

2022

# מיני פרויקט -בסיסי נתונים

מגישים:  
יקיר יוחנן ונועם שמחון

גרסה: 1.2

## תוכן

|    |              |
|----|--------------|
| 2  | הקדמה        |
| 3  | אפיון        |
| 5  | אכלוס טבלאות |
| 10 | שאלות        |
| 11 | נספח         |

# הקדמה

הפרויקט הכיתתי הוא בסיס נתונים של מוזיאון. וכחלק מהמוזיאון יש גם פרטים של חנות המזכרות ורשימת בעיות התחזוקה שיש לטפל בהן (וגם אלו שכבר טופלו). בנוסף על המידע הזה מתלווה מידע על אזורי המיקומים השונים במוזיאון לצורך איתור מיקום הבעיה המדויק, וכמו כן רשימת תקלות שניתן יהיה לבחור מתוכה את תיאור הבעיה הרלוונטית.

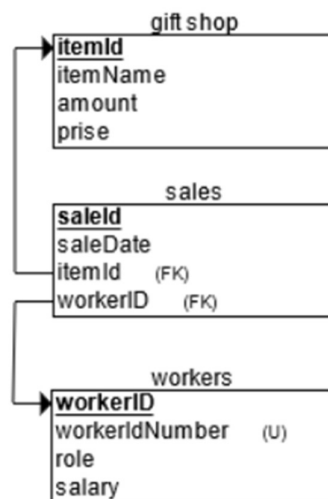
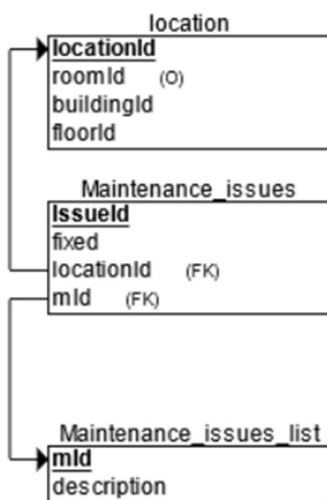
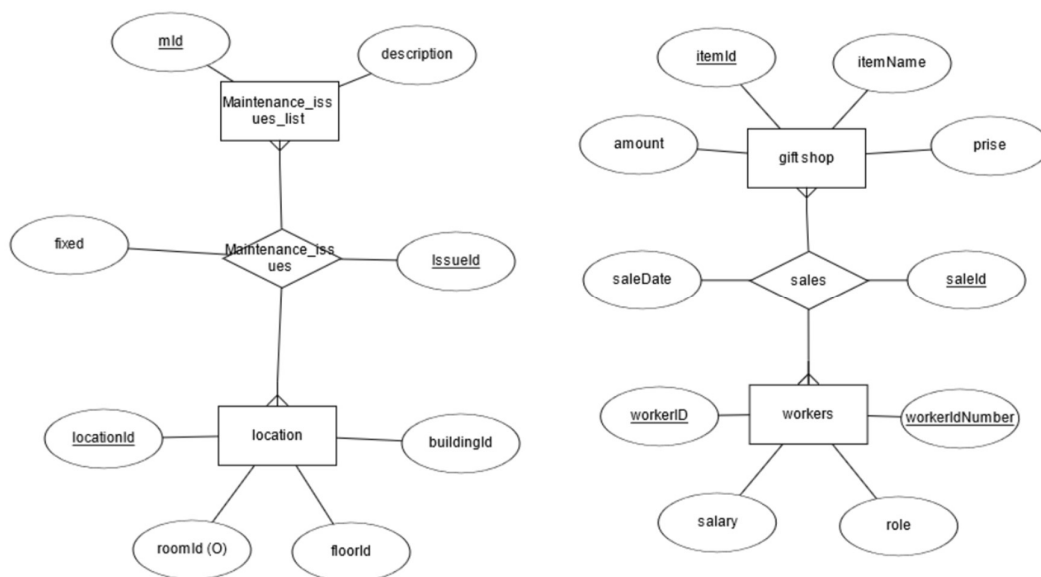
סך הכל אנו מטפלים בחמש טבלאות:

1. חנות מזכרות
2. מכירות
3. תיאורי תקלות
4. רשימת קריאות של תקלות
5. מיקומים במוזיאון (חלוקה לפי בניין, קומה, חדר)

קהל היעד להפקת מידע מהטבלאות הללו:

1. רואי חשבון של המוזיאון לצורך מעקב אחר הכנסות והוצאות של חנות המזכרות וכמות התקלות שיש במוזיאון
2. צוות התחזוקה – כדי לדעת איזו תקלה צריך לטפל היכן
3. מנהל חנות המזכרות – לצורך מעקב אחר מלאי נוכחי והזמנת מלאי חדש

# אפיון



```
CREATE TABLE gift_shop
(
  itemId INT NOT NULL,
  itemName VARCHAR(100) NOT
NULL,
  amount INT NOT NULL,
  prise INT NOT NULL,
  PRIMARY KEY (itemId)
);
```

```
CREATE TABLE location
(
  locationId INT NOT NULL,
  roomId INT,
  buildingId INT NOT NULL,
  floorId INT NOT NULL,
  PRIMARY KEY (locationId)
);
```

```
CREATE TABLE
Maintenance_issues_list
(
  mId INT NOT NULL,
  description VARCHAR(200)
NOT NULL,
  PRIMARY KEY (mId)
);
```

```
CREATE TABLE workers
(
  workerID INT NOT NULL,
  workerIdNumber INT NOT NULL,
  role VARCHAR(20) NOT NULL,
  salary INT NOT NULL,
  PRIMARY KEY (workerID),
  UNIQUE (workerIdNumber)
);
```

```
CREATE TABLE
Maintenance_issues
(
  fixed CHAR(1) NOT NULL,
  IssueId INT NOT NULL,
  locationId INT NOT NULL,
  mId INT NOT NULL,
  PRIMARY KEY (IssueId),
  FOREIGN KEY (locationId)
REFERENCES
location(locationId),
  FOREIGN KEY (mId)
REFERENCES
Maintenance_issues_list(mId)
);
```

```
CREATE TABLE sales
(
  saleId INT NOT NULL,
  saleDate DATE NOT NULL,
  itemId INT NOT NULL,
  workerID INT NOT NULL,
  PRIMARY KEY (saleId),
  FOREIGN KEY (itemId)
REFERENCES gift_shop(itemId),
  FOREIGN KEY (workerID)
REFERENCES workers(workerID)
);
```

# אכלוס טבלאות

את רשימת התקלות אכלסנו באמצעות text importer מאחר וכתבנו את רשימת התקלות באופן ידני בקובץ CSV.

The screenshot shows a software window titled 'רשימת תקלות.csv'. It has two tabs: 'Data from Textfile' (selected) and 'Data to Oracle'. The 'Data from Textfile' tab displays a list of 11 fault entries, each with an ID and a description. Below the list is a 'Configuration' section with various settings for importing the data. The 'General' section includes 'Fieldcount' (2), 'End at line-end' (checked), 'Name in header' (unchecked), and 'Skip empty lines' (checked). The 'Quote character' is set to double quotes, and 'Import lines' are set to 1. The 'Field Start' section has 'Relative position' selected with a value of 0. The 'Field End' section has 'Character' selected. The 'Result Preview' section shows a table with two columns: 'ID' and 'Description'. The first two rows are highlighted in blue.

| ID | Description          |
|----|----------------------|
| 1  | Broken light         |
| 2  | Broken door          |
| 3  | Short on electricity |
| 4  | A faulty computer    |
| 5  | A broken fence       |
| 6  | Peeled paint         |
| 7  | Broken protrusion    |
| 8  | Broken exhibition    |
| 9  | Internet issues      |
| 10 | Faulty exhibition    |
| 11 | Faulty touch sensor  |

Configuration:

General

Fieldcount: 2

☒ End at line-end

☐ Name in header

☒ Skip empty lines

Quote character: "

Comment line:

Import lines: 1 ..

Field1 (+0 .. ",")

Field2 (+0 .. ",")

Field Start

☒ Relative position

☐ Absolute position

☐ Character

Field End

☐ Length

☒ Character

Filter:

Result Preview

| ID | Description  |
|----|--------------|
| 1  | Broken light |
| 2  | Broken door  |

את טבלת המיקומים מילאנו באמצעות data generator והכל זה ערכים רנדומליים של מספר בניין קומה וחדר במוזיאון.

**LOCATION**

Owner: SYSTEM Table: LOCATION Number of records: 400

| Name       | Type   | Size | Data          |
|------------|--------|------|---------------|
| LOCATIONID | NUMBER |      | Sequence(1,1) |
| ROOMID     | NUMBER |      | Random(1,10)  |
| BUILDINGID | NUMBER |      | Random(1,10)  |
| FLOORID    | NUMBER |      | Random(1,5)   |
| *          |        |      |               |

את טבלת רשימת התקלות מילאנו ב data generator מאחר והכל זה קומבינאציות רנדומליות בין מיקומים ותקלות.

**MAINTENANCE\_ISSUES**

Owner: SYSTEM Table: MAINTENANCE\_ISSUES Number of records: 20000

| Name       | Type   | Size | Data          |
|------------|--------|------|---------------|
| ISSUEID    | NUMBER |      | Sequence(1,1) |
| FIXED      | CHAR   | 1    | Random(0,1)   |
| LOCATIONID | NUMBER |      | Random(1,400) |
| MID        | NUMBER |      | Random(1,13)  |
| *          |        |      |               |

WORKERS

|          |         |                   |
|----------|---------|-------------------|
| Owner    | Table   | Number of records |
| YIOHANAN | WORKERS | 500               |

| Name           | Type     | Size | Data  |
|----------------|----------|------|---|
| WORKERID       | NUMBER   |      | Sequence(1,1)                                   |
| WORKERIDNUMBER | NUMBER   |      | Random(321321122,765432122)                     |
| ROLE           | VARCHAR2 | 20   |   |
| SALARY         | NUMBER   |      | List(3000,3000,3000,3000,6000,9000,15000,50000) |
| *              |          |      |   |

את טבלת חנות המזכרות מלאנו גם ב־text importer מקובץ csv שמכיל רשימת מוצרים וגם מ־data generator שמספק ערכים רנדומליים לכמות ולמחיר.

GIFT\_SHOP

|          |           |                   |
|----------|-----------|-------------------|
| Owner    | Table     | Number of records |
| YIOHANAN | GIFT_SHOP | 500               |

| Name     | Type     | Size | Data           |
|----------|----------|------|----------------|
| ITEMID   | NUMBER   |      | Sequence(1,1)  |
| ITEMNAME | VARCHAR2 | 100  |                |
| AMOUNT   | NUMBER   |      | Random(20,100) |
| PRICE    | NUMBER   |      | Random(50,200) |
| *        |          |      |                |



newGiftShopTable.csv

Data from Textfile Data to Oracle

**General**

Owner: YYOHANAN Table: GIFT\_SHOP

Commit every...: 100

☒ Overwrite duplicates ☐ Delete records  
☐ Ignore duplicates ☐ Truncate table

Initializing Script:   
Finalizing Script:

**Fields**

Field1 -> ITEMID (NUMBER)  
Field2 -> ITEMNAME (VARCHAR2)  
Field3 -> AMOUNT (NUMBER)  
Field4 -> PRISE (NUMBER)  
Field5 ->

Field: PRISE (NUMBER)  
Fieldtype: Number  
Create SQL

SQL function:   
additional Oracle processing, for example: substr(%, 1, 20)

**Result Preview**

| 1 | 2  | 3  | 4   | 5                 |
|---|--|----|-----|-------------------|
| 1 | Ancient Hebrew Pendant with chain (silver) | 25 | 91  | AACXcoAGAAADR1AAA |
| 2 | Hamsa -Pendant with chain (silver)         | 63 | 172 | AACXcoAGAAADR1AAB |

Import Import to Script Close yyohanana@labdbwin 400 records imported in 0.469 seconds

--select count(\*) from mai ... YYOHANAN.SALES@LABDBWIN newSaleTable2.csv Data Generator

Data from Textfile Data to Oracle

**General**

Owner: YYOHANAN Table: SALES

Commit every...: 100

☒ Overwrite duplicates ☐ Delete records  
☐ Ignore duplicates ☐ Truncate table

Initializing Script:   
Finalizing Script:

**Fields**

Field1 -> SALEID (NUMBER)  
Field2 -> SALEDATE (DATE)  
Field3 -> ITEMID (NUMBER)  
Field4 -> WORKERID (NUMBER)

Field: WORKERID (NUMBER)  
Fieldtype: Number  
Create SQL

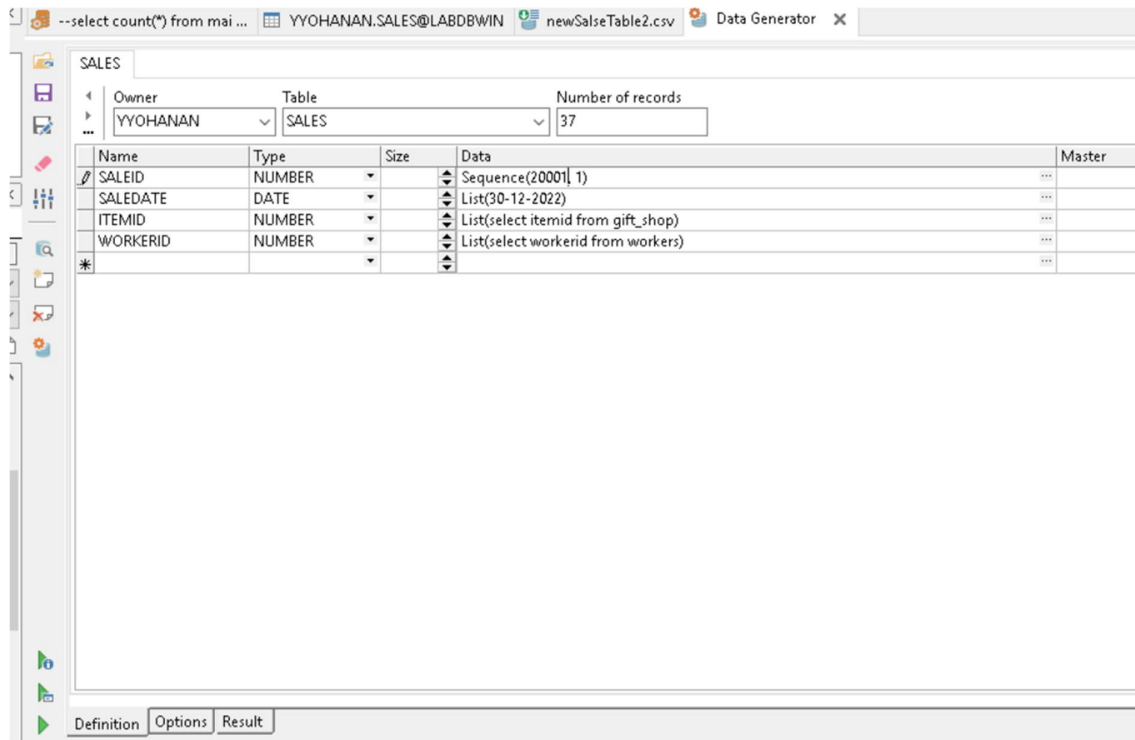
SQL function:   
additional Oracle processing, for example: substr(%, 1, 20)

**Result Preview**

| 1     | 2          | 3   | 4   |
|-------|------------|-----|-----|
| 19998 | 29/12/2022 | 252 | 391 |
| 19999 | 29/12/2022 | 133 | 66  |

Import Import to Script Close yyohanana@labdbwin 101 records imported in 1.094 seconds (27 duplicates)

את טבלת המכירות מלאו באמצעות data generator המספק ערכים מבין הטבלאות השונות. וגם מ text importer שלוקח מקובץ csv שנעשה באמצעות תכנית ב-c# שמכיל תאריכים של מכירות.



# שאלות

1. שאלתה המחזירה את תיאור התקלות שתמיד טופלו – בכדי שבעל המוזיאון ידע מאילו בעיות לא לדאוג כלל.  
שורה 1 – 0.065 שניות
2. שאלתה המחזירה את מיקום החדר ותיאור התקלה שלא טופלה בבנין 1 קומה 2. – כמובן שניתן להחליף את המיקום לפי דרישה. חשוב כדי להתכונן לאירוח אנשים חשובים וכדו'.  
57 שורות – 0.053 שניות
3. שאלתה המחזירה את מספרי 40 המוצרים שנמכרו הכי הרבה, מי מכר אותם הכי הרבה, וכמה פעמים הוא מכר – כדי לחלק פרסי מצטיינים  
40 שורות – 0.066 שניות
4. שאלתה המחזירה את הפריט שנמכר הכי הרבה וכמה החנות הרוויחה ממנו עד כה – כדי לדעת איזה פריט הכי רווחי כמובן.  
שורה אחת – 0.045 שניות
5. שאלתה המחזירה את פרטי המיקום שנפתחו בו הכי הרבה פניות לבעיות תחזוקה ואת מספר הפניות – כדי לדעת באיזה חדר לעשות את השיפוץ הכללי הבא.  
שורה אחת – 0.030 שניות
6. שאלתה המחזירה את התאריך ומספר המכירות שהתבצעו באותו יום – כדי לראות איזה יום היה הכי רווחי.  
1043 שורות – 0.587 שניות
7. שאלתה המחזירה את שמות הפריטים שצריך להזמין בקרוב (אלו בעלי כמות שפחותה מ-25) ואת הכמות הנוכחית של כל אחד מהם – כדי לדעת מה להזמין בהזמנה הקרובה כמובן.  
20 שורות – 0.532 שניות – עם אינדקס על הכמות 0.031 שניות
8. שאלתה המחזירה את תיאור בעיית התחזוקה שנקראה הכי הרבה פעמים ואת מספר הפעמים שנקראה – כדי לדעת האם יש בעיה שחייבת טיפול רוחבי במוזיאון  
שורה אחת – 0.026 שניות

שאלות אינטגרציה:

1. 40 המוצרים שנמכרו הכי הרבה ושם של מי מכר אותם הכי הרבה וכמה פעמים נמכרו
2. שם ושכר של העובד שמכר הכי הרבה מוצרים
3. שם ומייל של 40 העובדים המבוגרים ביותר במוזיאון

# נספח

.1

```
select distinct description
  from maintenance_issues_list m1, maintenance_issues m2
  where m1.mid = m2.mid
        and m2.mid not in (
            select distinct m1.mid
              from maintenance_issues_list m1, maintenance_issues m2
             where m1.mid = m2.mid
                   and m2.fixed = 0
        );
```

.2

```
select distinct l.roomid, m1.description
  from maintenance_issues_list m1, maintenance_issues m2, location l
  where m1.mid = m2.mid
        and m2.locationid = l.locationid
        and l.buildingid = 1
        and l.floorid = 2
        and m2.fixed = 0
        order by l.roomid;
```

.3

```
select workerid, itemid, count(saleid)
  from sales
  where itemid IN (select itemid
                    from sales
                   group by itemid
                  order by count(saleid) desc
                  FETCH FIRST 40 ROW only)
  group by workerid, itemid
  order by count(saleid) desc
  FETCH FIRST 40 ROW only;
```

.4

```
select itemname, prise* (select count(saleid)
                        from sales
                        group by itemid
                        order by count(saleid)
                        desc fetch first 1 row only) as profit
from gift_shop
where itemid = (
    select itemid
    from sales
    group by itemid
    order by count(saleid)
    desc fetch first 1 row only)
```

.5

```
select l.*, (select count(locationid)
            from maintenance_issues
            group by locationid
            order by count(locationid) desc
            fetch first 1 row only) as call_times
from location l
where locationid = (select locationid
                  from maintenance_issues
                  group by locationid
                  order by count(locationid) desc
                  fetch first 1 row only)
```

.6

```
select saledate, count(saleid) as sales_number
from sales
group by saledate
```

.7

```
select itemname, amount
from gift_shop
where amount < 25
```

.8

```
select m.description, (select count(mid)
                        from maintenance_issues
                        group by mid
                        order by count(mid) desc
                        fetch first 1 row only) as calls_number
from maintenance_issues_list m
where mid = (select mid
             from maintenance_issues
             group by mid
             order by count(mid) desc
             fetch first 1 row only)
```

שאלות אינטגרציה:

.1

```
-- 40 products most saled and who saled them the most
select w.name, itemid, count(saleid)
from sales, mklempne.worker w
where itemid IN (select itemid
                 from sales
                 group by itemid
                 order by count(saleid) desc
                 FETCH FIRST 40 ROW only)
AND
sales.workerid = w.workerid

group by w.name, itemid
order by count(saleid) desc
FETCH FIRST 40 ROW only;
```

.2

```
-- name and salary of the goodest worker
select w.name,w.salaryperhour
from sales, mklempne.worker w
where itemid IN (select itemid
                 from sales
                 group by itemid
                 order by count(saleid) desc
                 FETCH FIRST 1 ROW only)
AND
sales.workerid = w.workerid

group by w.name,w.salaryperhour
order by count(saleid) desc

FETCH FIRST 1 ROW only;
```

```
-- third query - the most oldest workers in the musium
select w.name, w.email
from mklemfne.worker w
where w.birthdate in (select w.birthdate
                      from mklemfne.worker w
                      order by w.birthdate desc
                      fetch first 40 row only )
```