

מינি פרויקט בסיסי נתוניים

תשיף



מגישים:

אבי קניגסברג 311223705

תמר גולדשטייד 207623224

שלב 1

1.1 מיאור המערכת

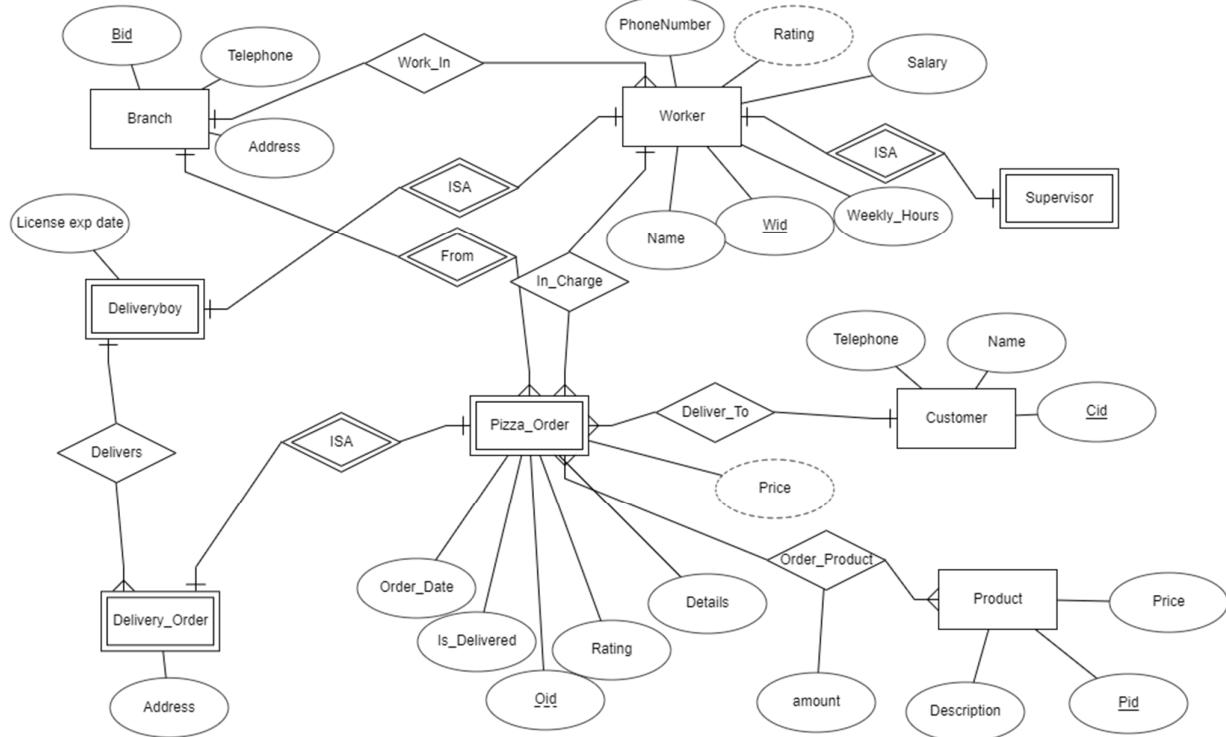
המערכת תשמש לניהול הזמינות ומשלוחים של רשת "פיצה קורונה" [Pizza Corona].

רשת "פיצה קורונה" נוסדה בסין והתפתחה מאוד באיטליה ולבסוף אף כבשה אזורים נרחבים בעולם כולל [ואף על ארצנו הקטנטונית לא פסחה], כאמור הרשת מכילה סניפים רבים, בכל סניף יש מספר עובדים, ביניהם גם מפקח ומשלוחנים.

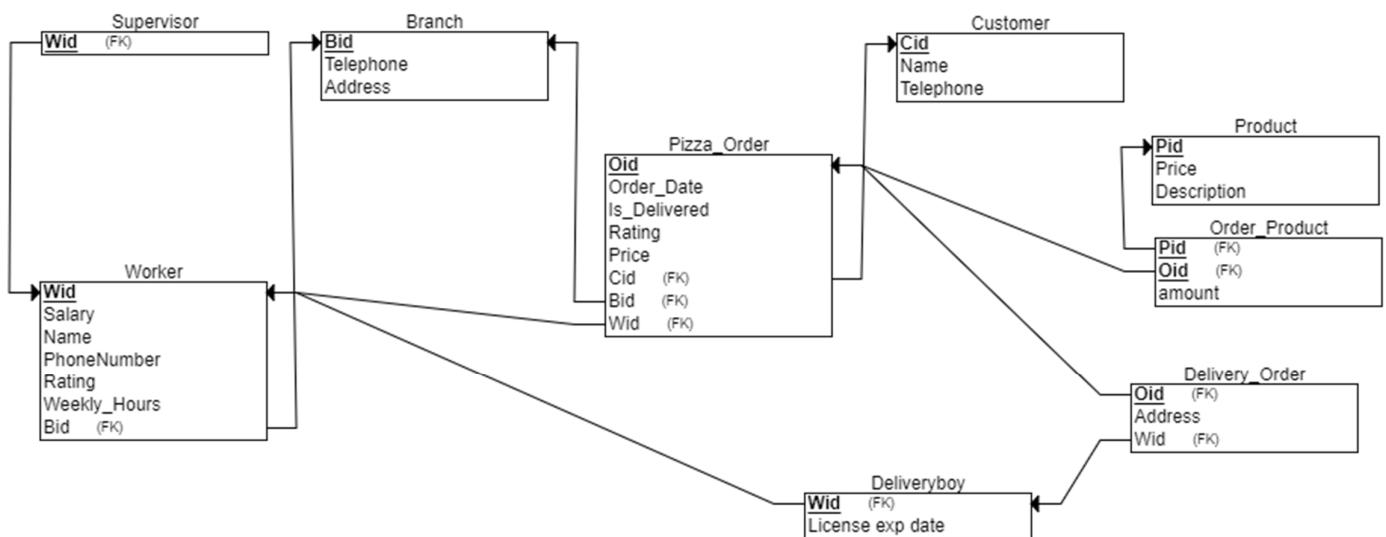
המערכת מבצעת את הדברים הבאים:

- מאפשרת לבצע הזמנות מתחਪרט הפייצריה, עם אפשרות לשלוח לבית הלוקה.
- שומרת את שם וכותבת הלוקה, למקרה שריצה להזמין שוב באותו.
- מאפשרת ללקוח מתחת לדירוג להזמנה (לעובד שטיפל בה).
- מחשבת את מחיר ההזמנה כולל על פי מחירי המוצריים שהיא כוללת.
- מחשבת את הדירוג של כל עובד על פי דירוג ההזנות שביצע.
- שומרת עבור כל משולחן את ההזנות שלו.
- מעדכנת עבור כל הזמינה אם היא נמסרה.

ERD הוראות 1.2



DSD הוראות 1.3



1.3.1 תיאור היחסיות

(סניף) - מתאפיין ב: ת.ז (מפתח ראשי), כתובת, טלפון.
 Branch (עובד) - מתאפיין ב: ת.ז (מפתח ראשי), כתובת, טלפון, שם, משכורת, דירוג, ת.ז סניף (מפתח זר).
 Worker (אחראי משמרת) - יורש את כל התוכנות של עובד באמצעות מפתח זר: ת.ז.
 Supervisor (שלוחן) - יורש את כל התוכנות של עובד באמצעות מפתח זר: ת.ז, תוכנות נוספות: תאrik תפוגה Deliveryboy (שלוחן) - רישון הגיהה.
 של רישון הגיהה.
 (לקוח) - מתאפיין ב: ת.ז (מפתח ראשי), שם, טלפון.
 Customer (מוצר) - מתאפיין ב: ת.ז (מפתח ראשי), מחיר, תיאור.
 Product Pizza_Order (הזמנה) - מתאפיין ב: ת.ז (מפתח ראשי), פרטי הזמנה, דירוג, תאrik, שעה, מחיר, האם_ນמטרה, ת.ז עובד אחראי הזמנה (מפתח זר).
 Delivery_Order (הזמנה פלוס שלוחן) - יורש את כל התוכנות של הזמנה פיצה באמצעות מפתח זר: ת.ז, תוכנות נוספות: כתובת, ת.ז שלוחן (מפתח זר).

1.3.2 תיאור הקשרים

- כל עובד מקשור לסניף אחד. (יתכן הרבה עובדים לסניף)
- כל אחראי משמרת מקשור לעובד אחד (ירושה). עובד מקשור לכל היוטר לאחראי משמרת אחד.
- כל שלוחן מקשור לעובד אחד (ירושה). עובד מקשור לכל היוטר ושלוחן אחד.
- כל מוצר יכול להיות מקשור להרבה הזמנות (יתכן הרבה מוצרים בהזמנה). כל הזמנה יכולה להיות מקושרת להרבה מוצרים, אך בהרבה מקושרת לפחות למוצר אחד. לשר יש מאפיין כמוות.
- כל הזמנה מקושרת ללקוח אחד (יתכן הרבה הזמנות ללקוח).
- כל הזמנה משולח מקושרת להזמנה אחת (ירושה). הזמנה מקושרת לכל היוטר להזמנה משולח אחת.
- כל הזמנה מקושרת לסניף אחד (יתכן הרבה הזמנות לסניף).
- כל הזמנה מקושרת לעובד אחד (יתכן הרבה הזמנה לעובד).

הערה: בגלל שינוי אפשרות שעבוד יעבור סניף, השארנו את הקישור של הזמנה לסניף וגם לעובד.

1.4 פקודות SQL לשיזיון וקשרים

1.4.1 יצרת טבלאות

```
CREATE TABLE Customer
(
    Name VARCHAR(20) NOT NULL,
    Cid INT NOT NULL,
    Telephone VARCHAR(10) NOT NULL,
    PRIMARY KEY (Cid)
);
CREATE TABLE Product
(
    Price NUMERIC(5,2) NOT NULL,
    Pid INT NOT NULL,
    Description INT NOT NULL,
    PRIMARY KEY (Pid)
);
CREATE TABLE Branch
(
    Bid INT NOT NULL,
    Telephone VARCHAR(10) NOT NULL,
    Address VARCHAR(25) NOT NULL,
    PRIMARY KEY (Bid)
);
CREATE TABLE Worker
(
    Wid INT NOT NULL,
    Salary NUMERIC(5,2) NOT NULL,
    Name VARCHAR(20) NOT NULL,
    PhoneNumber VARCHAR(10) NOT NULL,
    Rating INT NOT NULL,
    Bid INT NOT NULL,
    PRIMARY KEY (Wid),
    FOREIGN KEY (Bid) REFERENCES Branch(Bid)
);
CREATE TABLE Supervisor
(
    Wid INT NOT NULL,
    PRIMARY KEY (Wid),
    FOREIGN KEY (Wid) REFERENCES Worker(Wid)
);
CREATE TABLE Deliveryboy
(
```

```
License_exp_date DATE NOT NULL,  
Wid INT NOT NULL,  
PRIMARY KEY (Wid),  
FOREIGN KEY (Wid) REFERENCES Worker(Wid)  
);
```

```
CREATE TABLE Pizza_Order  
(  
    Order_Date DATE NOT NULL,  
    Oid INT NOT NULL,  
    Is_Delivered INT NOT NULL,  
    Rating INT NOT NULL,  
    Price NUMERIC(5,2) NOT NULL,  
    Cid INT NOT NULL,  
    Bid INT NOT NULL,  
    Wid INT NOT NULL,  
    PRIMARY KEY (Oid),  
    FOREIGN KEY (Cid) REFERENCES Customer(Cid),  
    FOREIGN KEY (Bid) REFERENCES Branch(Bid),  
    FOREIGN KEY (Wid) REFERENCES Worker(Wid)  
);
```

```
CREATE TABLE Delivery_Order  
(  
    Address VARCHAR(25) NOT NULL,  
    Oid INT NOT NULL,  
    Wid INT NOT NULL,  
    PRIMARY KEY (Oid),  
    FOREIGN KEY (Oid) REFERENCES Pizza_Order(Oid),  
    FOREIGN KEY (Wid) REFERENCES Deliveryboy(Wid)  
);
```

```
CREATE TABLE Order_Product  
(  
    amount INT NOT NULL,  
    Pid INT NOT NULL,  
    Oid INT NOT NULL,  
    PRIMARY KEY (Pid, Oid),  
    FOREIGN KEY (Pid) REFERENCES Product(Pid),  
    FOREIGN KEY (Oid) REFERENCES Pizza_Order(Oid)  
);
```

מחיקת טבלאות 1.4.2

```
drop table delivery_order;  
drop table order_product;  
drop table pizza_order;
```

```

drop table product;
drop table supervisor;
drop table deliveryboy;
drop table worker;
drop table branch;

```

3NF בדיקת 1.4.3

כפי שנלמד בקורס הקודם, טבלה מנורמלת ב-3NF אם עבור כל תלוות של הטבלה, התולוה (הצד השמאלי) הוא מפתח עליון, או שכל חלק מהתלווי (הצד הימני) מוכל בתולוה או מוכל במפתח כלשהו של הטבלה.

סניף: התלוויות הן bid → Telephone, Address ולבן הטבלה מנורמלת.

לקוחות: התלוויות הן cid → name, telephone ולבן הטבלה מנורמלת.

מוצר: התלוויות הן pid → price, description ולבן הטבלה מנורמלת.

עובד: התלוויות הן wid → salary, name, phonenum, rating, bid ולבן הטבלה מנורמלת.

מנהל משמרות: אין תלויות, ולבן הנירמול מתקיים באופן ריק.

שליח: התלוויות הן wid → license_exp_date ולבן הטבלה מנורמלת.

הזמנה: התלוויות הן oid, bid → is_delivered, rating, price, cid, wid, order_date ולבן הטבלה מנורמלת.

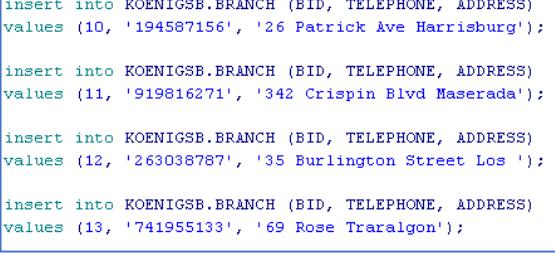
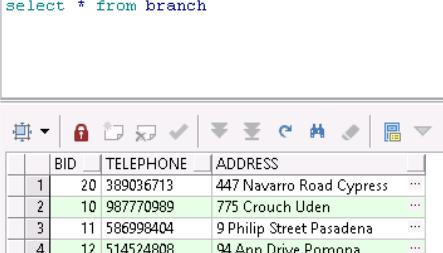
הזמנה במשולח: התלוויות הן oid, bid → address, wid ולבן הטבלה מנורמלת.

פרטי הזמנה: התלוויות הן .pid, oid → amount ולבן הטבלה מנורמלת.

טיעינת נתונים באמצעות Data Generator 1.5

Branch 1.5.1 טיענת נתונים לטבלה

BRANCH			
Owner	Table	Number of records	
KOENIGSB	BRANCH	10	
Name	Type	Size	Data
BID	NUMBER	10	sequence(10,1)
TELEPHONE	VARCHAR2	10	[999999999]
ADDRESS	VARCHAR2	25	Address1 + 'City'

<pre> insert into KOENIGSB.BRANCH (BID, TELEPHONE, ADDRESS) values (10, '194587156', '26 Patrick Ave Harrisburg'); insert into KOENIGSB.BRANCH (BID, TELEPHONE, ADDRESS) values (11, '919816271', '342 Crispin Blvd Maserada'); insert into KOENIGSB.BRANCH (BID, TELEPHONE, ADDRESS) values (12, '263038787', '35 Burlington Street Los '); insert into KOENIGSB.BRANCH (BID, TELEPHONE, ADDRESS) values (13, '741955133', '69 Rose Traralgon'); </pre>	<pre> select * from branch </pre>
	

Customer טעינת נתונים לטבלה 1.5.2

<p>CUSTOMER</p> <table border="1" style="width: 100%; border-collapse: collapse; font-family: monospace;"> <tr> <td style="width: 15%;">Owner</td> <td style="width: 15%;">Table</td> <td style="width: 70%;">Number of records</td> </tr> <tr> <td>KOENIGSB</td> <td>CUSTOMER</td> <td>800.1000</td> </tr> <tr> <td colspan="3"> <table border="1" style="width: 100%; border-collapse: collapse; font-family: monospace;"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Size</th> <th>Data</th> </tr> </thead> <tbody> <tr> <td>NAME</td> <td>VARCHAR2</td> <td>20</td> <td>FirstName + ''LastName</td> </tr> <tr> <td>CID</td> <td>NUMBER</td> <td></td> <td>sequence(2000,1)</td> </tr> <tr> <td>TELEPHONE</td> <td>VARCHAR2</td> <td>10</td> <td>[999999999]</td> </tr> </tbody> </table> </td> </tr> </table> <pre>insert into KOENIGSB.CUSTOMER (NAME, CID, TELEPHONE) values ('Max Sylvian', 2000, '047459367'); insert into KOENIGSB.CUSTOMER (NAME, CID, TELEPHONE) values ('Forest Salt', 2001, '925464704'); insert into KOENIGSB.CUSTOMER (NAME, CID, TELEPHONE) values ('Gord McCain', 2002, '407437857'); insert into KOENIGSB.CUSTOMER (NAME, CID, TELEPHONE) values ('Richard Henriksen', 2003, '563793735');</pre>	Owner	Table	Number of records	KOENIGSB	CUSTOMER	800.1000	<table border="1" style="width: 100%; border-collapse: collapse; font-family: monospace;"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Size</th> <th>Data</th> </tr> </thead> <tbody> <tr> <td>NAME</td> <td>VARCHAR2</td> <td>20</td> <td>FirstName + ''LastName</td> </tr> <tr> <td>CID</td> <td>NUMBER</td> <td></td> <td>sequence(2000,1)</td> </tr> <tr> <td>TELEPHONE</td> <td>VARCHAR2</td> <td>10</td> <td>[999999999]</td> </tr> </tbody> </table>			Name	Type	Size	Data	NAME	VARCHAR2	20	FirstName + ''LastName	CID	NUMBER		sequence(2000,1)	TELEPHONE	VARCHAR2	10	[999999999]	<p>select * from customer</p> <table border="1" style="width: 100%; border-collapse: collapse; font-family: monospace;"> <tr> <td style="width: 15%;">CID</td> <td style="width: 15%;">NAME</td> <td style="width: 15%;">TELEPHONE</td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> </tr> <tr> <td>1</td> <td>Claude Stampley</td> <td>2664</td> <td>169738694</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>Wade Ford</td> <td>2665</td> <td>443490714</td> <td></td> <td></td> <td></td> </tr> <tr> <td>3</td> <td>Aimee Dunn</td> <td>2666</td> <td>536173907</td> <td></td> <td></td> <td></td> </tr> <tr> <td>4</td> <td>Vendetta Zappacosta</td> <td>2667</td> <td>416381049</td> <td></td> <td></td> <td></td> </tr> <tr> <td>5</td> <td>Rosie Cale</td> <td>2668</td> <td>275688019</td> <td></td> <td></td> <td></td> </tr> <tr> <td>6</td> <td>Darius Coltrane</td> <td>2669</td> <td>635389802</td> <td></td> <td></td> <td></td> </tr> <tr> <td>7</td> <td>Mili Ribisi</td> <td>2670</td> <td>604387302</td> <td></td> <td></td> <td></td> </tr> <tr> <td>8</td> <td>Billy Cocker</td> <td>2671</td> <td>352623145</td> <td></td> <td></td> <td></td> </tr> </table>	CID	NAME	TELEPHONE					1	Claude Stampley	2664	169738694				2	Wade Ford	2665	443490714				3	Aimee Dunn	2666	536173907				4	Vendetta Zappacosta	2667	416381049				5	Rosie Cale	2668	275688019				6	Darius Coltrane	2669	635389802				7	Mili Ribisi	2670	604387302				8	Billy Cocker	2671	352623145			
Owner	Table	Number of records																																																																																							
KOENIGSB	CUSTOMER	800.1000																																																																																							
<table border="1" style="width: 100%; border-collapse: collapse; font-family: monospace;"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Size</th> <th>Data</th> </tr> </thead> <tbody> <tr> <td>NAME</td> <td>VARCHAR2</td> <td>20</td> <td>FirstName + ''LastName</td> </tr> <tr> <td>CID</td> <td>NUMBER</td> <td></td> <td>sequence(2000,1)</td> </tr> <tr> <td>TELEPHONE</td> <td>VARCHAR2</td> <td>10</td> <td>[999999999]</td> </tr> </tbody> </table>			Name	Type	Size	Data	NAME	VARCHAR2	20	FirstName + ''LastName	CID	NUMBER		sequence(2000,1)	TELEPHONE	VARCHAR2	10	[999999999]																																																																							
Name	Type	Size	Data																																																																																						
NAME	VARCHAR2	20	FirstName + ''LastName																																																																																						
CID	NUMBER		sequence(2000,1)																																																																																						
TELEPHONE	VARCHAR2	10	[999999999]																																																																																						
CID	NAME	TELEPHONE																																																																																							
1	Claude Stampley	2664	169738694																																																																																						
2	Wade Ford	2665	443490714																																																																																						
3	Aimee Dunn	2666	536173907																																																																																						
4	Vendetta Zappacosta	2667	416381049																																																																																						
5	Rosie Cale	2668	275688019																																																																																						
6	Darius Coltrane	2669	635389802																																																																																						
7	Mili Ribisi	2670	604387302																																																																																						
8	Billy Cocker	2671	352623145																																																																																						

Worker טעינת נתונים לטבלה 1.5.3

<p>WORKER</p> <table border="1" style="width: 100%; border-collapse: collapse; font-family: monospace;"> <tr> <td style="width: 15%;">Owner</td> <td style="width: 15%;">Table</td> <td style="width: 70%;">Number of records</td> </tr> <tr> <td>KOENIGSB</td> <td>WORKER</td> <td>1000.1200</td> </tr> <tr> <td colspan="3"> <table border="1" style="width: 100%; border-collapse: collapse; font-family: monospace;"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Size</th> <th>Data</th> </tr> </thead> <tbody> <tr> <td>WID</td> <td>NUMBER</td> <td></td> <td>sequence(3000,1)</td> </tr> <tr> <td>SALARY</td> <td>NUMBER</td> <td>5,2</td> <td>random(28.00,45.00)</td> </tr> <tr> <td>NAME</td> <td>VARCHAR2</td> <td>20</td> <td>FirstName + ''LastName</td> </tr> <tr> <td>PHONENUMBER</td> <td>VARCHAR2</td> <td>10</td> <td>[999999999]</td> </tr> <tr> <td>RATING</td> <td>NUMBER</td> <td></td> <td>random(1,10)</td> </tr> <tr> <td>BID</td> <td>NUMBER</td> <td></td> <td>list(select bid from branch)</td> </tr> </tbody> </table> </td> </tr> </table> <pre>insert into KOENIGSB.WORKER (WID, SALARY, NAME, PHONENUMBER, RATING, BID) values (3000, 42.77, 'Brothers Shue', '733639044', 1, 11); insert into KOENIGSB.WORKER (WID, SALARY, NAME, PHONENUMBER, RATING, BID) values (3001, 29.38, 'Angie Phifer', '066710423', 10, 18); insert into KOENIGSB.WORKER (WID, SALARY, NAME, PHONENUMBER, RATING, BID) values (3002, 39.90, 'Cevin Furtado', '628089643', 8, 20); insert into KOENIGSB.WORKER (WID, SALARY, NAME, PHONENUMBER, RATING, BID) values (3003, 36.65, 'Robin Bruce', '750251942', 5, 17); insert into KOENIGSB.WORKER (WID, SALARY, NAME, PHONENUMBER, RATING, BID) values (3004, 30.37, 'Suzanne Hampton', '880462200', 2, 20);</pre>	Owner	Table	Number of records	KOENIGSB	WORKER	1000.1200	<table border="1" style="width: 100%; border-collapse: collapse; font-family: monospace;"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Size</th> <th>Data</th> </tr> </thead> <tbody> <tr> <td>WID</td> <td>NUMBER</td> <td></td> <td>sequence(3000,1)</td> </tr> <tr> <td>SALARY</td> <td>NUMBER</td> <td>5,2</td> <td>random(28.00,45.00)</td> </tr> <tr> <td>NAME</td> <td>VARCHAR2</td> <td>20</td> <td>FirstName + ''LastName</td> </tr> <tr> <td>PHONENUMBER</td> <td>VARCHAR2</td> <td>10</td> <td>[999999999]</td> </tr> <tr> <td>RATING</td> <td>NUMBER</td> <td></td> <td>random(1,10)</td> </tr> <tr> <td>BID</td> <td>NUMBER</td> <td></td> <td>list(select bid from branch)</td> </tr> </tbody> </table>			Name	Type	Size	Data	WID	NUMBER		sequence(3000,1)	SALARY	NUMBER	5,2	random(28.00,45.00)	NAME	VARCHAR2	20	FirstName + ''LastName	PHONENUMBER	VARCHAR2	10	[999999999]	RATING	NUMBER		random(1,10)	BID	NUMBER		list(select bid from branch)	<p>select * from worker</p> <table border="1" style="width: 100%; border-collapse: collapse; font-family: monospace;"> <tr> <td style="width: 15%;">WID</td> <td style="width: 15%;">SALARY</td> <td style="width: 15%;">NAME</td> <td style="width: 15%;">PHONENUMBER</td> <td style="width: 15%;">RATING</td> <td style="width: 15%;">BID</td> </tr> <tr> <td>1</td> <td>3508</td> <td>37.55</td> <td>Natasha Shearer</td> <td>074900329</td> <td>1 18</td> </tr> <tr> <td>2</td> <td>3509</td> <td>44.65</td> <td>Ving Aiken</td> <td>616434686</td> <td>2 16</td> </tr> <tr> <td>3</td> <td>3510</td> <td>34.90</td> <td>Charles Young</td> <td>976055338</td> <td>1 17</td> </tr> <tr> <td>4</td> <td>3511</td> <td>29.04</td> <td>Chuck Tripplehorn</td> <td>422130944</td> <td>6 13</td> </tr> <tr> <td>5</td> <td>3512</td> <td>43.49</td> <td>Steven LaPaglia</td> <td>146588284</td> <td>5 12</td> </tr> <tr> <td>6</td> <td>3513</td> <td>31.73</td> <td>Helen Page</td> <td>681243723</td> <td>8 12</td> </tr> <tr> <td>7</td> <td>3514</td> <td>28.99</td> <td>Andie Doucette</td> <td>162802894</td> <td>1 16</td> </tr> <tr> <td>8</td> <td>3515</td> <td>42.83</td> <td>Ermymou Stampley</td> <td>286977070</td> <td>2 18</td> </tr> <tr> <td>9</td> <td>3516</td> <td>34.43</td> <td>Gena Sampson</td> <td>534727199</td> <td>9 19</td> </tr> <tr> <td>10</td> <td>3517</td> <td>37.39</td> <td>Stellan Manuski</td> <td>0003057364</td> <td>6 13</td> </tr> </table>	WID	SALARY	NAME	PHONENUMBER	RATING	BID	1	3508	37.55	Natasha Shearer	074900329	1 18	2	3509	44.65	Ving Aiken	616434686	2 16	3	3510	34.90	Charles Young	976055338	1 17	4	3511	29.04	Chuck Tripplehorn	422130944	6 13	5	3512	43.49	Steven LaPaglia	146588284	5 12	6	3513	31.73	Helen Page	681243723	8 12	7	3514	28.99	Andie Doucette	162802894	1 16	8	3515	42.83	Ermymou Stampley	286977070	2 18	9	3516	34.43	Gena Sampson	534727199	9 19	10	3517	37.39	Stellan Manuski	0003057364	6 13
Owner	Table	Number of records																																																																																																						
KOENIGSB	WORKER	1000.1200																																																																																																						
<table border="1" style="width: 100%; border-collapse: collapse; font-family: monospace;"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Size</th> <th>Data</th> </tr> </thead> <tbody> <tr> <td>WID</td> <td>NUMBER</td> <td></td> <td>sequence(3000,1)</td> </tr> <tr> <td>SALARY</td> <td>NUMBER</td> <td>5,2</td> <td>random(28.00,45.00)</td> </tr> <tr> <td>NAME</td> <td>VARCHAR2</td> <td>20</td> <td>FirstName + ''LastName</td> </tr> <tr> <td>PHONENUMBER</td> <td>VARCHAR2</td> <td>10</td> <td>[999999999]</td> </tr> <tr> <td>RATING</td> <td>NUMBER</td> <td></td> <td>random(1,10)</td> </tr> <tr> <td>BID</td> <td>NUMBER</td> <td></td> <td>list(select bid from branch)</td> </tr> </tbody> </table>			Name	Type	Size	Data	WID	NUMBER		sequence(3000,1)	SALARY	NUMBER	5,2	random(28.00,45.00)	NAME	VARCHAR2	20	FirstName + ''LastName	PHONENUMBER	VARCHAR2	10	[999999999]	RATING	NUMBER		random(1,10)	BID	NUMBER		list(select bid from branch)																																																																										
Name	Type	Size	Data																																																																																																					
WID	NUMBER		sequence(3000,1)																																																																																																					
SALARY	NUMBER	5,2	random(28.00,45.00)																																																																																																					
NAME	VARCHAR2	20	FirstName + ''LastName																																																																																																					
PHONENUMBER	VARCHAR2	10	[999999999]																																																																																																					
RATING	NUMBER		random(1,10)																																																																																																					
BID	NUMBER		list(select bid from branch)																																																																																																					
WID	SALARY	NAME	PHONENUMBER	RATING	BID																																																																																																			
1	3508	37.55	Natasha Shearer	074900329	1 18																																																																																																			
2	3509	44.65	Ving Aiken	616434686	2 16																																																																																																			
3	3510	34.90	Charles Young	976055338	1 17																																																																																																			
4	3511	29.04	Chuck Tripplehorn	422130944	6 13																																																																																																			
5	3512	43.49	Steven LaPaglia	146588284	5 12																																																																																																			
6	3513	31.73	Helen Page	681243723	8 12																																																																																																			
7	3514	28.99	Andie Doucette	162802894	1 16																																																																																																			
8	3515	42.83	Ermymou Stampley	286977070	2 18																																																																																																			
9	3516	34.43	Gena Sampson	534727199	9 19																																																																																																			
10	3517	37.39	Stellan Manuski	0003057364	6 13																																																																																																			

1.5.4 טעינת נתונים לטבלה Supervisor

SUPERVISOR

Owner	Table	Number of records
KOENIGSB	SUPERVISOR	50.150

Name	Type	Size	Data
WID	NUMBER		sequence(3508, 5)
*			

```

insert into KOENIGSB.SUPERVISOR (WID)
values (3508);

insert into KOENIGSB.SUPERVISOR (WID)
values (3513);

insert into KOENIGSB.SUPERVISOR (WID)
values (3518);

insert into KOENIGSB.SUPERVISOR (WID)
values (3523);

```

select * from supervisor

WID	
1	3508
2	3513
3	3518
4	3523
5	3528
6	3533
7	3538

1.5.5 טעינת נתונים לטבלה Deliveryboy

DELIVERYBOY

Owner	Table	Number of records
KOENIGSB	DELIVERYBOY	200.300

Name	Type	Size	Data
LICENSE_EXP_DATE	DATE		random(01/01/2021, 01/01/2030)
WID	NUMBER		sequence(3509, 2)
*			

```

insert into KOENIGSB.DELIVERYBOY (LICENSE_EXP_DATE, WID)
values (to_date('13-11-2023', 'dd-mm-yyyy'), 3509);

insert into KOENIGSB.DELIVERYBOY (LICENSE_EXP_DATE, WID)
values (to_date('26-02-2024', 'dd-mm-yyyy'), 3511);

insert into KOENIGSB.DELIVERYBOY (LICENSE_EXP_DATE, WID)
values (to_date('20-04-2023', 'dd-mm-yyyy'), 3513);

insert into KOENIGSB.DELIVERYBOY (LICENSE_EXP_DATE, WID)
values (to_date('02-06-2025', 'dd-mm-yyyy'), 3515);

```

select * from Deliveryboy

LICENSE_EXP_DATE	WID
26/12/2027	3509
29/10/2021	3511
23/02/2021	3513
24/07/2027	3515
10/06/2028	3517

Pizza_Order טעינת נתונים לטבלה 1.5.6

PIZZA_ORDER			
Owner	Table	Number of records	
KOENIGSB	PIZZA_ORDER	1500.2000	
Name	Type	Size	Data
ORDER_DATE	DATE		random(01/01/1993,02/23/2020) + ','random(00:00,23:59)
OID	NUMBER		sequence(5000, 1)
IS_DELIVERED	NUMBER		list('0'(1), '1'(20))
RATING	NUMBER		random(1,10)
PRICE	NUMBER	5,2	random(0,1)
CID	NUMBER		list(select cid from customer)
BID	NUMBER		list(select bid from branch)
WID	NUMBER		list(select wid from worker)

```
select * from Pizza_Order
```

ORDER_DATE	OID	IS_DELIVERED	RATING	PRICE	CID	BID	WID
28/06/2880 18:28:22	5392	1	6	0.80	2230	12	3597
08/01/2992 16:24:57	5393	1	4	0.80	2079	17	3693
09/03/3051 14:23:16	5394	1	1	0.69	2151	16	3885
12/12/2087 19:53:28	5395	1	10	0.48	2851	17	3212

```

insert into KOENIGSB.PIZZA_ORDER (ORDER_DATE, OID, IS_DELIVERED, RATING, PRICE, CID, BID, WID)
values (to_date('29-07-2344 04:00:37', 'dd-mm-yyyy hh24:miss')), 5000, 1, 2, 0.91, 2075, 14, 3629);

insert into KOENIGSB.PIZZA_ORDER (ORDER_DATE, OID, IS_DELIVERED, RATING, PRICE, CID, BID, WID)
values (to_date('08-11-2010 02:52:22', 'dd-mm-yyyy hh24:miss')), 5001, 1, 4, 0.20, 2293, 15, 3569);

insert into KOENIGSB.PIZZA_ORDER (ORDER_DATE, OID, IS_DELIVERED, RATING, PRICE, CID, BID, WID)
values (to_date('15-10-2261 04:46:59', 'dd-mm-yyyy hh24:miss')), 5002, 1, 9, 0.77, 2612, 15, 3315);

insert into KOENIGSB.PIZZA_ORDER (ORDER_DATE, OID, IS_DELIVERED, RATING, PRICE, CID, BID, WID)
values (to_date('02-08-2647 06:24:54', 'dd-mm-yyyy hh24:miss')), 5003, 1, 3, 0.96, 2293, 10, 3065);

insert into KOENIGSB.PIZZA_ORDER (ORDER_DATE, OID, IS_DELIVERED, RATING, PRICE, CID, BID, WID)
values (to_date('11-05-2631 22:18:54', 'dd-mm-yyyy hh24:miss')), 5004, 1, 9, 0.75, 2481, 15, 3001);

```

Delivery_Order טעינת נתונים לטבלה 1.5.7

DELIVERY_ORDER			
Owner	Table	Number of records	
KOENIGSB	DELIVERY_ORDER	800.1000	
Name	Type	Size	Data
ADDRESS	VARCHAR2	25	Address1 + ''City
OID	NUMBER		sequence(5000,2)
WID	NUMBER		list(select wid from deliveryboy)
*			

```
select * from Delivery_Order
```

ADDRESS	OID	WID
988 Loeb Plymouth Meeting	5378	3701
23rd Street Le chesnay	5380	3979
91 Tilly Road Oosterhout	5382	3947
54 Frances Milsons Point	5384	3699

```

insert into KOENIGSB.DELIVERY_ORDER (ADDRESS, OID, WID)
values ('88 Patton Road Rotterdam', 5000, 3685);

insert into KOENIGSB.DELIVERY_ORDER (ADDRESS, OID, WID)
values ('13 Ewan Blvd Santana do p', 5002, 3885);

insert into KOENIGSB.DELIVERY_ORDER (ADDRESS, OID, WID)
values ('2 Matarazzo Road Birmingh', 5004, 3563);

insert into KOENIGSB.DELIVERY_ORDER (ADDRESS, OID, WID)
values ('9 Kris Road Suffern', 5006, 3801);

```

Product (without generator) 1.5.8 טעינת נתונים לטבלה

```
insert into product(price, pid, Description )
values(55.00, 1, 'Plain Pie');

insert into product(price, pid, Description )
values(63.00, 2, 'Pie with extra cheese');

insert into product(price, pid, Description )
values(60.00, 3, 'Pie with extra veg');

insert into product(price, pid, Description )
values(65.00, 4, 'Pie with 2 extra veg');

insert into product(price, pid, Description )
values(68.00, 5, 'Pie with extra veg and cheese');

insert into product(price, pid, Description )
values(7.00, 6, 'Water Bottle');

insert into product(price, pid, Description )
values(9.00, 7, 'Soda Bottle');
```

```
select * from Product
```

	PRICE	PID	DESCRIPTION	...
▶	1	55.00	1 Plain Pie	...
▶	2	63.00	2 Pie with extra cheese	...
▶	3	60.00	3 Pie with extra veg	...
▶	4	65.00	4 Pie with 2 extra veg	...
▶	5	68.00	5 Pie with extra veg and cheese	...
▶	6	7.00	6 Water Bottle	...
▶	7	9.00	7 Soda Bottle	...
▶	8	4.00	8 Ice Pop	...
▶	9	8.00	9 Icecream	...
▶	10	10.00	10 Garlic Bread	...
▶	11	15.00	11 Focaccia	...
▶	12	7.00	12 Coffee	...
▶	13	16.00	13 Cheese Bites	...
▶	14	8.00	14 Coffee with Milk	...

Order_Product טעינת נתונים לטבלה 1.5.9

OWNER: KOENIGSB
TABLE: ORDER_PRODUCT
Number of records: 1936

Name	Type	Size	Data
AMOUNT	NUMBER		list('1'(10),'2'(3),'3'(1),'4'(1))
PID	NUMBER		list(select pid from product)
OID	NUMBER		sequence(5000,1)

```
select * from Order_product
```

```
insert into KOENIGSB.ORDER_PRODUCT (AMOUNT, PID, OID)
values (1, 11, 5000);

insert into KOENIGSB.ORDER_PRODUCT (AMOUNT, PID, OID)
values (1, 11, 5001);

insert into KOENIGSB.ORDER_PRODUCT (AMOUNT, PID, OID)
values (2, 1, 5002);

insert into KOENIGSB.ORDER_PRODUCT (AMOUNT, PID, OID)
values (3, 9, 5003);

insert into KOENIGSB.ORDER_PRODUCT (AMOUNT, PID, OID)
values (1, 14, 5004);
```

	AMOUNT	PID	OID
▶	1	3	6874
▶	2	1	6877
▶	3	1	6878
▶	4	2	6886
▶	5	1	6887
▶	6	1	6888
▶	7	3	6889

Text Importer 1.6 טעינת נתונים בעזרת טקסט

CustomersText - Notepad

File Edit Format View Help

Robert Morgan,68075659,9615
Gladys Ewing,32701085,9050
Ervin Shaw,85830236,7861

```
insert into customer (name, telephone, cid)
values ('Robert Morgan', '68075659', 9615);

insert into customer (name, telephone, cid)
values ('Gladys Ewing', '32701085', 9050);

insert into customer (name, telephone, cid)
values ('Ervin Shaw', '85830236', 7861);
```

Datas from Textfile: Data to Oracle

General
Owner: CUSTOMER
Table: CUSTOMER
Commit every: 0
Overwrite duplicates: Ignore duplicates:
Delete records: Truncate table:

Initializing Script:
Finalizing Script:

Fields
Field1 -> NAME NVARCHAR2
Field2 -> TELEPHONE NVARCHAR2
Field3 -> CID NUMBER

Field: CID (NUMBER)
Fieldtype: Number
Create SQL
SQL function: additional Oracle processing, for

Result Preview

1	2	3
Robert Morgan	68075659	9615
Gladys Ewing	32701085	9050
Ervin Shaw	85830236	7861

1.7 תיקון נתונים עבור התאמת מציאותית

```
update pizza_order p
set bid = (select bid from worker where wid = p.wid);

update worker w
set rating = (select avg(rating* is_delivered) from pizza_order group by wid having wid = w.wid)
where wid in (select wid from pizza_order);

update pizza_order p
set price = (select sum(price*amount)
             from order_product natural join product
             group by oid
             having oid = p.oid)
```

1.7.1 תפקיד הפקודה הראשונה היא לסדר שכל הזמנה Taboa מהסניף שבו עובד העובד האחראי על הזמנה. אמינו מוקדם שאין זה מוכרת, כי יתכן שעובד יעבד סניף, או שייעבוד באופן זמני בסניף אחר, אך עשינו זאת למען הסדר הטוב.

1.7.2 תפקיד הפקודה השנייה היא להזין דירוג לעובדים לפי ממוצע דירוג הזמנות שלהם שהגיעו ליעדם.

1.7.3 תפקיד הפקודה השלישית היא להזין להזמנה מחיר לפי סכום מכפלת מחירי המוצרים בהזמנה עם הכמות שללה.

שלב 2

Select 2.1

השאילתה הראשוונה

שאילתה שמטרתה להציג את פרטי הסניף שהכנסה שלו מכירות כי גדולה:

```
with sum_price_branch as (select bid, sum(total_price) as sumPrice
                           from pizza_order
                           group by bid)
select sumPrice as "Total Income", Telephone, Address
from sum_price_branch natural join branch
where sumPrice >= all(select sumPrice
                      from sum_price_branch);
```

התוצאה:

	Total Income	TELEPHONE	ADDRESS
▶ 1	13982	437867305	10 Gertner Street Fair La ...

השאילתה השנייה

שאילתה שמצוינה את פרטי המוצר הזול ביותר, ואת פרטי המוצר היקר ביותר:

```
select 'Most Expensive Product:' as Category , price, description from product
where price >= all(select price from product)
union
select 'Cheapest Product:' as Category , price, description from product
where price <= all(select price from product)
```

התוצאה:

	CATEGORY	PRICE	DESCRIPTION
▶ 1	Cheapest Product: ...	4.00	Ice Pop ...
2	Most Expensive Product: ...	68.00	Pie with extra veg and cheese ...

השאילתה השלישית

שאילתה שמרתה מהו המוצר הכי נמכר בכל סניף:

```
select bid, description as "Favorite product", count(pid)as "Number of times sold"
from order_product natural join product natural join pizza_order P
group by bid, description
having count(pid) >= all (select count(pid)
                           from order_product
                           natural join product
                           natural join pizza_order
                           where bid = P.Bid
                           group by bid,pid)
order by bid;
```

	BID	Favorite product	Number of times sold
▶ 1	10	Coffee ...	29
2	11	Pie with extra cheese ...	29
3	12	Pie with extra veg and cheese ...	18

התוצאה:

השאילתה הרביעית 2.1.4

שאילתה שטירה מהו המוצר שמכניס הכי הרבה ברשות, וכמה הוא הכנסה

```
select pid as "best selling product", description, sum(price*amount) as "Total Income"
from product natural join order_product
group by pid, description
having sum(price*amount) >= all(select sum(price*amount)
                                from order_product natural join product
                                group by pid)
```

התוצאה:

	best selling product	DESCRIPTION	Total Income
▶ 1	5	Pie with extra veg and cheese	21216

השאילתה החמישית 2.1.5

שאילתה שמציאה זוגות עובדים עם אותו שם משפחה (שימוש בביטויים רגולריים):

```
with splitname as (
    select wid,
           regexp_substr(name, '[^ ]*') as firstname,
           regexp_substr(name, ' .*') as lastname
      from worker)
select l1.firstname as "Worker 1",
       l2.firstname as "Worker 2",
       l1.lastname as "Common Lastname"
  from splitname l1, splitname l2
 where l1.lastname = l2.lastname
   and l1.wid > l2.wid
  order by l1.lastname;
```

התוצאה:

	Worker 1	Worker 2	Common Lastname
1	Chuck	France	Addy
2	Buddy	France	Addy
3	Buddy	Chuck	Addy
4	Jonatha	Zooey	Akins
5	Frank	Cledus	Alda
6	Paula	Albertina	Allen
7	Fairuza	Angela	Almond
8	John	Gaby	Alston

השאילתה השישית 2.1.6

שאילתה שמציגה עבור כל חודש את מחיר ההזמנה הממוצעת עבור אותו חודש, עם אפשרות לחישוב של כמה שנים אחרת על ידי פרמטר:

```
select month, round(avg(total_price),2) as avg_order_price
from (select to_char(order_date, 'MM') as month, total_price
      from pizza_order
     where order_date >=
           add_months(sysdate, -12*&years_to_calculate)) month_price
group by month
order by month;
```

התוצאות:

	MONTH	AVG_ORDER_PRICE
►	1 01	71.46
2	02	74.26
3	03	74.02
4	04	69.66
5	05	70.36
6	06	76.75
7	07	68.99
8	08	64.88
9	09	71.31
10	10	70.26
11	11	66.02
12	12	75.73

Variables	
Name	Value
&years_to_calculate	5

השאילתה השביעית 2.1.7

שאילתה שמציגה כמה ההזנות נעשו בשעה שהמשתמש בוחר:

```
select order_date from pizza_order
order by to_char(order_date, 'HH24');
select to_char(order_date, 'HH24') as hour, count(*) as total_orders
from pizza_order
group by to_char(order_date, 'HH24')
having to_char(order_date, 'HH24') = &hour;
```

	HOUR	TOTAL_ORDERS
►	1 23	66

Variables	
Name	Value
&hour	23

2.1.8 השאלתה השמינית

השאילתה הבאה מציגה עבור כל סניף את שם הסניף (תוך שימוש בכינוי רגולרי כדי להלץ את שם הרחוב והעיר מהכתובת) ואת מספר הזמנות מאותו סניף שמהירות עולה על המחיר הממוצע של הזמנות עבור אותו סניף, ועל המחיר הממוצע של הזמנות בכל הדרשת.

```
select regexp_substr(address, '.*') as "Branch Name",
       count(*) as "Number of orders above average"
  from branch natural join pizza_order
 where total_price >= (select avg(total_price)
                           from pizza_order)
        and total_price >= (select avg(total_price)
                               from pizza_order po
                             where po.bid = bid)
 group by address
 order by "Branch Name";
```

התוצאות:

		Branch Name	Number of orders above average
▶	1	Costa Spring Valley ...	62
	2	Gertner Street Fair La ...	66
	3	Gladys Drive Fambach ...	61
	4	Goiania Blvd Pirapora ...	50
	5	Hauer Blvd Waltham ...	65
	6	Hohenfels Alexandria ...	49
	7	Louise Street Barueri ...	52
	8	Ness Road Oulu ...	56
	9	Perrineau Street Read ...	39
	10	Skerritt Street Cheshi ...	36

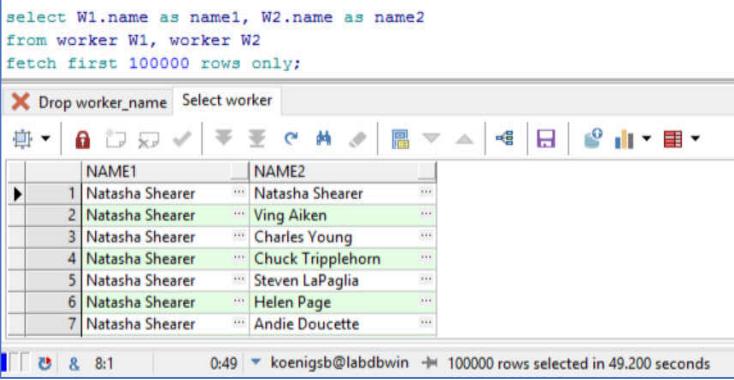
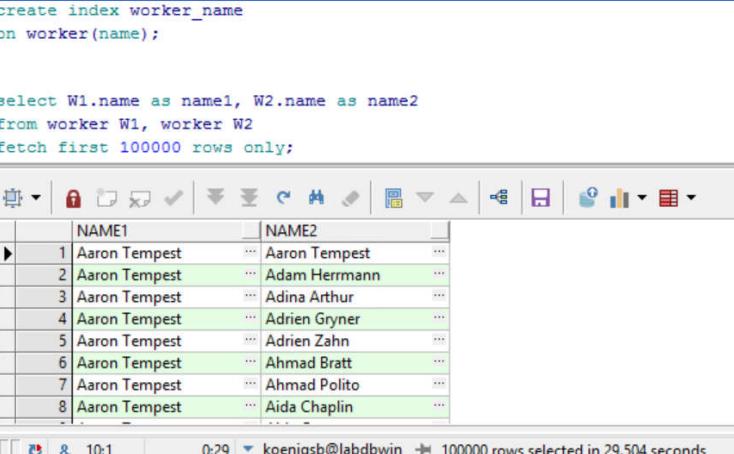
2.2 אינדקסים

```
create index worker_name  
in worker(name);  
  
create index customer_name  
on customer(name);
```

2.2.1 אינדקס עבור שמות העובדים.

2.2.2 אינדקס עבור שמות הלקוחות.

2.2.3 שיפור זמן הריצה של שאלתה בעקבות שימוש באינדקס:

<pre>select W1.name as name1, W2.name as name2 from worker W1, worker W2 fetch first 100000 rows only;</pre>		בלוי אינדקס
<pre>create index worker_name on worker(name); select W1.name as name1, W2.name as name2 from worker W1, worker W2 fetch first 100000 rows only;</pre>		עם אינדקס

Update 2.3 שאלות

```
-- defining the salary
update worker
set salary = 28 + (weekly_hours-16)/3;

-- bonus for supervisors
update worker
set salary = salary + 3
where wid in (select wid from supervisor);

--bonus for outstanding workers
update worker
set salary = salary*1.07
where wid in (select wid from outstanding_workers);
```

**2.3.1 שאלתה שמנדרה לכל עובד את
השכר השנתי שלו.**

2.3.2 שאילתת שנותנת בonus לכל אחראי משמרת.

2.3.3 שאלתה שנותנת בonus נוסף לעובדים מצטיינים.

הצורך: אם יש شيئاים במערך העובדים בחברה, צריך לעדכן את השכר (מינוי עובך כאחראי שומרת), והורדת אחראי שמרת לעובך רגיל, עובד שקיבל הצעינות, וכו').

לפני עדכון		2.2.1 אחרי		2.2.2 אחרי		2.2.3 לפני	
	WID		WID	SALARY		WID	SALARY
▶	1	3102	36.00	▶	1	3508	39.00
	2	3509	30.67		2	3509	30.67
	3	3510	33.33		3	3510	33.33
	4	3511	38.67		4	3511	38.67
	5	3512	37.33		5	3512	37.33
	6	3513	39.00		6	3513	36.00
	7	3514	29.33		7	3514	29.33
	8	3516	32.00		8	3516	32.00
	9	3517	34.67		9	3517	34.67
	10	3518	31.00		10	3518	28.00
	11	3519	28.00		11	3519	28.00
	12	3520	30.67		12	3520	30.67
	13	3521	33.33		13	3521	33.33
	14	3522	36.00		14	3522	36.00
					15	3523	36.00

Delete 2.4 שאלה

2.4.1 שאלתה שטרתת למחוק מספר עובדים מכל סניף שיש בו יותר ממאות מקסימלית נתונה של עובדים:

 Variables <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr><th>Name</th><th>Value</th></tr> <tr><td>upper_limit</td><td>100</td></tr> <tr><td>amount_to_delete</td><td>5</td></tr> </table>	Name	Value	upper_limit	100	amount_to_delete	5	<pre style="font-family: monospace; padding: 10px;">delete from worker where wid in (select wid from worker where bid in (select bid from worker group by bid having count(wid) > &upper_limit order by rating fetch first &amount to delete rows only);</pre>
Name	Value						
upper_limit	100						
amount_to_delete	5						

לפנינו תוצאות השאלה שסופרת מספר עובדים לכל סניף כאשר יש יותר מ100 עובדים בסניף.

הסבר	אחרי מהיקה	לפני מהיקה														
שני עובדים נמחקו מסניף 18, אחד נמחק מסניף 16, ושני עובדים נמחקו מסניף 15.	<table border="1" style="width: 100%; border-collapse: collapse; border: none;"> <thead> <tr><th>BID</th><th>COUNT(WID)</th></tr> </thead> <tbody> <tr><td>1</td><td>18</td></tr> <tr><td>2</td><td>16</td></tr> </tbody> </table>	BID	COUNT(WID)	1	18	2	16	<table border="1" style="width: 100%; border-collapse: collapse; border: none;"> <thead> <tr><th>BID</th><th>COUNT(WID)</th></tr> </thead> <tbody> <tr><td>1</td><td>18</td></tr> <tr><td>2</td><td>16</td></tr> <tr><td>3</td><td>15</td></tr> </tbody> </table>	BID	COUNT(WID)	1	18	2	16	3	15
BID	COUNT(WID)															
1	18															
2	16															
BID	COUNT(WID)															
1	18															
2	16															
3	15															

2.4.2 מחיקת שליחים שתוקף רשיונם פג לפני 4 חודשים (הממשיכים כעובדים רגילים):

```
delete from deliveryboy
where LICENSE_EXP_DATE < sysdate - 120;
```

לפנינו תוצאות השאלה שמציגה את השליחים ואת תוקף רשיונם לפי סדר עולה:

הסבר	אחרי מהיקה	לפני מהיקה																						
השליחים נעלמו מהתוצאות לאחר המהיקה.	<table border="1" style="width: 100%; border-collapse: collapse; border: none;"> <thead> <tr><th>WID</th><th>LICENSE_EXP_DATE</th></tr> </thead> <tbody> <tr><td>1</td><td>28/01/2021</td></tr> <tr><td>2</td><td>12/02/2021</td></tr> <tr><td>3</td><td>24/02/2021</td></tr> <tr><td>4</td><td>04/04/2021</td></tr> </tbody> </table>	WID	LICENSE_EXP_DATE	1	28/01/2021	2	12/02/2021	3	24/02/2021	4	04/04/2021	<table border="1" style="width: 100%; border-collapse: collapse; border: none;"> <thead> <tr><th>WID</th><th>LICENSE_EXP_DATE</th></tr> </thead> <tbody> <tr><td>1</td><td>3050 31/01/2006</td></tr> <tr><td>2</td><td>3073 09/02/2006</td></tr> <tr><td>3</td><td>3022 04/03/2006</td></tr> <tr><td>4</td><td>3049 27/03/2006</td></tr> <tr><td>5</td><td>3066 20/04/2006</td></tr> </tbody> </table>	WID	LICENSE_EXP_DATE	1	3050 31/01/2006	2	3073 09/02/2006	3	3022 04/03/2006	4	3049 27/03/2006	5	3066 20/04/2006
WID	LICENSE_EXP_DATE																							
1	28/01/2021																							
2	12/02/2021																							
3	24/02/2021																							
4	04/04/2021																							
WID	LICENSE_EXP_DATE																							
1	3050 31/01/2006																							
2	3073 09/02/2006																							
3	3022 04/03/2006																							
4	3049 27/03/2006																							
5	3066 20/04/2006																							

2.4.3 מחיקת לקוחות שלא הזמינו בחצי שנה האחרונות (חשב לזמן כי בטבלה `customer` על העמודה `cid`

כתבתי

:(on delete set null)

```
delete from customer
where cid not in (select distinct cid
                     from customer natural join pizza_order
                     where order_date > sysdate - 180);
```

לפנינו שאלתה שמציגה כל לקוחות ואת תאריך ההזמנה האחרון שלו, ממוין לפי התאריך:

הסבר	אחרי מחיקה	לפני מחיקה																																										
הלקוחות שהופיעו לפני המבוקש נעלמו בתוצאות אחרי המבוקש	<table border="1"><thead><tr><th>CID</th><th>LATEST_ORDER</th></tr></thead><tbody><tr><td>1 2704</td><td>21/01/2020 10:31:43</td></tr><tr><td>2 2036</td><td>23/11/2021 03:43:37</td></tr><tr><td>3 2449</td><td>10/06/2023 10:10:50</td></tr><tr><td>4 2678</td><td>29/10/2024 17:31:20</td></tr><tr><td>5 2481</td><td>05/02/2025 12:07:20</td></tr><tr><td>6 2544</td><td>18/07/2028 11:12:20</td></tr><tr><td>7 2445</td><td>13/04/2029 19:55:29</td></tr><tr><td>8 2418</td><td>05/11/2031 05:29:02</td></tr><tr><td>9 2525</td><td>02/12/2039 10:54:11</td></tr><tr><td>10 2692</td><td>05/10/2042 10:49:52</td></tr></tbody></table>	CID	LATEST_ORDER	1 2704	21/01/2020 10:31:43	2 2036	23/11/2021 03:43:37	3 2449	10/06/2023 10:10:50	4 2678	29/10/2024 17:31:20	5 2481	05/02/2025 12:07:20	6 2544	18/07/2028 11:12:20	7 2445	13/04/2029 19:55:29	8 2418	05/11/2031 05:29:02	9 2525	02/12/2039 10:54:11	10 2692	05/10/2042 10:49:52	<table border="1"><thead><tr><th>CID</th><th>LATEST_ORDER</th></tr></thead><tbody><tr><td>1 2431</td><td>15/04/1993 11:34:22</td></tr><tr><td>2 2001</td><td>03/08/1995 00:41:19</td></tr><tr><td>3 2107</td><td>01/02/1998 01:00:54</td></tr><tr><td>4 2327</td><td>08/10/2002 08:33:12</td></tr><tr><td>5 2714</td><td>31/01/2004 15:39:27</td></tr><tr><td>6 2060</td><td>20/12/2007 22:55:37</td></tr><tr><td>7 2736</td><td>22/08/2008 14:54:14</td></tr><tr><td>8 2897</td><td>02/12/2014 16:52:27</td></tr><tr><td>9 2474</td><td>01/09/2015 15:20:27</td></tr></tbody></table>	CID	LATEST_ORDER	1 2431	15/04/1993 11:34:22	2 2001	03/08/1995 00:41:19	3 2107	01/02/1998 01:00:54	4 2327	08/10/2002 08:33:12	5 2714	31/01/2004 15:39:27	6 2060	20/12/2007 22:55:37	7 2736	22/08/2008 14:54:14	8 2897	02/12/2014 16:52:27	9 2474	01/09/2015 15:20:27
CID	LATEST_ORDER																																											
1 2704	21/01/2020 10:31:43																																											
2 2036	23/11/2021 03:43:37																																											
3 2449	10/06/2023 10:10:50																																											
4 2678	29/10/2024 17:31:20																																											
5 2481	05/02/2025 12:07:20																																											
6 2544	18/07/2028 11:12:20																																											
7 2445	13/04/2029 19:55:29																																											
8 2418	05/11/2031 05:29:02																																											
9 2525	02/12/2039 10:54:11																																											
10 2692	05/10/2042 10:49:52																																											
CID	LATEST_ORDER																																											
1 2431	15/04/1993 11:34:22																																											
2 2001	03/08/1995 00:41:19																																											
3 2107	01/02/1998 01:00:54																																											
4 2327	08/10/2002 08:33:12																																											
5 2714	31/01/2004 15:39:27																																											
6 2060	20/12/2007 22:55:37																																											
7 2736	22/08/2008 14:54:14																																											
8 2897	02/12/2014 16:52:27																																											
9 2474	01/09/2015 15:20:27																																											

פקודות 2.5 Rollback- Commit

Rollback 2.5.1

בשאילתת המצורפת מהקטי את כל העובדים שਮופיעים בview שמציג את העובדים עם דירוג נמוך.

```
--1
select wid from low_rating_workers;
delete from worker
where wid in (select wid from low_rating_workers);
--2
select wid from low_rating_workers;
rollback;
--3
select wid from low_rating_workers;
```

להלן התוצאות:

	3	2	1																												
	<table border="1"> <thead> <tr><th></th><th>WID</th></tr> </thead> <tbody> <tr><td>1</td><td>3289</td></tr> <tr><td>2</td><td>4035</td></tr> <tr><td>3</td><td>3362</td></tr> <tr><td>4</td><td>3719</td></tr> <tr><td>5</td><td>3878</td></tr> <tr><td>6</td><td>3033</td></tr> <tr><td>7</td><td>3183</td></tr> <tr><td>8</td><td>3519</td></tr> <tr><td>9</td><td>3132</td></tr> <tr><td>10</td><td>3393</td></tr> <tr><td>11</td><td>3802</td></tr> <tr><td>► 12</td><td>3901</td></tr> </tbody> </table>		WID	1	3289	2	4035	3	3362	4	3719	5	3878	6	3033	7	3183	8	3519	9	3132	10	3393	11	3802	► 12	3901		<table border="1"> <thead> <tr><th></th><th>WID</th></tr> </thead> </table>		WID
	WID																														
1	3289																														
2	4035																														
3	3362																														
4	3719																														
5	3878																														
6	3033																														
7	3183																														
8	3519																														
9	3132																														
10	3393																														
11	3802																														
► 12	3901																														
	WID																														

Commit 2.5.2

בשאילתת המצורפת מinitiy עובד להיות אחראי משמרות. לאחר פקודה commit ניסיתי rollback, אך ללא הצלחה:

	WID
► 1	3102

	WID
► 1	3102

```
insert into supervisor(wid) values(3102);
--1
select * from supervisor
where wid = 3102;
commit;
rollback;
--2
select * from supervisor
where wid = 3102;
```

הרשאות 2.6

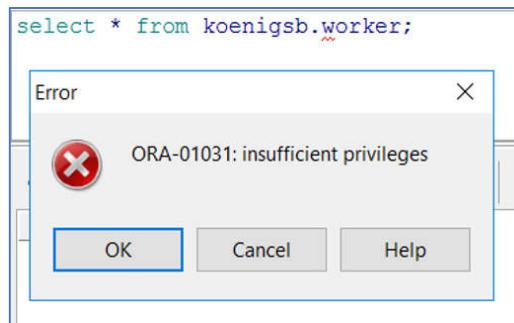
```
grant select, insert, update, delete on worker to tgoldshm;
revoke select, insert, update, delete on worker from tgoldshm;
```

:Grant נסיון של tgoldshm לוגשת לנתחים בטבלה worker, לאחר הרצת הפקודה 2.6.1

The screenshot shows the Oracle SQL Developer interface. The title bar says "select * from koenigsb.worker;". The tabs at the top are "SQL", "Output", and "Statistics". The SQL tab contains the query "select * from koenigsb.worker;". The Output tab displays a grid of 19 rows of data from the worker table. The columns are WID, SALARY, NAME, PHONENUMBER, RATING, BID, and WEEKLY_HOURS. The data includes various employee names like Natasha Shearer, Ving Aiken, Charles Young, etc., with their respective details. The bottom status bar shows "tgoldshm@labdbwin" and "19 rows selected in 0.049 seconds (more...)".

	WID	SALARY	NAME	PHONENUMBER	RATING	BID	WEEKLY_HOURS
▶	1	3508	39.00	Natasha Shearer	074900329	3	18
	2	3509	30.67	Ving Aiken	616434686	4	16
	3	3510	33.33	Charles Young	976055338	9	17
	4	3511	38.67	Chuck Tripplehorn	422130944	3	13
	5	3512	37.33	Steven LaPaglia	146588284	1	12
	6	3513	39.00	Helen Page	681243723	8	12
	7	3514	29.33	Andie Doucette	162802894	7	16
	8	3516	32.00	Gena Sampson	534727199	9	19
	9	3517	34.67	Stellan Meniketti	993057364	5	13
	10	3518	31.00	Nikka Chapman	042902933	7	13
	11	3519	28.00	Rip Tinsley	054723808	3	18
	12	3520	30.67	Mary Beth Bragg	021614157	6	14
	13	3521	33.33	Scarlett Diaz	281086878	3	13
	14	3522	36.00	Tony Pepper	157162329	9	19
	15	3523	31.00	Lisa Bacon	840694845	4	18
	16	3524	34.67	Allison Gaynor	722416730	6	19
	17	3525	36.00	Milvi Diarco	0964092772	5	13

:Revoke נסיון של tgoldshm לוגשת לטבלה worker, לאחר הרצת הפקודה 2.6.2



Checks אילוֹז 2.7

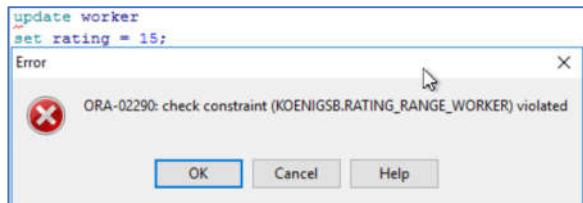
```
Alter table worker
add constraint rating_range_worker check (rating between 0 and 10);

Alter table pizza_order
add constraint rating_range_order check (rating between 0 and 10);

Alter table worker
add constraint weekly_hours_range check (weekly_hours between 16 and 52);

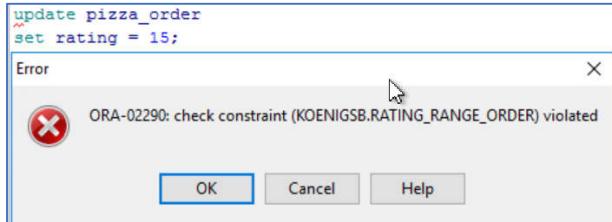
Alter table Order_Product
add constraint amount_check check (amount > 0);
```

.2.7.1 אילוֹז שמודא שדרוג עובד יהיה מספר בין 0 ל 10.



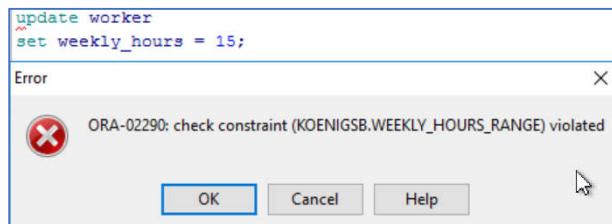
ניסוי הפרת האילוֹז:

.2.7.2 אילוֹז שמודא שדרוג הזמנה יהיה מספר בין 0 ל 10.



ניסוי הפרת האילוֹז:

.2.7.3 אילוֹז שמודא שעבוד לא יעבד פחות מ 16 שעות שבועית, וכן לא יעבד יותר מ 52 שעות שבועיות.



ניסוי הפרת האילוֹז:

.2.7.4 אילוֹז שמודא שלקשר בין מוצר לבין הזמנה לא יהיה כמות שווה ל 0.

Views 2.8

2.8.1 המבט הראשוני

הGBT הבא מציג מספר מזהה של כל סניף, יחד עם סכום הכנסות שלא מוזמינות.

GBT נועד לשימוש של המנהלים, עבור קבלת סטטיסטיות.

```
-- income by branches
create view sum_price_branch as
select bid, sum(total_price) as sumPrice
from pizza_order
group by bid;
```

2.8.1.1 השאלה הראשונה עבורGBT הראשוני

שאילתת שמציגה כתובת של סניף והכנסה בכתובות, עבור ניתוח גיאוגרפי של הכנסות.

	ADDRESS	SUMPRICE
1	59 McDowell Drive Rozenbu	9106
2	94 Ann Drive Pomona	9524
3	97 Tah Street Milan	9631
4	51 Thewlis Road Spresiano	9742
5	21st Street Amarillo	10184
6	9 Philip Street Pasadena	10296
7	775 Crouch Uden	10806
8	71 De Almeida Blvd Valenc	10966
9	33 McNeice Drive Giessen	13061
► 10	456 Denny Cedar Park	14177

2.8.1.2 השאלה השנייה עבורGBT הראשוני

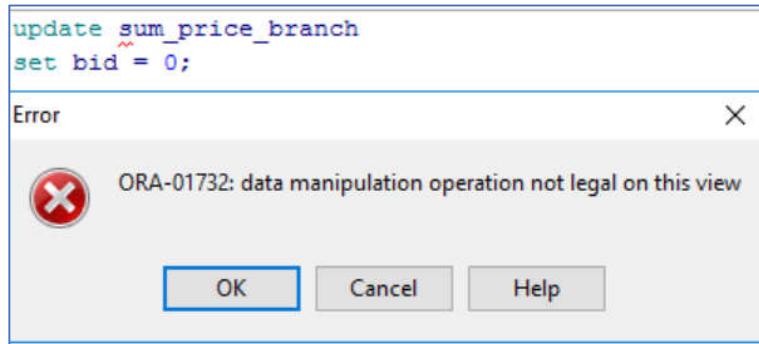
שאילתת שמציגה את מספר ושמות מנהלי המשמרות שעובדים בסניף שבו הכנסה הילדה מזומנים.

```
select wid, name
from worker natural join supervisor
where bid = (select bid
            from sum_price_branch
            where sumprice >= all(select sumprice from sum_price_branch));
```

	WID	NAME
► 1	3508	Natasha Shearer
2	3523	Lisa Bacon
3	3548	Olympia Suchet
4	3588	Stewart King
5	3723	Tara Addy
6	3773	Edie Eldard
7	3828	Sean Strathairn
8	3833	Aaron Tempest
9	3903	Leonardo Sedgwick
10	3983	Melanie Palmieri
11	3988	Chad Rea

2.8.1.3 עדכון המבט הראשוני

לא ניתן לעדכן את המבט, מפני שבגדרתו נעשה שימוש בcolumn aggregation.



2.8.2 המבט השני

המבט הבא מציג את העובדים המציגים. הקритריון לעובד מצטיין: הדירוג שלו מעל 8, והוא היה אחראי לכמויות הזמנות מעל הממוצע של הזמנות לעבוד. ייועד המבט הוא לשימוש המנהלים ורואי החשבון של הסניף, למטרות של נתינה מענקים ובונוסים.

```
-- Outstanding workers
create view outstanding_workers as
select wid, name, rating
from worker W
where rating > 8 and (select count(*) from pizza_order where wid=W.wid) >
(select avg(M.orders) from (select wid, count(*) as orders from pizza_order group by wid
having count(*)>2) M)
order by rating desc ,name;
```

2.8.2.1 השאלה הראשונה עבור המבט השני

השאילה מציגה את כל העובדים המציגים שהם גם אחראי משמרת:

	WID	NAME	RATING

2.8.2.2 השאלתה השנייה עבור המבט השני

השאלתה מציגה כמה עובדים מצטיננים יש בכל סניף

```
select bid, count(wid)
from branch natural left join (select * from worker
where wid in (select wid from outstanding_workers))
group by bid;
```

BID	COUNT(WID)
1	10
2	0
3	12
4	0
5	14
6	0
7	16
8	0
9	18
10	0
11	20

2.8.2.3 עדכון המבט השני

לאחר העדכון, המבט ריק, כי העובדים שהיו בו קיבלו דירוג נמוך שוחרג מתנאי ההשתיכות למבט.

```
update outstanding_workers
set rating = 2;
select * from outstanding_workers;
```

WID	NAME	RATING

2.8.2.4 הוספה למבט השני

כאן זה לא מתאפשר, מפני שבטבלה המקורית יש עמודות שתשארנה ריקות.

```
insert into outstanding_workers
^
values(10000, 'Avi Koenigsberg', 10);
```

Error

ORA-01400: cannot insert NULL into ("KOENIGSB"."WORKER"."SALARY")

OK Cancel Help

שלב 3

3.1 שאלות עם פרמטרים

3.1.1 השאלה הראשונה

שאילה שמטרתה להציג עובדים שעובדים בסניף נתון, בסדר שמודר על ידי המשתמש. הפרמטרים: מספר מזהה של סניף, עמודה למים, וסדר מיון.

```
select wid, name, phonenumbers, rating, salary, weekly_hours
from worker
where bid in (&<name= "choose branch number:" type = string
list ="select bid from branch" hint = "number between 11 - 20")
order by &<name= "sort by:"
list = "name, rating, weekly_hours" default = "wid"> &<name= "up"
list = "asc ,ascending order, desc , descending order" default = "asc"
description = "true" restricted ="true">;
```

תוצאה:

The screenshot shows a 'Variables' dialog box with three entries: 'choose branch number:' set to 13, 'sort by:' set to 'weekly_hours', and 'up' set to 'descending order'. Below the dialog is a table titled 'number between 11 - 20' containing five rows of worker data. The first row (WID 4084) is highlighted in red.

	WID	NAME	PHONENUMBER	RATING	SALARY	WEEKLY_HOURS
▶	1	4084 Ritchie Balin	643076050	9	40.00	52
	2	3407 Cledus Alda	016155955	2	40.00	52
	3	3342 Mary Beth Prinze	407927930	5	40.00	52
	4	3953 Leelee Coughlan	350963796	6	39.67	51
	5	3417 Kimberly Sampson	055000275	8	30.67	51

The screenshot shows a second 'Variables' dialog box with three entries: 'choose branch number:' set to 15, 'sort by:' set to 'name', and 'up' set to 'ascending order'. Below the dialog is a table titled 'number between 11 - 20' containing five rows of worker data. The first row (WID 3073) is highlighted in red.

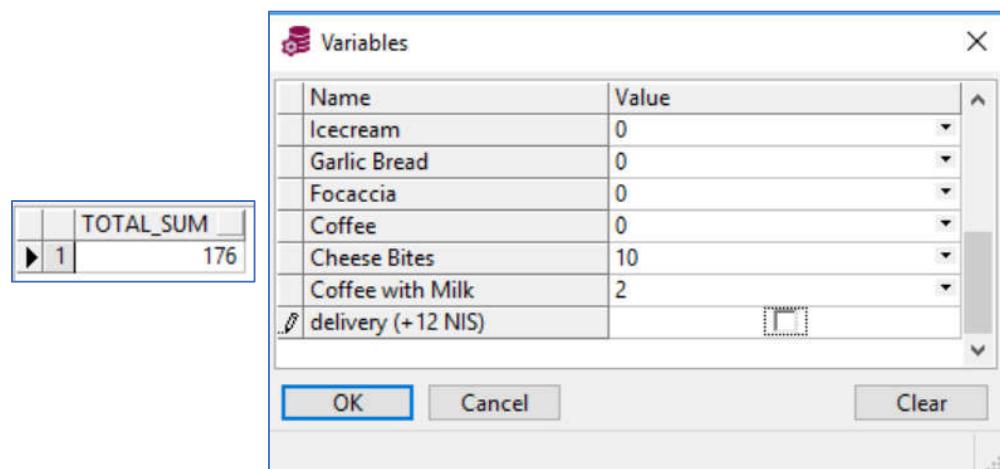
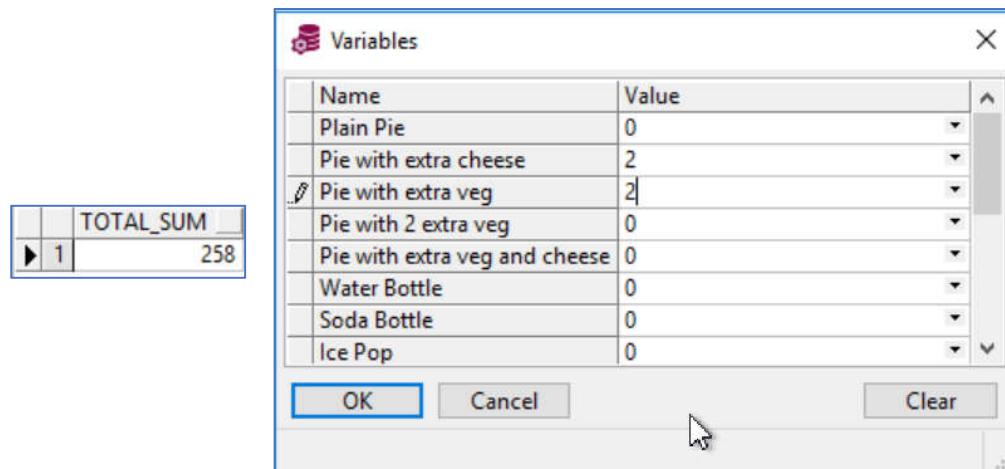
	WID	NAME	PHONENUMBER	RATING	SALARY	WEEKLY_HOURS
▶	1	3073 Adam Cale	610678856	8	36.67	42
	2	3168 Albert Tah	562914326	4	29.67	21
	3	3255 Alfred Crystal	016921293	8	37.00	43
	4	3422 Ali Cagle	794914192	10	31.00	25
	5	3186 Ann Rosas	668452122	8	30.33	23

השאילתה ה-3.1.2

שאילתה שמטרתה להציג למשתמש את התפריט, לחת לו לבחור כמה מכל מוצר, ולהציג בסוף את מחיר ההזמנה (השאילתה ארכוה, וחלוקת התוכנה). הפקטרים: עבור על מוצר, צריך לבחור את הכמות.

```
select sum(sum_1) as total_sum from
(select sum(price * &<name= "Plain Pie" type = Integer default = 0 hint = "What do you want to order?")>
from product
where pid = 1
union
select sum(price * &<name= "Pie with extra cheese" type = Integer default = 0>
from product
where pid = 2
union
select sum(price * &<name= "Pie with extra veg" type = Integer default = 0>) as sum_1
from product
where pid = 3
union
select sum(price * &<name= "Pie with 2 extra veg" type = Integer default = 0>)
from product
where pid = 4
union
```

התוצאה



3.1.3 השאלתה השלישית

שאילתה שמציגה את כל ההזמנות עבור לקוח שהמשתמש בוחר. הפרמטר הוא שם הלוקות.

```
select * from pizza_order
where cid in(select cid from customer where name = (&<name= "select:" type = string
list ="select name from customer order by name" hint = "select customer">))
order by cid, order_date desc;
```

התוצאה

Variables	
Name	Value
▶ select:	Allison Douglas

	ORDER_DATE	OID	IS_DELIVERED	RATING	TOTAL_PRICE	CID	BID	WID
▶ 1	11/07/2012 21:50:15	5716		1	7	28.00	2083	11
2	25/08/2012 10:06:31	5489		1	2	65.00	2083	10

Variables	
Name	Value
▶ select:	Barry Kenoly

	ORDER_DATE	OID	IS_DELIVERED	RATING	TOTAL_PRICE	CID	BID	WID
▶ 1	26/10/2012 22:35:54	5425		1	9	20.00	2174	15

3.1.4 השאלתה הרביעית

שאילתה שמציגה את כל ההזמנות בין שני תאריכים שהמשתמש בוחר, לפי סדר שהמשתמש בוחר. הפרמטרים: תאריך תחילה, תאריך סוף, וסדר מיון.

```
select * from pizza_order
where order_date between to_date(&From, 'dd/mm/yyyy')
and to_date(&To, 'dd/mm/yyyy')
order by order_date &<name= "Descending sort order?" checkbox = "desc, asc">;
```

התוצאה:

Variables

Name	Value
From	'01/01/2000'
To	'01/01/2020'
Descending sort order?	<input checked="" type="checkbox"/>

	ORDER_DATE	OID	IS_DELIVERED	RATING	TOTAL_PRICE	CID	BID	WID
► 1	13/11/2019 08:31:03	5674	1	7	49.00	2434	14	3899
2	20/08/2018 12:46:04	5604	1	7	9.00	2432	12	3873
3	16/08/2017 16:58:04	5839	1	6	60.00	2713	15	4112
4	13/08/2015 02:13:03	5293	1	3	8.00	2667	15	3148
5	18/01/2015 00:53:16	6507	1	1	16.00	2476	14	3170

3.2 דוחות

3.2.1 הדוח הראשון

הדווח שמציג ללקוח את כל ההזמנות שלו. הדוח מראה את תאריך ההזמנה, את דירוג ההזמנה, ואת מחיר ההזמנה.
3.1.3 מספרי ויהוי של ההזמנה, הלקוח, העובד האחראי, והסניף, אינם חשובים ללקוח. הדוח מבוסס על שאלתה מה

The screenshot shows a report configuration interface with two main sections:

- Fields Configuration:** A table where each row represents a field (Item) with columns for Description, Style, Header, Align, Format, Break, and Sum. Fields listed include ORDER_DATE, OID, IS_DELIVERED, RATING, TOTAL_PRICE, CID, BID, and WID. Most fields have 'Center' alignment and 'Default Field' style.
- Report Preview:** A table titled "All Orders of Customer" showing four rows of data. The columns are Order Date, Rating, and Total Price. The data is as follows:

Order Date	Rating	Total Price
04/09/2019 16:24:31	4	31.00
28/07/2019 10:11:42	8	40.00
23/07/2019 02:18:31	6	16.00
07/07/2019 20:52:22	7	204.00

3.2.2 הדוח השני

השאילתה הבאה מציגה את כל העובדים בדירוג גדול או שווה לנתחן, עם שעות שבועיות יותר מהנתון

```
select * from worker
where rating >= &rating
and weekly_hours >= &weekly_hours
order by rating;
```

להלן הדוח שבוסס על השאילתה זו. מטרת הדוח להציג למנג'ל המידע של עובדים על פי קритריוני הצעינות מסוימים.

The screenshot shows a report configuration interface with two main sections:

- Fields Configuration:** A table where each row represents a field (Item) with columns for Description, Style, Header, Align, Format, Break, and Sum. Fields listed include Report Title, Variables, Tabular Tables, Form Tables, and Default Field. The Report Title has a 'Blue Title' style.
- Report Preview:** A table titled "Workers with top rating and most hours" showing seven rows of data. The columns are Wid, Salary, Name, Phononenumber, Rating, Bid, and Weekly Hours. The data is as follows:

Wid	Salary	Name	Phonenumber	Rating	Bid	Weekly Hours
3944	39.67	Bridgette Hatchet	371838326	7	19	51
3600	42.67	Geena Chandler	289961788	7	19	51
3497	39.33	Juliana Bello	912064582	7	17	50
3919	39.33	Jane Keen	163279303	7	10	50
4099	39.33	Carolyn Prowse	434894436	7	17	50
3699	40.00	Maria Ferrell	226700710	7	15	52
3698	40.00	Joshua Salonga	968815502	7	15	52

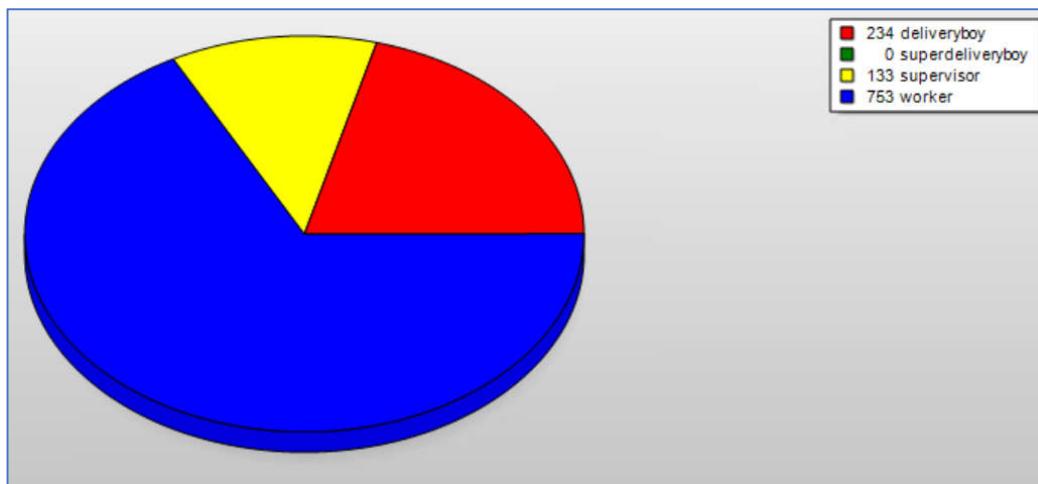
3.3 גראפים

3.3.1 הגרף הראשון

לפנינו שאלתה שמציגה את מספר העובדים מכל סוג.

```
select 'worker', count(*)  
from (select wid from worker  
minus  
select wid from supervisor  
minus  
select wid from deliveryboy)  
union  
select 'supervisor', count(*)  
from (select wid from supervisor  
minus select wid from deliveryboy)  
union  
select 'deliveryboy', count(*)  
from (select wid from deliveryboy  
minus select wid from supervisor)  
union  
select 'superdeliveryboy', count(*)  
from (select wid from deliveryboy  
intersect select wid from supervisor);
```

להלן הגרף המתאים. מטרתה להציג לנו כמה עובדים יש לנו מכל סוג.



הגרף השני 3.3.2

2.1.6 הגרף מציג את ממוצע מחיר הזמנה לפי חודש בשנה האחרונות. הגרף מבוסס על שאלתא

```
select month, round(avg(total_price),2) as avg_order_price
from (select to_char(order_date, 'MM') as month, total_price
      from pizza_order
     where order_date >=
           add_months(sysdate, -12*&years_to_calculate) month_price
   group by month
  order by month;
```



פונקציות 3.4

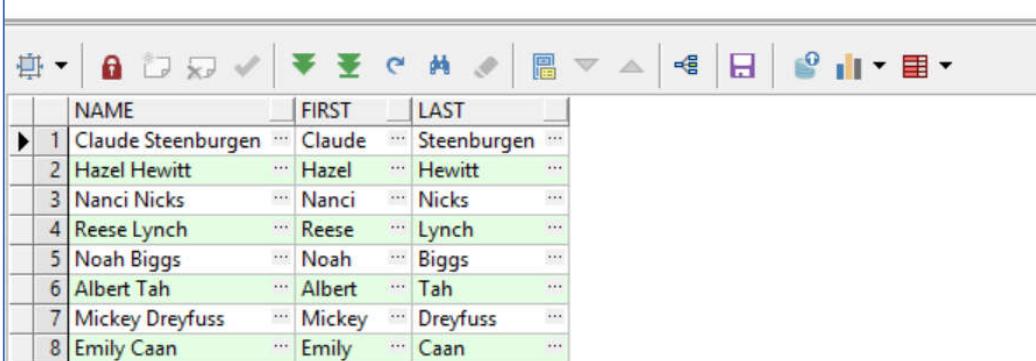
3.4.1 פונקציות לחילוץ שם פרטי ומשפחה מתוך שם מלא. הפונקציה משתמשת בביטויים רגולריים.

```
create or replace function Get_First_Name(Name in varchar2)
return varchar2 is
    FunctionResult varchar2(1000);
begin
    FunctionResult:= regexp_substr(Name, '\w*');
    return(FunctionResult);
end Get_First_Name;
```

```
create or replace function Get_Last_Name(Name in varchar2)
return varchar2 is
    Last_Name varchar2(1000);
    First_Name varchar(1000);
begin
    First_Name := Get_First_Name(Name);
    Last_Name:= regexp_replace(Name, First_Name, '');
    Last_Name:= regexp_replace(Last_Name, ' ', '');
    return(Last_Name);
end Get_Last_Name;
```

דוגמא לשימוש:

```
select name, get_first_name(name) as first, get_last_name(name) as last
from worker;
```



The screenshot shows the Oracle SQL Developer interface with a query editor and a results grid. The query editor contains the SQL statement: 'select name, get_first_name(name) as first, get_last_name(name) as last from worker;'. The results grid displays eight rows of data, each showing a name and its corresponding first and last names extracted by the functions.

	NAME	FIRST	LAST
1	Claude Steenburgen	Claude	Steenburgen
2	Hazel Hewitt	Hazel	Hewitt
3	Nanci Nicks	Nanci	Nicks
4	Reese Lynch	Reese	Lynch
5	Noah Biggs	Noah	Biggs
6	Albert Tah	Albert	Tah
7	Mickey Dreyfuss	Mickey	Dreyfuss
8	Emily Caan	Emily	Caan

3.4.2 פונקציה למציאת לקוחות אהובים על הרשת.

הfonקציה מקבלת מספר שלם ובודקת על כל לקוחות, אם בין כל שתי הזמינות רצופות שלו עברו יותר ימים מהמספר הנתון. אם לא, הוא מוחזר. הfonקציה משתמשת בלולאה והcursor.

```
create or replace type customer_record as object
(
    name varchar(30),
    phone varchar(10)
);
```

```
create or replace type customer_table as table of customer_record;
```

בשביל הfonקציה יצרנו שני type:

הראשון ישות שמייצגת שורה בטבלה שחוורת מהfonקציה, והשני טבלה.

הfonקציה:

```
1  create or replace function Get_Favorite_Customers(Interval in number) return customer_table is
2  begin
3      return_table customer_table;
4      cursor customers is --cursor for finding relevant customers
5          select cid, name, telephone, order_date
6              from customer natural join pizza_order
7                  order by cid, order_date;
8      customer_rec customers%rowtype;
9      is_favorite number; --variables in use for function
10     current_cid number;
11     current_date date;
12     begin
13         if not customers%isopen then
14             open customers;
15         end if;
16         return_table := customer_table();
17         fetch customers into customer_rec; --extract first row data
18         current_cid:= customer_rec.cid;
19         current_date:= customer_rec.order_date;
20         is_favorite:=1;
21         loop --for each customer, check if every 2 consecutive orders are within <Interval> days of each other
22             fetch customers into customer_rec;
23             exit when customers%notfound;
24             if customer_rec.cid = current_cid
25                 then
26                 if customer_rec.order_date - current_date > Interval
27                     then
28                         is_favorite:=0;
29                     end if;
30                 else
31                     if is_favorite = 1
32                         then
33                             return_table.extend(); --add the customer to the output table
34                             return_table(return_table.count):= customer_record(customer_rec.name, customer_rec.telephone);
35                         end if;
36                     current_cid:= customer_rec.cid;
37                 end if;
38             current_date:=customer_rec.order_date;
39         end loop;
40     return (return_table);
41 end Get_Favorite_Customers;
```

פעולות הפונקציה: ה`cursor` בשמם `customers` מוחזיק טבלה של הזמנות במילוי לפי ת"ז ללקוח ותאריך הזמנה. בהינתן טווח של ימים נקלט לפונקציה, מטבחצעת בדיקה אם לא עברו יותר מאותו מספר ימים בין הזמנה להזמנה. ההגדרה זו אומרת לנו אם לקוחות הוא ללקוח אהוב או לא. אותו לקוחות נוסף לטבלה שהולכת ונובנית בשם `return_table`, ואותה טבלה מוחזרת בסוף פעולה הפונקציה.

```
select *  
from table(get_favorite_customers(365));
```

The screenshot shows the Oracle SQL Developer interface. On the left, there is a code editor window containing the SQL query: `select * from table(get_favorite_customers(365));`. To the right of the code editor is a results grid window displaying the output of the query. The results grid has two columns: `NAME` and `PHONE`. The data consists of 14 rows, each containing a number from 1 to 14 followed by a name and a phone number. The names listed are: Miko Fiorentino, George Barnett, Kurt Hackman, Toshiro Olin, Gary Whitaker, Busta Martin, Peabo Peterson, Teena Macy, Brian Brothers, Petula Penn, Seth Hanks, King Finn, Rickie Applegate, and Ralph Palminteri. The phone numbers are: 850194280, 885472725, 518001511, 927677494, 422518453, 843010327, 802835574, 913480563, 253102582, 158997205, 801395569, 368680611, 273519088, and 194798446.

	NAME	PHONE
1	Miko Fiorentino	850194280
2	George Barnett	885472725
3	Kurt Hackman	518001511
4	Toshiro Olin	927677494
5	Gary Whitaker	422518453
6	Busta Martin	843010327
7	Peabo Peterson	802835574
8	Teena Macy	913480563
9	Brian Brothers	253102582
10	Petula Penn	158997205
11	Seth Hanks	801395569
12	King Finn	368680611
13	Rickie Applegate	273519088
14	Ralph Palminteri	194798446

משמאלי ניתן לראות תוצאה של שימוש בפונקציה.
השאילה מהציגת את הלוקוחות שאין בין הזמנה
להזמנה שלהם יותר משנה.

3.5 פרוצדורה

3.5.1 פרווצדורה שמקבלת כקלט מספר סניף, ומדפיסה דוח על הסניף

```
1  create or replace procedure Print_Status(Branch_id in Number) is
2
3      --variables for data extraction
4      num_of_workers number;
5      weekly_salary number;
6      num_of_supervisors number;
7      num_of_deliverors number;
8      num_of_orders number;
9      total_income number;
10     num_of_deliveries number;
11     b_address varchar2(50);
12     date_report date := sysdate;
13
14     --cursors for multiple data extraction
15     cursor promotables is
16         select wid, name, rating, weekly_hours
17             from worker
18             where bid = Branch_id
19             and rating >=8
20             and wid not in (select wid from supervisor)
21             order by dbms_random.value()
22             fetch first 3 rows only;
23
24     cursor firables is
25         select wid, name, rating, weekly_hours
26             from worker
27             where bid = Branch_id
28             and rating <=3
29             order by dbms_random.value()
30             fetch first 3 rows only;
31
32     begin
33         --local queries to get data
34         select count(*) into num_of_workers
35             from worker where bid = Branch_id;
36         select count(*) into num_of_supervisors
37             from worker natural join supervisor
38             where bid = Branch_id;
39         select count(*) into num_of_deliverors
40             from worker natural join deliveryboy
41             where bid = Branch_id;
42         select count(*) into num_of_orders
43             from pizza_order
44             where bid = Branch_id;
45         select count(*) into num_of_deliveries
46             from pizza_order natural join delivery_order
47             where bid = Branch_id;
48         select sum(total_price) into total_income
49             from pizza_order
50             where bid = Branch_id;
```

```

51   select sum(salary*weekly_hours) into weekly_salary
52   from worker where bid = Branch_id;
53   select address into b_address
54   from branch where bid = Branch_id;
55
56   --printing commands
57   dbms_output.put_line('===== Branch Number '
58   || Branch_id || ' Status Report =====');
59   dbms_output.put_line('Date of report : ' || date_report);
60   dbms_output.put_line('Address: '|| b_address);
61   dbms_output.put_line('Number of Workers: '||num_of_workers);
62   dbms_output.put_line('Number of Supervisors: '||num_of_supervisors);
63   dbms_output.put_line('Number of Delivery Boys: '|| num_of_deliverors);
64   dbms_output.put_line('Number of Orders: '|| num_of_orders);
65   dbms_output.put_line('Number of Deliveries: '|| num_of_deliveries);
66   dbms_output.put_line('Total Income: ' || total_income);
67   dbms_output.put_line('Total Weekly Salaries: ' || weekly_salary);
68   dbms_output.put_line('');
69
70   --loop over promotable workers
71   dbms_output.put_line('Suggested Promotions: ');
72   for worker in promotables loop
73     dbms_output.put_line(rpad(worker.name||',', 25)
74     || 'wid:'||lpad(worker.wid,5)
75     ||', rating:'
76     ||lpad(worker.rating,2)
77     ||', weekly hours:'
78     ||lpad(worker.weekly_hours,3));
79   end loop;
80   dbms_output.put_line('');
81
82   --loop over fireable workers
83   dbms_output.put_line('Suggested Layoffs: ');
84   for worker in firables loop
85     dbms_output.put_line(rpad(worker.name||',', 25)
86     || 'wid:'||lpad(worker.wid,5)
87     ||', rating:'
88     ||lpad(worker.rating,2)
89     ||', weekly hours:'
90     ||lpad(worker.weekly_hours,3));
91   end loop;
92
93   end Print_Status;

```

פעולה הפרוצדורה:

```

===== Branch Number 11 Status Report =====
Number of Workers: 112
Number of Supervisors: 16
Number of Delivery Boys: 31
Number of Orders: 177
Number of Deliveries: 0
Total Income: 13633
Total Weekly Salaries: 131279.15

Suggested Promotions:
King Schwarzenegger,      wid: 3185, rating:10, weekly hours: 33
Cate Guinness,            wid: 3305, rating:10, weekly hours: 37
Deborah Swank,             wid: 3225, rating:10, weekly hours: 22

Suggested Layoffs:
Brent McCracken,          wid: 3247, rating: 1, weekly hours: 29
Kirk Irving,               wid: 3500, rating: 1, weekly hours: 42
Ed Mirren,                wid: 3355, rating: 2, weekly hours: 43

```

לפונקציה יש שני סטנים מורכבים בשמות fireables ו promotables – כל אחד מצביע על טבלה שמכילה שלושה עובדים שמתאימים לקידום או לפיטורין בהתאם.

הפרוצדורה משתמשת בשאלות מקומיות בשבייל להלץ ערכיהם מתחזק טבלאות עברו: מספר עובדים, מספר אחראי משמרות, מספר שליחים, מספר הזמינות, מספר משלוחים, סך הכנסה, סך הוצאה על משכורות שבועיות.

הפרוצדורה מדפסת למסך את כל המידע המזוכר לעיל, ואת המידע על העובדים שמתאימים לקידום או לפיטורין.

דוגמא לשימוש משmaal:

3.5.2 פ្នצדורה שמדמה שליחת סמסים ללקוחות רגילים, עם מבצעים מיוחדים

3.4.2 הפרוצדורה מקבלת מהרוזת סמס וטבלה מסווג customer_table כפי שהוזג ב.

```
create or replace procedure Send_Texts(text_to_send in varchar,
c_table in customer_table) is
begin
    dbms_output.put_line('Sending Text...');
    dbms_output.put_line(text_to_send);

    for i in 1..c_table.count
    loop
        dbms_output.put_line('To: '|| rpad(c_table(i).name, 25)
                            || 'Number: '|| lpad(c_table(i).phone, 11));
    end loop;
end Send_Texts;
```

: שימוש:

```
declare message varchar(100) := 'To our favorite customers!
Only this week, you get a 30% discount on any order!
Pizza Corona';
begin
    send_texts(message,get_favorite_customers(1000));
end;
```

```
Sending Text...
To our favorite customers!
Only this week, you get a 30% discount on any order!
Pizza Corona
To: Miko Fiorentino      Number:  850194280
To: George Barnett       Number:  885472725
To: Kurt Hackman         Number:  518001511
To: Toshiro Olin          Number:  927677494
To: Gary Whitaker        Number:  422518453
To: Ruesta Martin         Number:  843010327
```

3.5.3 פרוצדורה שמקדמת עובד לתפקיד מנהל משמרת.

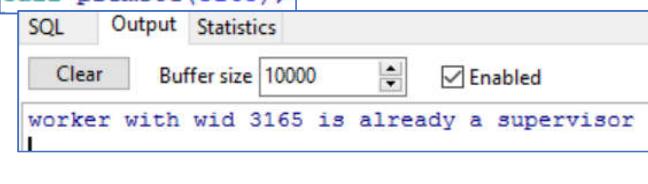
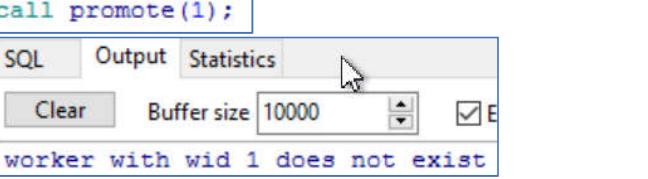
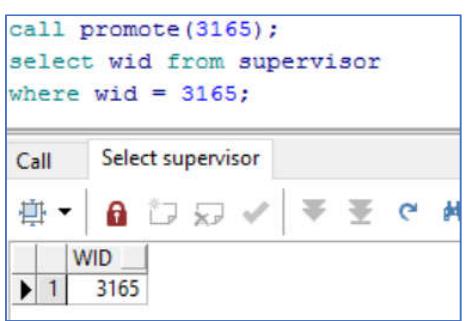
```

create or replace procedure Promote(in_Wid in number) is
    pk_constraint_e exception;
    local_wid number;
begin
    select count(*) into local_wid from worker where wid = in_Wid;
    if local_wid = 0
        then raise pk_constraint_e;
    end if;
    insert into supervisor
    values(in_Wid);
    exception
        when dup_val_on_index
            then dbms_output.put_line('worker with wid '||in_Wid
            ||' is already a supervisor');
        when pk_constraint_e
            then dbms_output.put_line('worker with wid '||in_Wid
            ||' does not exist');
end Promote;

```

בפרוצדורה הוי הגדרנו שגיאה e שמייעצת לטפל במצב שמנסים לקדם עובד שלא קיים.

דוגמאות לשימוש:

שימוש לא תקין	שימוש תקין
<pre>call promote(3165);</pre>  <pre>call promote(1);</pre> 	<pre>call promote(3165); select wid from supervisor where wid = 3165;</pre> 

טריגרים

3.6.1 טריגר שנותן ה

עלאה בשכר לכל עובד שמקבל קידום למנהל משמרת

```
create or replace trigger Promotion
  after insert
  on supervisor
  for each row
declare
  -- local variables here
begin
  update worker
  set salary = salary + 3
  where wid = :new.wid;
end Promotion;
```

דוגמא לשימוש (באמצעות פרוצדורת הקידום):

```
--1
select wid, salary from worker where wid = 3177;
--2
call promote(3177);
--3
select wid, salary from worker where wid = 3177;
```

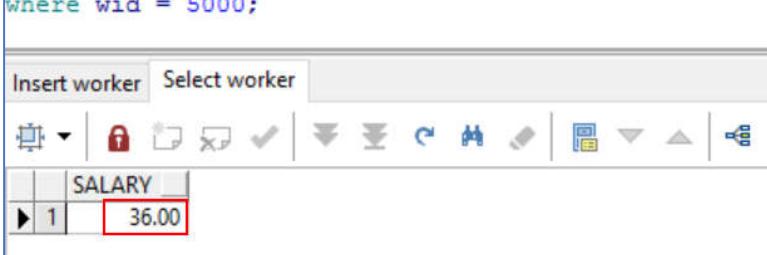
3			1				
	WID	SALARY		WID	SALARY		
▶	1	3177	36.67	▶	1	3177	33.67

3.6.2 טרייגר שמעדכן את המשכורות של עובד חדש

```
create or replace trigger Set_Salary
  before insert
  on worker
  for each row
declare
  -- local variables here
begin
  :new.salary:= 28 + (:new.weekly_hours-16)/3;
end Set_Salary;
```

דוגמא לשימוש:

```
insert into worker
values(5000, 5, 'Avi K', '0524591819', 3, 11, 40);
select salary from worker
where wid = 5000;
```

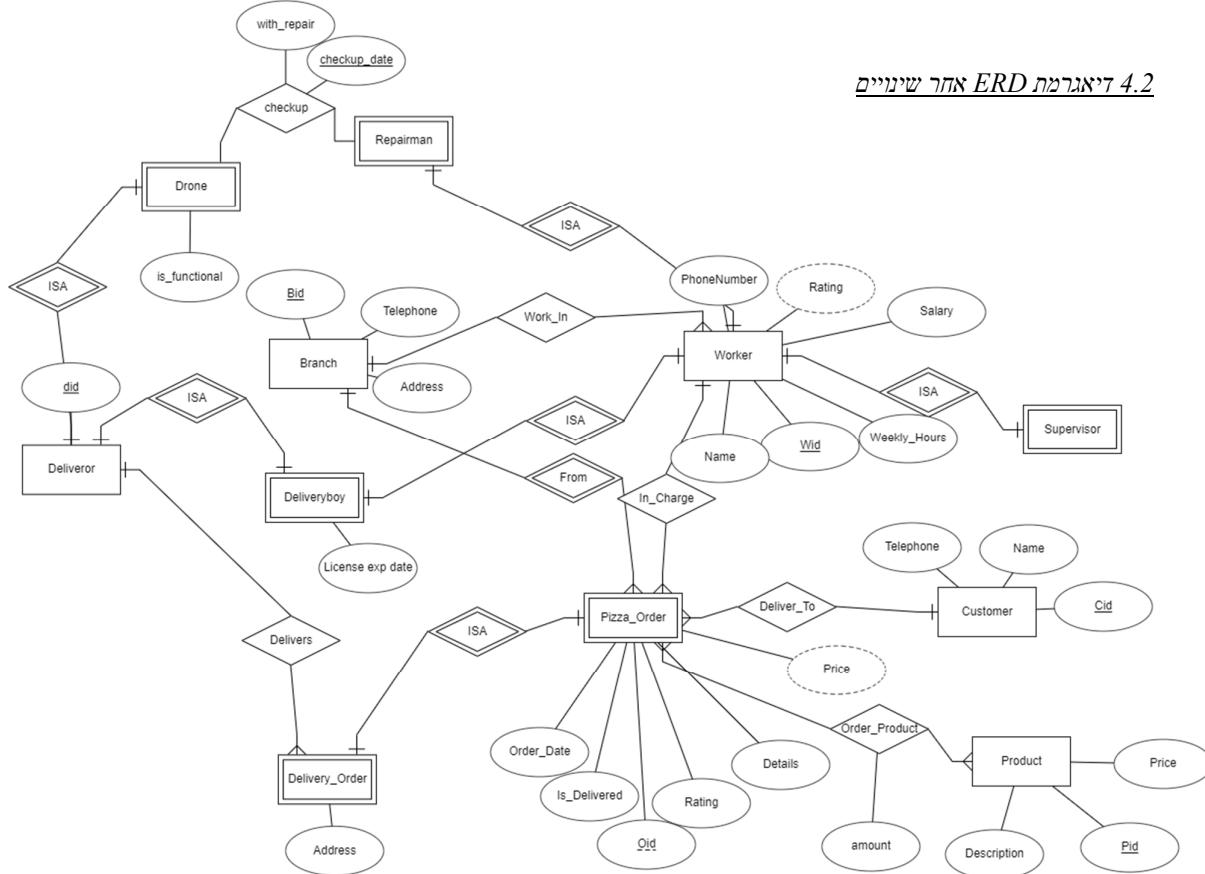


	SALARY
▶ 1	36.00

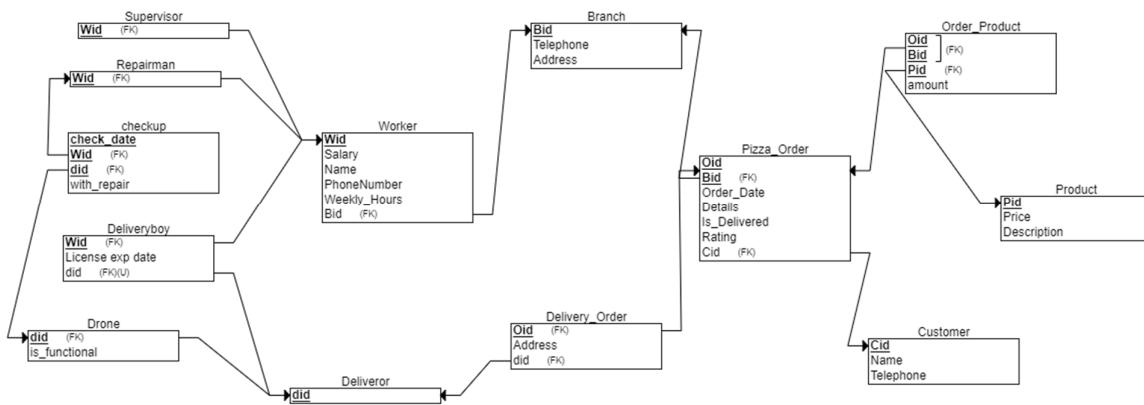
שלב 4

4.1 תקציר היחסים במערכת

הרשות קיבלה מספר רחפנים לצורך חסכוּן בכוח אדם למשלוחים. מדובר בתקופת נסיוּן, ולכן קיימים עדין שליחים אנושיים בראשת. לתחזוקה של הרחפנים, הרשות מעסיקה טכנאים לבדוקות שגרתיות ולתיקונים בעת הצורך.



4.2 דיאגרמת אחר שינויים



4.3.1 תיאור הישויות חדשות

טכני (Repairman) – יורש את כל התוכנות של עובד באמצעות מפתח זר: ת.ז.
מוסר (Deliveror) – מתאפיין ב: ת.ז (מפתח ראשי).

רחפן (Drone) – מתאפיין ב: האם_תיקון, יורש את כל התוכנות של מוסר באמצעות מפתח זר: ת.ז. הרחפן לא שירך לטעמי מסויים, בגלל

בדיקות_רחפן (Checkup) – מתאפיין ב: ת.ז טכני (מפתח זר), ת.ז רחפן (מפתח זר), תאrik בדיקה (מפתח ראשי, יחד עם ת.ז טכני ות.ז רחפן), האם_נדרש_תיקון.

4.3.2 תיאור שינויים לשינויות הקיימות

4.3.2.1 שליח: לפני השינוי, יורש את כל התוכנות של עובד באמצעות מפתח זר: ת.ז. אחרי השינוי, יורש גם את כל התוכנות של מוסר באמצעות מפתח זר: ת.ז.

4.3.2.2 הזמנה במשלוח – לפני השינוי, היה קשור עם שליח על ידי מפתח זר: ת.ז עובד. אחרי השינוי, במקום ת.ז עובד, קיים מפתח זר 'ת.ז מוסר' שמאפשר קישור לשיליח או לרחפן.

4.3.3 תיאור קשרים

- כל טכני מקשר לעובד אחד (ירושה). עובד מקשר לכל היוטר לטכני אחד.
- כל שליח מקשר למוסר אחד (ירושה). מוסר מקשר לכל היוטר לשיליח אחד.
- כל רחפן מקשר למושר אחד (ירושה). מושר מקשר לכל היוטר לרחפן אחד.
- כל משלוח מקשר למושר אחד (ייתכנו הרבה משלוחים מושר).
- כל בדיקת_רחפן מקשרת לרחפן אחד ולטכני אחד. ייתכנו כמה בדיקות_רחפן לכל זוג טכני-רחפן,
- תוק הגבלה של התאrik.

פקודות SQL עבור השינויים

4.4.1 יצרת טבלאות

```

CREATE TABLE Deliveror
(
    did INT NOT NULL,
    PRIMARY KEY (did)
);

CREATE TABLE Drone
(
    is_functional INT NOT NULL,
    did INT NOT NULL,
    PRIMARY KEY (did),
    FOREIGN KEY (did) REFERENCES Deliveror(did)
);

```

```

CREATE TABLE Repairman
(
    Wid INT NOT NULL,
    PRIMARY KEY (Wid),
    FOREIGN KEY (Wid) REFERENCES Worker(Wid)
);

CREATE TABLE checkup
(
    check_date DATE NOT NULL,
    with_repair INT NOT NULL,
    Wid INT NOT NULL,
    did INT NOT NULL,
    PRIMARY KEY (check_date, Wid, did),
    FOREIGN KEY (Wid) REFERENCES Repairman(Wid),
    FOREIGN KEY (did) REFERENCES Drone(did)
);

```

4.4.2 עדכונים לטבלאות

```

--1
alter table Deliveryboy
add did int;
--2
update Deliveryboy
set did = rownum;
commit;
--3
alter table Deliveryboy
add constraint db_fk_did foreign key(did) references deliveror(did);
--4
alter table Deliveryboy
add constraint db_unik_did unique(did);

```

- .1 הוספה עמודה did לטבלה שליה.
- .2 עדכון ערכי העמודה .did.
- .2.1 בשלב זהה הוננו נתונים לטבלה deliveror
- .3 הוספה אילוץ מפתח זר על העמודה did.
- .4 הוספה אילוץ unique על העמודה .did

```

--1
alter table delivery_order
add did int;

--2
update delivery_order do
set did = (select did from delivery_boy db where db.wid = do.wid);

--3
alter table delivery_order
drop column wid;

--4
alter table delivery_order
add constraint do_fk_did foreign key(did) references deliveror(did);

```

- .1. הוספה עמודה did לטבלה משלו.
- .2. עדכן העמודה did כך שהיא שווה לתז. מוסר של השליח.
- .3. מחייבת העמודה .wid.
- .4. הוספה אילוץ מפתח זר על העמודה did.

.3NF בדיקת 4.4.3

מוסר: אין תלויות. התנאים מתקיימים באופן ריק.

רchapn: התלויות הן did→is_functional ולקמן הטבלה מנורמלת.

טכני: אין תלויות. התנאים מתקיימים באופן ריק.

בדיקה רchapn: התלויות הן .wid, did, check_date → with_repair

Select 4.5 שאלות

4.5.1 השאלתה הראשונה

שאילתה שמצויה שם פרטן, שם משפחה, ותז. של כל טכני שבייע לפחות מספר מסויים של בדיקות, בין התאריכים ווּן עַל פִי הבדיקה. מטרת השאלה: המנהלים יכולים לראות את רמת הפעולות של הטכנאים בתקופה נתונה, ואיזה מהם פעילים יותר.

```
with rm_names as( select get_first_name(name) as "first name",
get_last_name(name) as "last name",
wid
from repairman rm natural join worker)
select * from rm_names rm
where &"Number of checkups" <= (select count(*)
from checkup cp
where cp.wid = rm.wid
and check_date between
to_date(&from, 'dd/mm/yyyy')
and to_date(&to, 'dd/mm/yyyy'));
```

Variables

Name	Value
▶ Number of checkups	3
from	'01/01/2020'
to	'01/03/2020'

	first name	last name	WID
▶ 1	Allan	Day	5002
2	Leonardo	Elwes	5012
3	Joaquin	Hagerty	5014

השאילתה השנייה:

4.5.2.1 פונקציה שמקבלת שני תאריכים ובודקת אם החודש והשנה שליהם תואימים.

הפונקציה משווה את מספר החודש ומספר השנה בשני התאריכים ומהזירה אמת במידה שיש התאמה.

```
create or replace function is_same_month(date1 in date,
                                         date2 in date)
                                         return number is
begin
  if extract(month from date1) = extract(month from date2)
    and extract(year from date1) = extract(year from date2)
  then
    return 1;
  end if;
  return 0;
end is_same_month;
```

4.5.2.2 פונקציה שמקבלת שני תאריכים ומספר ת.ז של רחפן, ובודקת האם בכל חודש בין שני התאריכים עבר הרחפן בדיקה (תוק שימוש בפונקציה (4.5.2.1).

```
create or replace function Routine_checks(Did1 in number,
                                         from_date in date,
                                         to_date in date)
                                         return number is
cursor months
is select add_months(from_date, rownum-1) m1
from all_objects_ae
where add_months(from_date, rownum-1)
between from_date
and to_date;
num_of_checks number;

begin
  for m in months
  loop
    select count(*) into num_of_checks
    from checkup
    where did = Did1
    and 1 = is_same_month(check_date, m.m1);
    if num_of_checks = 0 then
      return 0;
    end if;
  end loop;
  return 1;
end Routine_checks;
```

הפונקציה משתמשת בפונקציה שהגדכנו בסעיף הקודם

4.5.2.2 שאלתה שימושת בפונקציה

השאילה מבקשת מהמשתמש שני תאריכים, ומציגה את התוצאות של כל הרחפנים שעברו בדיקה לפחות פעם אחת במהלך החודש בין שני התאריכים הנ"ל. ניתן גם לשנות את ה-1 ל-0 ולקבל את כל הרחפנים שלא עברו בדיקה לפחות פעם אחת במהלך החודש). מטרת השאלה היא לקבל תמונה מצב של תוצאות הבדיקות של הטכנים.

```
select * from drone
where 1 = routine_checks(did,
to_date(&"Starting Date",'mm/yyyy'),
to_date(&"Ending Date",'mm/yyyy'))
order by did;
```

Name	Value
Starting Date	'01/2020'
Ending Date	'03/2020'

תוצאה:

	IS_FUNCTIONAL	DID
▶	1	258

הוכחת נכונות:

```
select did, check_date from checkup
where did = 258
order by check_date;
```

	DID	CHECK_DATE
1	258	21/01/2020
2	258	10/02/2020
▶ 3	258	05/03/2020

Views 4.6

4.5.1 מבט המבוסס על שאלה 4.6.1

```
create view active_repairmen1 as
with rm_names as( select get_first_name(name) as "first name",
get_last_name(name) as "last name",
wid
from repairman rm natural join worker)
select * from rm_names rm
where &"Number of checkups" <= (select count(*)
from checkup cp
where cp.wid = rm.wid
and check_date between
to_date(&from, 'dd/mm/yyyy')
and to_date(&to, 'dd/mm/yyyy'));
```

Variables	
Name	Value
▶ Number of checkups	2
from	'01/01/2020'
to	'01/01/2021'

4.5.2.3 מבט המבוסס על שאלה 4.6.2

```
create view good_drones1 as
select * from drone
where 1 = routine_checks(did,
to_date(&"Starting Date",'mm/yyyy'),
to_date(&"Ending Date",'mm/yyyy'))
order by did;
```

Variables		
	Name	Value
	Starting Date	'01/2020'
▶	Ending Date	'03/2020'

סיכום

הקורס-קדם התיאורטי הכין אותנו בחשיבה הבסיסי-נתונים-ית, אך הידע התיאורטי זהה לא ייעיל אם לא מיישמים אותו. בקורס הנוכחי, למדנו לישם את כל מה שלמדנו בקורס הקודם, והוספנו לדע זהה כמה שכברות של ידע מעשי במנועי SQL וספציפית במונע Oracle. למדנו ליצור טבלאות, להזין נתונים בכל מיני שיטות, לעורך ולמחוק טבלאות, לעורך ולמחוק נתונים, ולנסח שאלות בכל מיני שיטות.

למדנו להכין פרמטרים לשאלות, וכן פרמטרים מורכבים יותר. למדנו להגדיר מבטים, פונקציות, ופrozדורות.

למדנו להתנהל עם פרויקט בצורה מרוחקת. למדנו להתגבר על קשיים שלא דמיינו שנוצרך לעBOR בתור סטודנטים. הן בלימודים והן בחני היום-יום.

החויה הייתה מאוד מלמדת, ואני ניקח את כל הכלים שרכשנו בקורס ובסמינר הקורונה יחד איתנו לתעשייה ולעתיד הזורח בסיעתא דשמייה.