

# **מיני פרויקט בבסיסי נתונים בית דפוס**

**מגישות**

**רינת ארנפרוינד 213527823**

**זהבי פרלא 326381480**

## תוכן עניינים:

2	-----	תוכן עניינים
3	-----	תיאור בית הדפוס
4	-----	תרשים ERD
4	-----	תרשים DSD
5	-----	פקודות יצירת הטבלאות SQL
6	-----	יצירת טבלאות ופקודת DESC
7-9	-----	הכנסת נתונים
10	-----	גיבוי ושחזור
11-15	-----	שאלות
16-17	-----	שאלות עם פרמטרים
18	-----	אילוצים
19-22	-----	פרוצדורות ופונקציות

## תיאור בית הדפוס:

בפרויקט שלנו, יצרנו מערכת לניהול בית דפוס שמתארת את הקשרים בין הלקוחות, העובדים, ההזמנות, התשלומים, הקטלוג והמלאי. בנינו ישויות וקשרים שונים בכדי לתאר את בית הדפוס ואת פעילותו בצורה מלאה ויעילה:

### לקוחות-

ת"ז שם וטלפון, עבור כל לקוח יש פירוט האם הוא פרטי, עסקי, בית ספר וכדו'

### עובדים-

ת"ז שם מייל וטלפון, עבור כל עובד יש פירוט של התפקיד שלו והשכר החודשי שלו. כמו"כ המשמרת בה העובד עובד (בוקר, צהריים, ערב).

### הזמנות-

תאריך הזמנה, מס' הזמנה וכמות המוצרים המוזמנים. מידע על זמן האספקה- הזמן שבו יש לספק את ההזמנה. מידע על כל המעורבים בהזמנה- ת"ז של הלקוח, ת"ז של העובד האחראי על ההזמנה, מספר קטלוגי של המוצר ומספר מלאי.

### תשלומים-

מספר תשלום, תאריך תשלום- התאריך בו נעשה התשלום. מספר תשלומים. סטטוס: הסטטוס של התשלום (שולם או לא שולם). צורת תשלום: הצורה בה נעשה התשלום (כרטיס אשראי, העברה בנקאית וכדומה). מספר זהות לקוח: מזהה הלקוח שביצע את התשלום.

### קטלוג-

שם, מספר קטלוגי, ומחיר עבור יחידה.

### מלאי-

מזהה המלאי, המדף בו הוא נמצא, מס' קטלוגי וכמות בסטוק.

### טיפול בהזמנות:

העובדים מטפלים בהזמנות. כל עובד יכול לטפל בהרבה הזמנות, אבל כל הזמנה מנוהלת על ידי עובד אחד בלבד.

### הזמנת מוצר:

ההזמנות כוללות מוצרים מהקטלוג. כל מוצר יכול להיות בהזמנות רבות, אבל כל הזמנה מתייחסת למוצר אחד בלבד.

### פרטי מלאי:

הקטלוג והמלאי מקושרים ביניהם. כל פריט במלאי יכול להיות שייך לכמה קטלוגים, אבל כל פריט קטלוגי מופיע במלאי פעם אחת.

### עדכון מלאי:

ההזמנות מעדכנות את המלאי. כל הזמנה יכולה לעדכן כמה פריטים במלאי, אבל כל פריט במלאי יכול להיות מעודכן על ידי הזמנות רבות.

### ניהול הזמנה:

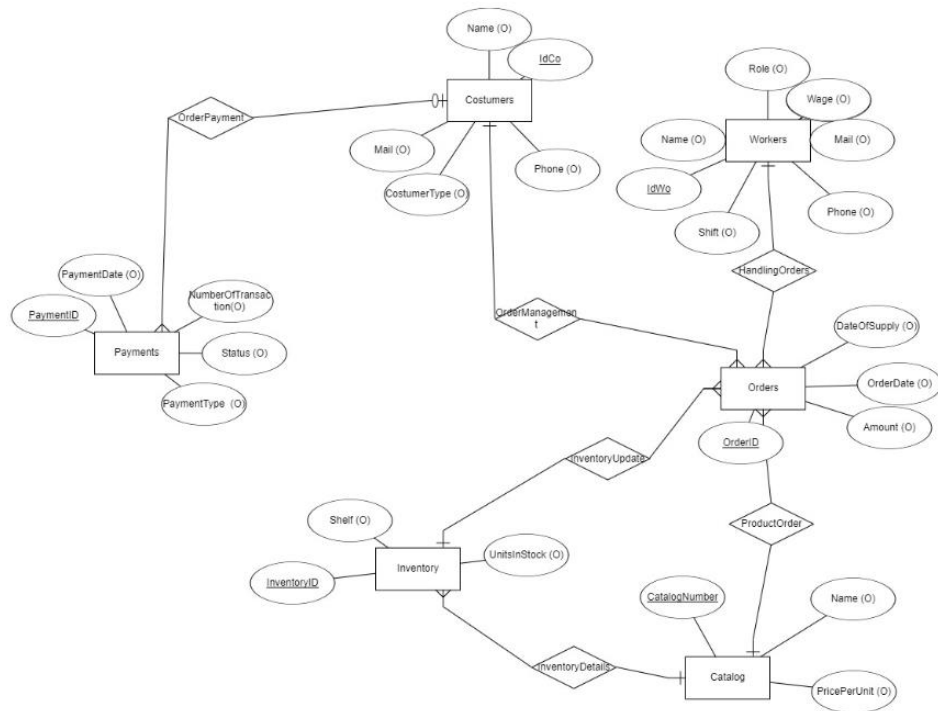
הלקוחות מבצעים הזמנות. כל לקוח יכול לבצע הרבה הזמנות, אבל כל הזמנה שייכת ללקוח אחד בלבד.

### תשלום הזמנה:

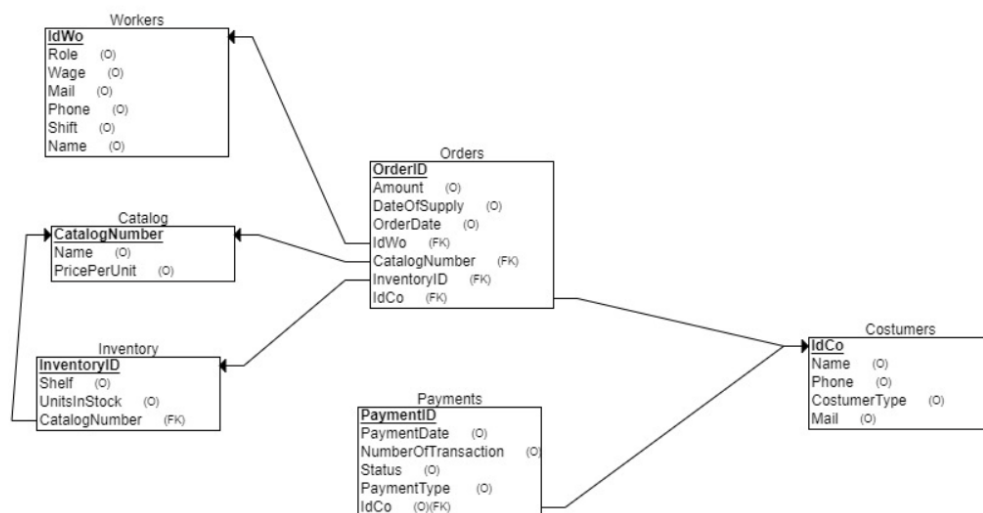
התשלומים קשורים ללקוחות. כל תשלום נעשה על ידי לקוח אחד, אבל כל לקוח יכול לבצע הרבה תשלומים.

בפרויקט שלנו, יצרנו מערכת מורכבת שמאפשרת ניהול מדויק ויעיל של כל הישויות השונות בבית הדפוס. המערכת כוללת מעקב אחרי לקוחות, ניהול עובדים, רישום הזמנות ותשלומים, ותחזוקת קטלוג ומלאי, כל זאת בכדי להבטיח שהעסק פועל בצורה חלקה ויעילה.

## תרשים ERD:



## תרשים DSD:



## פקודת יצירת טבלאות בSQL:

```

CREATE TABLE Workers
(
    WorkersRole VARCHAR(50),
    Wage NUMERIC,
    WorkersMail VARCHAR(50),
    WorkersPhone NUMERIC,
    Shift VARCHAR(50),
    IdWo NUMERIC NOT NULL,
    WorkersName VARCHAR(50),
    PRIMARY KEY (IdWo)
);

CREATE TABLE Customers
(
    CustomersName VARCHAR(50),
    IdCo NUMERIC NOT NULL,
    CustomersPhone NUMERIC,
    CustomersType VARCHAR(50),
    CustomersMail VARCHAR(50),
    PRIMARY KEY (IdCo)
);

CREATE TABLE NewCatalog
(
    NewCatalogNumber NUMERIC NOT NULL,
    NewCatalogName VARCHAR(50),
    PricePerUnit NUMERIC,
    PRIMARY KEY (NewCatalogNumber)
);

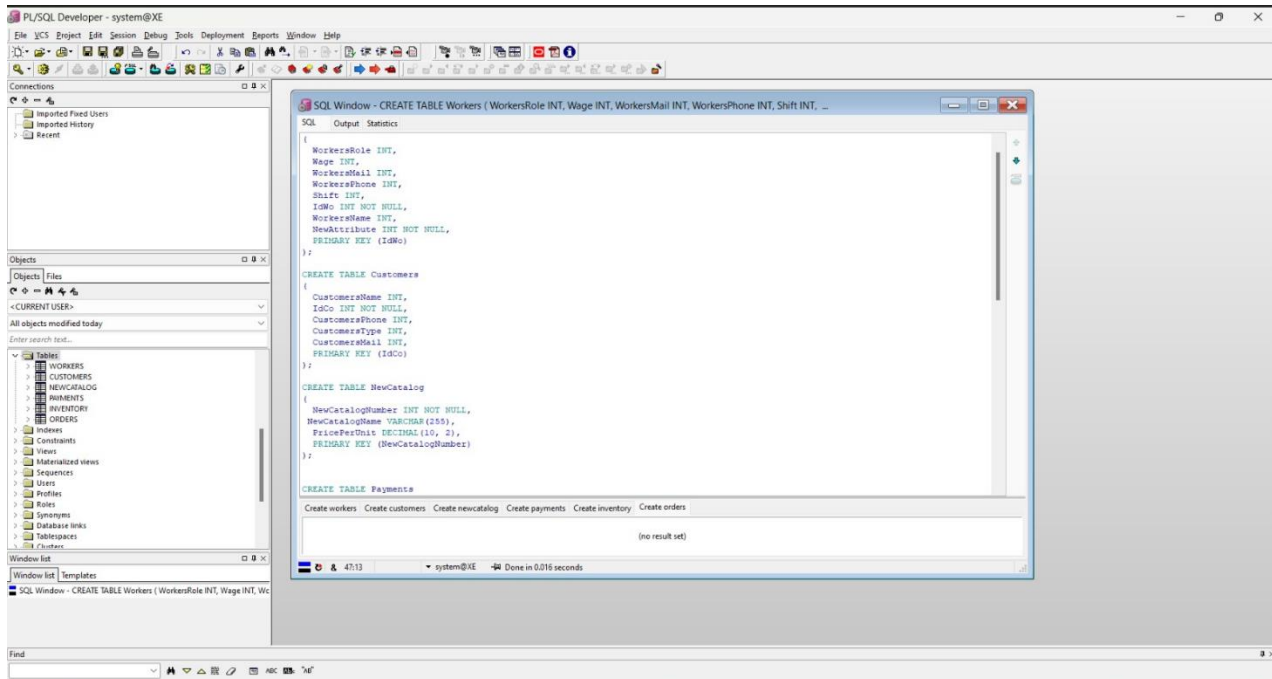
CREATE TABLE Payments
(
    PaymentDate DATE,
    NumberOfTransaction NUMERIC,
    Status VARCHAR(50),
    PaymentType VARCHAR(50),
    PaymentID NUMERIC NOT NULL,
    IdCo NUMERIC,
    PRIMARY KEY (PaymentID),
    FOREIGN KEY (IdCo) REFERENCES Customers(IdCo)
);

CREATE TABLE Inventory
(
    Shelf NUMERIC,
    UnitsInStock NUMERIC,
    InventoryID NUMERIC NOT NULL,
    CatalogNumber NUMERIC NOT NULL,
    PRIMARY KEY (InventoryID),
    FOREIGN KEY (CatalogNumber) REFERENCES NewCatalog(NewCatalogNumber)
);

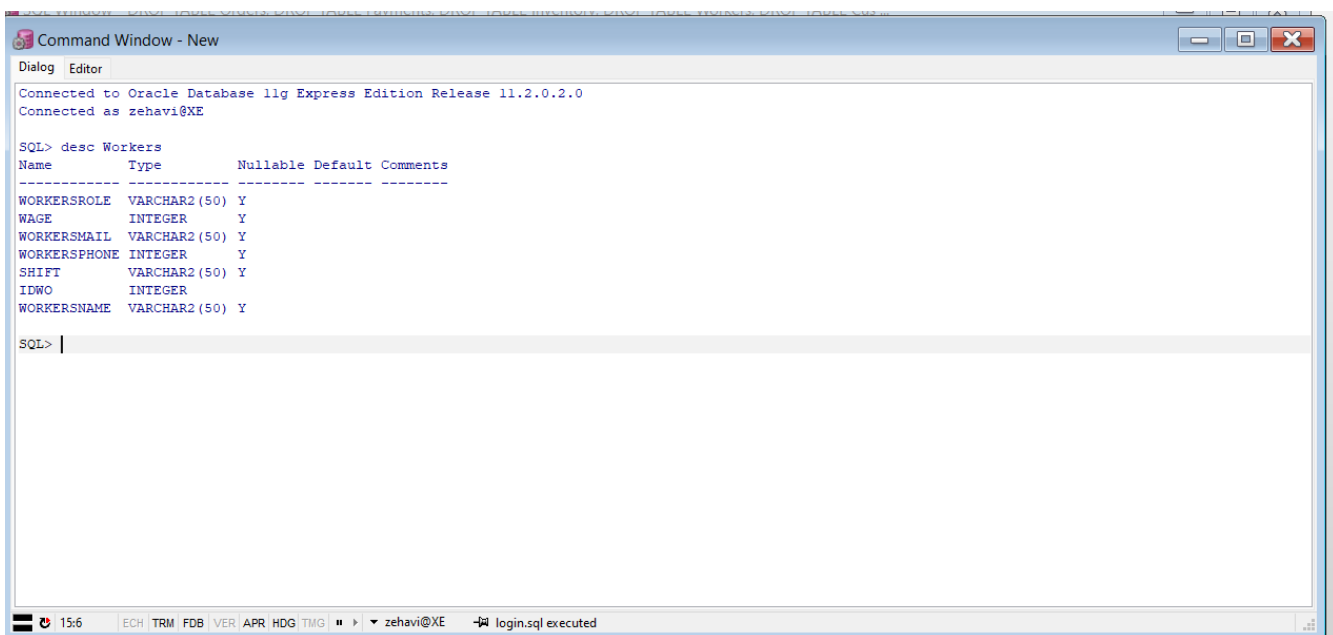
CREATE TABLE Orders
(
    Amount NUMERIC,
    DateOfSupply DATE,
    OrderDate DATE,
    OrderID NUMERIC NOT NULL,
    IdWo NUMERIC NOT NULL,
    CatalogNumber NUMERIC NOT NULL,
    InventoryID NUMERIC NOT NULL,
    IdCo NUMERIC NOT NULL,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (IdWo) REFERENCES Workers(IdWo),
    FOREIGN KEY (CatalogNumber) REFERENCES NewCatalog(NewCatalogNumber),
    FOREIGN KEY (InventoryID) REFERENCES Inventory(InventoryID),
    FOREIGN KEY (IdCo) REFERENCES Customers(IdCo)
);

```

## יצירת הטבלאות:

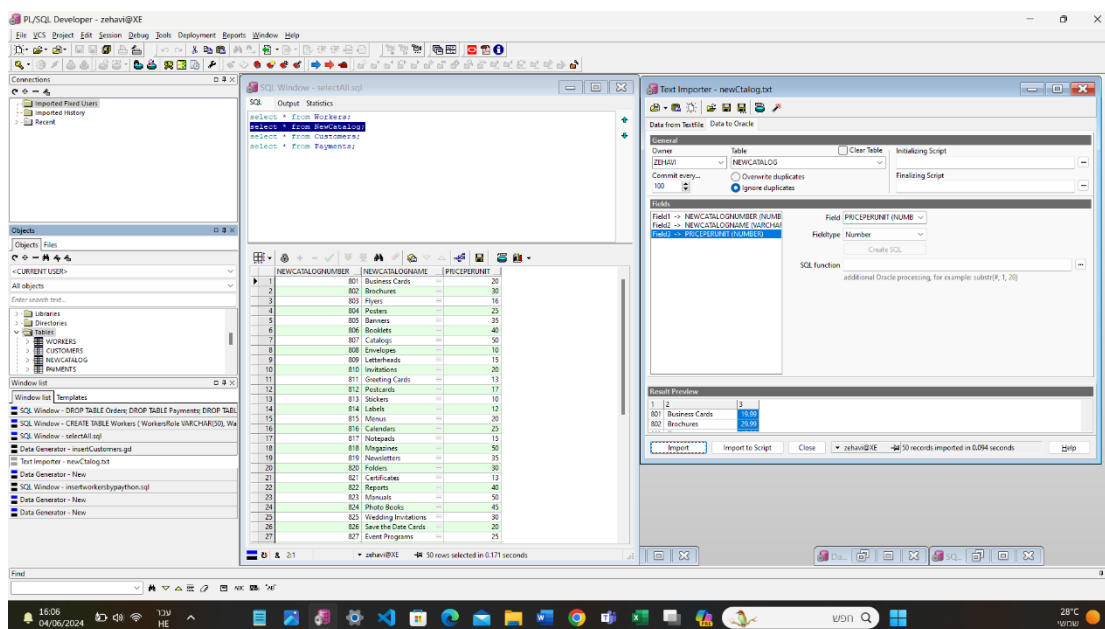


## פקודת DESC



## שיטות הכנסת הנתונים:

### הכנסה מקובץ טקסט:



## הכנסה בעזרת פייטון:

```

import random
import string

first_names = [
    "Noah", "Olivia", "Liam", "Emma", "Oliver", "Ava", "Elijah", "Sophia",
    "William", "Isabella", "James", "Charlotte", "Benjamin", "Mia", "Lucas",
    "Evelyn", "Mason", "Abigail", "Ethan", "Harper", "Aaliyah", "Alexander",
    "Amelia", "Jackson", "Ava", "Emily", "Avah", "Mason", "Olivia", "Sophia",
    "Liam", "Isabella", "Noah", "William", "James", "Benjamin", "Charlotte",
    "Elijah", "Mia", "Lucas", "Evelyn", "Mason", "Abigail", "Ethan", "Harper",
    "Avery", "Logan", "Ella", "Sofia", "Matthew", "Evelyn", "Aaliyah", "Alexander",
    "Amelia", "Emily", "Elizabeth", "Michael", "Avery", "Charlotte", "Avah",
    "Sophia", "Liam", "Ava", "Noah", "William", "James", "Benjamin", "Amelia",
    "Mason", "Evelyn", "Elijah", "Lucas", "Olivia", "Ethan", "Harper", "Avery",
    "Logan", "Ella", "Luna", "David", "Layla", "Camila", "Mateo", "Mia", "Evelyn",
    "Penelope", "Henry", "Riley", "Charlotte", "Aurora", "Levi", "Zoe", "Violet",
    "Elias", "Stella", "Scarlett", "Julian", "Maverick", "Claire"
]

last_names = [
    "Smith", "Johnson", "Williams", "Brown", "Jones", "Garcia", "Miller", "Davis",
    "Rodriguez", "Wilson", "Moore", "Clark", "Lewis", "Robinson", "Walker",
    "Allen", "Young", "Hall", "Wood", "Anderson", "Thomas", "Wright", "Hernandez",
    "King", "Scott", "Nelson", "Robinson", "Carter", "Campbell", "Adams", "Mitchell",
    "Bell", "Roberts", "Turner", "Phillips", "Green", "Baker", "Harris", "Lewis",
    "Lopez", "Martin", "Sanchez", "Clark", "Young", "Lee", "Allen", "Walker",
    "King", "Robinson", "Wright", "Scott", "Pierce", "Miller", "Diaz", "Cruz",
    "Murphy", "Harris", "Lewis", "Clark", "Hall", "Moore", "Robinson", "Garcia",
    "Rodriguez", "Anderson", "Thomas", "Jackson", "Campbell", "Young", "Allen",
    "Wright", "Diaz", "Walker", "Moore", "Robinson", "Nelson", "Carter", "Turner",
    "Garcia", "Johnson", "Davis", "Miller", "Rodriguez", "Lopez", "Hernandez", "Gonzalez",
    "Harris", "Clark", "Lewis", "Robinson", "Walker", "Allen", "Young", "King",
    "Wright", "Scott", "Pierce", "Moore", "Diaz", "Cruz", "Hall", "Bell", "Nelson",
    "Carter", "Campbell"
]

def generate_name():
    index1 = random.randint(0,98)
    first_name = first_names[index1]
    index2 = random.randint(0,98)
    last_name = last_names[index2]
    return first_name + " " + last_name

def generate_email():
    suffix = "@gmail.com"
    chars = string.ascii_letters + string.digits #array of all letters and numbers
    random_string = ''.join(random.choice(chars) for _ in range(8))
    return random_string + suffix

def generate_phone():
    # Fixed prefix (85)
    prefix = "05"
    # Generate remaining digits (8 random digits between 0 and 9)
    digits = ''.join(str(random.randint(0, 9)) for _ in range(8))
    # Combine prefix and digits
    return prefix + digits

# Define ranges for random values
workers_roles = ["Press Operator", "Prepress Technician", "Graphic Designer", "Customer Service Representative"]
wages_range = (7500, 15000)
shifts = ["Morning", "Afternoon", "Evening"]

# Generate 400 random records
for i in range(400):
    # Choose a random worker role
    workers_role = random.choice(workers_roles)

    # Generate a random wage
    wage = random.randint(*wages_range)

    # Generate a random email address
    workers_email = generate_email()

    # Generate a random phone number
    workers_phone = generate_phone()

    # Choose a random shift
    shift = random.choice(shifts)

    # Generate a random worker name
    workers_name = generate_name()

    # Create INSERT statement
    insert_query = f"INSERT INTO Workers (WorkersRole, Wage, WorkersMail, WorkersPhone, Shift, Idno, WorkersName) VALUES ({workers_role}, {wage}, '{workers_email}', {workers_phone}, '{shift}', {i+1}, '{workers_name}');"

    # Print INSERT statement
    print(insert_query)

```



SQL Window - insertworkersbypython.sql

```

SQL      Output  Statistics
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Prepress Technician', 8933, 'GAAD4Vv@gmail.com', 0526956196, 'Afternoon', 1,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Prepress Technician', 8919, '6nH7yav@gmail.com', 0546597191, 'Evening', 2,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Prepress Technician', 7696, 'g1BAUUI@gmail.com', 0531149209, 'Morning', 3,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 11895, 'YKSLDm@gmail.com', 0530481894, 'Afternoon', 4,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Customer Service Representative', 8712, 'lNYITP6@gmail.com', 0536732435, 'Ev
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Prepress Technician', 14350, 'o1ZD76k@gmail.com', 0571796735, 'Morning', 6,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Prepress Technician', 14530, 'e1alwd@gmail.com', 0538794510, 'Evening', 7,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Customer Service Representative', 10118, 'De4gtp5@gmail.com', 0569391226, 'E
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 10279, 'k4XJmNg@gmail.com', 0572740432, 'Morning', 9, 'A
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Prepress Technician', 12529, 'EadgOUI@gmail.com', 0576945371, 'Morning', 10,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 9180, 'WdusU08@gmail.com', 0536342312, 'Evening', 11, 'Oliv
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 9575, 'UHRBwkh@gmail.com', 0505264034, 'Evening', 12, 'Harp
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 10727, 'Aqqlc1H@gmail.com', 0595522957, 'Morning', 13, 'Men
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 14402, 'Y0ufmg@gmail.com', 0551200675, 'Afternoon', 14, 'B
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Prepress Technician', 8113, 'lgbDeK9@gmail.com', 0533586434, 'Afternoon', 15
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Prepress Technician', 8609, 'TqQIA8@gmail.com', 0552337492, 'Morning', 16,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 12521, 'bvlkx3S@gmail.com', 0514103974, 'Afternoon', 17,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 8437, 'Ie10Cq@gmail.com', 0529814090, 'Evening', 18, 'A
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 10277, 'YE1J0PX@gmail.com', 0508291301, 'Afternoon', 19,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Customer Service Representative', 10323, 'Y0PXA19@gmail.com', 0572240249, 'A
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 12730, 'YQ1U0DH@gmail.com', 0503384039, 'Morning', 20,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 9788, '89CFDL8I@gmail.com', 0533375134, 'Afternoon', 21,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 9346, 'A8t0k4K@gmail.com', 0597321717, 'Afternoon', 24,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 10977, 'ac2QWq@gmail.com', 0587834641, 'Evening', 23,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 7695, 'AiX7IyI@gmail.com', 0545811103, 'Evening', 26, 'Amel
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 12888, 'k1o5Rz@gmail.com', 0551157312, 'Afternoon', 27,

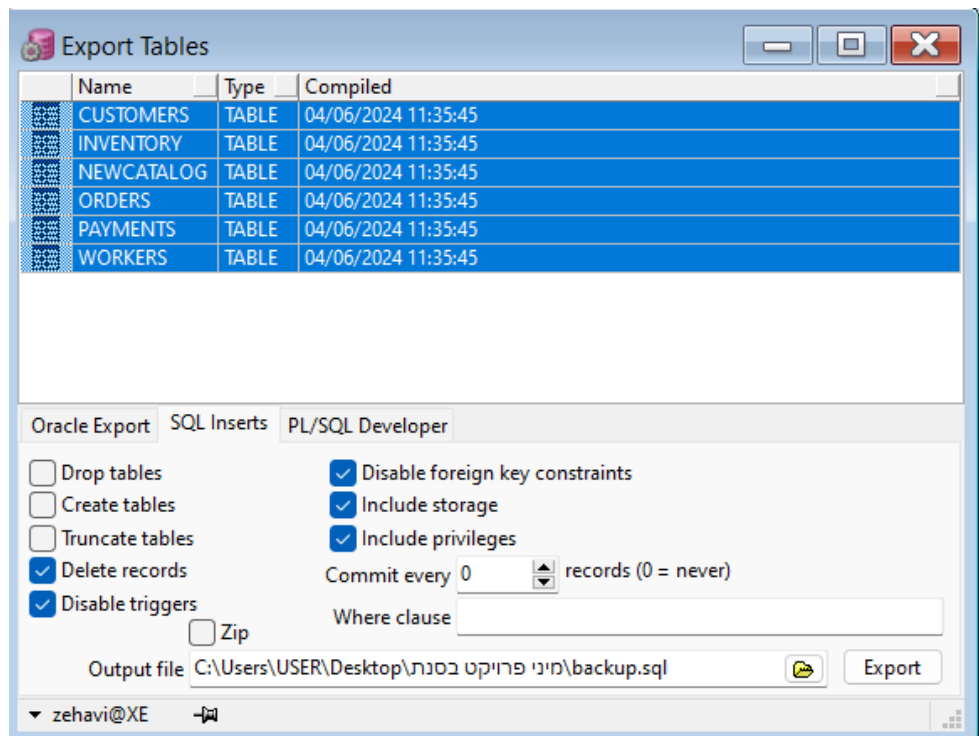
```

## הכנסה באמצעות DATAGENERATOR

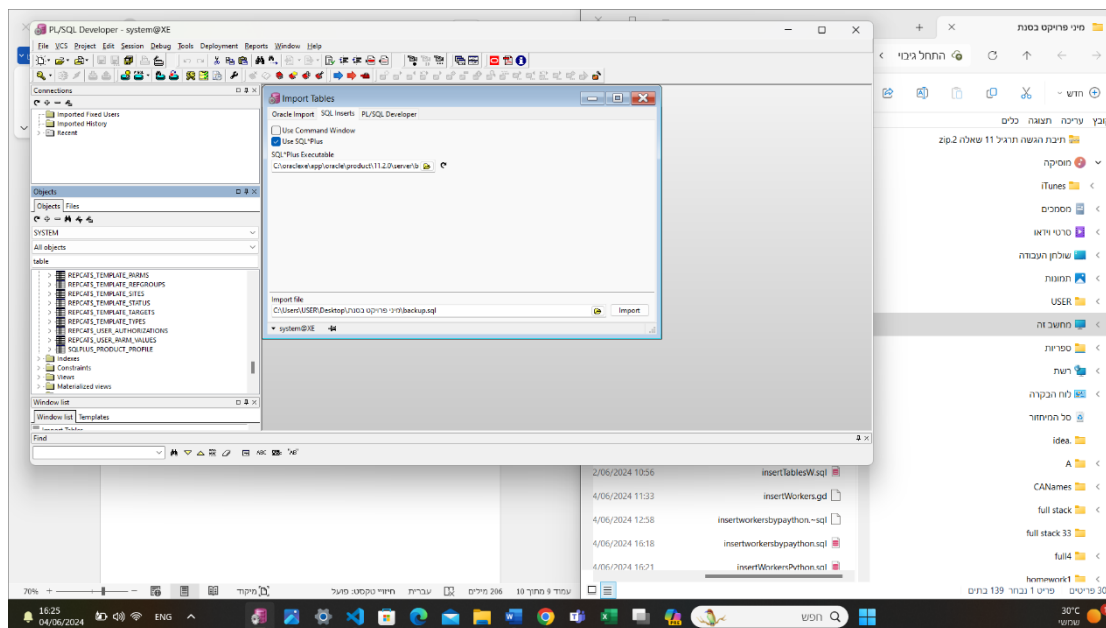
Data Generator - insertCustomers.gd

CUSTOMERSNAME	IDNO	CUSTOMERSPHONE	CUSTOMERSTYPE	CUSTOMERSMAIL
Rebecca-Vassar	401	53837886	school	rebecca.v@accesssystems.com
Howard-Saucedo	402	54954069	school	howard.saucedo@spimakersexploration.ca
Fisher-Warden	403	51038350	office	fisher.w@trafficmanagement.uk
Drew-McIntyre	404	58289789	private	d.mcintyre@gillette.de
Laure-Gordon	405	52511478	private	laure.gordon@kroger.ch
Sylvester-Roberts	406	55455478	private	sylvester.roberts@instruments.com
Dan-Hubbard	407	568179128	school	dan.hubbard@telaware.com
Celia-Makeba	408	57246399	private	cmakeba@ris.com
Rich-Alston	409	59884634	school	ralston@safesecurity.at
Madeline-Heslov	410	50721930	school	madeline.h@emergencyvehicles.com
Rade-Aida	411	54053729	office	rade.aida@hudsonwebcorp.ca
Ozzy-Giamatti	412	52302455	private	ozzy.giamatti@onesourceprinting.se
Richie-Day	413	51375646	office	richie.d@prosperitybancshares.de
Samantha-Michaels	414	55627354	private	samantha.michaels@meritagtechnologies.nl
Ellen-Roy Parnell	415	51392942	office	ellen@mms.com
Melba-Keaton	416	57655842	school	melba.keaton@stmaryland.at
Leon-Suchet	417	55341919	private	leon.suchet@pscifgroup.in
Arnold-Gjeld	418	54457956	private	arnold.gjeld@progressivedesigns.uk
George-Gracie	419	56221632	private	george.g@atlanticnet.com
Jimmie-Mandrell	420	58109068	school	jimmie.mandrell@bps.sg
Lois-Benson	421	55279045	private	lois.benson@ghsystems.de
Bryan-Roberts	422	52946164	private	bryan@vspan.com

## גיבוי נתונים:



## שחזור נתונים:



## שאלות

### SELECT

1. השאלתה מתמקדת במציאת מידע על עובדים מסוימים מתוך טבלת עובדים. היא מחפשת את כל העובדים שעובדים במשמרת לילה ושיש להם משכורת גבוהה מ-10,000. לאחר שהיא מוצאת את העובדים האלה, היא מציגה את שמותיהם, כתובות הדוא"ל שלהם, מספרי הטלפון שלהם, ואת המשכורות שלהם. בנוסף, השאלתה ממיינת את העובדים שנמצאו לפי המשכורת שלהם, מהגבוה לנמוך.

```
SELECT WorkersName, WorkersMail, WorkersPhone, Wage
FROM Workers
WHERE Shift = 'Evening' AND Wage > 12000
ORDER BY Wage desc;
```

הרצה:

	WORKERSNAME	WORKERSMAIL	WORKERSPHONE	WAGE
1	Alexander Clark	wR9vUehQ@gmail.com	509569472	14993
2	Abigail Robinson	ouDk88o5@gmail.com	577958343	14983
3	James Robinson	iabiRyfk@gmail.com	511197573	14967
4	Evelyn Turner	cIAJN0Nh@gmail.com	534741037	14802
5	William Clark	yptSwt2n@gmail.com	553610071	14797

תוצאה:

2. השאלתה הזו מתמקדת במציאת מידע על לקוחות והזמנות שהם ביצעו מאז ה-1 בינואר 2023. היא מציגה את שמות הלקוחות, כמה הזמנות כל אחד מהם ביצע, ומהו תאריך האספקה הראשון של כל לקוח עבור ההזמנות שבוצעו בתקופה זו.

הרצה:

```
SELECT c.CustomersName, COUNT(o.OrderID) AS TotalOrders, MIN(o.DateOfSupply) AS FirstSupplyDate
FROM Orders o
JOIN Customers c ON o.IdCo = c.IdCo
WHERE o.OrderDate > TO_DATE('01/01/2023', 'DD/MM/YYYY')
GROUP BY c.CustomersName
ORDER BY TotalOrders DESC;
```

תוצאה:

1	Noah-Gilliam	3	09/12/2020
2	Tia-Frost	2	13/03/2022
3	Pelvic-Hawthorne	2	18/09/2020
4	Dianne-Dalton	2	19/02/2021
5	Ed-Mattea	2	14/06/2021

3. השאלתה מתמקדת במציאת הזמנות שסכומן גבוה מ-50 ומציגה פרטים רלוונטיים על כל הזמנה כולל מספר ההזמנה, הסכום שלה, ושמות הלקוח והעובד שקשורים אליה. היא מציגה את התוצאות ממיונות לפי סכום ההזמנה בסדר יורד, כך שההזמנות עם הסכומים הגבוהים ביותר יופיעו ראשונות.

הרצה:

```
SELECT o.OrderID, o.Amount, c.CustomersName, w.WorkersName
FROM Orders o
JOIN Customers c ON o.IdCo = c.IdCo
JOIN Workers w ON o.IdWo = w.IdWo
WHERE o.Amount > 50
ORDER BY o.Amount DESC, o.OrderID ASC;
```

	ORDERID	AMOUNT	CUSTOMERSNAME	WORKERSNAME
1	2094	2000	Mandy-Conlee	Camila Robinson
2	2257	1996	Rufus-Wilder	Levi Walker
3	2185	1995	Isaiah-Snow	Abigail Moore
4	2174	1994	Jeanne-Ojeda	Charlotte Hernandez
5	2117	1993	Edie-Hart	Benjamin Walker

תוצאה:

4. השאילתה מציגה פרטים על תשלומים שנעשו על ידי לקוחות מסוג "office". היא משתמשת בשאילתא מקוננת כדי לבחור את רשימת הלקוחות מסוג "office" כאשר השאילתה רצה, היא מקבצת את התוצאות לפי מזהה התשלום, תאריך התשלום, סוג התשלום ושם הלקוח.

הרצה:

```
SELECT p.PaymentID, p.PaymentDate, p.PaymentType, c.CustomersName
FROM Payments p
JOIN Customers c ON p.IdCo = c.IdCo
WHERE c.IdCo IN (
    SELECT IdCo
    FROM Customers
    WHERE CustomersType = 'office'
)
GROUP BY p.PaymentID, p.PaymentDate, p.PaymentType, c.CustomersName;
```

תוצאה:

	PAYMENTID	PAYMENTDATE	PAYMENTTYPE	CUSTOMERSNAME
1	1063	29/01/2015	cash	Ewan-Unger
2	1065	18/04/2020	cash	Hank-Wagner
3	1080	26/09/2020	cash	Tracy-Tinsley
4	1100	14/11/2006	CreditCard	Wang-Viterelli
5	1103	06/12/2015	cash	Lisa-Mellencamp

## DELETE

1. השאילתה מבוססת על פקודת DELETE ומטרתה למחוק הזמנות מטבלת "Orders" ששייכות ללקוחות מסוג "office" כדי לקבוע אילו הזמנות למחוק, השאילתה משתמשת בשאילתה מקוננת שמביאה את רשימת הלקוחות מסוג "office" בנוסף, השאילתה משתמשת בפקודת GROUP BY כדי לקבץ את התוצאות לפי לקוחות וסכומי ההזמנה שלהם. בכך, השאילתה מבטיחה שהזמנות יוחזרו וימחקו לפי קבוצות של לקוחות וסכומי ההזמנה שלהם.

הרצה:

```
DELETE FROM Orders
WHERE IdCo IN (
    SELECT IdCo
    FROM Customers
    WHERE CustomersType = 'office'
);
```

# תוצאה:

	AMOUNT	DATEOFSUPPLY	ORDERDATE	ORDERID	IDWO	CATALOGNUMBER	INVENTORYID	IDCO
1	1133	29/04/2020	07/11/2021	2000	251	831	2040	611
2	1009	05/10/2023	12/02/2022	2001	111	848	2238	616
3	1949	18/09/2020	07/08/2023	2002	43	807	2100	514
4	1702	04/11/2021	03/05/2022	2003	19	850	2326	451
5	1198	22/09/2023	20/01/2022	2004	223	829	2137	675
6	1778	01/10/2020	27/02/2023	2005	218	827	2098	421
7	1401	15/08/2023	18/09/2023	2006	215	820	2269	430
8	1401	02/02/2022	25/04/2023	2007	197	803	2210	779
9	1297	30/07/2022	05/11/2020	2008	358	827	2185	698
10	1451	26/06/2020	09/05/2021	2009	382	816	2353	674
11	1527	01/04/2020	17/09/2023	2010	199	824	2398	777
12	1744	25/10/2022	18/07/2021	2011	31	831	2157	474
13	1415	19/08/2023	12/05/2020	2012	13	814	2146	557
14	1942	15/12/2022	18/06/2022	2013	82	810	2171	514

- תת-שאלתה בוחרת את כל ה-IdCo מהטבלה Customers של לקוחות מסוג private.
- השאלתה הראשית מוחקת מהטבלה Payments את כל הרשומות שבהן IdCo מופיע בתוצאות תת-השאלתה.

## הרצה:

```
DELETE FROM Payments
WHERE IdCo IN (
    SELECT IdCo
    FROM Customers
    WHERE CustomersType = 'private'
);
```

# תוצאה:

לפני המחיקה-

	PAYMENTDATE	NUMBEROFTRANSACTION	STATUS	PAYMENTTYPE	PAYMENTID	IDCO
1	26/11/2018	7	not paid	CreditCard	1000	534
2	11/03/2004	2	not paid	bank transfer	1001	485
3	07/08/2008	10	not paid	cash	1002	773
4	09/12/2020	2	Paid	CreditCard	1003	425
5	16/03/2017	8	Paid	bank transfer	1004	720
6	06/06/2018	9	not paid	cash	1005	654
7	24/12/2020	4	Paid	cash	1006	643
8	16/10/2005	7	not paid	CreditCard	1007	671
9	11/02/2012	1	not paid	bank transfer	1008	468
10	25/08/2006	1	not paid	bank transfer	1009	505
11	06/07/2023	5	Paid	CreditCard	1010	745
12	22/03/2012	9	not paid	CreditCard	1011	653
13	02/06/2012	2	Paid	CreditCard	1012	782
14	14/07/2012	8	not paid	cash	1013	446
15	04/01/2011	9	Paid	CreditCard	1014	629
16	21/11/2014	2	Paid	bank transfer	1015	683

## לאחר המחיקה-

	PAYMENTDATE	NUMBEROFTRANSACTION	STATUS	PAYMENTTYPE	PAYMENTID	IDCO
1	07/08/2008	10	not paid	cash	1002	773
2	09/12/2020	2	Paid	CreditCard	1003	425
3	16/03/2017	8	Paid	bank transfer	1004	720
4	06/06/2018	9	not paid	cash	1005	654
5	16/10/2005	7	not paid	CreditCard	1007	671
6	11/02/2012	1	not paid	bank transfer	1008	468
7	06/07/2023	5	Paid	CreditCard	1010	745
8	22/03/2012	9	not paid	CreditCard	1011	653
9	14/07/2012	8	not paid	cash	1013	446
10	04/01/2011	9	Paid	CreditCard	1014	629
11	21/11/2014	2	Paid	bank transfer	1015	683
12	06/04/2009	10	Paid	CreditCard	1016	473
13	06/03/2008	7	Paid	bank transfer	1021	535
14	19/09/2023	2	not paid	CreditCard	1022	624
15	06/05/2016	2	not paid	bank transfer	1025	440
16	30/12/2022	8	Paid	bank transfer	1026	566
17	09/06/2012	6	not paid	bank transfer	1027	549
18	16/05/2008	7	Paid	bank transfer	1031	430
19	23/04/2006	5	Paid	CreditCard	1032	687
20	12/03/2015	7	Paid	cash	1034	550

## UPDATE

- עדכון המשכורת של כל העובדים שעובדים במשמרת בוקר ב-10%, תוך כדי סיכום המשכורות לפי תפקיד העובד  
הרצה:

```
UPDATE Workers
SET Wage = Wage * 1.1
WHERE Shift = 'Morning';
```

## תוצאה:

לפני העדכון-

	WORKERSROLE	WAGE	WORKERSMAIL	WORKERSPHONE	SHIFT	IDWO	WORKERSNAME
1	Prepress Technician	8933	GAAD4Gvz@gmail.com	526956196	Afternoon	1	Elias Martin
2	Prepress Technician	8919	6nH7yatv@gmail.com	546597191	Evening	2	Oliver Young
3	Prepress Technician	7696	gi7BAUJU@gmail.com	531149208	Morning	3	Jackson Murphy
4	Graphic Designer	11895	Y5kCLDmE@gmail.com	539048194	Afternoon	4	Elijah Robinson
5	Customer Service Representative	8712	1hYiTP6g@gmail.com	536732435	Evening	5	David Carter
6	Prepress Technician	14350	oiTZv76k@gmail.com	571796738	Morning	6	Claire Miller
7	Prepress Technician	14530	eZslowdS@gmail.com	535796510	Evening	7	Aurora Turner
8	Customer Service Representative	10118	Jm4gMp5R@gmail.com	569391226	Evening	8	Emma Wright
9	Graphic Designer	10279	x4XJknNp@gmail.com	572740432	Morning	9	Amelia Campbell
10	Prepress Technician	12829	EsdgtOYs@gmail.com	576945371	Morning	10	Sophia King
11	Press Operator	9180	WCduuOBr@gmail.com	536342312	Evening	11	Olivia Scott
12	Press Operator	9575	UHRBwxhc@gmail.com	509286034	Evening	12	Harper Young
13	Press Operator	10727	A9qO1tjN@gmail.com	595522957	Morning	13	Henry Lopez
14	Press Operator	14402	78OfnqK@gmail.com	591200672	Afternoon	14	Benjamin Hall
15	Prepress Technician	8113	1gDaAe9@gmail.com	583586434	Afternoon	15	William Jones
16	Prepress Technician	8609	TgXQrlAR@gmail.com	552337692	Morning	16	Sophia Anderson
17	Graphic Designer	12521	BwlXs3S0@gmail.com	514103974	Afternoon	17	Claire Miller
18	Press Operator	8627	lelOC6pp@gmail.com	529814090	Evening	18	Elizabeth Lee
19	Graphic Designer	10277	YE1JOPXg@gmail.com	508291301	Afternoon	19	Avah Lewis
20	Prepress Technician	8961	pgMgEsc5@gmail.com	504357854	Afternoon	20	Luna Phillips
21	Customer Service Representative	10323	PUFX199h@gmail.com	572260269	Afternoon	21	Violet Young
22	Graphic Designer	12730	PQrlU0DH@gmail.com	503386039	Morning	22	Avery Davis
23	Graphic Designer	9758	8BCFLbEl@gmail.com	533375134	Afternoon	23	Oliver Turner
24	Graphic Designer	9366	An40xdkZ@gmail.com	597321717	Afternoon	24	Aaliyah Carter
25	Graphic Designer	10977	m20jWq7@gmail.com	557836241	Evening	25	Harper Baker
26	Press Operator	7605	Δix7uln@gmail.com	565811105	Evening	26	Amelia Diaz



## אחרי העדכון-

	WORKERSROLE	WAGE	WORKERSMAIL	WORKERSPHONE	SHIFT	IDWO	WORKERSNAME
1	Prepress Technician	8933	GAAD4Gvz@gmail.com	526956196	Afternoon	1	Elias Martin
2	Prepress Technician	8919	6nH7yav@gmail.com	546597191	Evening	2	Oliver Young
3	Prepress Technician	8466	gi78AUU@gmail.com	531149208	Morning	3	Jackson Murphy
4	Graphic Designer	11895	Y5kCLdME@gmail.com	539048194	Afternoon	4	Elijah Robinson
5	Customer Service Representative	8712	1hYiTP6g@gmail.com	536732435	Evening	5	David Carter
6	Prepress Technician	15785	oiTZv76k@gmail.com	571796738	Morning	6	Claire Miller
7	Prepress Technician	14530	eZslowdS@gmail.com	535796510	Evening	7	Aurora Turner
8	Customer Service Representative	10118	Jm4gMp5R@gmail.com	568391226	Evening	8	Emma Wright
9	Graphic Designer	11307	x4XJknNp@gmail.com	572740432	Morning	9	Amelia Campbell
10	Prepress Technician	14112	EsdtOYs@gmail.com	576945371	Morning	10	Sophia King
11	Press Operator	9180	WCduuOBr@gmail.com	536342312	Evening	11	Olivia Scott
12	Press Operator	9575	UHRBwxc@gmail.com	509286034	Evening	12	Harper Young
13	Press Operator	11800	A9qO1tjN@gmail.com	595522957	Morning	13	Henry Lopez
14	Press Operator	14402	78OfnqK@gmail.com	591200672	Afternoon	14	Benjamin Hall
15	Prepress Technician	8113	1gDaAe9@gmail.com	583586434	Afternoon	15	William Jones
16	Prepress Technician	9470	TgXQrAR@gmail.com	552337692	Morning	16	Sophia Anderson
17	Graphic Designer	12521	BwXs3S0@gmail.com	514103974	Afternoon	17	Claire Miller
18	Press Operator	8627	lelOC6pp@gmail.com	529814090	Evening	18	Elizabeth Lee
19	Graphic Designer	10277	YE1JOPXg@gmail.com	508291301	Afternoon	19	Avah Lewis
20	Prepress Technician	8961	ppMgEsc5@gmail.com	504357854	Afternoon	20	Luna Phillips
21	Customer Service Representative	10323	PUFX199h@gmail.com	572260269	Afternoon	21	Violet Young
22	Graphic Designer	14003	PQrIUODH@gmail.com	503386039	Morning	22	Avery Davis
23	Graphic Designer	9758	8BCFLbEl@gmail.com	533375134	Afternoon	23	Oliver Turner
24	Graphic Designer	9366	An40xdKZ@gmail.com	597321717	Afternoon	24	Aaliyah Carter
25	Graphic Designer	10977	m20jWqa7@gmail.com	557836241	Evening	25	Harper Baker

2. מעדכן את מצב התשלום (Status) בטבלת Payments ל"Completed" עבור תשלומים שבוצעו על ידי לקוחות מסוג office.

הרצה:

```
UPDATE Payments
SET Status = 'paid'
WHERE IdCo IN (
    SELECT IdCo
    FROM Customers
    WHERE CustomersType = 'office'
);
```

תוצאה:

לפני העדכון-

	PAYMENTDATE	NUMBEROFTRANSACTION	STATUS	PAYMENTTYPE	PAYMENTID	IDCO
1	07/08/2008	10	not paid	cash	1002	773
2	09/12/2020	2	Paid	CreditCard	1003	425
3	16/03/2017	8	Paid	bank transfer	1004	720
4	06/06/2018	9	not paid	cash	1005	654
5	16/10/2005	7	not paid	CreditCard	1007	671
6	11/02/2012	1	not paid	bank transfer	1008	468
7	06/07/2023	5	Paid	CreditCard	1010	745
8	22/03/2012	9	not paid	CreditCard	1011	653
9	14/07/2012	8	not paid	cash	1013	446
10	04/01/2011	9	Paid	CreditCard	1014	629
11	21/11/2014	2	Paid	bank transfer	1015	683
12	06/04/2009	10	Paid	CreditCard	1016	473
13	06/03/2008	7	Paid	bank transfer	1021	535
14	19/09/2023	2	not paid	CreditCard	1022	624
15	06/05/2016	2	not paid	bank transfer	1025	440
16	30/12/2022	8	Paid	bank transfer	1026	566
17	09/06/2012	6	not paid	bank transfer	1027	549
18	16/05/2008	7	Paid	bank transfer	1031	430
19	23/04/2006	5	Paid	CreditCard	1032	687
20	13/03/2015	7	not paid	cash	1034	660

אחרי העדכון-

	PAYMENTDATE	NUMBEROFTRANSACTION	STATUS	PAYMENTTYPE	PAYMENTID	IDCO
1	07/08/2008	10	paid	cash	1002	773
2	09/12/2020	2	Paid	CreditCard	1003	425
3	16/03/2017	8	Paid	bank transfer	1004	720
4	06/06/2018	9	not paid	cash	1005	654
5	16/10/2005	7	not paid	CreditCard	1007	671
6	11/02/2012	1	not paid	bank transfer	1008	468
7	06/07/2023	5	Paid	CreditCard	1010	745
8	22/03/2012	9	not paid	CreditCard	1011	653
9	14/07/2012	8	paid	cash	1013	446
10	04/01/2011	9	Paid	CreditCard	1014	629
11	21/11/2014	2	paid	bank transfer	1015	683
12	06/04/2009	10	paid	CreditCard	1016	473
13	06/03/2008	7	Paid	bank transfer	1021	535
14	19/09/2023	2	not paid	CreditCard	1022	624

## שאלות עם פרמטרים

1. בשאלתה זו אנו מקבלים את פרטי התשלומים יחד עם פרטי הלקוחות עבור תשלומים שלא הושלמו בתאריך מסוים ומחזירים את הטבלה עבור נתונים אלו.  
הרצה:

```
SELECT
    Payments.PaymentID,
    Payments.PaymentDate,
    Payments.PaymentType,
    Payments.Status,
    Customers.CustomerName,
    Customers.CustomerMail
FROM
    Payments
JOIN
    Customers ON Payments.IdCo = Customers.IdCo
WHERE
    Payments.Status = 'not paid' and payments.paymentdate=<name="newDate" type="date">;
```

תוצאה:

	PAYMENTID	PAYMENTDATE	PAYMENTTYPE	STATUS	CUSTOMERSNAME	CUSTOMERSMAIL
1	1007	16/10/2005	CreditCard	not paid	Olympia-Washington	olympia.w@unilever.at

2. בשאלתה הזו אנו מקבלים את פרטי ההזמנות, פרטי המוצרים ופרטי העובדים שביצעו את ההזמנות.  
באפשרותנו לבחור סוג לקוח מסוים (לקוח פרטי, בית ספר, משרד) עבורו נרצה לראות את כל הפרטים, התנאי הוא שסכום הזמנת הלקוח יהיה מעל 1500 ש"ח.  
השאלתה מסודרת לפי תאריך ההזמנה בסדר יורד.

הרצה:

```
SELECT
    Orders.OrderID,
    Orders.OrderDate,
    Orders.Amount,
    Customers.CustomerName,
    Customers.CustomerPhone,
    Customers.CustomerType,
    NewCatalog.NewCatalogName,
    NewCatalog.PricePerUnit,
    Workers.WorkerName,
    Workers.WorkerRole
FROM
    Orders
JOIN
    Customers ON Orders.IdCo = Customers.IdCo
JOIN
    NewCatalog ON Orders.CatalogNumber = NewCatalog.NewCatalogNumber
JOIN
    Workers ON Orders.IdWo = Workers.IdWo
WHERE
    Customers.CustomerType = <name="customerType" list="private,office,school" type="string" restricted="yes">
    AND Orders.Amount > 1500
ORDER BY
    Orders.OrderDate DESC;
```



כאן ניתן לראות את האפשרות לבחירת סוג הלקוח.

תוצאה:

ORDERID	ORDERDATE	AMOUNT	CUSTOMERSNAME	CUSTOMERSPHONE	CUSTOMERSTYPE	NEWCATALOGNAME	PRICEPERUNIT	WORKERSNAME	WORKERSROLE	
1	2212	15/12/2023	1582	Nanci-Garber	520402382	private	Announcement Cards	17	Olivia Lee	Graphic Designer
2	2037	30/11/2023	1857	Timothy-Jamal	599154120	private	Folders	30	Elijah Robinson	Graphic Designer
3	2157	15/11/2023	1647	Janeane-Van Shelton	549177111	private	Booklets	40	Ethan Nelson	Customer Service Representative
4	2035	18/10/2023	1657	Remy-Starr	569457944	private	Magazines	50	Charlotte Robinson	Prepress Technician
5	2022	12/10/2023	1825	Armand-Quinones	523110384	private	Rack Cards	18	William Jones	Prepress Technician

3. בשאלתה זו אנו מקבלים את פרטי המלאי יחד עם פרטי המוצרים כאשר כמות היחידות במלאי היא בין 10-30, ומסדרים לפי המדף בו נמצאים המוצרים.

הרצה:

```
SELECT
    Inventory.InventoryID,
    Inventory.Shelf,
    Inventory.UnitsInStock,
    NewCatalog.NewCatalogName,
    NewCatalog.PricePerUnit
FROM
    Inventory
JOIN
    NewCatalog ON Inventory.CatalogNumber = NewCatalog.NewCatalogNumber
WHERE
    Inventory.UnitsInStock < &<name="units" hint="select a number between 10-30" type="integer" required="true">
ORDER BY
    Inventory.Shelf;
```

תוצאה:

	INVENTORYID	SHELF	UNITSINSTOCK	NEWCATALOGNAME	PRICEPERUNIT
1	2002	1	10	Notepads	15
2	2018	1	3	Brochures	30
3	2063	1	19	Banners	35
4	2083	1	4	Business Forms	15
5	2145	1	15	Thank You Cards	13

4. שאלתה אשר ממינת את הטבלה לפי אחת משלושת העמודות הבאות המייל של העובד, המשמרת שבה הוא עבד או התפקיד שלו

הרצה:

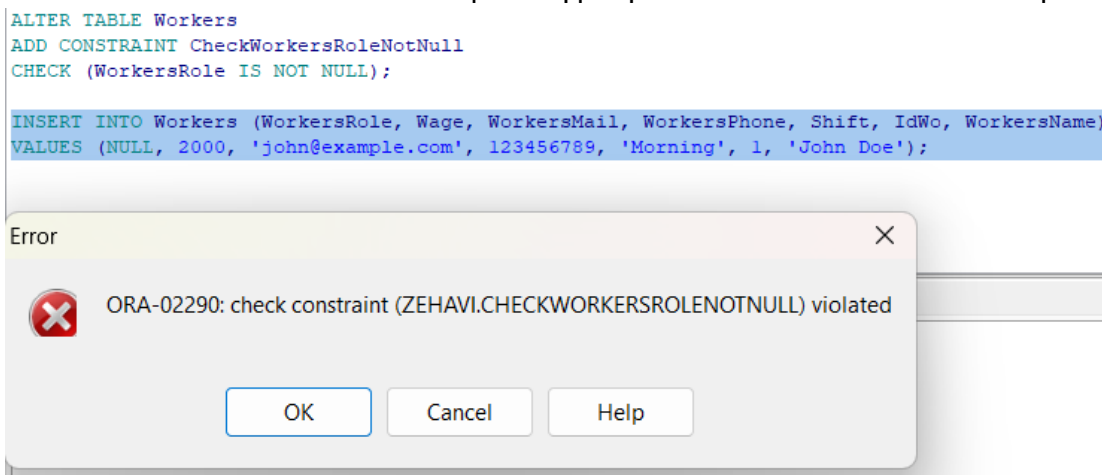
```
SELECT * FROM Workers
&<name="sorted" list="workersRole,workersMail,shift" prefix="order by " suffix=" desc">;|
```

תוצאה:

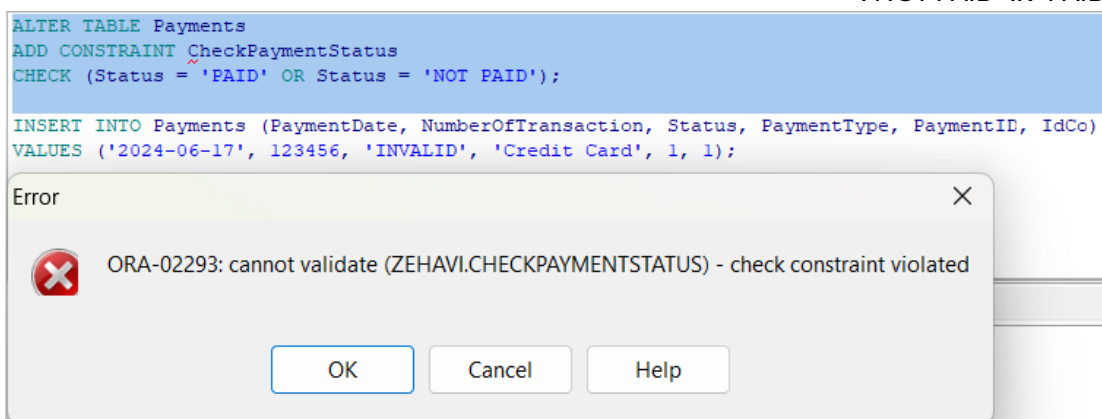
	WORKERSROLE	WAGE	WORKERSMAIL	WORKERSPHONE	SHIFT	IDWO	WORKERSNAME
2	Graphic Designer	10001	IDx57p3s@gmail.com	511729022	Morning	374	Julian Pierce
3	Graphic Designer	12095	g5Hx7qCv@gmail.com	596598587	Morning	375	Evelyn Robinson
4	Graphic Designer	8866	Ac40NdiY@gmail.com	536280685	Morning	378	Harper Wright
5	Graphic Designer	9261	4qAOCGvY@gmail.com	520820659	Morning	392	Harper Moore

## אילוצים

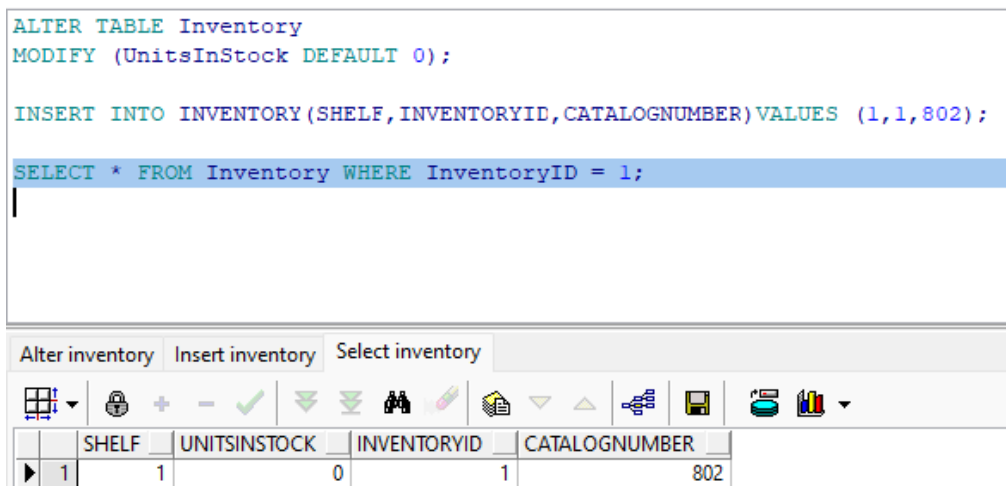
1. אילוץ זה בודק שערך בעמודת WorkersRole לא יכול להיות ריק (NULL). במילים אחרות, האילוץ מבטיח שכל רשומה בטבלה יכיל ערך תקין ולא ריק בעמודת WorkersRole.



2. אילוץ CHECK בטבלה Payments בודק שערך בעמודת Status הוא אחד מבין 'PAID' או 'NOT PAID'. במילים אחרות, האילוץ מוודא שערך ה-Status תקין ותואם לאחד מהערכים המותרים: 'PAID' או 'NOT PAID'.



3. אם אין ערך מוגדר של כמות במלאי עבור העמודה, הערך של ברירת המחדל יישמש במקומו.



## פונקציות ופרוצדורות

### תוכנית מס 1:

התוכנית ManageStockTransfer נועדה לנהל את המלאי במערכת ההדפסה על ידי העברת מוצרים ממדפים שיש בהם יותר מ-40 יחידות למדפים שיש בהם פחות מ-40 יחידות. היא משתמשת בפונקציה FindShelfUnder40 לאיתור מדף ריק שבו יש פחות מ-40 יחידות, ומבצעת עדכונים בבסיס הנתונים כדי להבטיח אופטימיזציה של המלאי.

### הפרוצדורה:

```

1 CREATE OR REPLACE PROCEDURE ManageStockTransfer AS
2   CURSOR overstock_cursor IS
3     SELECT Shelf, UnitsInStock
4     FROM Inventory
5     WHERE UnitsInStock > 40;
6
7   overstock_rec overstock_cursor%ROWTYPE;
8   target_shelf NUMERIC;
9   units_to_transfer NUMERIC;
10 BEGIN
11   OPEN overstock_cursor;
12   LOOP
13     FETCH overstock_cursor INTO overstock_rec;
14     EXIT WHEN overstock_cursor%NOTFOUND;
15
16     target_shelf := FindShelfUnder40();
17
18     IF target_shelf IS NOT NULL THEN
19       units_to_transfer := overstock_rec.UnitsInStock - 40;
20
21       -- עדכון ומדף המקורי
22       UPDATE Inventory
23       SET UnitsInStock = 40
24       WHERE Shelf = overstock_rec.Shelf;
25
26       -- עדכון המדף הריק
27       UPDATE Inventory
28       SET UnitsInStock = UnitsInStock + units_to_transfer
29       WHERE Shelf = target_shelf;
30     END IF;
31   END LOOP;
32   CLOSE overstock_cursor;
33 EXCEPTION
34   WHEN OTHERS THEN
35     DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
36 END;
37

```

**הפונקציה:**

```

CREATE OR REPLACE FUNCTION FindShelfUnder40 RETURN NUMERIC IS
    shelf_num NUMERIC;
BEGIN
    SELECT Shelf INTO shelf_num
    FROM Inventory
    WHERE UnitsInStock < 40
    AND ROWNUM = 1;

    RETURN shelf_num;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN NULL;
END;

```

**התוכנית הראשית:**

```

begin
    -- Call the procedure
    managestocktransfer;
end;

```

**התוצאה לאחר הפעלת התכנית:**

ניתן לראות שהתכנית סידרה את כל המדפים כך שבכל מדף יהיו מקסימום 40 מוצרים,

למעט מדף אחד בו יש 48 מוצרים, כיוון שיש בסך הכל 10 מדפים שצריכים להכיל 408 מוצרים, לכן במדף אחד יהיה 48 מוצרים.

```

SELECT shelf, COUNT(*) AS product_count
FROM Inventory
GROUP BY shelf
HAVING COUNT(*) >39;

```

	SHELF	PRODUCT_COUNT
1	1	40
2	6	40
3	2	40
4	5	40
5	4	40
6	8	40
7	7	48
8	3	40
9	9	40

## תוכנית מספר 2:

הפונקציה מחשבת את השכר של עובד על בסיס מספר ההזמנות שהוא טיפל בהן, והפרוצדורה מעדכנת את השכר הכולל של כל עובד בהתאם להזמנות שבוצעו. זה עוזר בניהול השכר של העובדים ובמתן תגמול על עבודה טובה.

### הפונקציה:

```

3 CREATE OR REPLACE FUNCTION CalculateWorkerWage(p_worker_id IN NUMBER)
RETURN NUMBER IS
3   v_order_count NUMBER;
   v_wage_per_order NUMBER := 50; -- סכום יבוי עבור כל הזמנה
   v_total_wage NUMBER;
BEGIN
   -- מחשב את מופר ההזמנות שטיפל בהן העובד
3   SELECT COUNT(*) INTO v_order_count
   FROM Orders
   WHERE IdWo = p_worker_id;

   -- מחשב את ושכר הכולל של העובד
   v_total_wage := v_order_count * v_wage_per_order;

   RETURN v_total_wage;
EXCEPTION
3   WHEN OTHERS THEN
   RETURN 0;
END;

```

### הפרוצדורה:

```

CREATE OR REPLACE PROCEDURE UpdateAllWorkersWages AS
   CURSOR c_workers IS
      SELECT IdWo
      FROM Workers;

   worker_rec c_workers%ROWTYPE;
   v_new_wage NUMBER;
BEGIN
   OPEN c_workers;
   LOOP
      FETCH c_workers INTO worker_rec;
      EXIT WHEN c_workers%NOTFOUND;

      -- מחשב את ושכר החדש של העובד
      v_new_wage := CalculateWorkerWage(worker_rec.IdWo);

      -- מעדכן או השכר של העובד בטבלת העובדים
      UPDATE Workers
      SET Wage = v_new_wage
      WHERE IdWo = worker_rec.IdWo;
   END LOOP;
   CLOSE c_workers;
END;

```

## התכנית הראשית:

```

DECLARE
  v_worker_id NUMBER := 1; -- ID מסוים של עובד מסוים
  v_new_wage NUMBER;
BEGIN
  -- קריאה לירוי דויה לעדיו וזן השכר הכולל של כל העובדים
  UpdateAllWorkersWages;

  -- קריאה לפונקציה לחיוב השכר החדש של עובד מסוים
  v_new_wage := CalculateWorkerWage(v_worker_id);
  DBMS_OUTPUT.PUT_LINE('The new wage for worker ID ' || v_worker_id || ' is ' || v_new_wage);

```

## התוצאה לאחר הפעלת התכנית:

SQL Window - -- שאילתה לבדוק את השכר של כל העובדים לאחר העדכון; SELECT IdWo, Wage FROM Workers;

SQL Output Statistics

-- שאילתה לבדוק את השכר של כל העובדים לאחר העדכון;  
 SELECT IdWo, Wage  
 FROM Workers;

	IDWO	WAGE
1	1	50
2	2	50
3	3	0
4	4	100
5	5	100
6	6	100
7	7	0
8	8	0
9	9	0
10	10	0
11	11	50
12	12	0
13	13	0
14	14	50
15	15	100
16	16	0
17	17	0
18	18	50

2:1 zehavi@XE wage, integer, optional