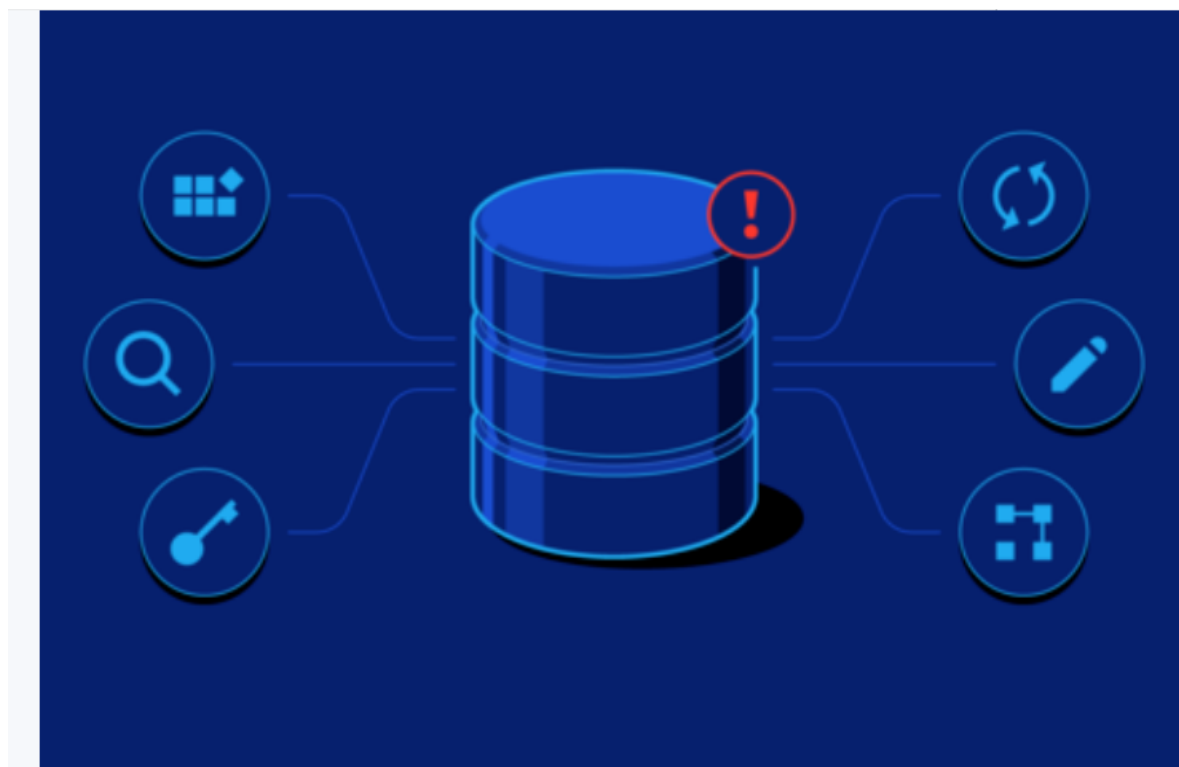


## מיני פרוייקט בסיסי נתונים



קישור לגיט

מגישות:  
שירה קרמר 209929835  
שירז כהן 319065074  
מרים פרידמן 207741810

## תוכן עניינים:

3.....	תיאור המערכת
3.....	הגרלת נתונים
4-5.....	דיאגרמות ERD ותיאור מילולי
6-9.....	שאלות
6.....	שאלתא 1
6.....	שאלתא 2
6.....	שאלתא 3
7.....	שאלתא 4
7.....	שאלתא 5
7.....	שאלתא 6
7.....	שאלתא 7
8.....	שאלתא 8
8.....	שאלתא 9
9.....	שאלתא 10
9.....	שאלתא 11
9.....	שאלתא 12
10-11.....	אינדקסים
12-13.....	שלב האינטגרציה
14-17.....	views
18-19.....	functions
20-21 .....	פרמוטציות

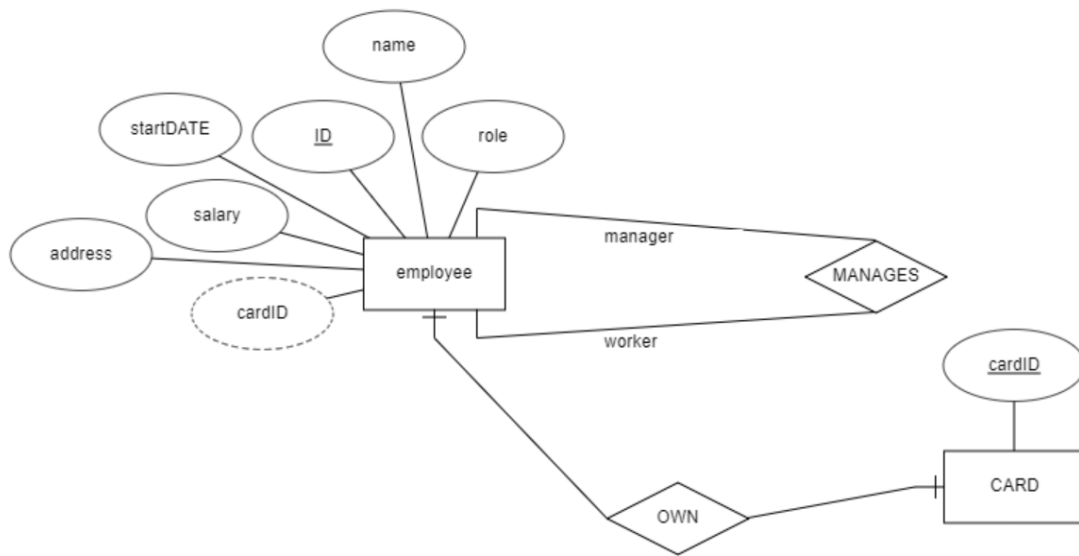
## **תיאור המערכת**

תיאור מילולי של הארגון  
בפרויקט נבנה בסיס נתונים ל-לונה פארק  
החברה כוללת מספר מחלקות, למשל: עובדים.מבקרים.מתקנים.מנויים וכו...  
חזון האתר: עיר השעשועים שמה לה למטרה להמשיך ולפתח את פארקי השעשועים במדינת ישראל,  
להתחדש ולהפתיע עם מתקנים חדשים, מרגשים ומיוחדים, בצד שמירה ללא סייג על הבטיחות והאבטחה.  
אצלו דואגים לחווית הלקוח החל ממחלקת אבדות . דוכני מזון. צוות רחב ידיים ומגוון מתקנים

## **הגרלת נתונים**

ע"מ להגריל נתונים השתמשנו ב-DATA-GENERATOR  
יצרנו 80.000-נתונים סה"כ  
נתקלנו ביצירת נתונים שאינם מקושרים לטבלה שמחזיקה ב- FOREIGN KEY  
כמו כן הגרלת נתונים מתוך רשימה אישית שלי

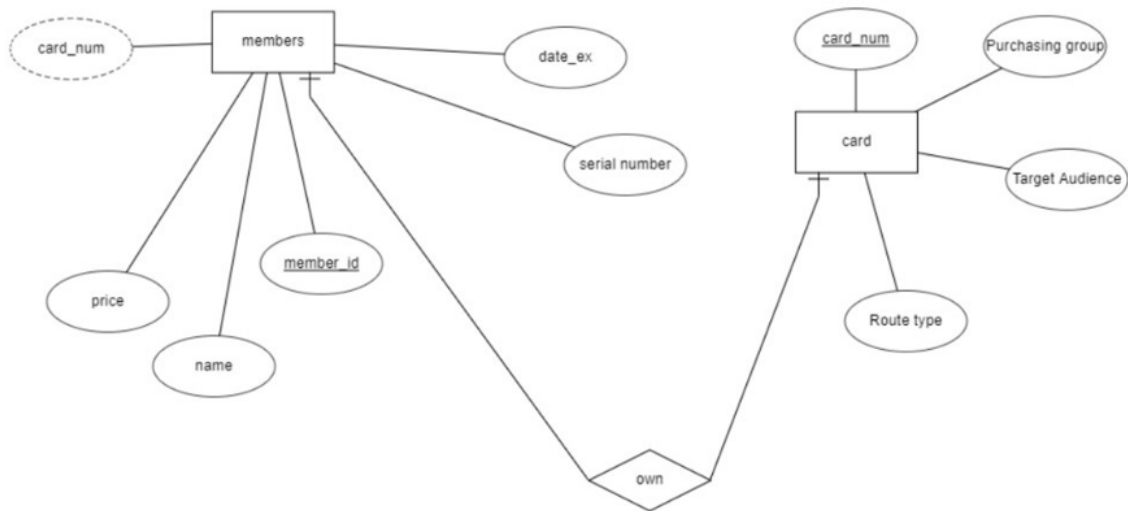
## דיאגרמות ERD וסכמות



```

CREATE TABLE CARD
)
,cardID INT NOT NULL
,cardNAME VARCHAR(35) NOT NULL
(PRIMARY KEY (cardID
;(

CREATE TABLE employee
)
,ID INT NOT NULL
,name VARCHAR(30) NOT NULL
,role VARCHAR(20) NOT NULL
,address VARCHAR(35) NOT NULL
,salary FLOAT NOT NULL
,startDATE DATE NOT NULL
,cardID INT NOT NULL
,(PRIMARY KEY (ID
(FOREIGN KEY (cardID) REFERENCES CARD(cardID
;(
    
```



```

CREATE TABLE members
)
,member_id INT NOT NULL
,date_ex DATE NOT NULL
,serial_number INT NOT NULL
,name INT NOT NULL
,price INT NOT NULL
(PRIMARY KEY (member_id
;
  
```

```

CREATE TABLE cardMEMBER
)
,card_num INT NOT NULL
,Target_Audience VARCHAR(30) NOT NULL
,Purchasing_group VARCHAR(30) NOT NULL
,Route_type VARCHAR(30) NOT NULL
,member_id INT NOT NULL
,(PRIMARY KEY (card_num
(FOREIGN KEY (member_id) REFERENCES members(member_id
;
  
```

## שאלות

### \*שאלת ראשונה

```
SELECT distinct member_id,name FROM MEMBERS NATURAL JOIN EMPLOYEE;
```

השאלתא יוצרת רשימת מנויים שהם גם עובדים בחברה (ללא כפילויות) ונותנת לנו את רשימת שמותם בצירוף מספר התעודת זהות  
השאלתא מחזירה לי 18 שורות והיא רצה בזמן של 110 שניות  
השאלתא משמשת את מנהלי המקום במידה וירצו לצ'פר עובד ולתת לו למשל מנוי מוזל.

### \*שאלת שניה

```
SELECT name,salary FROM EMPLOYEE WHERE role != 'manager' ;
```

השאלתא מחזירה לנו את רשימת העובדים שאינם מנהלים ומשכורתם  
השאלתא משמשת את מנהלי האתר למטרת דוחות של הפסדים ו-רווחים של האתר  
השאלתא מחזירה 16653  
ורצה בזמן של 5.024 שניות

### \*שאלת שלישית

```
SELECT name, member_id, date_ex FROM MEMBERS WHERE date_ex > '01-may-2022';
```

השאלתא מחזירה לנו את פרטי המנויים שיש להם מנוי בתוקף  
היא משמשת את שירות הלקוחות לדעת לאילו מנויים מומלץ להציע חידוש  
השאלתא מחזירה 2074 שורות  
ורצה בזמן של 0.664 שניות

### \*שאלת רביעית

```
SELECT name, serial_number, Purchasing_group FROM MEMBERS NATURAL JOIN cardMEMBER;
```

השאלתא מחזירה לנו את רשימת המנויים וקבוצת הרכישה שלהם  
השאלתא משמשת את ההנהלת חשבונות לקבל מידע על לקוחותיהם ועל קבוצות הרכישה שבחרו  
השאלתא מחזירה 20000 שורות  
ורצה בזמן של 5.982 sec

### **\*שאילתא חמישית**

```
SELECT name,card_num FROM cardMEMBER NATURAL JOIN MEMBERS WHERE  
ROUTE_TYPE = 'kid';
```

השאילתא מחזירה לנו את שמות ומספרי הכרטיס שקהל היעד שלהם הינו "ילד"  
השאילתא משמשת את מנהלי האתר להציע הטבות נוספות עבור קהל יעד זה.  
השאילתא מחזירה 6721 שורות  
ורצה בזמן של 1.970 sec

### **\*שאילתא שישית**

```
SELECT cM1.card_num, cM2.Card_Num, cM1.member_id, cM2.Member_Id FROM  
cardMEMBER cM1, cardMEMBER cM2 WHERE cM1.member_id= cM2.member_id  
and cM1.Purchasing_group = cM2.Purchasing_Group  
and cM1.Route_type <> cM2.Route_type;
```

השאילתא מחזירה לנו את מספרי הכרטיס ומספרי ת.ז. שקהל היעד שלהם באותו קבוצת רכישה אך לא  
באותו מסלול קהל יעד.  
השאילתא משמשת את מנהלי האתר לדעת אילו הטבות של קבוצת רכישה הכי נמכרות , להגדיל את שיתוף  
הפעולה שלהם עם אותה חברה , ולאיזה קהל יעד הכי משתלם להציע.  
השאילתא מחזירה 3428 שורות  
ורצה בזמן של 1.159sec

### **\*שאילתא שביעית**

```
SELECT ID, name,salary FROM employee WHERE salary<= 6000;
```

השאילתא מחזירה לנו את שמות העובדים, ת.ז והמשכורת של העובדים שהמשכורת שלהם קטנה מ6000.  
השאילתא משמשת את מנהלי הכספים לדעת איזה עובדים ראויים לקידום תפקיד.  
השאילתא מחזירה שורות 7914 ורצה בזמן של 2.404se

### \*שאילתא שמינית

```
SELECT e1.ID, e1.name, e1.salary, e2.ID, e2.name, e2.salary FROM employee e1,  
employee e2 WHERE e1.role='manager' and e2.role = 'manager' and e1.ID <> e2.ID and  
e1.salary= e2.salary;
```

השאילתא מחזירה את שמות העובדים שהתפקיד שלהם הוא מנהל, את ת"ז והמשכורות שלהם, כך שבכל שורה מופיע מנהלים שיש להם את אותה משכורת. מטרת השאילתא היא בעקבות החלטה שבמקום מנהל אחד לכל תפקיד יהיו 2 מנהלים לכל משימה מנהלית, ורוצים שבאותו תפקיד יהיו מנהלים עם אותה דרגה. השאילתא מחזירה 107102 שורות ורצה בזמן של 41.141sec

### \*שאילתא תשיעית

```
SELECT card_num, member_id, name, date_ex FROM cardMEMBER NATURAL JOIN  
MEMBERS WHERE date_ex >= '23-june-2022' and Route_type = 'adult';
```

שאילתא זו מחזירה את התאריך (לשם בדיקה) ת"ז, שמות, ומספרי כרטיס מנוי, של כל המבקרים המבוגרים שהתוקף מנוי שלהם גדול מסוף החודש הנוכחי, כדי לחלק לחסוך עבודה למזכירה ולהשתמש בשאילתא כשאילתת עזר שנוכל לעשות חיסור בין המבוגרים שכן עומד להסתיים להם המנוי, ובנוסף השאילתא מאפשרת לתרום מידע עבור ההצלחה בקרב המבוגרים שעדיין ממשיכים לחפוף במנוי ללונה פארק.

(שאילתת עזר, כלומר - כל המבוגרים, פחות ה"שאילתא תשע" = כל אלה שצריכים תזכורת לחדש את המנוי) השאילתא מחזירה 605 שורות ורצה בזמן של 0.284sec



### \*שאילתא עשירית

```
SELECT card_num, member_id, name, date_ex FROM cardMEMBER NATURAL JOIN  
MEMBERS WHERE Route_type = 'adult' MINUS  
SELECT card_num, member_id, name, date_ex FROM cardMEMBER NATURAL JOIN  
MEMBERS WHERE date_ex >= '23-june-2022' and Route_type = 'adult';
```

זו השאילתא שדיברנו עליה בשאילתא תשע-(מחזירה את מספר מנוי, ת"ז, שם, ותאריך) שעוזרת לנו לדעת איזה מבוגרים המנוי שלהם תם או עומד להסתיים, והמטרה של השאילתא הינה להזכיר למבוגרים לחדש את המנוי. אצלנו בלונה פארק מאמינים בלהשאיר את המבוגרים "צעירים לנצח". השאילתא מחזירה שורות 6055

ורצה בזמן של 2.210sec

### \*שאילתא אחד עשר

```
SELECT distinct member_id, serial_number, name, Purchasing_group FROM  
cardMEMBER NATURAL JOIN MEMBERS WHERE Purchasing_group != 'soldiers';
```

שאילתא זו מחזירה את הת"ז, מספר סידורי, ושם של כל המנויים שלא נמצאים בקבוצת רכישה השייכת להטבת חיילים. נרצה את השאילתא הזאת כדי שאם נחליט להעלות מחירים לא נעלה לחיילים את המחיר.

השאילתא מחזירה שורות 13327

ורצה בזמן של 4.443sec

### \*שאילתא שניים עשר

```
SELECT name, ID, salary, startDATE FROM EMPLOYEE WHERE startDATE <=  
'01-january-2018' and salary <= 4500;
```

שאילתא זו מחזירה שם, ת"ז, משכורת, ותאריך התחלת עבודה של כל העובדים שהתחילו לעבוד לפני הראשון לינואר 2018 והמשכורת שלהם קטנה או שווה ל4,500. השאילתא משמשת אותנו לדעת לאיזה עובדים יש ותק מעל 4 שנים ועל כן מגיע להם העלאה בשכר.

השאילתא מחזירה שורות 14380

ורצה בזמן של 4.968sec

## אינדקסים:

1. CREATE index cardM\_Purchasing\_group ON cardMEMBER (Purchasing\_group);

שאלתא:	לפני אינדקס מס' 1	אחרי
4	6.116	6.048
6	1.07	1.061
<u>10</u>	<u>4.267</u>	<u>2.084</u>

באינדקס זה, יש בשאלתא מס' 10 שיפור של 50% מהזמן. ובשאר השאלות שיפור לא כל כך משמעותי אך בכל זאת שיפור מסוים.

2. CREATE INDEX ro\_type\_index ON cardMEMBER (Route\_Type);

שאלתא:	לפני אינדקס מס' 2	אחרי
5	1.815	1.965
6	1.07	1.052
9	0.229	0.235
10	2.091	2.006

באינדקס זה, בשאלות 5 ו-9 הרעה מועטת מאוד של זמן, ובשאלות 6 ו-10 שיפור מועט מאוד. אז באופן כללי האינדקס לא הועיל.

3. CREATE INDEX first\_date\_emp ON employee(startdate);

שאלתא:	לפני אינדקס מס' 3	אחרי
12	4.863	4.6

באינדקס זה, יש בשאלתא מס' 12 שיפור מועט מאוד ולא משמעותי.

4. CREATE INDEX salary\_emp ON employee(salary);

<u>שאלתא:</u>	<u>לפני אינדקס מס' 4</u>	<u>אחרי</u>
2	4.879	4.689
7	6.082	6.132
<u>8</u>	<u>38.303</u>	<u>37.3</u>
12	4.611	4.635

באינדקס זה, יש בשאלתא מס' 8 שיפור של כמעט שניה. בשאלות 2 ו 7 יש שיפור מועט מאוד ובשאלתא 12 הרעה מועטה מאוד.

5. CREATE INDEX role\_emp ON employee(role);

<u>שאלתא:</u>	<u>לפני אינדקס מס' 5</u>	<u>אחרי</u>
2	4.88	4.6
<u>8</u>	<u>38.303</u>	<u>36.932</u>

באינדקס זה, יש בשאלתא מס' 8 שיפור של 2 שניות! ובשאלתא 2 שיפור מועט מאוד.

## שלב האינטגרציה:

### \*שאלתא ראשונה

```
SELECT distinct ID, name, employee_id, facilities_id FROM rofridma.facilities  
NATURAL JOIN employee WHERE ID <> EMPLOYEE_ID;
```

השאלתא מציגה (ללא כפילויות) את הת"ז עובד, והת"ז של המתקן והת"ז של העובדים בכללי, ושמותיהם. (כך שהת"ז לא אותו ת"ז באותה שורה.)  
מטרת השאלתא היא בשביל שנוכל לשבץ עבודה. (או מחדש עבור עובדים שכבר משובצים, או מהתחלה בשביל עובדים חדשים שעדיין לא משובצים).

### \*שאלתא שניה

```
select distinct visitor_phone, visitor_name, Route_type as visitor_type, visitor_ID as  
member_id FROM rofridma.visitor NATURAL JOIN cardMEMBER WHERE Route_type  
= 'adult';
```

השאלתא מציגה את כל מספרי טלפון, ואת שמות המבקרים, סוג המסלול שלהם והת"ז שלהם כך שמדובר במסלול של מבוגרים. מטרת השאלתא הינה לעזור לנו בקלות למצוא את מספרי הטלפון של המבוגרים הרשומים אצלנו, או שהיו רשומים בעבר, מכיוון שידוע לנו שהמבוגרים מתקשים יותר להתנהל עם אפליקציות בלבד ולכן נוכל לסייע להם בטלפון יותר מאשר דרך האפליקציה וכמו כן להציע להם לחדש את המנוי וכלל שירותים טלפונים שנרצה להציע להם.

### \*שאלתא שלישית

```
select distinct name, employee_id as ID, facilities_location, role FROM  
rofridma.facilities NATURAL JOIN employee WHERE facilities_location = 'north' and  
role = 'facility operator';
```

השאלתא מחזירה את שמות העובדים, ת"ז שלהם, והמיקום של המתקנים עליהם הם אחראיים, ואת שם התפקיד שלהם שנבחר להיות "מפעיל מתקן" (על מנת שנוכל להשוות בין הטבלאות) כך שהמיקום הנבחר הינו האזור הצפוני בפארק. מטרת השאלתא לעזור לנו לחלק את האוכל לעובדים שנמצאים במקום המרוחק יותר, ואולי גם לחלק לזוגות את האוכל כך שהשליחת אוכל תעשה בצורה מהירה יותר.

### **\*שאלתא רביעית**

```
select distinct name as name_mem, member_id, event_price as price, event_id,  
Purchasing_group FROM rofridma.event NATURAL JOIN members NATURAL JOIN  
cardMEMBER WHERE event_id = 1 or event_id = 5;
```

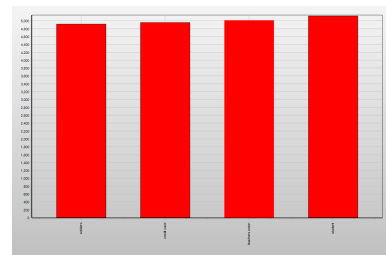
השאלתא מחזירה את שמות המנויים, ת"ז המנויים, מחיר האירוע, והת"ז של אירוע, ואת מסלול הרכישה, כך שנבחר את אירוע מספר 1 ואירוע מספר 5 ונגלה מה המחירים שלהם. מטרת השאלתא לוודא שכולם אכן שילמו את אותו מחיר עבור האירוע בעל מספר ת"ז זהה על מנת שלא ילקח כסף מהלקוחות בצורה לא הוגנת. או שלא תהיה הנחה שלא בכוונה.

## views

### \*שאלתא אחת

```
select Purchasing_group, count(*) as c_pruching from Cardmember GROUP BY  
Purchasing_group order by c_pruching
```

	PURCHASING_GROUP	C_PRUCHING
1	soldiers	4913
2	credit card	4954
3	teachers union	5012
4	student	5121



השאלתא מחזירה את מספר השורות שיש לכל מסלול רכישה עבור כל סוג מסלול רכישה שיש לטבלת המבקרים בלונה פארק ממויין לפי כמות. כך נדע כמה מבוקש כל מסלול רכישה ומה יותר נפוץ ומה פחות.

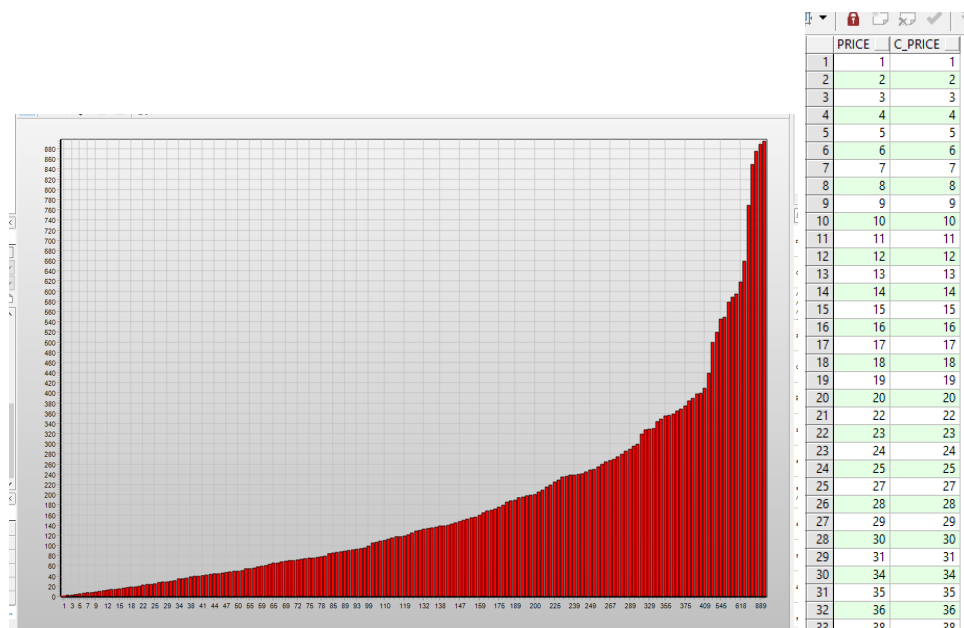
### \*שאלתא שניה

```
select date_ex, count(*) as c_date from MEMBERS GROUP BY date_ex ORDER BY  
c_date
```

השאלתא מחזירה את מספר השורות שיש לכל תאריך פג תוקף עבור כל סוג של תאריכים בטבלת המנויים, ממויין לפי תאריכים. כך נדע מי עומד להסתיים לו תוקף המנוי הכי מוקדם.

## \*שאלתא שלישית

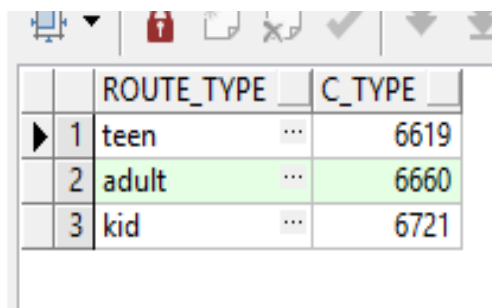
select price, max(price) as c\_price from MEMBERS GROUP BY price ORDER BY c\_price



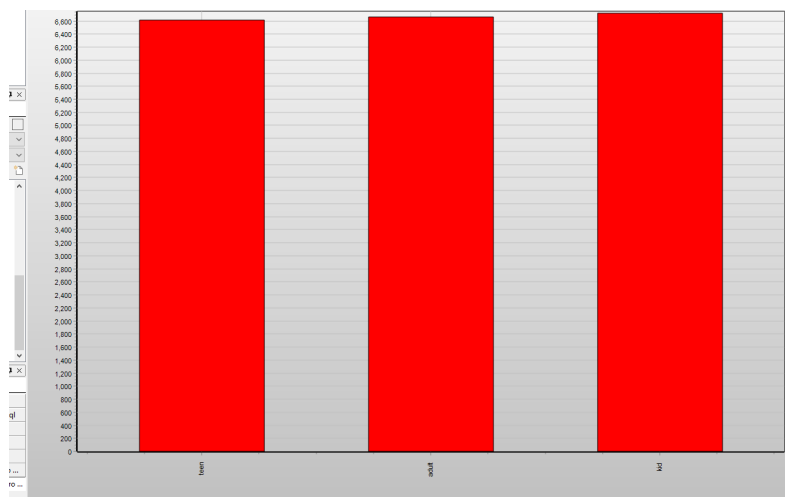
השאלתא מחזירה את כמות המחירים שיש מכל סוגי המחירים שיש לטבלת המבקרים בלונה פארק ממויין לפי גובה המחיר מהנמוך לגבוה.

### \*שאלתא רביעית

```
select Route_type, count(*) as c_type from cardMEMBER GROUP BY Route_type  
ORDER BY c_type
```



	ROUTE_TYPE	C_TYPE
1	teen	6619
2	adult	6660
3	kid	6721



השאלתא מחזירה את כמות האנשים שנמצאים בכל סוג מסלול (ילד, מתבגר, מבוגר)  
שיש לטבלת המבקרים בלונה פארק ממויין לפי כמות האנשים במסלולים השונים.



### **\*שאלתא חמישית**

```
select salary, count(*) as c_salary from EMPLOYEE GROUP BY salary ORDER BY  
c_salary
```

השאלתא מחזירה את כמות המשכורות מאותו גודל של העובדים שיש בטבלת העובדים בלונה פארק ממויין לפי גובה המחיר(המשכורת).

### **\*שאלתא שישית**

```
select role, count(*) as c_role from EMPLOYEE GROUP BY role ORDER BY c_role
```

השאלתא מחזירה את כמות האנשים שיש בכל סוג תפקיד בטבלת העובדים.

## functions:

first function

create or replace function update\_salaries

return number Ns is

cursor s1 (ID number) is select\*

from employee

where role='manager';

seniority number;

extra employee.salary%type;

overall\_update number default 0;

Result number (500);

begin

for w in s1(dep)

loop

seniority:=extract (year from startDate)-extract (year from w.startDate);

extra:=w.salary\* (seniority \*0.02):

overall\_update: overall\_update +extra:

update employee w2 set

w2.salary=w2.salary +extra

where w2.id=w.id;

commit;

end loop;

result:= 'the raise of salaries in department' || dep|| ' cost:' ||overall\_update;

return (Result);

end update\_salaries;

מעדכנת העלאה בשכר למנהלים הותיקים

**second function**

```
*create or replace function find_employee (employee_id_to_find in number) return  
employeeNAME out varchar2 is  
FunctionResult employeeNAME out varchar2;  
begin  
    select employeeNAME into employeeNAME;  
    from employee  
    where employee_id_to_find:=employee.id;  
    return (employeeNAME);  
end find_name;
```

**return employee name by his id**

**third function**

```
*create or replace function GetAmountKids return number amount  
is  
FUNCres number amount:=0;  
begin  
    select route_type into route_type;  
    from cardmember  
    where route_type='kid';  
    amount:=cont (cardmember.route_type);  
    return(FUNCres);  
end GetAmountKids;
```

**return the amount of kids**

## פרמוטציות

```
create or replace procedure findMANAGER (employeeID in number,ans out varchar2)
is
checker number;
ans varchar2 (5) NOT NULL:="False";
begin
select worker into checker
from employee
where employeeID=worker and employee.role='manager';
if (SQL%found)
then
ans:="True":
else
ans:="False":
end if;
end findMANAGER;
```

מקבלת מספר מזהה של עובד מוודא שהוא מנהל ומחזירה ערך בולאני מתאים

```
create or replace procedure isKID (memberID in number,ans out varchar2)
is
checker number;
ans varchar2 (5) NOT NULL:="False";
begin
select MEM into checker
from members
where memberID=MEM and cardmember.route_type ='kid';
if (SQL%found)
then
ans:="True":
else
ans:="False":
end if;
end isKID;
```

מקבלת מספר מזהה של מבקר מוודא שהוא ילד ומחזירה ערך בולאני מתאים

```

create or replace procedure findMEM ( serial_number in number,ans out varchar2)
is
checker number;
ans varchar2 (5) NOT NULL:="False";
begin
select memBERS into checker
from members
where serial_number=memBERS;
if (SQL%found)
then
ans:="True":
return members.name;
else
ans:="False":
end if;
end findMEM;

```

מקבלת מספר סידורי ומחזירה את שם המבקר