

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
23	-----	שילוב הפרויקטים

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:

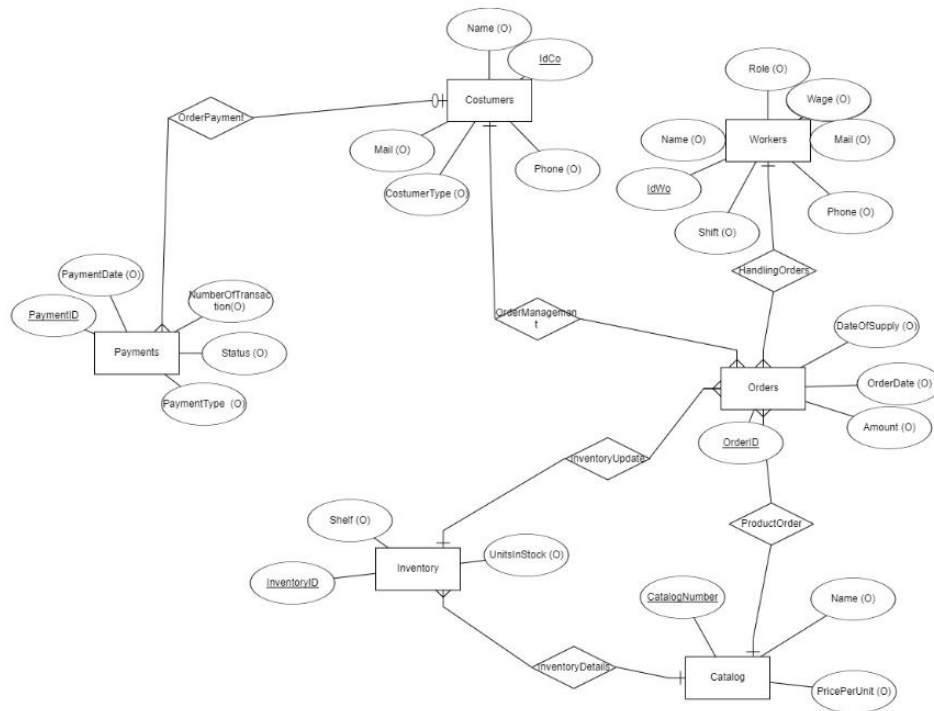
הלקוחות מבצעים הזמנות. כל לקוח יכול לבצע הרבה הזמנות, אבל כל הזמנה שייכת ללקוח אחד בלבד.

תשלום הזמנה:

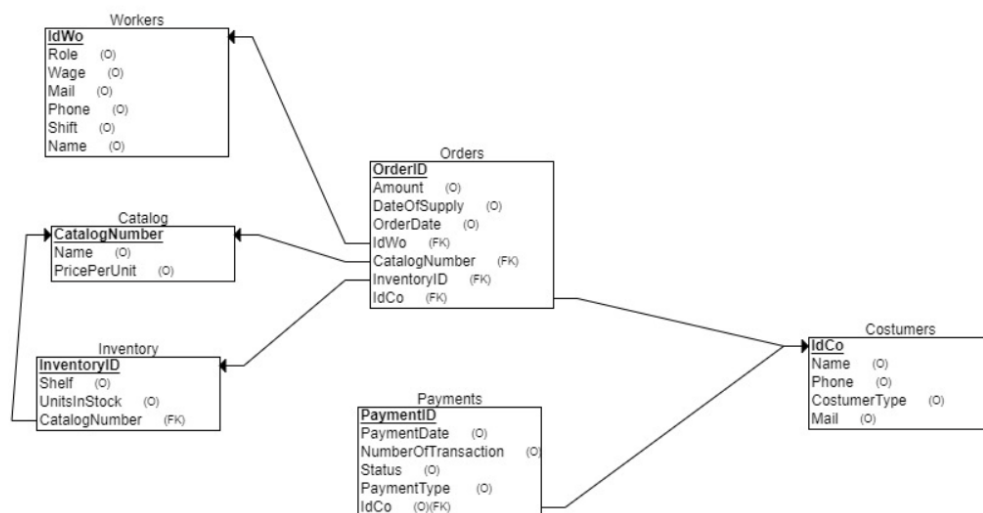
התשלומים קשורים ללקוחות. כל תשלום נעשה על ידי לקוח אחד, אבל כל לקוח יכול לבצע הרבה תשלומים.

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)
);

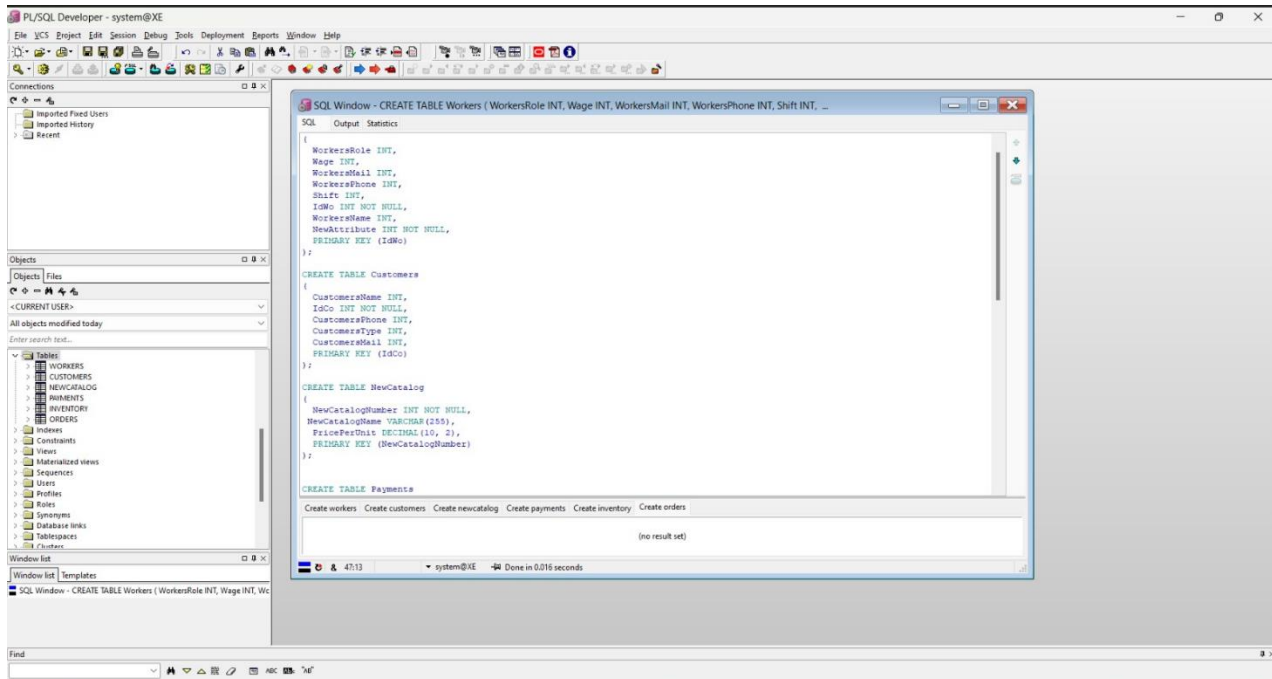
CREATE TABLE Payments
(
    PaymentDate DATE,
    NumberOfTransaction NUMERIC,
    Status VARCHAR(50),
    PaymentType VARCHAR(50),
    PaymentID NUMERIC NOT NULL,
    IdCo NUMERIC,
    PRIMARY KEY (PaymentID),
    FOREIGN KEY (IdCo) REFERENCES Customers(IdCo)
);

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

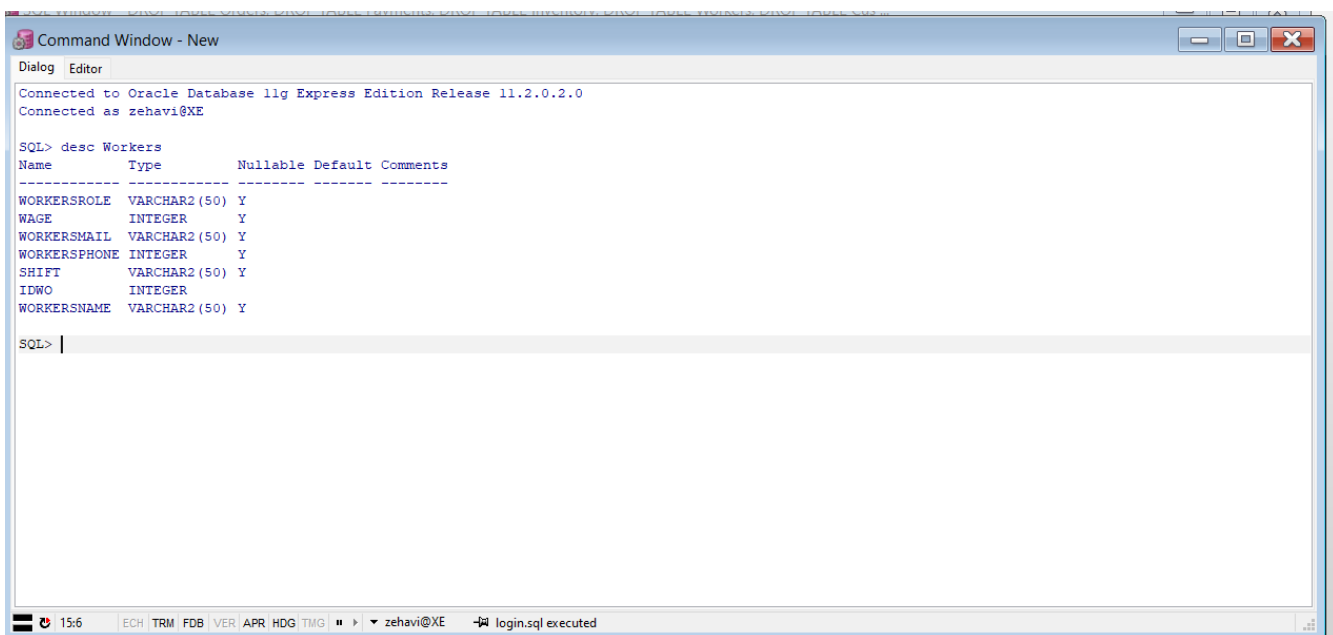
CREATE TABLE Orders
(
    Amount NUMERIC,
    DateOfSupply DATE,
    OrderDate DATE,
    OrderID NUMERIC NOT NULL,
    IdWo NUMERIC NOT NULL,
    CatalogNumber NUMERIC NOT NULL,
    InventoryID NUMERIC NOT NULL,
    IdCo NUMERIC NOT NULL,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (IdWo) REFERENCES Workers(IdWo),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

```

יצירת הטבלאות:

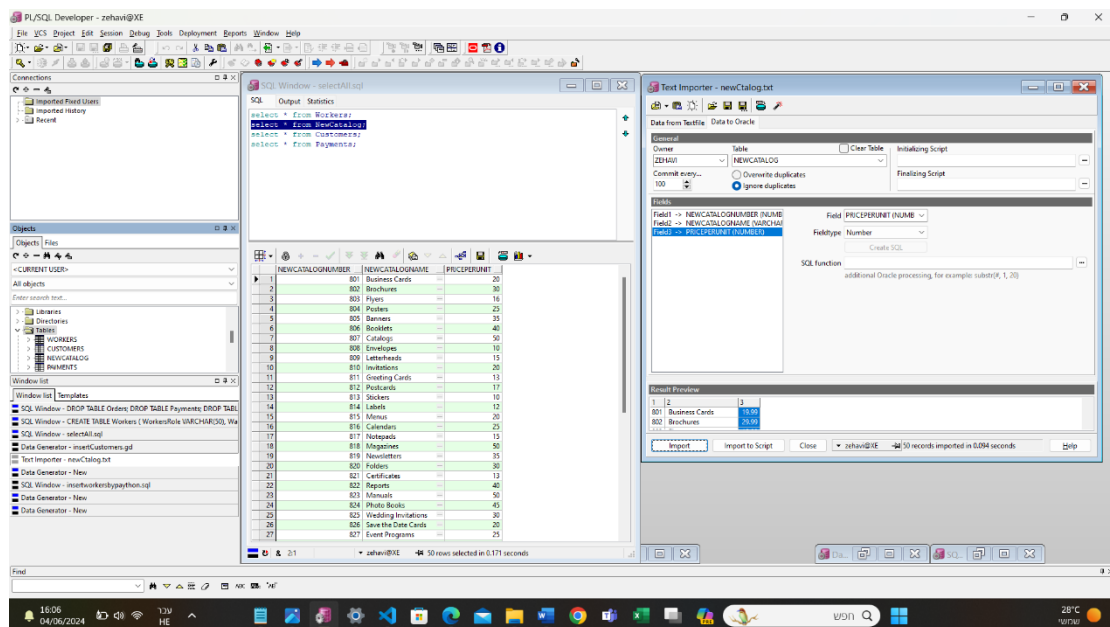


פקודת DESC



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 ('Press Operator', 8933, 'GAAD4Vv@gmail.com', 0526956196, 'Afternoon', 1,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 8919, '6nH7yav@gmail.com', 0544597191, 'Evening', 2,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 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 ('Press Operator', 14350, 'o1ZD76k@gmail.com', 0571796735, 'Morning', 6,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 14530, 'e1alwd@gmail.com', 0538794510, 'Evening', 7,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Customer Service Representative', 10118, 'De4gq58@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 ('Press Operator', 12259, '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 ('Press Operator', 8437, 'Ie10Cqpg@gmail.com', 0528914090, 'Evening', 15,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 8113, 'lgbDakE9@gmail.com', 0533586434, 'Afternoon', 16,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 8609, 'TqQXIA8@gmail.com', 0552337492, 'Morning', 17,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 12821, 'bvlkx3S@gmail.com', 0514103974, 'Afternoon', 18,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 8437, 'Ie10Cqpg@gmail.com', 0528914090, 'Evening', 19,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 10277, 'YE1J0PX@gmail.com', 0508291301, 'Afternoon', 20,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 8961, 'lgbDakE9@gmail.com', 0504837554, 'Afternoon', 21,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Customer Service Representative', 10323, 'Y0ufmg@gmail.com', 0572642649, 'A
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 12730, 'Y0ufmg@gmail.com', 0503384039, 'Morning', 22, '
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 9788, '89CFLdE@gmail.com', 0533375134, 'Afternoon', 23,
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Graphic Designer', 8946, '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', 25, '
INSERT INTO Workers (WorkerRole, Wage, WorkerEmail, WorkerPhone, Shift, IDNo, WorkerName) VALUES ('Press Operator', 7695, 'A1X7IyI@gmail.com', 0545811105, '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

Owner	Table	Number of records
ZEHAVI	CUSTOMERS	400

Definition Options Result

400 records generated in 1.047 seconds

SQL Window - selectAll.sql

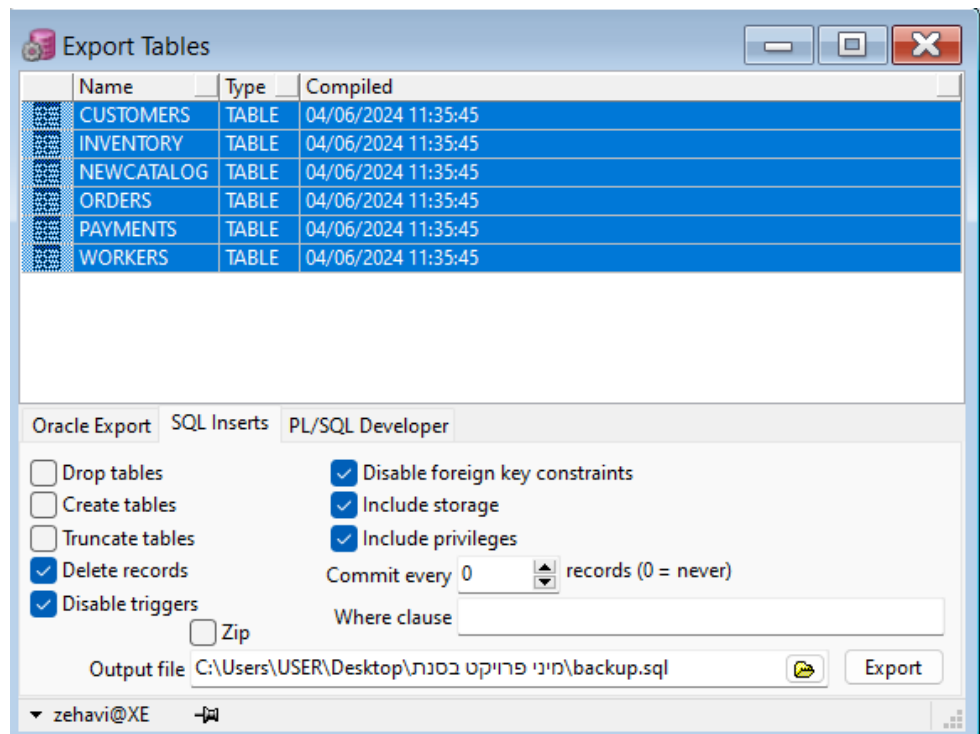
```

SQL      Output  Statistics
select * from Workers;
select * from NewCatalog;
select * from Customers;
select * from Payments;

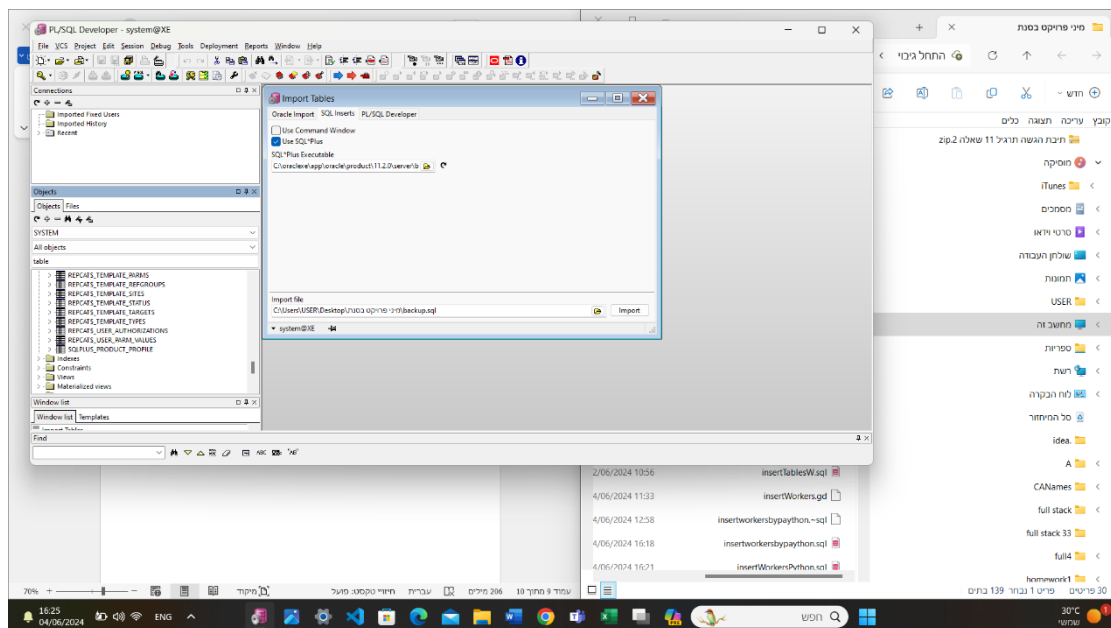
```

CUSTOMERSNAME	IDNO	CUSTOMERSPHONE	CUSTOMERSTYPE	CUSTOMERSMAIL
1 Rebecca-Vassar	401	53837886	school	rebecca.vassar@accesssystems.com
2 Howard-Saucedo	402	54954069	school	howard.saucedo@spimakerexploration.ca
3 Fisher-Warden	403	51038350	office	fisher.warden@trafficmanagement.uk
4 Drew-McIntyre	404	58289789	private	d.mcintyre@gillette.de
5 Laurie-Gordon	405	52511478	private	laurie.gordon@kroger.ch
6 Sylvester-Roberts	406	55455478	private	sylvester.roberts@systems.com
7 Dan-Hubbard	407	565179128	school	dan.hubbard@telaware.com
8 Celia-Makeba	408	57246339	private	cmakeba@ris.com
9 Rich-Alston	409	59884634	school	ralston@safesecurity.at
10 Madeleine-Heslov	410	50721930	school	madeleine.heslov@emergency.com
11 Rade-Aida	411	54053729	office	rade.aida@hudsonwebcorp.ca
12 Ozzy-Giamatti	412	52302455	private	ozzy.giamatti@onesourceprinting.se
13 Riche-Day	413	51375646	office	riche.day@prosperitybancshares.de
14 Samantha-Michaels	414	55627344	private	samantha.michaels@meritagtechnologies.nl
15 Ellen-Roy Parnell	415	51392942	office	ellen@mms.com
16 Melba-Keaton	416	57655942	school	melba.keaton@stmaryland.at
17 Leon-Suchet	417	55341919	private	leon.suchet@pscgroup.in
18 Arnold-Gjeld	418	54457956	private	arnold.gjeld@progressivedesigns.uk
19 George-Gracie	419	56221632	private	george.gracie@atlantictel.com
20 Jimmie-Mandrell	420	58109068	school	jimmie.mandrell@bps.sg
21 Lois-Benson	421	55279045	private	lois.benson@ghsystems.de
22 Bryan-Roberts	422	52946144	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

2. 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	12/02/2015	7	Paid	cash	1034	559

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	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	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	Δx7hln@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:

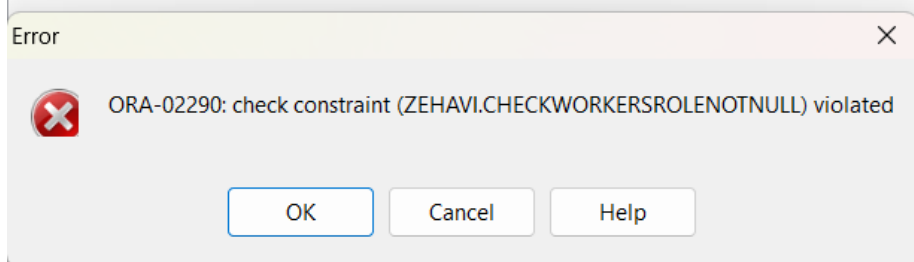
	WORKERSROLE	WAGE	WORKERSMAIL	WORKERSPHONE	SHIFT	IDWO	WORKERSNAME
2	Graphic Designer	10001	IDx57p3s@gmail.com	511729022	Morning	374	Julian Pierce
3	Graphic Designer	12095	g5Hx7qCv@gmail.com	596598587	Morning	375	Evelyn Robinson
4	Graphic Designer	8866	Ac40NdiY@gmail.com	536280685	Morning	378	Harper Wright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

```
ALTER TABLE Workers
ADD CONSTRAINT CheckWorkersRoleNotNull
CHECK (WorkersRole IS NOT NULL);

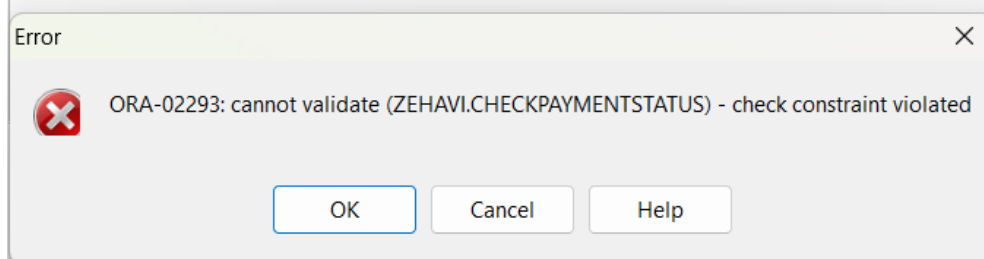
INSERT INTO Workers (WorkersRole, Wage, WorkersMail, WorkersPhone, Shift, IdWo, WorkersName)
VALUES (NULL, 2000, 'john@example.com', 123456789, 'Morning', 1, 'John Doe');
```



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'.

```
ALTER TABLE Payments
ADD CONSTRAINT CheckPaymentStatus
CHECK (Status = 'PAID' OR Status = 'NOT PAID');

INSERT INTO Payments (PaymentDate, NumberOfTransaction, Status, PaymentType, PaymentID, IdCo)
VALUES ('2024-06-17', 123456, 'INVALID', 'Credit Card', 1, 1);
```



















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

```
ALTER TABLE Inventory
MODIFY (UnitsInStock DEFAULT 0);

INSERT INTO INVENTORY (SHELF, INVENTORYID, CATALOGNUMBER) VALUES (1, 1, 802);

SELECT * FROM Inventory WHERE InventoryID = 1;
```

Alter inventory Insert inventory Select inventory

	SHELF	UNITSINSTOCK	INVENTORYID	CATALOGNUMBER
▶ 1	1	0	1	802

פונקציות ופרוצדורות**תוכנית מס 1:**

התוכנית נועדה למצוא את כל המדפים שיש בהם יותר מ-42 מוצרים ולהעביר את המוצרים העודפים למדפים שיש בהם פחות מ-42 מוצרים.

הפרוצדורה:

```

CREATE OR REPLACE PROCEDURE ManageStockTransfer IS
  CURSOR overstocked_shelves IS
    SELECT Shelf
    FROM Inventory
    GROUP BY Shelf
    HAVING COUNT(*) > 42
    ORDER BY Shelf;

  understocked_shelves SYS_REFCURSOR;

  v_over_shelf Inventory.Shelf%TYPE;
  v_under_shelf Inventory.Shelf%TYPE;
  v_inventory_id Inventory.InventoryID%TYPE;
  v_overstock_count INTEGER;
  v_understock_count INTEGER;
  v_transfer_units INTEGER;
BEGIN
  OPEN overstocked_shelves;

  LOOP
    FETCH overstocked_shelves INTO v_over_shelf;
    EXIT WHEN overstocked_shelves%NOTFOUND;

    -- Calculate the number of products on the overstocked shelf
    SELECT COUNT(*) INTO v_overstock_count FROM Inventory WHERE Shelf
    = v_over_shelf;

    -- Find understocked shelves
    understocked_shelves := FindShelfUnder40();

    LOOP
      FETCH understocked_shelves INTO v_under_shelf;
      EXIT WHEN understocked_shelves%NOTFOUND;

      -- Calculate the number of products on the understocked shelf
      SELECT COUNT(*) INTO v_understock_count FROM Inventory WHERE
      Shelf = v_under_shelf;

      -- Calculate how many products can be transferred
      v_transfer_units := LEAST(v_overstock_count - 42, 42 -
      v_understock_count);

      -- Transfer products
      FOR i IN 1 .. v_transfer_units LOOP
        SELECT InventoryID INTO v_inventory_id
        FROM Inventory
        WHERE Shelf = v_over_shelf AND ROWNUM = 1;

        UPDATE Inventory
        SET Shelf = v_under_shelf
        WHERE InventoryID = v_inventory_id;
      
```

```

-- Commit each update
COMMIT;

-- Update the overstock count
v_overstock_count := v_overstock_count - 1;

-- Exit if the overstocked shelf has been normalized
EXIT WHEN v_overstock_count <= 42;
END LOOP;

EXIT WHEN v_overstock_count <= 42;
END LOOP;

CLOSE understocked_shelves;

END LOOP;

CLOSE overstocked_shelves;

EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;
    RAISE;
END;
```

הפונקציה:

```

CREATE OR REPLACE FUNCTION FindShelfUnder40 RETURN SYS_REFCURSOR IS
  shelf_cursor SYS_REFCURSOR;
BEGIN
  OPEN shelf_cursor FOR
    SELECT Shelf
    FROM Inventory
    GROUP BY Shelf
    HAVING COUNT(*) < 42
    ORDER BY Shelf;

  RETURN shelf_cursor;
```

התוכנית הראשית:

```

begin
  -- Call the procedure
  managestocktransfer;
end;
```

התוצאה לאחר הפעלת התכנית:

ניתן לראות שהתכנית סידרה את כל המדפים כך שבכל מדף יהיו מקסימום 42 מוצרים,

	SHELF	NUMBER_OF_PRODUCTS
1	1	42
2	2	42
3	3	42
4	4	42
5	5	42
6	6	42
7	7	37
8	8	36
9	9	42
▶ 10	10	38

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

SQL Output Statistics

-- שאילתה לבדוק את השכר של כל העובדים לאחר העדכון

```

SELECT IdWo, Wage
FROM Workers;

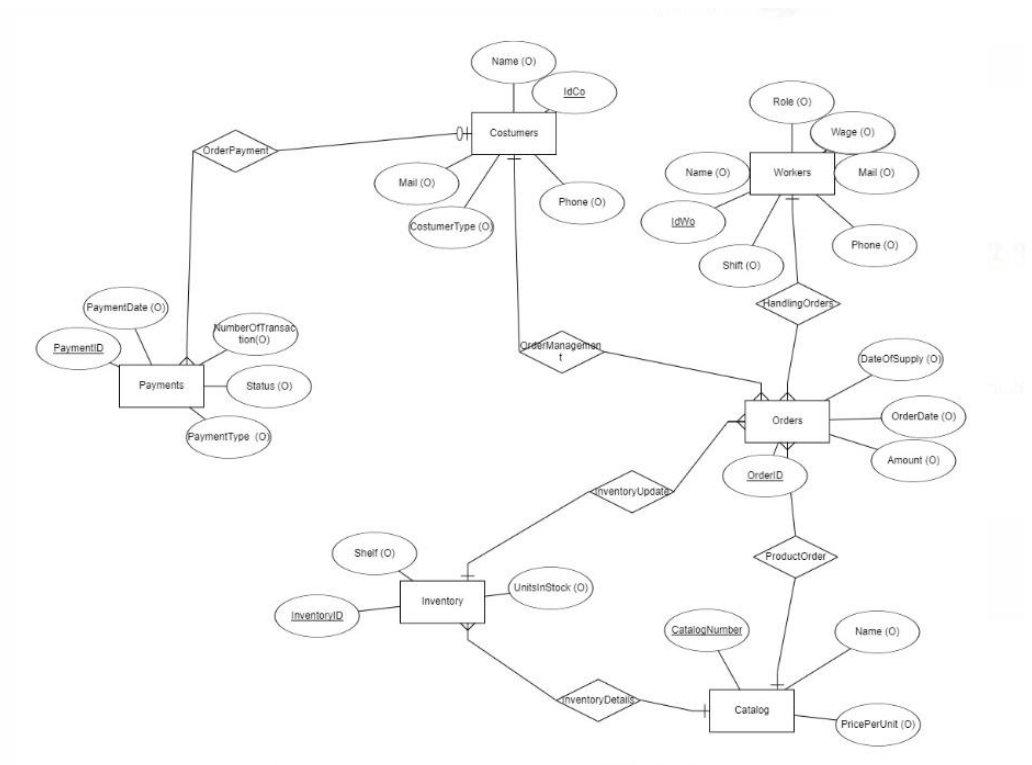
```

	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	0

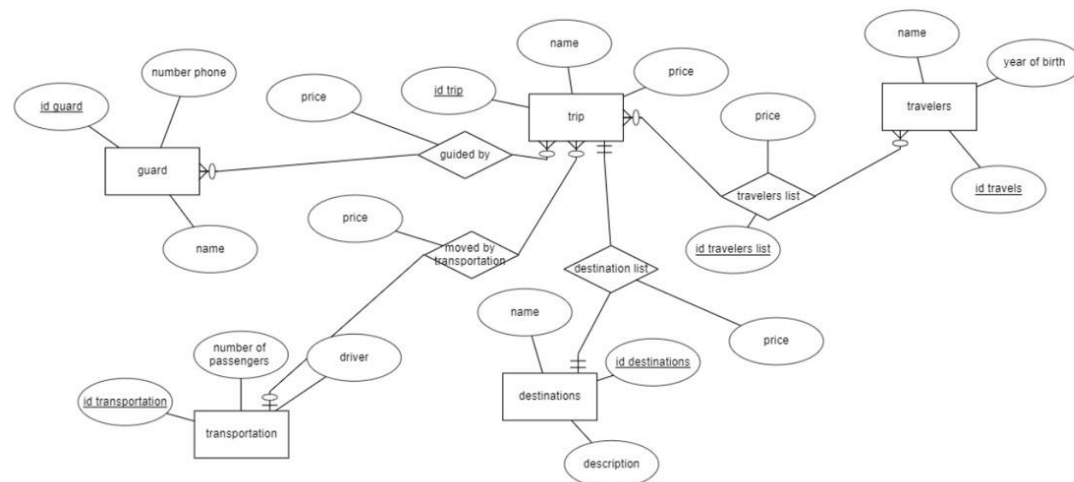
2:1 zehavi@XE wage, integer, optional

שילוב הפרויקטים

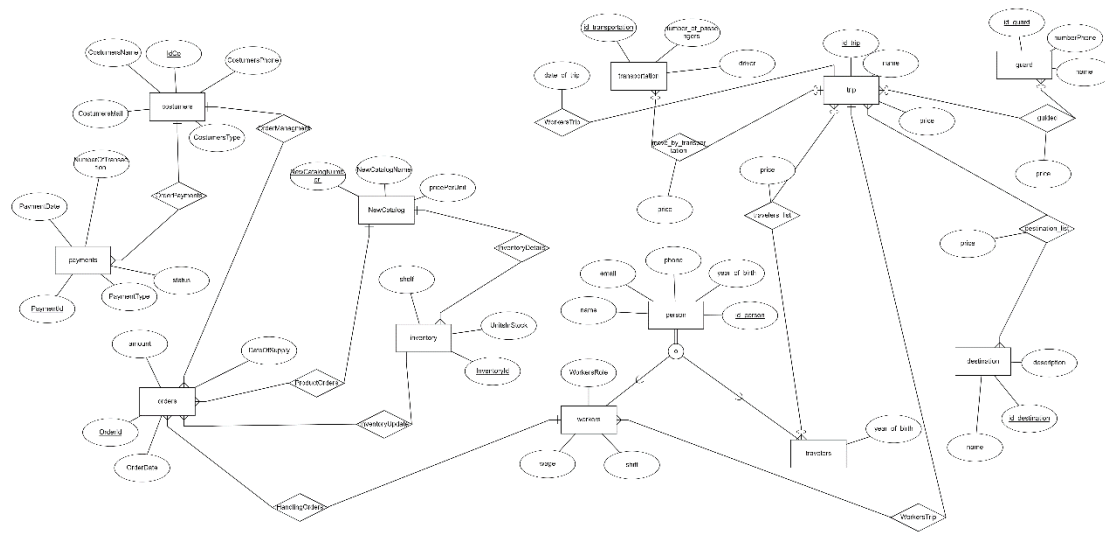
ERD של הבית דפוס לפני האינטגרציה:



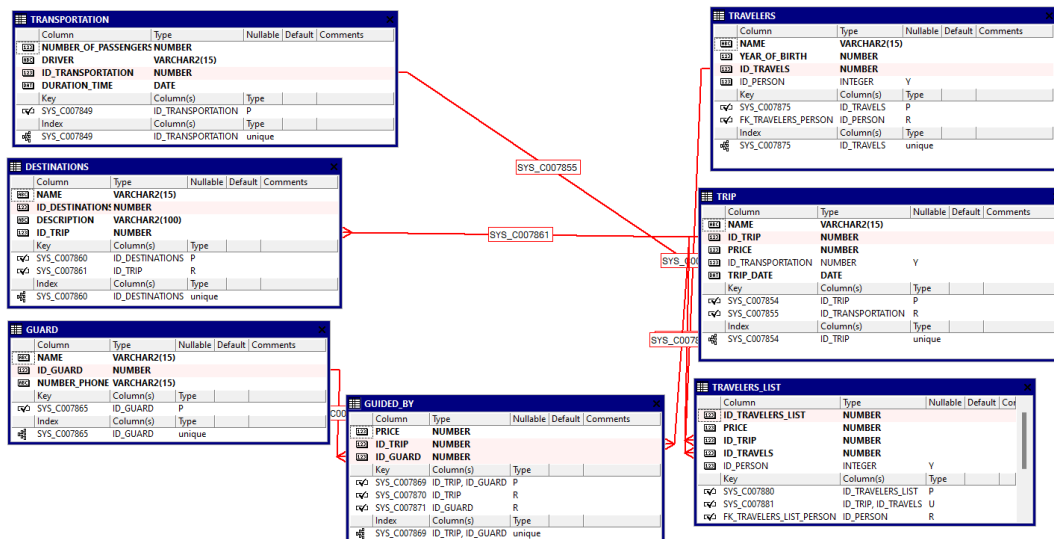
ERD של הטיולים לפני האינטגרציה:



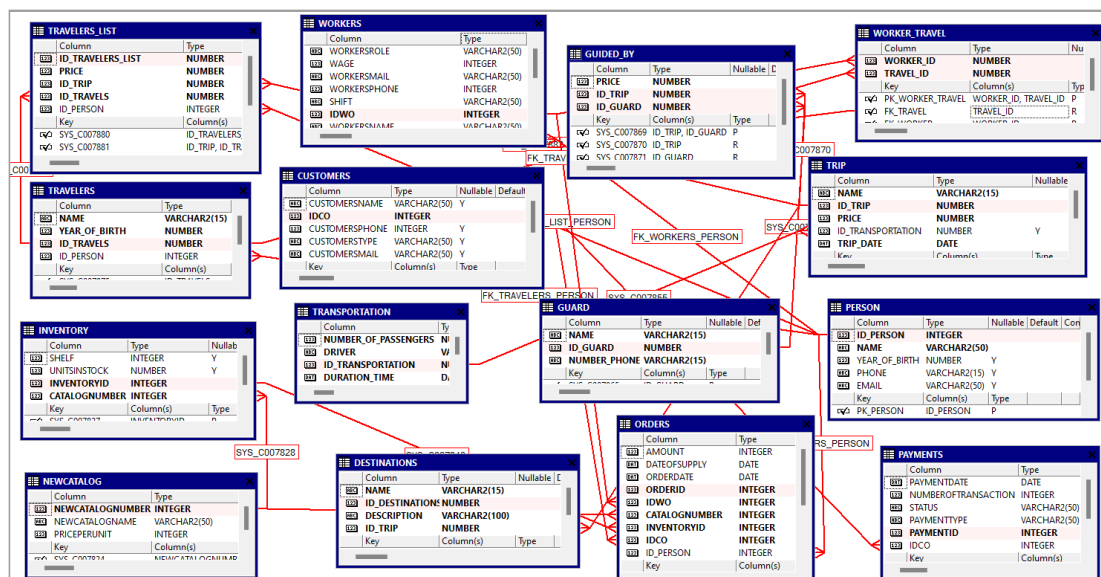
ERD משולב של הבית דפוס עם הטיולים:



DSD עבור טבלת המטיילים:



DSD משותף למטילים ולבית דפוס:



החלטות האינטגרציה:

הגדרנו ישות חדשה, PERSON כך ש-workers ו-TRAVELERS ירשו ממנה. בנינו קשר בשם WORKER_TRAVEL המקשר בין טבלת הטיולים לטבלת העובדים כך שכל עובד משויך לטיול מסוים.

התאמת הנתונים:

בPERSON ישנם השדות הבאים: NAME, ID_PERSON, EMAIL, PHONE, year_of_birth. ID_PERSON הוא המפתח הראשי. הוספנו את כל הנתונים של הטבלאות שיוורשות מ PERSON אל הטבלה PERSON.

```
BEGIN
EXECUTE IMMEDIATE 'CREATE TABLE PERSON
(
    id_person    INTEGER NOT NULL,
    name         VARCHAR2(50) NOT NULL,
    year_of_birth NUMBER,
    phone        VARCHAR2(15),
    email        VARCHAR2(50),
    CONSTRAINT pk_person PRIMARY KEY (id_person)
)';
END;
```

הסבר: יוצר טבלה בשם PERSON עם העמודות id_person, name, year_of_birth, phone, email.

id_person הוא המפתח הראשי.

```
BEGIN
-- הוספת TRAVELERS
BEGIN
EXECUTE IMMEDIATE 'ALTER TABLE TRAVELERS ADD (id_person INTEGER)';
EXCEPTION
WHEN OTHERS THEN
IF SQLCODE = -01430 THEN
NULL; -- העמודי כבר קיימת
ELSE
RAISE;
END IF;
END;

-- הוספת TRAVELERS
EXECUTE IMMEDIATE 'ALTER TABLE TRAVELERS ADD CONSTRAINT fk_travelers_person FOREIGN KEY (id_person) REFERENCES PERSON (id_person)';
END;
```

הסבר: מוסיף עמודת id_person לטבלת TRAVELERS.

מוסיף מפתח זר המקשר בין id_person בטבלת TRAVELERS ל id_person בטבלת PERSON.

```
BEGIN
-- הוספת WORKERS
BEGIN
EXECUTE IMMEDIATE 'ALTER TABLE WORKERS ADD (id_person INTEGER)';
EXCEPTION
WHEN OTHERS THEN
IF SQLCODE = -01430 THEN
NULL; -- העמודי כבר קיימת
ELSE
RAISE;
END IF;
END;

-- הוספת WORKERS
EXECUTE IMMEDIATE 'ALTER TABLE WORKERS ADD CONSTRAINT fk_workers_person FOREIGN KEY (id_person) REFERENCES PERSON (id_person)';
END;
```

הסבר: מוסיף עמודת id_person לטבלת WORKERS.

מוסיף מפתח זר המקשר בין id_person בטבלת WORKERS ל id_person בטבלת PERSON.

```
BEGIN
EXECUTE IMMEDIATE 'ALTER TABLE ORDERS ADD (id_person INTEGER)';
EXECUTE IMMEDIATE 'ALTER TABLE ORDERS ADD CONSTRAINT fk_orders_person FOREIGN KEY (id_person) REFERENCES PERSON (id_person)';
END;
```

הסבר: מוסיף עמודת id_person לטבלת ORDERS.

מגדיר מפתח זר המקשר בין id_person בטבלת ORDERS ל id_person בטבלת PERSON.

```
BEGIN
EXECUTE IMMEDIATE 'ALTER TABLE TRAVELERS_LIST ADD (id_person INTEGER)';
EXECUTE IMMEDIATE 'ALTER TABLE TRAVELERS_LIST ADD CONSTRAINT fk_travelers_list_person FOREIGN KEY (id_person) REFERENCES PERSON (id_person)';
END;
```

הסבר: מוסיף עמודת id_person לטבלת TRAVELERS_LIST.

מגדיר מפתח זר המקשר בין id_person בטבלת TRAVELERS_LIST ל id_person בטבלת PERSON.

```
BEGIN
-- סטת נתונים ת
-- סטת נתונים ת
BEGIN
FOR rec IN (SELECT id_travels, name, year_of_birth FROM TRAVELERS) LOOP
-- סטת נתונים ת-PERSON
EXECUTE IMMEDIATE 'INSERT INTO PERSON (id_person, name, year_of_birth, phone, email)
VALUES (' || rec.id_travels || ', ' || rec.name || ', ' || rec.year_of_birth || ', NULL, NULL)';

-- סטת נתונים ת-TRAVELERS
EXECUTE IMMEDIATE 'UPDATE TRAVELERS
SET id_person = ' || rec.id_travels || '
WHERE id_travels = ' || rec.id_travels;

END LOOP;
END;

-- סטת נתונים ת
-- סטת נתונים ת
BEGIN
FOR rec IN (SELECT IDWO, workersname, workersphone, workersmail FROM WORKERS) LOOP
-- סטת נתונים ת-PERSON
EXECUTE IMMEDIATE 'INSERT INTO PERSON (id_person, name, year_of_birth, phone, email)
VALUES (' || rec.IDWO || ', ' || rec.workersname || ', NULL, ' || rec.workersphone || ', ' || rec.workersmail || ')';

-- סטת נתונים ת-WORKERS
EXECUTE IMMEDIATE 'UPDATE WORKERS
SET id_person = ' || rec.IDWO || '
WHERE IDWO = ' || rec.IDWO;

END LOOP;
END;

-- סטת נתונים ת-TRAVELERS
BEGIN
FOR rec IN (SELECT id_travels, year_of_birth FROM TRAVELERS) LOOP
EXECUTE IMMEDIATE 'UPDATE PERSON
SET year_of_birth = ' || rec.year_of_birth || '
WHERE id_person = ' || rec.id_travels;

END LOOP;
END;

-- סטת נתונים ת-WORKERS
BEGIN
FOR rec IN (SELECT IDWO, workersphone, workersmail FROM WORKERS) LOOP
EXECUTE IMMEDIATE 'UPDATE PERSON
SET phone = ' || rec.workersphone || ',
email = ' || rec.workersmail || '
WHERE id_person = ' || rec.IDWO;

END LOOP;
END;
```

הסבר:

מקבל נתונים מהטבלאות TRAVELERS ו-WORKERS ומכניס אותם לטבלת PERSON.

מעדכן את id_person בטבלאות TRAVELERS ו-WORKERS בהתאמה.

```
BEGIN
EXECUTE IMMEDIATE 'UPDATE TRAVELERS t
SET id_person = (SELECT id_person FROM PERSON p WHERE p.id_person = t.id_travels)';
END;

-- סטת נתונים ת-WORKERS
-- סטת נתונים ת-WORKERS
BEGIN
EXECUTE IMMEDIATE 'UPDATE WORKERS w
SET id_person = (SELECT id_person FROM PERSON p WHERE p.id_person = w.IDWO)';
END;
```

מעדכן את עמודת id_person בטבלאות TRAVELERS ו-WORKERS עם הנתונים מהטבלת PERSON.

```
BEGIN
-- יצירת טולת קשר בין עובדים לטיולים
EXECUTE IMMEDIATE 'CREATE TABLE WORKER_TRAVEL (
worker_id NUMBER NOT NULL,
travel_id NUMBER NOT NULL,
CONSTRAINT pk_worker_travel PRIMARY KEY (worker_id, travel_id),
CONSTRAINT fk_worker FOREIGN KEY (worker_id) REFERENCES WORKERS(IDWO),
CONSTRAINT fk_travel FOREIGN KEY (travel_id) REFERENCES TRAVELERS(id_travels)
)';
END;
```

הסבר: יוצר טבלה חדשה בשם WORKER_TRAVEL עם עמודות worker_id ו-travel_id.

מגדיר מפתחות זרים המקשרים בין worker_id לטבלת WORKERS ו-travel_id לטבלת TRAVELERS.

```
BEGIN
-- סטת נתונים ת-TRAVELERS
-- סטת נתונים ת-TRAVELERS
FOR i IN 1..30 LOOP
INSERT INTO WORKER_TRAVEL (worker_id, travel_id)
select
(SELECT IDWO FROM (SELECT IDWO FROM WORKERS ORDER BY DBMS_RANDOM.VALUE) WHERE ROWNUM = 1),
(SELECT id_travels FROM (SELECT id_travels FROM TRAVELERS ORDER BY DBMS_RANDOM.VALUE) WHERE ROWNUM = 1)
FROM DUAL;

END LOOP;
END;
```

הסבר: הוספת קשרים בין עובדים לטיולים מתוך TRAVELER ו WORKERS

מבטים:

מבט 1

```
CREATE OR REPLACE VIEW TRIP_DETAILS AS
SELECT
    T.ID_TRIP,
    T.NAME AS TRIP_NAME,
    T.PRICE AS TRIP_PRICE,
    T.TRIP_DATE,
    TRAN.NUMBER_OF_PASSENGERS AS TRANSPORTATION_PASSENGERS,
    TRAN.DRIVER AS TRANSPORTATION_DRIVER,
    DEST.NAME AS DESTINATION_NAME,
    DEST.DESCRPTION AS DESTINATION_DESCRIPTION
FROM
    TRIP T
JOIN
    TRANSPORTATION TRAN ON T.ID_TRANSPORTATION = TRAN.ID_TRANSPORTATION
JOIN
    DESTINATIONS DEST ON T.ID_TRIP = DEST.ID_TRIP;
```

מציג את פירוטי הטיולים כולל מידע על התחבורה והיעדים

SELECT * FROM TRIP_DETAILS;

ID_TRIP	TRIP_NAME	TRIP_PRICE	TRIP_DATE	TRANSPORTATION_PASSENGERS	TRANSPORTATION_DRIVER	DESTINATION_NAME	DESTINATION_DESCRIPTION
1	Jeremy	2192	12/11/2117	38	Cary	Kirsten	The journey begins: odio expedita nostrud est quidem commodo. qui aut ...
2	Grant	2917	19/08/2011	26	Clay	Dean	The journey begins: maxime debitis accusamus mollitia tempor et. ...
3	Sigourney	3912	24/06/2415	22	Harris	Bebe	The journey begins: dolores non in et qui incididunt. id. ...
4	Holly	1607	25/09/2525	32	Clarence	Danni	The journey begins: elit distinctio in nulla in eos. in placeat est consequat ...
5	Dwight	3745	26/02/2723	24	Bernie	Allan	The journey begins: delectus sed. ...
6	Ivan	2267	26/01/2531	37	Rosanna	Terrence	The journey begins: ex ad aut debitis perferendis optio. assumenda aut ne ...

שאלתא 1:

```
SELECT
    T.TRIP_NAME,
    T.TRIP_PRICE,
    T.TRIP_DATE,
    NVL(GL.GUIDED_BY_COUNT, 0) AS GUIDED_BY_COUNT
FROM
    TRIP_DETAILS T
LEFT JOIN
    (SELECT ID_TRIP, COUNT(*) AS GUIDED_BY_COUNT FROM GUIDED_BY GROUP BY ID_TRIP) GL
    ON T.ID_TRIP = GL.ID_TRIP
WHERE
    T.ID_TRIP = 101;
```

הצגת פרטים על טיול מסוים.

	TRIP_NAME	TRIP_PRICE	TRIP_DATE	GUIDED_BY_COUNT
1	Maria	2368	08/08/1977	1

שאלתא 2:

```

SELECT
    T.ID_TRIP,
    T.NAME AS TRIP_NAME,
    T.PRICE AS TRIP_PRICE,
    EXTRACT(YEAR FROM T.TRIP_DATE) AS TRIP_YEAR
FROM TRIP T JOIN (SELECT
    EXTRACT(YEAR FROM TRIP_DATE) AS TRIP_YEAR,
    MAX(PRICE) AS MAX_PRICE
    FROM TRIP
    GROUP BY EXTRACT(YEAR FROM TRIP_DATE)
) MaxPrices
ON EXTRACT(YEAR FROM T.TRIP_DATE) = MaxPrices.TRIP_YEAR
AND T.PRICE = MaxPrices.MAX_PRICE
ORDER BY TRIP_YEAR;

```

מחזירה את הטיול הכי יקר בכל שנה

	ID_TRIP	TRIP_NAME	TRIP_PRICE	TRIP_YEAR
1	39	Miriam	2708	1901
2	144	Tal	2234	1902
3	351	Udo	3223	1907
4	294	Ricardo	3317	1910
5	355	Sara	3727	1914
6	93	Swoosie	2292	1919
7	277	Martin	3159	1921
8	154	Trick	2811	1922
9	166	William	3204	1924
10	353	Carole	3221	1930
11	325	Embeth	3868	1931
12	100	Joseph	3872	1932
13	117	Chalee	2938	1936
14	347	Christmas	3902	1938
15	9	Regina	3995	1941
16	150	Laura	1742	1943
17	258	Rhona	2404	1947
18	187	Courtney	1163	1949
19	213	Hilary	3775	1961
20	66	Donna	3559	1962
21	289	Bruce	3888	1963
22	8	Jean	3058	1964
23	32	Mint	2372	1965

מבט 2:

```
CREATE OR REPLACE VIEW WORKER_ORDERS AS
SELECT
    O.ORDERID,
    O.AMOUNT AS ORDER_AMOUNT,
    O.DATEOFSUPPLY,
    O.ORDERDATE,
    W.WORKERSNAME AS WORKER_NAME,
    W.WORKERSROLE AS WORKER_ROLE,
    W.WORKERSPHONE AS WORKER_PHONE,
    C.CUSTOMERSNAME AS CUSTOMER_NAME,
    C.CUSTOMERSPHONE AS CUSTOMER_PHONE
FROM
    ORDERS O
JOIN
    WORKERS W ON O.IDWO = W.IDWO
JOIN
    CUSTOMERS C ON O.IDCO = C.IDCO;
```

מציג את פירטי ההזמנות כולל פירטי העובדים שקשורים להזמנות

SELECT * FROM WORKER_ORDERS;

	ORDERID	ORDER_AMOUNT	DATEOFSUPPLY	ORDERDATE	WORKER_NAME	WORKER_ROLE	WORKER_PHONE	CUSTOMER_NAME	CUSTOMER_PHONE
1	2225	1379	24/09/2021	27/03/2020	Ava Lopez	Press Operator	527082662	Ike-Atlas	534307377
2	2226	1817	28/04/2021	09/05/2021	Zoe Young	Prepress Technician	593438995	Thelma-Red	568235070
3	2227	1886	10/10/2022	27/04/2020	Liam Young	Customer Service Representative	582066880	Micky-Biehn	547017150
4	2228	1169	10/06/2021	24/05/2021	Mason Phillips	Customer Service Representative	531052830	Dermot-Dunst	588912942
5	2229	1111	11/10/2021	02/12/2020	Abigail Harris	Graphic Designer	577858680	Bernard-Hackman	564455789
6	2230	1211	26/08/2022	07/12/2022	Camila Young	Prepress Technician	538306078	Breckin-Inving	572795249
7	2231	1201	13/05/2020	15/08/2022	Zoe Harris	Prepress Technician	595609972	Colin-Close	574447014
8	2233	1394	24/01/2020	28/02/2021	Aaliyah Carter	Graphic Designer	597321717	Alan-Ryder	516991327
9	2234	1678	05/01/2020	11/10/2023	Luna Robinson	Customer Service Representative	504583403	Thelma-Red	568235070
10	2235	1204	03/11/2023	18/10/2023	Oliver Gonzalez	Press Operator	595570776	Alan-Cumming	570966307

שאלתא 1:

```
SELECT
    W.workersname AS WORKER_NAME,
    COUNT(O.orderid) AS NUMBER_OF_ORDERS
FROM
    WORKER_ORDERS O
JOIN
    WORKERS W ON O.WORKER_NAME = W.WORKERSNAME
GROUP BY
    W.workersname
ORDER BY
    NUMBER_OF_ORDERS DESC;
```

מסיר את ההזמנות שכל עובד טיפל בהם

		WORKER_NAME	NUMBER_OF_ORDERS
▶	1	James Moore	9
	2	Luna Robinson	8
	3	Zoe Young	8
	4	Lucas Bell	8
	5	Evelyn Wright	6
	6	Logan Allen	6
	7	William Jones	6
	8	Noah Scott	6
	9	Sofia Thomas	6
	10	Avah Robinson	6
	11	Evelyn Rodriguez	6
	12	Emma Clark	4
	13	Harper Young	4
	14	Elijah Robinson	4
	15	Logan Young	4
	16	Sofia Lewis	4
	17	Claire Miller	4
	18	Evelyn Moore	4
	19	Charlotte Hall	3
	20	Mason Scott	3
	21	Evelyn Diaz	3
	22	Ethan Young	3

שאלת 2:

```

SELECT
    C.customersname AS CUSTOMER_NAME,
    C.customersphone AS CUSTOMER_PHONE,
    C.customersmail AS CUSTOMER_EMAIL,
    SUM(O.ORDER_AMOUNT) AS TOTAL_AMOUNT
FROM
    WORKER_ORDERS O
JOIN
    CUSTOMERS C ON O.CUSTOMER_NAME = C.CUSTOMERSNAME
GROUP BY
    C.customersname,
    C.customersphone,
    C.customersmail
HAVING
    SUM(O.ORDER_AMOUNT) > 1000
ORDER BY
    TOTAL_AMOUNT DESC;

```

כל הלקוחות שסכום ההזמנה שלהם היא מעל 1000 שח

		CUSTOMER_NAME	CUSTOMER_PHONE	CUSTOMER_EMAIL	TOTAL_AMOUNT
▶	1	Edie-Hart	550990425	edie.h@outsourcgroup.com	8142
	2	Denise-Matarazzo	597649129	denise@fpf.de	7864
	3	Pelvic-Hawthorne	595997674	pelvic.h@mercantilebank.com	6891
	4	Sander-Cetera	508856448	sander.c@manhattanassociates.au	6735
	5	Devon-Atlas	523248613	devon.atlas@keymark.nl	6037
	6	Sander-Gallagher	518378374	sgallagher@summitenergy.ca	5773
	7	Gwyneth-Rea	511271596	gwyneth.rea@marathonheater.ar	5596
	8	Anthony-Cassidy	509185607	anthony.cassidy@denalventures.com	5129
	9	Noah-Gilliam	556485339	noah@appriss.com	4768
	10	Natacha-Stiles	512912100	nstiles@travizon.ch	4555

