

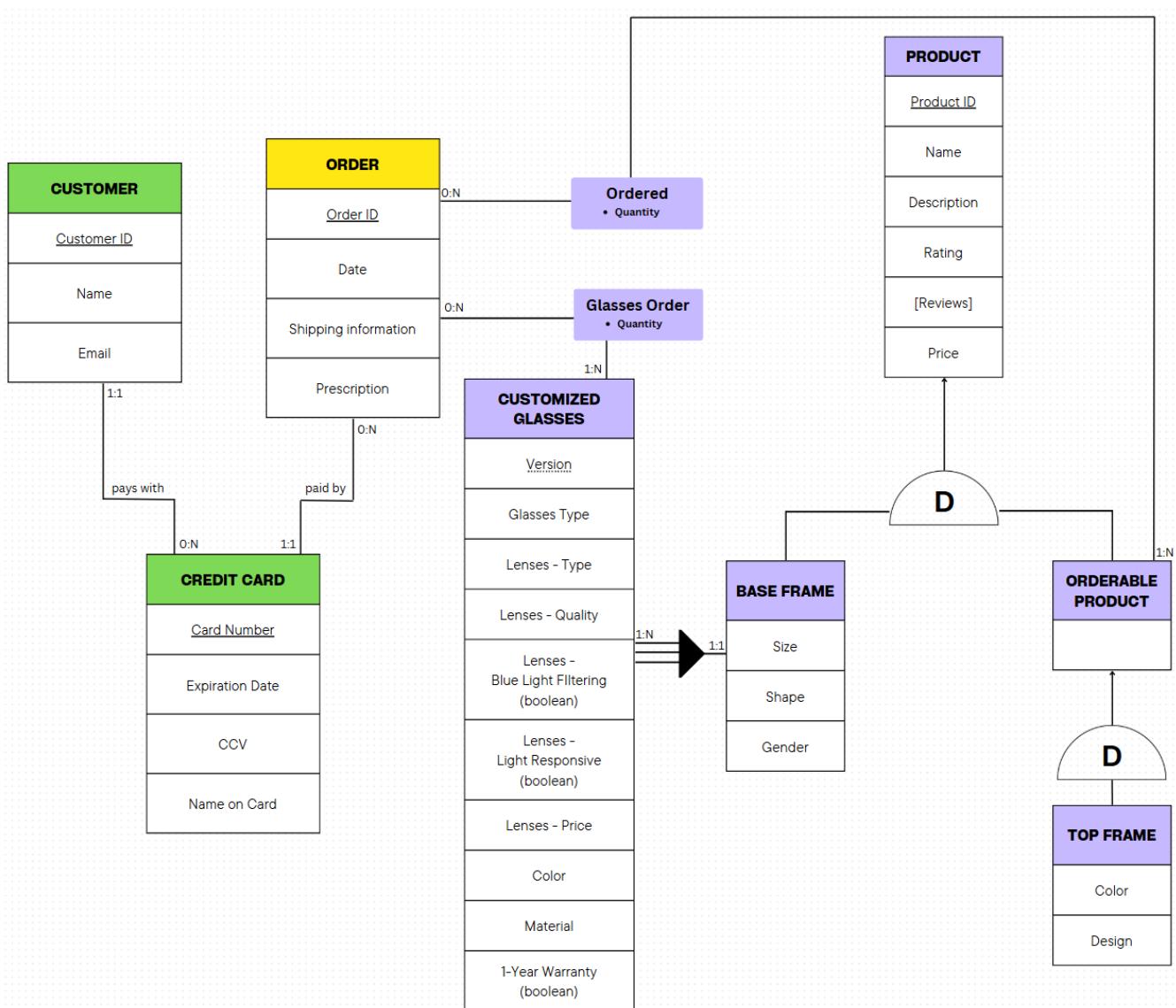


תאריך הגשה	אתר	מספר קבוצה
22/05/2025	www.paireyewear.com	15

## פרויקט בסיסי נתונים - חלק ב'

מטרה מקדימה – ה-ERD שהוגש בחלק א'

התרשים בהגשה הקודמת



הנחות שנרשמו בהגשה הקודמת

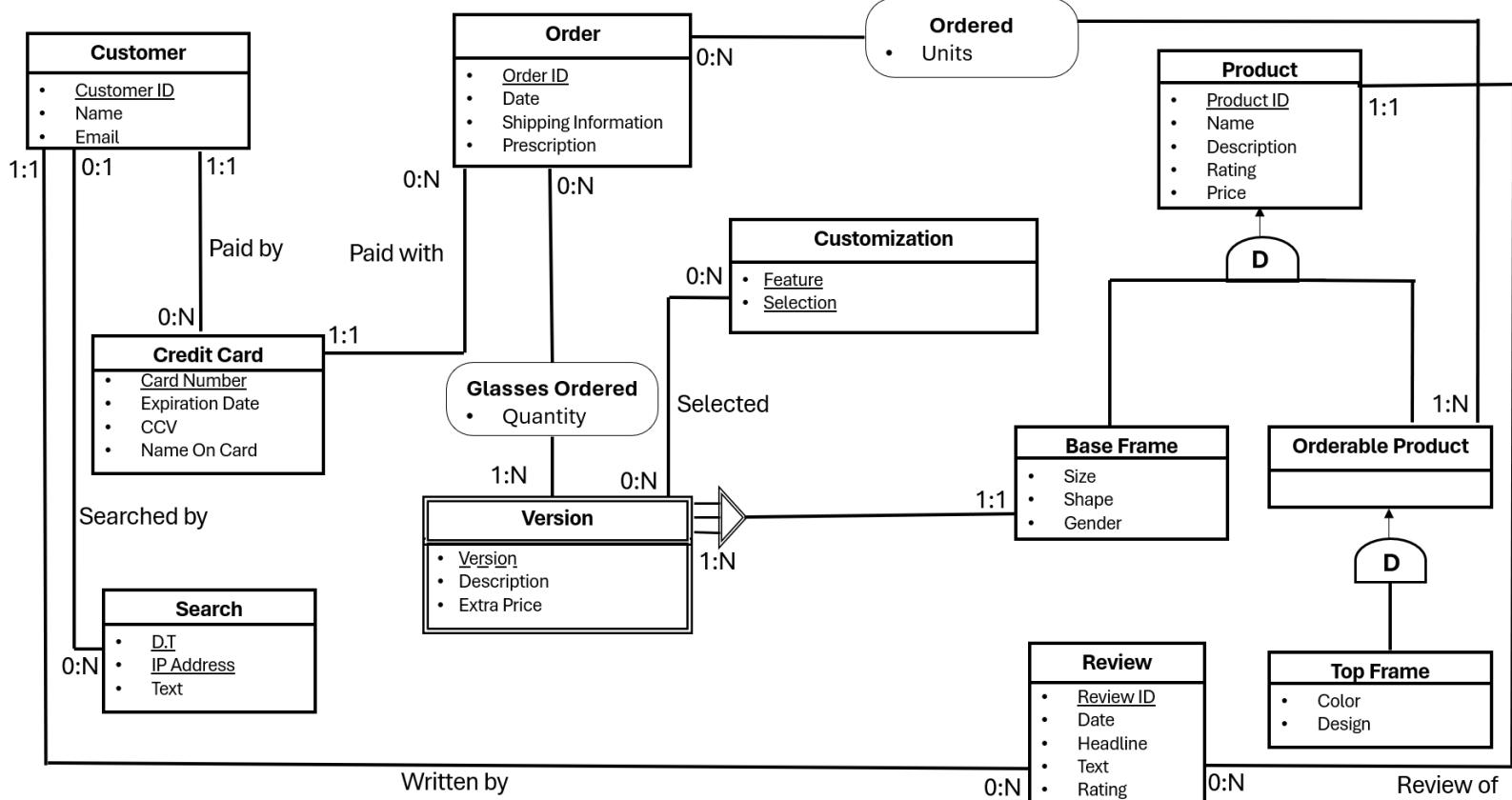


- הנחנו כי לכל וריאציה של מסגרת בסיס בהתאם אישית (צבע, חומר, תכונות העדשה וכו') יש מספר סידורי: Version.
- הנחנו כי הדרך היחידה לשלם באתר היא באמצעות כרטיס אשראי.
- יתכן שיש מגבלה כלשהי באתר לשמר מס' מוגבל של כרטיסי אשראי פר לקוח, אך לא בדקנו זאת. במקרה, הנחנו כי能夠 יכול לשמור מס' בלתי מוגבל של כרטיסי אשראי (מ-0 ועד אינסוף) ובעת ביצוע הזמנה לבחור באיזה מהם הוא רוצה להשתמש.

#### ההערות שניתנו להגשה הקודמת

- זה לא הפורמט שעובדים איתו בקורס.
- חסר ישות חיפוש.
- חוות דעת היא אם תוכן וגם ציון מה גם שחשיבות לדעת מי רשום את חוות הדעת لكن לא מתאים כשדה מרובה ערכיים.
- העיצוב אישי שלכם אפשרי אך ניתן לבצע התאמה בשביל שהייה יותר טוב. אפשר לראות דוגמה שפורסמה במודול בשwil לקבל השראה.

#### מטרה 1 (40%) – עיצוב קונספטואלי (תרשים יישוות-קשרים)



## הנחות מודל ה- ERD

מרכיב	הנחות
המזהה של היחסות VERSION	הנחנו כי לכל וריאציה של מסגרת בסיס בהתאם אישית (צבע, חומר, תכונות העדשה וכו') יש מספר סידורי: Version
היחסות CREDIT CARD	הנחנו כי הדרך היחידה תשלום היא באמצעות כרטיס אשראי.
הקשר בין כרטיס אשראי ללקוח	יתכן שיש מגבלה כלשהי באתר לשימרת מספר מגבל של כרטיסי אשראי פר לקוח, אך לא בדקנו זאת. במקום, הנחנו כי לקוח יכול לשמור מס' בלתי מגבל של כרטיסי אשראי (מ-0 ועד אינסוף), ובעת ביצוע הזמנה לבחור באיזה מהם הוא רוצה להשתמש.
היחסות ORDERABLE PRODUCT	אתר יש מוצרים נלוויים שניתן להזמין (לדוגמא: ערכת ניקוי לעדשות), והוא תחת יישות זו.



## מטרה 2 (30%) – עיצוב לוגי (מודל נתונים טבלאי)

- **CUSTOMERS** ( Customer\_ID, First\_Name, Last\_Name, Email )
- **CREDIT\_CARDS** ( Card\_Number, Expiration\_Date, CCV, Name\_On\_Card, Customer\_ID (CUSTOMERS) )
- **SEARCHES** ( DT, IP\_Address, Text, Customer\_ID (CUSTOMERS) )
- **ORDERS** ( Order\_ID, Date, Country, City, Street\_Name, Street\_Number, Apartment, ZIP\_Code, Prescription, Card\_Number (CREDIT\_CARDS) )
- **PRODUCTS** ( Product\_ID, Name, Description, Rating, Price )
- **ORDERABLE\_PRODUCTS** ( Product\_ID (PRODUCTS) )
- **TOP\_FRAMES** ( Product\_ID (ORDERABLE\_PRODUCTS), Color, Design )
- **ORDERED** ( Product\_ID (ORDERABLE\_PRODUCTS), Order\_ID (ORDERS), Units )
- **BASE\_FRAMES** ( Product\_ID (PRODUCTS), Size, Shape, Gender )
- **VERSIONS** ( Product\_ID (BASE\_FRAMES), Version, Description, Extra\_Price )
- **CUSTOMIZATIONS** ( Feature, Selection )
- **GLASSES\_ORDERED** ( {Product\_ID, Version} (VERSIONS), Order\_ID (ORDERS), Units )
- **SELECTED** ( {Product\_ID, Version} (VERSIONS), {Feature, Selection} (CUSTOMIZATIONS) )
- **REVIEWS** ( Review\_ID, Date, Headline, Text, Rating, Customer\_ID (CUSTOMERS), Product\_ID (PRODUCTS) )

### הנחות והסברים נוספים:

- נירמול השדה - **Name** השדה Name בישות Customer במודל ה-ERD פוצל ל-2 שדות אוטומים: First และ Last על מנת למנוע הפרת נרמול מסדר ראשון.
- נירמול השדה - **Shipping Information** השדה Shipping Information בישות Order במודל ה-ERD פוצל ל-6 שדות אוטומים על מנת למנוע הפרת נרמול מסדר ראשון. Country, City, Street\_Name, Street\_Number, Apartment, ZIP\_Code בנוסף, בהוראות הקורס כתוב לא להתייחס לביעות נרמול הנbowות מהשדה "מיקוד" ולכך לא להתיחסנו אליו.
- הקשר בין **City** לבין **Country** הנחנו כי ייתכן קיומו של 2 ערים עם אותו השם ב-2 מדינות שונות, ולכן אין תלות פונקציונלית בין השדה City לשדה Country, ומכאן שאין הפרת נרמול מסדר שלישי.
- הקשר בין **CUSTOMER** לבין **CREDIT CARD** - הנחנו שלכל לקוח (CUSTOMER) יכול להיות יותר מכרטיס אשראי אחד אך כל כרטיס שיר ללקוח אחד בלבד. לכן, שדה Customer\_ID מופיע בטבלה CREDIT\_CARDS כמפתח זר.



- הקשר בין **VERSION** לבין **CUSTOMIZATION** - הנקנו שכל וריאציה של מסגרת בסיס (VERSION) מקוורת במספר התאמות אישיות (Customization) היוצרות את הוריאציה.
- כМОבן שגם מס' וריאציות של מסגרות בסיס יכולות להיות מקושرات להתקאה אישית אחת - לכן מדובר בקשר רבים לרבים, ייצרנו לכך טבלה נפרדת, בשם SELECTED.
- קשר בין שם הלוקוח לשם על כרטיס האשראי - הנקנו כי השם על כרטיס האשראי (Name\_On\_Card) עשוי להיות שונה מהשם בטבלת הלוקוחות (לדוגמא – בן משפחה שמשתמש בכרטיס אחר), לכן לא נעשו ניסיון אחד בון השדות.
- חוסר שימוש ב-**NULL** במידת האפשר – במודול נעשה ניסיון לצמצם שימוש בשדות NULL, בעיקר בטבלאות קשר בהן שמננו דגש על הוספנו שדות רק אם הם הכרחיים למודול. לדוגמא, בטבלת SELECTED כל שדה הוא חלק מהמפתח ולכן לא יכול להכיל ערכים חסרים.

### פירוט שדות בסיס הנתונים והערכת גודלם

Table	Field	Data Type	Size (Bytes)
<b>CUSTOMERS</b>	<u>Customer_ID</u>	Int	4
	First_Name	Varchar(20)	20
	Last_Name	Varchar(20)	20
	Email	Varchar(40)	40
	<b>Total Record Size (Bytes)</b>		<b>84</b>
<b>CREDIT_CARDS</b>	<u>Card_Number</u>	Varchar(19)	19
	Expiration_Date	Date	3
	CCV	Varchar(3)	3
	Name_On_Card	Varchar(40)	40
	<u>Customer_ID (CUSTOMERS)</u>	Int	4
	<b>Total Record Size (Bytes)</b>		<b>69</b>
<b>SEARCHES</b>	<u>DT</u>	DateTime	8
	<u>IP_Address</u>	Varchar(15)	15
	Text	Varchar(200)	200

Table	Field	Data Type	Size (Bytes)
	Customer_ID ( <b>CUSTOMERS</b> )	Int	4
	<b>Total Record Size (Bytes)</b>		
<b>ORDERS</b>	<u>Order_ID</u>	Int	4
	Date	Date	3
	Country	Varchar(20)	20
	City	Varchar(20)	20
	Street_Name	Varchar(40)	40
	Street_Number	Int	4
	Apartment	Int	4
	ZIP_Code	Varchar(10)	10
	Prescription	Varchar(512)	512
	Card_Number ( <b>CREDIT CARDS</b> )	Varchar(19)	19
	<b>Total Record Size (Bytes)</b>		
<b>PRODUCTS</b>	<u>Product_ID</u>	Int	4
	Name	Varchar(20)	20
	Description	Varchar(200)	200
	Rating	Decimal(2,1)	5
	Price	Money	8
	<b>Total Record Size (Bytes)</b>		
<b>ORDERABLE_PRODUCTS</b>	<u>Product_ID (<b>PRODUCTS</b>)</u>	Int	4
	<b>Total Record Size (Bytes)</b>		
<b>TOP_FRAMES</b>	<u>Product_ID (<b>ORDERABLE_PRODUCTS</b>)</u>	Int	4
	Color	Varchar(20)	20
	Design	Varchar(20)	20



Table	Field	Data Type	Size (Bytes)
	<b>Total Record Size (Bytes)</b>		<b>44</b>
<b>ORDERED</b>	<u>Product_ID</u> <b>(ORDERABLE_PRODUCTS)</b>	Int	4
	<u>Order_ID</u> <b>(ORDERS)</b>	Int	4
	Units	Int	4
	<b>Total Record Size (Bytes)</b>		<b>12</b>
<b>BASE_FRAMES</b>	<u>Product_ID</u> <b>(PRODUCTS)</b>	Int	4
	Size	Varchar(20)	20
	Shape	Varchar(20)	20
	Gender	Char	1
	<b>Total Record Size (Bytes)</b>		<b>45</b>
<b>VERSIONS</b>	<u>Product_ID</u> <b>(BASE_FRAMES)</b>	Int	4
	<u>Version</u>	Int	4
	Description	Varchar(200)	200
	Extra_Price	Money	8
	<b>Total Record Size (Bytes)</b>		<b>216</b>
<b>CUSTOMIZATIONS</b>	<u>Feature</u>	Varchar(40)	40
	<u>Selection</u>	Varchar(40)	40
	<b>Total Record Size (Bytes)</b>		<b>80</b>
<b>GLASSES_ORDERED</b>	<u>Product_ID</u> <b>(VERSIONS)</b>	Int	4
	<u>Version</u> <b>(VERSIONS)</b>	Int	4
	<u>Order_ID</u> <b>(ORDERS)</b>	Int	4
	Units	Int	4
	<b>Total Record Size (Bytes)</b>		<b>16</b>
<b>SELECTED</b>	<u>Product_ID</u> <b>(VERSIONS)</b>	Int	4

Table	Field	Data Type	Size (Bytes)
	<u>Version (VERSIONS)</u>	Int	4
	<u>Feature (CUSTOMIZATIONS)</u>	Varchar(40)	40
	<u>Selection (CUSTOMIZATIONS)</u>	Varchar(40)	40
	<b>Total Record Size (Bytes)</b>		<b>88</b>
REVIEWS	<u>Review_ID</u>	Int	4
	Date	Date	3
	Headline	Varchar(40)	40
	Text	Varchar(200)	200
	Rating	Int	4
	<u>Customer_ID (CUSTOMERS)</u>	Int	4
	<u>Product_ID (PRODUCTS)</u>	Int	4
	<b>Total Record Size (Bytes)</b>		<b>259</b>

#### הנחות והסבירים נוספים:

- האתר מציג דירוגים של מוצרים עד ספרה אחת לאחר הנקודה (לדוגמא: 4.6), לכן בחרנו את הטיפוס DECIMAL(2,1) עבור השדה Rating בישיות: Product, Orderable\_Product, Base\_Frame, Top\_Frame.
- הנחנו כי כל מסגרת בסיס (Base\_Frame) מתאימה למגדל אחד בלבד (זכר או נקבה).
- הנחנו כי רחוב אין אותיות, לדוגמא: אין רחוב שמספרו A21. לכן, עבור השדה Street\_Number בישות ORDERS בтип Int.
- הנחנו כי המספר הסודי על כרטיס האשראי הוא באורך של 3 ספרות בלבד, ועל כן סוג השדה הוא VARCHAR(3).
- הנחנו כי תאריך התפוגה של כרטיס האשראי נשמר כטיפוס Date. כאשר משתמש מזין תאריך תפוגה, לדוגמא: 06/06/2026, אז תאריך התפוגה שנשמר במסד הנתונים הוא 01/06/2026 והתאריך האחרון בו היה ניתן להשתמש בכרטיס האשראי הוא 31/05/2026.



- הנחנו כי את מרשם האופטומטריסט שהמשתמש מעלה לבסיס הנתונים - הוא מעלה כ קישור לתמונה (כליום כתובת URL), לכן בחרנו בטיפוס VARCHAR.
- כל בחירת תחומי התווים עבור שדות מסווג VARCHAR נעשתה תחת היגיון בריא והבנת הצורך של השדה הרלוונטי. לדוגמה, הנחנו כי הגבלה לטקסט חופשי של ביקורת על מוצר כלשהו היא עד 200 תוויים ולכן בחרנו טיפוס VARCHAR(200) עבור השדה Text בישות Review.



### מטרה 3 (30%) - עיצוב פיסי (בסיס נתונים פעיל על שרת **RDBMS**)

#### אלוצי מרחיב ערכים אלוצי בדיקה (CHECK)

- אלוץ על EMAIL: (CUSTOMERS)

```
CONSTRAINT CK_Email CHECK (Email LIKE '%@%.%'),  
CONSTRAINT UQ_Email UNIQUE (Email))
```

הוספנו אלוץ לבדוק שכותבת המail מכילה את הסימנים '@' ו'.'; במבנה בסיסי של כתובת mail. כתובת המail משמשת לצורכי זיהוי, שליחת פרטי הזמנה ותקשורת עם הלקוח, ולכן חייבי לוודא שהיא כתובה בפורמט תקני. בנוסף וידנו שכותבת המail הנה **יחודית** ולא נמצאת במערכת שלנו אצל לקוחות אחרים.

- אלוץ על PRICE: (PRODUCTS)

```
CONSTRAINT CK_Price CHECK (Price >= 0),
```

מחיר המוצר לא יכול להיות שלילי.

- אלוץ על RATING: (PRODUCTS)

```
CONSTRAINT CK_Products_Rating CHECK (Rating BETWEEN 0 AND 5))
```

דירוג מוצר חייב להיות בין 0 ל-5, על פי הסולם המקובל של דירוגים. נתן זה נועד לשקף את חוות דעת הלקוחות, ולכן יש להגביל אותו לטווח הגיוני.

- אלוצים על EXPIRATION DATE & CARD NUMBER & CVV: (CREDIT\_CARDS)

```
CONSTRAINT CK_CCV CHECK (CCV LIKE '[0-9][0-9][0-9]'),  
CONSTRAINT CK_NUMBER CHECK (Card_Number LIKE '[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]-  
[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]'),  
CONSTRAINT CK_Expiration CHECK (Expiration_Date > GETDATE())()
```

1. קוד אבטחה (CCV) חייב להיות בן שלוש ספרות.
2. מספר כרטיס האשראי חייב להיות בפורמט של 16 ספרות המחולקות ל-4 קבוצות, מופרדות באמצעות מקף.



3. תאריך התפוגה של הלקוח חיב להיות בעתיד. לא ניתן להשתמש בכרטיס שפג תוקפו ולכן יש לוודא שהוא בתוקף.

• **אילוצים על (ORDERS) :ZIP CODE & APARTMENT & STREET NUMBER**

```
CONSTRAINT CK_STREET_NUMBER CHECK (Street_Number > 0),  
CONSTRAINT CK_APARTMENT CHECK (Apartment > 0),  
CONSTRAINT CK_ZIP_CODE CHECK (Zip_Code NOT LIKE '%[^0-9]%' )
```

1. מספר בית חייב להיות חיובי. לא ניתן לשלוח האמנה לכתובת עם מספר שלילי או אפס.
2. מספר דירה גם הוא חייב להיות חיובי.
3. קוד דואר חייב להכיל ספרות בלבד. אין לאפשר תווים שאינם מספריים על מנת לוודא שליחת משלוחים בצורה תקינה.

• **אילוץ על (ORDERED) :UNITS**

```
CONSTRAINT CK_Units CHECK (Units > 0)
```

מספר היחידות בהזמנה חייב להיות לפחות 1. לא ניתן להזמין כמות שלילית כמוון או 0.

• **אילוצים על (BASE\_FRAMES) :GENDER & SIZE & SHAPE**

```
CONSTRAINT CK_Gender CHECK (Gender IN ('M', 'F')),  
CONSTRAINT CK_Size CHECK (Size IN ('Extra Narrow', 'Narrow', 'Medium', 'Wide', 'Extra Wide')),  
CONSTRAINT CK_Shape CHECK (Shape IN ('Rectangular', 'Round')) )
```

1. כפי שציינו בהנחות, המשקפיים מסווגים לפי מגדר – גבר או אישה ולכן יש לוודא שהערך שירך לאחד מהם.
2. הגדרת מידת המסגרת מוגבלת למידות סטנדרטיות בלבד, כדי להבטיח התאמת תקנית למוצרים קיימים.
3. הצורה של המשקפיים חייבת להיות אחת מהצורות הנתמכות – עגולה או מלבנית – כדי לשמר על תקניות העיצוב.

• **אילוץ על (VERSIONS) : EXTRA PRICE**

```
Constraint CK_Extra_Price CHECK (Extra_Price >= 0) )
```

תוספת למחיר עבור גרסה מסוימת של מוצר חייבת להיות אפס או יותר.



• **AILUZ על (GLASSES\_ORDERED) :GLASSES ORDERED UNTIS**

`CONSTRAINT CK_GLASSES_ORDERED_Units CHECK (Units > 0)`

בדומה להזנות רגילים – גם כאן יש לוודא שכמה משקפים שהזמנה חיובית ולא אפס.

• **AILUZ על (REVIEWS) :REVIEWS**

`CONSTRAINT CK_Reviews_Rating CHECK (Rating BETWEEN 1 AND 5 )`

גם כאן, הדירוג בביקורת חייב להיות בין 1 ל-5, לפי סולם הדירוג.

**טבלאות חיפוש (Lookup Tables)**

- **טבלת חיפוש ל-Colors:** יצרנו טבלת חיפוש נפרדת עבור צבעים מתוך מטרה לוודא שכל צבע שנבחר במסגרת הגדרת מוצר (למשל מסגרת משקפים או עדשות) יהיה מתוך רשימת צבעים תקפה בלבד.

```
CREATE TABLE COLORS (
Color Varchar(20) NOT NULL,
CONSTRAINT PK_COLORS PRIMARY KEY (Color),
)

INSERT INTO COLORS (Color) VALUES
('Black'),
('Tortoise'),
('White'),
('Blue'),
('Red'),
('Gray'),
('Green'),
('Rainbow'),
('Lavender')
```

```
ALTER TABLE TOP_FRAMES
ADD CONSTRAINT FK_TOP_FRAMES_COLORS foreign key(Color) references COLORS(Color)
```



## לבנית הטבלאות **SCRIPT**

```
CREATE TABLE CUSTOMERS (
    Customer_ID Int NOT NULL,
    First_Name Varchar(20),
    Last_Name Varchar(20),
    Email Varchar(40),
    CONSTRAINT PK_CUSTOMERS primary key(Customer_ID),
    CONSTRAINT CK_Email CHECK (Email LIKE '%@%.%'),
    CONSTRAINT UQ_Email UNIQUE (Email))

CREATE TABLE PRODUCTS (
    Product_ID Int NOT NULL,
    [Name] Varchar(20),
    [Description] Varchar(200),
    Rating Decimal(2,1),
    Price Money,
    CONSTRAINT PK_PRODUCTS primary key(Product_ID),
    CONSTRAINT CK_Price CHECK (Price >= 0),
    CONSTRAINT CK_Products_Rating CHECK (Rating BETWEEN 0 AND 5))

CREATE TABLE CREDIT_CARDS (
    Card_Number Varchar(19) NOT NULL,
    Expiration_Date Date,
    CCV Varchar(3),
    Name_On_Card Varchar(40),
    Customer_ID Int,
    CONSTRAINT PK_CREDIT_CARDS primary key(Card_Number),
    CONSTRAINT CK_CCV CHECK (CCV LIKE '[0-9][0-9][0-9]'),
    CONSTRAINT CK_NUMBER CHECK (Card_Number LIKE '[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9]'),
    CONSTRAINT CK_Expiration CHECK (Expiration_Date > GETDATE()))

ALTER TABLE CREDIT_CARDS
    ADD CONSTRAINT FK_CREDIT_CARDS_CUSTOMERS foreign key(Customer_ID)
references CUSTOMERS(Customer_ID)

CREATE TABLE SEARCHES (
    DT DateTime NOT NULL,
    IP_Address Varchar(15),
    [Text] Varchar(200),
    Customer_ID Int,
    CONSTRAINT PK_SEARCHES primary key(DT,IP_Address))

ALTER TABLE SEARCHES
    ADD CONSTRAINT FK_SEARCHES_CUSTOMERS foreign key(Customer_ID) references
CUSTOMERS(Customer_ID)
```



```
CREATE TABLE ORDERS (
    Order_ID Int NOT NULL,
    [Date] Date,
    Country Varchar (20),
    City Varchar (20),
    Street_Name Varchar (40),
    Street_Number Int,
    Apartment Int,
    Zip_Code Varchar(10),
    Prescription Varchar(512),
    Card_Number Varchar(19),
    CONSTRAINT PK_ORDERS primary key(Order_ID),
    CONSTRAINT CK_STREET_NUMBER CHECK (Street_Number > 0),
    CONSTRAINT CK_APARTMENT CHECK (Apartment > 0),
    CONSTRAINT CK_ZIP_CODE CHECK (Zip_Code NOT LIKE '%[^0-9]%' ))
```

```
ALTER TABLE ORDERS
    ADD CONSTRAINT FK_ORDERS_CREDIT_CARDS foreign key (Card_Number) references
CREDIT_CARDS (Card_Number)
```

```
CREATE TABLE ORDERABLE_PRODUCTS(
    Product_ID int NOT NULL,
    CONSTRAINT PK_ORDERABLE_PRODUCTS primary key(Product_ID) )
```

```
ALTER TABLE ORDERABLE_PRODUCTS
    ADD CONSTRAINT FK_ORDERABLE_PRODUCTS_PRODUCTS foreign key (Product_ID)
references PRODUCTS (Product_ID)
```

```
CREATE TABLE COLORS (
    Color Varchar(20) NOT NULL,
    CONSTRAINT PK_COLORS PRIMARY KEY (Color),
    )
```

```
CREATE TABLE TOP_FRAMES(
    Product_ID int NOT NULL,
    Color Varchar(20),
    Design Varchar(20),
    CONSTRAINT PK_TOP_FRAMES primary key(Product_ID) )
```

```
ALTER TABLE TOP_FRAMES
    ADD CONSTRAINT FK_TOP_FRAMES_ORDERABLE_PRODUCTS foreign key(Product_ID)
references ORDERABLE_PRODUCTS (Product_ID)
```

```
ALTER TABLE TOP_FRAMES
    ADD CONSTRAINT FK_TOP_FRAMES_COLORS foreign key(Color) references
COLORS(Color)
```

```
CREATE TABLE ORDERED(
    Product_ID Int NOT NULL,
    Order_ID Int,
    Units Int,
```



```

CONSTRAINT PK_ORDERED primary key(Product_ID, Order_ID),
CONSTRAINT CK_Units CHECK (Units > 0)
)

ALTER TABLE ORDERED
    ADD CONSTRAINT FK_ORDERED_ORDERABLE_PRODUCTS foreign key(Product_ID)
references ORDERABLE_PRODUCTS(Product_ID)
ALTER TABLE ORDERED
    ADD CONSTRAINT FK_ORDERED_ORDERS foreign key(Order_ID) references
ORDERS(Order_ID)

CREATE TABLE BASE_FRAMES (
    Product_ID Int NOT NULL,
    Size Varchar(20),
    Shape Varchar(20),
    Gender Char,
    CONSTRAINT PK_BASE_FRAMES primary key(Product_ID),
    CONSTRAINT CK_Gender CHECK (Gender IN ('M', 'F')),
    CONSTRAINT CK_Size CHECK (Size IN ('Extra Narrow', 'Narrow', 'Medium',
'Wide', 'Extra Wide')),
    CONSTRAINT CK_Shape CHECK (Shape IN ('Rectangular', 'Round')) )

ALTER TABLE BASE_FRAMES
    ADD CONSTRAINT FK_BASE_FRAMES_PRODUCTS foreign key(Product_ID) references
PRODUCTS(Product_ID)

CREATE TABLE VERSIONS(
    Product_ID Int NOT NULL,
    [Version] Int NOT NULL,
    [Description] Varchar(200),
    Extra_Price Money,
    Constraint PK_VERSIONS primary key(Product_ID,[Version]),
    Constraint CK_Extra_Price CHECK (Extra_Price >= 0) )

ALTER TABLE VERSIONS
    ADD CONSTRAINT FK_VERSIONS_BASE_FRAMES foreign key(Product_ID) references
BASE_FRAMES(Product_ID)

CREATE TABLE CUSTOMIZATIONS(
    Feature Varchar (40) NOT NULL,
    Selection Varchar (40) NOT NULL ,
    CONSTRAINT PK_CUSOMIZATIONS primary key (Feature, Selection))

CREATE TABLE GLASSES_ORDERED(
Product_ID Int NOT NULL,
[Version] Int NOT NULL,
Order_ID Int NOT NULL,

```



```
Units Int,  
CONSTRAINT PK_GLASSES_ORDERERED primary key (Product_ID, [Version], Order_ID),  
CONSTRAINT CK_GLASSES_ORDERED_Units CHECK (Units > 0)  
)  
  
ALTER TABLE GLASSES_ORDERED  
    ADD CONSTRAINT FK_GLASSES_ORDERED VERSIONS foreign key (Product_ID,  
[Version]) references VERSIONS (Product_ID, [Version])  
  
ALTER TABLE GLASSES_ORDERED  
    ADD CONSTRAINT FK_GLASSES_ORDERED_ORDERS foreign key (Order_ID) references  
ORDERS (Order_ID)  
  
CREATE TABLE SELECTED(  
    Product_ID Int NOT NULL,  
    [Version] Int NOT NULL,  
    Feature Varchar(40) NOT NULL,  
    Selection Varchar(40) NOT NULL,  
    Constraint PK_SELECTED primary key(Product_ID,[Version], Feature,  
Selection))  
  
ALTER TABLE SELECTED  
    ADD CONSTRAINT FK_SELECTED VERSIONS foreign key (Product_ID, [Version])  
references VERSIONS (Product_ID, [Version])  
  
ALTER TABLE SELECTED  
    ADD CONSTRAINT FK_SELECTED_CUSTOMIZATIONS foreign key (Feature, Selection)  
references CUSTOMIZATIONS (Feature, Selection)  
  
CREATE TABLE REVIEWS(  
    Review_ID Int NOT NULL,  
    [Date] Date,  
    HeadLine Varchar(40),  
    [Text] Varchar(200),  
    Rating Int,  
    Customer_ID Int,  
    Product_ID Int,  
    CONSTRAINT PK_REVIEWS primary key(Review_ID),  
    CONSTRAINT CK_Reviews_Rating CHECK (Rating BETWEEN 1 AND 5) )  
  
ALTER TABLE REVIEWS  
    ADD CONSTRAINT FK_REVIEWS_CUSTOMERS foreign key (Customer_ID) references  
CUSTOMERS (Customer_ID)  
  
ALTER TABLE REVIEWS  
    ADD CONSTRAINT FK_REVIEWS_PRODUCTS foreign key (Product_ID) references  
PRODUCTS (Product_ID)
```



## להנת רשומות SCRIPT

```
INSERT INTO COLORS (Color) VALUES
('Black'),
('Tortoise'),
('White'),
('Blue'),
('Red'),
('Gray'),
('Green'),
('Rainbow'),
('Lavender')

INSERT INTO CUSTOMERS VALUES
(1, 'Alice', 'Levi', 'alice.levi@gmail.com'),
(2, 'Ben', 'Cohen', 'ben.cohen@bgu.com'),
(3, 'Dana', 'Mizrahi', 'dana.mizrahi@walla.com'),
(4, 'Eli', 'Shalev', 'eli.shalev@gmail.com'),
(5, 'Noa', 'Bar', 'noa.bar@bgu.com')

INSERT INTO CREDIT_CARDS VALUES
('1234-5678-9012-3456', '2026-07-01', '123', 'Alice Levi', 1),
('2345-6789-0123-4567', '2027-01-01', '456', 'Ben Cohen', 2),
('3456-7890-1234-5678', '2028-12-01', '789', 'Dana Mizrahi', 3),
('4567-8901-2345-6789', '2026-10-01', '147', 'Eli Shalev', 4),
('5678-9012-3456-7890', '2029-05-01', '258', 'Noa Bar', 5)

INSERT INTO SEARCHES VALUES
('2025-05-01 10:30', '192.168.0.1', 'blue light glasses', 1),
('2025-05-02 09:45', '192.168.0.2', 'sunglasses for men', 2),
('2025-05-03 14:15', '192.168.0.3', 'clear frame', 3),
('2025-05-04 18:20', '192.168.0.4', 'smart glasses', 4),
('2025-05-05 21:00', '192.168.0.5', 'flexible glasses', 5)

INSERT INTO PRODUCTS (Product_ID, Name, Description, Rating, Price) VALUES
(101, 'The Gemstone', 'Top Frame in blue camo pattern.', 4.3, 25),
(102, 'The Black', 'A top frame for those who wants to stay classy.', 4.6, 30),
(103, 'The Navy', 'A top frame for true lovers of the sea.', 4.5, 28),
(104, 'Glasses Case', 'Our protective glasses case will make you feel safer.', 4.8, 20),
(105, 'The Kirby', 'This medium rectangle frame is stylish and timeless.', 4.5, 70),
(106, 'The Larkin', 'This narrow, modified rectangle frame is a fan-favorite.', 4.8, 70),
(107, 'The Murphy', 'This wide frame has an oversized, new-fashioned feel.', 4.8, 70),
(108, 'The Finley', 'This narrow frame is extremely versatile and effortlessly cool.', 4.4, 70)

INSERT INTO ORDERABLE_PRODUCTS (Product_ID) VALUES
```



(101),  
(102),  
(103),  
(104)

```
INSERT INTO TOP_FRAMES (Product_ID, Color, Design) VALUES  
(101, 'Rainbow', 'Prints'),  
(102, 'Black', 'Classic'),  
(103, 'Blue', 'Pattern')
```

```
INSERT INTO BASE_FRAMES (Product_ID, Size, Shape, Gender) VALUES  
(105, 'Medium', 'Rectangular', 'M'),  
(106, 'Narrow', 'Rectangular', 'M'),  
(107, 'Wide', 'Round', 'F'),  
(108, 'Narrow', 'Round', 'M')
```

```
INSERT INTO ORDERS (Order_ID, [Date], Country, City, Street_Name, Street_Number,  
Apartment, Zip_Code, Prescription, Card_Number) VALUES  
(1, '2025-05-15', 'USA', 'New York', '5th Avenue', 123, 10, '1000123456',  
'https://picsum.photos/seed/1/600/400', '1234-5678-9012-3456'),  
(2, '2025-05-16', 'Canada', 'Toronto', 'Queen St', 45, 22, '2000123456',  
'https://picsum.photos/seed/2/600/400', '2345-6789-0123-4567'),  
(3, '2025-05-17', 'UK', 'London', 'Baker St', 221, 5, '3000123456',  
'https://picsum.photos/seed/3/600/400', '3456-7890-1234-5678'),  
(4, '2025-05-18', 'Australia', 'Sydney', 'George St', 10, 1, '4000123456',  
'https://picsum.photos/seed/4/600/400', '4567-8901-2345-6789'),  
(5, '2025-05-19', 'Germany', 'Berlin', 'Unter den Linden', 55, 12, '5000123456',  
NULL, '5678-9012-3456-7890')
```

```
INSERT INTO ORDERED VALUES  
(101, 1, 1),  
(102, 2, 2),  
(103, 3, 1),  
(104, 4, 3)
```

```
INSERT INTO VERSIONS VALUES  
(105, 1, 'Material: Metal, Lenses: Single Vision , Lens Type: Premium, Lenses:  
Blue-Light Filtering', 100),  
(105, 2, 'Material: Acetate, Lenses: Progressive Lenses, Lens Type: Plus ', 150),  
(106, 1, ' Material: Acetate, Lenses: Light Responsive ', 0),  
(107, 1, 'Lenses: Blue-Light Filtering, Lenses: Readers ', 50),  
(108, 1, 'Lens Color: Black', 0)
```

```
INSERT INTO CUSTOMIZATIONS (Feature, Selection) VALUES  
('Material', 'Metal'),  
('Material', 'Acetate'),  
(('Lenses', 'Single Vision'),  
(('Lenses', 'Progressive Lenses'),  
(('Lens Type', 'Premium'),  
(('Lens Type', 'Standard'),  
(('Lenses', 'Blue-Light Filtering'),  
(('Lenses', 'Light Responsive'),  
(('Lens Color', 'Black'),  
(('Lens Color', 'Brown'))
```

```
INSERT INTO GLASSES_ORDERED VALUES
```



```
(105, 1, 1, 1),  
(106, 1, 2, 2),  
(107, 1, 3, 1),  
(108, 1, 4, 3)  
  
INSERT INTO SELECTED_VALUES  
(105, 1, 'Lenses', 'Light Responsive'),  
(105, 2, 'Lens Type', 'Premium'),  
(106, 1, 'Lenses', 'Progressive Lenses'),  
(107, 1, 'Material', 'Acetate'),  
(108, 1, 'Lens Type', 'Standard')  
  
INSERT INTO REVIEWS VALUES  
(1, '2025-05-11', 'Excellent!', 'Very smart and useful', 5, 1, 107),  
(2, '2025-05-12', 'Nice glasses', 'They filter the blue light well', 4, 2, 105),  
(3, '2025-05-13', 'Cool shades', 'Stylish and effective', 5, 3, 106),  
(4, '2025-05-14', 'So-so', 'Very basic! cant say that I LOVE wearing it to be  
honest...', 2, 4, 102),  
(5, '2025-05-15', 'Flexible but light', 'Comfortable to wear', 4, 5, 105),  
(6, '2025-05-17', 'Best case ever', 'I LOVE this case! Awesome material and very  
protective', 5, 5, 104)
```

---

## להסרת טבלאות ואילוצים **SCRIPT**

```
DROP TABLE SELECTED  
DROP TABLE GLASSES_ORDERED  
DROP TABLE REVIEWS  
DROP TABLE ORDERED  
DROP TABLE TOP_FRAMES  
DROP TABLE VERSIONS  
DROP TABLE BASE_FRAMES  
DROP TABLE ORDERABLE_PRODUCTS  
DROP TABLE ORDERS  
DROP TABLE SEARCHES  
DROP TABLE CREDIT_CARDS  
DROP TABLE PRODUCTS  
DROP TABLE CUSTOMERS  
DROP TABLE CUSTOMIZATIONS  
DROP TABLE COLORS
```



## נספח SCRIPT הכלל את כל הקוד – יצרת טבלאות (כולל Lookup) עם אילוצים הכנסת נתונים

```
DROP TABLE IF EXISTS SELECTED
DROP TABLE IF EXISTS GLASSES_ORDERED
DROP TABLE IF EXISTS REVIEWS
DROP TABLE IF EXISTS ORDERED
DROP TABLE IF EXISTS TOP_FRAMES
DROP TABLE IF EXISTS VERSIONS
DROP TABLE IF EXISTS BASE_FRAMES
DROP TABLE IF EXISTS ORDERABLE_PRODUCTS
DROP TABLE IF EXISTS ORDERS
DROP TABLE IF EXISTS SEARCHES
DROP TABLE IF EXISTS CREDIT_CARDS
DROP TABLE IF EXISTS PRODUCTS
DROP TABLE IF EXISTS CUSTOMERS
DROP TABLE IF EXISTS CUSTOMIZATIONS
DROP TABLE IF EXISTS COLORS

CREATE TABLE CUSTOMERS (
    Customer_ID Int NOT NULL,
    First_Name Varchar(20),
    Last_Name Varchar(20),
    Email Varchar(40),
    CONSTRAINT PK_CUSTOMERS primary key(Customer_ID),
    CONSTRAINT CK_Email CHECK (Email LIKE '%@%.%'),
    CONSTRAINT UQ_Email UNIQUE (Email))

CREATE TABLE PRODUCTS (
    Product_ID Int NOT NULL,
    [Name] Varchar(20),
    [Description] Varchar(200),
    Rating Decimal(2,1),
    Price Money,
    CONSTRAINT PK_PRODUCTS primary key(Product_ID),
    CONSTRAINT CK_Price CHECK (Price >= 0),
    CONSTRAINT CK_Products_Rating CHECK (Rating BETWEEN 0 AND 5))

CREATE TABLE CREDIT_CARDS (
    Card_Number Varchar(19) NOT NULL,
    Expiration_Date Date,
    CCV Varchar(3),
    Name_On_Card Varchar(40),
    Customer_ID Int,
    CONSTRAINT PK_CREDIT_CARDS primary key(Card_Number),
    CONSTRAINT CK_CCV CHECK (CCV LIKE '[0-9][0-9][0-9]'),
    CONSTRAINT CK_NUMBER CHECK (Card_Number LIKE '[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]'),
    CONSTRAINT CK_Expiration CHECK (Expiration_Date > GETDATE()))

ALTER TABLE CREDIT_CARDS
    ADD CONSTRAINT FK_CREDIT_CARDS_CUSTOMERS foreign key(Customer_ID)
references CUSTOMERS(Customer_ID)
```



```
CREATE TABLE SEARCHES (
    DT DateTime NOT NULL,
    IP_Address Varchar(15),
    [Text] Varchar(200),
    Customer_ID Int,
    CONSTRAINT PK_SEARCHES primary key(DT,IP_Address))  
  
ALTER TABLE SEARCHES
    ADD CONSTRAINT FK_SEARCHES_CUSTOMERS foreign key(Customer_ID) references
CUSTOMERS(Customer_ID)  
  
CREATE TABLE ORDERS (
    Order_ID Int NOT NULL,
    [Date] Date,
    Country Varchar (20),
    City Varchar (20),
    Street_Name Varchar (40),
    Street_Number Int,
    Apartment Int,
    Zip_Code Varchar(10),
    Prescription Varchar(512),
    Card_Number Varchar(19),
    CONSTRAINT PK_ORDERS primary key(Order_ID),
    CONSTRAINT CK_STREET_NUMBER CHECK (Street_Number > 0),
    CONSTRAINT CK_APARTMENT CHECK (Apartment > 0),
    CONSTRAINT CK_ZIP_CODE CHECK (Zip_Code NOT LIKE '%[^0-9]%' ))  
  
ALTER TABLE ORDERS
    ADD CONSTRAINT FK_ORDERS_CREDIT_CARDS foreign key (Card_Number) references
CREDIT_CARDS (Card_Number)  
  
CREATE TABLE ORDERABLE_PRODUCTS(
    Product_ID int NOT NULL,
    CONSTRAINT PK_ORDERABLE_PRODUCTS primary key(Product_ID) )  
  
ALTER TABLE ORDERABLE_PRODUCTS
    ADD CONSTRAINT FK_ORDERABLE_PRODUCTS_PRODUCTS foreign key (Product_ID)
references PRODUCTS (Product_ID)  
  
CREATE TABLE COLORS (
Color Varchar(20) NOT NULL,
CONSTRAINT PK_COLORS PRIMARY KEY (Color),
)  
  
CREATE TABLE TOP_FRAMES(
    Product_ID int NOT NULL,
    Color Varchar(20),
    Design Varchar(20),
    CONSTRAINT PK_TOP_FRAMES primary key(Product_ID) )
```



```

ALTER TABLE TOP_FRAMES
    ADD CONSTRAINT FK_TOP_FRAMES_ORDERABLE_PRODUCTS foreign key(Product_ID)
references ORDERABLE_PRODUCTS(Product_ID)

ALTER TABLE TOP_FRAMES
    ADD CONSTRAINT FK_TOP_FRAMES_COLORS foreign key(Color) references
COLORS(Color)

CREATE TABLE ORDERED(
    Product_ID Int NOT NULL,
    Order_ID Int,
    Units Int,
    CONSTRAINT PK_ORDERED primary key(Product_ID, Order_ID),
    CONSTRAINT CK_Units CHECK (Units > 0)
)

ALTER TABLE ORDERED
    ADD CONSTRAINT FK_ORDERED_ORDERABLE_PRODUCTS foreign key(Product_ID)
references ORDERABLE_PRODUCTS(Product_ID)
ALTER TABLE ORDERED
    ADD CONSTRAINT FK_ORDERED_ORDERS foreign key(Order_ID) references
ORDERS(Order_ID)

CREATE TABLE BASE_FRAMES (
    Product_ID Int NOT NULL,
    Size Varchar(20),
    Shape Varchar(20),
    Gender Char,
    CONSTRAINT PK_BASE_FRAMES primary key(Product_ID),
    CONSTRAINT CK_Gender CHECK (Gender IN ('M', 'F')),
    CONSTRAINT CK_Size CHECK (Size IN ('Extra Narrow', 'Narrow', 'Medium',
'Wide', 'Extra Wide')),
    CONSTRAINT CK_Shape CHECK (Shape IN ('Rectangular', 'Round')) )

ALTER TABLE BASE_FRAMES
    ADD CONSTRAINT FK_BASE_FRAMES_PRODUCTS foreign key(Product_ID) references
PRODUCTS(Product_ID)

CREATE TABLE VERSIONS(
    Product_ID Int NOT NULL,
    [Version] Int NOT NULL,
    [Description] Varchar(200),
    Extra_Price Money,
    Constraint PK_VERSIONS primary key(Product_ID,[Version]),
    Constraint CK_Extra_Price CHECK (Extra_Price >= 0) )

ALTER TABLE VERSIONS
    ADD CONSTRAINT FK_VERSIONS_BASE_FRAMES foreign key(Product_ID) references
BASE_FRAMES(Product_ID)

```



```
CREATE TABLE CUSTOMIZATIONS(
    Feature Varchar (40) NOT NULL,
    Selection Varchar (40) NOT NULL ,
    CONSTRAINT PK_CUSOMIZATIONS primary key (Feature, Selection))  
  
CREATE TABLE GLASSES_ORDERED(
    Product_ID Int NOT NULL,
    [Version] Int NOT NULL,
    Order_ID Int NOT NULL,
    Units Int,
    CONSTRAINT PK_GLASSES_ORDERERD primary key (Product_ID, [Version], Order_ID),
    CONSTRAINT CK_GLASSES_ORDERED_Units CHECK (Units > 0)
)  
  
ALTER TABLE GLASSES_ORDERED
    ADD CONSTRAINT FK_GLASSES_ORDERED VERSIONS foreign key (Product_ID,
    [Version]) references VERSIONS (Product_ID, [Version])  
  
ALTER TABLE GLASSES_ORDERED
    ADD CONSTRAINT FK_GLASSES_ORDERED_ORDERS foreign key (Order_ID) references
ORDERS (Order_ID)  
  
CREATE TABLE SELECTED(
    Product_ID Int NOT NULL,
    [Version] Int NOT NULL,
    Feature Varchar(40) NOT NULL,
    Selection Varchar(40) NOT NULL,
    Constraint PK_SELECTED primary key(Product_ID,[Version], Feature, Selection))  
  
ALTER TABLE SELECTED
    ADD CONSTRAINT FK_SELECTED VERSIONS foreign key (Product_ID, [Version])
references VERSIONS (Product_ID, [Version])  
  
ALTER TABLE SELECTED
    ADD CONSTRAINT FK_SELECTED_CUSTOMIZATIONS foreign key (Feature, Selection)
references CUSTOMIZATIONS (Feature, Selection)  
  
CREATE TABLE REVIEWS(
    Review_ID Int NOT NULL,
    [Date] Date,
    HeadLine Varchar(40),
    [Text] Varchar(200),
    Rating Int,
    Customer_ID Int,
    Product_ID Int,
    CONSTRAINT PK_REVIEWS primary key(Review_ID),
    CONSTRAINT CK_Reviews_Rating CHECK (Rating BETWEEN 1 AND 5) )
```



```
ALTER TABLE REVIEWS
    ADD CONSTRAINT FK_REVIEWS_CUSTOMERS foreign key (Customer_ID) references
CUSTOMERS (Customer_ID)

ALTER TABLE REVIEWS
    ADD CONSTRAINT FK_REVIEWS_PRODUCTS foreign key (Product_ID) references
PRODUCTS (Product_ID)

INSERT INTO COLORS (Color) VALUES
('Black'),
('Tortoise'),
('White'),
('Blue'),
('Red'),
('Gray'),
('Green'),
('Rainbow'),
('Lavender')

INSERT INTO CUSTOMERS VALUES
(1, 'Alice', 'Levi', 'alice.levi@gmail.com'),
(2, 'Ben', 'Cohen', 'ben.cohen@bgu.com'),
(3, 'Dana', 'Mizrahi', 'dana.mizrahi@walla.com'),
(4, 'Eli', 'Shalev', 'eli.shalev@gmail.com'),
(5, 'Noa', 'Bar', 'noa.bar@bgu.com')

INSERT INTO CREDIT_CARDS VALUES
('1234-5678-9012-3456', '2026-07-01', '123', 'Alice Levi', 1),
('2345-6789-0123-4567', '2027-01-01', '456', 'Ben Cohen', 2),
('3456-7890-1234-5678', '2028-12-01', '789', 'Dana Mizrahi', 3),
('4567-8901-2345-6789', '2026-10-01', '147', 'Eli Shalev', 4),
('5678-9012-3456-7890', '2029-05-01', '258', 'Noa Bar', 5)

INSERT INTO SEARCHES VALUES
('2025-05-01 10:30', '192.168.0.1', 'blue light glasses', 1),
('2025-05-02 09:45', '192.168.0.2', 'sunglasses for men', 2),
('2025-05-03 14:15', '192.168.0.3', 'clear frame', 3),
('2025-05-04 18:20', '192.168.0.4', 'smart glasses', 4),
('2025-05-05 21:00', '192.168.0.5', 'flexible glasses', 5)

INSERT INTO PRODUCTS (Product_ID, Name, Description, Rating, Price) VALUES
(101, 'The Gemstone', 'Top Frame in blue camo pattern.', 4.3, 25),
(102, 'The Black', 'A top frame for those who wants to stay classy.', 4.6, 30),
(103, 'The Navy', 'A top frame for true lovers of the sea.', 4.5, 28),
(104, 'Glasses Case', 'Our protective glasses case will make you feel safer.', 4.8, 20),
(105, 'The Kirby', 'This medium rectangle frame is stylish and timeless.', 4.5, 70),
(106, 'The Larkin', 'This narrow, modified rectangle frame is a fan-favorite.', 4.8, 70),
```



```

(107, 'The Murphy', 'This wide frame has an oversized, new-fashioned feel.', 4.8,
70),
(108, 'The Finley', 'This narrow frame is extremely versatile and effortlessly
cool.', 4.4, 70)

INSERT INTO ORDERABLE_PRODUCTS (Product_ID) VALUES
(101),
(102),
(103),
(104)

INSERT INTO TOP_FRAMES (Product_ID, Color, Design) VALUES
(101, 'Rainbow', 'Prints'),
(102, 'Black', 'Classic'),
(103, 'Blue', 'Pattern')

INSERT INTO BASE_FRAMES (Product_ID, Size, Shape, Gender) VALUES
(105, 'Medium', 'Rectangular', 'M'),
(106, 'Narrow', 'Rectangular', 'M'),
(107, 'Wide', 'Round', 'F'),
(108, 'Narrow', 'Round', 'M')

INSERT INTO ORDERS (Order_ID, [Date], Country, City, Street_Name, Street_Number,
Apartment, Zip_Code, Prescription, Card_Number) VALUES
(1, '2025-05-15', 'USA', 'New York', '5th Avenue', 123, 10, '1000123456',
'https://picsum.photos/seed/1/600/400', '1234-5678-9012-3456'),
(2, '2025-05-16', 'Canada', 'Toronto', 'Queen St', 45, 22, '2000123456',
'https://picsum.photos/seed/2/600/400', '2345-6789-0123-4567'),
(3, '2025-05-17', 'UK', 'London', 'Baker St', 221, 5, '3000123456',
'https://picsum.photos/seed/3/600/400', '3456-7890-1234-5678'),
(4, '2025-05-18', 'Australia', 'Sydney', 'George St', 10, 1, '4000123456',
'https://picsum.photos/seed/4/600/400', '4567-8901-2345-6789'),
(5, '2025-05-19', 'Germany', 'Berlin', 'Unter den Linden', 55, 12, '5000123456',
NULL, '5678-9012-3456-7890')

INSERT INTO ORDERED VALUES
(101, 1, 1),
(102, 2, 2),
(103, 3, 1),
(104, 4, 3)

INSERT INTO VERSIONS VALUES
(105, 1, 'Material: Metal, Lenses: Single Vision , Lens Type: Premium, Lenses:
Blue-Light Filtering', 100),
(105, 2, 'Material: Acetate, Lenses: Progressive Lenses, Lens Type: Plus ', 150),
(106, 1, ' Material: Acetate, Lenses: Light Responsive ', 0),
(107, 1, 'Lenses: Blue-Light Filtering, Lenses: Readers ', 50),
(108, 1, 'Lens Color: Black', 0)

INSERT INTO CUSTOMIZATIONS (Feature, Selection) VALUES
('Material', 'Metal'),
('Material', 'Acetate'),
('Lenses', 'Single Vision'),
('Lenses', 'Progressive Lenses'),
('Lens Type', 'Premium'),
('Lens Type', 'Standard'),
('Lenses', 'Blue-Light Filtering'),

```



```
( 'Lenses', 'Light Responsive' ),  
( 'Lens Color', 'Black' ),  
( 'Lens Color', 'Brown' )  
  
INSERT INTO GLASSES_ORDERED VALUES  
(105, 1, 1, 1),  
(106, 1, 2, 2),  
(107, 1, 3, 1),  
(108, 1, 4, 3)  
  
INSERT INTO SELECTED VALUES  
(105, 1, 'Lenses', 'Light Responsive'),  
(105, 2, 'Lens Type', 'Premium'),  
(106, 1, 'Lenses', 'Progressive Lenses'),  
(107, 1, 'Material', 'Acetate'),  
(108, 1, 'Lens Type', 'Standard')  
  
INSERT INTO REVIEWS VALUES  
(1, '2025-05-11', 'Excellent!', 'Very smart and useful', 5, 1, 107),  
(2, '2025-05-12', 'Nice glasses', 'They filter the blue light well', 4, 2, 105),  
(3, '2025-05-13', 'Cool shades', 'Stylish and effective', 5, 3, 106),  
(4, '2025-05-14', 'So-so', 'Very basic! cant say that I LOVE wearing it to be  
honest...', 2, 4, 102),  
(5, '2025-05-15', 'Flexible but light', 'Comfortable to wear', 4, 5, 105),  
(6, '2025-05-17', 'Best case ever', 'I LOVE this case! Awesome material and very  
protective', 5, 5, 104)
```