מגישות: אילה חיים – 325302271

תמר גפנר - 213451073

השכרת

שמלות

תוכן עניינים

והסבר.....................................................................................................................2 ERD

3………………………………………………………………………………………………………………………………………DSD

3…..……………………………………………………………………………………………………………..…CREATE TABLES

5……………..………………………………………………………………………………………………………………………DESC

6…………………………………………………………………………………………………………………DATA GENERATOR

6………………………………………………………………………………………………………………… Data Imports File

6…………………………………………………………….……………………………………………………………… Mockaroo

7…………………………………………………………….……………………………………………………………… גיבוי.........

7…………………………………………………………….……………………………….………………………… שחזור נתונים

8…………………………………………………………….……………………………………………………………… שאילתות..

16………………………………………………………….……………………………………………………………… אילוצים.....

19………………………………………………………….………………………….………………… שאילתות עם פרמטרים

23………………………………………………………….………………………….………………… תוכנית ראשונה...........

41………………………………………………………….………………………….……………………… תוכנית שניה.........

46………………………………………………………….………………………….………… נתונים של האגף החדש.......

47………………………………………………………….………………………….……אינטגרציה................... החלטות

48………………………………………………………….…………….……………………………… אופן האינטגרציה.........

53………………………………………………………….…………….……………………………………………………… Views

**השכרת שמלות**

תמונה שמכילה שרטוט, אומנות קווים, תרשים, לבן

התיאור נוצר באופן אוטומטי**תרשים ERD:**

פירוט:

הישות המרכזית היא הזמנות שהיא מחברת בין כולם. יש לה קשר עם לקוחות שמסמן איזה לקוח

הזמין את ההזמנה. בנוסף יש לה קשר עם עובדים שמציין איזה עובד ביצע את ההזמנה.

גם לשמלות יש קשר שמסמן איזה שמלה הוזמנה על ידי הלקוח, בכל הזמנה יש רק סוג שמלה אחת.

ולכל הזמנה יש תשלום אחד שכולל בתוכו תשלום על קנס במידה והיה קנס.

וכן קשר לכביסה שכל שמלה שחוזרת מהלקוח הולכת ישר למכבסה.

תמונה שמכילה טקסט, צילום מסך, תרשים, תצוגה

התיאור נוצר באופן אוטומטי**תרשים DSD:**

**Create tables:**

**תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי**

**תמונה שמכילה טקסט, צילום מסך, מסמך, גופן

התיאור נוצר באופן אוטומטי**

**Desc:**

**תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, צילום מסך, מסמך, גופן

התיאור נוצר באופן אוטומטי**

**תמונה שמכילה טקסט, צילום מסך, מספר, גופן

התיאור נוצר באופן אוטומטיData Generator:**

**Data Imports File:**

**תמונה שמכילה טקסט, צילום מסך

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, צילום מסך, תוכנה, סמל מחשב

התיאור נוצר באופן אוטומטיMockaroo:**

**תמונה שמכילה טקסט, צילום מסך, תוכנה, סמל מחשב

התיאור נוצר באופן אוטומטיגיבוי:**

**שחזור נתונים:**

**תמונה שמכילה טקסט, צילום מסך, תוכנה, תצוגה

התיאור נוצר באופן אוטומטי**

**שאילתות:**

1.הסבר: השאילתה מחזירה עבור כל לקוח במידה והשכיר שמלה והיה לו לפחות קנס אחד, כמה קנסות היו לו ומה הסכום הכולל של הקנסות.

השאילתה:

select C.CID, COUNT(C.CID) AS N\_FINES,SUM(P.FINEPRICE) AS SUM\_FINES

FROM CUSTOMER C JOIN ORDERS O ON (C.CID = O.OID) JOIN PAYMENT P ON (O.OID = P.OID)

WHERE P.FINEPRICE != 0

GROUP BY C.CID

תוצאת השאילתה והזמן:

תמונה שמכילה טקסט, צילום מסך, תצוגה, תוכנה

התיאור נוצר באופן אוטומטי

2.הסבר שאילתה: השאילתה מחזירה את שמות העובדים שטפלו בלפחות 50 הזמנות שונות שמורכבות מלפחות 2 לקוחות שונים. ואת סך המשכורת שהרוויחו עבור כל ההזמנות הללו

השאילתה:

SELECT E.EID,E.ENAME, SUM(SALARY) AS PRICE

FROM EMPLOYEE E JOIN ORDERS O ON O.EID = E.EID

WHERE E.EID in(SELECT O.EID

FROM Orders O

WHERE O.EID in (SELECT O.EID

FROM Orders O

GROUP BY O.CID,O.EID

HAVING COUNT(\*) >=2)

GROUP BY O.EID

HAVING COUNT(\*) >=50)

GROUP BY E.EID,E.ENAME

ORDER BY E.ENAME;

תמונה שמכילה טקסט, צילום מסך, תצוגה, גופן

התיאור נוצר באופן אוטומטי

3. הסבר השאילתה:

השאילתה תחזיר את כל הלקוחות ששילמו עבור כל ההזמנות שלהם.

השאילתה:

SELECT C.CID AS ID, C.CNAME AS NAME

FROM CUSTOMER C

JOIN ORDERS O ON C.CID = O.CID

JOIN PAYMENT P ON O.OID = P.OID

WHERE P.ISPAID = 1

MINUS

SELECT C.CID AS ID, C.CNAME AS NAME

FROM CUSTOMER C

JOIN ORDERS O ON C.CID = O.CID

JOIN PAYMENT P ON O.OID = P.OID

WHERE P.ISPAID = 0;

תמונה שמכילה טקסט, צילום מסך, תצוגה, תוכנה

התיאור נוצר באופן אוטומטי

: הסבר השאילתה4.

השאילתה תחזיר את הלקוח שסכום מחיר כל ההזמנות שלו הוא הגבוה ביותר מבין כל הלקוחות.

השאילתה:

SELECT C.CID, C.CNAME,SUM(P.PRICE) AS PRICE

FROM CUSTOMER C

JOIN ORDERS O ON C.CID = O.CID

JOIN PAYMENT P ON P.OID = O.OID

GROUP BY C.CID, C.CNAME

HAVING SUM(P.Price) >= (

SELECT MAX(total\_price)

FROM (

SELECT SUM(P2.Price) total\_price

FROM ORDERS O2 JOIN PAYMENT P2 ON P2.OID = O2.OID

GROUP BY O2.CID

) max\_total\_price\_subquery

);

תמונה שמכילה טקסט, צילום מסך, תצוגה, תוכנה

התיאור נוצר באופן אוטומטי

5. הסבר השאילתה: השאילתה תעדכן את מחיר ההזמנה שיהיה תואם למחיר השמלה שהוזמנה בהזמנה.

השאילתה:

UPDATE Payment

SET Price =

(

SELECT D.Price

FROM Orders O JOIN Dress D ON O.DId = D.DId

WHERE O.OId = Payment.OId

);

נתונים לפני הרצת השאילתה (דוגמא ספציפית):

תשלום של הזמנה מספר 740 והמחיר שהוכנס הוא 954.

לאחר עדכון:

תמונה שמכילה טקסט, צילום מסך, גופן, קו

התיאור נוצר באופן אוטומטי

אפשר לראות שהמחיר החדש הוא 985.

נבדוק שאכן זה מחיר השמלה של ההזמנה.

הזמנה מספר 740:



נסתכל על השמלה של ההזמנה (שמלה מספר 937):



וכמו שניתן לראות מחיר השמלה הוא 985 כפי שעודכן, כלומר העדכון נכון.

6. הסבר השאילתה: השאילתה מעדכנת שכל ההזמנות שעדיין לא שולמו ועבר כבר שבוע מתאריך החזרה, מעדכנת את הסטטוס שלהן לשולם כלומר מ-0 ל-1.

השאילתה:

UPDATE Payment

SET IsPaid = 1

WHERE IsPaid = 0

AND OId IN (

SELECT O.OId

FROM Orders O

WHERE O.ReturnDate <= (SYSDATE - INTERVAL '7' DAY)

);

נתונים לפני הרצת השאילתה (דוגמא ספציפית):



יש כאן תשלום של הזמנה מספר 303 שהסטטוס שלה שעדיין לא שולמה.



וכמו שניתן לראות כבר עבר יותר משבוע מתאריך החזרה.

אחרי עדכון של השאילתה:



אפשר לראות שסטטוס ההזמנה השתנה ל-1, כלומר שולם.

7. הסבר השאילתה: השאילתה מוחקת את כל התשלומים המיותרים לאותה הזמנה, כלומר בעבור כל הזמנה יש תשלום אחד והשאילתה משאירה את התשלום עם המזהה הכי נמוך ואת כל השאר היא מוחקת.

השאילתה:

DELETE FROM Payment

WHERE PId NOT IN (

SELECT MIN(PId)

FROM Payment

GROUP BY OId

);

נתונים לפני הרצת השאילתה (דוגמא ספציפית):

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

אפשר לראות שלהזמנה 303 יש 4 תשלומים שבפועל הם אותו דבר.

אחרי הרצת השאילתה:

תמונה שמכילה טקסט, צילום מסך, קו, גופן

התיאור נוצר באופן אוטומטי

ניתן לראות שנשאר רק התשלום בעל המספר המזהה הנמוך ביותר(271) והשאר נמחקו מהטבלה.

8. הסבר השאילתה: השאילתה מוחקת את כל הכביסות המיותרות לאותה הזמנה, כלומר בעבור כל הזמנה יש כביסה אחת והשאילתה משאירה את הכביסה עם המזהה הכי נמוך ואת כל השאר היא מוחקת.

השאילתה:

DELETE FROM Laundry

WHERE LId NOT IN (

SELECT MIN(LId)

FROM Laundry

GROUP BY OId

);

נתונים לפני הרצת השאילתה (דוגמא ספציפית):

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

כאן רואים שלהזמנה מספר 534 יש 4 פעמים שכבסו וזה מיותר, נרצה שרק הראשון(42) יישאר.

אחרי הרצת השאילתה:

תמונה שמכילה טקסט, גופן, קו, צילום מסך

התיאור נוצר באופן אוטומטי

ניתן לראות שנשאר רק הכביסה בעלת המזהה הנמוך ביותר.

אילוצים:

1. הסבר: אילוץ מסוג check שבודק שהערך של התכונה הבוליאנית ispaid בטבלה PAYMENT הוא 0 או 1. (1-שולם 0-לא שולם)

האילוץ:

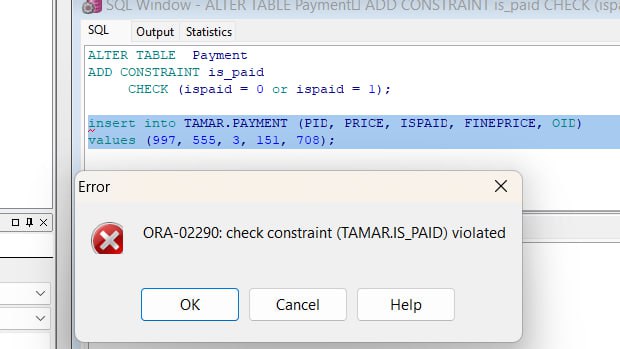
ALTER TABLE Payment

ADD CONSTRAINT is\_paid

CHECK (ispaid = 0 or ispaid = 1);

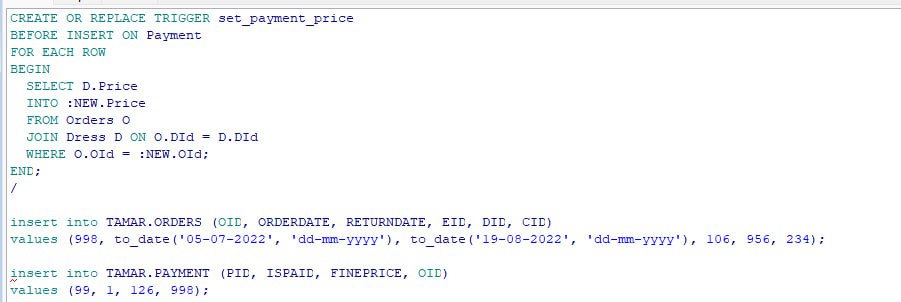
insert into TAMAR.PAYMENT (PID, PRICE, ISPAID, FINEPRICE, OID)

values (997, 555, 3, 151, 708);



ניסינו להכניס תשלום שהערך ispaid הוא 3 וקיבלנו הודעת שגיאה.

1. כאשר נוסף תשלום להזמנה אז מחיר ההזמנה יהיה כמחיר השמלה שהלקוח הזמין. (שימוש באילוץ רגיל לא היה מספיק, היה נדרש Trigger).



כמו שניתן לראות הוספנו הזמנה חדשה עם השמלה 956 וגם תשלום להזמנה ולא ציינו את המחיר, לאחר ההרצה התשלום הוא כמו מחיר השמלה:

תמונה שמכילה טקסט, גופן, קו, מספר

התיאור נוצר באופן אוטומטי

התשלום:

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

השמלה:

.3 האילוץ בודק עבור לקוח את התוכנה email האם הוא תקין. (מכיל @, נגמר בסיומת של .com)

ALTER TABLE Customer

ADD CONSTRAINT chk\_customer\_email

CHECK (

REGEXP\_LIKE(

CEmail,

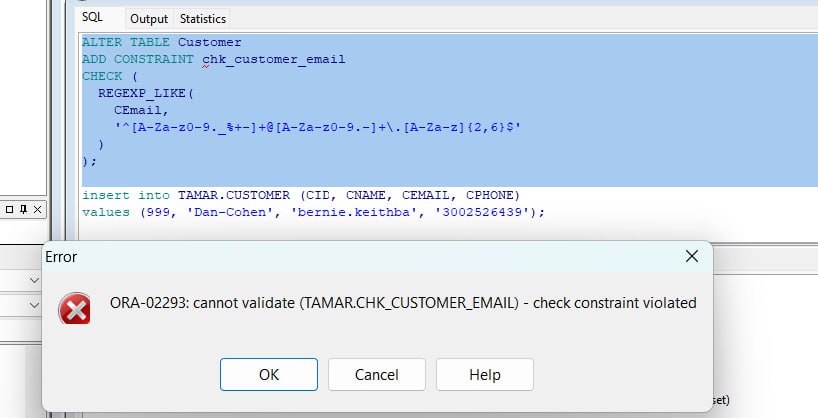
'^[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,6}$'

)

);

insert into TAMAR.CUSTOMER (CID, CNAME, CEMAIL, CPHONE)

values (999, 'Dan-Cohen', 'bernie.keithba', '3002526439');



שאילתות עם פרמטרים:

1. הסבר: השאילתה מקבלת כפרמטרים שני תאריכים ומחזירה עבור כל עובד את כמות ההזמנות שעשה בין התאריכים האלה ואת הסכום הכולל שהרוויח בזמן הזה.

השאילתה:

SELECT ENAME, COUNT(\*) AS n\_orders, SUM(SALARY) AS SALARY

FROM EMPLOYEE E JOIN ORDERS O ON E.EID = O.EID

WHERE O.ORDERDATE BETWEEN &<name= "d\_from" type= date required=true> and &<name= "d\_to" type= date required=true>

GROUP BY E.EID, E.ENAME;

עבור הפרמטרים:

תמונה שמכילה טקסט, צילום מסך, תוכנה, תצוגה

התיאור נוצר באופן אוטומטי

תוצאת השאילתה:

תמונה שמכילה טקסט, צילום מסך, תצוגה, תוכנה

התיאור נוצר באופן אוטומטי

2. הסבר: השאילת מקבלת כפרמטר שם של לקוח מתוך רשימה שהמשתמש בוחר ומחזירה את כל השמלות שאותו לקוח הזמין. בבחירת הלקוח רואים את השם שלו אבל השאילתה מקבלת את התעודת זהות שלו.

השאילתה:

SELECT D.DID, D.LENGTH, D.COLOR,D.PRICE,O.ORDERDATE, O.RETURNDATE

FROM CUSTOMER C

JOIN ORDERS O ON C.CID = O.CID

JOIN DRESS D ON O.DID = D.DID

WHERE C.CID = &<name="CUSTOMER\_NAME"

list="SELECT DISTINCT C.CID,C.CNAME FROM CUSTOMER C JOIN ORDERS O ON C.CID = O.CID ORDER BY CNAME"

description="yes" restricted="yes" >;

עבור הפרמטר:

תמונה שמכילה טקסט, צילום מסך, תצוגה, תוכנה

התיאור נוצר באופן אוטומטי

תוצאת השאילתה:

תמונה שמכילה טקסט, צילום מסך, גופן, תוכנה

התיאור נוצר באופן אוטומטי

3. הסבר: השאילתה מקבלת כפרמטר שם של מכבסה, ניתן לבחור אחת או יותר ומחזירה עבור כל אותן מכבסות שנבחרו את כל ההזמנות שנעשו במכבסה שלכל הזמנה היא מראה את הנתונים הבאים:

מסשם המכבסהת מספר ההזמנה, תאריך איסוף מהמכבסה ומספר שמלה שכבסו.

השאילתה:

select l.lname,o.oid,l.pickingdate,d.did

from laundry l

join orders o on l.oid = o.oid

join dress d on o.did = d.did

where l.lid in

(&<name="laundry\_name"

list="select lid,lname from laundry order by lname"

description="yes" multiselect="yes" >)

עבור הפרמטרים:

תמונה שמכילה טקסט, צילום מסך, גופן, תצוגה

התיאור נוצר באופן אוטומטי

תוצאת השאילתה:

תמונה שמכילה טקסט, צילום מסך, תצוגה, גופן

התיאור נוצר באופן אוטומטי

4. הסבר: השאילתה מקבלת כפרמטר מחיר ותאריך ומחזירה את כל הלקוחות שהזמינו הזמנה שהיא לפחות במחיר שהוכנס ותאריך החזרה שלה היה אחרי התאריך שהוכנס(או באותו היום) וסדר הלקוחות מופיע לפי בקשת המשתמש בסדר עולה או יורד.

השאילתה:

SELECT C.CID, C.CNAME

FROM CUSTOMER C

JOIN ORDERS O ON O.CID = C.CID

JOIN PAYMENT P ON P.OID = O.OID

WHERE P.PRICE >= &<NAME="Price" IFEMPTY=0>

AND O.ORDERDATE >= &<NAME="Return Date" TYPE=DATE REQUIRED="TRUE">

ORDER BY C.CNAME &<NAME="Descending" CHECKBOX="DESC, ASC">;

תמונה שמכילה טקסט, צילום מסך, תצוגה, מספר

התיאור נוצר באופן אוטומטיעבור הפרמטרים:(מיון יורד)

תוצאת השאילתה:

תמונה שמכילה טקסט, צילום מסך, תצוגה, תוכנה

התיאור נוצר באופן אוטומטי

ועבור הפרמטרים:(מיון עולה)

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

תוצאת השאילתה:

תמונה שמכילה טקסט, צילום מסך, תצוגה, תוכנה

התיאור נוצר באופן אוטומטי

הוספת עמודות לטבלאות:

-- Add a column to track the number of rentals for each dress

ALTER TABLE Dress ADD (RentalCount NUMBER(5) DEFAULT 0);

-- Add a column to track customer loyalty points

ALTER TABLE Customer ADD (LoyaltyPoints NUMBER(5) DEFAULT 0);

-- Add a column to track employee performance

ALTER TABLE Employee ADD (PerformanceRating NUMBER(2,1) DEFAULT 3.0);

**תוכנית ראשית ראשונה:**

פרצדורה:

מקבלת תעודת זהות של עובד ומעדכנת את הדירוג שלו באופן הבא:

הדירוג הוא מספר בין 3-5 ולכן החישוב מתבצע כך:

**3+0.2 \* (כמות ההזמנות שביצע)**

הקוד:

create or replace procedure update\_employee\_performance(p\_eid IN NUMBER) IS

CURSOR emp\_orders IS

SELECT COUNT(\*) AS order\_count

FROM Orders

WHERE EId = p\_eid AND OrderDate >= ADD\_MONTHS(SYSDATE, -3);

v\_order\_count NUMBER;

v\_new\_rating NUMBER(2,1);

-- Custom exception for employee not found

employee\_not\_found EXCEPTION;

BEGIN

-- Check if employee exists

DECLARE

v\_employee\_count NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_employee\_count

FROM Employee

WHERE EId = p\_eid;

IF v\_employee\_count = 0 THEN

RAISE employee\_not\_found;

END IF;

END;

OPEN emp\_orders;

FETCH emp\_orders INTO v\_order\_count;

CLOSE emp\_orders;

-- Calculate new rating based on order count

v\_new\_rating := LEAST(5.0, 3.0 + (v\_order\_count \* 0.2));

-- Update employee performance rating

UPDATE Employee

SET PerformanceRating = v\_new\_rating

WHERE EId = p\_eid;

COMMIT;

EXCEPTION

WHEN employee\_not\_found THEN

DBMS\_OUTPUT.PUT\_LINE('Employee with ID ' p\_eid ' does not exist.');

ROLLBACK;

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No orders found for the employee with ID: ' p\_eid);

ROLLBACK;

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error updating employee performance: ' SQLERRM);

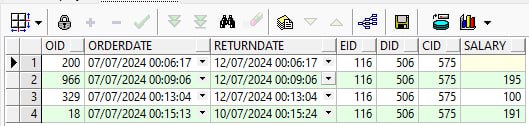
ROLLBACK;

END;

דוגמת הרצה:

עובד 116 כפי שניתן לראות הדירוג שלו הוא לפי הברירת מחדל:

עובד זה ביצע 4 הזמנות:

נריץ את הטסט:

תמונה שמכילה טקסט, צילום מסך, מספר, קו

התיאור נוצר באופן אוטומטי

ונראה את הדירוג של העובד:

תמונה שמכילה טקסט, קו, גופן, צילום מסך

התיאור נוצר באופן אוטומטי

הפונקציה:

הפונקציה מחזירה את חמשת העובדים הכי טובים על פי הקריטריונים של כמות ההזמנות שעשו וממוצע הסכום שלהן.

הקוד:

CREATE OR REPLACE FUNCTION get\_top\_5\_employees RETURN SYS\_REFCURSOR IS

result\_cursor SYS\_REFCURSOR;

BEGIN

OPEN result\_cursor FOR

SELECT EId, EName, PerformanceRating

FROM (

SELECT EId, EName, PerformanceRating

FROM (

SELECT e.EId, e.EName,AVG(p.Price) AS AvgOrderSum, e.performancerating

FROM Employee e

JOIN Orders o ON e.EId = o.EId

JOIN Payment p ON o.OId = p.OId

GROUP BY e.EId, e.EName,e.performancerating

ORDER BY e.performancerating DESC, AvgOrderSum DESC

)

WHERE ROWNUM <= 5;

RETURN result\_cursor;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No data found.');

RETURN NULL;

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Too many rows returned.');

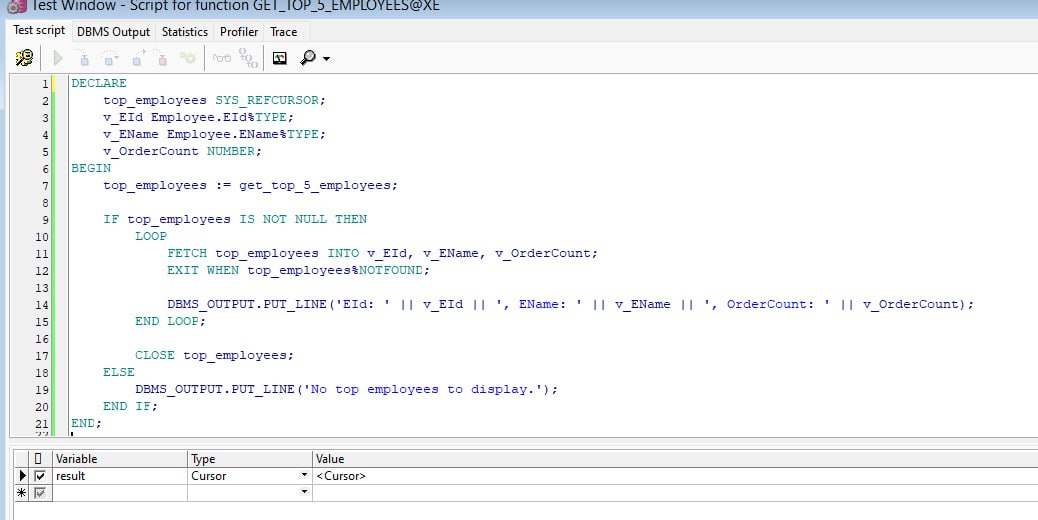
RETURN NULL;

WHEN OTHERS THEN

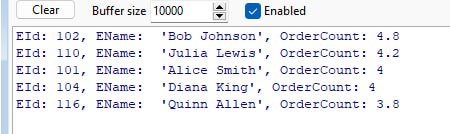
DBMS\_OUTPUT.PUT\_LINE('An error occurred: ' || SQLERRM);

RETURN NULL;

END;

הרצת הפונקציה:

התוצאה:



התוכנית הראשית:

עבור כל עובד מחשבת את הדירוג שלו בעזרת הפרצדורה ואחר כך מפעילה את הפונקציה שמחזירה את חמשת העובדים הכי טובים ומדפיסה את הפרטים שלהם.

הקוד:

DECLARE

v\_eid Employee.EId%TYPE;

v\_ename Employee.EName%TYPE;

v\_rating Employee.PerformanceRating%TYPE;

top\_employees SYS\_REFCURSOR;

BEGIN

-- Update performance rating for all employees

FOR emp IN (SELECT EId FROM Employee) LOOP

update\_employee\_performance(emp.EId);

END LOOP;

-- Get the top 5 employees after updating their ratings

top\_employees := get\_top\_5\_employees;

LOOP

FETCH top\_employees INTO v\_eid, v\_ename, v\_rating;

EXIT WHEN top\_employees%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Employee Performance Review');

DBMS\_OUTPUT.PUT\_LINE('Employee ID: ' v\_eid);

DBMS\_OUTPUT.PUT\_LINE('Employee Name: ' v\_ename);

DBMS\_OUTPUT.PUT\_LINE('Current Rating: ' v\_rating);

END LOOP;

CLOSE top\_employees;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No data found.');

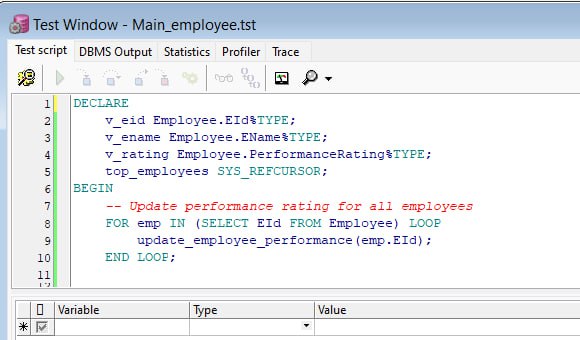
WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Too many rows returned.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error in main program: ' SQLERRM);

END;

 הרצת התוכנית:

התוצאה:

תמונה שמכילה טקסט, צילום מסך, גופן, מסמך

התיאור נוצר באופן אוטומטי

**תוכנית ראשית שניה:**

**הפונקציה:**

הפונקציה מחזירה מספר הזמנה חדשה שלא נמצאת במאגר על מנת להוסיף הזמנה חדשה למאגר.

הקוד:

CREATE OR REPLACE FUNCTION generate\_unique\_oid RETURN NUMBER IS

v\_new\_oid Orders.OId%TYPE;

v\_count NUMBER;

BEGIN

LOOP

-- Generate random number between 0 and 999

v\_new\_oid := ROUND(DBMS\_RANDOM.VALUE(0, 999));

-- Check if this OId is already used

SELECT COUNT(\*)

INTO v\_count

FROM Orders

WHERE OId = v\_new\_oid;

-- If not used, return the new OId

IF v\_count = 0 THEN

RETURN v\_new\_oid;

END IF;

END LOOP;

END generate\_unique\_oid;

הרצת הפונקציה:

תמונה שמכילה טקסט, צילום מסך, קו, גופן

התיאור נוצר באופן אוטומטיהתוצאה שחזרה:

תמונה שמכילה טקסט, צילום מסך, גופן, קו

התיאור נוצר באופן אוטומטיפונקציה שמחזירה מספר תשלום חדש שלא קיים במאגר בכדי להוסיף תשלום חדש למאגר

הקוד:

CREATE OR REPLACE FUNCTION generate\_unique\_pid RETURN NUMBER IS

v\_new\_pid Payment.PId%TYPE;

v\_count NUMBER;

BEGIN

LOOP

-- Generate random number between 0 and 999

v\_new\_pid := ROUND(DBMS\_RANDOM.VALUE(0, 999));

-- Check if this PId is already used

SELECT COUNT(\*)

INTO v\_count

FROM Payment

WHERE PId = v\_new\_pid;

-- If not used, return the new PId

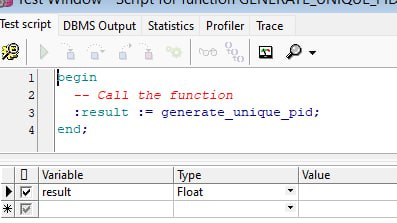
IF v\_count = 0 THEN

RETURN v\_new\_pid;

END IF;

END LOOP;

END generate\_unique\_pid;

הרצת הפונקציה:

תמונה שמכילה טקסט, צילום מסך, מספר, גופן

התיאור נוצר באופן אוטומטיהתוצאה שחזרה:

פרצדורה:

מעדכנת לכל שמלה את מספר הפעמים שהזמינו אותה

הקוד:

CURSOR dress\_orders IS

SELECT DId, COUNT(\*) AS OrderCount

FROM Orders

GROUP BY DId;

v\_did Dress.DId%TYPE;

v\_order\_count NUMBER;

BEGIN

FOR dress\_rec IN dress\_orders LOOP

v\_did := dress\_rec.DId;

v\_order\_count := dress\_rec.OrderCount;

-- Update rental count for the dress

UPDATE Dress

SET RentalCount = v\_order\_count

WHERE DId = v\_did;

END LOOP;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No data found.');

WHEN TOO\_MANY\_ROWS THEN

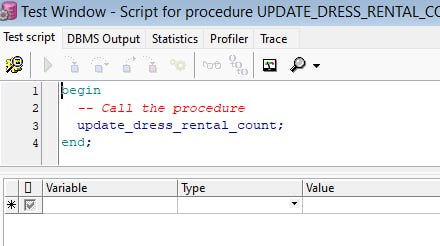
DBMS\_OUTPUT.PUT\_LINE('Too many rows returned.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error updating dress rental count: ' || SQLERRM);

ROLLBACK;

END;

הרצת הפרצדורה:

לדוגמא שמלה מספר 280:

תמונה שמכילה טקסט, גופן, קו, מספר

התיאור נוצר באופן אוטומטיובאמת ניתן לראות שהיא הוזמנה 3 פעמים:

תמונה שמכילה טקסט, גופן, מספר, תוכנה

התיאור נוצר באופן אוטומטיפרצודורה:

מעדכנת עבור כל לקוח את הדירוג שלו לכל הזמנה הוא מקבל דירוג של 10 נקודות

הקוד:

CREATE OR REPLACE PROCEDURE update\_customer\_loyalty IS

CURSOR customer\_orders IS

SELECT CId, COUNT(\*) AS OrderCount

FROM Orders

GROUP BY CId;

v\_cid Customer.CId%TYPE;

v\_order\_count NUMBER;

BEGIN

FOR customer\_rec IN customer\_orders LOOP

v\_cid := customer\_rec.CId;

v\_order\_count := customer\_rec.OrderCount;

-- Calculate loyalty points

UPDATE Customer

SET LoyaltyPoints = v\_order\_count \* 10

WHERE CId = v\_cid;

END LOOP;

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No data found.');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Too many rows returned.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error updating customer loyalty: ' || SQLERRM);

ROLLBACK;

END;

הרצת הפרצדורה:

תמונה שמכילה טקסט, גופן, קו, מספר

התיאור נוצר באופן אוטומטי

עבור לקוח מספר 73 הוא קיבל דירוג של 40:

תמונה שמכילה טקסט, תוכנה, גופן, קו

התיאור נוצר באופן אוטומטיניתן לראות שיש לו באמת 4 הזמנות:

פונקציה

הפונקציה מקבלת מספר זיהוי של שמלה כמות ימי השכרה ודירוג של הלקוח ומחשבת לפיהם את המחיר להשכרה.

מחיר השמלה+(מספר ימי השכרה)\*10

אם זו שמלה שהושכרה בתדירות גבוהה (מעל 7 פעמים) יש הנחה של 10%

אם זו שמלה שהושכרה בתדירות בינונית (מעל 4 פעמים) יש הנחה של 5%

תמונה שמכילה טקסט, מספר, גופן, תוכנה

התיאור נוצר באופן אוטומטיאם הלקוח עם דירוג גבוה (מעל 50) יש הנחה של 10%

אם הלקוח עם דירוג בינוני (מעל 30) יש הנחה של 5%

הקוד:

CREATE OR REPLACE FUNCTION calculate\_rental\_price(p\_did NUMBER, p\_days NUMBER, c\_loyalty NUMBER)

RETURN NUMBER IS

v\_base\_price Dress.Price%TYPE;

v\_rental\_count Dress.RentalCount%TYPE;

v\_final\_price NUMBER;

BEGIN

-- Get the base price and rental count of the dress

SELECT Price, RentalCount INTO v\_base\_price, v\_rental\_count

FROM Dress

WHERE DId = p\_did;

-- Calculate the final price based on rental count and duration

v\_final\_price := v\_base\_price + p\_days\*10;

-- Apply discount for frequently rented dresses

IF v\_rental\_count > 7 THEN

v\_final\_price := v\_final\_price \* 0.9; -- 10% discount

ELSIF v\_rental\_count > 4 THEN

v\_final\_price := v\_final\_price \* 0.95; -- 5% discount

END IF;

-- Apply discount for loyalty points

IF c\_loyalty > 50 THEN

v\_final\_price := v\_final\_price \* 0.9; -- 10% discount

ELSIF c\_loyalty > 30 THEN

v\_final\_price := v\_final\_price \* 0.95; -- 5% discount

END IF;

IF v\_final\_price > 999 THEN

v\_final\_price := 999;

END IF;

RETURN ROUND(v\_final\_price, 2);

EXCEPTION

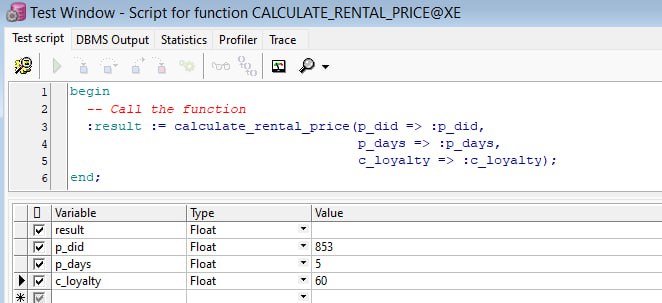
WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Dress not found');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Error calculating rental price: ' || SQLERRM);

END;

הרצת הפונקציה:

עבור שמלה מספר 853:



מחיר השמלה הוא 924 ונתון שיש 5 ימי השכרה + דירוג של 60 של הלקוח

יתבצע החישוב הבא:

**(924+(10\*5))\*0.95\*0.90=832.77**

הנחה של 5 אחוז כי השמלה הושכרה מעל 4 פעמים

והנחה של 10 אחוז כי הדירוג הוא מעל 50

תוצאת ההרצה:

תמונה שמכילה טקסט, צילום מסך, תוכנה, מספר

התיאור נוצר באופן אוטומטי

פרצדורה:

הפרצדורה מקבלת מספרים מזהים של לקוח שמלה כמות ימי השכרה ומחיר ההשכרה

הפורצדורה מוסיפה למאגר הזמנה חדשה עם הפרטים שהתקבלו ותשלום מתאים ומעדכנת את הדירוג של הלקוח ואת כמות הפעמים שהשמלה הושכרה.

הקוד:

CREATE OR REPLACE PROCEDURE process\_order(

p\_cid NUMBER,

p\_did NUMBER,

p\_eid NUMBER,

p\_rental\_days NUMBER,

p\_rental\_price NUMBER

) IS

v\_oid Orders.OId%TYPE;

v\_pid Payment.PId%TYPE;

v\_order\_date DATE := SYSDATE;

v\_return\_date DATE := SYSDATE + p\_rental\_days;

BEGIN

-- Generate unique IDs

v\_oid := generate\_unique\_oid();

v\_pid := generate\_unique\_pid();

-- Insert new order

INSERT INTO Orders (OId, OrderDate, ReturnDate, EId, DId, CId,Salary)

VALUES (v\_oid, v\_order\_date, v\_return\_date, p\_eid, p\_did, p\_cid, p\_rental\_price \* 0.2);

-- Insert payment record

INSERT INTO Payment (PId, Price, IsPaid, FinePrice, OId)

VALUES (v\_pid, p\_rental\_price, 0, 0, v\_oid);

UPDATE Payment

SET Price = p\_rental\_price

WHERE PId = v\_pid;

-- Update dress rental count

UPDATE Dress

SET RentalCount = RentalCount + 1

WHERE DId = p\_did;

-- Update customer loyalty points

UPDATE Customer

SET LoyaltyPoints = LoyaltyPoints + 10

WHERE CId = p\_cid;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Order processed successfully. Order ID: ' v\_oid);

DBMS\_OUTPUT.PUT\_LINE('Payment ID: ' v\_pid);

DBMS\_OUTPUT.PUT\_LINE('Rental price: $' || p\_rental\_price);

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

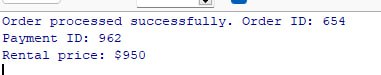
DBMS\_OUTPUT.PUT\_LINE('Error processing order: ' SQLERRM);

END;

הרצת הפרצדורה:

תמונה שמכילה טקסט, צילום מסך, מספר, תוכנה

התיאור נוצר באופן אוטומטי

הפלט:

התווספו הזמנה חדשה ותשלום חדש בהתאם לנתונים שהוכנסו:





**התוכנית הראשית השניה:**

התוכנית בוחרת בצורה רנדומלית לקוח מסוים עובד מספר ימי השכרה ושמלה ובמידה והשמלה פנויה להשכרה בטווח הנבחר התוכנית קוראת לפונקציה שמחשבת את מחיר ההשכרה וקוראת לפרצדורה שיוצאת הזמנה חדשה.

הקוד:

DECLARE

v\_cid Customer.CId%TYPE;

v\_did Dress.DId%TYPE;

v\_eid Employee.EId%TYPE;

v\_rental\_days NUMBER;

v\_estimated\_price NUMBER;

v\_rental\_price NUMBER;

v\_cid\_loyalty NUMBER;

e\_dress\_not\_available EXCEPTION;

BEGIN

-- Select a random customer, employee, dress, and rental duration

SELECT CId , loyaltypoints INTO v\_cid, v\_cid\_loyalty

FROM (SELECT CId, loyaltypoints FROM Customer ORDER BY DBMS\_RANDOM.VALUE)

WHERE ROWNUM = 1;

SELECT EId INTO v\_eid

FROM (SELECT EId FROM Employee ORDER BY DBMS\_RANDOM.VALUE)

WHERE ROWNUM = 1;

v\_rental\_days := FLOOR(DBMS\_RANDOM.VALUE(1, 8)); -- Random rental duration between 1 and 7 days

-- Select a random dress

SELECT DId INTO v\_did

FROM (SELECT DId FROM Dress ORDER BY DBMS\_RANDOM.VALUE)

WHERE ROWNUM = 1;

-- Check if the selected dress is available

DECLARE

v\_order\_count NUMBER;

BEGIN

SELECT COUNT(\*)

INTO v\_order\_count

FROM Orders

WHERE DId = v\_did AND ReturnDate >= SYSDATE;

IF v\_order\_count > 0 THEN

RAISE e\_dress\_not\_available;

END IF;

END;

DBMS\_OUTPUT.PUT\_LINE(CHR(10) 'Processing Order:');

DBMS\_OUTPUT.PUT\_LINE('------------------');

DBMS\_OUTPUT.PUT\_LINE('Customer ID: ' v\_cid);

DBMS\_OUTPUT.PUT\_LINE('Employee ID: ' v\_eid);

DBMS\_OUTPUT.PUT\_LINE('Dress ID: ' v\_did);

DBMS\_OUTPUT.PUT\_LINE('Rental Days: ' v\_rental\_days);

-- Calculate rental price

v\_rental\_price := calculate\_rental\_price(v\_did, v\_rental\_days,v\_cid\_loyalty);

-- Process the order

process\_order(v\_cid, v\_did, v\_eid, v\_rental\_days , v\_rental\_price);

EXCEPTION

WHEN e\_dress\_not\_available THEN

DBMS\_OUTPUT.PUT\_LINE('The selected dress is currently unavailable. Please choose another dress.');

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No data found. Make sure there are records in Customer, Employee, and Dress tables.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error in main program: ' SQLERRM);

END;

הרצת התוכנית:

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

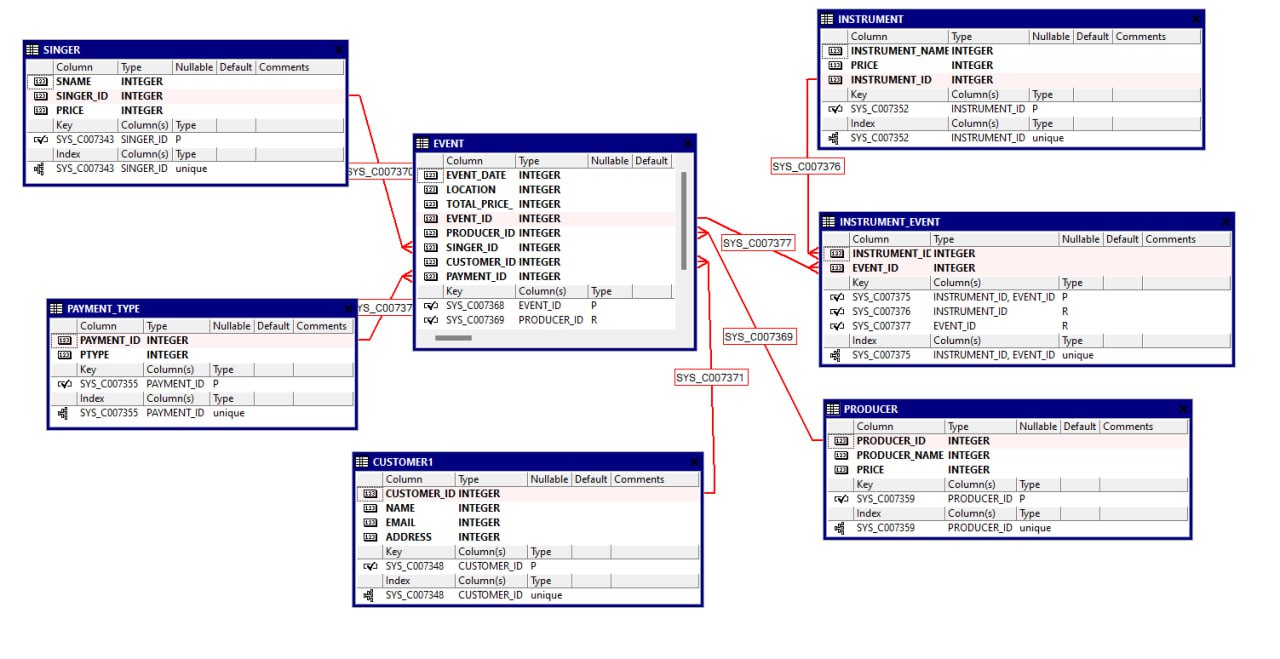
ההזמנה החדשה והתשלום שהוכנסו למאגר בהתאם לנתונים שהוגרלו:

שלב 4

האגף החדש: של DSD

האגף החדש: של ERD

תמונה שמכילה ציור, שרטוט, תרשים, אומנות קווים

התיאור נוצר באופן אוטומטיבהתחלה יצרנו יישויות ולכל יישות את התכונות שלה ולפי המפתחות הזרים הצלחנו להבין את הקשרים בין היישויות וכך הצלחנו ליצור את התרשים.

**החלטות:**

1. הוספנו את התכונה כתובת ליישות של לקוח כדי לקבל את כל התכונות המשותפות ללקוח בתרשים המשולב.

ליישות של תשלום בכדי לקבל את כל התכונות המשותפות\_type הוספנו את התכונה.2

3.מחקנו את התכונה מחיר מהיישות של אירוע כי בתרשים שלנו ביישות של תשלום התכונה מחיר כבר קיימת.

4. שינינו את סוג הקשר בין תשלום להזמנות שיהיה קשר של יחיד לרבים ולא יחיד ליחיד שיהיה מסקרן לפי התרשים שקיבלנו.

חדש:ERD

תמונה שמכילה שרטוט, אומנות קווים, לבן, ציור

התיאור נוצר באופן אוטומטי

DSD חדש:

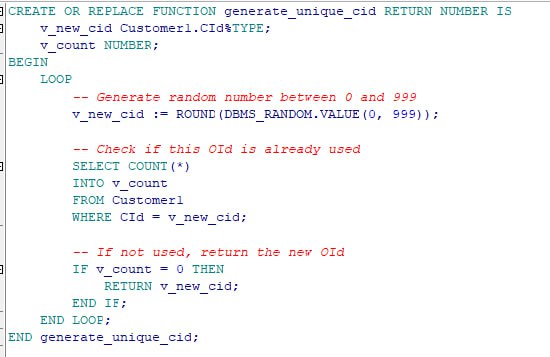
תמונה שמכילה טקסט, תרשים, צילום מסך, קו

התיאור נוצר באופן אוטומטי

**אופן האינטגרציה:**

**חיבור טבלאות הלקוחות:**

הוספנו פונקציה שמחזירה id פנוי כלומר מחזירה id שעוד לא קיים בטבלה:



תחילה הוספנו עמודה של כתובת לטבלת הלקוחות על מנת לקבל את התכונה שהייתה גם בתרשים שקיבלנו, אחר כך הוספנו עמודה לטבלת האירועים שתהיה המפתח הזר החדש במקום המפתח הזר הקיים לטבלת הלקוחות שלהם. הוספנו את כל הרשומות של הטבלה שהתקבלה לטבלה שלנו, אחר כך עדכנו את התכונות הקשורות בטבלת האירועים עם הid המתאים ולבסוף מחקנו את המפתח הזר מטבלת האירועים,הוספנו מפתח זר חדש שאנחנו יצרנו, מחקנו את העמודה של המפתח הזר הקודם ומחקנו את הטבלה שהתקבלה.

תמונה שמכילה טקסט, צילום מסך, מסמך, תפריט

התיאור נוצר באופן אוטומטי

**שינוי הקשר בין טבלת התשלומים תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטילטבלת** **ההזמנות**:

במקום שלתשלום יהיה מפתח זר של הזמנות להזמנות יש מפתח זר של תשלום וכך תשלום יכול לשמש גם כתשלום לאירועים וגם תשלום להזמנות.

הוספנו עמודה חדשה להזמנות שתשמש כמפתח הזר החדש לתשלומים עדכנו את כל הנתונים בהתאם בטבלת ההזמנות. מחקנו את העמודה הקודמת של המפתח הזר והפכנו את העמודה שהוספנו למפתח זר.

תמונה שמכילה טקסט, צילום מסך, מסמך, גופן

התיאור נוצר באופן אוטומטי

**מיזוג טבלאות התשלומים:**

הוספנו עמודה חדשה לטבלת התשלומים, הפכנו את התכונות השונות בין הטבלאות לאופציונליות. הוספנו עמודה שתשמש למפתח הזר החדש, עדכנו את הנתונים בטבלה. הוספנו מפתח זר חדש שאנחנו יצרנו מחקנו את העמודה של המפתח הזר הקודם, מחקנו את הטבלה שהתקבלה ולבסוף מחקנו את העמודה של מחיר סופי מטבלת האירועים כי העברנו את המחיר לתשלום המתאים של האירוע.

תמונה שמכילה טקסט, צילום מסך, מסמך, גופן

התיאור נוצר באופן אוטומטי

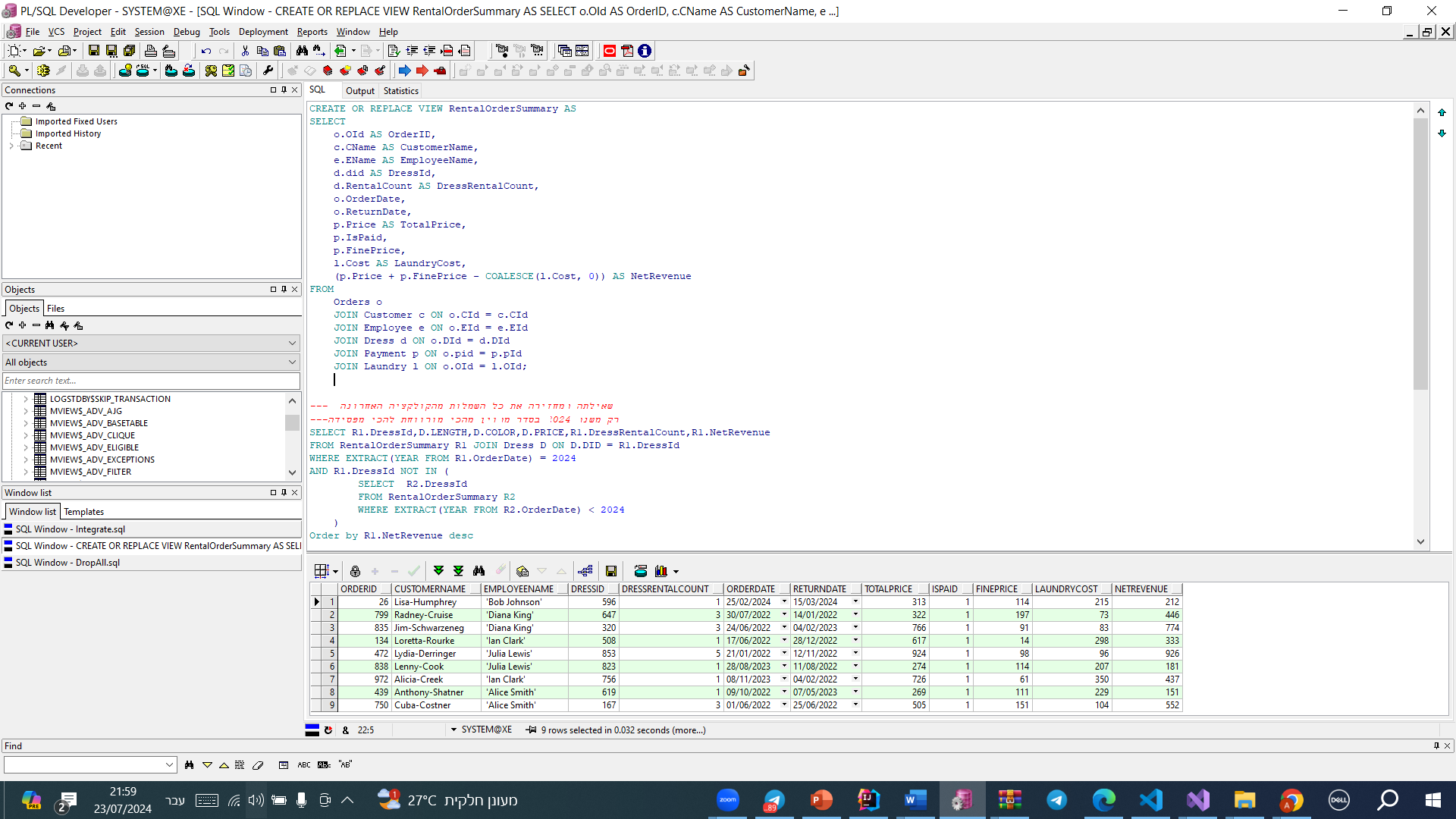
תמונה שמכילה טקסט, צילום מסך, מסמך, תפריט

התיאור נוצר באופן אוטומטי

**Views**

1. יצרנו VIEW אשר מראה עבור כל הזמנה איזה לקוח הזמין אותה מי העובד שטיפל בה מהו המחיר הכולל האם ההזמנה שולמה והאם היה עליה קנס ובצורה כללית יותר טבלה אשר נותנת את כל המידע עבור הזמנה.

יצירת הטבלה:



**שאילתות על הטבלה:**

**תמונה שמכילה טקסט, צילום מסך, מספר, גופן

התיאור נוצר באופן אוטומטישאילתה ראשונה**

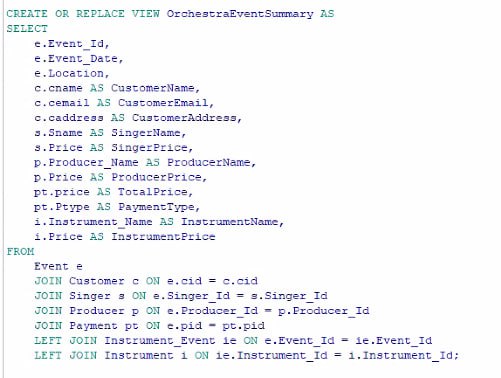
**שאילתה שניה**

**תמונה שמכילה טקסט, צילום מסך, תוכנה, דף אינטרנט

התיאור נוצר באופן אוטומטי**

**2**

**.** יצרנו VIEW אשר מראה עבור כל אירוע איזה לקוח הזמין אותה מה הוזמן מהו המחיר הכולל ובצורה כללית יותר טבלה אשר נותנת את כל המידע עבור אירוע.

****

**שאילתות על הטבלה:**

**תמונה שמכילה טקסט, צילום מסך, מספר, גופן

התיאור נוצר באופן אוטומטישאילתה ראשונה**

**שאילתה שניה**

**תמונה שמכילה טקסט, צילום מסך, תצוגה, תוכנה

התיאור נוצר באופן אוטומטי**