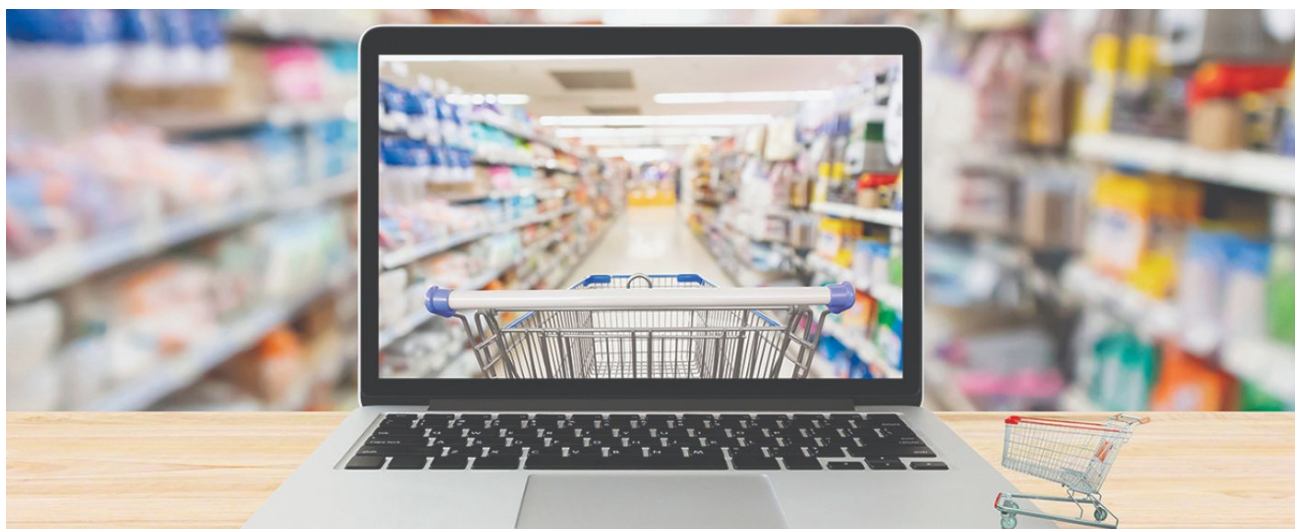


מיני פרויקט בסיסי נתונים

ניהול אתר קניות באינטרנט



מכון לב - 2020
מרצה מנחה: מר יאיר גולדשטיין

מגישים:

אלון שאולי - 322372632
מתן מועלם - 322889114
אופיר שמואלי - 212071567

מטרת הפרויקט

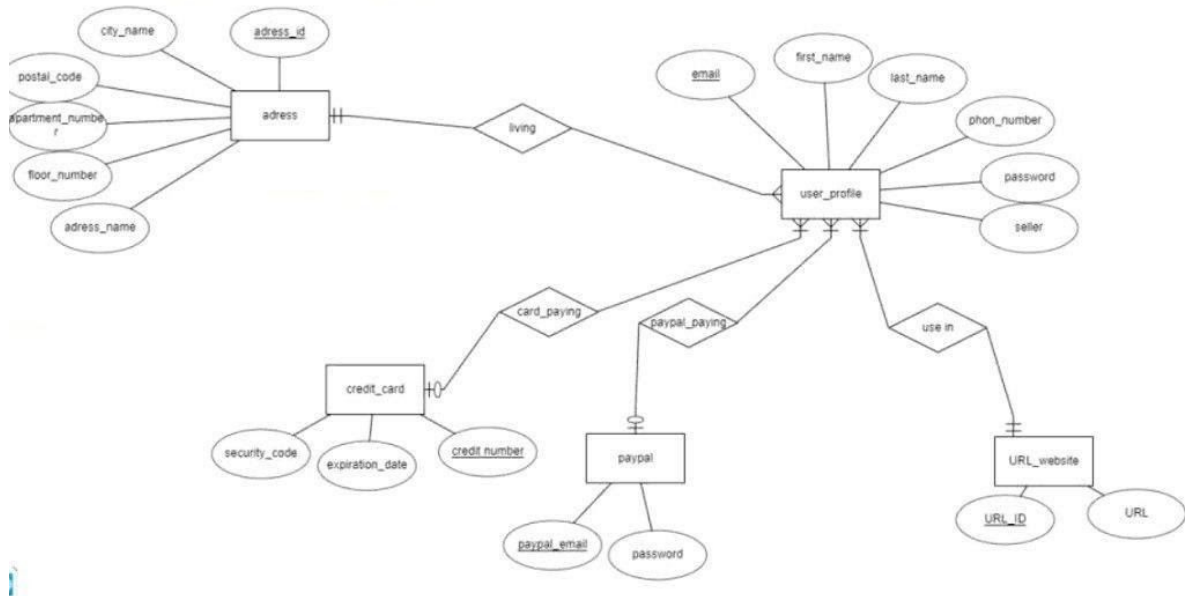
מערכת ניהול הזמנת קניות ללקוחות דרך אתר אינטרנט, שמטרתה העיקרית לנהל את הקניות ללקוח פוטנציאלי תוך כדי שמירת מידע מקיף על פרטי הלקוח.

ישויות

- **Adress**: ישות חזקה - המכילה את כל פרטי הכתובת.
address_id - מפתח מספרי ייחודי לזיהוי הכתובת.
 city_name - שם העיר.
 floor_number - מספר הקומה.
 apartment_number - מספר הדירה.
 postal_code - מיקוד הדירה לשליחת דואר.
 adress_name - שם הרחוב.
- credit_card - ישות חזקה - ישות זו אחראית על כל הכרטיסים.
credit_number - מפתח ראשי המייצג את מספר הכרטיס.
 expiration_date - תאריך תפוגת הכרטיס.
 security_code - קוד הבטחון בן 3 ספרות.
- paypal - ישות חזקה - ישות זו אחראית על כל החשבונות.
paypal_email - מפתח ראשי המייצג פרטי משתמש Paypal
 password - סיסמא המשמשת לאימות בעל החשבון.
- URL_website - ישות חזקה - ישות זו אחראית על כל האתרים.
URL_ID - מפתח ראשי המשמש לזיהוי של האתר.
 URL - כתובת האתר.

- user_profile - ישות חזקה - ישות זו אחראית על פרטי המשתמשים.
 - email - מפתח ראשי המייצג את כתובת האימייל של המשתמש.
 - first_name - שם פרטי.
 - last_name - שם משפחה.
 - phon_number - מספר טלפון.
 - password - סיסמא לזיהוי המשתמש.
 - seller - מאפיין המשתמש האם הינו מוכר או קונה.
 - adress_id - מספר לזיהוי כתובת המשתמש.
 - credit_number - מספר כרטיס האשראי של המשתמש.
 - paypal_email - כתובת האימייל לזיהוי חשבון הPaypal
 - URL_ID - מספר לזיהוי כתובת אתר המשתמש.

תרשים ERD



הנימוק שהסכמה עומדת בדרישות 3NF

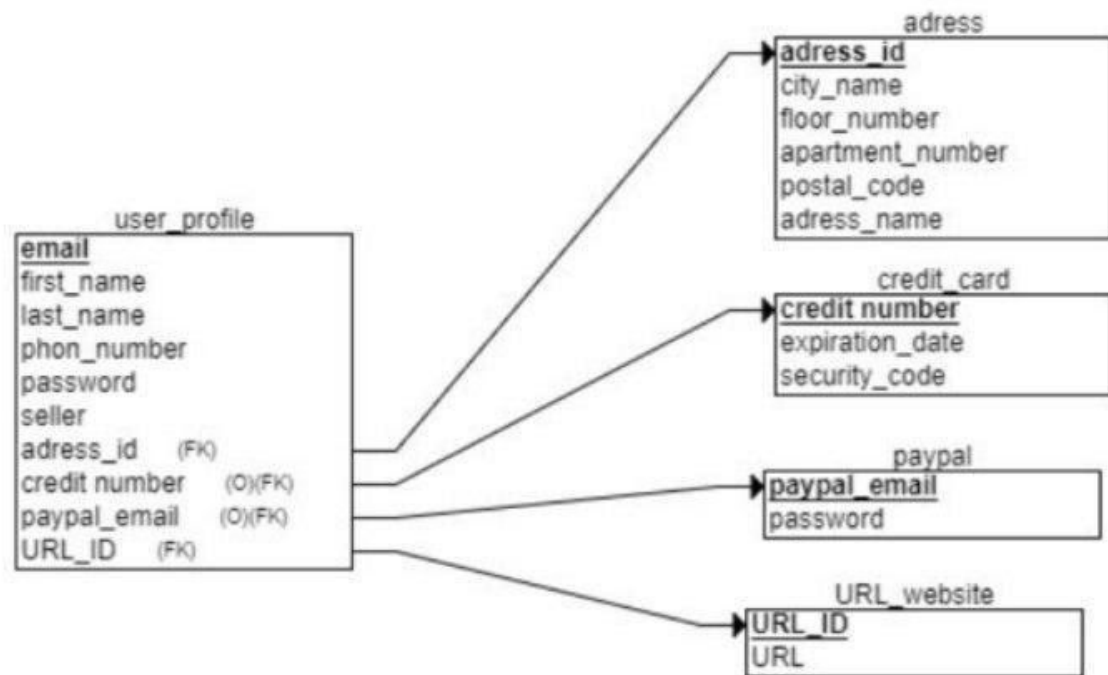
כל הטבלאות עומדות בהגדרת 3NF מכיוון שהתלויות היחידות של כל טבלה הן שהמפתחות תולים את שאר התכונות.

א"כ לכל טבלה עבור כל תלות $X \rightarrow Y$ (איקס תולה y) בהכרח X הוא מפתח בטבלה.

תיאור הקשרים

- כל משתמש, נמצא בטבלה user_profile פעם אחת בדיוק.
- לכל משתמש חייב להיות כתובת אחת בדיוק.
- לכל משתמש חייב להיות אתר אחד בלבד.
- לכל משתמש יכול להיות עד כרטיס אשראי 1.
- לכל משתמש יכול להיות עד חשבון אחד.

תרשים DSD



נרמול הטבלאות

Adress(address_id, city_name, floor_number, apartment_number, postal_code, adress_name)

Credit_card (credit_number, expiration_date, security_code)

Paypal (paypal_email, password)

user_profile (first_name, last_name, phon_number, password, email, seller, address_id, credit_number, paypal_email, URL_ID)

URL_website (URL_ID, URL)

יצירת הטבלאות

טבלת address:

```
CREATE TABLE address
(
    address_id INT NOT NULL,
    city_name VARCHAR(20) NOT NULL,
    floor_number NUMERIC(2) NOT NULL,
    apartment_number NUMERIC(4) NOT NULL,
    postal_code NUMERIC(10) NOT NULL,
    address_name VARCHAR(20) NOT NULL,
    PRIMARY KEY (address_id)
);
```

טבלת credit_card:

```
CREATE TABLE credit_card
(
    credit_number VARCHAR(20) NOT NULL,
    expiration_date DATE NOT NULL,
    security_code NUMERIC(3) NOT NULL,
    PRIMARY KEY (credit_number)
);
```

טבלת paypal:

```
CREATE TABLE paypal
(
    paypal_email VARCHAR(30) NOT NULL,
    password VARCHAR(6) NOT NULL,
    PRIMARY KEY (paypal_email)
);
```

טבלת URL_website:

```
CREATE TABLE URL_website
(
    URL_ID INT NOT NULL,
    URL VARCHAR(30) NOT NULL,
    PRIMARY KEY (URL_ID)
);
```

טבלת user_profile:

```
CREATE TABLE user_profile
(
    first_name VARCHAR(20) NOT NULL,
    last_name VARCHAR(20) NOT NULL,
    phon_number NUMERIC(10) NOT NULL,
    password VARCHAR(6) NOT NULL,
    email VARCHAR(30) NOT NULL,
    seller VARCHAR(3) NOT NULL,
    adress_id INT NOT NULL,
    credit_number VARCHAR(20),
    paypal_email VARCHAR(30),
    URL_ID INT NOT NULL,
    PRIMARY KEY (email),
    FOREIGN KEY (adress_id) REFERENCES adress(adress_id),
    FOREIGN KEY (credit_number) REFERENCES credit_card(credit_number),
    FOREIGN KEY (paypal_email) REFERENCES paypal(paypal_email),
    FOREIGN KEY (URL_ID) REFERENCES URL_website(URL_ID)
);
```

הכנסת הנתונים לטבלאות

- בטבלת user_profile יצרנו 20000 שורות בעזרת data generator בPLSQL.
- בטבלת adress יצרנו 17000 שורות בעזרת data generator בPLSQL.
- בטבלת credit_card יצרנו 10000 שורות בעזרת data generator בPLSQL.
- בטבלת paypal עשינו 981 (19 ירדו בגלל הגבלות של הטבלה) בmockaroo ועוד 9019 עשינו בעזרת data generator בPLSQL אז סה"כ 10000.
- בטבלת URL_websie יצרנו 30 אתרים בעזרת python script שכתבנו.

שאלות

שאלת 1:

נרצה לדעת את פרטי המוכרים ברשת הקניות אונליין כדי שנוכל לתקשר איתם. לצורך כך נכתוב שאלתא המאגדת את פרטי המשתמש של המוכרים ומחזירה לנו את השם המלא עם כתובת המייל ומספר הטלפון שלהם.

```
select last_name, first_name, phon_number, email
from user_profile
where seller='Yes'
order by last_name;
```

		LAST_NAME	FIRST_NAME	PHON_NUMBER	EMAIL
1		Aaron	Sammy	8941235170	sammya@trx.com
2		Aaron	Marie	1030454559	marie.aaron@operationaltechnol
3		Addy	Cherry	8885082469	cherry.addy@providentbancorp.c
4		Adler	Gabrielle	2901005551	gabrielle@younginnovations.com
5		Akins	Andy	6536648368	andy.akers@cima.com
6		Albright	Jennifer	1644466235	j.albright@meritagetechnologie
7		Alexander	Boyd	8194341500	boyd.alexander@afs.ca
8		Allen	Norm	9914810964	nallen@monitronicsinternationa
9		Anderson	Shannon	5519081943	shannon.anderson@activeservice
10		Arkenstone	Frank	7926596025	frank.a@qls.de
11		Armstrong	Nina	608445549	narmstrong@entelligence.com
12		Arthur	Nicky	6402207262	nicky.arthur@cartermortgage

1 of 238 SYSTEM [11:05:40] 238 rows selected in 0.163 seconds

שאלתה 2

נרצה לדעת כמה משתמשים יש בכל עיר אשר יש להם חשבון Paypal לצורך ניתוח התפלגות המשתמשים לפי ערים. כך שמנהלי האתר יוכלו לדעת את מצב השוק. לשם כך נכתוב שאלתא אשר מחזירה לנו את כל הערים ואת כמות המשתמשים באותה עיר כנדרש, ממוינים לפי מספר המשתמשים באותה עיר.

```
--How many PayPal users are there in each city?--
select distinct city_name, count paypal_email) as numberOfPaypalUsers
from user_profile natural join address
group by address.city_name
order by count paypal_email) desc;
```

		CITY_NAME		NUMBEROFPAYPALUSERS
▶	1	Los Angeles	...	6
	2	Akron	...	5
	3	Richmond	...	4
	4	Reno	...	4
	5	Utrecht	...	4
	6	Glendale	...	4
	7	Vista	...	4
	8	Koufu	...	4
	9	Fountain Hills	...	4
	10	Mason	...	4

6:1 & SYSTEM [11:13:50] 313 rows selected in 0.510 seconds

שאלתה 3

נרצה לדעת מי הם המשתמשים באתר אשר הם רשומים במערכת שהם רק קונים באתר, לצורך שליחת הודעות פרסום כדי לעניין אותם לקנות דברים. לשם כך נכתוב שאלתה אשר מחזירה לנו פרטים על כל הלקוחות שיש לנו באתר.

```
--Users who also have a credit card and PayPal--
select first_name,last_name,credit_number,paypal_email
from user_profile
where user_profile.seller='No';
```

		FIRST_NAME		LAST_NAME		CREDIT_NUMBER		PAYPAL_EMAIL	
▶	1	Ronnie	...	Neuwirth	...	8702 9350 8700 9349	...	nikki.henriksen@avr.no	...
	2	Brooke	...	Baldwin	...	7346 8672 7344 8671	...	nellye@columbiabancorp.dk	...
	3	Dick	...	English	...	8166 9082 8164 9081	...	scotth@priorityexpress.au	...
	4	Terry	...	Burton	...	8102 9050 8100 9049	...	latlas@intraspheretechnologies	...
	5	Nikka	...	Rhames	...	6530 8264 6528 8263	...	mos.church@ataservices.es	...
	6	Randy	...	Quatro	...	9526 9762 9524 9761	...	laurence@pepsico.es	...
	7	Carlene	...	Posey	...	9610 9804 9608 9803	...	gaby.warren@insurmark.com	...
	8	Frederic	...	Dreyfuss	...	6122 8060 6120 8059	...	mmorales@consultants.si	...
	9	Cornell	...	Bright	...	6170 8084 6168 8083	...	millie.ellis@voicelog.ch	...
	10	Ron	...	Remar	...	6202 8100 6200 8099	...	n.gallagher@mai.at	...
	11	Gene	...	Fishburne	...	9698 9848 9696 9847	...	lpepper@nissanmotor.uk	...
	12	Laranz	...	Bichter	...	7222 9660 7220 9659	...	carrie.pedman@madamius.com	...

4:32 SYSTEM [11:19:51] 262 rows selected in 0.236 seconds

שאלתה 4

מנהל האתר ביקש לדעת עבור כל אתר כמה לקוחות קונים ממנו, כך שלפי מספר המכירות נקבע את התשלום עבור פרסומת בכל אתר.
 לשם כך נכתוב שאלתה שמחזירה לפי סדר ממזין את שם האתרים ועבור כל אתר כמה לקוחות יש לו.

```
--For each site, gives the number of users--
select url_website.url, count(seller) as numberOfUsers
from user_profile natural join URL_WEBSITE
where seller = 'No'
group by url_website.url
order by count(seller) desc;
```

	URL	NUMBEROFUSERS
1	Brooklinen.com	17
2	Etsy.com	14
3	FreePeople.com	13
4	macys.com	13
5	homdepot.com	13
6	Wish.com	11
7	Zappos.com	10
8	Asos.com	10
9	Overstock.com	10
10	kohls.com	10

6:29 SYSTEM [11:34:30] 30 rows selected in 0.075 seconds

שאלתה 5.

לצורך ניתוח התפלגות המוכרים באתרים השונים לפי העיר שבה הם מוכרים.

לשם כך נכתוב שאלתה שעבור כל עיר מחזירה לנו את מספר המוכרים שיש בה בסדר ממזין.

```
--sum of sellers for each city--
select city_name,count(*) as sum_of_sellers
from address_natural join user_profile
where user_profile.seller='Yes'
group by city_name
order by count(*)desc;
```

		CITY_NAME		SUM_OF_SELLERS
▶	1	Akron	...	4
	2	Richmond	...	3
	3	Szazhalombatta	...	3
	4	Koufu	...	3
	5	Reno	...	3
	6	Los Angeles	...	3
	7	Rochester	...	3
	8	Santa rita sapucaí	...	3
	9	Augst	...	2
	10	Aberdeen	...	2
	11	Heubach	...	2
	12	Wuerzburg	...	2

6:22 SYSTEM [11:41:08] 188 rows selected in 0.171 seconds

שאלתה 6:

האתר שלנו הוקם לפני שנים רבות כך שחלק מהמשתמשים מחזיקים כרטיס אשראי שפג תוקפו. מנהל האתר ביקש למצוא את כל הלקוחות שאכן פג להם תוקף כרטיסי האשראי.

לצורך כך נכתוב שאלתה אשר בכל פעם שנריץ אותה היא תעבור על כל התאריכים של הכרטיסים ותמצא את המשתמש שעבר לו התוקף. השאלתה תחזיר לנו את שמו המלא והתאריך כדי שנוכל לעדכן את הלקוח.

```
--gives all the users that their credit card was expired--
```

```
ALTER SESSION SET TIME_ZONE = '+2:0';
```

```
select email, last_name, first_name, expiration_date
from user_profile natural join credit_card, dual
where CURRENT_DATE >= expiration_date
```

Alter session Select user_profile

	EMAIL	LAST_NAME	FIRST_NAME	EXPIRATION_DATE
1	bbaldwin@dillards.com	Baldwin	Brooke	06/07/2020
2	todd.k@ccfholding.mx	Keaton	Todd	16/08/2020
3	doug.palmieri@timberlanewoodcr	Palmieri	Doug	14/06/2020
4	julianna.clarkson@mitsubishimo	Clarkson	Julianna	27/07/2020
5	mchilton@uem.jp	Chilton	Mos	29/06/2020
6	mikeg@cyberthink.com	Gunton	Mike	14/06/2020
7	mika.ceasar@team.au	Ceasar	Mika	10/07/2020
8	terry.askew@aco.com	Askew	Terry	18/05/2020
9	talvin.melvin@jma.com	Melvin	Talvin	18/05/2020
10	rachel.stowe@data.au	Stowe	Rachel	25/07/2020
11	mwainwright@bestever.uk	Wainwright	Mary-Louise	01/08/2020
12	reesed@morganresearch.com	Duncan	Reese	10/08/2020

2:38 SYSTEM [12:25:27] 29 rows selected in 0.076 seconds

שאלתה 7:

ידוע שלמשתמשים קשה למצוא הרבה סיסמאות אשר אותם הם יזכרו, לכן הרבה אנשים מגדירים את אותה סיסמא להרבה אתרים. מה שגורם בעיות אבטחה כיוון שהפורצים יכולים לגלות את הסיסמא וככה לפרוץ את כל האתרים.

לשם כך כתבנו שאלתא שבודקת עבור כל המשתמשים באתר אם הסיסמא שהגדירו לחשבון הPaypal שלהם זה הסיסמא שהגדירו לכניסה לאתר. השאלתה מחזירה את הפרטים של המשתמש כדי שנוכל לעדכן אותו.

```
--get the user that have the same password on paypal--
select u.paypal_email,last_name, first_name, phon_number, u.password
from user_profile u, paypal p
where p.paypal_email= u.paypal_email
and u.password = p.password
```

	PAYPAL_EMAIL	LAST_NAME	FIRST_NAME	PHON_NUMBER	PASSWORD
1	fred.urban@computersource.br	Hedaya	Brent	2646095646	100003
2	tori.s@genghisgrill.de	Serbedzija	Chuck	6430163093	100018
3	p.heatherly@morganresearch.uk	Wilson	Clea	1149346313	100033
4	cherry.loggins@lynksystems.ca	Winger	Gene	443512721	100036
5	terry.weaving@streetglow.fr	Hirsch	Willie	5740076885	100051
6	nikka.neville@y2marketing.dk	Rooker	Alessandro	106809395	100057
7	jeanne.h@electricalsolutions.j	Hong	Kirk	9507539929	100066
8	mili.day@sps.com	Osbourne	Edward	2176227193	100072
9	rachael.keen@telecheminternati	Quinones	Dick	3574548461	100078
10	embeth.posey@learningvoyage.co	Connery	Angie	8267207989	100084
11	manu.swayze@ivci.es	Evanswood	Fred	5301729298	100090
12	melba.mantegna@ahl.com	Crow	Casey	3114142146	100096

5:28 SYSTEM [13:01:34] 220 rows selected in 0.161 seconds

שאלת 8:

לשם בטיחות האתר נרצה לבדוק שאכן הסיסמאות עומדות בתקן מסויים של רמת אבטחה.

נכתוב שאלתה שמחזירה לנו את הפרטים של כל המשתמשים אשר אורך הסיסמא שלהם קטן מ-6.

```
--get the user that have the same password on paypal--
select u.paypal_email,last_name, first_name, phon_number, p.password
from user_profile u, paypal p
where u.paypal_email =p.paypal_email and length(p.password) <6
```

	PAYPAL_EMAIL	LAST_NAME	FIRST_NAME	PHON_NUMBER	PASSWORD
1	2k.hiatt@younginnovations.ca	Teng	Ramsey	1356040341	1112
2	2sdupree@spd.com	Belushi	Kirk	4754218275	1139
3	2denise.numan@jcpenney.com	Ronstadt	Victor	9684005102	1130
4	2vienna.w@advertisingventures.	Neville	Crispin	3280629370	1082
5	2roscoe.tripplehorn@businesspl	Ticotin	Bonnie	4692445081	1100
6	2isuvvari@mainstreetbanks.com	Jovovich	Carlene	8995802581	1118
7	2phil@networkdisplay.hu	Farris	Brothers	4676131521	1064
8	2denise.numan@jcpenney.com	Dalton	Allan	2865555438	1130
9	2a.boone@ccb.za	Popper	Rich	3210886551	1103
10	2kay.mcdiarmid@irissoftware.ca	Spiner	Cesar	8403050577	1046
11	2hcazale@fordmotor.de	Rispoli	Patty	3293203903	1010
12	2k.hiatt@younginnovations.ca	Venora	Joaquin	424730343	1112

3:30 SYSTEM [13:23:37] 50 rows selected in 0.059 seconds

שאלת 9:

לצורך שיפור שירות האתר בעל האתר ביקש שניצור פיצ'ר שישימש את משתמשי האתר, כך שהמשתמש יכניס קלט שם של אתר וכתוצאה הוא יקבל כמה כניסות היו לאתר מכל עיר.

לשם כך יצרנו את השאלת המקבלת כקלט שם של אתר ובתגובה מחזירה את כמות המשתמשים בכל עיר בסדר ממזרח.

```
select city_name, count(*) as sum_of_users
from url_website natural join address natural join user_profile
where url_id = &< name= "url" list = "select url_id, url from url_website" description= "yes">
group by city_name
order by city_name;
```

	CITY_NAME	SUM_OF_USERS
1	Amherst	1
2	Bad Oeynhausen	1
3	Banbury	1
4	Belo Horizonte	1
5	Bkk	1
6	Boise	1
7	Caguas	1
8	Calgary	1
9	Carlin	1
10	Cedar Rapids	1
11	Cheshire	1

4:19 SYSTEM [13:41:01] 54 rows selected in 0.047 seconds

Variables

Name	Value
url	Etsy.com

OK

Cancel

Clear

שאלתה 10:


מנהל האתר שוקל להוסיף שירות שליחויות ללקוחות. מנהל האתר ביקש שנבנה לו שאלתא אשר תקבל כקלט מספר ותחזיר לנו את הפרטים של כל המשתמשים שגרים מעל קומה זו.


לשם כך כתבנו שאלתא המקבלת כקלט מספר קומה ומחזירה לנו את שמות המשתמשים ומספר הטלפון שלהם ממוינים לפי מספר הקומה בסדר עולה.

```
select first_name, last_name, phon_number, floor_number
from user_profile natural join adress
where floor_number > &<name= "floor number" type = "integer">
group by floor_number, first_name, last_name, phon_number
order by floor_number desc;
```

		FIRST_NAME		LAST_NAME		PHON_NUMBER		FLOOR_NUMBER
▶	1	Arturo	...	Tate	...	7724593878		99
	2	Carlene	...	Shocked	...	6400675493		99
	3	Dean	...	Cornell	...	7158966515		99
	4	Demi	...	Berkoff	...	2227569320		99
	5	Denzel	...	Sledge	...	5684307758		99
	6	Edgar	...	Hubbard	...	4259617635		99
	7	Emma	...	Cleary	...	8661900166		99
	8	Hank	...	Tucker	...	2947132973		99
	9	Jason	...	Hackman	...	4024743421		99
	10	Judge	...	Stuart	...	4417865196		99
	11	Lindsay	...	Shaw	...	9137531340		99
	12	Miranda	...	Reno	...	6522784280		99

5:28 SYSTEM [13:50:53] 1326 rows selected in 0.581 seconds

 Variables ✕

Name	Value
 floor number	3

OK

Cancel

Clear

אינדקסים

אינדקס 1:

אינדקס על עמודת סוג המשתמש האם הינו או מוכר או לקוח (SELLER) בטבלה של המשתמשים, שם האינדקס: IDX_SELLER.

הגדרת האינדקס:

```
create index IDX_SELLER
on USER_PROFILE (SELLER);
```

נראה את השיפור שמבצע האינדקס על שאלתא מספר 1.

לפני הגדרת האינדקס:

```
--Name and adress of each seller--
select last_name, first_name, phon_number, email
from user_profile
where seller='Yes'
order by last_name;
```

		LAST_NAME	FIRST_NAME	PHON_NUMBER	EMAIL
▶	1	Aaron	Mia	8290153611	1mia.aaron@paintedword.com
	2	Aaron	Marie	1030454559	marie.aaron@operationaltechnol
	3	Aaron	Sammy	8941235170	sammya@trx.com
	4	Abraham	Paul	346109444	1p.abraham@americanpan.com
	5	Addy	Cherry	8885082469	cherry.addy@providentbancorp.c
	6	Adkins	Hugo	1420290544	1hugo.adkins@pinnaclestaffing.
	7	Adler	Gabrielle	2901005551	gabrielle@younginnovations.com
	8	Aiken	Cheryl	6240524952	1c.aiken@sweetproductions.il
	9	Akins	Andy	6536648368	andy.akers@cima.com
	10	Akins	Isabella	2334188054	1isabella@accurateautobody.ch

6:1 SYSTEM [17:28:00] 657 rows selected in 0.630 seconds

אחרי הגדרת האינדקס:

```
--Name and adress of each seller--
select last_name, first_name, phon_number, email
from user_profile
where seller='Yes'
order by last_name;
```

	LAST_NAME	FIRST_NAME	PHON_NUMBER	EMAIL
1	Aaron	Mia	8290153611	1mia.aaron@paintedword.com
2	Aaron	Marie	1030454559	marie.aaron@operationaltechnol
3	Aaron	Sammy	8941235170	sammya@trx.com
4	Abraham	Paul	346109444	1p.abraham@americanpan.com
5	Addy	Cherry	8885082469	cherry.addy@providentbancorp.c
6	Adkins	Hugo	1420290544	1hugo.adkins@pinnaclestaffing.
7	Adler	Gabrielle	2901005551	gabrielle@younginnovations.com
8	Aiken	Cheryl	6240524952	1c.aiken@sweetproductions.il
9	Akins	Andy	6536648368	andy.akers@cima.com
10	Akins	Isabella	2334188054	1isabella@accurateautobody.ch

6:1 SYSTEM [17:25:54] 657 rows selected in 0.443 seconds

כפי שניתן לראות הגדרת האינדקס עזרה להפחית את זמן הריצה.

אינדקס 2:

אינדקס על עמודת חשבון משתמש Paypal בטבלת המשתמשים,
שם האינדקס: IDX_PAYPAL.

הגדרת האינדקס:

```
create index IDX_PAYPAL
on USER_PROFILE (PAYPAL_EMAIL);
```

נראה את השיפור שמבצע האינדקס על שאילתא מספר 8.

לפני הגדרת האינדקס:

```
--get the user that have the same password on paypal--
select u.paypal_email,last_name, first_name, phon_number, p.password
from user_profile u, paypal p
where u.paypal_email =p.paypal_email and length(p.password) <6
```

	PAYPAL_EMAIL	LAST_NAME	FIRST_NAME	PHON_NUMBER	PASSWORD
1	2k.hiatt@younginnovations.ca	Teng	Ramsey	1356040341	1112
2	2sdupree@spd.com	Belushi	Kirk	4754218275	1139
3	2denise.numan@jcpenny.com	Ronstadt	Victor	9684005102	1130
4	2vienna.w@advertisingventures.	Neville	Crispin	3280629370	1082
5	2roscoe.tripplehorn@businesspl	Ticotin	Bonnie	4692445081	1100
6	2isuvvari@mainstreetbanks.com	Jovovich	Carlene	8995802581	1118
7	2phil@networkdisplay.hu	Farris	Brothers	4676131521	1064
8	2denise.numan@jcpenny.com	Dalton	Allan	2865555438	1130
9	2a.boone@ccb.za	Popper	Rich	3210886551	1103
10	2kay.mcdiarmid@irissoftware.ca	Spiner	Cesar	8403050577	1046
11	2hcazale@fordmotor.de	Risnoli	Pattv	3293203903	1010

3:30 SYSTEM [21:07:44] 50 rows selected in 0.087 seconds

אחרי הגדרת האינדקס:

```
--get the user that have the same password on paypal--
select u.paypal_email,last_name, first_name, phon_number, p.password
from user_profile u, paypal p
where u.paypal_email =p.paypal_email and length(p.password) <6
```

	PAYPAL_EMAIL	LAST_NAME	FIRST_NAME	PHON_NUMBER	PASSWORD
1	2k.hiatt@younginnovations.ca	Teng	Ramsey	1356040341	1112
2	2sdupree@spd.com	Belushi	Kirk	4754218275	1139
3	2denise.numan@jcpenny.com	Ronstadt	Victor	9684005102	1130
4	2vienna.w@advertisingventures.	Neville	Crispin	3280629370	1082
5	2roscoe.tripplehorn@businesspl	Ticotin	Bonnie	4692445081	1100
6	2isuvari@mainstreetbanks.com	Jovovich	Carlene	8995802581	1118
7	2phil@networkdisplay.hu	Farris	Brothers	4676131521	1064
8	2denise.numan@jcpenny.com	Dalton	Allan	2865555438	1130
9	2a.boone@ccb.za	Popper	Rich	3210886551	1103
10	2kay.mcdiarmid@irissoftware.ca	Spiner	Cesar	8403050577	1046
11	2hcazale@fordmotor.de	Risnoli	Pattv	3293203903	1010

3:30 SYSTEM [21:12:56] 50 rows selected in 0.045 seconds

כפי שניתן לראות הגדרת אינדקס עזרה להפחית את זמן הריצה, מפני שהשאלתא עושה בדיקה לפי העמודה שעליה הגדרנו את האינדקס.

אינדקס 3:

אינדקס על עמודת ת.ז. אתר האינטרנט (URL_ID) בטבלת המשתמשים, שם האינדקס: IDX_URLID.

הגדרת האינדקס:

```
create index IDX_URLID
on USER_PROFILE (URL_ID);
```

נראה את השיפור שמבצע האינדקס על שאילתא מספר 9.

לפני הגדרת האינדקס:

```
--gives the sum of users per city for selected website--|
select city_name, count(*) as sum_of_users
from url_website natural join address natural join user_profile
where url_id = &< name= "url" list = "select url_id, url from url_website" description= "yes">
group by city_name
order by city_name;
```

	CITY_NAME	SUM_OF_USERS
1	Akron	1
2	Angers	1
3	Augst	1
4	Barueri	1
5	Belp	1
6	Berlin	1
7	Breda	1
8	Carlsbad	1
9	Chaam	1

1:57 SYSTEM [21:22:14] 35 rows selected in 0.093 seconds

אחרי הגדרת האינדקס:

```
--gives the sum of users per city for selected website--
select city_name, count(*) as sum_of_users
from url_website natural join address natural join user_profile
where url_id = &< name= "url" list = "select url_id, url from url_website" description= "yes">
group by city_name
order by city_name;
```

	CITY_NAME	SUM_OF_USERS
1	Akron	1
2	Angers	1
3	Augst	1
4	Barueri	1
5	Belp	1
6	Berlin	1
7	Breda	1
8	Carlsbad	1
9	Chaam	1

6:8

SYSTEM [21:28:11] 35 rows selected in 0.034 seconds

כפי שניתן לראות הגדרת אינדקס עזרה להפחית את זמן הריצה, מפני שהשאלתא עושה בדיקה לפי העמודה שעליה הגדרנו את האינדקס.

טבלת התפלגות

אינדקס	שאלתא	לפני הגדרת האינדקס	אחרי הגדרת האינדקס	חיסכון בשניות	חיסכון באחוזים
1	1	0.63	0.43	0.2	32%
2	8	0.087	0.045	0.042	48%
3	9	0.093	0.034	0.059	64%

אינטגרציה

הוחלט להרחיב את המערכת ולהוסיף ישות מוצר כך שתהיה אפשרות לקונה לראות את מחיר המוצר ופרטים נוספים. בנוסף יהיה אפשרות לקונים לתת ביקורת על המוצר. בנוסף הוחלט להרחיב את המערכת כך שעבור כל מוצר שנקנה יהיה גם אחריות.

תיאור הקשרים

- לכל מוצר יש מזהה והסבר קצר.
- לכל מוצר תינתן אפשרות לתת עליו ביקורות עם אופציית דירוג ואפשרות תגובה מילולית.
- לכל מוצר יש אחריות תוך כדי שמירת תאריך הקניה ופרטי המוכר והלוקח.

פירוט תכונות הסכמות שנוספו (קו תחתון - מפתח)

- **Product:** ישות חזקה - ישות זו אחראית על כל מוצר.
 - pid - מפתח מספרי ייחודי לזיהוי המוצר.
 - brief_description - תיאור המוצר בקצרה.
 - nom - שם המוצר.
 - additional_description - פרטי מוצר נוספים.
- **Customer_Review:** ישות חלשה התלויה בישות מוצר - אחראית על חוות דעת המשתמשים.
 - cid - מפתח ראשי המשמש לזיהוי הביקורת.
 - pid - מפתח ראשי זר המשמש לזיהוי המוצר
 - comments - תוכן התגובה.
- **Guarantee:** ישות חלשה - מייצגת את האחריות על המוצר.
 - gid - מספר מזהה של האחריות.
 - pid - מפתח ראשי זר למוצר.
 - email_seller - כתובת אימייל של המוכר.
 - email_buyer - כתובת אימייל של הלקוח.
 - _date - תאריך הקניה.
 - _level - דרגת האחריות.

שאלות

שאלת 1:

לצורך הנפקת כרטיס vip לכל לקוח שקנה מעל 3 מוצרים לשם כך נכתוב שאלת המחזירה את שם הלקוח כמות המוצרים ופרטים ליצירת קשר

```
select last_name,first_name,phon_number, email_buyer,count(*)
from user_profile u , guarantee g
where u.email = g.email_buyer
group by last_name,first_name,phon_number, email_buyer
having count(*)>=3
order by last_name;
```

	LAST_NAME	FIRST_NAME	PHON_NUMBER	EMAIL_BUYER	COUNT(*)
1	Heston	Jon	8534165017	jon.heston@allegiantbancorp.de	4
2	Ledger	Rebeka	115426336	rebeka.ledger@inzone.au	3

שאלתא 2:

לצורך בקרה נרצה לדעת איזה מוצרים הכי חמים באתר. נעשה שאלתה הנותנת למשתמש לבחור פריט מתוך כל סך הפריטים והשאלתא תחזיר לנו את כל המוכרים שמכרו את המוצר וכמות הפעמים שהוא נמכר.

```
from guarantee natural join product, user_profile u
where email = email_seller
and pid = &<name="product" list="select pid,nom from product" description="yes">
group by u.last_name,u.first_name, nom
order by u.last_name
```

	LAST_NAME	FIRST_NAME	NOM	COUNT(*)
1	Barry	Minnie	samarium	1
2	Gagnon	Kristin	samarium	1
3	Ranger	Campbell	samarium	1

שאלת 3

נרצה לקבל מידע על כל המוצרים שקיבלו ממוצע דירוג לפי בחירת המשתמש. לצורך כך ניצור את השאלתה הבאה אשר מקבלת כקלט מספר שלם מהמשתמש ומחזירה את הפרטים על כל המוצרים אשר הדירוג שלהם מעל המספר אשר קיבלנו כקלט.

```
select pid, nom, avg(rank_) as avarage, min(rank_) as minimum, max(rank_) as maximum
from product natural join customer_review
group by pid, nom
having avg(rank_)>= &<name= "rank" type="integer" hint="type rank from 1 to 10">
order by nom;
```

	PID	NOM	AVARAGE	MINIMUM	MAXIMUM
1	72915	fermium	5	5	5
2	72947	fluorine	6	6	6
3	72996	fluorine	7	7	7
4	72975	lithium	5.5	3	8
5	72933	manganese	7	7	7
6	72998	nickel	8	8	8
7	72929	radium	10	10	10
8	72917	rutherford	10	10	10
9	72902	samarium	7	7	7

7:1 & SYSTEM [22:02:33] 12 rows selected in 0.035 seconds

תרשימים

דוח 1:

לצורך בקרה על המוכרים באתר נרצה לראות את התפלגות המחירים לפי כמות המוצרים. לשם כך השאילתה ממיינת את המוצרים לפי המחירים ובודקת כמה יש מכל מוצר.

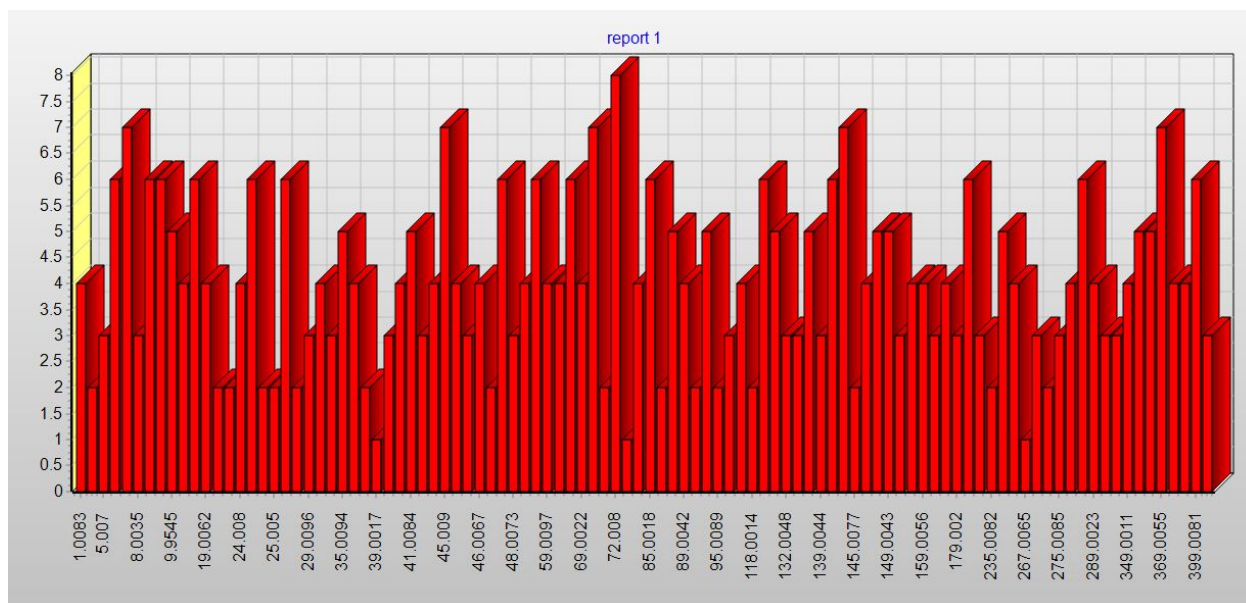
תוכן השאילתא:

```
select price, count(*) as sum
from product
group by price
order by price
```

Price	Sum
1.0083	4
3.0087	2
5.007	3
5.0071	6
8.0025	7
8.0035	3
8.0057	6
9.009	6
9.9545	5
14.006	4
18.0021	6

ייצוג גרפי:

על מנת שהדוח הנ"ל יהיה יותר מוחשי, נייצג אותו באמצעות גרף. כאשר ציר X בגרף מייצג לנו את מחיר המוצר, ואילו ציר Y מייצג לנו את הכמות המוצרים במחיר הזה. לדוגמא בגרף הנתון ניתן לראות שקיימים 3 מוצרים עם מחיר 5007.



דוח 2:

לצורך בקרה על מנגנון מתן האחריות עבור המוכרים באתר נרצה לראות את התפלגות המוכרים נותני האחריות לפי כמות המכירות. לשם כך השאילתה ממיינת את המוצרים לפי כתובות האימייל של המוכרים ובודקת את כמות הנפקת האחריות עבור כל מוכר.

תוכן השאילתא:

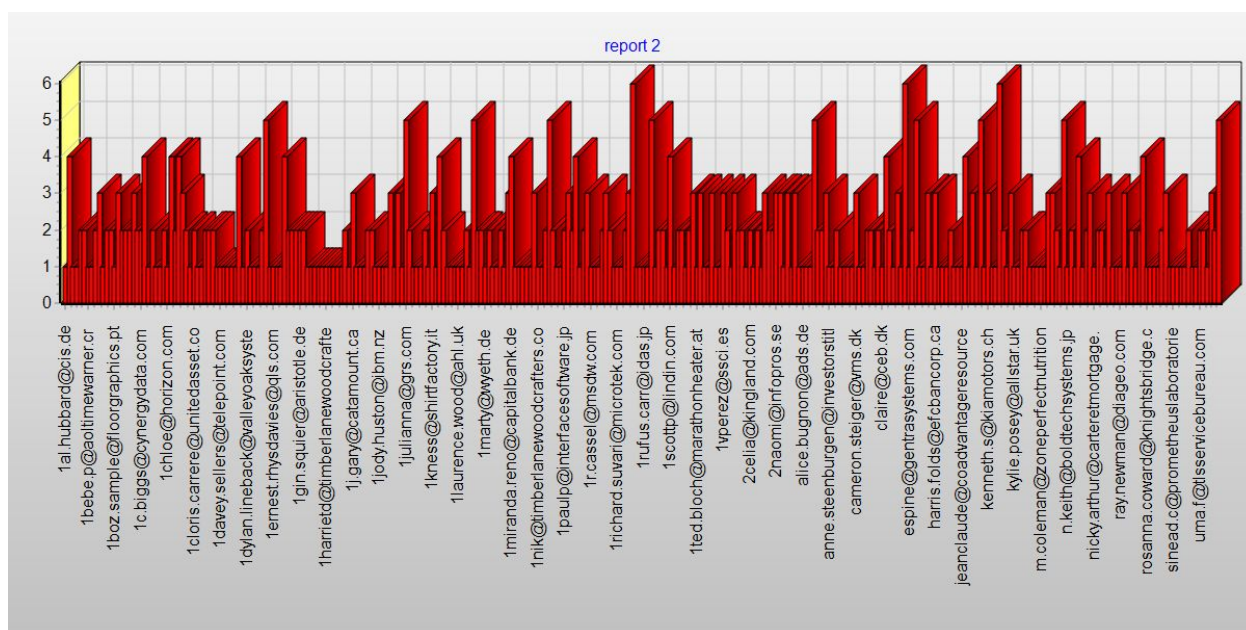
```
select email, count(*) as sum
from user_profile, guarantee g
where email = g.email_seller
group by email
order by email
```

Email	Sum
1al.hubbard@cis.de	1
1alice.h@blueoceansoftware.com	4
1alicia.p@onesourceprinting.tr	1
1amanda.gooding@summitenergy.d	1
1andrae.m@callhenry.com	2
1armin.thomson@scriptsave.ec	2
1bebe.p@aoltimewarner.cr	1
1bernard.gooding@гна.com	1
1bette.harrison@bradleypharmac	2
1bhedaya@anheuserbusch.com	3
1bo26@mastercardinternational.	1

1:1 SYSTEM [21:46:52] 306 rows selected in 0.285 seconds

ייצוג גרפי:

על מנת שהדוח הנ"ל יהיה יותר מוחשי, נייצג אותו באמצעות גרף. כאשר ציר X בגרף מייצג לנו את כתובות המוכרים, ואילו ציר Y מייצג לנו את כמות המוצרים שמכר. לדוגמא בגרף הנתון ניתן לראות שקיימים 3 מוכרים שהנפיקו 6 תעודות אחריות עבור מוצריהם.



Views

:view 1

נגדיר מבט לצורך קבלת מידע על כל סוכני הנסיעות והמכירות שנמכרו על ידם, לצורך בקרה על מכירת מלאי הכרטיסים.

יצירת המבט:

```
create view view1 as
select u.last_name, u.first_name, u.email, pid, price, date_
  from product natural
  join Guarantee p, user_profile u
 where u.email = p.email_seller;
```

view 2

נרצה ליצור מבט לצורך קבלת מידע על כל המוכרים, כדי לדעת את כל פרטי המוכר לצורך שליחת דואר רגיל (על ידי כתובת המוכר ומקום מגוריו) כתובת האתר ודואר אלקטרוני.

יצירת המבט:

```
create view view2 as
select url,last_name,first_name,phon_number,email,paypal_email,
       city_name,apartment_number,adress_name
  from user_profile natural
 join adress natural
 join URL_website
 where seller = 'Yes';
```

view 3

נרצה ליצור מבט לצורך קבלת מידע על כל הביקורות שקיבלו המוצרים, עבור מחלקת שביעות רצון הלקוחות.

יצירת המבט:

```
create view view3 as
select pid, nom, brief_description, rank_, comments
  from product natural
  join customer_review;
```

view 4

נרצה ליצור מבט לצורך קבלת מידע על כל משתמש שנמצא במערכת, עבור שירות הלקוחות שיוכלו לדלות מידע על החשבונות השונים שיש למשתמשים.

יצירת המבט:

```
create view view4 as
select url,u.last_name, u.first_name, u.phon_number, u.email,
c.credit_number,
       c.expiration_date, c.security_code, p.paypal_email
from user_profile u natural
join URL_website
left outer join paypal p
  on u.paypal_email = p.paypal_email
left outer join credit_card c
  on u.credit_number = c.credit_number
```

פונקציות

פונקציה 1 - product_price_changed_by_parameter:
 עקב משבר הקורונה והמשבר הכלכלי הבינלאומי, נוצר משבר בענף הקניות. לשם כך הוחלט לאפשר שינויים במחירי המוצרים.
 הפונקציה product_price_changed_by_parameter מקבלת שני פרמטרים: הראשון - מספר מוצר שעליו נדרש לבצע עדכון מחיר, השני - האחוזים שמעוניינים להפחית/להוסיף למחיר הקיים. בנוסף, הפונקציה מבצעת בדיקה שאכן המחיר החדש לא שלילי או שווה ל-0, במידה וכן העדכון לא יתבצע.

תוכן הפונקציה:

```
create or replace function
product_price_changed_by_parameter(productid in int,
percentage_change in float)
return integer is
  cnt integer default 0;

  cursor product_cursor is
  select pid,price
  from product
  where pid = productid
  for update;
  iterator product % rowtype;
begin
  for iterator in product_cursor
  loop
    if iterator.price * percentage_change >1
    then
      update product
      set price = iterator.price * percentage_change
      where current of product_cursor;
      cnt:=cnt+1;
    end if;
  end loop;
  return(cnt);
end product_price_changed_by_parameter;
```

פונקציה 2 - calculating_seller_bonus:

בעקבות ירידה במכירות ברבעון האחרון של השנה, הוחלט בסוכנות הנסיעות לתת תמריץ למוכרי מוצרים רבים. לשם כך הפונקציה calculating_seller_bonus מקבלת כפרמטר את האימייל של המוכר, סוכמת את מספר המוצרים שהוא מכר, ובהתאם לכך מחזירה את סכום הבונוס שמגיע לו (אפשר להגדיר את הבונוס לפי בקשת המשתמש). הפונקציה בודקת שאכן הפרופיל שקיבלנו הוא מוגדר כמוכר במערכת, הפונקציה מחזירה את הבונוס שמגיע לסוכן.

```
create or replace function calculating_seller_bonus(email_recieved in
string)
return float is
    bonus float;
    amount int;
    production_bonus float default 1.15;
    is_seller string(20);
begin
    select seller into is_seller
    from user_profile
    where email_recieved = email;
    if is_seller = 'Yes'
        then
            select count(*) into amount
            from guarantee g
            where email_recieved = g.email_seller;
            bonus := amount*production_bonus;
            dbms_output.put_line('the user ' + email_recieved + 'sales ' +
amount);
        end if;
    return (bonus);
end calculating_seller_bonus;
```


פרוצדורות

פרוצדורה 1 - sellers_bonus_list :

כל רבעון נרצה לחשב את הבונוסים המגיעים לכל המוכרים באתר. לשם כך ניצור את הפרוצדורה sellers_bonus_list שעוברת על כל המוכרים הקיימים במאגר ומפעילה את הפונקציה לעיל המחשבת את הבונוס לפי מכירות, עבור כל מוכר. הפרוצדורה מדפיסה את שם המוכר, ואת הפרטים שבמערכת ואת הבונוס שמגיע לו. הפרוצדורה לא מקבלת פרמטרים (עוברת על כל הסוכנים פעם ברבעון - כאשר נפעיל אותה), מכיוון שמדובר בפרוצדורה היא עושה הדפסה למסך ולא מחזירה ערך.

תוכן הפרוצדורה:

```
create or replace procedure sellers_bonus_list is
bonus float default 0;
cursor users_cursor is
  select last_name, first_name, phon_number, email
  from user_profile
  where seller = 'Yes';
  iterator user_profile % rowtype;
begin
  for iterator in users_cursor
  loop
    bonus:= calculating_seller_bonus(iterator.email);
    dbms_output.put_line('the user '|| iterator.email || ' rewarded
' || bonus);
  end loop;
end sellers_bonus_list;
```

פרוצדורה 2 - sellers_bonus_list :

נוהל שינוי סיסמא למשתמש באתר הקניות מחייב את המשתמש להכניס את מזהה המשתמש ואת הסיסמא הישנה וכן את הסיסמא החדשה. הפרוצדורה sellers_bonus_list מקבלת 3 פרמטרים: הראשון - האימייל המזהה של המשתמש באתר, השני - ערך הסיסמא הישנה, והשלישי - ערך הסיסמא החדשה אותה נרצה לעדכן. הפרוצדורה מוצאת את המשתמש, ובודקת האם הסיסמא הישנה תואמת לערך השמור אצלנו במערכת. במידה ואכן הסיסמא שווה לסיסמא הקודמת יודפס למשתמש הודעה שאכן השינוי בוצע בהצלחה. אם לא בוצע שינוי, המשתמש מקבל הודעת שגיאה מדוע לא בוצע השינוי.

תוכן הפרוצדורה:

```
create or replace procedure user_password_change(email_recieved in
string, old_password in string, new_password in string) is
exist int default 0;
current_password string(30);
begin
  select password, count(*) into current_password, exist
  from user_profile
  where email = email_recieved
  group by password;
  if exist > 0
  then
    if current_password=old_password
    then
      update user_profile
      set password = new_password
      where email = email_recieved;
      dbms_output.put_line('the current password you sent changed
successfully');
    else
      dbms_output.put_line('the current password you sent does
not matched');
    end if;
  else
    dbms_output.put_line('the '||email_recieved||'does not exist');
  end if;
end user_password_change;
```