

מיני פרויקט בסיסי נתונים

מערכת לניהול פרסום של מאפייה



מגישים

יהודה שני

ניב ברטוב

תוכן עניינים

3.....	תיאור הארגון
3.....	תרשים ERD
4.....	תרשים סכמה:
4.....	סוגי הישויות:
6.....	הטבלאות - פירוט יצירתן והכנסת הנתונים
6.....	יצירת הטבלאות בSQL
6.....	Generate SQL
7.....	מילוי הטבלאות בעזרת DataGenerator
11.....	שאילתות
11.....	שאילתא 1
11.....	שאילתא 2
12.....	שאילתא 3
12.....	שאילתא 4
13.....	דו"חות וגרפים
13.....	שאילתא 1
13.....	שאילתא 2
14.....	עדכן בסיס הנתונים באמצעות תכניות בשפתPL/SQL
14.....	עדכן
14.....	מחיקה
15.....	שימוש rollback
16.....	שאילתות ב- PLSQL
18.....	פקודת Grant
18.....	זמן ריצה
20.....	אינדקסים
21.....	פרצדרות
21.....	פרצדרה 1
22.....	פרצדרה 2
23.....	ישום משק משתמש
23.....	הציג המערכת- משק המשתמש
23.....	תיבת התצוגה
24.....	תיבת השאלת החופשית
25.....	תיבת השאלות בנויות
27.....	הציג המערכת- הקוד
27.....	קוד הפיתון של העמוד הראשי:
28.....	קוד הפיתון עבור הצגת הטבלאות:
29.....	קובץ HTML של דף הראשי:
31.....	קובץ HTML של הצגת הטבלאות

תיאור הארגון

ניהול הפרסום של מאפייה. רישום ומעקב אחריו פרסומים באמצעות המדיות השונות. המערכת שומרת את כל הפרטמים וכן את מידת הצלחתם.

הישויות הראשיות:

קמפיין - מתאר פרטיים של קמפיין מסוים.

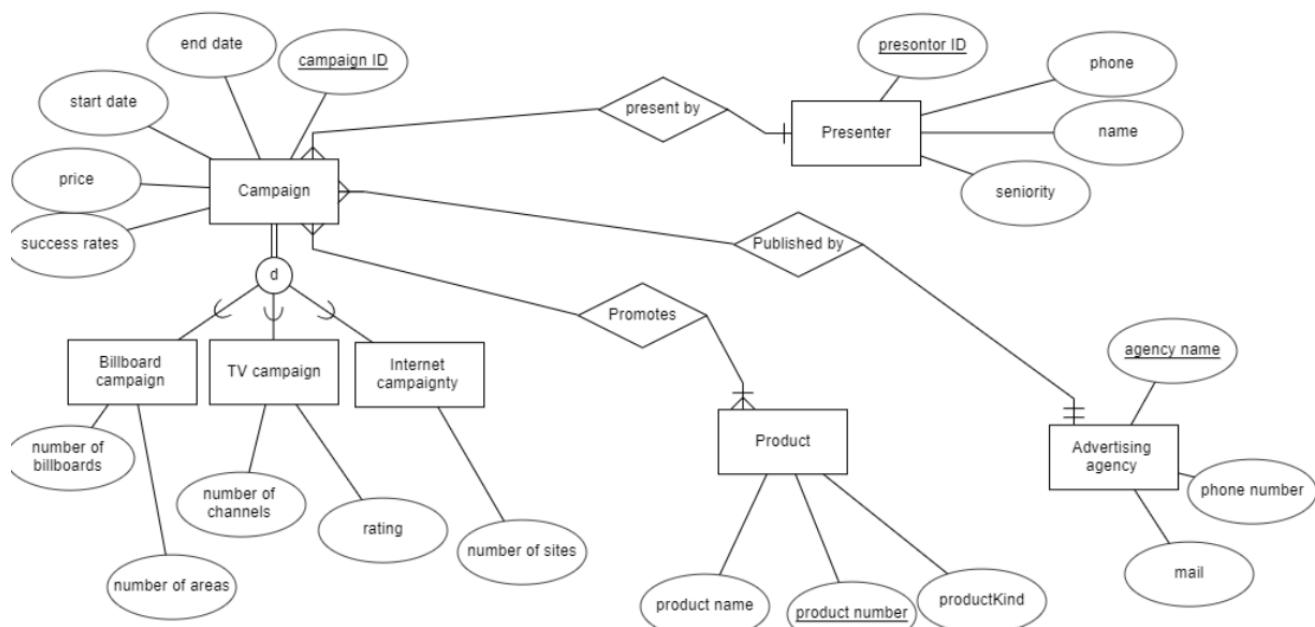
יורשים ממנה שלוש סוגי קמפיין שונים- א. קמפיין אינטרנט. ב. קמפיין טלוויזיוני. ג. קמפיין במודעות.

מוציארים- מתאר את המוציארים הנמכרים במאפייה.

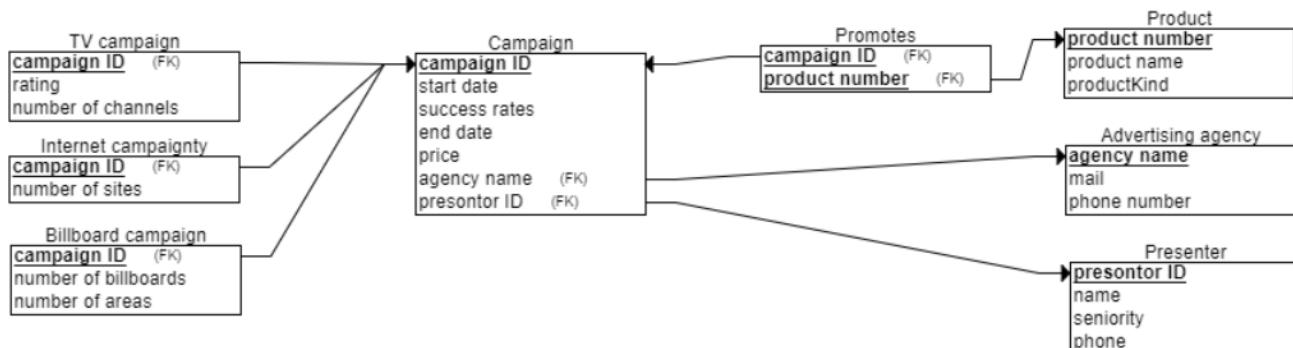
SOCIALITES - מתאר את סוכנויות הפרסום שבאמצעותם אפשר לעלות קמפיין.

פרזנטטור - מייצג ידוען לפרסום המוציאר.

תרשים ERD



תרשים סכמה



סוגי הישויות:

בהתבסס על תרשימים ERD שיצרנו בשלבים מוקדמים של הפרויקט, ניצור מאגר מידע מתאים תוך התחשבות בכל הקשרים והישויות. בדוח להלן נציג ונפרט את היחסות והקשרים העיקריים.

משרד פרסום - Advertising_agency

שם הסוכנות (מפתח)	AgencyName
אימייל	Mail
מספר טלפון	PhoneNumber

פרזנטור - Presenter

מחזה פרזנטטור (מפתח)	presenterID
שם הפרזנטטור	Name
טלפון	Phone
ニיסוין	Seniority

מוצר - Product

מחזה המוצר (מפתח)	Productid
שם מוצר	ProductName
סוג מוצר	Kind

קמפייל - Campaign

מחזה מסע פרסום (מפתח)	campaign ID
תאריך התחלה	start_date
תאריך סיום	end_date
אחוזי הצלחה	success_rates
מחיר	Price
שם הסוכנות (Foreign KEY)	AgencyName
מחזה מציג (מפתח זר)	PresontorID

קמפיין טלוויזיה - TV_campaign

מזהה מסע פרסום (מפתח)	campaign ID
מספר ערוצים	number of channels
דרוג	Rating

קמפיין של לוח מודעות - Billboard_campaign

מזהה מסע פרסום (מפתח)	campaign ID
מספר שלטי חוצות	number of billboards
מספר אזורים	number of areas

קמפיין אינטרנט - Internet_campaign

מזהה מסע פרסום	campaign ID
מספר אתרים	number of sites

קשר - מפרסם מוצר - promotes

many to many

מזהה מסע פרסום	campaign ID
מזהה המוצר (מפתח)	Productid

הטבלאות - פירוט יצרתן והכנסת הנתונים

יצירת הטבלאות ב SQL

Generate SQL

```

CREATE TABLE Advertising_agency
(
    agency_name VARCHAR(30),
    mail VARCHAR(30),
    phone_number INT NOT NULL,
    PRIMARY KEY (agency_name)
);

CREATE TABLE Presenter
(
    name VARCHAR(30),
    seniority INT NOT NULL,
    phone INT NOT NULL,
    presontor_ID INT NOT NULL,
    PRIMARY KEY (presontor_ID)
);

CREATE TABLE Product
(
    product_name VARCHAR(30),
    product_number INT NOT NULL,
    productKind VARCHAR(30),
    PRIMARY KEY (product_number)
);

CREATE TABLE Campaign
(
    start_date DATE NOT NULL,
    success_rates INT NOT NULL,
    campaign_ID INT NOT NULL,
    end_date DATE NOT NULL,
    price INT NOT NULL,
    agency_name VARCHAR(30) NOT NULL,
    presontor_ID INT NOT NULL,
    PRIMARY KEY (campaign_ID),
    FOREIGN KEY (agency_name) REFERENCES
    Advertising_agency(agency_name),
    FOREIGN KEY (presontor_ID) REFERENCES Presenter(presontor_ID)
);

CREATE TABLE TV_campaign
(
    rating INT NOT NULL,
    number_of_channels INT NOT NULL,
    campaign_ID INT NOT NULL,
    PRIMARY KEY (campaign_ID),
    FOREIGN KEY (campaign_ID) REFERENCES Campaign(campaign_ID)
);

```

```

CREATE TABLE Billboard_campaign
(
    number_of_billboards INT NOT NULL,
    number_of_areas INT NOT NULL,
    campaign_ID INT NOT NULL,
    PRIMARY KEY (campaign_ID),
    FOREIGN KEY (campaign_ID) REFERENCES Campaign(campaign_ID)
);

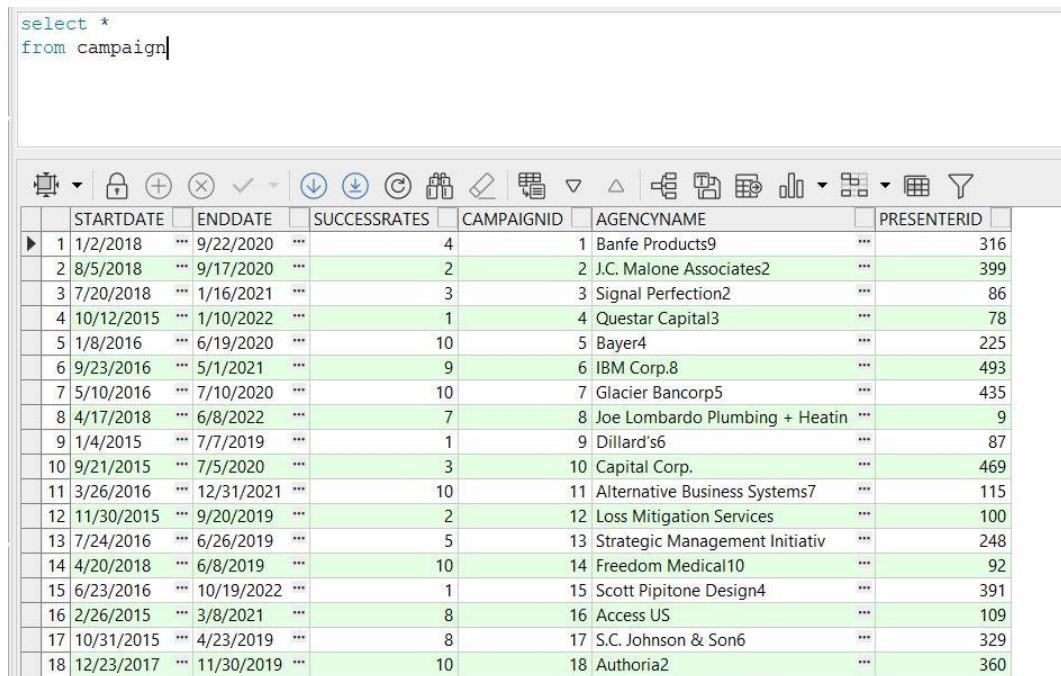
CREATE TABLE Internet_campaign
(
    number_of_sites INT NOT NULL,
    campaign_ID INT NOT NULL,
    PRIMARY KEY (campaign_ID),
    FOREIGN KEY (campaign_ID) REFERENCES Campaign(campaign_ID)
);

CREATE TABLE Promotes
(
    campaign_ID INT NOT NULL,
    product_number INT NOT NULL,
    PRIMARY KEY (campaign_ID, product_number),
    FOREIGN KEY (campaign_ID) REFERENCES Campaign(campaign_ID),
    FOREIGN KEY (product_number) REFERENCES Product(product_number)
);

```

מילוי הטבלאות באמצעות DataGenerator

א. טבלת הקמפיינים. למילוי טבלה זו יצרנו מספר רשומה המציג את המספר המזהה של הקמפיין, כמו כן מס' בין אחד לעשר המציג את דירוג הצלחת הקמפיין, בנוסף עמודה המכילה את שם סוכנות הפרסום (מטבלת סוכניות הפרסום) ועמודה המכילה מס' מזהה של הפרזנטור. ועוד שני עמודות בהם הגרנו שני שדות של תאריך ליצוג התחלת וסיום הקמפיין.



The screenshot shows a database interface with a query editor at the top containing the SQL command:

```
select * from campaign;
```

Below the query editor is a table titled "campaign" with 18 rows of data. The columns are: #, STARTDATE, ENDDATE, SUCCESSRATES, CAMPAIGNID, AGENCYNAME, PRESENTERID. The data is as follows:

#	STARTDATE	ENDDATE	SUCCESSRATES	CAMPAIGNID	AGENCYNAME	PRESENTERID
1	1/2/2018	9/22/2020	4	1	Banfe Products9	316
2	8/5/2018	9/17/2020	2	2	J.C. Malone Associates2	399
3	7/20/2018	1/16/2021	3	3	Signal Perfection2	86
4	10/12/2015	1/10/2022	1	4	Questar Capital3	78
5	1/8/2016	6/19/2020	10	5	Bayer4	225
6	9/23/2016	5/1/2021	9	6	IBM Corp.8	493
7	5/10/2016	7/10/2020	10	7	Glacier Bancorp5	435
8	4/17/2018	6/8/2022	7	8	Joe Lombardo Plumbing + Heatin	9
9	1/4/2015	7/7/2019	1	9	Dillard's6	87
10	9/21/2015	7/5/2020	3	10	Capital Corp.	469
11	3/26/2016	12/31/2021	10	11	Alternative Business Systems7	115
12	11/30/2015	9/20/2019	2	12	Loss Mitigation Services	100
13	7/24/2016	6/26/2019	5	13	Strategic Management Initiativ	248
14	4/20/2018	6/8/2019	10	14	Freedom Medical10	92
15	6/23/2016	10/19/2022	1	15	Scott Pipitone Design4	391
16	2/26/2015	3/8/2021	8	16	Access US	109
17	10/31/2015	4/23/2019	8	17	S.C. Johnson & Son6	329
18	12/23/2017	11/30/2019	10	18	Authoria2	360

מקמפניין יורשים שלושה סוגי קמפניינים. ולכן בכל טבלה שיוורשת יש מפתח זר הנלכח מטבלת קמפניין קמפניין טלייזוני. בטבלה זו יש עמודות ריביטינג- בה הגרלנו מס' בין 1 ל 10. ועמודה של מספר הערכאים בהם הוצג הפרסום וגם בה הגרלנו מס' בין אחד לעשר.

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, 'Data' is selected. Below it, the 'TV_CAMPAGN' table is selected. The table structure is displayed with columns: Name, Type, Size, Data, and Master. The 'Data' column for the 'RATING' column is set to 'Random(1, 10)'. The 'Data' column for the 'NUMBEROFCCHANNELS' column is set to 'Random(1, 10)'. The 'Data' column for the 'CAMPAIGNID' column is set to 'SEQUENCE(LIST(SELECT CAMPAIGNID FROM CAMPAIGN), [1])'. The 'Master' column has a dropdown arrow icon. The 'Number of records' is set to 4000. A PL/SQL code editor window on the right contains the following code:

```
PL/SQL CI
5/10/2023
TV_camp
Enter filter
TABLE
Campaign
```

```
select *
from tv_campaign|
```

התוצאות:

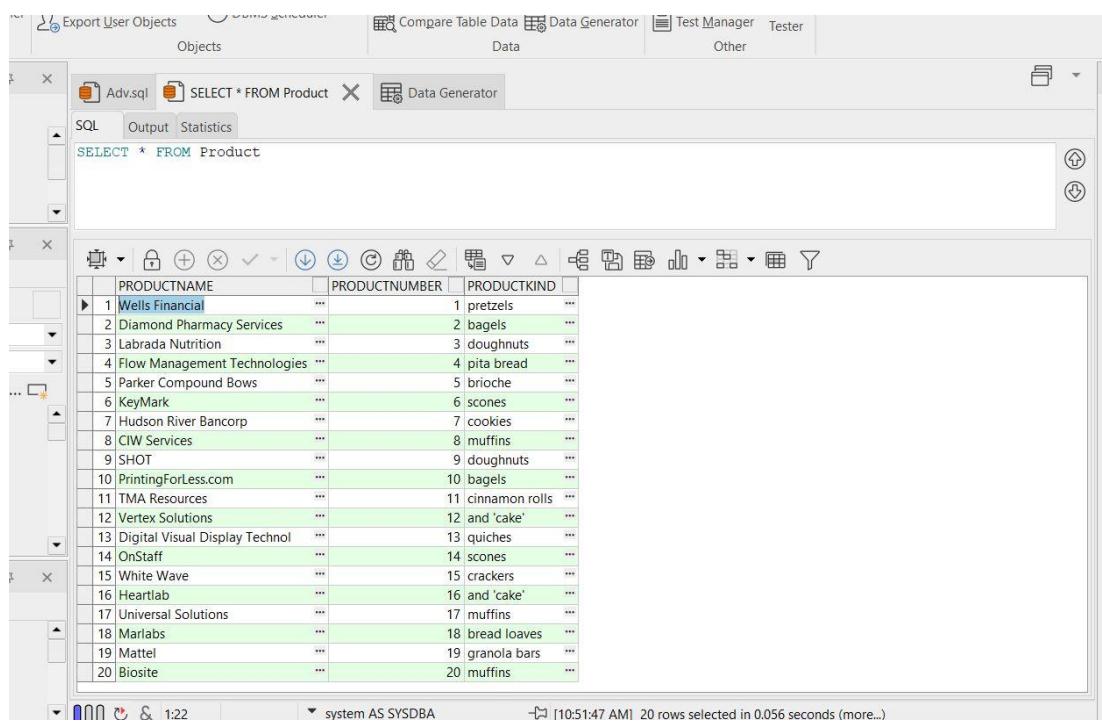
	RATING	NUMBEROFCCHANNELS	CAMPAIGNID
▶	1	3	8
2	1	8	8789
3	1	7	2213
4	5	8	1198
5	2	9	4815
6	7	1	8339
7	8	10	1617
8	6	7	6273
9	5	8	4867
10	1	1	7354
11	1	6	4267
12	7	10	4179
13	7	3	1543
14	6	9	6457
15	1	8	8121

טבלת קמפניין נוספת- בנוספ' למפתח זר של מס' זהה של הקמפניין יש בו עמודה של מס' איזורי פרסום, בה הגרלנו מס' בין אחד לחמש. ועמודה של מס' מודעות בה הגרלנו מס' בין אחד לעשרים.

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, 'Data' is selected. Below it, the 'billboard_campaign' table is selected. The table structure is displayed with columns: NUMBEROFBILLBOARDS, NUMBEROFAREAS, and CAMPAIGNID. The data in the table is as follows:

	NUMBEROFBILLBOARDS	NUMBEROFAREAS	CAMPAIGNID
▶	1	17	2
2	2	3	6001
3	1	2	6002
4	17	2	6003
5	9	4	6004
6	5	1	6005
7	10	1	6006
8	13	5	6007
9	14	5	6008
10	15	2	6009
11	20	5	6010
12	4	4	6011
13	16	5	6012
14	4	2	6013
15	7	3	6014

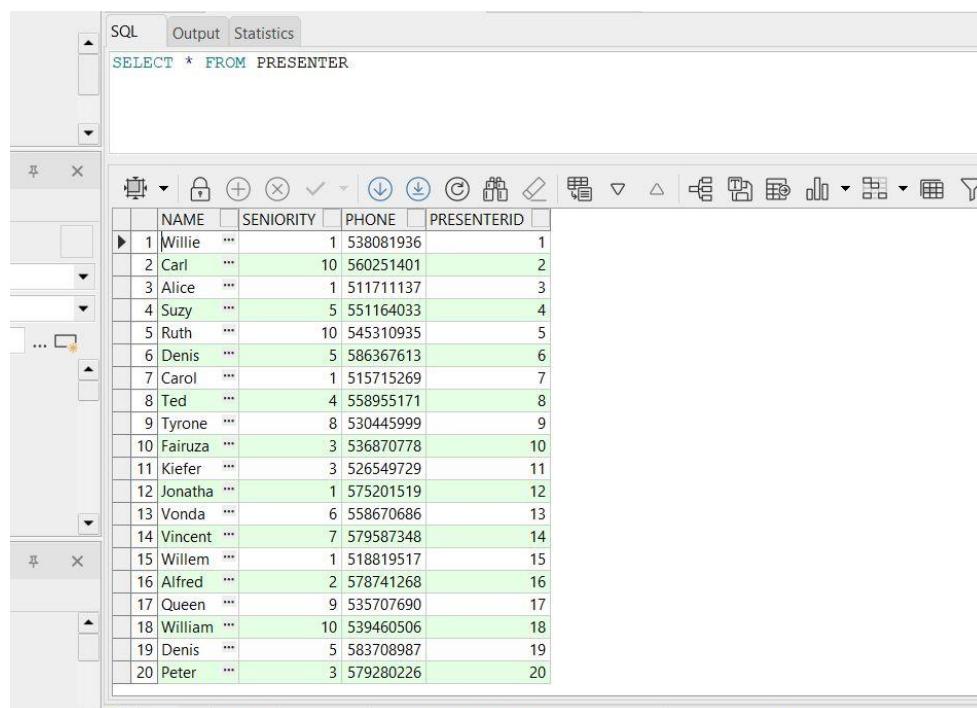
ב. טבלת מוצרים - מכילה עבור כל מוצר מספר מזהה למוצר, שם המוצר וסוג של המוצר.



The screenshot shows the SQL Server Management Studio interface with the 'Data' tab selected. A query window displays the results of the SQL command: 'SELECT * FROM Product'. The results are presented in a grid table with three columns: PRODUCTNAME, PRODUCTNUMBER, and PRODUCTKIND. The data includes 20 rows of product information, such as Wells Financial (pretzels), Diamond Pharmacy Services (bagels), and various types of bread and pastries.

	PRODUCTNAME	PRODUCTNUMBER	PRODUCTKIND
1	Wells Financial	...	1 pretzels
2	Diamond Pharmacy Services	...	2 bagels
3	Labrada Nutrition	...	3 doughnuts
4	Flow Management Technologies	...	4 pita bread
5	Parker Compound Bows	...	5 brioche
6	KeyMark	...	6 scones
7	Hudson River Bancorp	...	7 cookies
8	CIW Services	...	8 muffins
9	SHOT	...	9 doughnuts
10	PrintingForLess.com	...	10 bagels
11	TMA Resources	...	11 cinnamon rolls
12	Vertex Solutions	...	12 and 'cake'
13	Digital Visual Display Technol	...	13 quiches
14	OnStaff	...	14 scones
15	White Wave	...	15 crackers
16	Heartlab	...	16 and 'cake'
17	Universal Solutions	...	17 muffins
18	Marlabs	...	18 bread loaves
19	Mattel	...	19 granola bars
20	Biosite	...	20 muffins

ג. טבלת פرزנטורים - מכילה את שם השחקן/יידוע שמספרם את המוצר. ת"ז שלו. ניסין בתחום מספר פלאפון להתקשרות עמו.



The screenshot shows the SQL Server Management Studio interface with the 'Data' tab selected. A query window displays the results of the SQL command: 'SELECT * FROM PRESENTER'. The results are presented in a grid table with four columns: NAME, SENIORITY, PHONE, and PRESENTERID. The data includes 20 rows of presenter information, such as Willie (seniority 1, phone 538081936), Carl (seniority 10, phone 560251401), and Alice (seniority 1, phone 511711137).

	NAME	SENIORITY	PHONE	PRESENTERID
1	Willie	...	1 538081936	1
2	Carl	...	10 560251401	2
3	Alice	...	1 511711137	3
4	Suzy	...	5 551164033	4
5	Ruth	...	10 545310935	5
6	Denis	...	5 586367613	6
7	Carol	...	1 515715269	7
8	Ted	...	4 558955171	8
9	Tyrone	...	8 530445999	9
10	Fairuza	...	3 536870778	10
11	Kiefer	...	3 526549729	11
12	Jonatha	...	1 575201519	12
13	Vonda	...	6 558670686	13
14	Vincent	...	7 579587348	14
15	Willem	...	1 518819517	15
16	Alfred	...	2 578741268	16
17	Queen	...	9 535707690	17
18	William	...	10 539460506	18
19	Denis	...	5 583708987	19
20	Peter	...	3 579280226	20

ד. טבלת סוכנות פרסום- מכילה שדה של שם סוכנות הפרסום. מייל ומספר פלאפון להתקשרות עימם.

The screenshot shows the Oracle SQL Developer interface with the following details:

- Toolbar: Includes icons for Create Table, Drop Table, Select, and Data Generator.
- Tab Bar: Shows tabs for CREATE TABLE Advertising_a ..., DROP TABLE Advertising_age ..., SELECT * FROM ADVERTISING_..., and Data Generator.
- Bottom Tab: SELECT * FROM ADVERTISING_AGENCY
- Table View: Displays the ADVERTISING_AGENCY table with 17 rows of data.

	AGENCYNAME	MAIL	PHONENUMBER
1	All Star Consulting	wallace.dench@allstar.ch	578285081
2	National Heritage Academies	r.soda@nha.com	522569585
3	Alogen	marylouise@alogen.com	556705931
4	SPS Commerce	juliannap@sps.com	514953628
5	Cherokee Information Services	maria.g@cis.br	577932203
6	Office Depot	steve.katt@officedepot.com	559869214
7	Trainersoft	l.elwes@trainersoft.com	587542070
8	CARBO Ceramics	pierce.k@carboceramics.jp	558929157
9	Capital Corp.	eddie@capital.com	576486024
10	Montpelier Plastics	scott.e@montpelierplastics.com	538699291
11	Gentra Systems	m.malone@gentrasystems.com	537451784
12	Bradley Pharmaceuticals	jcale@bradleypharmaceuticals.h	521478691
13	National Home Health Care	vertical.winstone@nhhc.de	525431079
14	Aristotle	gena.h@aristotle.com	575321679
15	Mission West Properties	jake@mwp.com	528475752
16	Sweet Productions	thelma.donovan@sweetproduction	511969641
17	Street Glow	iudd.samplev@streetglow.com	576604627

ה. טבלה - Promotes- מייצגת את המוצרים שэмפורסמים בקמפיין מסוים. מכילה שתי שדות- א. מזהה קמפיין. ב. מזהה מוצר.

The screenshot shows the Oracle SQL Developer interface with the following details:

- Toolbar: Includes icons for Create Table, Drop Table, Select, and Data Generator.
- Tab Bar: Shows tabs for Adv.sql, SELECT * FROM Promotes, Data Generator, and SYS.PRODUCT.
- Bottom Tab: SELECT * FROM Promotes
- Table View: Displays the PROMOTES table with 18 rows of data.

	CAMPAIGNID	PRODUCTNUMBER
1	7774	387
2	7955	31
3	740	415
4	4877	108
5	2621	360
6	2764	876
7	2808	344
8	1703	66
9	4779	538
10	1505	199
11	813	68
12	5218	431
13	8507	533
14	8918	233
15	4563	285
16	8292	722
17	8046	42
18	8063	463

שאלות

שאילה 1

החברה רוצה לעשות קמפיין חדש עבור מוצר מסווג חלה. יוצי הפרסום יעוזו למתג את המוצר בעזרת פרזנטור. בראצוננו לבדוק איזה פרזנטור בעל אחוזי הצלחה הגבוהים ביותר בקמפיינים עבור מוצרים מסווגי חלה.

The screenshot shows a SQL query window and a results grid. The query is:

```
SQL Output Statistics
select presenterid, count(P2.PRODUCTNAME)
from campaign natural join promotes natural join product P2
where P2.Productkind LIKE '%challah%'
group by presenterid
order by count(*) Desc;
```

The results grid displays the count of products for each presenterid, ordered by count in descending order. The data is as follows:

PRESENTERID	COUNT(P2.PRODUCTNAME)
1	347
2	136
3	258
4	408
5	458
6	495
7	431
8	141
9	75
10	165
11	107
12	57
13	471
14	464
15	320
16	248
17	390
18	361
19	423

שאילה 2

בעקבות עליית מחירי הפרסום בערוצי המדיה, המאפייה מעוניינת ליעל את הפרסום בטלוויזיה. לשם כך נבדוק אלו קמפיינים טלוויזיוניים היי עם אחוז צפיה גבוהים אמנים צוין הקמפיין שלהם צלח פחות מאשר המזופה. השאילתא בודקת ALSO סוג מוצרים יש הפרש גדול בין אחוז הצפיה שלהם לבין הצלחת הקמפיין במחירות.

The screenshot shows a SQL query window and a results grid. The query is:

```
SQL Output Statistics
select productKind, avg(rating - successrates)
from tv_campaign natural join campaign natural join promotes natural join product
group by productKind
order by avg(rating - successrates) DESC
```

The results grid displays the average rating minus success rates for each product kind, ordered by the average rating minus success rates in descending order. The data is as follows:

PRODUCTKIND	Avg(Rating-SuccessRates)
?clairs	1.625
pretzels	1.125
biscuits	1.04347826086957
cupcakes	0.9
baguettes	0.760869565217391
sourdough	0.551020408163265
crackers	0.542857142857143
muffins	0.45
quiches	0.34375
cookies	0.3333333333333333
brownies	0.32
Danish pastries	0.203703703703704
fruit pies	0.196428571428571
Croissants	0.185185185185185
and 'cake'	0.18
pizza crust	0.0227272727272727
pita bread	0
bagels	-0.101180361703128

שאילתה 3

בעקבות חילופי זכיניהם מנהל המאפייה התחלף והוא מעוניין לדעת באלו סוגים מוצריים כדאי השקיע קמפני ל夸רת תקופת החגים. ניצור שאלה שבודקת לאלו סוגים מוצריים יש הכל הרבה קמפניים בחודש ספטמבר.

```

Select productkind, count(*)
From product natural join campaign natural join promotes
where EXTRACT(MONTH FROM startDate)=9 and EXTRACT(MONTH FROM endDate)=10
group by productkind
order by count(*) DESC

```

PRODUCTKIND	COUNT(*)
1 fruit pies	8
2 biscuits	6
3 quiches	6
4 energy balls	6
5 sourdough	6
6 protein bars	5
7 scones	5
8 and 'cake'	4
9 Danish pastries	4
10 pretzels	4
11 cookies	3
12 cinnamon rolls	3
13 doughnuts	3
14 bagels	3
15 muffins	3
16 challah	3

שאילתה 4

החברה מעוניינת לעשות קמפני מיוחד, בה תיקח חמיש מוצריים מסוימים שיש להם ממוצע היכי גבוה בקמפניים שהפיקו למוצרים מאותו סוג בעבר. נחפש את החמש המוצריים היכי מוצלחים מהבחן הצעת.

```

SELECT productkind, productname, avg(successrates) as average_rating
FROM campaign natural join product natural join promotes
GROUP BY productkind, productname
HAVING avg(successrates) = (
    SELECT MAX(avg_rating)
    FROM (
        SELECT productname, AVG(successrates) as avg_rating,
        FROM campaign natural join product natural join promotes
        GROUP BY productname,
    )
);

```

דו"חות וגרפים

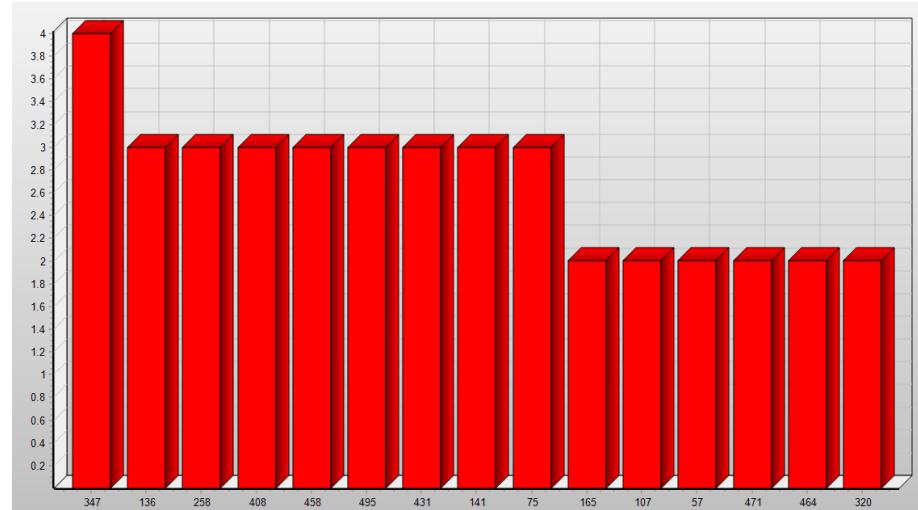
שאילתא 1

עבור השאלה המחזירה את הפרנצטורים המוצלחים ביותר עבור קמפיינים של מוצר חלה. נציג את הפרנצטורים ואת מידת הצלחה שלהם בגרף:

```
select presenterid, count(P2.PRODUCTNAME)
from campaign natural join promotes natural join product P2
where P2.Productkind LIKE '%challah%'
group by presenterid
order by count(*) Desc
```

השאילתא:

הגרף:



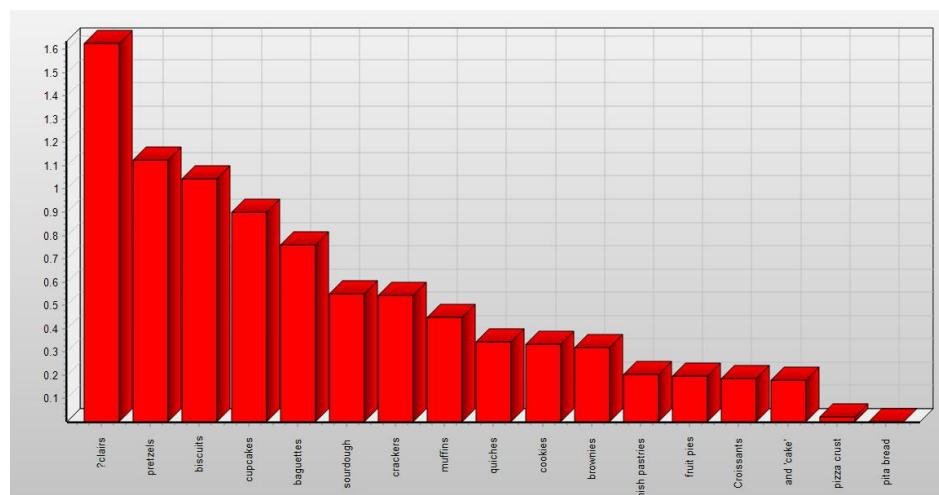
שאילתא 2

נציג גרף המציג את סוג המוצרים ואת הפרש בין אחוזי הצפיה שלהם לבין דירוג הצלחת הקמפיין שלהם ע"י קר נוכל לראות ויזואלית את סוג המוצרים שהקמפיינים הטולזיוניים שלהם פחות צלחו.

```
select productKind, avg(rating - successrates)
from tv_campaign natural join campaign natural join promotes natural join product
group by productKind
order by avg(rating - successrates) DESC
```

השאילתא:

הגרף:



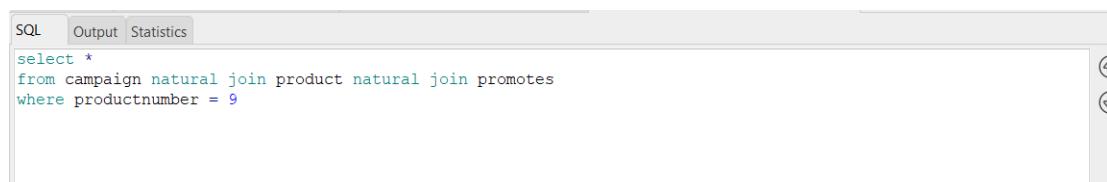
עדכון בסיס הנתונים באמצעות תכניות בשפת SQL/PL

עדכון

נעשה עדכון לכל הקמפיינים שמכילים את המוצר שמספרו הוא 9, ונחליף את הציון של הקמפיינים
האלה להיות 8 תמיד

```
UPDATE campaign
SET successrates = 8.0
WHERE campaignid in (
    select campaignid
    from campaign natural join product natural join promotes
    where productnumber = 9);
Commit;
```

ונבדוק האם זה עבד:



The screenshot shows the SQL tab of the Oracle SQL Developer interface. The query entered is:

```
select *
from campaign natural join product natural join promotes
where productnumber = 9
```

Below the query, the results are displayed in a grid:

CAMPAIGNID	PRODUCTNUMBER	STARTDATE	ENDDATE	SUCCESSRATES	AGENCYNAME	PRESENTERID	PRODUCTNAME
1	4375	9/2/2016	12/20/2019	8	Bradley Pharmaceuticals5	...	SHOT
2	101	9/1/2016	1/22/2021	8	Heartlab9	...	424 SHOT
3	4055	9/8/2015	6/30/2020	8	Verizon Communications	...	336 SHOT
4	2693	9/4/2016	8/22/2020	8	Active Services	...	294 SHOT
5	722	9/2/2016	12/16/2022	8	Creditors Interchange	...	204 SHOT
6	5455	9/3/2017	2/18/2019	8	Laboratory Management Systems1	...	320 SHOT
7	481	9/28/2015	10/8/2019	8	Shuffle Master	...	151 SHOT
8	4660	9/4/2016	9/23/2020	8	Hencie4	...	214 SHOT
9	6056	9/1/2018	7/12/2022	8	USA Environmental Management7	...	98 SHOT
10	2697	9/6/2017	3/1/2022	8	Marathon Heater6	...	122 SHOT
11	4704	9/11/2017	10/21/2020	8	Merck & Co.10	...	303 SHOT
12	2303	9/12/2015	8/15/2020	8	WorldCom1	...	245 SHOT

מחיקה

התגלגה שריה טעות, ובעצם בפועל מוצר שהיה רשום בתוך קמפיין לא השתתף, נרצה להוריד אותו מהקמפיין:



The screenshot shows the SQL tab of the Oracle SQL Developer interface. The query entered is:

```
select *
from promotes
```

Below the query, the results are displayed in a grid:

CAMPAIGNID	PRODUCTNUMBER
1	7774
2	7955
3	740
4	4877
5	2621
6	2764
7	2808
8	1703
9	4779
10	1505
11	813
12	5218
13	8507
14	8918
15	4563

עכשו נמחק את המוצר הראשון מהקמפיין הזה:



The screenshot shows the SQL tab of the Oracle SQL Developer interface. The query entered is:

```
delete from promotes
where campaignid = 7774 and productnumber = 387;
commit;
```

ונחפש לראות האם זה באמת נמוך:

The screenshot shows a database interface with a SQL editor at the top containing the query:

```
SQL Output Statistics
select *
from promotes
```

Below the editor is a table viewer displaying the results of the query:

	CAMPAIGNID	PRODUCTNUMBER
▶ 1	7955	31
2	740	415
3	4877	108
4	2621	360
5	2764	876
...

שימוש ב rollback

בעל המאפייה גילה שיש לנו מוצרים שנשארו מהם עודפים וכן ברצונו להוסיף מוצרים אלו לקמפיין מסוים שרצ עכשו. נסיף לקמפיין קיימ עד מוצרים באמצעות פקודה `insert`

The screenshot shows a database interface with a SQL editor containing a transaction script:

```
SQL Output Statistics
insert into promotes(campaignid, productnumber)
values (7955, 415);

insert into promotes(campaignid, productnumber)
values (7955, 416);

insert into promotes(campaignid, productnumber)
values (7955, 417);

savepoint insert_sp;

update promotes
set productnumber = 418
where campaignid = 7955 and productnumber = 417;
|
rollback to savepoint insert_sp;

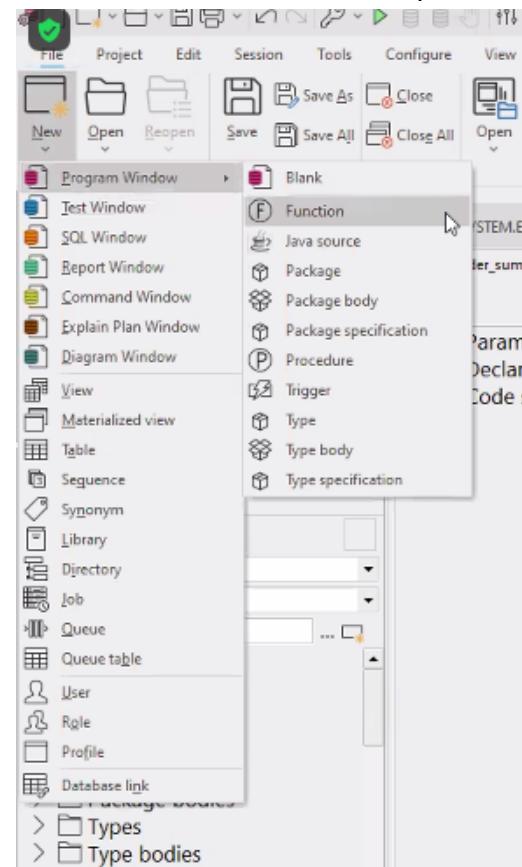
select * from promotes where campaignid = 7955
```

Below the editor is a table viewer displaying the results of the final select query:

	CAMPAIGNID	PRODUCTNUMBER
▶ 1	7955	31
2	7955	415
3	7955	416
4	7955	417

שאלות ב- PLSQL

נפתח פונקציה חדשה בשפת PLSQL:

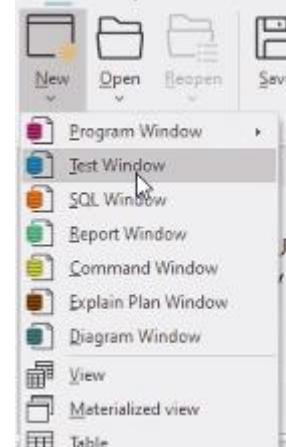


נבחר את כל סוגי מוצרים שעבורם יש יותר קמפיינים מהמינימום המבוקש.

הפונקציה מקבלת מספר של מינימום קמפיינים ומחזירה את סוגי המוצרים שמשם הקמפיינים שלהם גודלה ממספר זה.

```
create or replace function get_amount_by_kind
    (minimumCampaigns in Number)
    RETURN SYS_REFCURSOR
IS
    v_cursor SYS_REFCURSOR;
begin
    open v_cursor for
        select productKind, count(*)
        from campaign natural join product natural join promotes
        group by productKind
        having minimumCampaigns < count(*)
        order by count(*) DESC;
    return v_cursor;
END;
```

קריאה והדפסה של התוכן:
נפתח חלון חדש שדרכו נקרא לפונקציה:



נירץ אתה הפונקציה על מנת שתיה מוכרת למערכת.
בחילון זה נצהיר על המשתנים שיקבלו את תוצאות השאלה:
1. Cursor קולט את תוכו את ערך החזרה מהפונקציה. נחלץ את ערך החזרה שמכיל שני

- פרמטרים:
- 2. שם
- 3. כמות

את התוצאה נדפס.

```

1 -- Created on 6/21/2023 by YEHUDA
2 declare
3     result_cursor SYS_REFCURSOR;
4     productKind Varchar (30);
5     amount NUMBER;
6 begin
7     result_cursor := get_amount_by_kind(300);
8
9     LOOP
10        fetch result_cursor into productKind, amount;
11        exit when result_cursor%NOTFOUND;
12
13        DBMS_OUTPUT.put_line( 'Product Kind: ' || productKind ||
14                               'Amount: ' || amount);
15
16    END LOOP;
17
18    close result_cursor;
19
20
21 end;

```

ההדפסה:

The screenshot shows a database trace window with the following content:

```

Test script DBMS Output Statistics Profiler Trace
Clear Buffer size 10000 Enabled
Product Kind: energy ballsAmount: 616
Product Kind: pretzelsAmount: 504
Product Kind: cinnamon rollsAmount: 503
Product Kind: fruit piesAmount: 497
Product Kind: and 'cake'Amount: 495
Product Kind: Danish pastriesAmount: 471
Product Kind: browniesAmount: 462
Product Kind: pizza crustAmount: 442
Product Kind: bagelsAmount: 431
Product Kind: pita breadAmount: 429
Product Kind: biscuitsAmount: 428
Product Kind: sconesAmount: 426
Product Kind: baguettesAmount: 424
Product Kind: doughnutsAmount: 419
Product Kind: cookiesAmount: 419
Product Kind: focacciaAmount: 407
Product Kind: CroissantsAmount: 391
Product Kind: granola barsAmount: 364
Product Kind: crackersAmount: 359
Product Kind: sourdoughAmount: 358
Product Kind: naanAmount: 338
Product Kind: quichesAmount: 333
Product Kind: cupcakesAmount: 330
Product Kind: bread loavesAmount: 329

```

פקודת Grant

בעזרת פקודות מהציג זהה נוכל להוסיף הרשות למשתמשים נוספים:

```
grant select on campaign to Michael;
grant select on product to Joe;
grant update on promotes to Eliezer;
```

ובאמצעות פקודות מסוג revoke נוכל להוציא את הרשותות האלה:

```
revoke select on campaign to Michael;
revoke select on product to Joe;
revoke update on promotes to Eliezer;
```

זמן ריצה

נבדוק את זמן ריצה של כמה שאילתות:

```
select productkind, avg(rating - successrates)
from tv_campaign natural join campaign natural join promotes natural join product
group by productkind
order by avg(rating - successrates) DESC;
```

The screenshot shows the Oracle SQL Developer interface. At the top, there is a toolbar with various icons. Below the toolbar is a results grid displaying the output of the query. The grid has two columns: 'PRODUCTKIND' and 'AVG(RATING-SUCCESSRATES)'. The data rows are numbered from 1 to 17. The results are as follows:

PRODUCTKIND	AVG(RATING-SUCCESSRATES)
1 ?clairs	1.625
2 pretzels	1.125
3 biscuits	1.04347826086957
4 cupcakes	0.9
5 baguettes	0.760869565217391
6 sourdough	0.551020408163265
7 crackers	0.542857142857143
8 muffins	0.45
9 quiches	0.34375
10 cookies	0.333333333333333
11 brownies	0.32
12 Danish pastries	0.203703703703704
13 fruit pies	0.196428571428571
14 Croissants	0.185185185185185
15 and 'cake'	0.18
16 pizza crust	0.0227272727272727
17 pita bread	0

At the bottom of the interface, there is a status bar showing the session information: system AS SYSDBA, [5:57:42 PM], 17 rows selected in 0.129 seconds (more...).

```
select productkind, avg(rating - successrates)
from tv_campaign natural join campaign natural join promotes
natural join product
group by productkind
order by avg(rating - successrates) DESC
```

והנה זמן ריצה של הפקנץיה שכתבנו לעיל:

The screenshot shows the Oracle SQL Developer interface. At the top, there is a toolbar with various icons. Below the toolbar is a script editor window containing a PL/SQL block. The code is as follows:

```
-- Created on 6/21/2023 by YEHUDA
declare
  result_cursor SYS_REFCURSOR;
  productKind Varchar (30);
  amount NUMBER;
begin
  result_cursor := get_amount_by_kind(300);
  LOOP
    fetch result_cursor into productKind, amount;
    exit when result_cursor%NOTFOUND;
    DBMS_OUTPUT.put_line( 'Product Kind: ' || productKind ||
                          'Amount: ' || amount);
  END LOOP;
  close result_cursor;
end;
```

Below the script editor is a variable viewer window showing a single row with a checked checkbox. At the bottom of the interface, there is a status bar showing the session information: system AS SYSDBA, [10:01:37 AM], Executed in 0.012 seconds.

```

declare
    result_cursor SYS_REFCURSOR;
    productKind Varchar (30);
    amount NUMBER;
begin
    result_cursor := get_amount_by_kind(300);

    LOOP
        fetch result_cursor into productKind, amount;
        exit when result_cursor%NOTFOUND;

        DBMS_OUTPUT.put_line( 'Product Kind: ' || productKind ||
                            'Amount: ' || amount);

    END LOOP;

    close result_cursor;

end;

```

אינדקסים

מוסיף לכמה שדות אינדקסים, ורואים שיש השפעה על זמן הריצה:

```

create index idx_campaign_id on campaign(campaignid);
create index product_id on product(productnumber);

```

פרוצדורה

1 פרוצדורה

מנכ"ל החברה הגיע לבקשת מושרד הקמפיינים ורוצה לדעת כמה מוצרים השתתפו בקמפיין עם תוצאות מעולות. נממש פרוצדורה שתמצא את כמה המוצרים הנ"ל.

```

declare procedure excelent_products_amount
as
    amount Number;

BEGIN
    SELECT count (*) into amount
    From campaign natural join product natural join promotes
    where successrates >9;

    DBMS_OUTPUT.PUT_LINE('The number of products that
        participated in excellent campaigns is: ' || amount);
End;

Begin|
    excelent_products_amount;
END;

```



```

declare procedure excelent_products_amount
as
    amount Number;

BEGIN
    SELECT count (*) into amount
    From campaign natural join product natural join
promotes
        where successrates >9;

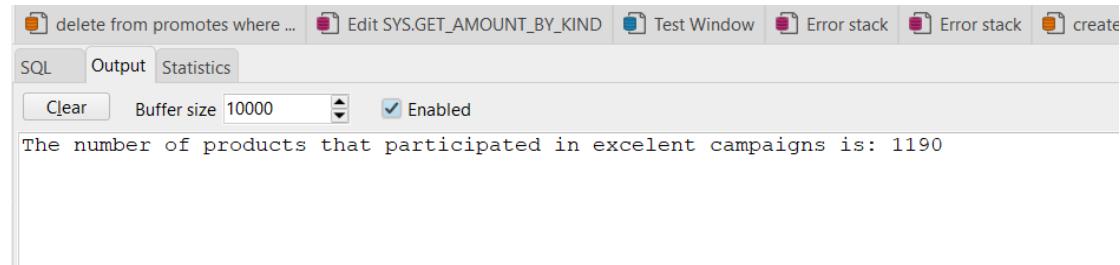
    DBMS_OUTPUT.PUT_LINE('The number of products that
        participated in excellent campaigns is: ' || amount);
End;

Begin
    excelent_products_amount;

```

END;

הפלט של התוכנית הוא:



The screenshot shows the Oracle SQL developer interface with the following details:

- Toolbar:** Includes icons for delete from, edit, test window, error stack, and create.
- Tab Bar:** Shows tabs for SQL, Output, and Statistics. The SQL tab is selected.
- Tool Buttons:** Clear, Buffer size (set to 10000), and Enabled checkbox.
- Output Window:** Displays the result of the procedure execution: "The number of products that participated in excellent campaigns is: 1190".

פראצ'זרה 2

הגיע סוף שנה, ועכשו מנכ"ל החברה הולך לبشر את החדשות שכולן חיכו להם: מי הבמאי שגרם להכי הרבה הצלחות בחברה (בממוצע של כל הקמפיינים שלו), נमמש פראצ'זרה שתמצא אותו:

```

Declare procedure find_best_presenter
as
    presenterName varchar(30);

Begin
    select name into presenterName
    from presenter P1
    where P1.presenterid = (
        select presenterId
        from presenter natural join campaign P2
        group by presenterid
        having avg(successrates) = (
            select max(average)
            from (
                select avg(successrates) as average
                from presenter natural join campaign P2
                group by presenterid)
            ));

```



```

        DBMS_OUTPUT.PUT_LINE('The Best presenter is ' || presenterName);
End;

Begin
    find_best_presenter;
END;
```



```

Declare procedure find_best_presenter
as
    presenterName varchar(30);

Begin
    select name into presenterName
    from presenter P1
    where P1.presenterid = (
        select presenterId
        from presenter natural join campaign P2
        group by presenterid
        having avg(successrates) = (
            select max(average)
            from (
                select avg(successrates) as average
                from presenter natural join campaign P2
                group by presenterid)
            )));

```

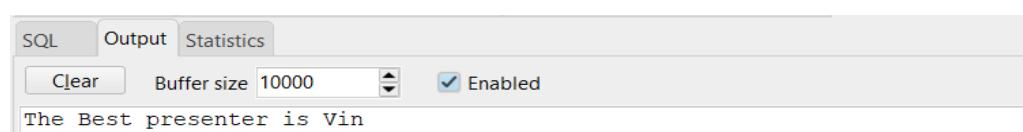


```

DBMS_OUTPUT.PUT_LINE('The Best presenter is ' || presenterName);
End;

Begin
    find_best_presenter;
END;
```

והפלט של התוכנית הוא:



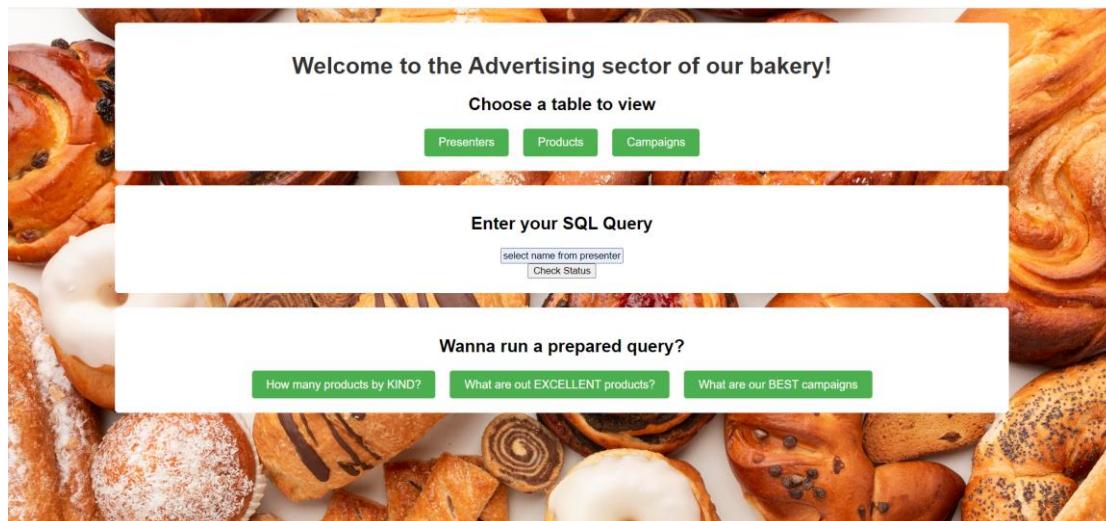
יישום ממשק משתמש

יצרנו ממשק משתמש **web** על מנת לגשת לבסיס הנתונים. לשם כך כתבנו תכנית בפייתון עם ספריית **Flask** ועיצבנו עמוד בעזרת **HTML, JavaScript**.

הציגת המערכת - ממשק המשתמש

המשתמש נכנס למסך הראשי בו מוצגים לו שלושה תיבות:

- א. תיבה המכילה לחיצים להציג הנתונים בטבלאות
- ב. תיבה המכילה תיבת טקסט בה הוא יכול להקליד בקשה SQL בהתאם לרצונו החופשי.
- ג. תיבה המכילה לחיצים של בקשות מוגנות להציג מידע שימושי.



היבת התצוגה

הציגת טבלת פרזנטורים

בלחיצה על הלחץ **Presenters** נפתח חלון בו מופיעה טבלת הפרזנטורים הקיימים במערכת:

Showing the table for presenter

name	seniority	phone	presenter id
Willie	1	538081936	1
Carl	10	560251401	2
Alice	1	511711137	3
Suzy	5	551164033	4
Ruth	10	545310935	5
Denis	5	586367613	6
Carol	1	515715269	7
Ted	4	558955171	8
Tyrone	8	530445999	9
Fairuza	3	536870778	10
Kiefer	3	526549729	11
Jonatha	1	575201519	12
Vonda	6	558670686	13
Vincent	7	579587348	14
Willem	1	518819517	15
Allred	2	578741268	16

הציגת טבלת המוצרים
בלחיצה על הלחצן Products נפתח חלון בו מופיעה טבלת המוצרים הקיימים במערכת:

Showing the table for product

product name	product number	product kind
Wells Financial	1	pretzels
Diamond Pharmacy Services	2	bagels
Labrada Nutrition	3	doughnuts
Flow Management Technologies	4	pita bread
Parker Compound Bows	5	brioche
KeyMark	6	scones
Hudson River Bancorp	7	cookies
CIW Services	8	muffins
SHOT	9	doughnuts
PrintingForLess.com	10	bagels
TMA Resources	11	cinnamon rolls
Vertex Solutions	12	and 'cake'
Digital Visual Display Technol	13	quiches
OnStaff	14	scones
White Wave	15	crackers
Heartlab	16	and 'cake'
...

הציגת טבלת קמפיינים
בלחיצה על הלחצן Campaigns נפתח חלון בו מופיעה טבלת הקמפיינים הקיימים במערכת:

Showing the table for campaign

start date	end date	success rates	campaign id	agency name	presenter id
2018-01-02	2020-09-22	4	1	Banfe Products9	316
2018-08-05	2020-09-17	2	2	J.C. Malone Associates2	399
2018-07-20	2021-01-16	3	3	Signal Perfection2	86
2015-10-12	2022-01-10	1	4	Questar Capital3	78
2016-01-08	2020-06-19	10	5	Bayer4	225
2016-09-23	2021-05-01	9	6	IBM Corp.8	493
2016-05-10	2020-07-10	10	7	Glacier Bancorp5	435
2018-04-17	2022-06-08	7	8	Joe Lombardo Plumbing + Heatin	9
2015-01-04	2019-07-07	1	9	Dillard's6	87
2015-09-21	2020-07-05	3	10	Capital Corp.	469
2016-03-26	2021-12-31	10	11	Alternative Business Systems7	115
2015-11-30	2019-09-20	2	12	Loss Mitigation Services	100
2016-07-24	2019-06-26	5	13	Strategic Management Initiativ	248
2018-04-20	2019-06-08	10	14	Freedom Medical10	92
2016-06-23	2022-10-19	1	15	Scott Pipitone Design4	391
2015-02-26	2021-03-08	8	16	Access US	109
2014-10-24	2019-01-29	0	17

היבת השאלה החופשית
בתיבה זו ניתנת האפשרות למשתמש לכתוב שאלתא חופשית. ובלחיצה על **'ויצג'** המידע שבקש. לדוגמה עבור השאלה: select name from presenter **'ויצג'**:



Showing the table for select name from presenter

NAME
Willie
Carl
Alice
Suzy
Ruth
Denis
Carol
Ted
Tyrone
Fairuza
Kiefer
Jonatha
Vonda
Vincent
Willem
Alfred

תיבת השאלות בנויה
בתיבה זו ישנו לחצנים של שלוש שאלות נפוצות.

שאילתת סוג המוצרים
בלחיצה על לחץ זה תופיע טבלה המכילה את שמות של סוגים המוצרים ואת מס' המוצרים הקיימים
 מכל סוג.

How many products by KIND?

Showing the table for kind

PRODUCTKIND	COUNT(*)
energy balls	616
pretzels	504
cinnamon rolls	503
fruit pies	497
and 'cake'	495
Danish pastries	471
brownies	462
pizza crust	442
bagels	431
pita bread	429
biscuits	428
scones	426
baguettes	424
doughnuts	419
cookies	419
focaccia	407

שאילתת מוצרים עם קמפיינים מוצלחים
בלחיצה על לחץ זה תופיע טבלה המכילה את פרטי המוצרים שהיו להם קמפיינים מוצלחים. עם עמודה שבה יופיע מס' הקמפיינים המוצלחים שהיו לכל מוצר.

What are our EXCELLENT products?

Showing the table for our products with excellent ratings

PRODUCTNUMBER	PRODUCTNAME	PRODUCTKIND	amount of excellent campaigns
1	Wells Financial	pretzels	2
2	Diamond Pharmacy Services	bagels	1
3	Labrada Nutrition	doughnuts	4
4	Flow Management Technologies	pita bread	2
5	Parker Compound Bows	brioche	2
6	KeyMark	scones	1
10	PrintingForLess.com	bagels	1
11	TMA Resources	cinnamon rolls	1
12	Vertex Solutions	and 'cake'	1
13	Digital Visual Display Technol	quiches	1
14	OnStaff	scones	1
16	Heartlab	and 'cake'	4
18	Mariabs	bread loaves	2
19	Mattel	granola bars	1
20	Biosite	muffins	3
21	Bigdough.com	biscuits	1
...

שאילתת קמפיינים מוצלחים
בלחיצה על לחץ זה תופיע טבלה המכילה את סוג המוצרים שהיו להם קמפיינים מוצלחים. בעמודה הראשונה יהיה כתוב סוג המוצר ובעמודה השנייה יהיה כתוב הממוצע של הקמפיינים המוצלחים עבור סוג המוצר הנ"ל. הנתונים יהיו ממוחאים בסדר יורד ע"פ עמודה זו.

What are our BEST campaigns

Showing the table for our best campaigns

PRODUCTKIND	AVG(RATING)
?clairs	7.125
pretzels	6.229166666666667
biscuits	5.478260869565218
cupcakes	5.9
baguettes	5.934782608695652
sourdough	6.285714285714286
crackers	5.714285714285714
muffins	5.75
quiches	5.625
cookies	6.055555555555555
brownies	5.94
Danish pastries	5.796296296296297
fruit pies	5.428571428571429
Croissants	5.833333333333333
and 'cake'	5.02
pizza crust	5.795454545454546

הציגת המערכת - הקוד

קוד הפיתון של העמוד הראשי:

```

from flask import Flask, render_template, request, abort
import Oracle_connection

app = Flask(__name__)

entities = [
    {'name': 'presenter', 'display_name': 'Presenter'},
    {'name': 'product', 'display_name': 'Product'},
    {'name': 'agency', 'display_name': 'Agency'},
    {'name': 'tv_campaign', 'display_name': 'TV Campaign'},
    {'name': 'billboard_campaign', 'display_name': 'Billboard Campaign'},
    {'name': 'internet_campaign', 'display_name': 'Internet Campaign'}
]

@app.route('/')
def menu():
    return render_template('main.html')

# get table from post request, then get table data from oracle connection
@app.route('/table', methods=['GET'])
def table():
    print("request.form")
    # get table name from post request
    table_name = request.args.get('button')
    print(table_name)
    # get table data from oracle connection
    if table_name == 'presenter':
        table_data = Oracle_connection.convert_to_json(*Oracle_connection.get_presenters())
    elif table_name == 'product':
        table_data = Oracle_connection.convert_to_json(*Oracle_connection.get_products())
    elif table_name == 'campaign':
        table_data = Oracle_connection.convert_to_json(*Oracle_connection.get_campaigns())
    else:
        abort(404)

    return render_template('display_table.html', table_data=table_data, table_name=table_name)

@app.route('/query', methods=['GET'])
def query():
    print("request.form")
    # get table name from post request
    query = request.args.get('query')
    print(query)
    # get table data from oracle connection
    table_data = Oracle_connection.convert_to_json(*Oracle_connection.apply_query(query))
    return render_template('display_table.html', table_data=table_data, table_name=query)

@app.route('/prepared', methods=['GET'])
def prepared():
    # get table name from post request
    query = request.args.get('query')
    print(query)
    if query == 'kind':
        table_data = Oracle_connection.convert_to_json(*Oracle_connection.get_amount_by_kind())
        return render_template('display_table.html', table_data=table_data, table_name=query)
    elif query == 'excellent':
        table_data = Oracle_connection.convert_to_json(*Oracle_connection.get_excellent_products())
        return render_template('display_table.html', table_data=table_data, table_name="our products with excellent "
                                         "ratings")
    elif query == 'campaigns':
        table_data = Oracle_connection.convert_to_json(*Oracle_connection.best_campaigns())
        return render_template('display_table.html', table_data=table_data, table_name="our best campaigns")

if __name__ == '__main__':
    app.run(debug=True)

```

קוד הפיתון עבור הצגת הטבלאות:

```

1  from datetime import datetime
2
3  import cx_Oracle
4  import json
5
6  connection = cx_Oracle.connect(
7      user="sys",
8      password="password",
9      mode=cx_Oracle.SYSDBA)
10
11 cursor = connection.cursor()
12
13 # define columns
14 campaign_columns = ["start date", "end date", "success rates", "campaign id", "agency name", "presenter id"]
15 presenter_columns = ["name", "seniority", "phone", "presenter id"]
16 product_columns = ["product name", "product number", "product kind"]
17
18 # define functions that will return tables in json format
19
20 def get_campaigns():
21     cursor.execute("SELECT * FROM campaign")
22     result = cursor.fetchall()
23     return campaign_columns, result
24
25
26 def get_presenters():
27     cursor.execute("SELECT * FROM presenter")
28     result = cursor.fetchall()
29     return presenter_columns, result
30
31
32 def get_products():
33     cursor.execute("SELECT * FROM product")
34     result = cursor.fetchall()
35     return product_columns, result
36
37
38 cursor.execute(query)
39 result = cursor.fetchall()
40 print(result)
41 columns = [i[0] for i in cursor.description]
42 return columns, result
43
44
45 def get_excellent_products():
46     query = """SELECT productnumber, productname, productkind, count(*)
47         From campaign natural join product natural join promotes
48         where successrates > 9
49         group by productnumber, productname, productkind
50         """
51
52     cursor.execute(query)
53     result = cursor.fetchall()
54     print(result)
55     columns = [i[0] for i in cursor.description]
56     columns[3] = "amount of excellent campaigns"
57     return columns, result
58
59
60 def best_campaigns():
61     query = """select productkind, avg(rating)
62     from tv_campaign natural join campaign natural join promotes natural join product
63     group by productkind
64     order by avg(rating - successrates) DESC
65         """
66
67     cursor.execute(query)
68     result = cursor.fetchall()
69     print(result)
70     columns = [i[0] for i in cursor.description]
71     return columns, result
72
73
74

```

```

def convert_to_json(column_names, result):
    # Create an empty list to store the row data
    json_data = []

    for row in result:
        data = {}
        for i, column in enumerate(row):
            # Check if column is a datetime object
            if isinstance(column, datetime):
                # Convert datetime object to yyyy-mm-dd format
                column = column.strftime("%Y-%m-%d")
            # Add column value to the data dictionary
            data[column_names[i]] = column

        # Add the data dictionary to the JSON list
        json_data.append(data)

    # Serialize the JSON list
    json_result = json.dumps(json_data)

    # save json to file
    with open('static/data.json', 'w') as outfile:
        json.dump(json_data, outfile)

    return json_result

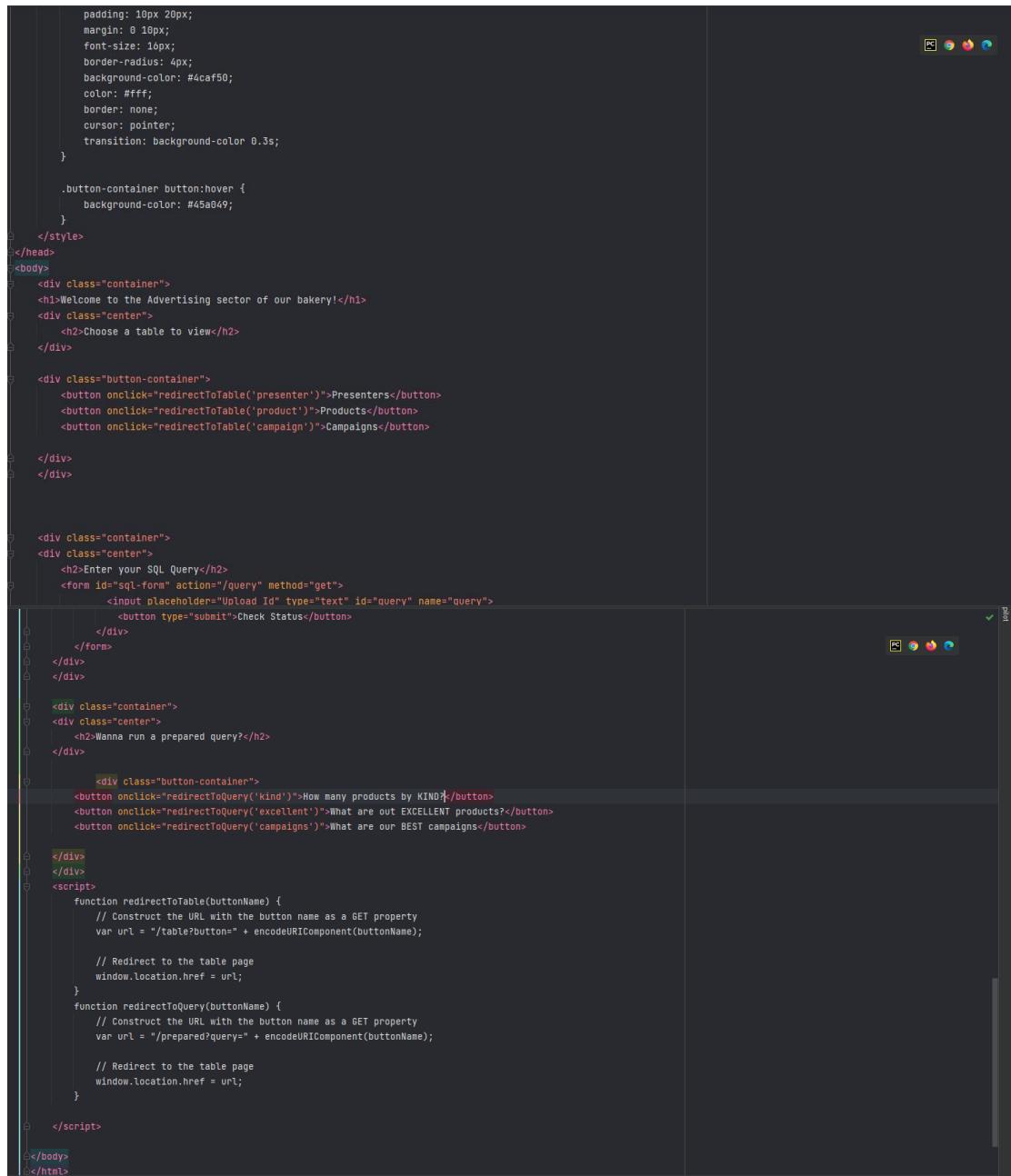
```

קובץ HTML של הדף הראשי:

```

<!DOCTYPE html>
<html>
<head>
    <title>Home Page</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f1f1f1;
            margin: 0;
            padding: 20px;
            background-image: url('../static/bakery.jpg');
            background-size: cover;
            background-repeat: no-repeat;
        }
        .container {
            width: 80%;
            margin: 0 auto;
            background-color: #fff;
            padding: 20px;
            border-radius: 5px;
            margin-bottom: 20px;
        }
        h1 {
            text-align: center;
            color: #333;
        }
        .center {
            text-align: center;
        }
        .button-container {
            display: flex;
            justify-content: center;
            margin-top: 20px;
        }
    </style>

```



```

padding: 10px 20px;
margin: 0 10px;
font-size: 16px;
border-radius: 4px;
background-color: #4CAF50;
color: #fff;
border: none;
cursor: pointer;
transition: background-color 0.3s;
}

.button-container button:hover {
background-color: #45a049;
}

</style>
</head>
<body>
<div class="container">
<h1>Welcome to the Advertising sector of our bakery!</h1>
<div class="center">
<h2>Choose a table to view</h2>
</div>

<div class="button-container">
<button onclick="redirectToTable('presenter')">Presenters</button>
<button onclick="redirectToTable('product')">Products</button>
<button onclick="redirectToTable('campaign')">Campaigns</button>
</div>
</div>

<div class="container">
<div class="center">
<h2>Enter your SQL Query</h2>
<form id="sql-form" action="/query" method="get">
<input placeholder="Upload Id" type="text" id="query" name="query">
<button type="submit">Check Status</button>
</form>
</div>
</div>

<div class="button-container">
<button onclick="redirectToQuery('kind')">How many products by KIND?</button>
<button onclick="redirectToQuery('excellent')">What are our EXCELLENT products?</button>
<button onclick="redirectToQuery('campaigns')">What are our BEST campaigns?</button>
</div>
</div>
</div>
<script>
function redirectToTable(buttonName) {
// Construct the URL with the button name as a GET property
var url = "/table?button=" + encodeURIComponent(buttonName);

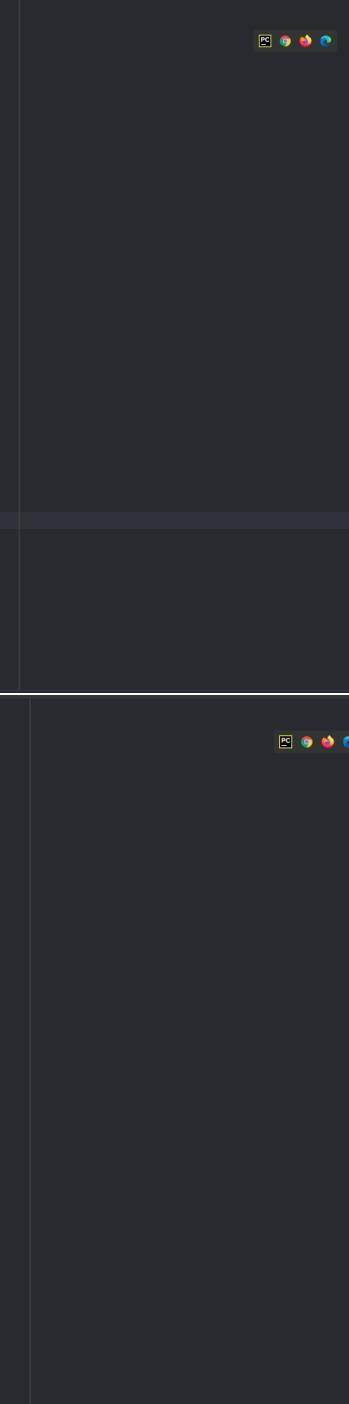
// Redirect to the table page
window.location.href = url;
}

function redirectToQuery(buttonName) {
// Construct the URL with the button name as a GET property
var url = "/prepared?query=" + encodeURIComponent(buttonName);

// Redirect to the table page
window.location.href = url;
}
</script>
</body>
</html>

```

קובץ HTML של הצגת הטבלאות



```

<!DOCTYPE html>
<html>
  <head>
    <title>Table Viewer</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        padding: 20px;
      }

      table {
        border-collapse: collapse;
        width: 100%;
      }

      th, td {
        border: 1px solid #ddd;
        padding: 8px;
      }

      th {
        background-color: #f2f2f2;
      }

      .table-container {
        overflow-x: auto;
      }
    </style>
  </head>
  <body>
    <h1>Showing the table for {{table_name}}</h1>
    <div class="table-container">
      <table id="table-data"></table>
    </div>

    <script type="module">
      import data from '../static/data.json' assert { type: 'json' };
      console.log(data);

      import data from '../static/data.json' assert { type: 'json' };
      console.log(data);

      // JSON data representing the table
      var jsonData = data;

      // Function to generate the table from JSON data
      function generateTable(data) {
        var table = document.getElementById('table-data');

        // Create table header
        var headerRow = table.insertRow();
        for (var key in data[0]) {
          if (data[0].hasOwnProperty(key)) {
            var headerCell = document.createElement('th');
            headerCell.textContent = key;
            headerRow.appendChild(headerCell);
          }
        }

        // Create table rows
        for (var i = 0; i < data.length; i++) {
          var row = table.insertRow();
          for (var key in data[i]) {
            if (data[i].hasOwnProperty(key)) {
              var cell = row.insertCell();
              cell.textContent = data[i][key];
            }
          }
        }
      }

      // Call the function to generate the table
      generateTable(jsonData);
    </script>
  </body>
</html>

```