

# סופר מרקט-אונליין



---

מגישות:  
תהלה כהן ושרה מנסור

---

---

תוכן עניינים

|              |    |
|--------------|----|
| מבוא         | 3  |
| תרשים ERD    | 4  |
| תרשים DSD    | 4  |
| החלטות עיצוב | 5  |
| הכנסת ערכים  | 6  |
| גיבוי נתונים | 13 |
| שחזור נתונים | 14 |

---

## מבוא

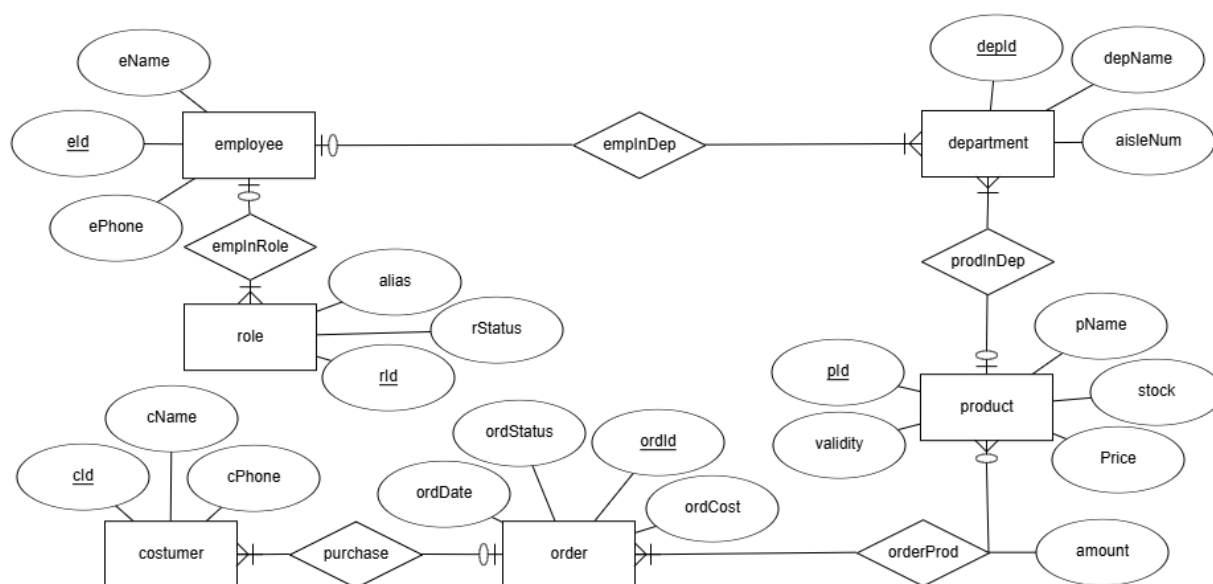
המערכת לניהול סופרמרקט אונליין נועדה לייעל את תהליכי הניהול והמעקב אחר כל מרכיבי העסק, כולל עובדים, תפקידים, מחלקות, לקוחות, מוצרים והזמנות. המערכת מספקת פתרון מקיף לניהול חויית הקנייה בצורה נוחה ויעילה.

### עיקרי המערכת:

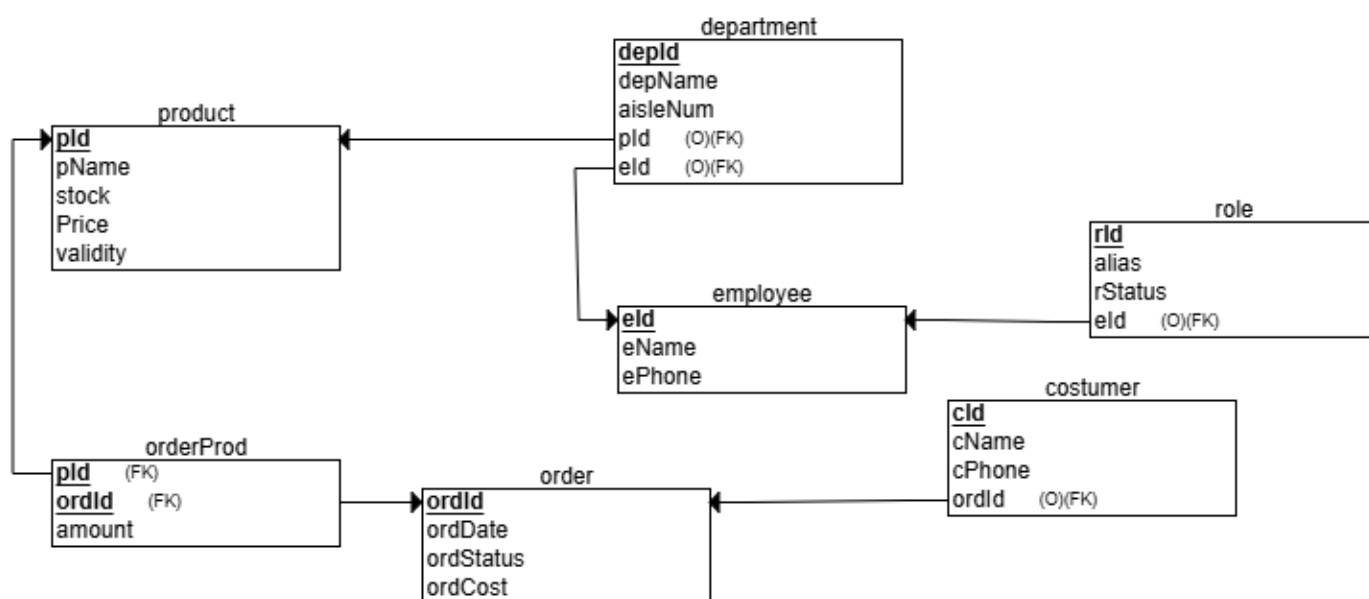
- **ניהול עובדים ותפקידיהם במחלקות השונות** – המערכת מאפשרת רישום ומעקב אחר העובדים, כולל שיוכם לתפקידים ולמחלקות הרלוונטיות.
- **מעקב אחר לקוחות ורכישותיהם** – ניתן לשמור נתוני לקוחות והיסטוריית קניות.
- **ניהול מלאי מוצרים לצורך הספקתם לפי צורך** – מעקב אוטומטי אחר זמינות מוצרים, עדכון מצב המלאי בזמן אמת, על מנת לספק סחורה חדשה בעת הצורך בהתאם לכמות במלאי ולמנוע חוסרים.
- **ניהול הזמנות אונליין ושיוך מוצרים להזמנות לקוח** – תמיכה בתהליך הרכישה המקוונת, כולל שיוך פריטים להזמנות, עלות הרכישה הכוללת ומעקב אחר סטטוס ההזמנה עד להגעתו ללקוח.
- **ארגון המחלקות של הסופרמרקט, כולל מיקום פיזי ואזורי** – תכנון וניהול מבנה החנות, כולל סידור המחלקות והמוצרים בהתאם לצרכים תפעוליים ולנוחות הלקוחות.

.....

## תרשים ERD



## תרשים DSD



# החלטות עיצוב

הסכימה מורכבת ממשויות עיקריות: לקוחות, הזמנות, מוצרים, מחלקות, עובדים ותפקידים, עם קשרים שונים ביניהן.

לקוח יכול לבצע אפס או יותר הזמנות, אך כל הזמנה חייבת להיות משויכת ללקוח אחד. קשר של אחד לרבים – חובה בצד ההזמנה, אופציונלי בצד הלקוח.

הזמנה יכולה לכלול כמה מוצרים, כאשר כל מוצר יכול להופיע במספר הזמנות, דרך טבלת קשר המנהלת את הכמות של כל מוצר בכל הזמנה קשר של רבים לרבים.

מוצר חייב להשתייך למחלקה אחת, ומחלקה יכולה לכלול מספר מוצרים. קשר של אחד לרבים – חובה בצד המוצר, אופציונלי בצד המחלקה.

עובד חייב להשתייך למחלקה אחת בלבד, בכל מחלקה יכולים להיות הרבה עובדים. קשר של אחד לרבים – חובה בצד העובד, אופציונלי בצד המחלקה.

עובד מאייש תפקיד אחד בלבד, לעומת זאת ישנם תפקידים שיכולים להיות מאוישים על ידי יותר מעובד אחד. קשר של יחיד לרבים – חובה בצד העובד, אופציונלי בצד התפקיד, כיון שיכולים להיות תפקידים שאינם מאוישים עדיין או שמושבתים.

## מגבלות עיצוב:

- סטטוס ההזמנה חייב להיות מתוך רשימה מוגדרת מראש ("הזמנה", "נשלחה", "הגיעה" או "בוטלה"), כמו כן תפקיד יכול להיות פעיל או לא פעיל או מושהה.
- כמות במלאי לא יכולה להיות שלילית.
- תאריך תפוגה של מוצר חייב להיות בעתיד.

## יתרונות:

- מבנה גמיש לניהול מחלקות, מוצרים ועובדים.
- תמיכה בהיסטוריית רכישות ויכולת ניהול דינמית של תפקידים.
- שימוש נכון בקשרים של אחד לרבים ורבים לרבים לניהול מידע מורכב.

## סיכום:

הסכימה מאפשרת ניהול חכם ויעיל של סופרמרקט אונליין, תוך התחשבות בגמישות וניהול מידע מתקדם.

# הכנסת נתונים

בחרנו להכניס נתונים בשיטת קובץ סקריפט בפייתון, קבצי אקסל וmockaroo.

1) פייתון

צילום Table.py

```
Table.py ×
import random
from datetime import datetime, timedelta

# מבנה הטבלאות ושמות השדות שלהן
tables = {
    "product": ["pId", "pName", "stock", "price", "validity"],
    "employee": ["eId", "eName", "ePhone"],
    "role": ["rId", "alias", "rStatus", "eId"],
    "orders": ["ordId", "ordDate", "ordStatus", "ordCost"],
    "customer": ["cId", "cName", "cPhone", "ordId"],
    "orderProd": ["amount", "pId", "ordId"],
    "department": ["depId", "depName", "aisleNum", "pId", "eId"]
}

products = [
    "Milk", "Bread", "Eggs", "Cheese", "Butter", "Tomato", "Potato", "Chicken", "Beef", "Fish",
    "Rice", "Pasta", "Sugar", "Salt", "Oil", "Yogurt", "Orange", "Apple", "Banana", "Grapes",
    "Carrot", "Cucumber", "Lettuce", "Strawberry", "Blueberry", "Peach", "Pear", "Mango", "Coconut",
    "Honey", "Jam", "Chocolate", "Tea", "Coffee", "Water", "Juice", "Soda", "Cereal", "Flour"
]

employees = [(i, f"Employee_{i}") for i in range(1, 500)]
customers = [(i, f"Customer_{i}") for i in range(1, 500)]
roles = ["Manager", "Cashier", "Stocker"]

output_file = "generated_insert.sql"
```

```

employee_dict = {e[0]: e[1] for e in employees} # מיון מזהי עובדים לשמות

def generate_random_value(field_name, ord_id=0):
    if "pName" in field_name:
        return f'"{random.choice(products)}"'
    elif "eName" in field_name:
        return f'"{employees.pop()[1]}"'
    elif "cName" in field_name:
        return f'"{customers.pop()[1]}"'
    elif "eId" in field_name:
        return random.choice(list(employee_dict.keys()))
    elif "cId" in field_name:
        return customers.pop()[0]
    elif "id" in field_name.lower():
        return random.randint(1, 10000)
    elif "phone" in field_name.lower():
        return f'"{random.randint(10000, 99999)}"'
    elif "ordDate" in field_name:
        start_date = datetime.today() - timedelta(days=365)
        random_date = start_date + timedelta(days=random.randint(1, 730))
        return f'"{random_date.strftime('%Y-%m-%d')}"'
    elif "validity" in field_name.lower():
        return f'"{(datetime.today() + timedelta(days=random.randint(30, 365))).strftime('%Y-%m-%d')}"'

with open(output_file, "w", encoding="utf-8") as f:
    order_totals = {}

    for table, fields in tables.items():
        for _ in range(400):
            values = [str(generate_random_value(field)) for field in fields]
            sql_insert = f"INSERT INTO {table} ({', '.join(fields)}) VALUES ({', '.join(values)});\n"

            # מחשב את הסכום של כל פריטי הזמנה לסכום הזמנה
            if table == "orderProd":
                amount = int(values[0])
                pId = int(values[1])
                ordId = int(values[2])
                price = round(random.uniform(10, 100), 2)

                if ordId not in order_totals:
                    order_totals[ordId] = 0
                order_totals[ordId] += amount * price

            f.write(sql_insert)

    # עדכון סכומי הזמנות
    for ordId, total_cost in order_totals.items():
        f.write(f"UPDATE orders SET ordCost = {round(total_cost, 2)} WHERE ordId = {ordId};\n")

print(f"SQL insert statements saved to {output_file}")

```

צילומי קובץ פלט:



```

1 INSERT INTO product (pId, pName, stock, price, validity) VALUES (8562, 'Pear', 421, 30.02, 7148);
2 INSERT INTO product (pId, pName, stock, price, validity) VALUES (7759, 'Milk', 78, 29.04, 4343);
3 INSERT INTO product (pId, pName, stock, price, validity) VALUES (9842, 'Eggs', 454, 91.01, 7440);
4 INSERT INTO product (pId, pName, stock, price, validity) VALUES (1300, 'Lettuce', 479, 22.87, 8794);
5 INSERT INTO product (pId, pName, stock, price, validity) VALUES (1000, 'Oil', 437, 86.7, 5935);
6 INSERT INTO product (pId, pName, stock, price, validity) VALUES (1607, 'Blueberry', 115, 60.63, 3668);
7 INSERT INTO product (pId, pName, stock, price, validity) VALUES (1057, 'Mango', 209, 78.19, 6554);
8 INSERT INTO product (pId, pName, stock, price, validity) VALUES (5208, 'Grapes', 260, 86.14, 6269);
9 INSERT INTO product (pId, pName, stock, price, validity) VALUES (1600, 'Chocolate', 457, 64.09, 8462);
10 INSERT INTO product (pId, pName, stock, price, validity) VALUES (540, 'Pasta', 496, 85.91, 218);
11 INSERT INTO product (pId, pName, stock, price, validity) VALUES (1480, 'Juice', 210, 43.28, 4612);
12 INSERT INTO product (pId, pName, stock, price, validity) VALUES (8029, 'Tomato', 234, 23.21, 3757);
13 INSERT INTO product (pId, pName, stock, price, validity) VALUES (8890, 'Tea', 199, 48.18, 7031);
14 INSERT INTO product (pId, pName, stock, price, validity) VALUES (883, 'Bread', 141, 31.87, 6616);
15 INSERT INTO product (pId, pName, stock, price, validity) VALUES (1060, 'Yogurt', 137, 36.21, 1508);
16 INSERT INTO product (pId, pName, stock, price, validity) VALUES (4949, 'Butter', 441, 57.17, 5096);
17 INSERT INTO product (pId, pName, stock, price, validity) VALUES (5288, 'Milk', 374, 40.63, 7832);
18 INSERT INTO product (pId, pName, stock, price, validity) VALUES (778, 'Butter', 288, 51.98, 9042);
19 INSERT INTO product (pId, pName, stock, price, validity) VALUES (1256, 'Cheese', 234, 23.21, 3757);
20 INSERT INTO product (pId, pName, stock, price, validity) VALUES (8322, 'Orange', 485, 81.71, 715);
21 INSERT INTO product (pId, pName, stock, price, validity) VALUES (4371, 'Potato', 220, 89.35, 6331);
22 INSERT INTO product (pId, pName, stock, price, validity) VALUES (7642, 'Flour', 259, 27.96, 5287);
23 INSERT INTO product (pId, pName, stock, price, validity) VALUES (2873, 'Salt', 493, 98.17, 9396);
24 INSERT INTO product (pId, pName, stock, price, validity) VALUES (8559, 'Butter', 367, 49.21, 1047);
25 INSERT INTO product (pId, pName, stock, price, validity) VALUES (634, 'Bread', 342, 46.67, 5305);
26 INSERT INTO product (pId, pName, stock, price, validity) VALUES (7349, 'Soda', 422, 60.92, 4589);
27 INSERT INTO product (pId, pName, stock, price, validity) VALUES (8742, 'Apple', 73, 68.78, 6896);
28 INSERT INTO product (pId, pName, stock, price, validity) VALUES (7150, 'Potato', 197, 74.12, 7900);
29 INSERT INTO product (pId, pName, stock, price, validity) VALUES (1651, 'Mango', 265, 51.25, 8666);
30 INSERT INTO product (pId, pName, stock, price, validity) VALUES (5864, 'Eggs', 453, 36.91, 8743);
31 INSERT INTO product (pId, pName, stock, price, validity) VALUES (4894, 'Butter', 284, 71.54, 2792);
32 INSERT INTO product (pId, pName, stock, price, validity) VALUES (8467, 'Flour', 343, 98.88, 4525);
33 INSERT INTO product (pId, pName, stock, price, validity) VALUES (7186, 'Yogurt', 373, 71.69, 9472);
34 INSERT INTO product (pId, pName, stock, price, validity) VALUES (9975, 'Yogurt', 246, 72.48, 2834);
35 INSERT INTO product (pId, pName, stock, price, validity) VALUES (649, 'Sugar', 83, 66.56, 1405);
36 INSERT INTO product (pId, pName, stock, price, validity) VALUES (2610, 'Potato', 126, 23.9, 2306);
37 INSERT INTO product (pId, pName, stock, price, validity) VALUES (7538, 'Yogurt', 93, 40.99, 9750);
38 INSERT INTO product (pId, pName, stock, price, validity) VALUES (7834, 'Cucumber', 151, 96.66, 3251);
39 INSERT INTO product (pId, pName, stock, price, validity) VALUES (7905, 'Jam', 333, 78.53, 1281);
40 INSERT INTO product (pId, pName, stock, price, validity) VALUES (6866, 'Blueberry', 10, 90.1, 2047);
41 INSERT INTO product (pId, pName, stock, price, validity) VALUES (4215, 'Orange', 174, 51.02, 1997);
42 INSERT INTO product (pId, pName, stock, price, validity) VALUES (2695, 'Beef', 134, 77.15, 5728);
43 INSERT INTO product (pId, pName, stock, price, validity) VALUES (6456, 'Eggs', 60, 76.04, 4013);
44 INSERT INTO employee (eId, eName, ePhone) VALUES (82, 'Employee_499', '68548');
45 INSERT INTO employee (eId, eName, ePhone) VALUES (55, 'Employee_498', '73450');
46 INSERT INTO employee (eId, eName, ePhone) VALUES (77, 'Employee_497', '90714');
47 INSERT INTO employee (eId, eName, ePhone) VALUES (16, 'Employee_496', '58387');
48 INSERT INTO employee (eId, eName, ePhone) VALUES (282, 'Employee_495', '28766');
49 INSERT INTO employee (eId, eName, ePhone) VALUES (96, 'Employee_494', '31715');
50 INSERT INTO employee (eId, eName, ePhone) VALUES (439, 'Employee_493', '88807');
51 INSERT INTO employee (eId, eName, ePhone) VALUES (193, 'Employee_492', '88906');
52 INSERT INTO employee (eId, eName, ePhone) VALUES (183, 'Employee_491', '38791');
53 INSERT INTO employee (eId, eName, ePhone) VALUES (109, 'Employee_490', '25900');
54 INSERT INTO employee (eId, eName, ePhone) VALUES (159, 'Employee_489', '56278');
55 INSERT INTO employee (eId, eName, ePhone) VALUES (320, 'Employee_488', '90174');
56 INSERT INTO employee (eId, eName, ePhone) VALUES (107, 'Employee_487', '84932');
57 INSERT INTO employee (eId, eName, ePhone) VALUES (218, 'Employee_486', '79682');
58 INSERT INTO employee (eId, eName, ePhone) VALUES (485, 'Employee_485', '15576');

```

(2) צילומי קבצי אקסל:

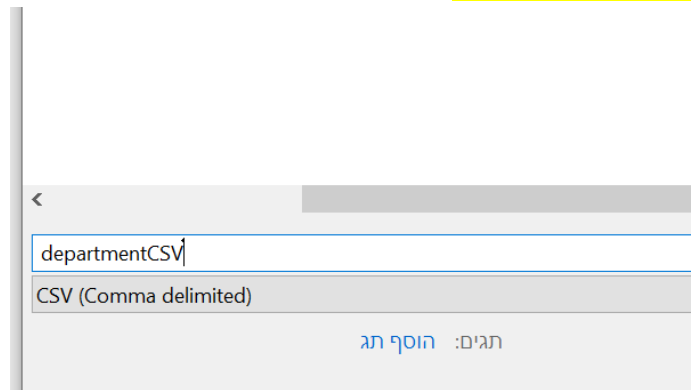


| E   | D   | C        | B          | A     |    |
|-----|-----|----------|------------|-------|----|
| eid | pid | aislenum | depname    | depid | 1  |
| 63  | 259 | 4        | Bakery     | 1     | 2  |
| 349 | 298 | 16       | Grocery    | 2     | 3  |
| 280 | 90  | 15       | Beverages  | 3     | 4  |
| 201 | 93  | 7        | Frozen     | 4     | 5  |
| 89  | 3   | 8        | Vegetables | 5     | 6  |
| 376 | 356 | 14       | Bakery     | 6     | 7  |
| 81  | 277 | 3        | Meat       | 7     | 8  |
| 299 | 223 | 14       | Beverages  | 8     | 9  |
| 259 | 319 | 17       | Frozen     | 9     | 10 |
| 356 | 372 | 18       | Dairy      | 10    | 11 |
| 358 | 176 | 10       | Frozen     | 11    | 12 |
| 239 | 335 | 9        | Dairy      | 12    | 13 |
| 364 | 108 | 17       | Beverages  | 13    | 14 |
| 121 | 369 | 3        | Meat       | 14    | 15 |
| 147 | 2   | 19       | Meat       | 15    | 16 |
| 50  | 257 | 4        | Beverages  | 16    | 17 |
| 136 | 25  | 5        | Bakery     | 17    | 18 |
| 160 | 154 | 5        | Vegetables | 18    | 19 |
| 338 | 33  | 13       | Vegetables | 19    | 20 |

| E | D     | C        | B          | A   |    |
|---|-------|----------|------------|-----|----|
|   | ordid | cphone   | cname      | cid | 1  |
|   | 257   | 6.42E+08 | Itay Levi  | 1   | 2  |
|   | 212   | 3.41E+08 | Yosef Levi | 2   | 3  |
|   | 164   | 7.34E+08 | Shira Pere | 3   | 4  |
|   | 352   | 8.17E+08 | Omer Gab   | 4   | 5  |
|   | 249   | 7.77E+08 | Yosef Seg  | 5   | 6  |
|   | 280   | 2.19E+08 | Yonatan S  | 6   | 7  |
|   | 251   | 5.32E+08 | Miriam Ma  | 7   | 8  |
|   | 161   | 7.38E+08 | Omer Malk  | 8   | 9  |
|   | 238   | 7.2E+08  | Shira Tzac | 9   | 10 |
|   | 230   | 1.57E+08 | Natan Coh  | 10  | 11 |
|   | 135   | 5.31E+08 | Miriam Pe  | 11  | 12 |
|   | 364   | 9.44E+08 | Esther Ezz | 12  | 13 |
|   | 45    | 6.46E+08 | Eli Dayan  | 13  | 14 |
|   | 79    | 8.75E+08 | Natan Gab  | 14  | 15 |
|   | 344   | 8.82E+08 | Rachel Tza | 15  | 16 |
|   | 70    | 4.55E+08 | Omer Tzac  | 16  | 17 |
|   | 159   | 6.7E+08  | Daniel Per | 17  | 18 |
|   | 106   | 7.77E+08 | Avigail Ma | 18  | 19 |
|   | 360   | 4.55E+08 | Shira Seg  | 19  | 20 |

|  | D | C        | B          | A   |    |
|--|---|----------|------------|-----|----|
|  |   | ephone   | ename      | eid | 1  |
|  |   | 9.97E+08 | Itay Levi  | 1   | 2  |
|  |   | 8.34E+08 | Yosef Levi | 2   | 3  |
|  |   | 8.4E+08  | Shira Pere | 3   | 4  |
|  |   | 2.64E+08 | Omer Gab   | 4   | 5  |
|  |   | 5.89E+08 | Yosef Seg  | 5   | 6  |
|  |   | 5.71E+08 | Miriam Ma  | 6   | 7  |
|  |   | 6.75E+08 | Yonatan S  | 7   | 8  |
|  |   | 2.59E+08 | Omer Malk  | 8   | 9  |
|  |   | 2.31E+08 | Shira Tzac | 9   | 10 |
|  |   | 3.05E+08 | Natan Coh  | 10  | 11 |
|  |   | 9.34E+08 | Miriam Pe  | 11  | 12 |
|  |   | 9.45E+08 | Esther Ezn | 12  | 13 |
|  |   | 9.24E+08 | Eli Dayan  | 13  | 14 |
|  |   | 9.88E+08 | Natan Gab  | 14  | 15 |
|  |   | 6.73E+08 | Rachel Tza | 15  | 16 |
|  |   | 9.21E+08 | Avigail Ma | 16  | 17 |
|  |   | 2.8E+08  | Daniel Per | 17  | 18 |
|  |   | 5.1E+08  | Omer Tzac  | 18  | 19 |
|  |   | 3.9E+08  | Shira Seg  | 19  | 20 |
|  |   | 2.4E+08  | Omer Per   | 20  | 21 |

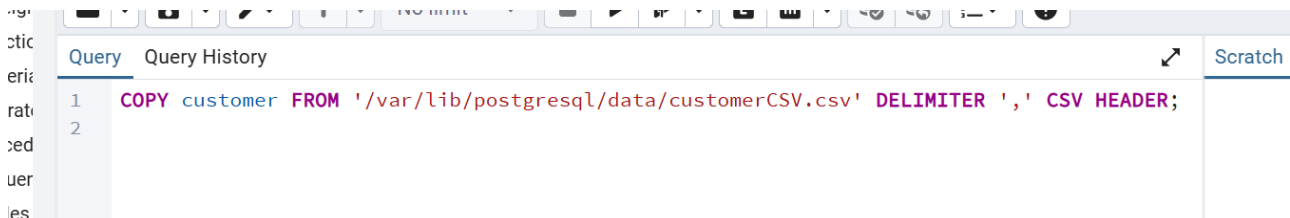
## שמירה כ- CSV



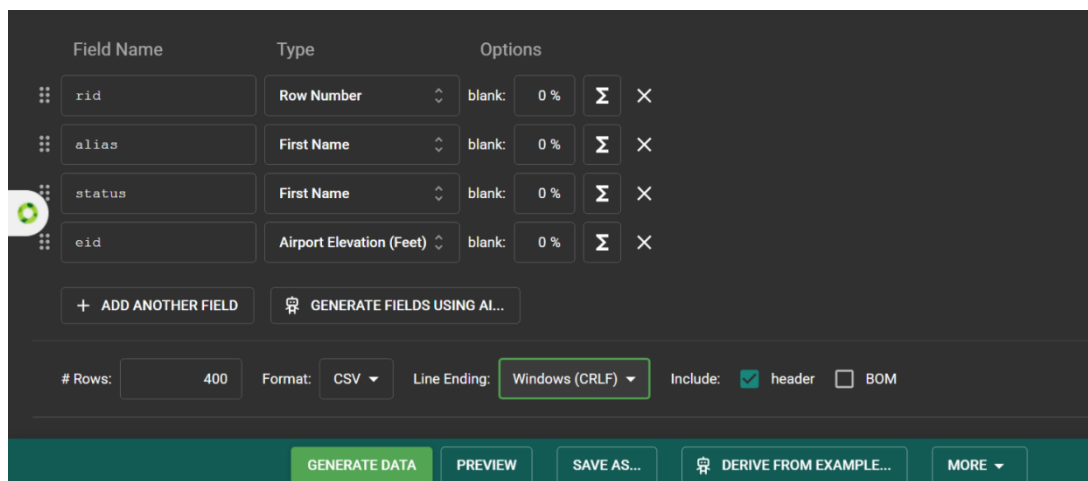
## ההלעת הקובץ ל- pgAdmin:

```
C:\Users\XXXXXX>COPY departmentCSV (depid, depname, aislenum, pid, eid) FROM 'C:\dbProject\stage1\excel\XXXX XXXX.csv' DELIMITER ',' CSV HEADER;
```

## שליפה בpgAdmin



## (3) יצירת נתונים באמצעות mockaroo

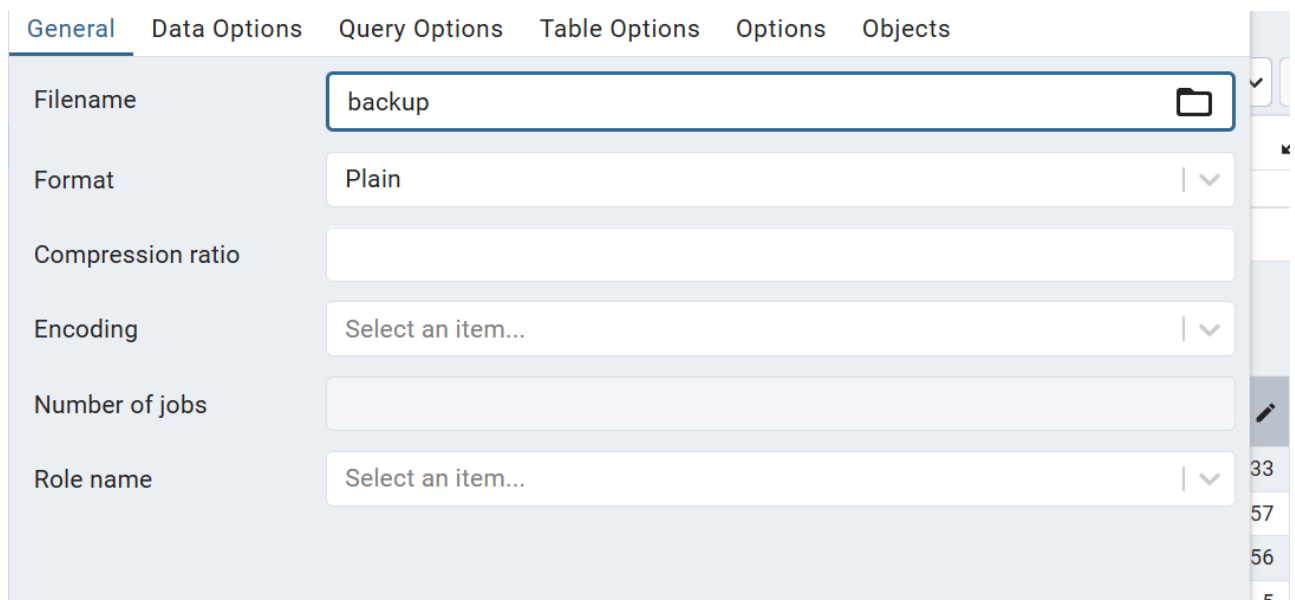
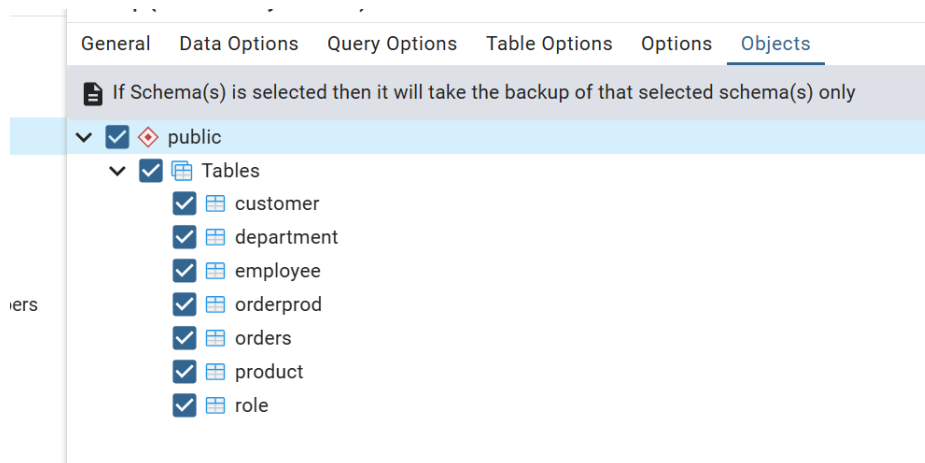
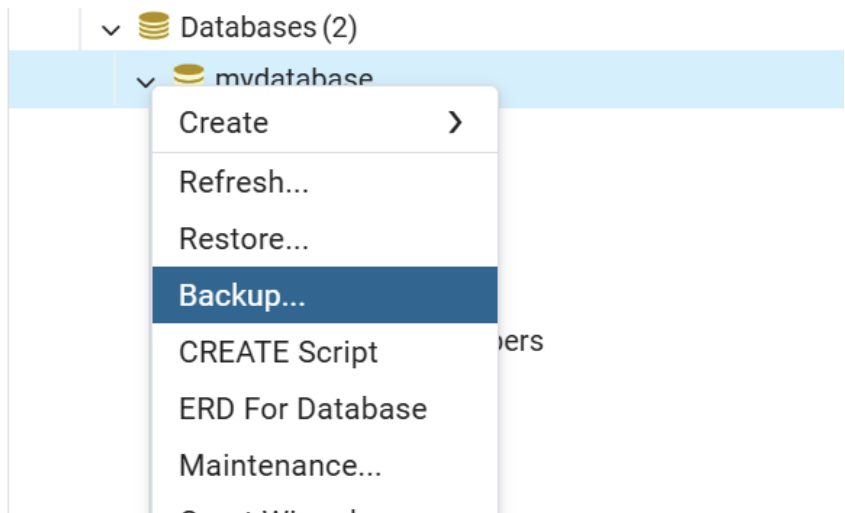


## הורדת קובץ ה-CSV

| F | E | D    | C         | B         | A   |    |
|---|---|------|-----------|-----------|-----|----|
|   |   | eid  | status    | alias     | rid | 1  |
|   |   | 300  | Theobald  | Steven    | 1   | 2  |
|   |   | 20   | Tulley    | Zoe       | 2   | 3  |
|   |   | 38   | Wylie     | Ariela    | 3   | 4  |
|   |   | 1633 | Kimberley | Hendrick  | 4   | 5  |
|   |   | 3347 | Cloris    | Persis    | 5   | 6  |
|   |   | 215  | Maurizia  | Amelita   | 6   | 7  |
|   |   | 394  | Odilia    | Rollin    | 7   | 8  |
|   |   | 13   | Gualterio | Maxine    | 8   | 9  |
|   |   | 1477 | Nissa     | Alisander | 9   | 10 |
|   |   | 12   | Beulah    | Abigale   | 10  | 11 |
|   |   | 72   | Millard   | Annmaria  | 11  | 12 |
|   |   | 7    | Valentijn | Dicky     | 12  | 13 |
|   |   | 914  | Patrick   | Issiah    | 13  | 14 |
|   |   | 23   | Kassi     | Jamey     | 14  | 15 |
|   |   | 8365 | Talbert   | Deborah   | 15  | 16 |
|   |   | 203  | Gaultiero | Shelia    | 16  | 17 |
|   |   | 900  | Onfroi    | Saleem    | 17  | 18 |
|   |   | 20   | Bent      | Dedie     | 18  | 19 |
|   |   | 1911 | Nicholle  | Sigfried  | 19  | 20 |

ההעלאה של קבצי ה-CSV כנ"ל בשורת הפקודה כמו בשיטה השנייה של קבצי האקסל לאחר שהועברו ל-CSV.

# גיבוי נתונים



|           | PID | Type          | Server                  | Object     | Start Time          | Status   | Time Taken (s) |
|-----------|-----|---------------|-------------------------|------------|---------------------|----------|----------------|
| ses (2)   | 927 | Backup Object | PostgreSQL Server (...) | mydatabase | 24.3.2025, 18:29:38 | Finished | 0.49           |
| atabase   | 274 | Backup Object | PostgreSQL Server (...) | mydatabase | 24.3.2025, 10:21:39 | Finished | 0.58           |
| asts      | 220 | Backup Object | PostgreSQL Server (...) | mydatabase | 24.3.2025, 9:49:56  | Finished | 0.09           |
| atalogs   |     |               |                         |            |                     |          |                |
| ent Trigg |     |               |                         |            |                     |          |                |
| tensions  |     |               |                         |            |                     |          |                |

C:\Users\srhm1>docker cp <my\_postgres>:/var/lib/pgadmin/storage/<cotehila>/<backup>.backup C:\Users\srhm1\Downloads\

```

קובץ עריכה עיצוב תצוגה עזרה
PGDMP
# 17.4 (Debian 17.4-1.pgdg120+2) 17.4 mydatabase
;SET client_encoding = 'UTF8' ENCODING ENCODING 0 0 0 0
a 0 0 false 0 0 0 0 0 0 0 0
0 0 0
STDSTRINGS
; ( SET standard_conforming_strings = 'on' STDSTRINGS
b 0 0 false 0 0 0 0 0 0 0 0
0 0 0
SEARCHPATH
;SELECT pg_catalog.set_config('search_path', '', false) 8 SEARCHPATH
c 0 0 false 0 0 0 0 0 0 0 0
16384 1262
CREATE DATABASE mydatabase WITH TEMPLATE = template0 ENCODING = 'UTF8' LOCALE_PROVIDER = libc LOCALE = 'en_US.utf8' DATABASE mydatabase
;DROP DATABASE mydatabase
; CREATE SCHEMA public SCHEMA public 2200 2615 0 0 0 0 false cotehila 0 0 0 0 0
;DROP SCHEMA public
d 0 0 false pg_database_owner 0 0 0 0 0 0
0 0 0
;COMMENT ON SCHEMA public IS 'standard public schema' 60 COMMENT SCHEMA public
CREATE TABLE public.customer TABLE customer 16426 12580 0 0 0 0 false pg_database_owner 0 0 0 0 0

```

# שחזור נתונים

העברת הקובץ לקונטיינר:

```

C:\Users\srhm1>docker exec -it 7494465e303d bash
root@7494465e303d:/# ls /backup/
ls: cannot access '/backup/': No such file or directory
root@7494465e303d:/# mkdir /backup/
exit
exit

C:\Users\srhm1>docker cp "C:\Users\srhm1\Downloads\backup_24.3.25.backup" 7494465e303d:/backup/
Successfully copied 39.9kB to 7494465e303d:/backup/

C:\Users\srhm1>|

```

## שליפה בpgAdmin

mydatabase/cotehila@PostgreSQL Server

Query Query History

```
1 COPY * FROM '/var/lib/postgresql/data/' DELIMITER ',' CSV HEADER;
2
```

FTS Te  
Foreign  
Func  
Materi  
Operat  
Proced  
Sequer  
Tables  
cust

## הצגת הנתונים בpgAdmin

pgAdmin File Object Tools Help

Object Explorer

- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (7)
  - customer
    - department
    - employee
    - orderprod
    - orders
    - product
    - role
  - Trigger Functions
  - Types
  - Views
- Subscriptions
- postgres
  - Casts
  - Catalogs

Dashboard Properties SQL Statistics Dependencies Dependents Processes public.customer/mydatabase/smansur@postgresql1

Query Query History

```
1 SELECT * FROM public.customer
2 ORDER BY cid ASC
```

Scratch Pad

Data Output Messages Notifications

Showing rows: 1 to 400 Page No: 1 of 1

|    | cid<br>[PK] numeric (5) | cname<br>character varying (15) | cphone<br>integer | ordid<br>numeric (15) |
|----|-------------------------|---------------------------------|-------------------|-----------------------|
| 1  | 1                       | Itay Levi                       | 641820540         | 257                   |
| 2  | 2                       | Yosef Levi                      | 341248848         | 212                   |
| 3  | 3                       | Shira Peretz                    | 733618619         | 164                   |
| 4  | 4                       | Omer Gabbay                     | 817204040         | 352                   |
| 5  | 5                       | Yosef Segal                     | 776966980         | 249                   |
| 6  | 6                       | Yonatan Shimon                  | 218669717         | 280                   |
| 7  | 7                       | Miriam Malka                    | 532097065         | 251                   |
| 8  | 8                       | Omer Malka                      | 738187948         | 161                   |
| 9  | 9                       | Shira Tzadok                    | 720252868         | 238                   |
| 10 | 10                      | Natan Cohen                     | 156683979         | 230                   |

Total rows: 400 Query complete 00:00:00.211 CRLF Ln 1, Col 1