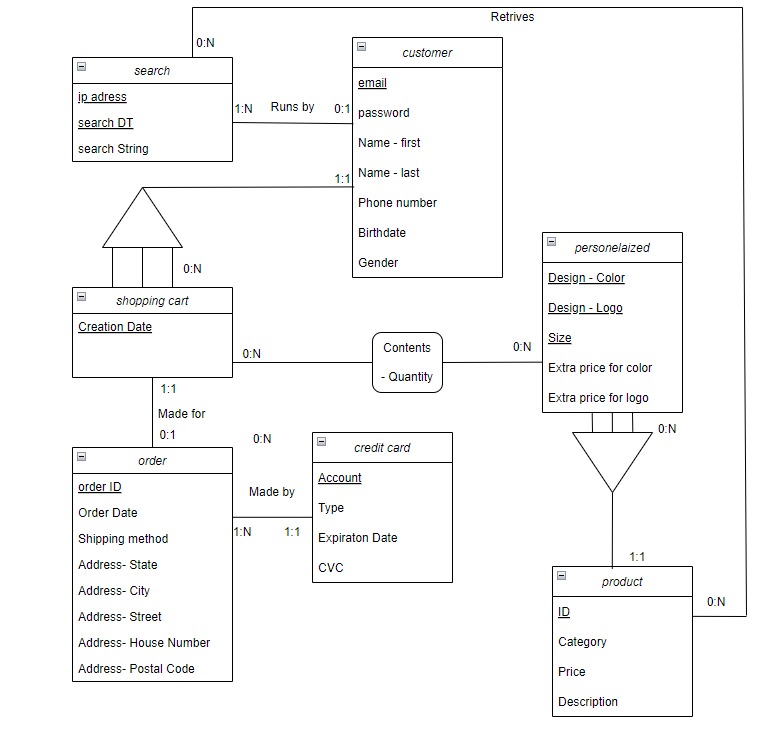
|  |
| --- |
| אתר |
| [בגדי ספורט בעיצוב אישי](http://www.personalisedsports.com) |

**פרויקט בסיסי נתונים – חלק ג'**



**• CUSTOMERS** (Email, password, Name- first, Name- last, phone number, Birthdate, Gender)

**• ORDERS** (order ID, DT, Address- State, Address- City, Address- Street, Address- House Number, Address- Postal Code, Shipping Method, {Creation Date, Email} (SHOPPING CARTS), Account (CREDIT CARDS))

**• SEARCHES** (IP Address, Search DT, Email (CUSTOMERS))

**• SEARCH STRING**({IP Address, Search DT} (SEARCHES), Search String)

**• SHOPPING CARTS** (Creation Date, Email (CUSTOMERS))

**• CREDIT CARDS** (Account, Type, Expiration Date, CVC)

**• PRODUCTS** (ID, Category, Size, Price, Description)

**• PERSONALISATIONS** (Design- Color(COLORS), Design – Logo(LOGOS), ID (PRODUCT))

**• COLORS** (Design – Color, Extra Price)

**• LOGOS** (Design – Logo, Extra Price)

**• RETRIVALS** ({IP Address, Search DT} (SEARCHES), ID (PRODUCT))

**• CONTENT** ({Creation Date, Email} (SHOPPING CARTS), ID (PRODUCTS), Quantity)

**מטלה 1- שאילתות**

**שתי שאילתות SELECT ללא קינון**

1. **אילו מוצרים החברה הרוויחה עליהם מעל 1000 דולר**

**ומחיר יחידה קטן מ100?**מוצרים זולים שהחברה מרוויחה עליהם יחסית הרבה (נקנים בכמויות) – רווחי לייצר בכמויות גדולות.

select c.ID, [Total Price]=sum((c.Quantity)\*(COLORS.[Extra Price]+ LOGOS.[Extra Price]+ p.Price))

from ORDERS as o join CONTENT as c on o.Email = c.Email AND o.[Creation Date]=c.[Creation Date]

join COLORS on COLORS.[Design – Color]= c.[Design – Color]

join logos on logos.[Design – Logo]= c.[Design – Logo]

join PRODUCTS as P on p.ID=c.ID

where p.Price < 100

group by c.ID

having sum((c.Quantity)\*(COLORS.[Extra Price]+ LOGOS.[Extra Price]+ p.Price))>1000

order by [Total Price] desc

פלט השאילתה-

Table

Description automatically generated

2. **מה המוצר שנקנה הכי הרבה על ידי נשים, בשנה הנוכחית?**  
היגיון עסקי – איזה מוצר שווה לקדם מול לקוחות שאנו יודעים שהן נשים (למשל נעלי נשים) , ישתנה בהתאם לשנה.

select top 1 [current year]=year(o.[Order Date]), c.ID, [sum] = sum(c.Quantity), CU.Gender

from ORDERS as o join CONTENT as c on o.Email = c.Email AND o.[Creation Date]=c.[Creation Date]

join CUSTOMERS as CU on o.Email = CU.Email

where year(o.[Order Date]) = year(getdate()) and CU.Gender like 'female'

group by year(o.[Order Date]),c.ID, CU.Gender

order by sum desc

פלט השאילתה-

A picture containing table

Description automatically generated

**שתי שאילתות SELECT מקוננות**

1. **פילוח אחוז הלקוחות לפי קבוצות גיל.**  
היגיון עסקי- נרצה לדעת כמה כסף להשקיע בפרסום בכל פלטפורמה ביחס לאחוז הלקוחות בכל גיל. למשל, עבור לקוחות בגילאי 16-30 נעדיף להתמקד בפרסומים בפלטפורמת אינסטגרם, לעומת זאת עבור קהל יעד של בני 60+ נמקד את הפרסום בטלוויזיה.

SELECT total = count(\*)

INTO SUMMARY

FROM CUSTOMERS

select age ,amount=count(\*)

, total, ratio = round(cast(count(\*) as float)/cast(total as float),2) from

(select age= case when datediff(yy, c.Birthdate, getDate()) between 16 and 30 then '16 - 30'

when datediff(yy, c.birthdate, getDate()) between 31 and 60 then '31 - 60'

when datediff(yy, c.birthdate, getDate()) >=60 then '61+'

else '0-15' end

from CUSTOMERS as c) as ages, SUMMARY

group by age, total

order by amount desc

drop table SUMMARY

פלט השאילתה-

Table

Description automatically generated

2. **איזה מוצרים לא נמכרו בכלל בשנה האחרונה?**

היגיון עסקי- תכנון תכנית עסקית לקו הייצור של השנה הבאה- זיהוי הרגלי צריכה שהיו ונעלמו, אופנות חולפות וכו'- כל הללו יגרמו לנו לשאוף להפחית בייצור מוצרים אלו במידת האפשר, על מנת לחסוך עלויות מיותרות.

SELECT distinct C.ID

FROM ORDERS as O JOIN CONTENT AS C ON O.[Creation Date] = c.[Creation Date] and o.Email = c.Email

WHERE Year(o.[Order Date]) not in(

select distinct C.ID

from ORDERS as O JOIN CONTENT AS C ON O.[Creation Date] = c.[Creation Date] and o.Email = c.Email

WHERE year(O.[Order Date])=Year(GetDate()))

פלט השאילתה-

Table

Description automatically generated

**110 רשומות**

**שתי שאילתות מקוננות תוך שימוש במרכיבים נוספים**

1. **שאילתת UPDATE המקטלגת מוצרים לפי כמות פעמים שהם נמכרו – best sellers, normal, not sold.**   
ההיגיון העסקי הוא שחשוב לנו כחברה לדעת מהם המוצרים החזקים ביותר, נשתמש במידע זה בתוכניות העסקיות – אם מבחינת שיווק, תמחור (כיוון שהביקוש להם הוא הגבוה ביותר, כנראה שלקוחות יסכימו לשלם עליהם מחירים גבוהים ועל כן נוכל בעבור אותו המוצר להגדיל את הרווח) וכו'.

alter table products add [product rate] varchar(11)

update PRODUCTS

set [product rate] = case when t.amount > 10 then 'best seller' when t.amount between 1 and 10 then 'normal'

else 'not sold'

end

from(

select c.ID, amount = sum(c.Quantity)

from ORDERS as o join CONTENT as c on o.Email = c.Email AND o.[Creation Date]=c.[Creation Date]

group by c.ID

union

select PRODUCTS.ID, amount = 0

from PRODUCTS

except(

select distinct c.ID,amount = 0

from ORDERS as o join CONTENT as c on o.Email = c.Email AND o.[Creation Date]=c.[Creation Date]

join PRODUCTS as p on p.ID = c.ID

)

) as t

where t.ID = PRODUCTS.ID

פלט טבלה מעודכנת:

Table

Description automatically generated

2. **שאילתה שבודקת מה הם העיצובים (צבע, לוגו) שמחירם גבוה מהמחיר הממוצע ומעולם לא הוזמנו.**  
ההיגיון העסקי – ייתכן שהם לא הוזמנו מעולם כיוון שאנשים לא מוכנים לשלם עליהם את המחיר הנוכחי שלהם, ועל כן נשקול להוריד את מחירם כדי שיימכרו, או אם הדבר לא משתלם לנו, להוריד את האופציה להוסיף אותם, לגמרי.

select a.[Design – Color] as Design

from

(select [Design – Color] , [Extra Price] from COLORS

where [Extra Price] > (select avg([Extra Price]) from COLORS)

union

select \* from LOGOS

where [Extra Price] > (select avg([Extra Price]) from LOGOS)) a

except(

select b.[Design – Color] as Design

from

(select distinct [Design – Color]

from CONTENT

union

select distinct [Design – Logo]

from CONTENT) b

)

פלט השאילתה:

Table

Description automatically generated

**75 רשומות**

**מטלה 2- יישומי כלים מתקדמים**

**VIEW**

יצרנו 2 טבלאות VIEW שמציגות את כרטיסי האשראי השמורים בבסיס הנתונים. הטבלה הראשונה, הכללית, מיועדת לעובדים שנרצה שתהיה להם גישה לכרטיסי האשראי במקרה הצורך אך נרצה להסתיר מהם נתונים שלא נרצה שתהיה להם גישה אליהם, כמו הקוד הסודי.

ה-VIEW השני מיועד למורשים בלבד ומכיל את כל הנתונים של כרטיסי האשראי.

----------view that shows us only credit card numbers, without details we wouldn't want our employees (who need the information about the credit cards numbers) to know.

create View [v\_CreditCard] as

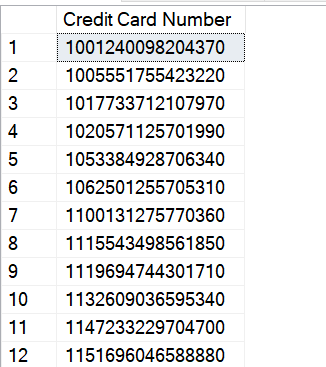
select [Credit Card Number]= Account

From [CREDIT CARDS]

group by Account

select \* from v\_CreditCard

פלט הVIEW-



create View [v\_CreditCardsForAuthorizedOnly] as

select Distinct \*

From [CREDIT CARDS]

select \* from v\_CreditCardsForAuthorizedOnly

פלט הVIEW השני-

Table

Description automatically generated

**FUNCTION**

1. **פונקציה אשר מקבלת מוצר ומחזירה סקלר = כמות הפעמים שהוא נמכר.**  
ההיגיון העסקי – נרצה לדעת כמה מוצר פופולרי יותר או פחות, לצורך תכנון התוכניות העסקיות לתקופות הבאות – איזה מוצרים כדאי לקדם יותר, להעלות את מחירם וכו'.

create function dbo.AmountSoldByProduct(@ID int)

returns int

as begin

declare @amount int

select @amount = sum(c.Quantity)

from ORDERS as o join CONTENT as c on o.Email = c.Email AND o.[Creation Date]=c.[Creation Date]

join PRODUCTS as p on p.ID=c.ID

where p.ID = @id

return @amount

end

דוגמא לשימוש בפונקציה:

select id, [Total sales]= dbo.AmountSoldByProduct(53684)

from PRODUCTS

where PRODUCTS.id = 53684

פלט הפונקציה-

Table

Description automatically generated with medium confidence

2. **פונקציה המקבלת מספר, חודש ושנה ומחזירה טבלה המכילה מדינות שמכרו מספר מוצרים הגדול או שווה למספר שהוזן, בחודש המוזן ובשנה המוזנת**.

היגיון עסקי – נרצה לזהות הרגלי צריכה: אם יש מוצרים מסוימים שבחודשים ספציפיים הם פופולריים יותר או פחות (לכן נשווה ביניהם בחודשים שונים באותה שנה) לצורך תכנון תכניות עסקיות רבעוניות או שנתיות (עם דגש על התקופות השונות).

CREATE FUNCTION TopProduct ( @x int , @month int, @year int)

RETURNS TABLE

AS RETURN

SELECT o.[Address- State], amount = sum(c.Quantity)

FROM ORDERS as o join CONTENT as c on o.Email = c.Email AND o.[Creation Date]=c.[Creation Date]

where month(o.[Order Date]) = @month and year(o.[Order Date]) = @year

group by o.[Address- State]

HAVING sum (c.Quantity) >= @x

דוגמא להפעלת הפונקציה:

select \*

from dbo.TopProduct(5,3,2021)

פלט הפונקציה-

Table

Description automatically generated

**TRIGGER**

**הטריגר עוזר לנו לדעת מה המחיר הכולל של עגלות הקנייה בין שנקנו ובין שלא נקנו,** למשל במידה ועגלות מסוימות התמלאו במוצרים מסוימים אך הלקוח לא השלים את תהליך ההזמנה נרצה לשלוח מייל המתזכר את הלקוח אודות העגלה.

טריגר זה הכרחי מאחר ונרצה לעקוב אחר פעולות הלקוחות וכן למשוך חזרה לקוחות שהחלו תהליך רכישה והתחרטו, לא היו בטוחים וכו'.

alter table [shopping carts] add total money

create trigger update\_shopping\_carts

on content

for insert, update, delete

as update [SHOPPING CARTS]

set total = (select sum(PRODUCTS.Price + COLORS.[Extra Price] + LOGOS.[Extra Price])

from CONTENT join COLORS on CONTENT.[Design – Color]=COLORS.[Design – Color]

join LOGOS on CONTENT.[Design – Logo] = LOGOS.[Design – Logo]

join PRODUCTS on CONTENT.ID = PRODUCTS.ID

where CONTENT.[Creation Date] = [SHOPPING CARTS].[Creation Date] and CONTENT.Email = [SHOPPING CARTS].Email

)

where

[SHOPPING CARTS].Email in(select distinct Email from inserted union select distinct Email from deleted)

דוגמא להפעלת הטריגר:

insert into CONTENT(Email,[Creation Date],ID,[Design – Color],[Design – Logo],Size,Quantity)

Values ('abeavenf8@home.pl', '2018-07-16 13:09:00.000', 17564, 'brown', '716030824', 'm',1)

delete from CONTENT where Email = 'abeavenf8@home.pl' and ID = 17564

**Stored Procedure**

**הפרוצדורה תאפשר לשליחים של החברה גישה למידע שרלוונטי להם, מבלי לחשוף אותם לכל בסיס הנתונים של החברה.** למשל, אם יש בעיה עם משלוח מסוים – כתובת מסוימת לא נמצאה או נדרש קוד לכניסה לבניין, יוכל השליח להגיע למס' הטלפון של הלקוח מבלי להיחשף למידע אחר שלו (כמו סיסמה) שלא נרצה שתהיה לו גישה אליו.

פרוצדורה זו הכרחית על מנת לשמור על חיסיון הלקוח ועל אבטחת המידע.

create procedure SP\_GetInfo @Email varchar(40) as

select [Name- first], [Name- last], Email, [Phone Number]

from CUSTOMERS

where (Email = @Email)

הדגמת השימוש בפרוצדורה:

execute sp\_GetInfo 'aclubley60@typepad.com'

פלט הפרוצדורה-

Graphical user interface, text, application

Description automatically generated

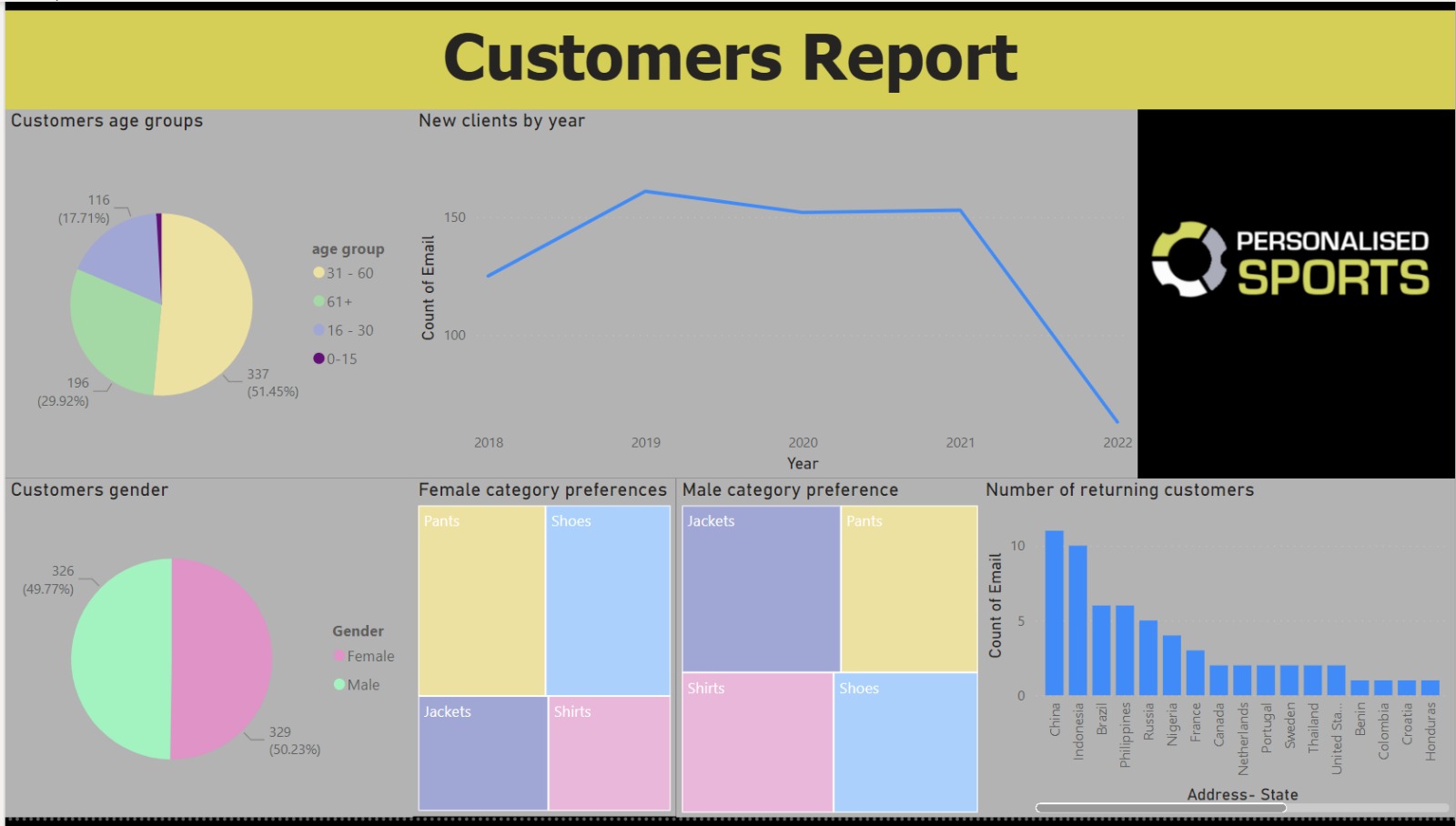
**מטלה 3- כלים להצגת נתונים**

**דו"ח עסקי**



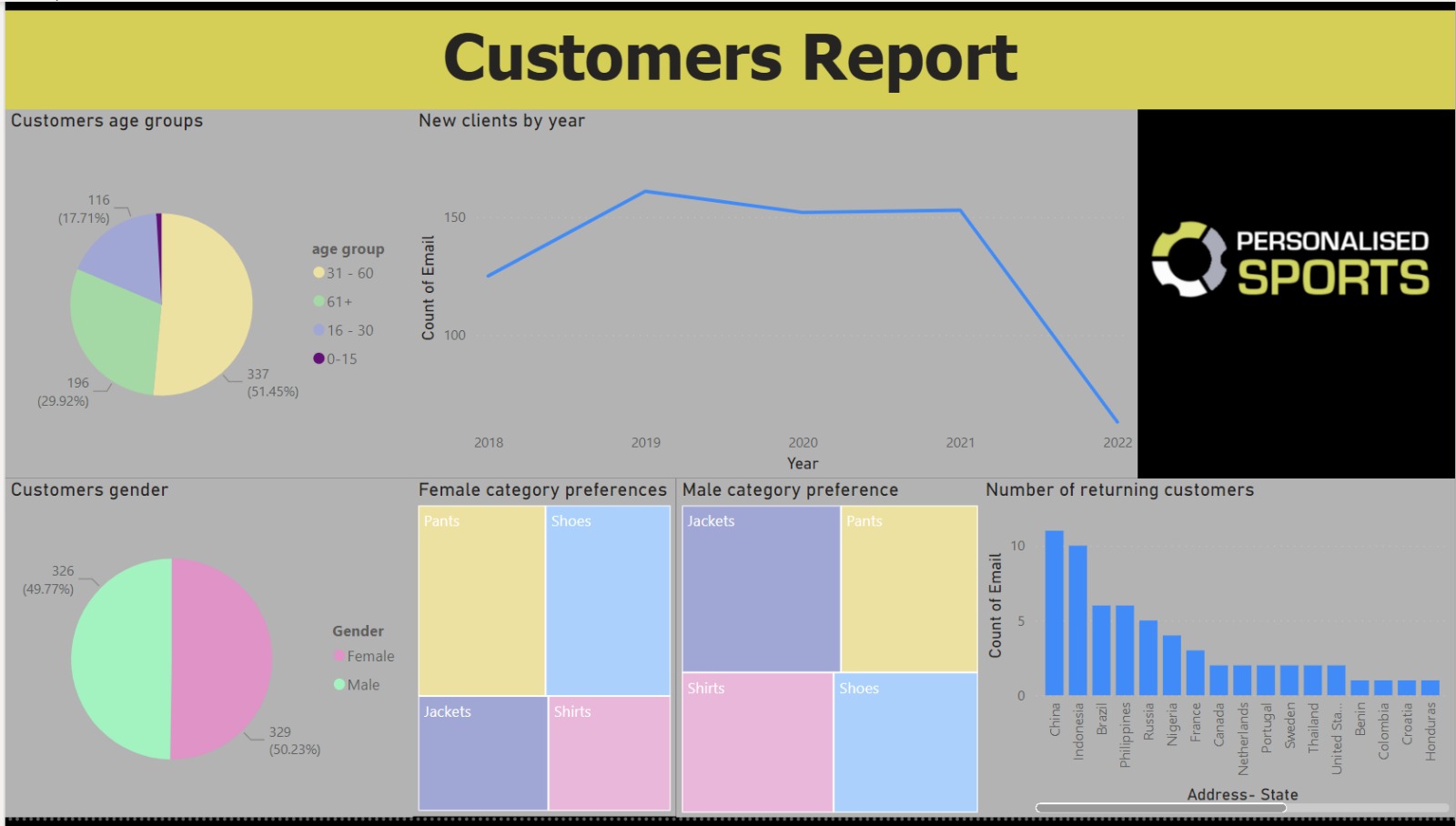
בחרנו לבנות דו"ח בנושא לקוחות החברה, שיוצג למנהלי החברה. מטרתו היא לספק נתונים אודות לקוחות החברה, שיסייעו בתכנון תוכניות עסקיות רלוונטיות ומיטביות עבורנו, שכן ניתוח והבנת קהל היעד של החברה הינה כלי חשוב עבור החברה וחיונית להצלחתה העסקית.

**הדו"ח כולל את הפרטים הבאים:**

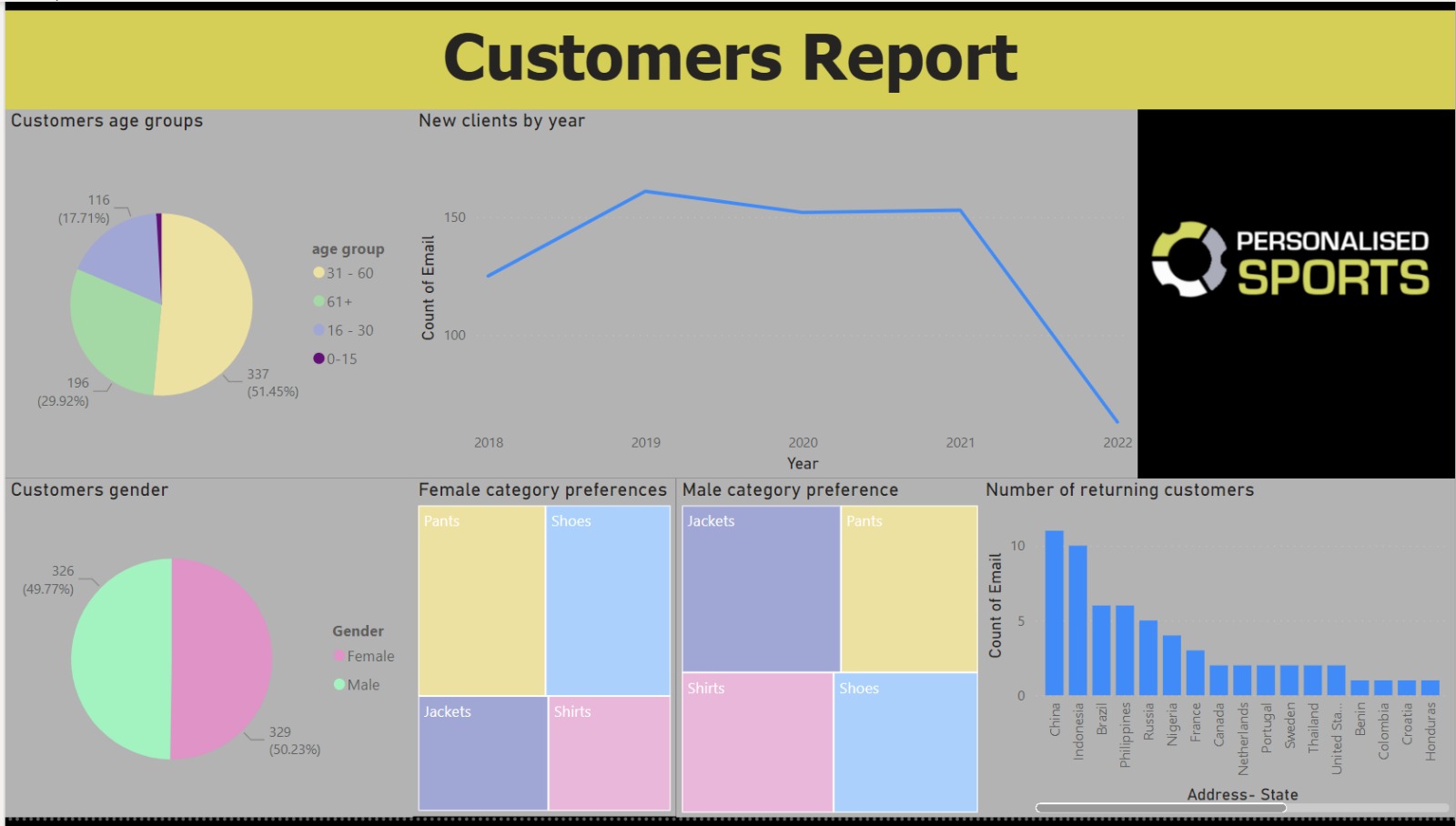


פילוח הלקוחות ע"פ קבוצות גיל – נרצה לדעת את התפלגות קבוצות הגילאים של לקוחות החברה.

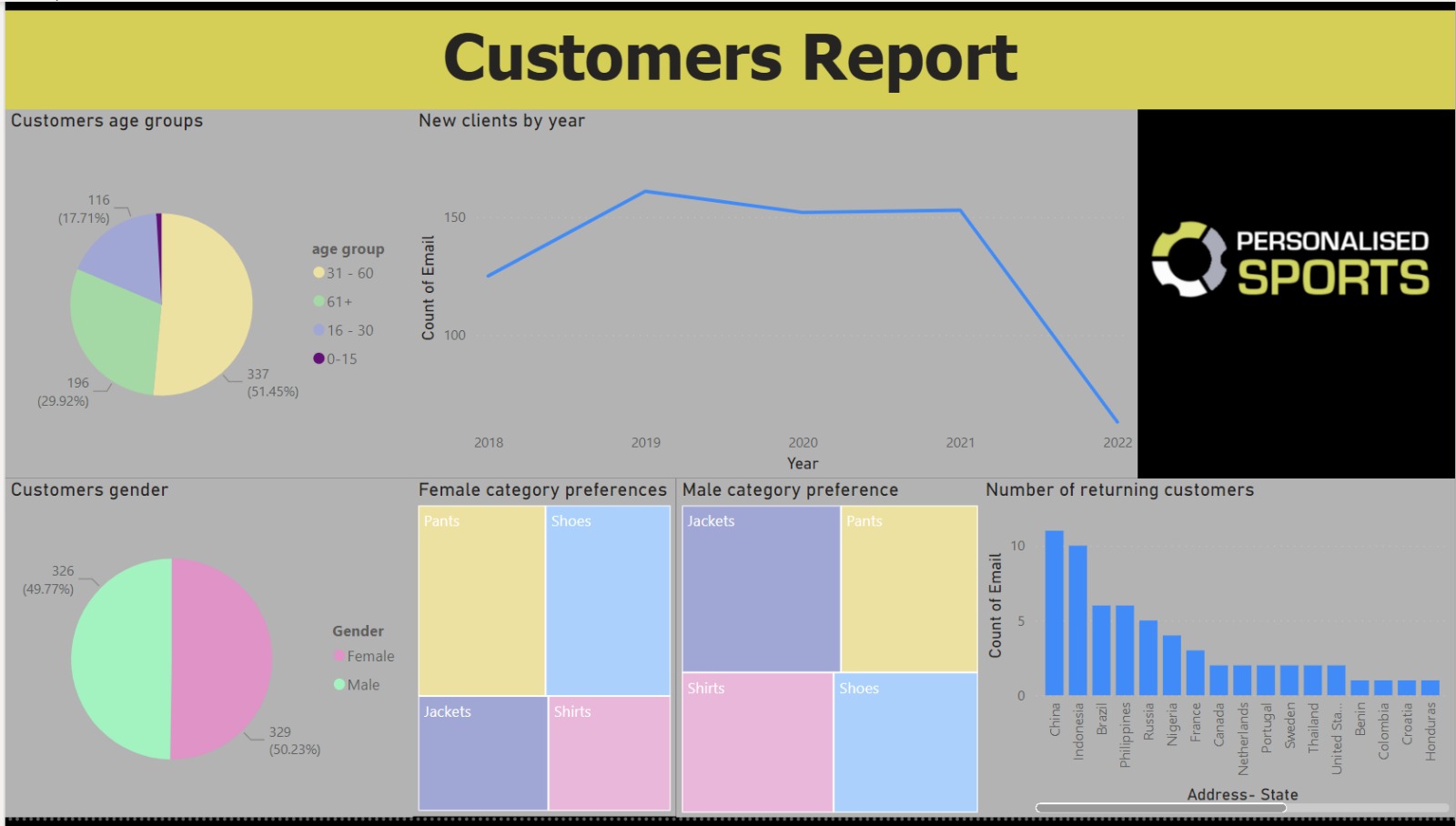
מידע זה יסייע בתכנון תוכניות עסקיות רלוונטיות, הן מבחינת השיווק (בחירת פלטפורמת שיווק מתאימה לפי גיל, עיצוב האתר והנגשת המידע – היות וקבוצת הגיל המובילה שלנו היא 31-60 נרצה להגדיל את הפונטים בהם אנו משתמשים באתר, נבחר בפלטפורמת טלוויזיה/ פייסבוק לפרסום שכן שם נמצא רוב קהל היעד שלנו וכו') והן מבחינת תכנון הייצור (למשל, היות וקבוצת הגיל 0-15 מהווה פלח קטן מאוד מלקוחות החברה, לא ישתלם להוציא ליין בגדי ילדים נרחב – כנראה שלא יהיה לו ביקוש רב).

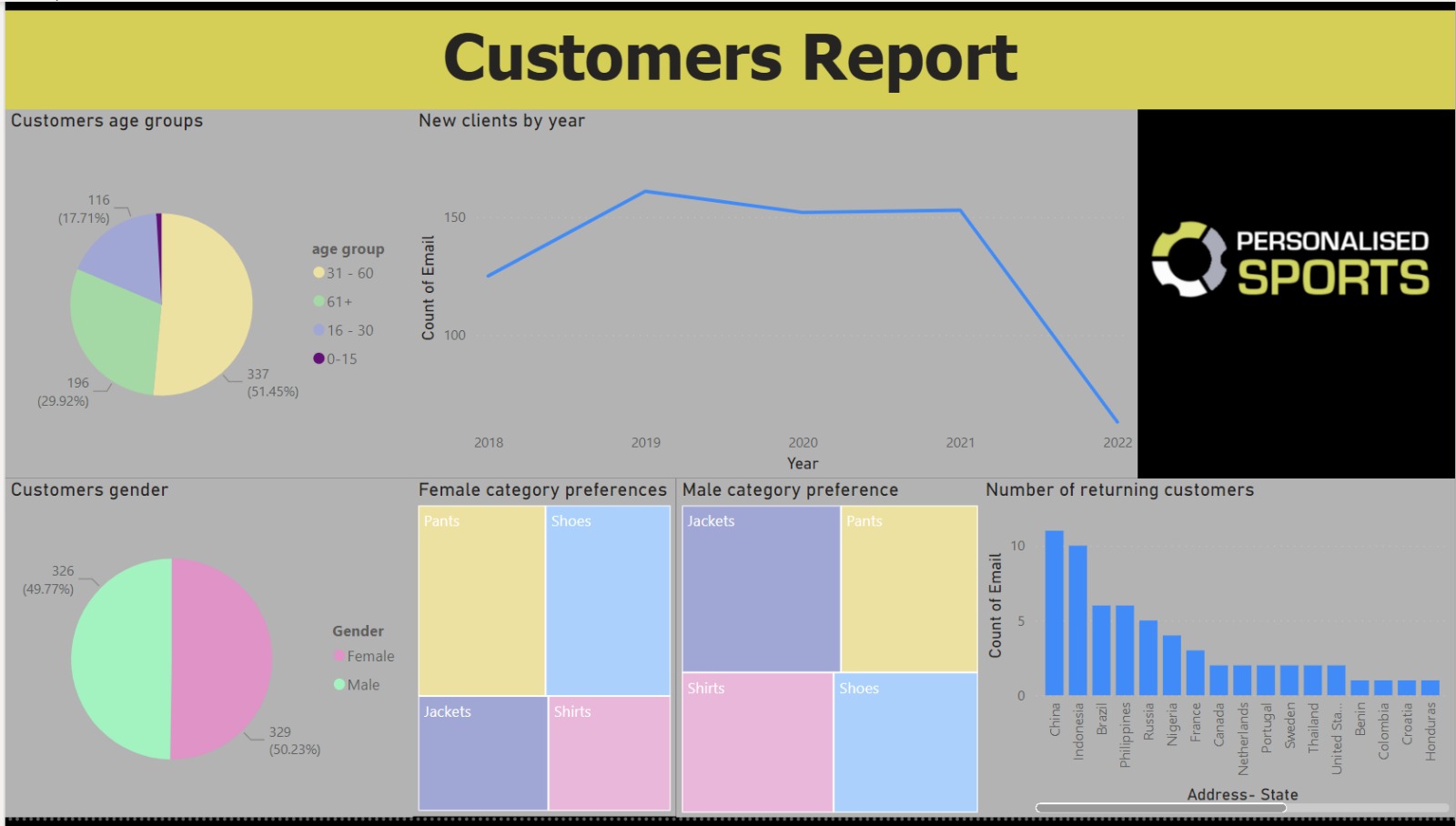


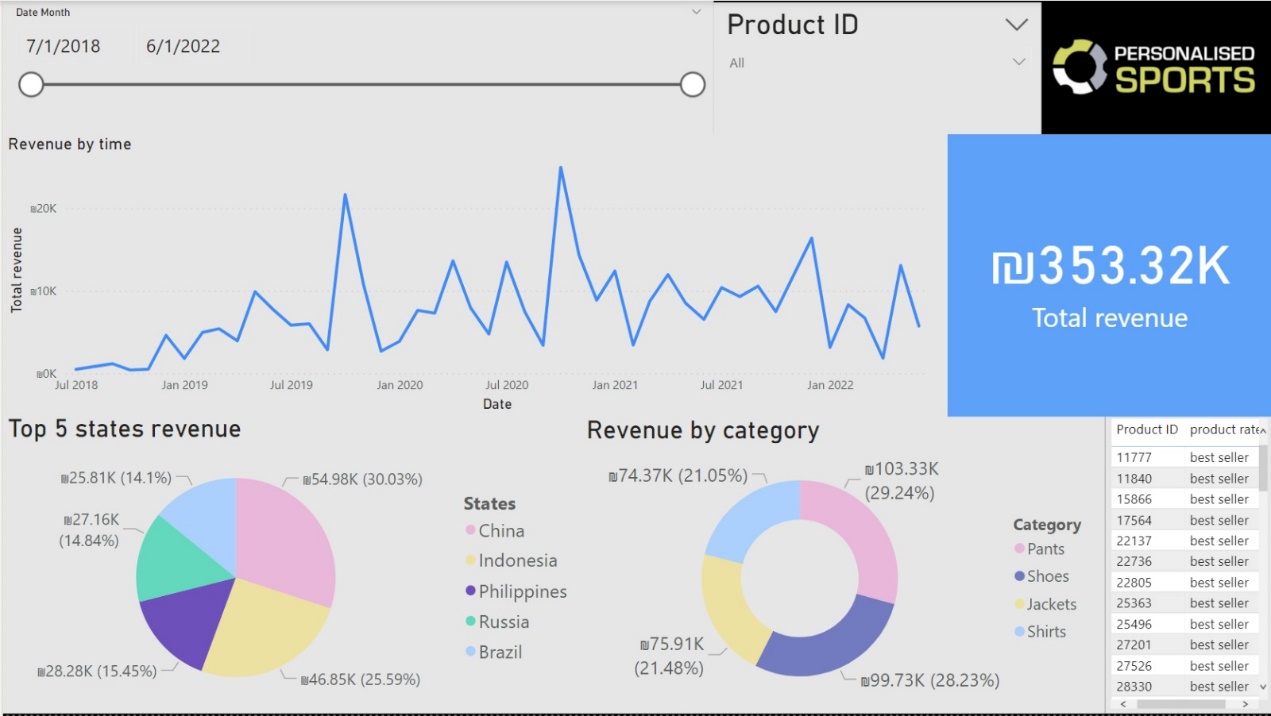
פילוח ע"פ מין – בדומה לפילוח ע"פ גילאים, נרצה לדעת לאיזה קהל אנו פונים יותר, לצורך תכנון האסטרטגיה התפעולית והשיווקית של החברה.



העדפות לפי קטגוריית מוצרים – עבור נשים ועבור גברים.

כמות לקוחות מצטרפים (ע"פ שנה). נוכל לנתח את פעילויותינו באותה שנה (למשל הוספת ליין מוצרים, הרחבת אזורי התפוצה, הורדת התמקדות ממוצר מסוים) ולהבין מה גרם לאותה מגמה, במטרה לשמור על הפונקציה עולה תמיד.

לקוחות חוזרים (ע"פ מדינה). נוכל לראות את כמויות הלקוחות החוזרים – אלו הם הלקוחות המרוצים, שהצלחנו לשמר בחברה. נראה שישנן מדינות בהן יותר לקוחות נשארים נאמנים לחברה – נוכל לנתח זאת, מה גורם לאותה מדינה להיות חזקה יותר, האם שירות הלקוחות באותה מדינה טוב יותר? או מצד שני למשל, אם במדינה מסוימת כמות הלקוחות החוזרים נמוכה, אולי יש לבחון את מערך המשלוחים באותה מדינה וכו'.

**לוח מחוונים**

לוח המחוונים מוצג למנהלי החברה ומכיל נתוני מאקרו על ההתנהלות באתר. הלוח מהווה "מבט על" על החברה – ומטרתו היא לבחון את תמונת המצב של החברה, במטרה לגבש תכנית עסקית תקופתית.

**לוח המחוונים כולל את הפרטים הבאים- 2 פילטרים:**

פילטר מוצרים – תוצאות לפי כלל המוצרים, או מוצרים ספציפיים

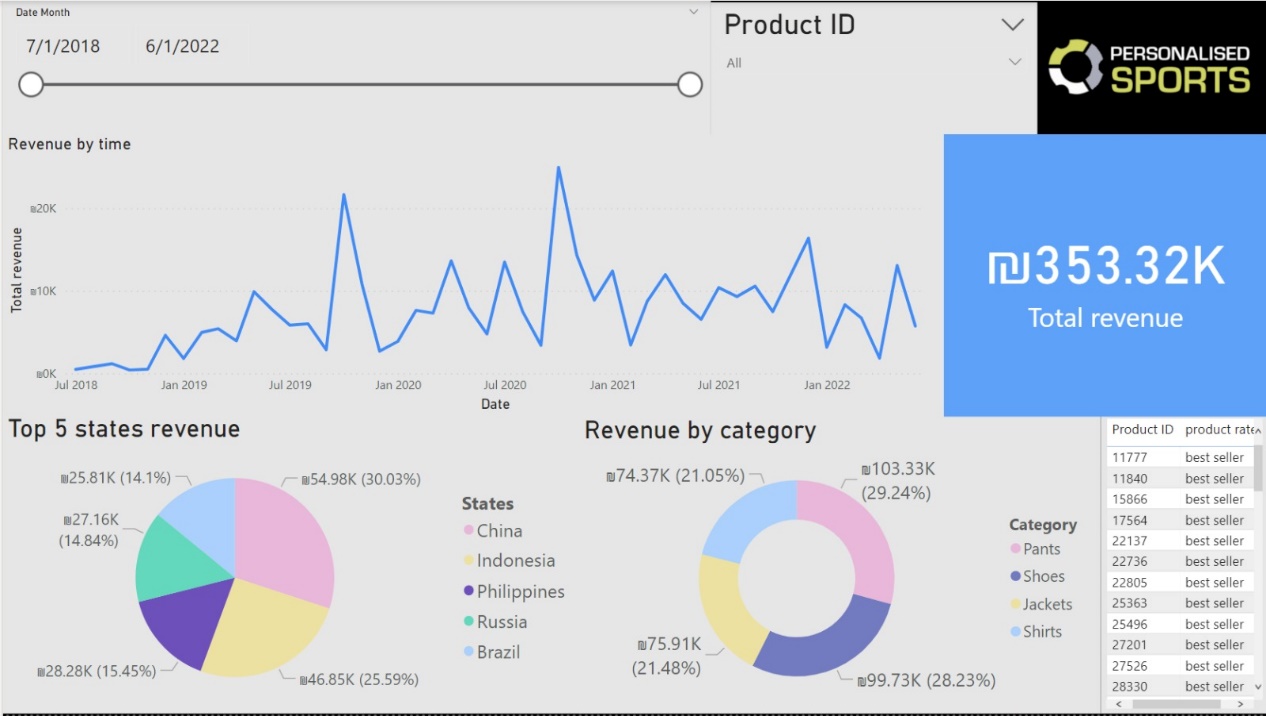
Graphical user interface, application, Teams

Description automatically generated

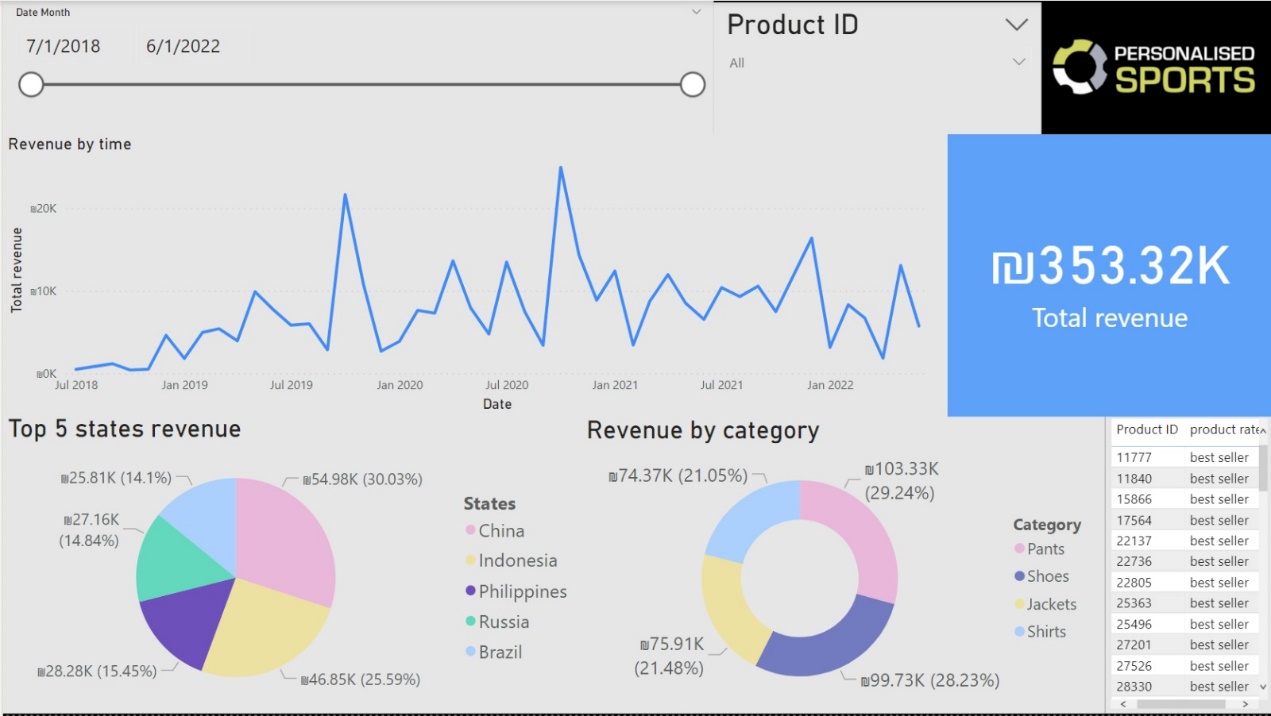
פילטר זמן – אפשרות לבדיקת הנתונים על זמן מוגדר

Shape, rectangle

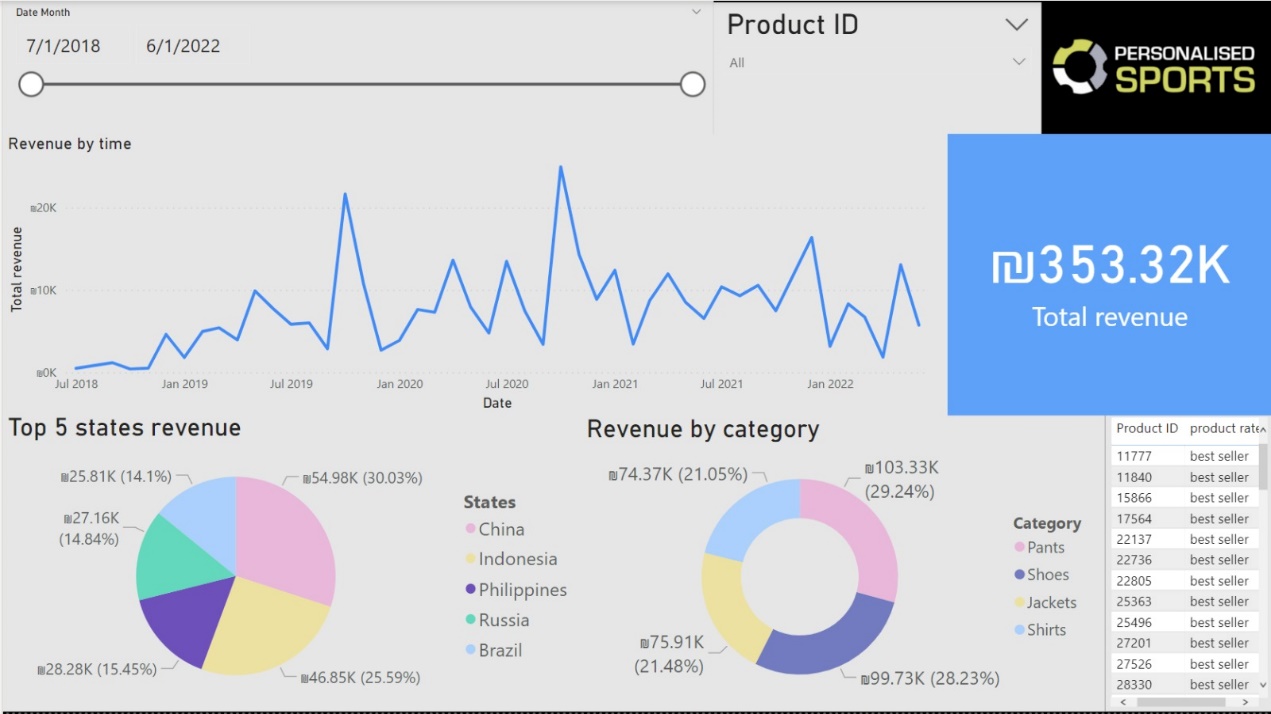
Description automatically generated with medium confidence



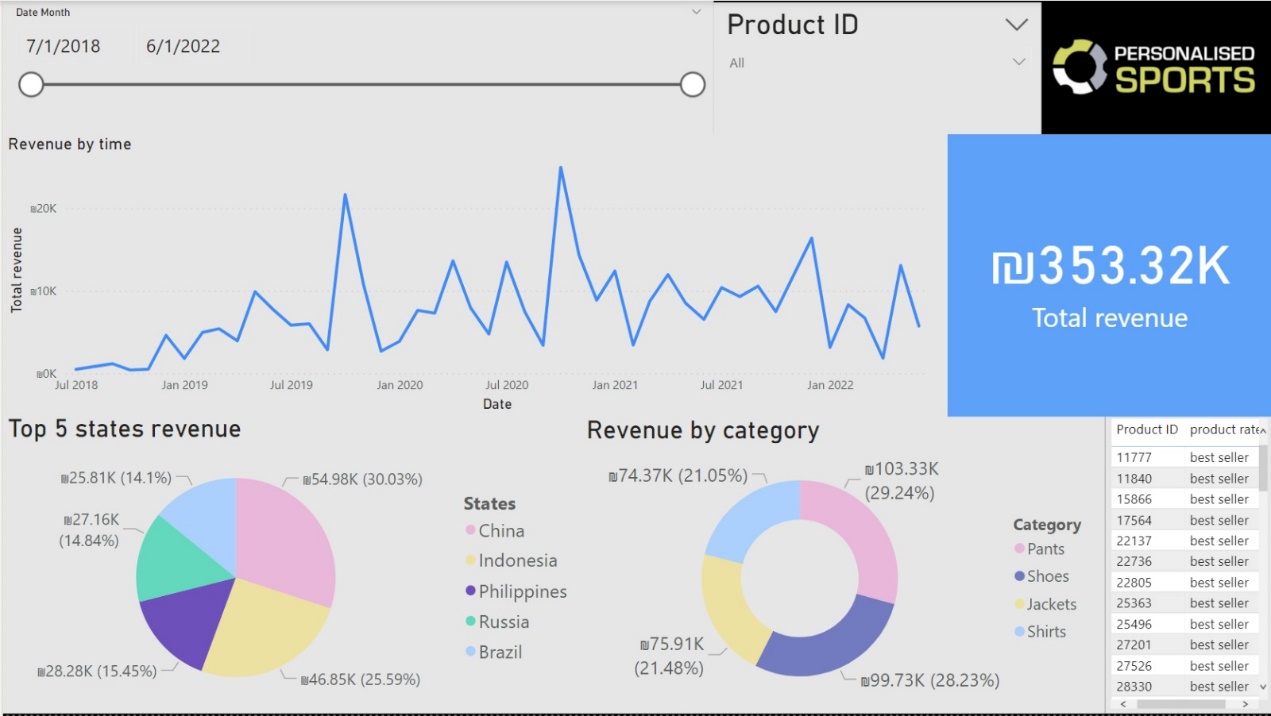
סך הכנסות החברה



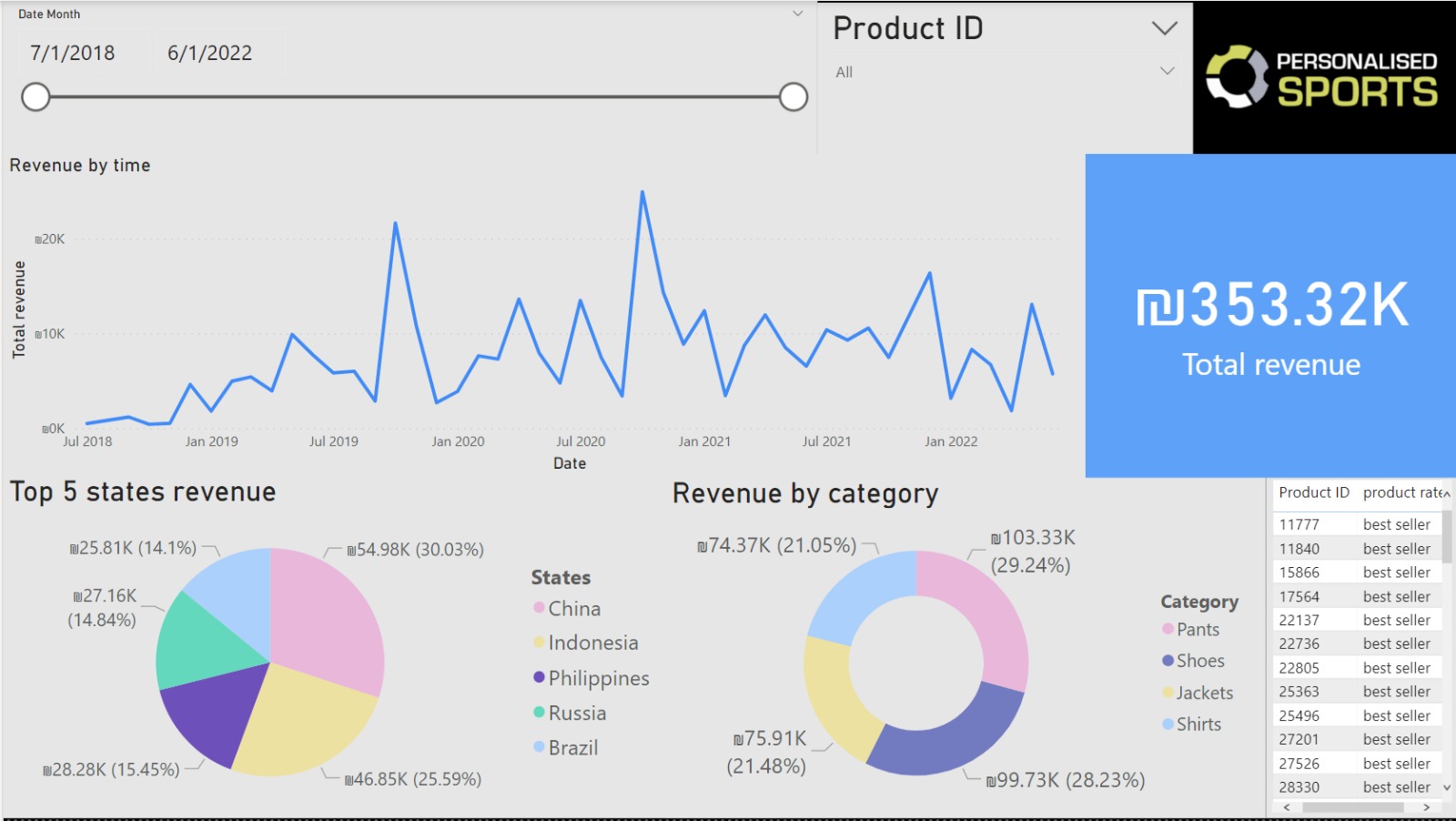
סך הרווחים של החברה פר חודש.



חמש המדינות המובילות בהכנסות.



סך ההכנסות של החברה, לפי קטגוריה.



רשימת המוצרים המובילים באתר – כלומר אלו

שנמכרים הכי הרבה.

המידע לעיל מלמד את הדרג הניהולי של החברה על הנעשה בה ומסייע לקבלת החלטות מושכלות מבחינה תכנונית – בין אם בקווי בייצור (ביקוש למוצרים - מה לייצר יותר, ביקוש לפי תקופות וכו') ובין אם אסטרטגיה שיווקית (באיזה מדינות יש לקדם את החברה, למשל) או אסטרטגיה ניהולית (אם מוצר מסוים מבוקש מאוד, אולי כדאי להרחיב את אפשרויות המכירה שלו – בצבעים, במידות נוספות – או למשל, להרחיב את שירות הלקוחות במדינה מובילה במכירות ע"מ לשמר את שביעות רצון הלקוחות ולוודא לקוחות חוזרים, וכולי).

**מטלה 4- יישומים מתקדמים**

**Window Functions**

1. שאילתה המספקת את **הרווח השנתי של החברה, רווח החברה בשנה הקודמת, מהי עליית הרווח באחוזים ובאיזה שנה הייתה העלייה המשמעותית ביותר בחברה.**

הגיון עסקי- נרצה למצוא שנים רווחיות לעומת שנים שקדמו להן כדי לנתח כיצד פעלנו באותה שנה ביחס לשנים הקודמות לה ולראות איך אנו שומרים על התנהלות זו, או לחלופין בשנים שבהן הייתה ירידת מכירות נרצה לדעת מה היו הגורמים לכך ולטפל בכך כדי שהמצב לא יחזור שנית. למשל- אם היו חסרים מוצרים מסוימים, או שמוצרים מסוימים לא נמכרו בערים מסוימות ועוד.

השאילתה מכילה את הפונקציות- RANK() וLAG():

select [year], TotalSales , PrevYearSales , [yearlyGrowth (%)],

yearRank =RANK() OVER (ORDER BY [yearlyGrowth (%)] desc)

from

(select \*, [yearlyGrowth (%)]= round((TotalSales)/(PrevYearSales), 2)

from

(SELECT \*,

PrevYearSales =round( LAG(TotalSales , 1) over (order by [year]),2)

from

(select [year]= year(ORDERS.[order date]) ,

TotalSales=round( sum(CONTENT.Quantity\*(COLORS.[Extra Price]+LOGOS.[Extra Price]+PRODUCTS.Price)), 2)

from orders join CONTENT on orders.Email= content.Email and

ORDERS.[Creation Date]=CONTENT.[Creation Date]

join COLORS on CONTENT.[Design – Color]=COLORS.[Design – Color]

join LOGOS on CONTENT.[Design – Logo] = LOGOS.[Design – Logo]

join PRODUCTS on CONTENT.ID = PRODUCTS.ID

group by year(ORDERS.[order date])) AS V

) as f)

as g

order by yearRank

פלט השאילתה-

Table

Description automatically generated

2. **שאילתה המחשבת את פרק הזמן בין הרכישה הראשונה של לקוח, לבין הרכישה האחרונה שלו. עבור לקוחות שמשך זמן זה (שהגדרנו כמשך חברות) שווה או גדול מ-2, נרצה לשלוח מתנת חברות ע"מ לשמר לקוחות נאמנים.**

נשים לב שהחברות נמדדת בין אם היו רכישות נוספות בטווח זמנים זה ובין אם לא, כדי שלקוחות חזקים של החברה יתוגמלו אך גם לקוחות חוזרים שנרצה לשמר ולהדק את היחסים עימם.

הפונקציות שהשאילתה משתמשת בהם – First\_Value(), Last\_Value(), Lead().

select Email, [Friendship Period]

from

(select \*, [Friendship Period] = ABS(LEAD (DateDiff(yy, LastPurchase,FirstPurchase), 1) OVER( ORDER BY FIRSTPURCHASE))

from (select c.email, FirstPurchase = First\_Value(O.[Order Date]) OVER (Partition By c.Email Order By [Order date]), LastPurchase = Last\_Value(O.[Order Date]) OVER (Partition By c.Email Order By [Order date])

from ORDERS AS O JOIN CUSTOMERS AS C ON O.Email = C.Email

) as full\_table

where firstpurchase <> LastPurchase)

as Friendship

where Friendship.[Friendship Period] >= 2

פלט השאילתה-

Graphical user interface, text

Description automatically generated

**70 רשומות**

**שילוב מערכתי של מספר כלים**

פונקציה חשובה שנרצה להוסיף לאתר שלנו היא פונקציית **ניהול הטרנזקציות**. כאשר לקוח רוצה לסיים את ההזמנה, הוא עובר שרשרת פעולות שנרצה שתגיע כיחידה עסקית אחת ויחידה- כלומר שאם אחת הפעולות נכשלת, כל הטרנזקציה תיכשל (roll back), רק אם רצף הפעולות מושלם בהצלחה אז "בלוק" הפעולות מושלם (כדי למנוע מצב למשל, שכרטיס אשראי לא תקין והכסף לא מועבר, אך הלקוח הצליח להשלים את תהליך ההזמנה). לכן, נוסיף Stored Procedure שמאחדת את תהליך ההזמנה של לקוח.

התהליך יורכב משרשרת הפעולות הבאה: קבלת עגלת הקניות העדכנית -> קבלת כרטיס אשראי, בדיקת תקינות, בדיקת תקינות העגלה (לוודא שאינה ריקה)-> השלמת ההזמנה (ליצור ישות חדשה של ORDER בתוך הטבלה) -> יצירת עגלה חדשה.

create PROCEDURE dbo.Transaction\_Management @email varchar(40), @AddressState varchar(20),@AddressCity varchar(20), @AddressStreet varchar(20), @AddressHouseNumber varchar(10), @AddressPostalCode varchar(10), @ShippingMethod varchar(20), @Account varchar(20)

as

declare @ORDERID bigint, @expirationM int, @expirationY int

set @expirationM = (select Month([Expiration Date]) from [CREDIT CARDS] where Account = @Account)

set @expirationY = (select Year([Expiration Date]) from [CREDIT CARDS] where Account = @Account)

IF (@expirationY < Year(GETDATE()) or (@expirationY = Year(GETDATE()) AND @expirationM <= Month(GETDATE())))

Begin

-- if credit card has expired:

RAISERROR ('TRANSACTION IS INVALID. CREDIT CARD HAS EXPIRED', 18, 0)

RETURN

END

--check if the shopping cart that is being sent is filled with products

-- if not, break the transaction

IF ((@Email not in(select Email from CONTENT))OR (dbo.GetLastSC(@email) not in (select [Creation Date] from CONTENT)))

BEGIN

RAISERROR('NO ITEMS IN CART ', 18, 0)

RETURN

END

-- insert into ORDER table

insert into ORDERS ([order ID], [Order Date], [Address- State], [Address- City], [Address- Street], [Address- House Number], [Address- Postal Code], [Shipping Method],Email,[Creation Date],Account)

values(dbo.GetLastoRDERid (),GETDATE(), @AddressState , @AddressCity, @AddressStreet, @AddressHouseNumber, @AddressPostalCode, @ShippingMethod,@email,dbo.GetLastSC(@email),@Account)

הפרוצדורה משתמשת בפונקציה שעבור מייל של משתמש מסוים, מחזירה את התאריך של עגלת הקניות העדכנית ביותר שלו – היא זו שנשלח עבורה הזמנה במקרה של השלמת הזמנה.

--function the returns the email and last date of shopping cart --> most relevant shopping cart.

CREATE FUNCTION GetLastSC (@email varchar(40))

RETURNS DateTime

AS BEGIN

DECLARE @MaxDate DateTime

select @MaxDate = MAX([Creation Date]) from [SHOPPING CARTS] where Email like @email

RETURN @MaxDate

END

דוגמא להפעלת הפונקציה:

select Distinct Email, LastDate = dbo.GetLastSC('abeavenf8@home.pl')

from [SHOPPING CARTS]

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

בנוסף הפרוצדורה משתמשת גם בפונקציה שמטרתה להחזיר לנו מספר הזמנה חדש (ORDER ID) עבור ההזמנה שאנו יוצרים.

כדי להבטיח שמס' ההזמנה יהיה ייחודי, נבחר את מס' ההזמנה המקסימלי ונוסיף לו 1. מס' זה יהווה את מס' ההזמנה החדשה. (כך גם נוכל לשמור את כמות ההזמנות שבוצעו באתר, בכמות מספרית שמהווה גם ה-ID).

CREATE FUNCTION GetLastoRDERid ()

RETURNS BIGINT

AS BEGIN

DECLARE @MaxOrderID BIGINT

select @MaxOrderID = MAX([order ID]) from ORDERS

RETURN (@MaxOrderID + 1)

END

דוגמא להפעלת הפרוצדורה:

execute dbo.Transaction\_Management 'akmiecet@cbslocal.com', 'Israel', 'Hertzliya', 'Ha Universita', '3', '498573', 'Express', '1001240098204370'

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטיטבלת ORDERS המעודכנת:

כעת נוסיף טריגר שבעת ביצוע הזמנה תיווצר עגלת קניות חדשה וריקה, עבור המשתמש.

CREATE TRIGGER insert\_new\_SC on ORDERS for insert as insert into [SHOPPING CARTS] (Email, [Creation Date])

Values((Select Distinct Email from inserted), GETDATE())

נדגים את פעולת הטריגר:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

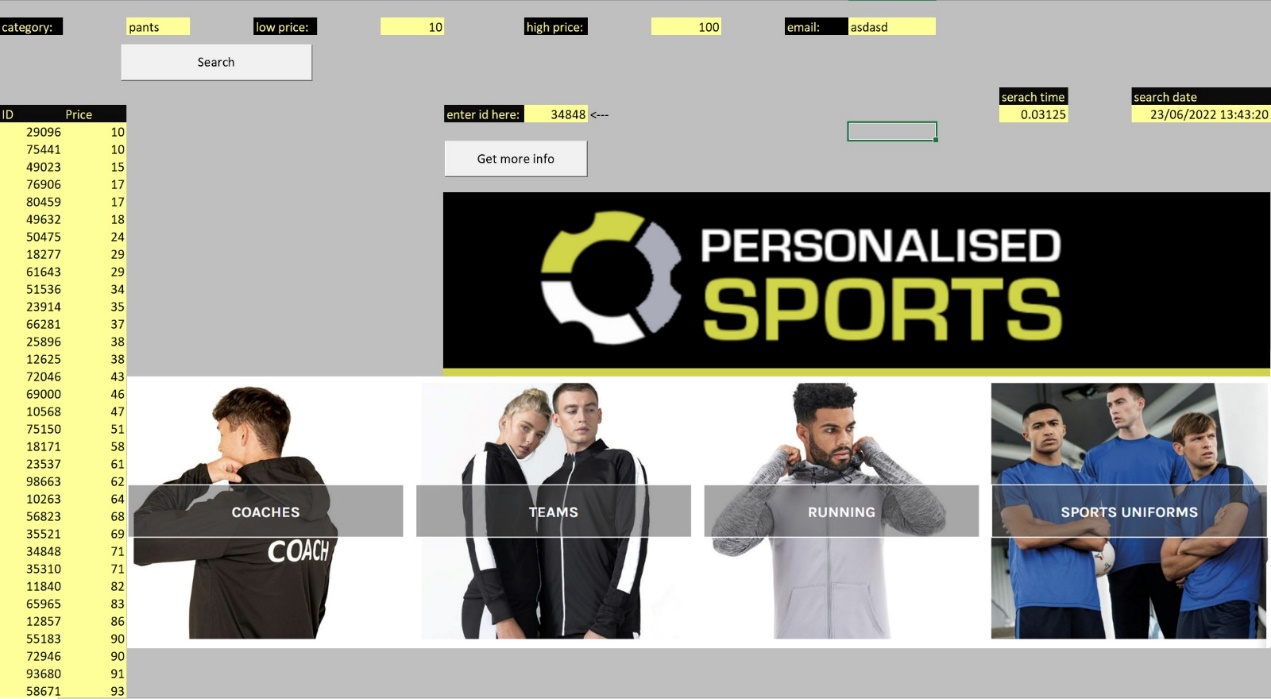
(השאילתה להפעלת הפרוצדורה אכן הופעלה בשעה 21:21:15 ביום 21.6.2022)

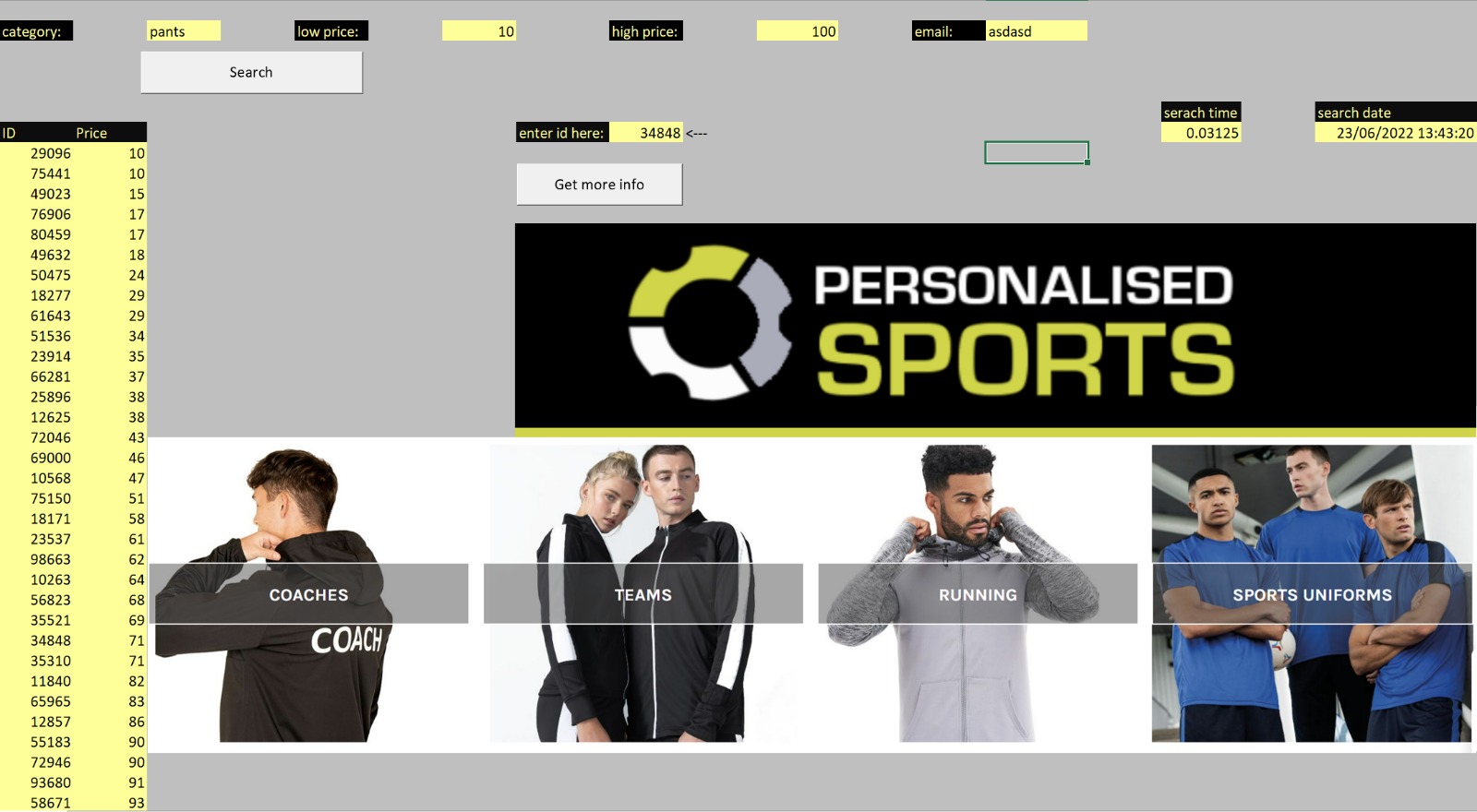
נשים לב שאם העגלה ריקה, ההזמנה לא תישלח והטריגר לא יופעל.

(התראת NO ITEMS IN CART )

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

**יישום מנוע חיפוש בעזרת גיליון אלקטרוני**



החלטנו לבנות מנוע חיפוש ללקוחות האתר, שיקל על תהליך החיפוש באתר ויאפשר גישה נוחה למידע הרלוונטי על המוצרים שבהם מעוניינים הלקוחות.

במנוע החיפוש, המשתמש מתבקש להזין את קטגוריית פריט הלבוש שבו הוא מעניין וטווח המחירים שהוא מוכן לשלם בעבורו (LB שמהווה המחיר הנמוך שהוא מוכן לשלם ו – UB שמהווה הגבלת הסכום שהוא מוכן לשלם).

בחיפוש זה, תיפתח לו חלונית כפי שניתן לראות מצד שמאל (להלן תקריב):

המפרטת את המוצרים העונים על הדרישות של הלקוח.

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטיבלחיצה על מוצר מסוים, תיפתח לו חלונית המספקת לו מידע נוסף אודות המוצר.

המטרה של מנוע החיפוש היא לאפשר חיפוש נוח, יעיל ומהיר עבור המשתמש, במטרה לשפר את שביעות רצון הלקוח, ולשמר את לקוחות החברה.

**תמונה של הטבלה לאחר מספר חיפושים**

**תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי**