

מיני פרויקט בבסיסי נתונים בית דפוס

מגישות

רינת ארנפרוינד 213527823

זהבי פרלא 326381480

תוכן עניינים:

2	-----	תוכן עיניים
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

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:

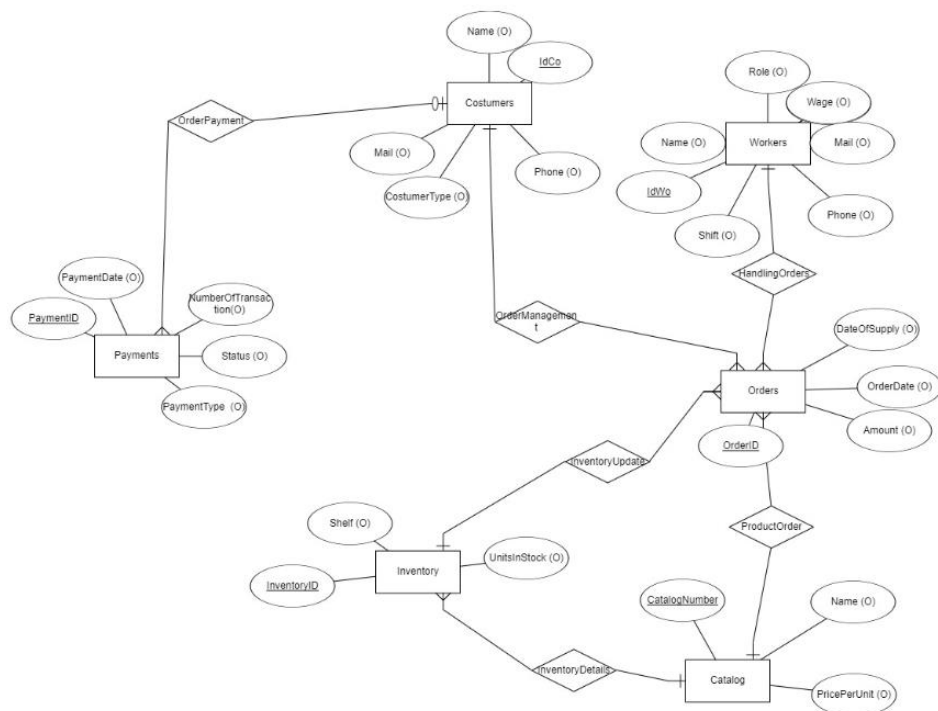
הלקוחות מבצעים הזמנות. כל לקוח יכול לבצע הרבה הזמנות, אבל כל הזמנה שייכת ללקוח אחד בלבד.

תשלום הזמנה:

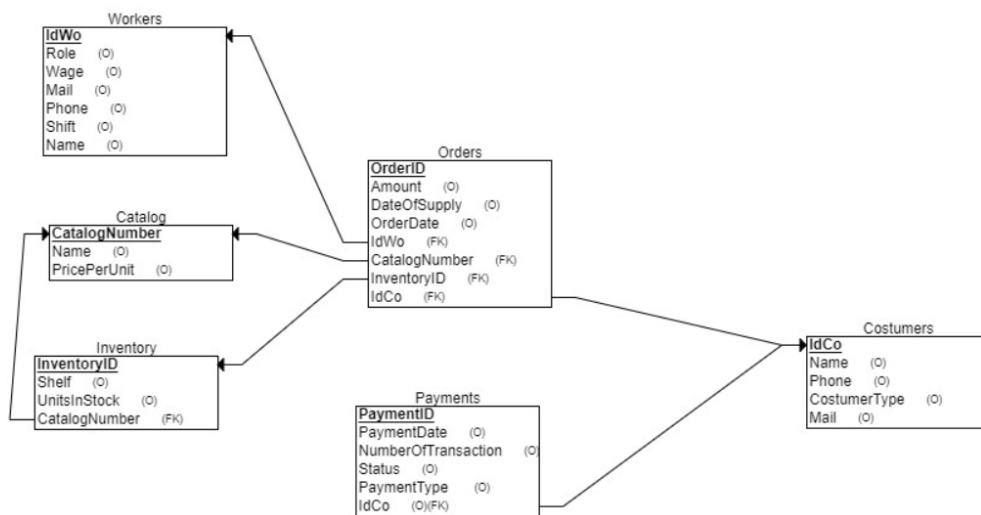
התשלומים קשורים ללקוחות. כל תשלום נעשה על ידי לקוח אחד, אבל כל לקוח יכול לבצע הרבה תשלומים.

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)
);

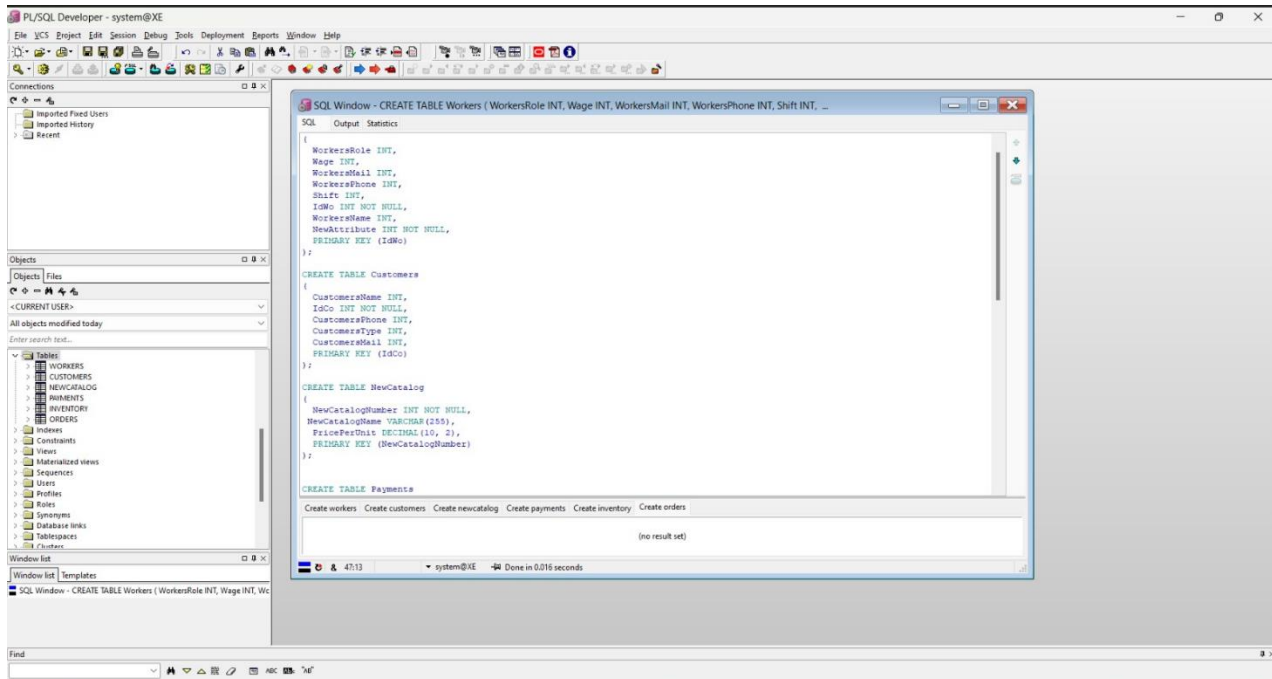
CREATE TABLE Payments
(
    PaymentDate DATE,
    NumberOfTransaction NUMERIC,
    Status VARCHAR(50),
    PaymentType VARCHAR(50),
    PaymentID NUMERIC NOT NULL,
    IdCo NUMERIC,
    PRIMARY KEY (PaymentID),
    FOREIGN KEY (IdCo) REFERENCES Customers(IdCo)
);

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

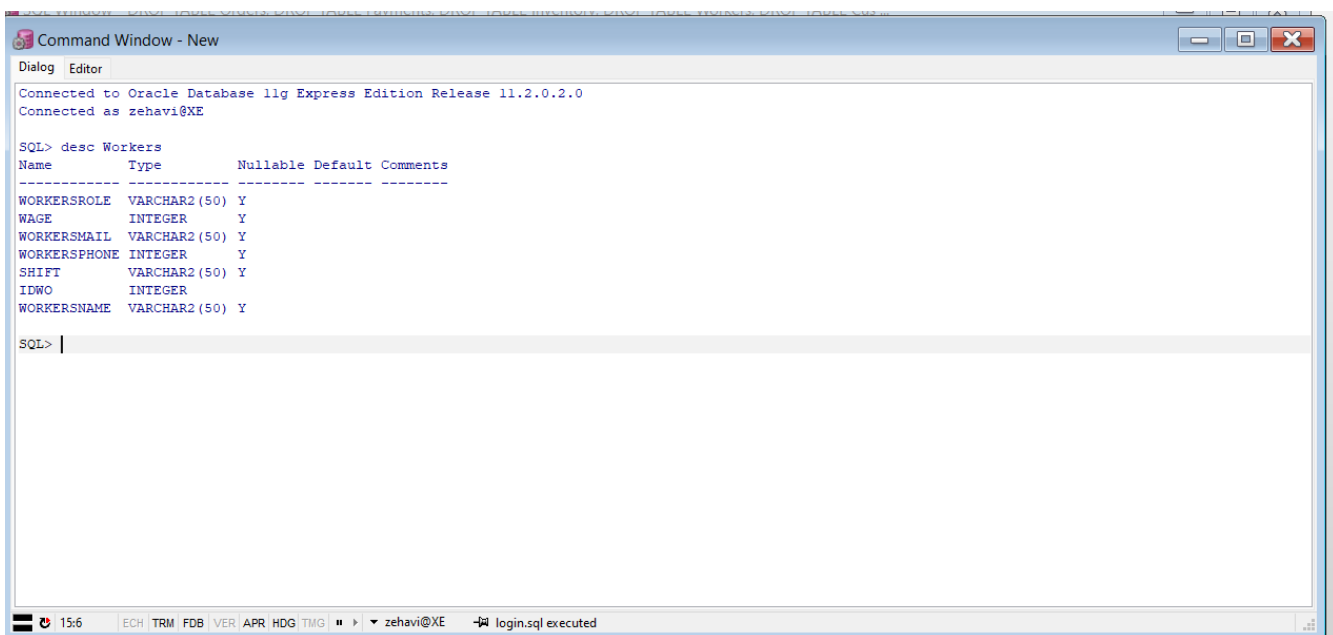
CREATE TABLE Orders
(
    Amount NUMERIC,
    DateOfSupply DATE,
    OrderDate DATE,
    OrderID NUMERIC NOT NULL,
    IdWo NUMERIC NOT NULL,
    CatalogNumber NUMERIC NOT NULL,
    InventoryID NUMERIC NOT NULL,
    IdCo NUMERIC NOT NULL,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (IdWo) REFERENCES Workers(IdWo),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

```

יצירת הטבלאות:

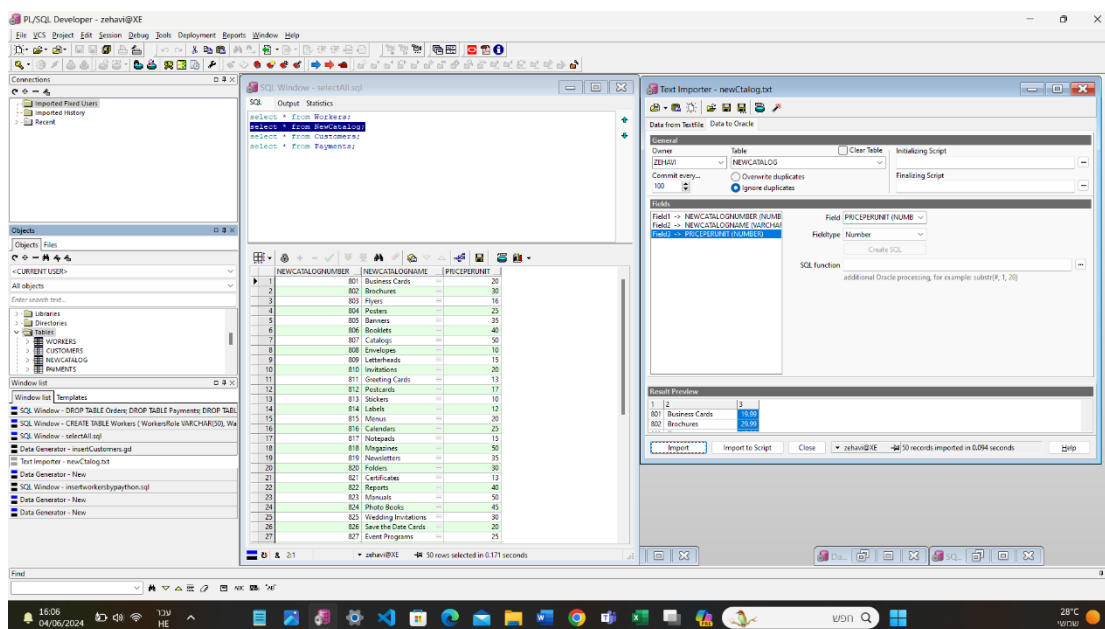


פקודת DESC



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()

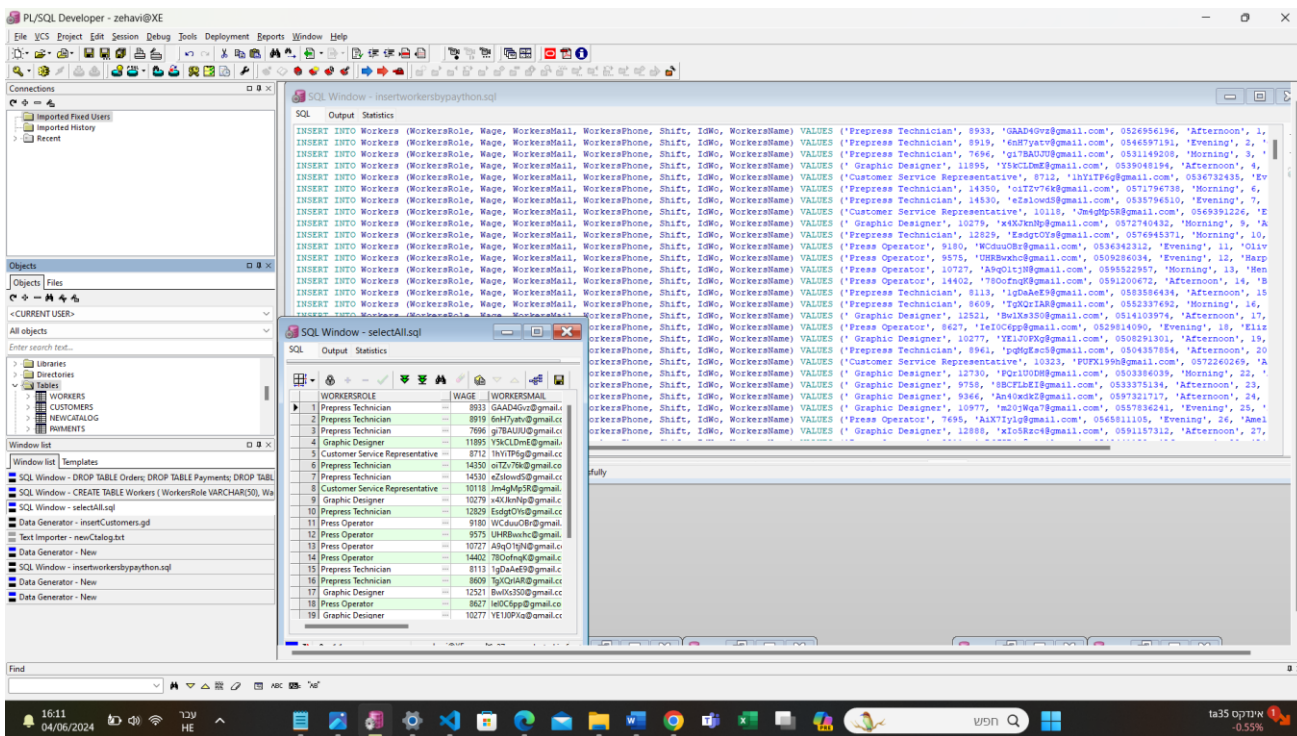
    # Choose a random shift
    shift = random.choice(shifts)

    # Generate a random worker name
    workers_name = generate_name()

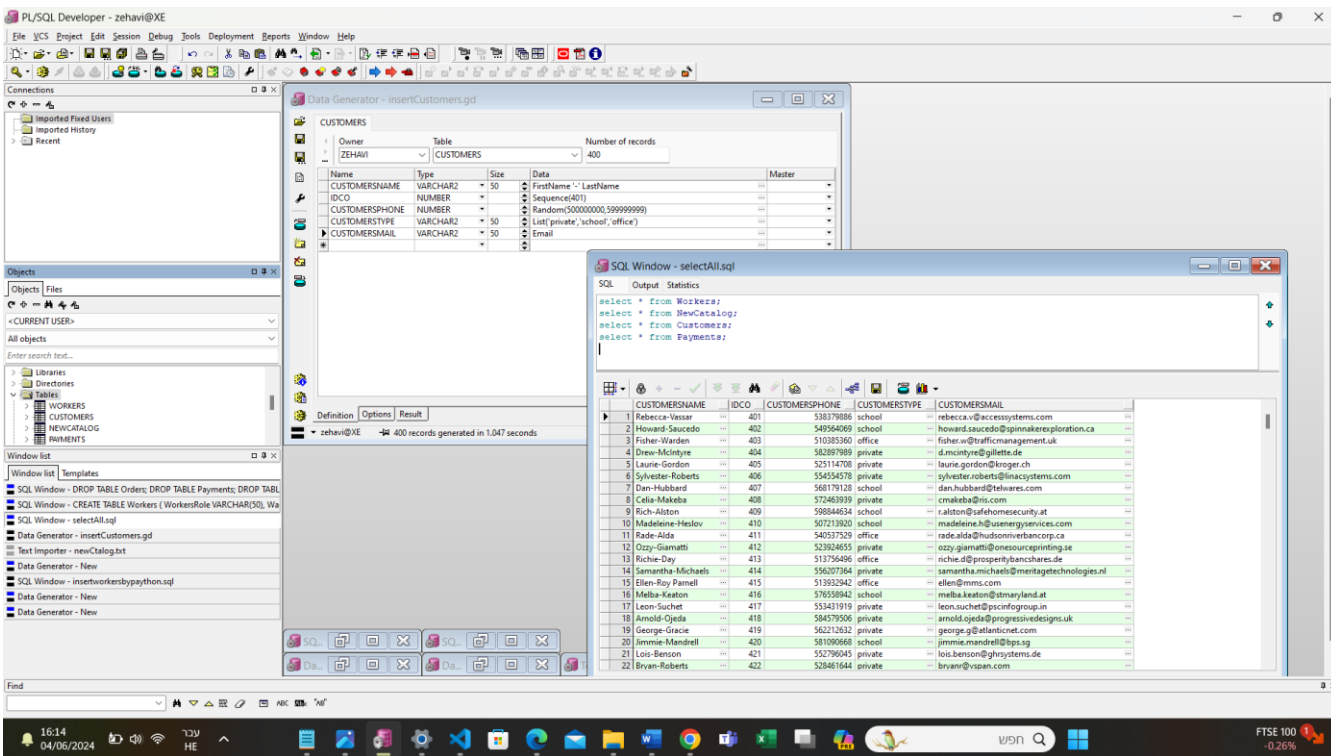
    # Create INSERT statement
    insert_query = f"INSERT INTO Workers (WorkersRole, Wage, WorkersMail, WorkersPhone, Shift, IdNo, WorkersName) VALUES ({workers_role}, {wage}, '{workers_email}', {workers_phone}, '{shift}', {i+1}, '{workers_name}');"

    # Print INSERT statement
    print(insert_query)

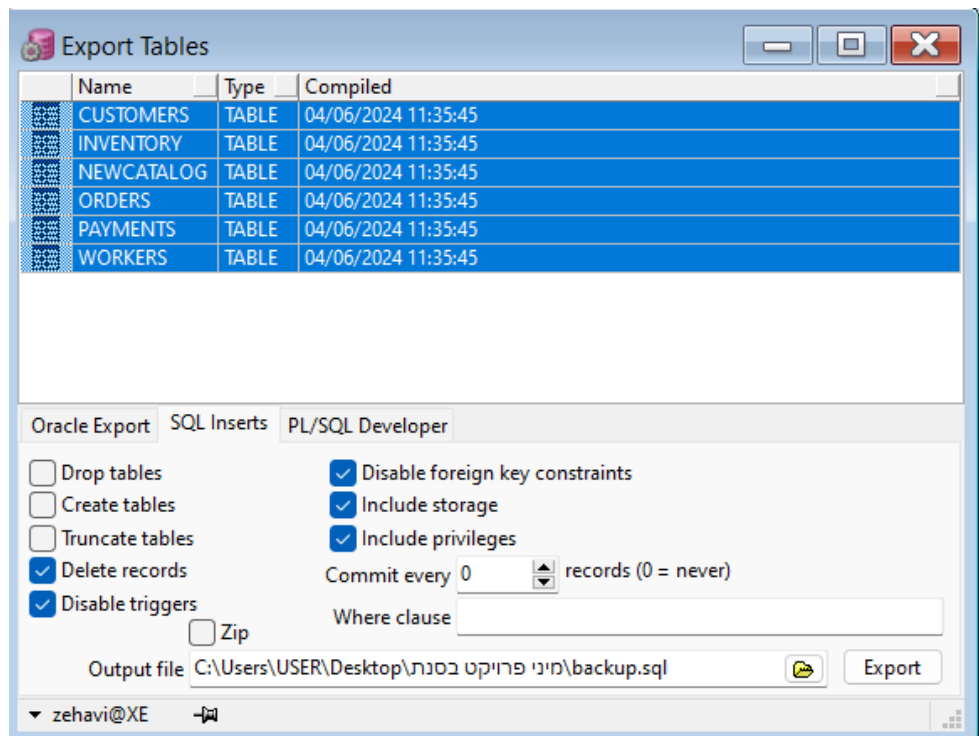
```

הכנסה באמצעות DATAGENERATOR



גיבוי נתונים:



שחזור נתונים:

