

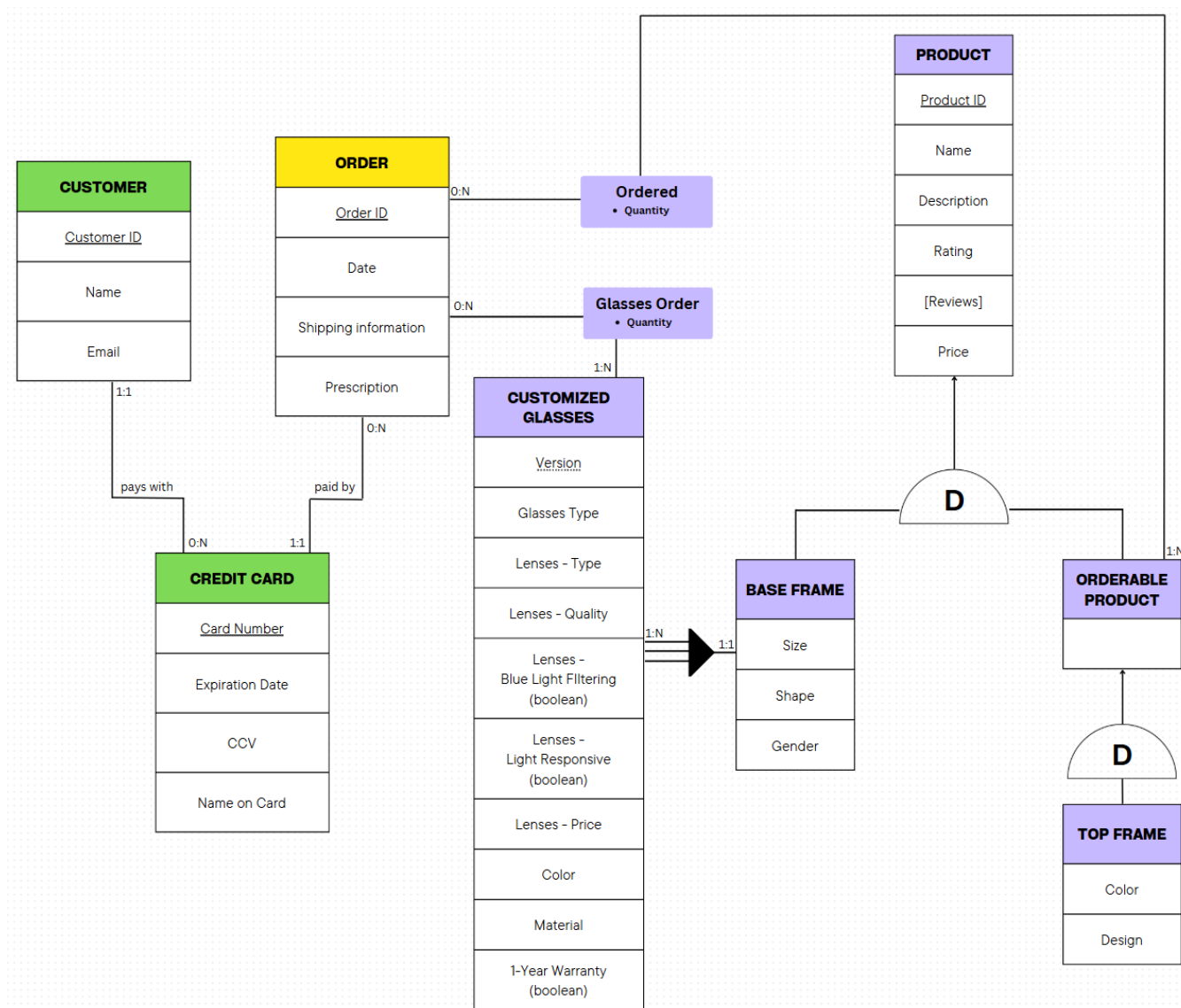


מס' קבוצה	אתר	תאריך הגשה
15	www.paireyewear.com	22/05/2025

פרויקט בסיסי נתונים - חלק ב'

מטלה מקדימה – ה-ERD שהוגש בחלק א'

התרשים בהגשה הקודמת





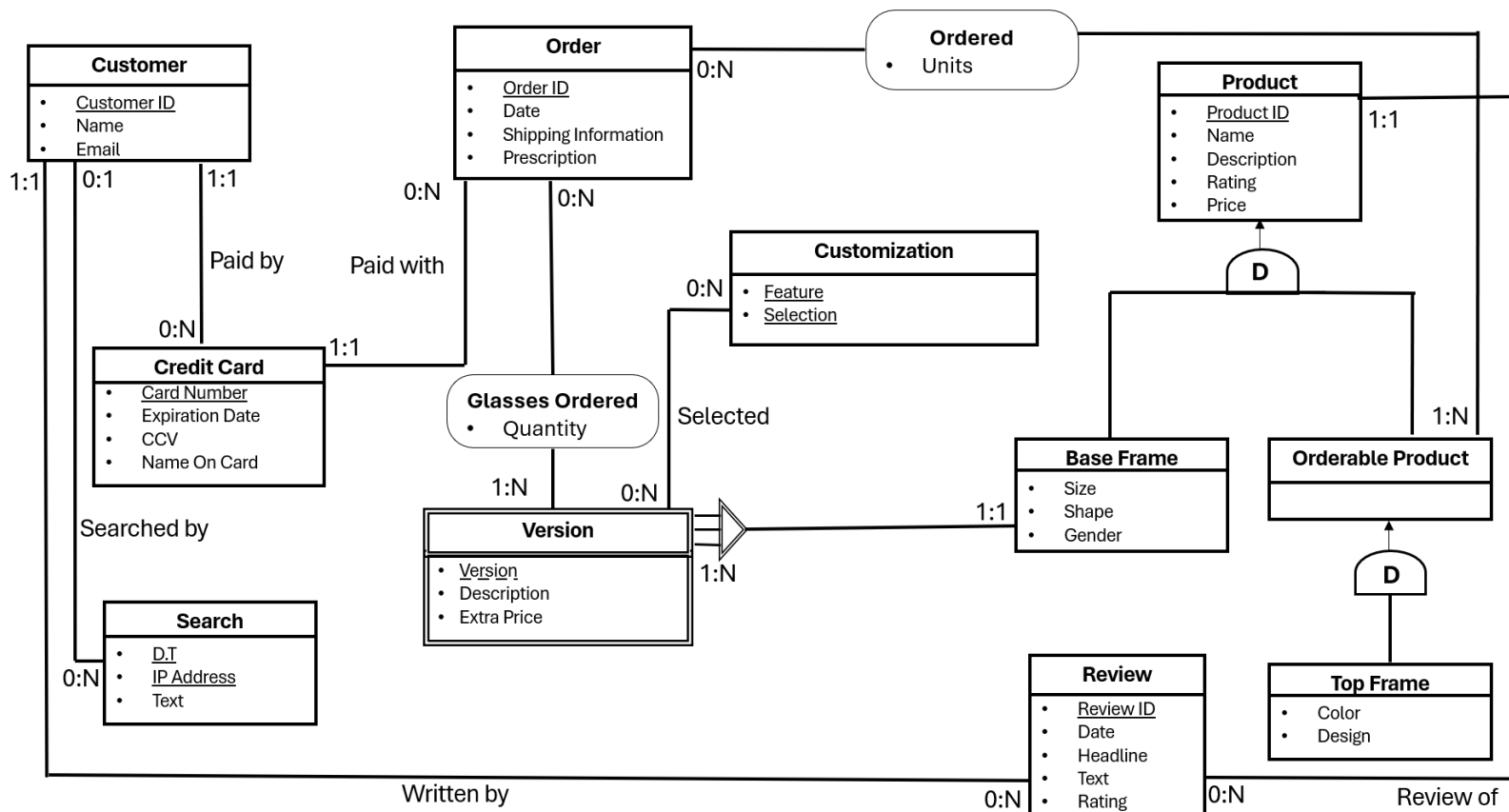
ההנחות שנרשמו בהגשה הקודמת

- הנחנו כי לכל וריאציה של מסגרת בסיס בהתאמה אישית (צבע, חומר, תכונות העדשה וכו') יש מספר סידורי: Version.
- הנחנו כי הדרך היחידה לשלם באתר היא באמצעות כרטיס אשראי.
- ייתכן שיש מגבלה כלשהי באתר לשמירת מס' מוגבל של כרטיסי אשראי פר לקוח, אך לא בדקנו זאת. במקום, הנחנו כי לקוח 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 גם תוכן וגם ציון מה גם שחשוב לדעת מי רשם את חוות הדעת לכן לא מתאים כשדה מרובה ערכים.
- העיצוב אישי שלכם אפשרי אך ניתן לבצע התאמה בשביל שיהיה יותר טוב. אפשר לראות דוגמה שפורסמה במודל בשביל לקבל השראה.

מטלה 1 (40%) – עיצוב קונספטואלי (תרשים ישויות-קשרים)





הנחות מודל ה- ERD

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



מטלה 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 אטומים: First ו-Last על מנת למנוע הפרת נרמול מסדר ראשון.
- נירמול השדה **Shipping Information** השדה Shipping Information בישות Order במודל ה-ERD פוצל ל-6 שדות אטומים על מנת למנוע הפרת נרמול מסדר ראשון:
Country, City, Street_Name, Street_Number, Apartment, ZIP_Code
בנוסף, בהוראות הקורס כתוב לא להתייחס לבעיות נרמול הנובעות מהשדה "מיקוד" ולכן לא התייחסנו אליהן.
- הקשר בין **City** לבין **Country** הנחנו כי ייתכן קיומן 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 **CREDIT CARD** לבין **CUSTOMER** - הנחנו שלכל לקוח (CUSTOMER) יכול להיות יותר מכרטיס אשראי אחד אך כל כרטיס שייך ללקוח אחד בלבד. לכן, שדה Customer_ID מופיע בטבלת CREDIT_CARDS כמפתח זר.



- הקשר בין **VERSION** לבין **CUSTOMIZATION** - הנחנו שכל וריאציה של מסגרת בסיס (VERSION) מקושרת למספר התאמות אישיות (Customization) היוצרות את הוריאציה. כמובן שגם מס' וריאציות של מסגרות בסיס יכולות להיות מקושרות להתאמה אישית אחת - לכן מדובר בקשר רבים לרבים, ויצרנו לכך טבלה נפרדת, בשם **SELECTED**.
- קשר בין שם הלקוח לשם על כרטיס האשראי - הנחנו כי השם על כרטיס האשראי (Name_On_Card) עשוי להיות שונה מהשם בטבלת הלקוחות (לדוגמה - בן משפחה שמשמש בכרטיס אחר), לכן לא נעשה ניסיון לאחד בין השדות.
- חוסר שימוש ב-**NULL** במידת האפשר - במודל נעשה ניסיון לצמצם שימוש בשדות NULL, בעיקר בטבלאות קשר בהן שמנו דגש על הוספת שדות רק אם הם הכרחיים למודל. לדוגמה, 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)
CUSTOMERS	<u>Customer_ID</u>	Int	4
	First_Name	Varchar(20)	20
	Last_Name	Varchar(20)	20
	Email	Varchar(40)	40
	Total Record Size (Bytes)		84
CREDIT_CARDS	<u>Card_Number</u>	Varchar(19)	19
	Expiration_Date	Date	3
	CCV	Varchar(3)	3
	Name_On_Card	Varchar(40)	40
	Customer_ID (CUSTOMERS)	Int	4
	Total Record Size (Bytes)		69
SEARCHES	<u>DT</u>	DateTime	8
	<u>IP_Address</u>	Varchar(15)	15
	Text	Varchar(200)	200



Table	Field	Data Type	Size (Bytes)
	Customer_ID (CUSTOMERS)	Int	4
	Total Record Size (Bytes)		227
ORDERS	<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 (CREDIT CARDS)	Varchar(19)	19
	Total Record Size (Bytes)		636
PRODUCTS	<u>Product_ID</u>	Int	4
	Name	Varchar(20)	20
	Description	Varchar(200)	200
	Rating	Decimal(2,1)	5
	Price	Money	8
	Total Record Size (Bytes)		237
ORDERABLE_PRODUCTS	<u>Product_ID</u> (PRODUCTS)	Int	4
	Total Record Size (Bytes)		4
TOP_FRAMES	<u>Product_ID</u> (ORDERABLE_PRODUCTS)	Int	4
	Color	Varchar(20)	20
	Design	Varchar(20)	20



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



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

הנחות והסברים נוספים:

- האתר מציג דירוגים של מוצרים עד ספרה אחת לאחר הנקודה (לדוגמא: 4.6), לכן בחרנו את הטיפוס DECIMAL(2,1) עבור השדה Rating בישויות:
- Product, Orderable_Product, Base_Frame, Top_Frame
- הנחנו כי כל מסגרת בסיס (Base_Frame) מתאימה למגדר אחד בלבד (זכר או נקבה).
- הנחנו כי במס' רחוב אין אותיות, לדוגמא: אין רחוב שמספרו A21. לכן, עבור השדה Street_Number בישות ORDERS בחרנו בטיפוס Int.
- הנחנו כי המספר הסודי שעל כרטיס האשראי הוא באורך של 3 ספרות בלבד, ועל כן סוג השדה הוא VARCHAR(3).
- הנחנו כי תאריך התפוגה של כרטיס אשראי נשמר כטיפוס Date. כאשר המשתמש מזין תאריך תפוגה, לדוגמא: 06/26, אז תאריך התפוגה שנשמר בבסיס 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 כקישור לתמונה (כלומר כתובת URL), לכן בחרנו בטיפוס VARCHAR.
- כל בחירת תחומי התווים עבור שדות מסוג VARCHAR נעשתה תחת היגיון בריא והבנת הצורך של השדה הרלוונטי. לדוגמא, הנחנו כי ההגבלה לטקסט חופשי של ביקורת על מוצר כלשהו היא עד 200 תווים ולכן בחרנו טיפוס VARCHAR(200) עבור השדה Text בישות Review.



מטלה 3 (30%) - עיצוב פיס (בסיס נתונים פעיל על שרת RDBMS)

אילוצי מרחב ערכים

אילוצי בדיקה (CHECK)

- אילוץ על EMAIL: (CUSTOMERS)

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

הוספנו אילוץ לבדוק שכתובת המייל מכילה את הסימנים '@' ו-'.', במבנה בסיסי של כתובת מייל. כתובת המייל משמשת לצורכי זיהוי, שליחת פרטי הזמנה ותקשורת עם הלקוח, ולכן חיוני לוודא שהיא כתובה בפורמט תקני. בנוסף ווידאנו שכתובת המייל הנה **ייחודית** ולא נמצאת במערכת שלנו אצל לקוחות אחרים.

- אילוץ על PRICE: (PRODUCTS)

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

מחיר המוצר לא יכול להיות שלילי.

- אילוץ 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. נתון זה נועד לשקף את חוות דעת הלקוחות, ולכן יש להגביל אותו לטווח הגיוני.

- אילוצים 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]'),  
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 הכרטיס חייב להיות בעתיד. לא ניתן להשתמש בכרטיס שפג תוקפו ולכן יש לוודא שהוא בתוקף.



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

```
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 הזמנה 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 **UNTIS**: (ORDERED)

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

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

- אילוצים על **GENDER & SIZE & SHAPE**: (BASE_FRAMES)

```
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 שייך רק לאחד מהם.
2. הגדרת מידת המסגרת מוגבלת למידות סטנדרטיות בלבד, כדי להבטיח התאמה תקנית למוצרים קיימים.
3. הצורה של המשקפיים חייבת להיות אחת מהצורות הנתמכות – עגולה או מלבנית – כדי לשמור על תקניות העיצוב.

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

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

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



- אילוף על **GLASSES ORDERED UNTIS**: (GLASSES_ORDERED)

CONSTRAINT CK_GLASSES_ORDERED_Units **CHECK** (Units > 0)

בדומה להזמנות רגילות – גם כאן יש לוודא שכמות המשקפיים שהוזמנה חיובית ולא אפס.

- אילוף 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 משקפיים 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)
```



SCRIPT לבניית הטבלאות

```
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]'),  
    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)
```



SCRIPT להזנת רשומות

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)
```

SCRIPT להסרת טבלאות ואילוצים

```
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** הכולל את כל הקוד – יצירת 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]'),
    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)
```