



SQL פרויקט גמר בבסיסי נתונים ושפת

מגיש: דניאל אלחרר
תעודת זהות: 326570520

מכללה: אורט סינגאפובסקי
תאריך: 16\04\2023

תוכן עניינים:

4. עולם התוכן.....	4
עולם התוכן שלי הוא ניהול חנות מחשבים:.....	4
5. האתגרים בעולם התוכן, מטרות בסיס הנתונים.....	5
מהם האתגרים העקרוניים בעולם התוכן שלכם, שבסיס הנתונים עשוי לסייע בפתרונם?.....	5
מהן המטרות אותן בסיס הנתונים שלכם מממש- המטרות צריכות לעזור להתמודד עם האתגרים של עולם התוכן?.....	5
6. המשתמשים של בסיס הנתונים.....	6
מי המשתמשים של בסיס הנתונים שלכם, לפי המטרות שהוגדרו?.....	6
7. הנתונים שיאוחסנו בבסיס הנתונים.....	7
ישויות:.....	
1. מוצרים - מזהה מוצר, שם מוצר, צבע, משקל, מחיר, יצרן, מלאי, מזהה קטגוריה.....	7
8. ישויות ו"טבלת איקסים".....	8
ישויות:.....	
1. מוצרים.....	8
טבלת איקסים:.....	8
erd: תרשים.....	9
10. סכמת בסיס נתונים.....	10
טבלאות ישויות:.....	10
מוצרים:.....	10
ספקים:.....	10
קטגוריות:.....	10
עובדים:.....	10
לקוחות:.....	11
הזמנות:.....	11
בן אדם:.....	11
טבלאות קשר:.....	11
מוצרים הזמנות:.....	11
טבלאות עזר:.....	11
טבלת מוצרי חודש של לקוח(עגלה חודשית):.....	11
פונקציות:.....	12
פונקציה בדיקה האם מוצר זמין במלאי:.....	12
פונקציה חישוב משכורת לעובד:.....	13
פונקציה חישוב מחיר של הזמנה מלאה:.....	15
פונקציה חישוב המכירות של מוצר מסוים:.....	16
פונקציה בדיקה האם לקוח הוא ותיק:.....	17
פונקציית בדיקה כמה הנחה ניתן לתת על מוצר:.....	18
פונקציה מילוי המלאי:.....	19
פרוצדורות:.....	21

פרוצדורת יצירת העגלה החודשית של לקוח:	21
שימוש:	
לפני:	22
פרוצדורת חישוב הזמנה אחרי הנחה:	23
פרוצדורת שלושת הקטגוריות הנמכרות ביותר:	25
פרוצדורת שלושת המוצרים הנמכרים ביותר:	27
טריגרים:	29
טריגר בדיקת דרך תשלום:	29
טריגר שינוי המלאי למוצר:	30
נספחים	31
טבלת מוצרים:	31
יצירה:	31
אכלוס:	31
טבלת ספקים:	33
יצירה:	33
אכלוס:	33
טבלת קטגוריות:	34
יצירה:	34
אכלוס:	34
טבלת עובדים:	35
יצירה:	35
אכלוס:	35
טבלת לקוחות:	36
יצירה:	36
אכלוס:	36
טבלת הזמנות:	37
יצירה:	37
אכלוס:	37
טבלת בן אדם:	39
יצירה:	39
אכלוס:	39
טבלת מוצרים-הזמנות:	40
יצירה:	40
אכלוס:	41
טבלת עגלת לקוח חודשית:	42
יצירה:	42
אכלוס:	42

עולם התוכן

עולם התוכן שלי הוא ניהול חנות מחשבים:

החנות מכילה מגוון מוצרי מחשב מקטגוריות שונות שאותם יכול לקוח לקנות ולשלם באמצעות כמה אפשרויות כמו: ויזה, אשראי, מזומן, כרטיס מתנה, חיוב ישיר (DEBIT), אפליקציה ועוד... בנוסף על החנות לשלם לעובדים שלה מה שמגיע להם כלומר שכר בסיס חודשי + חמישה אחוז על כל מכירה שעזרו בה, את המוצרים ניתן להזמין מספקים שונים.

האתגרים בעולם התוכן, מטרות בסיס הנתונים

מהם האתגרים העקרוניים בעולם התוכן שלכם, שבסיס הנתונים עשוי לסייע בפתרונם?

- ⇐ כמה מוצרים להזמין בעת ההזמנה מאיזה מוצר - ניתן לדעת בהתאם לכמות המחירות שנמכרו בחודש האחרון מאותו מוצר
- ⇐ כמה משלמים לעובד בסוף חודש?
- ⇐ האם לקוח יכול לקבל הנחה? במקרה ושאלה לגביה
- ⇐ האם להפסיק להזמין מוצר מסוים?

מהן המטרות אותן בסיס הנתונים שלכם מממש - המטרות צריכות לעזור להתמודד עם האתגרים של עולם התוכן?

- ⇐ זיהוי מוצרים נמכרים לעומת לא נמכרים
- ⇐ זיהוי האם ניתן לתת הנחה על מוצר ואם כן אז כמה
- ⇐ זיהוי הקטגוריות הכי נמכרות
- ⇐ מציאת העגלה של לקוח בחודש מסוים
- ⇐ זיהוי מצבים - לקוח ותיק, כמות מכירות של מוצר

המשתמשים של בסיס הנתונים

מי המשתמשים של בסיס הנתונים שלכם, לפי המטרות שהוגדרו?

לפי המטרות המוגדרות לבסיס הנתונים ניתן לראות כי בסיס הנתונים משרת את ההנהלה וגם את הלקוחות כך שיוכלו לדעת מה מתרחש בחנות בתקופות זמן מסוימות.

הנתונים שיאוחסנו בבסיס הנתונים

הנתונים אותם אתם מאחסנים בבסיס הנתונים שלכם, כדי לתמוך במטרות שהוגדרו.

ישויות:

1. **מוצרים** - מזהה מוצר, שם מוצר, צבע, משקל, מחיר, יצרן, מלאי, מזהה קטגוריה
2. **לקוחות** - מזהה לקוח, כתובת
3. **קטגוריות** - מזהה קטגוריה, שם קטגוריה, תיאור
4. **ספקים** - מזהה ספק, ניסיון
5. **הזמנות** - מזהה הזמנה, מזהה לקוח, תאריך הזמנה, שיטת תשלום, מזהה עובד
6. **עובדים** - מזהה עובד, משכורת, עבודה, תאריך התחלה

ישויות ו"טבלת איקסים"

ישויות:

1. מוצרים

2. לקוחות

3. קטגוריות

4. ספקים

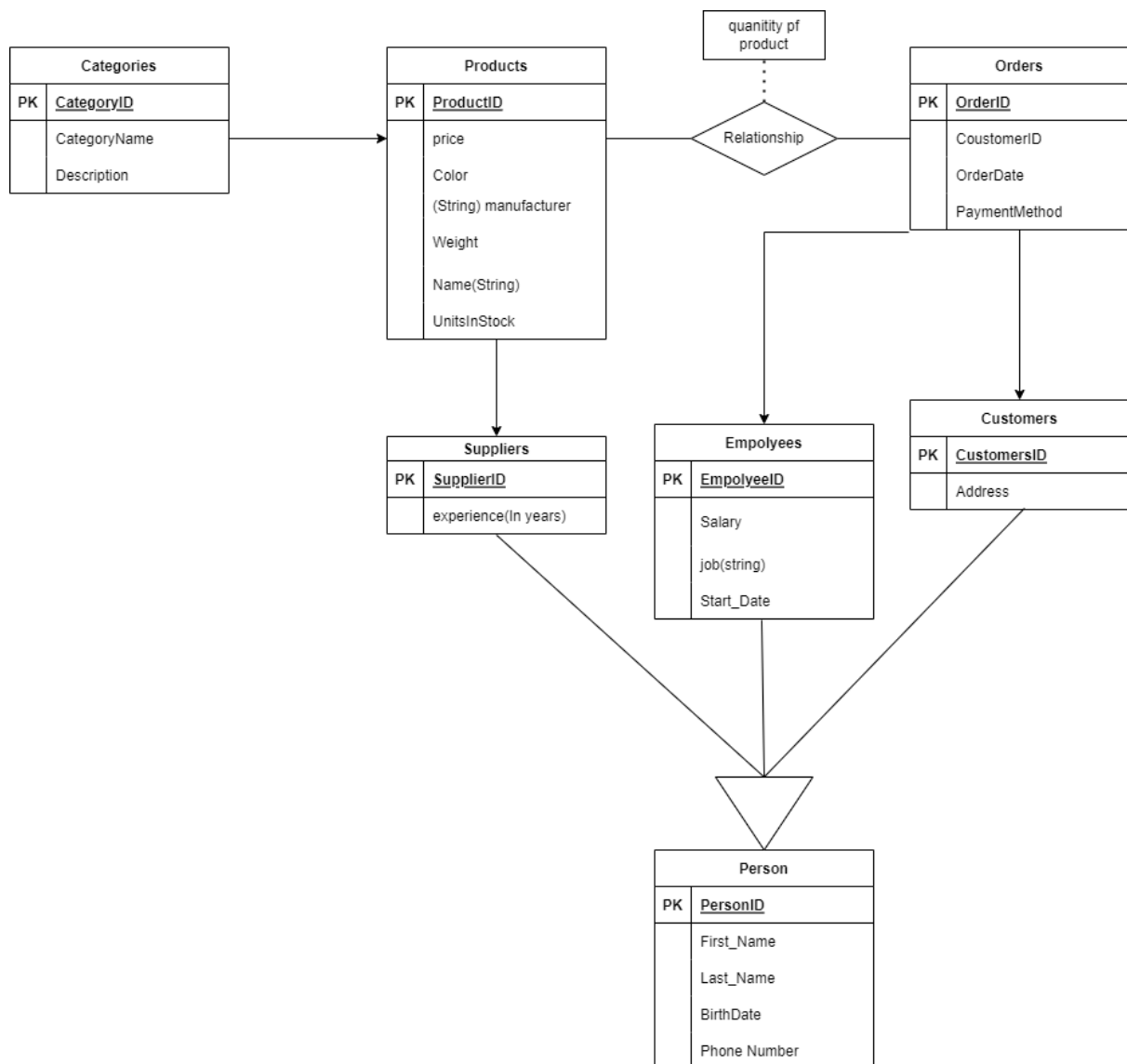
5. הזמנות

6. עובדים

טבלת איקסים:

	מוצרים	לקוחות	קטגוריות	ספקים	הזמנות	עובדים
מוצרים	X	אין קשר	m:n	1:n מוצר \rightarrow n	m:n	אין קשר
לקוחות	X	X	אין קשר	אין קשר	1:n לקוחות \rightarrow n	אין קשר
קטגוריות	X	X	X	אין קשר	אין קשר	אין קשר
ספקים	X	X	X	X	אין קשר	אין קשר
הזמנות	X	X	X	X	X	אין קשר
עובדים	X	X	X	X	X	X

תרשים :erd



סכמת בסיס נתונים

הטבלאות כפי שנגזרות מתוך ה erd כולל זיהוי מפתחות ראשיים, מפתחות זרים וטבלאות קשר לפי צורך

– ניתן לפרט כאן גם לגבי טבלאות העזר או views

– פרטו גם טריגרים, פונקציות ופרוצדורות

טבלאות ישויות:

מוצרים:

Weight	UnitsInStock	(String) manufacturer	Color	Size	Price	Name(String)	*SupplierID	<u>Product_ID</u>

ספקים:

experience	<u>*Person_ID</u>

קטגוריות:

Description	Category_Name	<u>Category_ID</u>

עובדים:

Start_Date	job(string)	Salary	<u>*Person_ID</u>

לקוחות:

Phone Number	Address	* <u>Person_ID</u>

הזמנות:

*employeeID	PaymentMethood	OrderDate	*Customer_ID	<u>Order_Id</u>

בן אדם:

BirthDate	Last_Name	First_Name	<u>Person_ID</u>

טבלאות קשר:**מוצרים הזמנות:**

Product_Amount	*Order_ID	*Product_ID

טבלאות עזר:**טבלת מוצרי חודש של לקוח(עגלה חודשית):**

UnitAmount	UnitPrice	ProductName	*OrderDate	<u>Order_Id</u>

פונקציות:

פונקציה בדיקה האם מוצר זמין במלאי:

הפונקציה מקבלת שם של מוצר ובודקת האם המוצר זמין במלאי או לא.

הסבר עסקי:

מטרת הפונקציה היא להחזיר להדפיס למשתמש האם מוצר זמין או לא כך לקוח יכול לדעת איזה מוצרים לא זמינים לקניה וההנהלה יכולה לדעת שנגמר לה המלאי של אותו המוצר וצריך להזמין.

הסבר טכני:

הפונקציה מחפשת את המוצר לפי השם באמצעות פקודות WHERE מכניסה את המשתנה unitsInStock למשתנה ובודקת אם המשתנה גדול מ 0 אם כן הפונקציה מדפיסה ומחזירה "Available" אחרת תדפיס "Not Available".

```
CREATE DEFINER=`root`@`location` FUNCTION
`function_checkProductAvailability`(productName varchar(45))
RETURNS varchar(50) CHARSET utf8mb4
DETERMINISTIC
BEGIN
    DECLARE stockAmount INT;
    DECLARE result varchar(50);
    select unitsInStock into stockAmount from products
    where products.name = productName limit 1;

    if stockAmount > 0 Then
        SET result = "Available";
    ELSE
        SET result = "Not Available";
    END IF;
    RETURN result;
END
```

שימוש:

```
select computershop.function_checkProductAvailability('NZXT H510')
as 'IsInStock';
```

	IsInStock
►	Available

פונקציה חישוב משכורת לעובד:

הפונקציה מקבלת מזהה עובד, חודש ושנה הפונקציה תחזיר את המשכורת של אותו עובד באותו חודש ושנה.

הסבר עסקי: מטרת הפונקציה היא להפיק את השכר החודשי של עובד באמצעות חישוב המשכורת הקבועה של העובד לחודש והוספה של 5% על כל מחירה שעזר עובד כך יכול בעל עסק לדעת כמה צריך לשלם לכל עובד.

הסבר טכני: תחילה הפונקציה מחפשת את השכר ההתחלתי של עובד לפי מזהה העובד באמצעות שאילתא פשוטה, לאחר מכן הפונקציה מבצעת JOIN בין טבלת הזמנות לבין טבלת הקשר הזמנות-מוצרים מחפשת את כל ההזמנות שאותו העובד השתתף בהם(עזר)

ועל כל הזמנה אנחנו משתמשים בפונקציה אחרת שנותנת לנו את סך כל ההזמנה ומחשבת 5 אחוז ממנה סוכמת ומוסיפה לשכר הדיפולטיבי וזהו השכר של העובד בסוף החודש.

```
CREATE DEFINER='root'@'localhost' FUNCTION
`function_employeeMonthlySalary`(employeeID INT, monthSalary INT,
yearSalary INT) RETURNS int
    DETERMINISTIC
BEGIN
    DECLARE total_salary DECIMAL(10,2) DEFAULT 0;
    DECLARE default_salary DECIMAL(10,2);

    IF monthSalary < 1 OR monthSalary > 12 THEN
        RETURN -1;
    ELSE
        -- Get default salary of employee
        SELECT salary INTO default_salary FROM employees WHERE personID
= employeeID;

        -- Calculate total salary of employee for given month and year
        SELECT SUM(function_GetOrderTotalPrice(products_orders.orderID)
* 0.05) INTO total_salary
        FROM products_orders
        INNER JOIN orders ON products_orders.orderID = orders.orderID
        WHERE orders.employeeID = employeeID AND MONTH(orders.OrderDate)
= monthSalary AND YEAR(orders.OrderDate) = yearSalary;

        IF total_salary IS NULL THEN
            RETURN default_salary;
        ELSE
            RETURN total_salary + default_salary;
        END IF;
    END IF;
```

```
END IF;  
END
```

שימוש:

```
select  
computershop.function_employeeMonthlySalary(310248652, 6,  
2022);
```

	computershop.function_employeeMonthlySalary(3 6, 2022)
▶	3295

פונקציה חישוב מחיר של הזמנה מלאה:

הסבר עסקי: מטרת הפונקציה היא להציג מהו המחיר הכולל של הזמנה שלמה, פונקציה שכזו יכולה לעזור גם ללקוח וגם להנהלה, הלקוח יכול לדעת כמה יצאה לו ההזמנה, והנהלה יכולה לראות כמה היא הכניסה מאותו לקוח.

הסבר טכני: הפונקציה משתמשת בשאילתת JOIN שמאחדת את הטבלאות מוצרים ומוצרים-הזמנות על מנת לקחת את המחיר של מוצר להכפיל אותו בכמות שהוזמנה ולבצע הכל על כל המוצרים בהזמנה וסוכם את כל החישוב וזה המחיר הסופי של ההזמנה.

```
CREATE DEFINER=`root`@`localhost` FUNCTION
`function_GetOrderTotalPrice` (orderID int) RETURNS int
DETERMINISTIC
BEGIN
    DECLARE total_price DECIMAL(10,2) DEFAULT 0;

    -- Calculate total price of the order
    SELECT SUM(p.price * po.productAmount) INTO total_price
    FROM products p
    INNER JOIN products_orders po ON p.productID = po.productID
    WHERE po.orderID = orderID;

    RETURN total_price;
END
```

שימוש:

```
select computershop.function_GetOrderTotalPrice(17);
```

	computershop.function_GetOrderTotalPrice(17)
▶	2080

פונקצית חישוב המכירות של מוצר מסוים:

הסבר עסקי: מטרת הפונקציה היא להציג את כמות היחידות שנמכרו ממוצר מסוים בחודש מסוים ובשנה מסוימת הפונקציה הזאת יכולה להראות לנו איזה מוצרים נחשבים "חזקים" כלומר נמכרים יותר לצד איזה מוצרים נמכרים פחות.

הסבר טכני: הפונקציה משתמש בJOIN כפול בין הטבלאות מוצרים-הזמנות, מוצרים, הזמנות על מנת למצוא באותו החודש שקיבלנו מהמשתמש לפונקציה את כל ההזמנות שבהן הופיע המוצר שנשלח לפונקציה וסוכמים את השדה PRODUCTAMOUNT כדי למצוא כמה פעמים נמכר אותו המוצר בהזמנות של אותו חודש.

```
CREATE DEFINER=`root`@`localhost` FUNCTION
`function_GetProductSales`(productName VARCHAR(45), yearCheck INT,
monthCheck INT) RETURNS int
DETERMINISTIC
BEGIN
    DECLARE productSales INT DEFAULT 0;

    SELECT SUM(products_orders.productAmount) INTO productSales
    FROM products_orders
    JOIN products ON products_orders.productID =
products.productID
    JOIN orders ON products_orders.orderID = orders.orderID
    WHERE products.name = productName
    AND YEAR(orders.OrderDate) = yearCheck
    AND MONTH(orders.OrderDate) = monthCheck;

    RETURN productSales;
END
```

שימוש:

```
select computershop.function_GetProductSales('Corsair Headset Stand',
2022, 6);
```

	computershop.function_GetProductSales('Corsair Headset Stand', 2022, 6)
▶	2

פונקצית בדיקה האם לקוח הוא ותיק:

הסבר עסקי: מטרת הפונקציה היא בדיקה אם לקוח הוא ותיק, על מנת לברר אם הוא זכאי להנחה של 5 אחוז, דבר זה ישמר לנו על לקוחות קבועים שיעדיפו להמשיך לקנות אצלנו.

הסבר טכני: הפונקציה מקבלת מזהה לקוח, איך מזהים ותק(ימים), הפונקציה משתמשת בSELECT פשוט על מנת למצוא את הקנייה הראשונה שעשה לקוח אצלנו בחנות אנחנו מוציאים את התאריך ומשתמשים בפונקציית DATEDIFF על הפעם הראשונה שקנה ועל התאריך של היום ואם ההפרש בימים גדול מהותק שקיבלנו הלקוח נחשב כ-ותיק ויהיה זכאי להנחה.

```
CREATE DEFINER=`root`@`localhost` FUNCTION
`function_IsCustomerVeteran`(customerID INT, vetranDays INT)
RETURNS tinyint
DETERMINISTIC
BEGIN
    DECLARE firstOrderDate DATE;
    DECLARE diff INT;

    SELECT OrderDate INTO firstOrderDate
    FROM orders
    WHERE CustomerID = customerID
    ORDER BY OrderDate ASC
    LIMIT 1;

    SET diff = DATEDIFF(CURDATE(), firstOrderDate);

    IF diff > vetranDays THEN
        RETURN 1;
    ELSE
        RETURN 0;
    END IF;
END
```

שימוש:

```
select computershop.function_IsCustomerVeteran(320607254, 730);
```

	computershop.function_IsCustomerVeteran(320607254, 730)
▶	1

פונקציית בדיקה כמה הנחה ניתן לתת על מוצר:

הסבר עסקי: מטרת הפונקציה היא במידה והלקוח יבקש הנחה נוכל לבדוק כמה הנחה ניתן להביא על אותו מוצר בהתאם לכמות המכירות שעשה אותו מוצר באותו חודש, דבר זה גם מראה על שימור לקוחות אם הם יראו שיש הנחות הם ירצו לבוא עוד.

הסבר טכני: הפונקציה מקבלת שם של מוצר, שנה וחודש הפונקציה משתמשת בפונקציה שמוצאת את כמות היחידות שנמכרו מאותו מוצר ככל שנמכר פחות יש יותר הנחה אם נמכרו פחות מ-10 יחידות המוצר זכאי ל-15 אחוז הנחה אם נמכרו בין 10 ל-20 המוצר זכאי ל-10 אחוז הנחה ואם נמכרו בין 20 ל-40 המוצר זכאי ל-5 אחוז הנחה אחרת המוצר לא זכאי להנחה.

```
CREATE DEFINER=`root`@`localhost` FUNCTION
`function_ProductAvailableDiscount`(productName VARCHAR(45),
yearCheck INT, monthCheck INT) RETURNS int
DETERMINISTIC
BEGIN
    DECLARE salesAmount INT DEFAULT 0;
    DECLARE discount INT DEFAULT 0;

    SELECT function_GetProductSales(productName, yearCheck,
monthCheck)
    INTO salesAmount;

    CASE
        WHEN salesAmount < 10 THEN SET discount = 15;
        WHEN salesAmount >= 10 AND salesAmount <= 20 THEN SET
discount = 10;
        WHEN salesAmount > 20 AND salesAmount <= 40 THEN SET
discount = 5;
    END CASE;

    RETURN discount;
END
```

שימוש:

```
select computershop.function_ProductAvailableDiscount('AMD Ryzen 7', 2022, 6);
```

computershop.function_ProductAvailableDiscount(Ryzen 7, 2022, 6)
0

computershop.function_ProductAvailableDiscount(OfficeJet Pro', 2022, 6)
15

```
select computershop.function_ProductAvailableDiscount('HP OfficeJet Pro', 2022, 6);
```

פונקצית מילוי המלאי:

הסבר עסקי: מטרת הפונקציה היא לדעת כמה להזמין מאותו מוצר ביחס לכמה שאמורים להזמין בהתאם למה שקרה בחודש האחרון לפי כמות המכירות.

הסבר טכני: הפונקציה משתמשת בשתי פרוצדורות, פרוצדורה ראשונה מחזירה לנו את שלושת הקטגוריות שנמכרו הכי הרבה בחודש האחרון הפרוצדורה השנייה עושה אותו דבר רק למוצרים, לאחר מכן הפונקציה רק בודקת האם המוצר נמצא בין 3 המוצרים הנמכרים ביותר בחודש האחרון אם כן ניקח את הMINIMALSTOCK ונזמין 25% יותר מזה אותו דבר נעשה אם הקטגוריה של המוצר נמצאת בין הקטגוריות הנמכרות ביותר אבל אם המוצר עונה על שניהם נזמין 50% יותר מהMINIMALSTOCK.

```
CREATE DEFINER='root'@'localhost' FUNCTION
`function_RefillProductStock`(productID INT, minimalStock INT) RETURNS int
DETERMINISTIC
BEGIN
    DECLARE checkDate DATE;
    DECLARE prod1 VARCHAR(45);
    DECLARE prod2 VARCHAR(45);
    DECLARE prod3 VARCHAR(45);
    DECLARE cat1 VARCHAR(45);
    DECLARE cat2 VARCHAR(45);
    DECLARE cat3 VARCHAR(45);
    DECLARE ProductNameToCheck VARCHAR(45);
    DECLARE ProductCategoryToCheck VARCHAR(45);
    DECLARE supplyMul INT DEFAULT 1;
    SET checkDate = DATE(NOW());
    CALL procedure_TopThreeSellingProducts(YEAR(checkDate), MONTH(checkDate),
@prod1, @prod2, @prod3);
    CALL procedure_topThreeSellersCategories(YEAR(checkDate), MONTH(checkDate),
@cat1, @cat2, @cat3);
    SET ProductNameToCheck = (SELECT products.name FROM products WHERE productID
= productID limit 1);
    SET ProductCategoryToCheck = (SELECT categoryName FROM categories c JOIN
products p ON c.categoryID = p.CategoryID WHERE p.productID = productID limit
1);
    IF ProductNameToCheck IN (@prod1, @prod2, @prod3) THEN
        SET supplyMul = supplyMul + 0.25;
    END IF;
    IF ProductCategoryToCheck IN (@cat1, @cat2, @cat3) THEN
        SET supplyMul = supplyMul + 0.25;
    END IF;
    UPDATE products SET unitsInStock = (minimalStock * supplyMul) WHERE
products.productID = productID;
```

```

SET minimalStock = minimalStock*supplyMul;
RETURN minimalStock;
END

```

שימוש:

לפני:

27	MSI Gaming GeFor...	black	1.5	499.99	MSI	6	8
----	---------------------	-------	-----	--------	-----	---	---

```
select computershop.function_RefillProductStock(27, 15);
```

אחרי:

27	MSI Gaming GeFor...	black	1.5	499.99	MSI	15	8
----	---------------------	-------	-----	--------	-----	----	---

פרוצדורות:

פרוצדורת יצירת העגלה החודשית של לקוח:

הסבר עסקי: מטרת הפרוצדורה היא ליצור טבלה המראה את המוצרים שקנה לקוח בחודש מסוים.

הסבר טכני: נכריז CURSOR שיכיל את ה-ID של ההזמנות של אותו לקוח, תאריך הזמנה, שם מוצר, מחיר וכמה הוזמן הדבר יעשה באמצעות JOIN כפול בין טבלאות הזמנות, מוצרים והזמנות - מוצרים לאחר מכן זורקים את הטבלה יוצרים אותה מחדש בשביל הלקוח החדש לאחר מכן פותחים את ה-CURSOR ועושים INSERTS לתוך הטבלה הזאת.

```
CREATE DEFINER='root'@'localhost' PROCEDURE
`procedure_CustomerMonthlyOrders` (IN customerID int, IN monthReport
int, IN yearReport int)
BEGIN
    --
    DECLARE done INT DEFAULT FALSE;
    DECLARE orderID INT;
    DECLARE OrderDate DATE;
    DECLARE prodName VARCHAR(255);
    DECLARE price INT;
    DECLARE productAmount INT;

    DECLARE CustomerMonthlyCart_Cursor CURSOR FOR
    SELECT orders.orderID as 'orderID', orders.OrderDate as
'OrderDate', products.name as 'name', products.price 'price',
products_orders.productAmount as 'productAmount'
    FROM orders
    JOIN products_orders ON orders.orderID =
products_orders.orderID
    JOIN products ON products_orders.productID =
products.productID
    WHERE orders.Customer_ID = customerID
    AND YEAR(orders.OrderDate) = yearReport
    AND MONTH(orders.OrderDate) = monthReport
    ORDER BY orders.orderID;
```

```

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

DROP TABLE IF EXISTS TEMP_TABLE_CustomerMonthlyCart;
CREATE TABLE TEMP_TABLE_CustomerMonthlyCart(OrderID int
unsigned not null, Order_Date date not null, ProductName
varchar(255) not null, UnitPrice int not null, UnitAmount int not
null);

OPEN CustomerMonthlyCart_Cursor;

CustomerMonthlyCart_loop: LOOP
    FETCH CustomerMonthlyCart_Cursor INTO orderID,
OrderDate, prodName, price, productAmount;

    IF done THEN
        LEAVE CustomerMonthlyCart_loop;
    END IF;

    INSERT INTO TEMP_TABLE_CustomerMonthlyCart(OrderID,
Order_Date, ProductName, UnitPrice, UnitAmount)
        VALUES(orderID, OrderDate, prodName, price,
productAmount);
    END LOOP;

CLOSE CustomerMonthlyCart_Cursor;

SELECT * FROM TEMP_TABLE_CustomerMonthlyCart;

END

```

שימוש:

לפני:

OrderID	Order_Date	ProductName	UnitPrice	UnitAmount
---------	------------	-------------	-----------	------------

call computershop.procedure_CustomerMonthlyOrders(325222058, 6, 2022);

אחרי:

42 מתוך 22 עמוד

OrderID	Order_Date	ProductName	UnitPrice	UnitAmount
26	2022-06-12	Seagate BarraCuda	80	2
26	2022-06-12	Secretlab Omega	350	1
39	2022-06-25	Logitech G Pro	70	4
39	2022-06-25	Netgear Nighthawk	130	3

פרוצדורת חישוב הזמנה אחרי הנחה:

הסבר עסקי: מטרת הפרוצדורה היא במידה וביקש לקוח הנחה על ההזמנה שלו אנחנו רוצים לראות מה אנחנו יכולים לתת לו כדי שירצה לחזור פעם הבאה כשיראה שיש פה הנחות.

הסבר טכני: הפרוצדורה משתמש בCURSOR הנבנה על ידי JOIN בין טבלת מוצרים והזמנות-מוצרים CURSOR משתמש כעזר בפונקציה של כמה הנחנה יכול מוצר לקבל לאחר מכן פותחים את CURSOR וסוכמים את כל המחירים החדשים של כל מה שהזמין וזה ההזמנה אחרי הנחה.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `procedure_OrderDiscount`(IN orderID
INT, IN orderDate DATE, OUT discountedPrice DECIMAL(10,2))
```

```
BEGIN
```

```
    DECLARE totalPrice DECIMAL(10,2) DEFAULT 0;
    DECLARE productPrice DECIMAL(10,2) DEFAULT 0;
    DECLARE productAmount INT DEFAULT 0;
    DECLARE productDiscount DECIMAL(10,2) DEFAULT 0;
    DECLARE customerID INT;
    DECLARE done INT DEFAULT FALSE;
```

```
    -- Loop over each product in the order and apply its respective discount
percentage
```

```
    DECLARE prodctAvailableDiscount_Cursor CURSOR FOR
        SELECT p.Price, po.ProductAmount,
function_ProductAvailableDiscount(p.name, YEAR(orderDate), MONTH(orderDate))
        FROM products p
        INNER JOIN products_orders po ON p.ProductID = po.ProductID
        WHERE po.OrderID = orderID;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
```

```
    -- Calculate total price of the order
```

```
    OPEN prodctAvailableDiscount_Cursor;
```

```
    prodctAvailableDiscount_loop: LOOP
        FETCH prodctAvailableDiscount_Cursor INTO productPrice, productAmount,
productDiscount;
        IF done THEN
            LEAVE prodctAvailableDiscount_loop;
        END IF;

        IF (productPrice IS NULL) THEN
```

```

        LEAVE prodctAvailableDiscount_loop;
    END IF;
    select totalPrice;
    SET totalPrice = totalPrice + ((productPrice * ((100 -
productDiscount)/100)) * productAmount);

END LOOP;

CLOSE prodctAvailableDiscount_Cursor;

#check if the customer is veteran in order to give him additional discount
of 5%
select customerID into customerID from orders where orders.OrderID =
orderID;

IF(function_IsCustomerVeteran(customerID) = 1) THEN
    SET discountedPrice = totalPrice*0.95;
ELSE
    SET discountedPrice = totalPrice;
END IF;
END

```

שימוש:

```

3 • select function_GetOrderTotalPrice(17);
4 • set @discountedPrice = 0;
5 • call computershop.procedure_OrderDiscount(17, '2022-06-03', @discountedPrice);
6 • select @discountedPrice;
7

```

	function_GetOrderTotalPrice(17)
▶	2080

	totalPrice
▶	1359.99

פרוצדורת שלושת הקטגוריות הנמכרות ביותר:

הסבר עסקי: מטרת הפרוצדורה היא להציג ללקוח את הקטגוריות הכי נמכרות שלנו שזו בעצם המלצה במה החנות שלנו מצטיינת, אמנם להנהלה ישנם שתי אפשרויות להבין מזה אפשר להבין אילו קטגוריות צריך לקדם ואפשר להבין שמהקטגוריות האלה כדי להזמין יותר כי הקהל מתחבר לקטגוריות האלו.

הסבר טכני: הפרוצדורה יוצר CURSOR המייצג את כמות המוצרים שנמכרות מאותה קטגוריה בחודש מסוים מסדרת מהגדול לקטן ומשאירה רק 3 לאחר מכן מכניסה את 3 הקטגוריות ל3 משתנים שונים ומחזירה.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE
`procedure_topThreeSellersCategories`(IN yearCheck INT, IN monthCheck
INT, OUT cat1 VARCHAR(45), OUT cat2 VARCHAR(45) , OUT cat3 VARCHAR(45))
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE temp_category VARCHAR(45) default null;

    DECLARE TopCategories_Cursor CURSOR FOR
    SELECT c.categoryName
    FROM (
        SELECT distinct productID, SUM(productAmount) AS total_amount
        FROM products_orders
        JOIN orders ON products_orders.orderID = orders.orderID
        WHERE YEAR(orders.OrderDate) = yearCheck
        AND MONTH(orders.OrderDate) = monthCheck
        GROUP BY productID
        ORDER BY total_amount DESC
        LIMIT 3
    ) po
    JOIN products p ON po.productID = p.productID
    JOIN categories c ON p.categoryID = c.categoryID
    GROUP BY p.categoryID;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    set cat1 = null;
    set cat2 = null;
    set cat3 = null;

    OPEN TopCategories_Cursor;
    TopCategories_loop : LOOP
        FETCH TopCategories_Cursor INTO temp_category;
```

```

        IF done THEN
            LEAVE TopCategories_loop;
        END IF;

        IF cat1 IS NULL THEN
            SET cat1 = temp_category;
        ELSEIF cat2 IS NULL THEN
            SET cat2 = temp_category;
        ELSE
            SET cat3 = temp_category;
        END IF;
    END LOOP;
CLOSE TopCategories_Cursor;
END

```

שימוש:

```

1 • set @cat1 = '0';
2 • set @cat2 = '0';
3 • set @cat3 = '0';
4 • call computershop.procedure_topThreeSellersCategories(2022, 6, @cat1, @cat2, @cat3);
5 • select @cat1, @cat2, @cat3;
6

```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	@cat1	@cat2	@cat3
►	Mouses	Speakers	NULL

פרוצדורת שלושת המוצרים הנמכרים ביותר:

הסבר עסקי: מטרת הפרוצדורה היא להציג ללקוח את המוצרים הכי נמכרים שלנו שזו בעצם המלצה איזה מוצרים אצלנו בחנות הכי טובים, אמנם להנהלה ישנם שתי אפשרויות להבין מזה אפשר להבין אילו מוצרים אפשר לקדם ואפשר להבין שמהמוצרים האלה כדי להזמין יותר כי הקהל מתחבר למוצרים האלה.

הסבר טכני: הפרוצדורה יוצר CURSOR המייצג את כמות היחידות שנמכר מכל מוצר בחודש מסוים מסדרת את המוצריפ מהגדול לקטן לפי כמות היחידות ומשאירה רק 3 לאחר מכן מכניסה את 3 המוצרים ל3 משתנים שונים ומחזירה אותם.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE
`procedure_TopThreeSellingProducts`(IN yearCheck INT, IN monthCheck INT,
OUT prodName1 VARCHAR(45), OUT prodName2 VARCHAR(45), OUT prodName3
VARCHAR(45))
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE prodName VARCHAR(45);

    DECLARE TopSellingProducts_Cursor CURSOR FOR
    SELECT p.name
    FROM products_orders po
    JOIN products p ON po.productID = p.productID
    JOIN orders o ON po.orderID = o.orderID
    WHERE YEAR(o.OrderDate) = yearCheck AND MONTH(o.OrderDate) =
monthCheck
    GROUP BY po.productID
    ORDER BY SUM(po.productAmount) DESC
    LIMIT 3;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    SET prodName1 = NULL;
    SET prodName2 = NULL;
    SET prodName3 = NULL;

    OPEN TopSellingProducts_Cursor;

    TopSellingProducts_loop: LOOP
        FETCH TopSellingProducts_Cursor INTO prodName;

        IF done THEN
            LEAVE TopSellingProducts_loop;
```

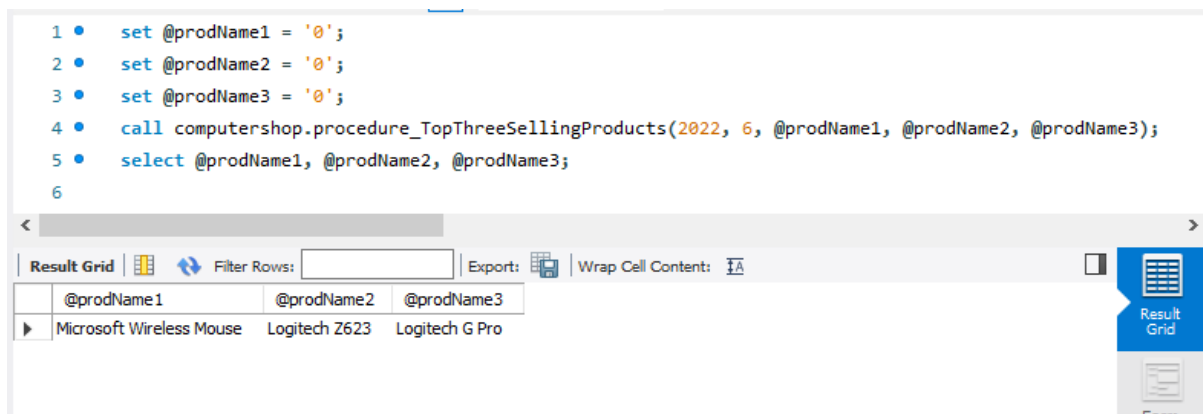
```
END IF;

IF prodName1 IS NULL THEN
    SET prodName1 = prodName;
ELSEIF prodName2 IS NULL THEN
    SET prodName2 = prodName;
ELSEIF prodName3 IS NULL THEN
    SET prodName3 = prodName;
END IF;
END LOOP;

CLOSE TopSellingProducts_Cursor;

END
```

שימוש:



The screenshot displays a SQL query execution window. The query is as follows:

```
1 • set @prodName1 = '0';
2 • set @prodName2 = '0';
3 • set @prodName3 = '0';
4 • call computershop.procedure_TopThreeSellingProducts(2022, 6, @prodName1, @prodName2, @prodName3);
5 • select @prodName1, @prodName2, @prodName3;
6
```

Below the query, the 'Result Grid' is visible, showing the output of the select statement:

@prodName1	@prodName2	@prodName3
Microsoft Wireless Mouse	Logitech Z623	Logitech G Pro

טריגרים:

טריגר בדיקת דרך תשלום

הסבר טכני: הטריגר בודק לפני הכנסת ערכים לטבלת הזמנות האם שיטת התשלום עומדת בתקנים מבין דרכי התשלום, משתמשים בתנאי שבודק האם מה שיטה החדשה נמצאת בשיטות שברשימה אם כן אז אפשרי להיכנס אם לא השורה החדשה שהייתה אמור להיות תהיה מ-NULL.

```
CREATE DEFINER=`root`@`localhost` TRIGGER
`orders_payment_method_check` BEFORE INSERT ON `orders` FOR EACH
ROW BEGIN
    DECLARE payment_method_valid TINYINT DEFAULT 0;

    IF NEW.PaymentMethod IN ('Visa', 'Gift Card', 'Credit',
    'Cash', 'App') THEN
        SET payment_method_valid = 1;
    END IF;

    IF NOT payment_method_valid THEN
        SET NEW = NULL;
    END IF;
END
```

טריגר שינוי המלאי למוצר:

הסבר טכני: הטריגר מעדכן את כמות היחידות שנשארו בחנות לאחר הכנסה חדשה לטבלת הזמנות מוצרים, נוריד מההUNITSINSTOCK של מוצר מסוים את כמות היחידות שנקנו מאותו מוצר.

```
CREATE DEFINER=`root`@`localhost` TRIGGER
`products_orders_AFTER_INSERT` AFTER INSERT ON
`products_orders` FOR EACH ROW BEGIN
    UPDATE products
    SET unitsInStock = unitsInStock - NEW.productAmount
    WHERE productID = NEW.productID;
END
```

נספחים

טבלת מוצרים:

יצירה:

```
CREATE TABLE `products` (
  `productID` int NOT NULL AUTO_INCREMENT,
  `name` varchar(45) NOT NULL,
  `color` varchar(45) NOT NULL,
  `weight` double NOT NULL,
  `price` double NOT NULL,
  `manufacturer` varchar(45) NOT NULL,
  `unitsInStock` varchar(45) NOT NULL,
  `CategoryID` int NOT NULL,
  PRIMARY KEY (`productID`),
  UNIQUE KEY `productID_UNIQUE` (`productID`),
  KEY `FR_ProdCat_CategoryID_idx` (`CategoryID`),
  CONSTRAINT `FR_ProdCat_CategoryID` FOREIGN KEY (`CategoryID`)
REFERENCES `categories` (`categoryID`)
) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

אכלוס:

```
INSERT INTO `products` VALUES (1,'HP
Pavilion','silver',2.5,899.99,'HP','50',18),(2,'Logitech G
Pro','black',0.2,69.99,'Logitech','25',3),(3,'AMD Ryzen
7','silver',0.5,329.99,'AMD','10',7),(4,'Dell
Ultrasharp','black',3.5,449.99,'Dell','15',1),(5,'Razer
Blackwidow','black',1.5,119.99,'Razer','20',2),(6,'SteelSeries
Arctis','white',0.6,99.99,'SteelSeries','30',4),(7,'Nvidia RTX
3080','black',1.2,799.99,'Nvidia','5',8),(8,'Bose
Companion','white',1.8,249.99,'Bose','10',5),(9,'Netgear
Nighthawk','black',0.8,129.99,'Netgear','12',14),(10,'Seagate
BarraCuda','silver',0.6,79.99,'Seagate','8',12),(11,'Microsoft Wireless
Mouse','black',0.1,29.99,'Microsoft','40',3),(12,'HP OfficeJet
Pro','white',9,299.99,'HP','5',17),(13,'Corsair Headset
Stand','black',0.3,39.99,'Corsair','15',4),(14,'Secretlab
Omega','black',30,349.99,'Secretlab','7',15),(15,'Asus ROG
Zephyrus','gray',2.5,1599.99,'Asus','8',18),(16,'Logitech
Z623','black',4,119.99,'Logitech','20',5),(17,'Logitech
Z906','black',20,395,'Logitech','38',5),(18,'Xbox Elite Wireless
Controller','black',0.4,179.99,'Microsoft','12',13),(19,'HyperX Fury
S','black',0.5,19.99,'HyperX','30',3),(20,'Logitech
G432','black',0.8,49.99,'Logitech','15',4),(21,'Redragon
S101-BA','black',2,29.99,'Redragon','23',2),(22,'NZXT
H510','white',7.5,79.99,'NZXT','10',15),(23,'Noctua
NH-D15','brown',1.3,89.99,'Noctua','5',11),(24,'Corsair Vengeance
```

```
LPX','black',0.1,69.99,'Corsair','15',9),(25,'ASUS Prime  
Z590-A','black',1.5,249.99,'ASUS','8',10),(26,'EVGA SuperNOVA  
850W','black',2,149.99,'EVGA','10',15),(27,'MSI Gaming GeForce RTX  
;(3060','black',1.5,499.99,'MSI','6',8
```


טבלת ספקים:**יצירה:**

```
CREATE TABLE `suppliers` (  
  `personID` int NOT NULL,  
  `experience` int NOT NULL,  
  PRIMARY KEY (`personID`),  
  CONSTRAINT `INH_SuppPer_personID` FOREIGN KEY (`personID`) REFERENCES  
  `person` (`personID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

אכלוס:

```
INSERT INTO `suppliers` VALUES  
(321278112,2),(323600844,3),(323616182,2),(324107886,1),(325222058,5),(3  
26015663,6),(326249751,10),(326570520,1),(326672599,1),(327772385,3);
```

טבלת קטגוריות:

יצירה:

```
CREATE TABLE `categories` (
  `categoryID` int NOT NULL,
  `categoryName` varchar(45) NOT NULL,
  `description` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`categoryID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

אכלוס:

```
INSERT INTO `categories` VALUES (1,'Screens','electronic displays that
allow users to view digital content'),(2,'Keyboards','input devices that
allow users to input text and commands'),(3,'Mouses','input devices that
allow users to move a cursor on a computer'),(4,'Headsets','Headsets are
audio equipment that combines headphones'),(5,'Speakers','Speakers are
audio devices'),(6,'Mics','Mics are input devices that allow users to
record and transmit audio content'),(7,'CPU','The CPU is the primary
component of a computer that performs most of the processing and
calculations for running software and operating systems. '),(8,'GPU','The
GPU is a specialized processor that is designed to handle complex
calculations and graphics rendering'),(9,'RAM','RAM is a type of
computer memory that provides fast access to frequently used data and
instructions'),(10,'MotherBoard','The motherboard is the main circuit
board in a computer that connects and controls all of the other
components'),(11,'Fans','Fans are components that are used to cool down
computer components'),(12,'Discs','Discs are storage media that use
laser technology to read and write digital
content'),(13,'Consoles','Consoles are dedicated gaming devices that
connect to a TV or monitor and allow users to play video
games'),(14,'Ethernet','Ethernet is a type of wired networking
technology that allows devices to connect to a local area network (LAN)
or the internet'),(15,'Case','The case is the outer shell of a computer
that encloses and protects the internal components, typically made of
metal or plastic'),(16,'Cables','Cables are used to connect different
components in a computer or electronic device'),(17,'Printers','Printers
are output devices that allow users to produce physical copies of
digital content, such as text, images, or graphics'),(18,'Laptops','
Laptops are portable computers that are designed for use on the go,
typically featuring a built-in screen, keyboard, and touchpad or
pointing stick. '),(19,'Desktops','Desktops are stationary computers that
are designed for use at a desk or workstation');
```

טבלת עובדים:**יצירה:**

```
CREATE TABLE `employees` (
  `personID` int NOT NULL AUTO_INCREMENT,
  `salary` int NOT NULL,
  `job` varchar(45) NOT NULL,
  `startDate` date NOT NULL,
  PRIMARY KEY (`personID`),
  UNIQUE KEY `personID_UNIQUE` (`personID`),
  CONSTRAINT `INH_EmpPer_personID` FOREIGN KEY (`personID`) REFERENCES
`person` (`personID`)
) ENGINE=InnoDB AUTO_INCREMENT=313775423 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

אכלוס:

```
INSERT INTO `employees` VALUES
(300877090,4500,'manager','2017-07-21'),(301789107,3400,'guard','2018-05-12'),(301836035,3150,'cashier','2019-05-12'),(302822727,3200,'salesman','2020-02-18'),(304446235,3400,'guard','2018-05-03'),(305298407,3200,'salesman','2021-12-12'),(306621797,3600,'stockkeeper','2020-08-18'),(307671623,4200,'manager','2015-07-29'),(308750124,2800,'cleaner','2018-04-06'),(310248652,3200,'salesman','2018-02-03'),(312290162,3150,'cashier','2018-09-07'),(313084174,2800,'cleaner','2018-08-16'),(313775422,3600,'stockkeeper','2015-08-16');
```

טבלת לקוחות:

יצירה:

```
CREATE TABLE `customers` (  
  `personID` int NOT NULL,  
  `address` varchar(45) NOT NULL,  
  PRIMARY KEY (`personID`),  
  CONSTRAINT `INH_personID` FOREIGN KEY (`personID`) REFERENCES `person`  
  (`personID`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

אכלוס:

```
INSERT INTO `customers` VALUES (315287490,'816 Springview  
Trail'),(315974582,'3 Almo Place'),(317460529,'6 Doe Crossing  
Alley'),(318148468,'4 Marcy Alley'),(318447589,'161 John Wall  
Alley'),(319737108,'9 Towne Junction'),(320607254,'0672 Elgar  
Parkway'),(321278112,'673 Fair Oaks Plaza'),(323600844,'12689 Mallard  
Court'),(323616182,'42166 International Road'),(324107886,'8 Southridge  
Alley'),(325222058,'8 Goodland Place'),(326015663,'32 Carioca Park');
```

טבלת הזמנות:

יצירה:

```
CREATE TABLE `orders` (
  `orderID` int NOT NULL AUTO_INCREMENT,
  `Customer_ID` int unsigned NOT NULL,
  `OrderDate` date NOT NULL,
  `PaymentMethod` varchar(45) NOT NULL,
  `employeeID` int NOT NULL,
  PRIMARY KEY (`orderID`,`employeeID`),
  UNIQUE KEY `orderID_UNIQUE` (`orderID`),
  KEY `FR_employeeID_idx` (`employeeID`),
  CONSTRAINT `FR_employeeID` FOREIGN KEY (`employeeID`) REFERENCES
`employees` (`personID`)
) ENGINE=InnoDB AUTO_INCREMENT=81 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

אכלוס:

```
INSERT INTO `orders` VALUES
(1,315287490,'2016-05-20','Credit',313084174),(2,315974582,'2017-02-14',
'Debit',300877090),(3,317460529,'2018-09-05','Cash',313775422),(4,318148
468,'2022-01-01','App',302822727),(5,318447589,'2022-02-14','Gift
Card',313775422),(6,319737108,'2023-03-01','Credit',301789107),(7,320607
254,'2023-02-14','Debit',310248652),(8,321278112,'2023-01-15','Cash',308
750124),(9,323600844,'2022-12-01','App',310248652),(10,323616182,'2022-1
1-05','Gift
Card',301836035),(11,324107886,'2022-10-02','Credit',301836035),(12,3252
22058,'2022-09-10','Debit',313775422),(13,326015663,'2022-08-15','Cash',
301789107),(14,315287490,'2022-07-20','App',308750124),(15,315287490,'20
22-06-01','Credit',313775422),(16,315974582,'2022-06-02','Debit',3102486
52),(17,317460529,'2022-06-03','Cash',312290162),(18,318148468,'2022-06-
04','App',312290162),(19,318447589,'2022-06-05','Gift
Card',304446235),(20,319737108,'2022-06-06','Credit',304446235),(21,3206
07254,'2022-06-07','Debit',313775422),(22,321278112,'2022-06-08','Cash',
308750124),(23,323600844,'2022-06-09','App',301789107),(24,323616182,'20
22-06-10','Gift
Card',312290162),(25,324107886,'2022-06-11','Credit',313084174),(26,3252
22058,'2022-06-12','Debit',304446235),(27,326015663,'2022-06-13','Cash',
312290162),(28,315287490,'2022-06-14','App',300877090),(29,315974582,'20
22-06-15','Gift
Card',313084174),(30,317460529,'2022-06-16','Credit',308750124),(31,3181
48468,'2022-06-17','Debit',301836035),(32,318447589,'2022-06-18','Cash',
313775422),(33,319737108,'2022-06-19','App',307671623),(34,320607254,'20
22-06-20','Gift
```

Card', 308750124), (35, 321278112, '2022-06-21', 'Credit', 313775422), (36, 323600844, '2022-06-22', 'Debit', 301789107), (37, 323616182, '2022-06-23', 'Cash', 313775422), (38, 324107886, '2022-06-24', 'App', 313084174), (39, 325222058, '2022-06-25', 'Gift
 Card', 300877090), (40, 326015663, '2022-06-26', 'Credit', 313084174), (41, 315287490, '2022-06-27', 'Debit', 313775422), (42, 315974582, '2022-06-28', 'Cash', 305298407), (43, 317460529, '2022-06-29', 'App', 306621797), (44, 318148468, '2022-06-30', 'Gift
 Card', 302822727), (45, 318447589, '2022-07-01', 'Credit', 302822727), (46, 319737108, '2022-07-02', 'Debit', 302822727), (47, 320607254, '2022-07-03', 'Cash', 306621797), (48, 321278112, '2022-07-04', 'App', 306621797), (49, 323600844, '2022-07-05', 'Gift
 Card', 304446235), (50, 323616182, '2022-07-06', 'Credit', 313775422), (51, 324107886, '2022-07-07', 'Debit', 301789107), (52, 325222058, '2022-07-08', 'Cash', 302822727), (53, 326015663, '2022-07-09', 'App', 300877090), (54, 315287490, '2022-07-10', 'Gift
 Card', 306621797), (55, 315974582, '2022-07-11', 'Credit', 313775422), (56, 317460529, '2022-07-12', 'Debit', 313775422), (57, 318148468, '2022-07-13', 'Cash', 301836035), (58, 318447589, '2022-07-14', 'App', 313775422), (59, 319737108, '2022-07-15', 'Gift
 Card', 301789107), (60, 320607254, '2022-07-16', 'Credit', 300877090), (61, 321278112, '2022-07-17', 'Debit', 301789107), (62, 323600844, '2022-07-18', 'Cash', 307671623), (63, 323616182, '2022-07-19', 'App', 313775422), (64, 324107886, '2022-07-20', 'Gift
 Card', 304446235), (65, 325222058, '2022-07-21', 'Credit', 313775422), (66, 326015663, '2022-07-22', 'Debit', 307671623), (67, 315287490, '2022-07-23', 'Cash', 313775422), (68, 315974582, '2022-07-24', 'App', 307671623), (69, 317460529, '2022-07-25', 'Gift
 Card', 302822727), (70, 318148468, '2022-07-26', 'Credit', 300877090), (71, 318447589, '2022-07-27', 'Debit', 305298407), (72, 319737108, '2022-07-28', 'Cash', 302822727), (73, 320607254, '2022-07-29', 'App', 300877090), (74, 321278112, '2022-07-30', 'Gift
 Card', 301836035), (75, 323600844, '2022-07-31', 'Credit', 305298407), (76, 323616182, '2022-08-01', 'Debit', 304446235), (77, 324107886, '2022-08-02', 'Cash', 302822727), (78, 325222058, '2022-08-03', 'App', 301789107), (79, 326015663, '2022-08-04', 'Gift
 Card', 300877090), (80, 315287490, '2022-08-05', 'Credit', 301836035);

טבלת בן אדם:

יצירה:

```
CREATE TABLE `person` (
  `personID` int NOT NULL,
  `FirstName` varchar(45) NOT NULL,
  `LastName` varchar(45) NOT NULL,
  `PhoneNumber` varchar(10) NOT NULL,
  `BirthDate` date NOT NULL,
  PRIMARY KEY (`personID`),
  UNIQUE KEY `personID_UNIQUE` (`personID`),
  UNIQUE KEY `PhoneNumber_UNIQUE` (`PhoneNumber`),
  KEY `INH_personID` (`personID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

אכלוס:

```
INSERT INTO `person` VALUES
(300877090, 'Noam', 'Kadosh', '0533210987', '1994-03-27'), (301789107, 'Shira',
, 'Ben-Arie', '0502109876', '2001-06-08'), (301836035, 'Shachar', 'Dahan', '054
1234567', '1990-01-01'), (302822727, 'Raz', 'Ben-David', '0587778899', '1985-1
1-18'), (304446235, 'Liran', 'Sasson', '0547654321', '1987-09-12'), (305298407
, 'Noa', 'Ohana', '0539991122', '1973-08-12'), (306621797, 'Noam', 'Cohen', '055
6667777', '1992-12-12'), (307671623, 'Ori', 'Ben-Aroya', '0587654321', '1976-0
6-25'), (308750124, 'Yael', 'Klein', '0532109876', '1984-03-09'), (310248652, '
Hadar', 'Nir', '0581112222', '1985-12-25'), (312290162, 'Yonatan', 'Bar-Yosef',
, '0509998888', '1989-11-21'), (313084174, 'Or', 'Amit', '0559876543', '2003-06
-02'), (313775422, 'Adi', 'Golan', '0543210987', '1998-01-01'), (314436520, 'Da
niel', 'Ohana', '0543154748', '2004-06-18'), (314436521, 'Shaked', 'Dror', '054
3254748', '2005-03-01'), (314436522, 'Elad', 'Balva', '0543184748', '2006-02-0
7'), (314436523, 'Yanir', 'Shmul', '0533154748', '2007-11-16'), (314436524, 'Kf
ir', 'levi', '0548154748', '2008-12-12'), (314436525, 'Yuval', 'Rashman', '0547
154748', '2012-07-13'), (315287490, 'Itay', 'Pinto', '0543334455', '1978-12-03
'), (315974582, 'Maya', 'Yaniv', '0504321098', '1971-07-07'), (317460529, 'Eita
n', 'Sela', '0509876543', '1969-10-27'), (318148468, 'Shira', 'Levi', '05777788
88', '1978-02-14'), (318447589, 'Noam', 'Cohen', '0580987654', '1964-12-30'), (
319737108, 'Lior', 'Bachar', '0535554444', '2002-12-04'), (319817459, 'Tom', 'S
halev', '0545432109', '2000-02-16'), (320607254, 'Or', 'Barnea', '0559012345',
, '2000-08-05'), (321278112, 'Avraham', 'Levi', '0505556677', '2001-09-30'), (32
3600844, 'Shaked', 'Dror', '0548824231', '2004-05-28'), (323616182, 'Bar', 'Ben
-Shushan', '0558765432', '1981-02-14'), (324107886, 'Tom', 'Levi', '0525678901
', '1962-11-01'), (325222058, 'Elad', 'Cohen', '0522345678', '1992-05-08'), (32
6015663, 'Michal', 'Shapira', '0521112233', '1989-04-22'), (326249751, 'Tal', '
Sharon', '0529876543', '1992-05-10'), (326570520, 'Daniel', 'Elharar', '054444
```

```
3615', '2004-06-18'), (326672599, 'Tomer', 'Adi', '0506256826', '2004-08-08'),  
(327772385, 'Maya', 'Levy', '0526543210', '1979-02-11');
```

טבלת מוצרים-הזמנות:

יצירה:

```
CREATE TABLE `products_orders` (
```



```

`productID` int NOT NULL,
`orderID` int NOT NULL,
`productAmount` int NOT NULL,
PRIMARY KEY (`productID`, `orderID`),
KEY `INDX_products_orders_ordID` (`orderID`) /*!80000 INVISIBLE */,
CONSTRAINT `orderID_FK` FOREIGN KEY (`orderID`) REFERENCES `orders`
(`orderID`),
CONSTRAINT `productID_FK` FOREIGN KEY (`productID`) REFERENCES
`products` (`productID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

```

אכלוס:

```

INSERT INTO `products_orders` VALUES
(1,1,3),(1,19,1),(1,47,5),(1,48,3),(1,64,3),(2,8,1),(2,29,3),(2,35,4),(2,
,39,4),(2,51,3),(3,7,1),(3,48,5),(3,55,4),(3,69,4),(3,71,4),(3,72,5),(4,
9,4),(4,11,4),(4,21,4),(4,62,2),(4,77,3),(5,7,3),(5,38,2),(5,46,5),(5,68
,3),(6,45,5),(6,49,2),(6,53,5),(6,58,2),(6,62,3),(7,16,1),(7,52,1),(7,63
,1),(7,75,2),(8,11,3),(8,15,5),(8,52,5),(8,57,5),(9,12,2),(9,18,1),(9,39
,3),(9,60,1),(9,74,3),(9,76,2),(10,26,2),(10,41,4),(10,70,5),(11,12,1),(
11,16,5),(11,24,3),(11,42,5),(11,75,2),(12,22,4),(12,44,4),(12,49,2),(12
,61,2),(12,77,5),(13,34,2),(13,51,3),(13,53,2),(13,54,4),(13,76,1),(14,2
6,1),(14,36,2),(14,56,1),(14,79,4),(15,17,1),(15,47,2),(15,50,1),(15,70,
2),(16,10,5),(16,17,4),(16,25,4),(16,29,2),(16,35,2),(16,50,5),(17,33,4)
,(17,44,5),(17,48,3),(17,76,3),(18,14,2),(18,23,1),(18,28,5),(18,32,5),(
18,69,5),(19,5,2),(19,30,1),(19,61,2),(19,64,2),(19,66,1),(20,8,1),(20,3
8,4),(20,57,4),(20,58,1),(20,67,5),(21,1,2),(21,31,3),(21,37,3),(21,43,4)
),(21,77,4),(22,2,4),(22,11,2),(22,56,5),(22,58,3),(23,8,5),(23,14,4),(2
3,24,5),(23,31,1),(23,72,1),(24,3,1),(24,11,2),(24,33,3),(24,45,2),(24,5
4,3),(25,12,1),(25,30,3),(25,40,1),(25,70,4),(26,27,1),(26,43,3),(26,46,
2),(26,63,3),(26,68,5),(27,25,1),(27,60,4),(27,78,5),(27,79,1);

```

טבלת עגלת לקוח חודשית:

יצירה:

```
CREATE TABLE `temp_table_customermonthlycart` (  
  `OrderID` int unsigned NOT NULL,  
  `Order_Date` date NOT NULL,  
  `Name` varchar(255) NOT NULL,  
  `UnitPrice` int NOT NULL,  
  `UnitAmount` int NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

אכלוס:

*האכלוס משתנה בהתאם ללקוח שנשלח לפרוצדורה שיוצר את טבלת העזר הזאת

דוגמא:

```
INSERT INTO `temp_table_customermonthlycart` VALUES  
(17,'2022-06-03','Asus ROG  
Zephyrus',1600,1),(17,'2022-06-03','Logitech  
Z623',120,4),(30,'2022-06-16','HyperX Fury  
S',20,1),(30,'2022-06-16','ASUS Prime  
Z590-A',250,3),(43,'2022-06-29','Redragon  
S101-BA',30,4),(43,'2022-06-29','EVGA SuperNOVA  
850W',150,3);
```