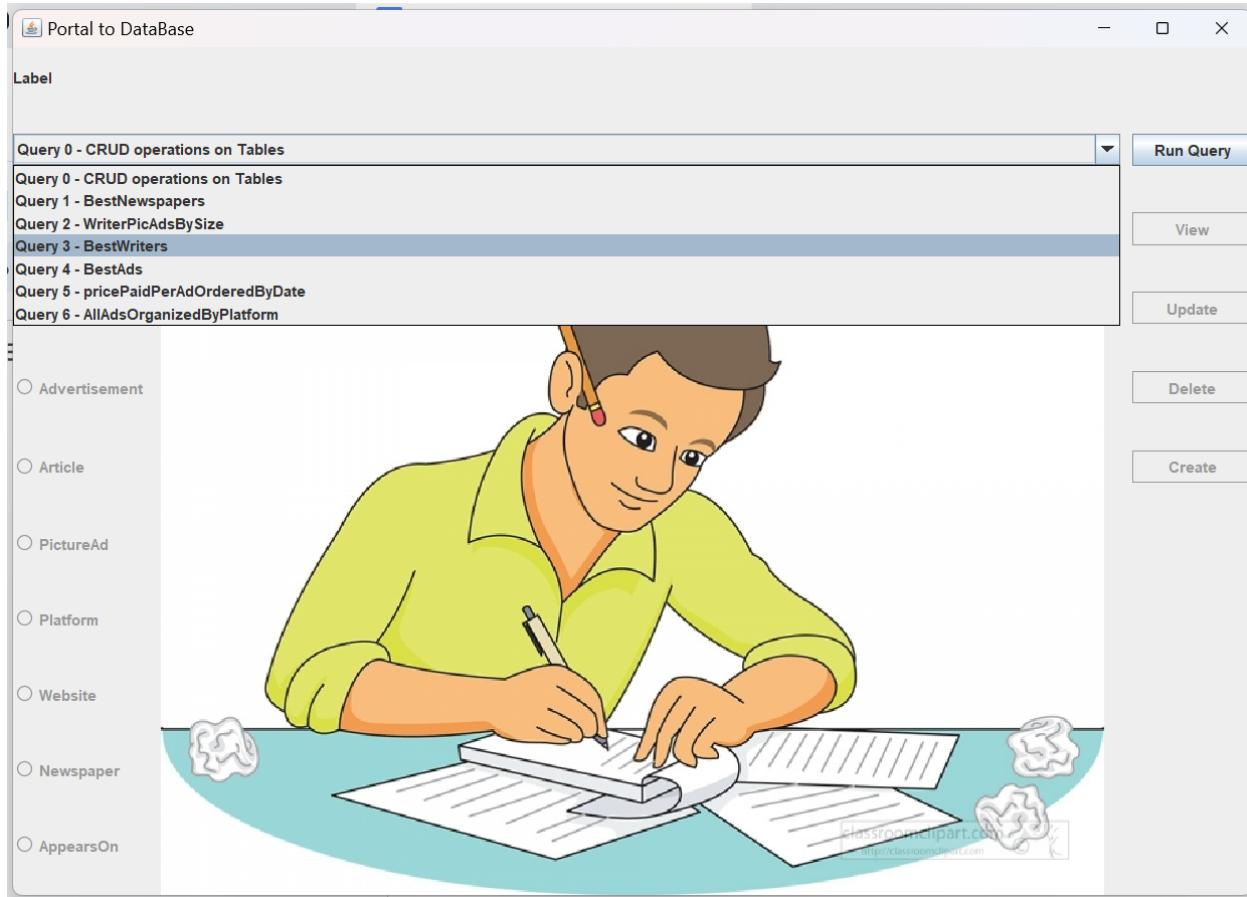
```

## ממשק GUI לטפל בסיס נתונים:

פיתחנו ממשק GUI (כלו כתוב ב-*Java*) ע"פ עיצוב תוכנה מסודר ומוצלח. הקוד נמצא בקישור [github](#) המצוין למטה.



אפשר לבחור שאלתא שורצים להרץ (נראה כמה דוגמאות):



## שאילה 5

Portal to DataBase

Query 5 - pricePaidPerAdOrderedByDate

Run Query

View

Writer

WrittenBy

Advertisement

Article

PictureAd

Platform

Website

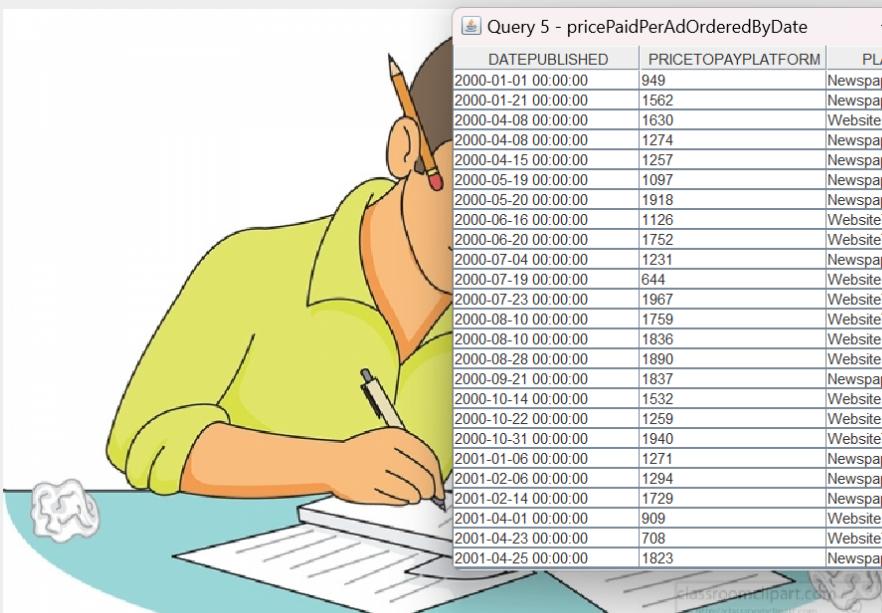
Newspaper

AppearsOn

Query 5 - pricePaidPerAdOrderedByDate

DATEPUBLISHED	PRICETOPAYPLATFORM	PLATFORMTYPE
2000-01-01 00:00:00	949	NewspaperType
2000-01-21 00:00:00	1562	NewspaperType
2000-04-08 00:00:00	1630	Website
2000-04-08 00:00:00	1274	NewspaperType
2000-04-15 00:00:00	1257	NewspaperType
2000-05-19 00:00:00	1097	NewspaperType
2000-05-20 00:00:00	1918	NewspaperType
2000-06-16 00:00:00	1126	WebsiteType
2000-06-20 00:00:00	1752	WebsiteType
2000-07-04 00:00:00	1231	NewspaperType
2000-07-19 00:00:00	644	Website
2000-07-23 00:00:00	1967	WebsiteType
2000-08-10 00:00:00	1759	WebsiteType
2000-08-10 00:00:00	1836	Website
2000-08-28 00:00:00	1890	Website
2000-09-21 00:00:00	1837	NewspaperType
2000-10-14 00:00:00	1532	Website
2000-10-22 00:00:00	1259	Website
2000-10-31 00:00:00	1940	WebsiteType
2001-01-06 00:00:00	1271	NewspaperType
2001-02-06 00:00:00	1294	NewspaperType
2001-02-14 00:00:00	1729	NewspaperType
2001-04-01 00:00:00	909	Website
2001-04-23 00:00:00	708	WebsiteType
2001-04-25 00:00:00	1823	NewspaperType

classroomclipart.com  
http://classroomclipart.com



## שאילה 3

Portal to DataBase

Label

Query 3 - BestWriters

Run Query

Writer

WrittenBy

Advertisement

Article

PictureAd

Platform

Website

Newspaper

AppearsOn

Query 3 - BestWriters

WRITERNAME	NUMYEARSOFEXPERIEN...	RAITING	NUMOFAFFS
Brad Moriarty	10	22.73	1
Mekhi Horizon	9	21.43	3
Whoopi Jonze	10	20.83	7
Jaime Stuermer	7	18.92	4
Al Easton	7	17.95	2
Janeane Phillippe	6	17.65	5
Mindy Culkin	7	16.28	1
Elle Rifkin	6	13.64	3
Laurie Tomei	5	13.51	5
Jesse Dorn	4	13.33	2
Lila Damon	10	12.82	3
Brenda Strong	9	12.33	2
Hank Levert	7	12.07	3
Jean-Luc Kershaw	6	12	1
Judi Keener	9	11.54	1
April Tankard	9	10.84	4
Charles Iglesias	4	10.53	7
Kurt Levine	8	10.39	4
Glen Karyo	8	10.26	2
Janice Hall	9	10.11	2
Sissy Turner	3	10	6
Reese McElhone	5	10	4
Sheryl Thomas	5	9.09	1
Cheryl Craddock	5	8.47	1
Russell Hayek	10	8.33	3

classroomclipart.com  
http://classroomclipart.com

## שאילה 4

Portal to DataBase

Label

Writer

WrittenBy

Advertisement

Article

PictureAd

Platform

Website

Newspaper

AppearsOn

Query 4 - BestAds

Run Query

View

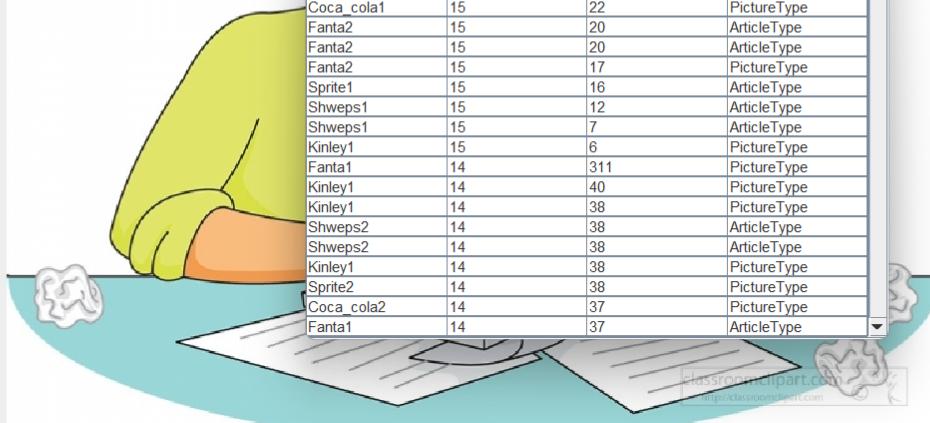
Update

Delete

Create

Query 4 - BestAds

ADNAME	NUMYEARSOFE...	HOURSTOWRITE	ADTYPEWRITTEN
Fanta1	15	37	ArticleType
Kinley2	15	30	PictureType
Fanta1	15	30	ArticleType
Fanta2	15	29	PictureType
Kinley2	15	26	PictureType
Kinley2	15	26	PictureType
Fanta2	15	24	PictureType
Sprite2	15	23	ArticleType
Coca_cola1	15	22	PictureType
Fanta2	15	20	ArticleType
Fanta2	15	20	ArticleType
Fanta2	15	17	PictureType
Sprite1	15	16	ArticleType
Shweps1	15	12	ArticleType
Shweps1	15	7	ArticleType
Kinley1	15	6	PictureType
Fanta1	14	311	PictureType
Kinley1	14	40	PictureType
Kinley1	14	38	PictureType
Shweps2	14	38	ArticleType
Shweps2	14	38	ArticleType
Kinley1	14	38	PictureType
Sprite2	14	38	PictureType
Coca_cola2	14	37	PictureType
Fanta1	14	37	ArticleType



classroomclipart.com  
http://classroomclipart.com

הממשק תומך בפעולות CRUD.  
הנה דוגמא של VIEW הממשק גם מאפשר לראות כמה טבלאות בו בזמן:

The screenshot shows a database interface with three tables displayed simultaneously:

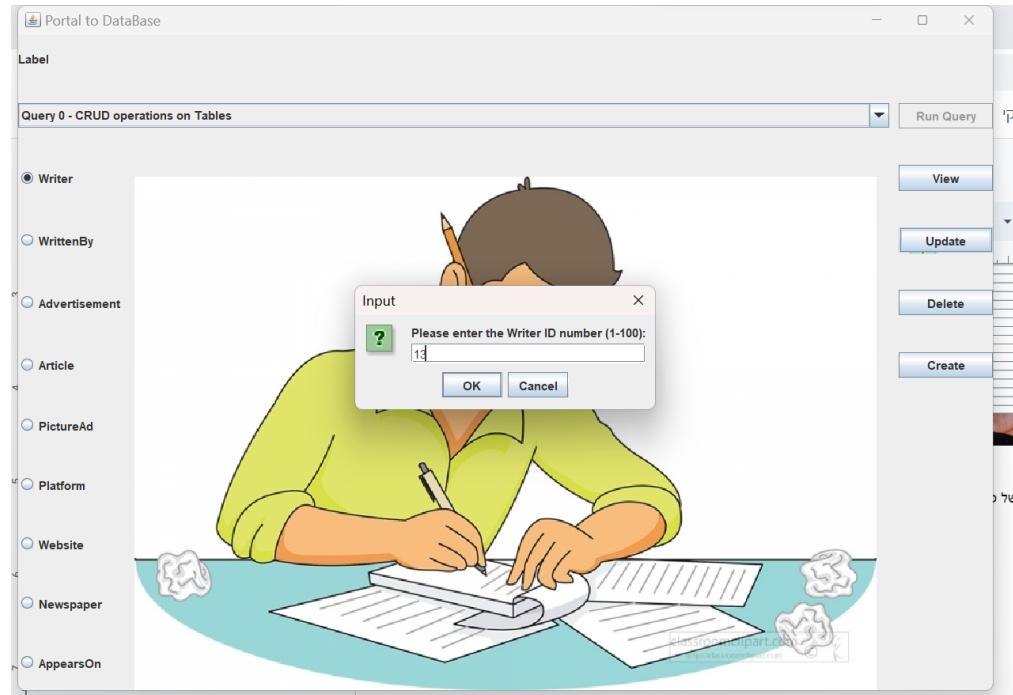
- WrittenBy** (Top Table):
 

ADTYPEWRITTEN	ADID	WRITERID
ArticleType	51	36
ArticleType	51	41
ArticleType	52	29
ArticleType	54	35
ArticleType	58	7
ArticleType	58	59
ArticleType	59	11
ArticleType	59	22
ArticleType	59	29
ArticleType	60	59
ArticleType	61	1
ArticleType	61	80
ArticleType	61	87
ArticleType	61	95
ArticleType	62	19
ArticleType	62	83
ArticleType	65	17
ArticleType	65	51
- Advertisement** (Middle Table):
 

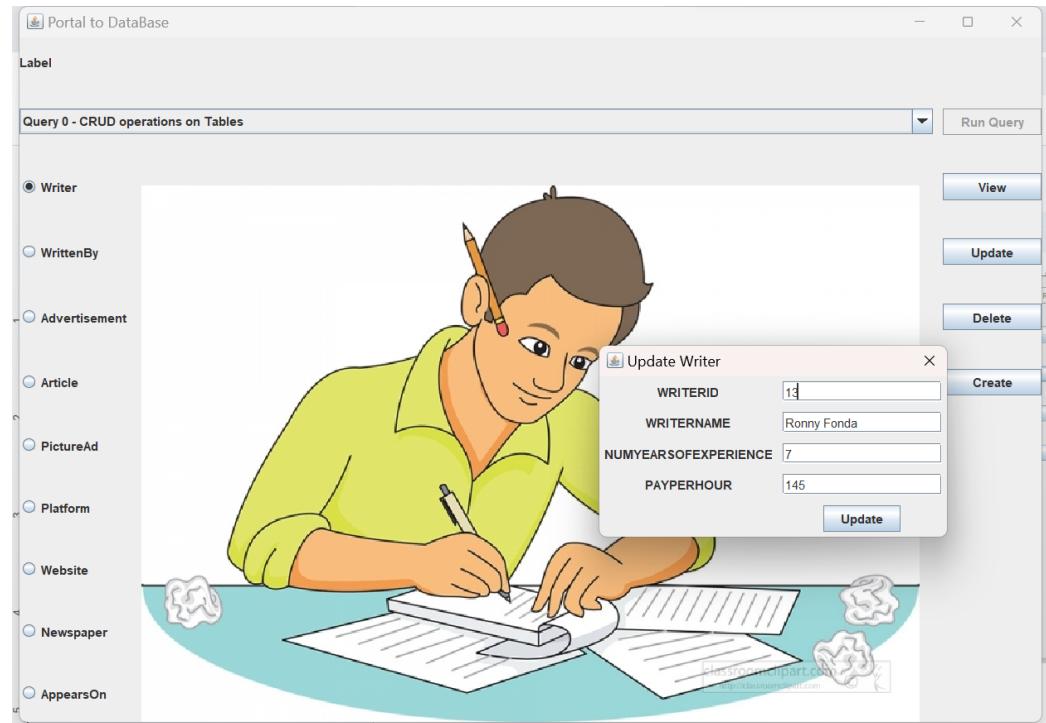
ADID	HOURSTOWRITE	ADNAME
166	23	Sprite2
167	20	Sprite2
168	11	Kinley1
169	30	Fanta1
170	13	Coca_cola1
171	13	Shweps1
172	13	Sprite1
173	33	Kinley2
174	30	Fanta1
175	23	Fanta1
176	7	Shweps1
177	17	Coca_cola1
178	27	Sprite1
179	34	Sprite1
180	6	Shweps1
181	11	Kinley2
182	33	Kinley1
183	18	Coca_cola2
184	10	Fanta2
- Writer** (Bottom Table):
 

WRITERID	WRITERNAME	NUMYEARSOFEX...	PAYPERHOUR
1	Jesse Dorn	4	30
2	Jake Nash	8	127
3	Whoopi Jonze	10	48
4	Ariell	8	130
5	Collin Coyote #2	7	129
6	Lari Sevenfold	7	127
7	Franco Macy	4	99
8	Rosanne Speaks	12	2088
9	Ralph Rhys-Davies	5	104
10	Mel Hayes	4	108
11	Curt Hewitt	11	2068
12	Javon Webb	11	2030
13	Ronny Fonda	7	145
14	Al Easton	7	39
15	Brenda Strong	9	73
16	Cheryl Craddock	5	59
17	Phil Berkeley	12	2117

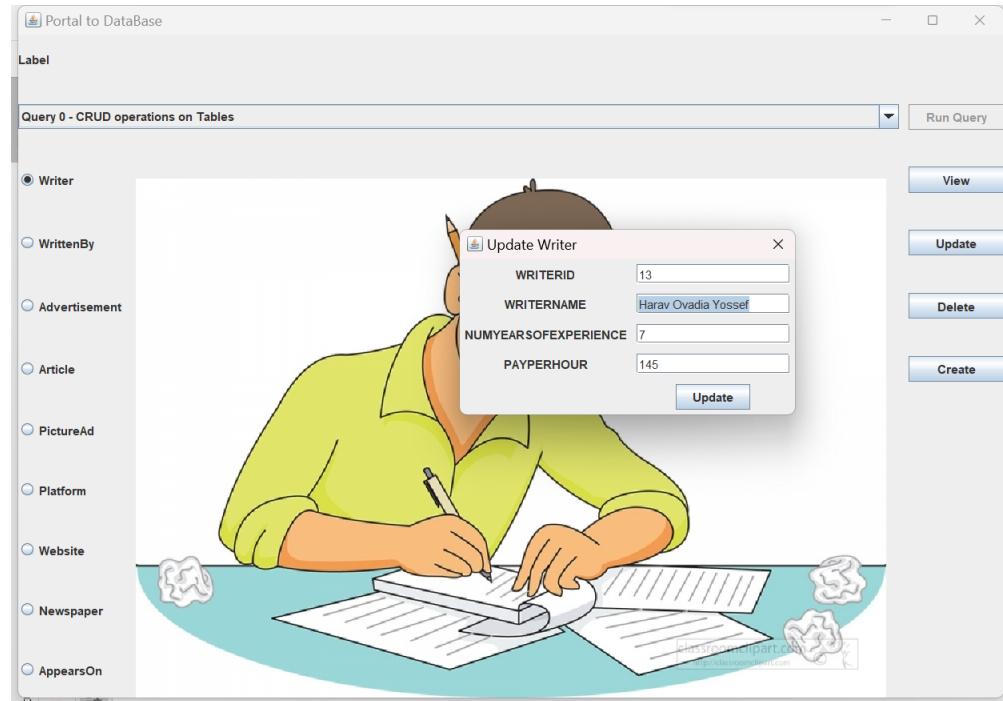
מימוש של פעולה Update עבור **Writer**, **advertisement** וכו':



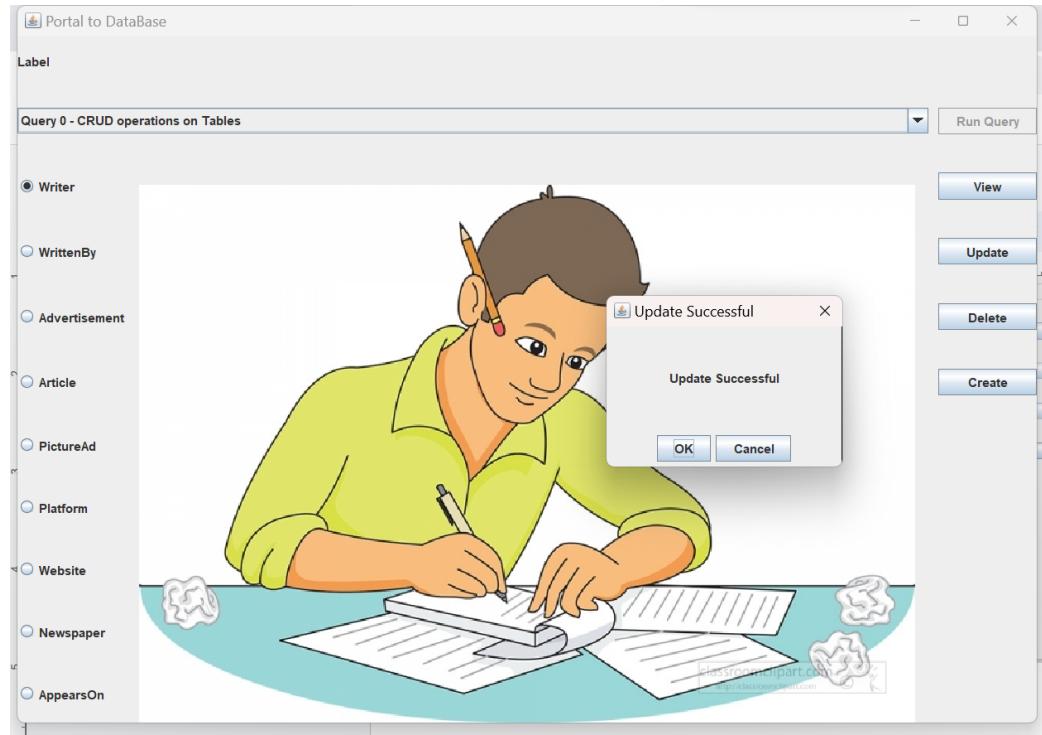
קובץ חלון עם המידע הקיים:



נשנה את השם לרוב עובדיה ז"ל



לוחצים על Update



ורואים את התוצאות המעודכנות:

The screenshot shows a database application window titled "Portal to DataBase". On the left, there is a sidebar menu with the following items:

- Writer
- WrittenBy
- Advertisement
- Article
- PictureAd
- Platform
- Website
- Newspaper
- AppearsOn

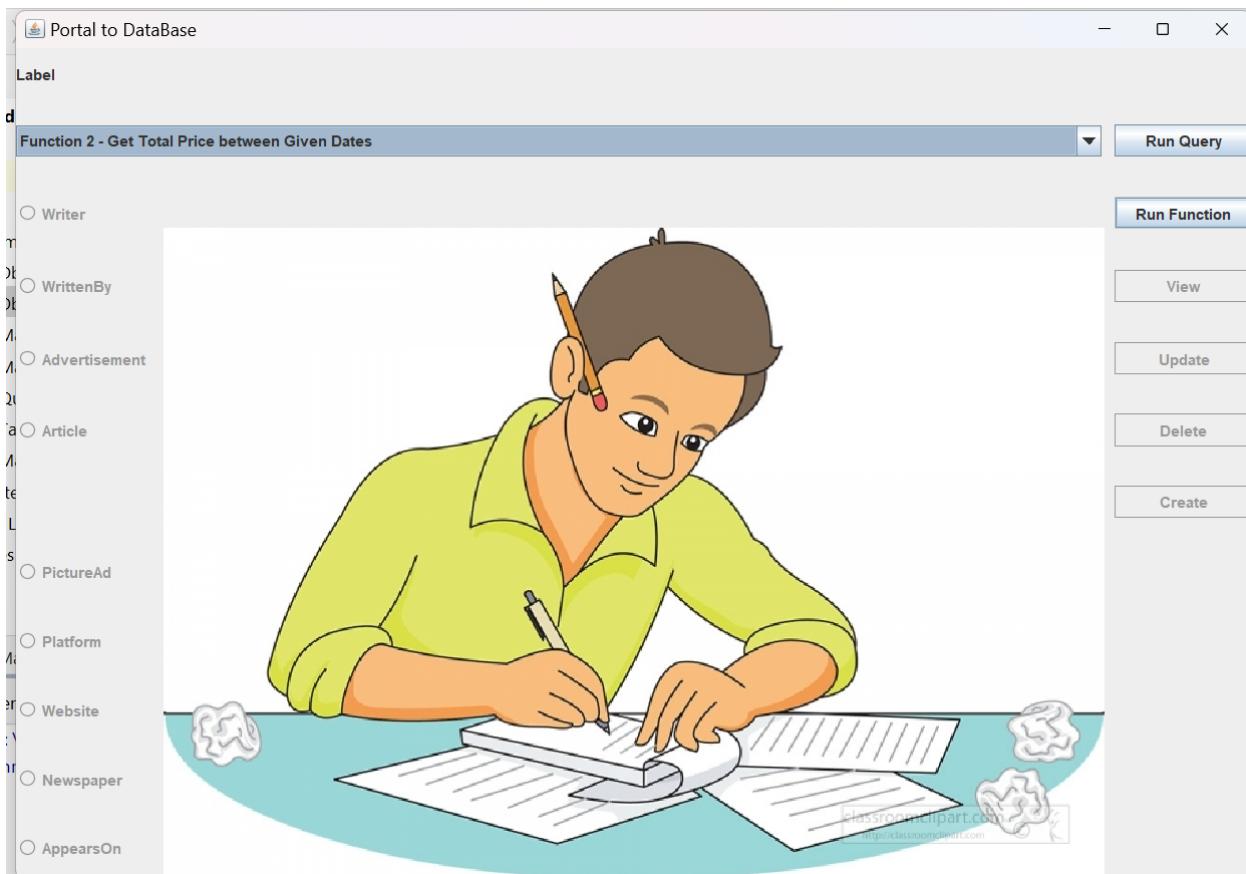
The main area displays a table titled "Writer" with the following data:

WRITERID	WRITERNAME	NUMYEARSOFEXPE...	PAYPERHOUR
1	Jesse Dom	4	30
2	Jake Nash	8	127
3	Whoopi Jonze	10	48
4	Ariell	8	130
5	Collin Coyote #2	7	129
6	Lari Sevenfold	7	127
7	Franco Macy	4	99
8	Rosanne Speaks	12	2088
9	Ralph Rhys-Davies	5	104
10	Mel Hayes	4	108
11	Curt Hewitt	11	2068
12	Javon Webb	11	2030
13	Harav Ovadia Yossef	7	145
14	Al Easton	7	39
15	Brenda Strong	9	73
16	Cheryl Craddock	5	59
17	Phil Berkeley	12	2117
18	Keith Coward	15	2122
19	April Tankard	9	83
20	Rene Hartnett	3	142
21	Jean Levert	11	2034
22	Elias Coleman	3	57
23	Jaimie Stuermer	7	37
24	Natasha Gough	15	2103
25	Sheryl Thomas	5	55

The background of the application features a cartoon illustration of a person writing at a desk with a stack of papers.

ICI'

**קריאה לפונקציה מ-PL/SQL:**  
בוחרים את הפונקציה  
מקליקים על "Run Function"



מכוונים את הנתונים המבוקשים:

The screenshot shows a software application window titled "Portal to DataBase". On the left, there is a sidebar with various labels: Label, Query 0, WrittenBy, Advertisement, Article, PictureAd, Platform, Website, Newspaper (which is selected), and AppearsOn. A central area displays a table titled "Newspaper" with columns "NUMCOPIESSPRINTED" and "PLATFORMNAME". The table contains four rows of data. To the right of the table is a "Run Query" button. Below the table, there is a "Run Function" button. A cartoon illustration of a person sitting at a desk, writing in a notebook, is overlaid on the interface. A modal dialog box titled "Get Total Price of Newspaper Adds From Given Dates:" is open in the center. It has three input fields: "Newspaper Name:" with "Yated\_neeman" selected, "Start Date (YYYY-MM-DD):" with "2004-01-01" entered, and "End Date (YYYY-MM-DD):" with "2007-02-01" entered. At the bottom of the dialog is a "Run This Function" button. The background of the application window features a blue gradient.

NUMCOPIESSPRINTED	PLATFORMNAME
3836	Mishpacha
1898	Yated_neeman
4736	Besheva
2185	Yediot_acharonot

Run Query

Run Function

View

Update

Delete

Create

Get Total Price of Newspaper Adds From Given Dates:

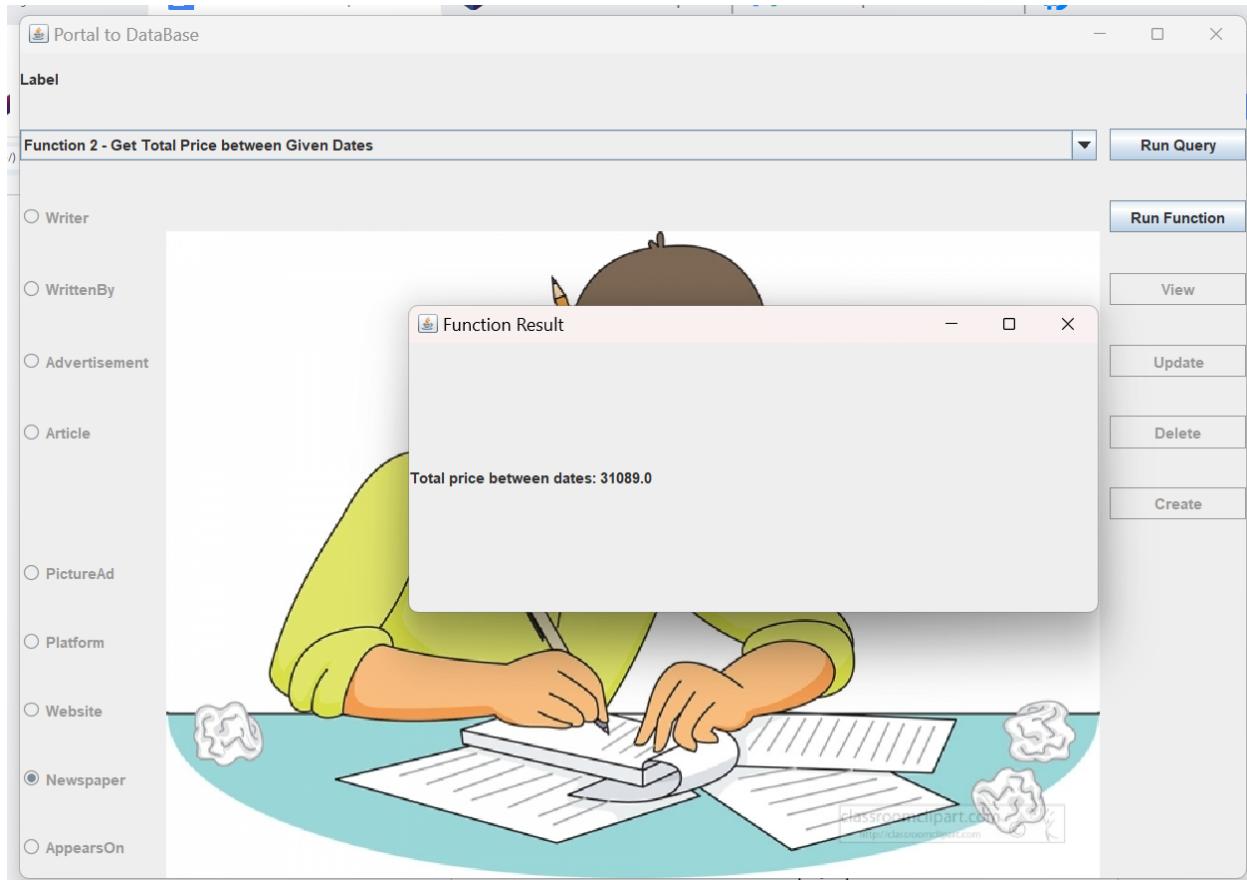
Newspaper Name: Yated\_neeman

Start Date (YYYY-MM-DD): 2004-01-01

End Date (YYYY-MM-DD): 2007-02-01

Run This Function

ורואים את התוצאה!



\* מימושו רק פונקציה אחת ולא את שאר הפונקציות\*

## קטעי קוד נבחרים

(שוב, אפשר לראות את הכל בגייט...):

קבלת הנתונים מבוסיס הנתונים, ויצירת רשימת המידע בצורה דינמית:

```
2 usages
public List<List<String>> selectAllFromTable(String tableName, Optional<String> optionalCond) {
    try (Connection connection = DriverManager.getConnection(jdbcUrl, username, password)) {
        List<List<String>> queryResults = new ArrayList<>();

        // Execute the query
        Statement statement = connection.createStatement();
        String query = "SELECT * FROM " + tableName;
        if( !Objects.isNull(optionalCond) && optionalCond.isPresent() && !optionalCond.get().isEmpty() ) {
            query += " " + optionalCond.get();
        }
        ResultSet resultSet = statement.executeQuery( query );
        ResultSetMetaData rsmd = resultSet.getMetaData();
        int columnCount = rsmd.getColumnCount();
        List<String> colNames = new ArrayList<>();
        // The column count starts from 1
        for (int i = 1; i <= columnCount; i++ ) {
            colNames.add(rsmd.getColumnName(i));
        }
        queryResults.add(colNames);
        // Retrieve the values from the result set
        while (resultSet.next()) {
            List<String> rowValues = new ArrayList<>();
            for (int i = 0; i < columnCount; i++) {
                rowValues.add(resultSet.getString(colNames.get(i)));
            }
            queryResults.add(rowValues);
        }
    }
}
```

יצירת השאלה באופן דינמי (כדי לאפשר אותן שאלות ליטראליות שונות):

```
1 usage
public void runUpdateQuery(String tableName, List<List<String>> lst, List<String> idList) {
    String query = "UPDATE " + tableName + "\n";
    query += "Set ";
    for (int i = 0; i < lst.get(0).size(); i++) {
        query += lst.get(0).get(i) + " = '" + lst.get(1).get(i) + "', ";
    }
    //remove last comma:
    query = query.substring(0, query.length() - 2);
    query += "\n Where " + idList.get(0) + " = " + idList.get(1) + "";

    try (Connection connection = DriverManager.getConnection(jdbcUrl, username, password)) {

        // Execute the query
        Statement statement = connection.createStatement();
        ResultSet resultSet = statement.executeQuery( query );
        ResultSetMetaData rsmd = resultSet.getMetaData();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

יצירת הטבלאות לצפייה בנתונים בצורה דינמית לפי התוצאות:

```
private void showTable(List<List<String>> dataToDisplay, String queryTitle){  
  
    // Create the JFrame and set its properties  
    JFrame frame = new JFrame(queryTitle);  
    // Create the JTable with a DefaultTableModel  
    JTable table = new JTable();  
    DefaultTableModel model = new DefaultTableModel();  
    // Get the column names from the first row of the data  
    List<String> columnNames = dataToDisplay.get(0);  
    for (String columnName : columnNames) {  
        model.addColumn(columnName);  
    }  
    // Populate the rows in the table  
    for (int i = 1; i < dataToDisplay.size(); i++) {  
        List<String> rowData = dataToDisplay.get(i);  
        model.addRow(rowData.toArray());  
    }  
    table.setModel(model);  
    JScrollPane scrollPane = new JScrollPane(table);  
    frame.add(scrollPane);  
    frame.pack();  
    frame.setVisible(true);  
}
```

..תיעוד