

# דוח משוב לסטודנט

תאריך בחינה  
יום שישי, 27 בדצמ' 2024

מרצה  
רועי פלאוס

ניקוד שאלות פתוחות	ציון מבחן מקורי	ציון מבחן סופי
84.00	84.00	84.00

## סיכום

מספר שאלה	הערות	ניקוד	ניקוד מירבי
1		28.00	40.00
2		25.00	30.00
3	יפה!	30.00	30.00
4	בנוס- קוד מסודר וקריא	1.00	10.00

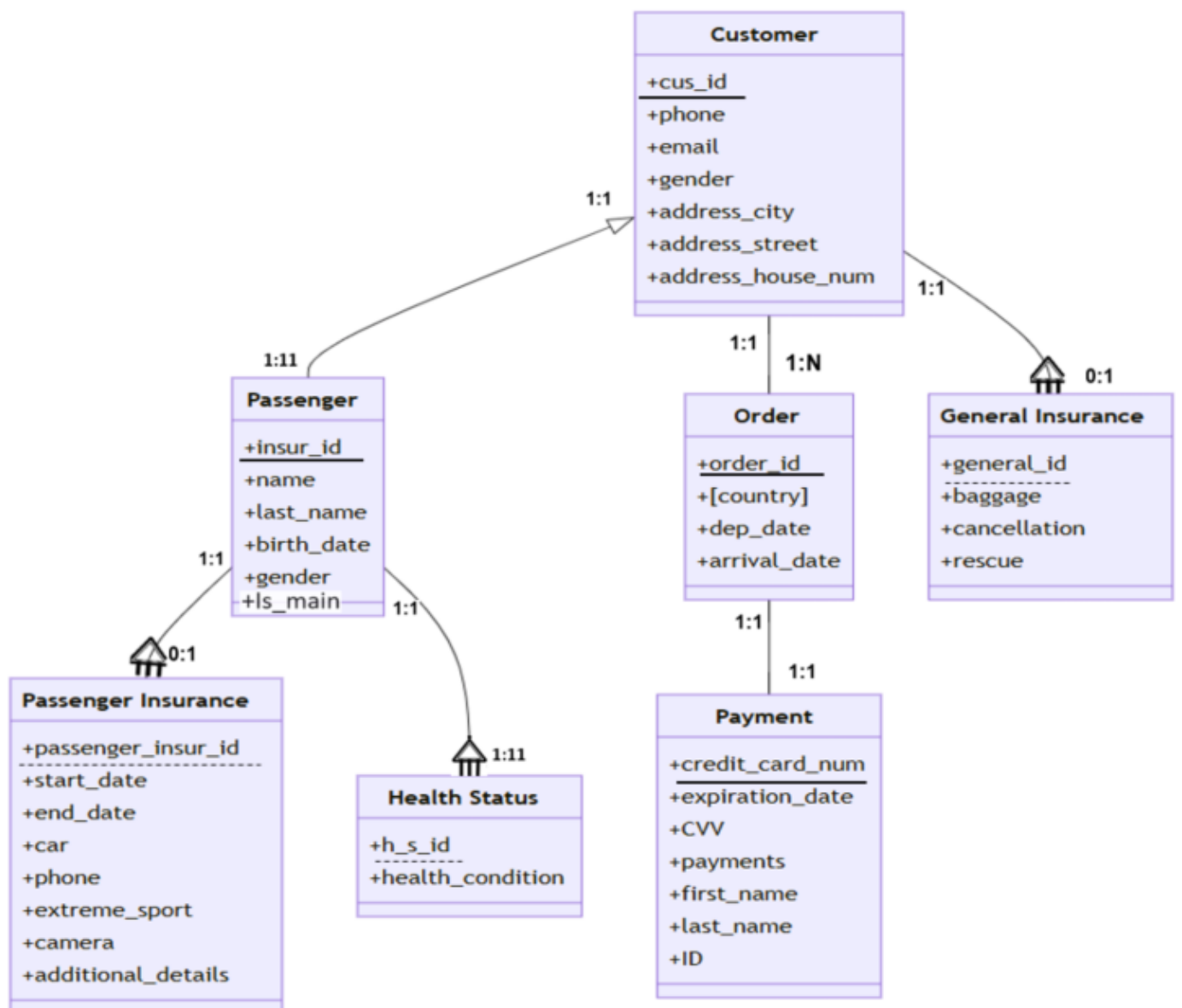


מס' קבוצה	אתר	תאריך הגשה
23	<a href="https://smart.fnx.co.il/Travel/client/stageOne/whereWeFlight">https://smart.fnx.co.il/Travel/client/stageOne/whereWeFlight</a>	16/12/2024
חברי הצוות – מספרי ת.ז.		
206836686	206947400	207975699

## פרויקט בסיסי נתונים – דוגמא להגשת חלק ב'

### מטלה מקדימה – ה-ERD שהוגש בחלק א'

#### התרשים בהגשה הקודמת





## **ההנחות שנרשמו בהגשה הקודמת**

הנחות על המודל:

- תחת costumer – ייתכן שמזמין הביטוח כבר הזמין ביטוח בעבר ופרטיו קיימים במערכת והם מוזנים אוטומטית על ידי הזדהות בטלפון של אותו לקוח, אחרת יש להזין את הפרטים ורק בסוף ההזמנה הפרטים החדשים נכנסים למאגר הלקוחות הקבוע.
- Costumer – ישות זו אינה מתארת נוסע ספציפי אלא ישות חזקה שעל בסיסה מוזנים כל הנוסעים תחת ישות החלשה passenger והאבחנה בין המזמין הראשי לשאר הנוסעים מתבצעת על ידי השדה הבינארי "is main".
- Passenger insurance – ישות זו מייצגת את הביטוחים האישיים שמוכללים רק על הנוסעים שבחרו להרחיב את הביטוח הבסיסי.
- General insurance – ישות זו מייצגת את הביטוחים הקבוצתיים, כלומר הרחבה כוללת לביטוח בלי אופציה להפריט אנשים ספציפיים ממנה.
- Payment – הנחנו כמה הנחות בנוגע לאמצעי התשלום:
  - האתר תומך רק בכרטיסי אשראי כאמצעי תשלום (בנוסף מפורט בהנחות הנוספות).
  - רמת הפירוט הנדרשת הינה מספר כרטיס אשראי, תוקף, cvv, תעודת זהות ושם מלא.
  - הנחנו שהתשלום מתבצע באתר בחלק מהתהליך העסקי ולא על ידי גורם שלישי (איש שירות, paypal וכו')

## **ההערות שניתנו להגשה הקודמת**

1. מה זו צורת הכתיבה של החץ הריק? הקפידו להבא על סגנון הרישום שנלמד בקורס.
2. שימו לב שלישות CUSTOMER יש את אותן תכונות כמו של PASSENGER ובנוסף אליהן תכונות נוספות - זה רומז לנו על מודל של הורשה (אולי לזה התכוונתם עם החץ הריק)
3. שימו לב לקרדינליות של הקשרים - אם יש לנו את מספר ההזמנה נרצה לדעת מי הלקוחות שנכללים בה, מה הצהרת הבריאות שלהם בהזמנה הזו ואילו סוגי ביטוחים הם קנו - ומכאן נחשב את המחיר הסופי. המודל שלכם לא מאפשר לי להבין אילו ביטוחים/ הצהרות בריאות קשורים להזמנה
4. לפי המידול שלכם לכל לקוח יש 0 או 1 ביטוחים כלליים, אבל זה לא הגיוני. יכול להיות שאותו לקוח יבצע כמה הזמנות ובכל אחת מהן הוא יוסיף General Insurance והמודל שלכם אינו מאפשר זאת.
5. כבודה, ביטול נסיעה וחילוץ הם כולם שמות של ביטוחים כלליים - כלומר אלו מופעים שונים של הישות.
6. אפשר לייצר ישות חזקה של "ביטוח" שהמזהה שלה יהיה שם הביטוח (ביטול טיסה, טלפון, כבודה) והיו לה שדות נוספים אם תרצו (למשל, מחיר בסיסי ליום). כיישות חזקה תוכלו לקשר אותה לישויות אחרות (חישבו איך) כך שנוכל לדעת איזה לקוח רכש איזה ביטוח בכל הזמנה שביצע.
7. קרדינליות לא נכונה. אפשר לשלם עם אותו אשראי על כמה הזמנות שונות, ולכן לא הגיוני שזה קשר יחיד ליחיד
8. למה זו ישות חלשה של PASSENGER? עכשיו אותו מטייל יזמין כמה פעמים ביטוח, איך נדע איזו הצהרת בריאות שייכת לאיזו פוליסת ביטוח?



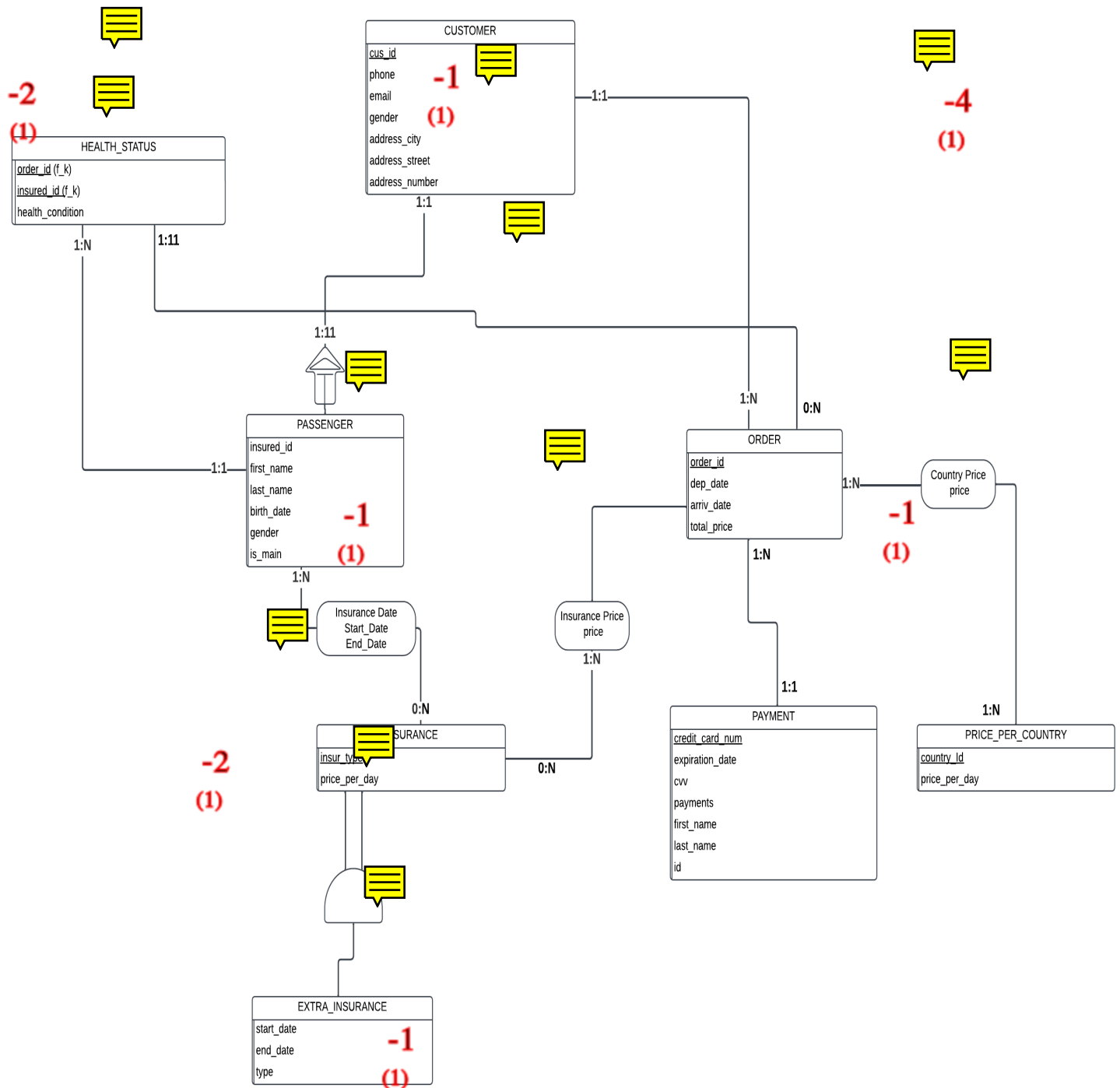
9. מה לגבי מחירים? לכל סוג כיסוי יש מחיר כלשהו, וחלקם משתנים לפי הצהרת הבריאות של האדם, למשל יקרים יותר לנשים בהריון/ מבוגרים.
10. במודל שלכם כרגע אין איך להבין כמה כל הזמנה עלתה
11. לא מאוד מובן למה General Insurance הוא יישות חלשה של לקוח, ואילו Passenger Insurance הוא יישות חלשה של מטייל.
12. שימו לב שבשניהם יש מחיר ליחידה (גם בכיסוי הבסיסי, אם תוסיפו עוד אנשים המחיר יעלה) ובשניהם נרצה לדעת איזה כיסויים כל נוסע רכש. נראה שמדובר באותה היישות ורק הסתבכתם עם המידול.
13. אותה ההערה כמו - General Insurance ביטוח רכב, ביטוח טלפון וכו' הם סוגים של ביטוחים שהנוסע יכול לעשות, ולכן יותר הגיוני שהם יהיו המזהה של ישות חזקה ולא שדות שלה.
14. בנוסף, תוכלו לראות שלביטוחי ספורט רכב האתר מבקש תאריכים, כלומר מדובר בסוג של ביטוח שיש לו עוד תכונות (תאריך התחלה וסיום) - זה רומז לנו על מודל של הורשה



28  
(1)

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

> תרשים ERD המתוקן <





## הנחות מודל ה- ERD

מרכיב	הנחות
הוספת is_main כשדה ב"passenger"	הוספנו שדה זה בכדי לדעת האם הנוסע הוא המזמין הראשי שאחראי על התשלום או נוסע נוסף.
השארת שתי הישויות "customer" & "passenger"	בחרנו להשאיר את שתי הישויות האלו מכיוון שלנוסע הראשי (המזמין) יש תכונות נוספות שלנוסע רגיל אין.
קישור ישות "health status" ל"passenger" ו"order"	קישרנו את הישויות מכיוון שלכל נוסע יכול להיות הצהרות בריאות שונות בהזמנות שונות.
פיצול הישות "country" לשתי ישויות	פיצלנו את הישות בכדי לנרמל מסדר ראשון שדה "country" הוא מרובה ערכים מכיוון שבכל הזמנה יכולות להיות כמה מדינות לכן פיצלנו לעוד ישות של "price per day"
שדה "date" בקשר בין "insurance" ל"passenger"	שדה זה מסמל את תאריך ההמראה ונחיתה וכאשר יורדים לישות "extra insurance" ניתן לעשות את הביטוחים המתאימים בתאריכים ספציפיים (כגון השכרת רכב, ביטוח ואתגרי וכו').





25

(2)

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

**customers** (cus\_Id, phone, email, gender, address\_city, address\_street, address-number)**passengers** (cus\_Id (customers), insured\_Id, first\_name, last\_name, Birth\_date, gender, is\_main)**orders** (order\_Id, depdate, arrivdate, price, cus\_Id (customers), credit\_card\_num (payments))**prices per country** (Country\_id, Price per day)

-1

**payments** (credit\_card\_num, expiration\_date, CVV, payments, first\_name, last\_name, Id)

(2)

**Healthstatus** (order\_Id (orders), {insurance\_id, cus\_Id} (passenger), health\_condition)**insurances** (insurance\_type, price per day)**insurance\_dates** (insur\_type (insurances), {insured\_id, cus\_Id} (passenger), start\_date, end\_date)

-1

**insurance\_prices** (insur\_type (insurances), order\_Id (orders), Price)

(2)

**extra\_insurances** (extra\_insurance\_type (insurance), start\_date, end\_date, type)**countries price** (country\_Id (prices\_per\_country), order\_Id (orders), price)**הנחות והסברים נוספים:**

- **ייצוג קשר רבים לרבים בין insurance\_prices ל-orders ו-insurance** - כל הזמנה עשויה לכלול מספר סוגי ביטוח וסוגי ביטוח עשויים להופיע במספר הזמנות שונות. הבחירה לממש קשר זה באמצעות טבלה התקבלה מתוך שיקול של מניעת שדות מיותר בשתי הישויות ושומרת על המודל גמיש ומאורגן בצורה שמאפשרת חישובים מהירים ויעילים.
- **שיקול חלוקה בין טבלת insurance\_dates ל-insurance** - הוחלט להפריד בין מידע על סוג הביטוח ובין תאריכים הביטוח מתוך הרצון למנוע כפילויות בסוגי הביטוחים ובמחירים שלהם ובנוסף לאפשר גמישות בניהול תאריכים עבור סוגי ביטוחים שונים לנוסעים שונים, החלטה זו נבחרה לאחר שנשקלה חלופה שבה היו מאחדים את הטבלאות אך היא נדחתה משום שהייתה גורמת לכפילות מידע.
- **קשר רבים לרבים בין orders ו-country ל-country-price** - כל הזמנה עשויה לכלול מספר מדינות ולכל מדינה יש מספר הזמנות, לכן בכדי למנוע שדות מיותרים בשתי הישויות בחרנו ביישום טבלה על הקשר כך המודל יהיה מאורגן ויעיל.



• נרמול:

**פיצול טבלת order גם ל-price\_per\_country** - פיצלנו את הישיות לשתי טבלאות לאחר נרמול מסדר ראשון, מכיוון שבכל ביטוח יכולים להיכלל כמה מדינות זה שדה מורכב ולכן צריך לפצל אותו לישות משל עצמו.

## פירוט שדות בסיס הנתונים והערכת גודלם

### Database Table Schema

#### Customers

Table	Field	Data Type	Size (Bytes)	Key
<b>Customers</b>	<u>Cus_ID</u>	Integer	4	Primary Key
	Phone	Varchar(20)	20	
	Email	Varchar(50)	50	
	Gender	Varchar(10)	10	
	Adress_City	Varchar(20)	20	
	Adress_Street	Varchar(20)	20	
	Adress_Number	Varchar(10)	10	
	<b>Total Record Size (Bytes)</b>		<b>134</b>	

-1  
(2)

#### Passengers

Table	Field	Data Type	Size (Bytes)	Key
<b>Passengers</b>	<u>Cus_ID</u> (Customers)	Integer	4	Primary Key, Foreign Key
	<u>Insured_ID</u>	Integer	4	Primary Key
	First_Name	Varchar(20)	20	
	Last_Name	Varchar(20)	20	
	Birthdate	DateTime	8	

-2  
(2)





	Gender	Varchar(10)	10	
	Is_Main	Bit	1	
	<b>Total Record Size (Bytes)</b>		<b>67</b>	

## Order


Table	Field	Data Type	Size (Bytes)	Key
<b>Order</b>	<u>Order_ID</u>	Integer	4	Primary Key
	Dep_Date	DateTime	8	
	Arriv_Date	DateTime	8	
	Total_Price	Decimal(10,2)	9	
	<u>Cus_ID</u> (Customers)	Integer	4	Foreign Key
	<u>Credit Card Num</u> (Payment)	Varchar(16)	16	Foreign Key
	<b>Total Record Size (Bytes)</b>		<b>49</b>	

## Price\_Per\_Country

Table	Field	Data Type	Size (Bytes)	Key
<b>Price_Per_Country</b>	<u>Country_ID</u>	Varchar(20)	20	Primary Key, Foreign Key
	Price_Per_Day	Decimal(10,2)	9	
	<b>Total Record Size (Bytes)</b>		<b>29</b>	



## Payment

Table	Field	Data Type	Size (Bytes)	Key
<b>Payment</b>	<u>Credit Card Number</u>	Varchar(16)	16	Primary Key
	Exp_Date	DateTime	8	
	CVV	Varchar(3)	3 	
	Payments	Integer	4	
	First_Name	Varchar(20)	20	
	Last_Name	Varchar(20)	20	
	ID	Integer	4	
	<b>Total Record Size (Bytes)</b>		<b>75</b>	

## Health\_Status

Table	Field	Data Type	Size (Bytes)	Key
<b>Health_Status</b>	<u>Order ID (Orders)</u>	Integer	4	Primary Key, Foreign Key
	<u>Insur ID (Passengers)</u>	Integer	4	Primary Key, Foreign Key
	<u>Cus ID (Passengers)</u>	Integer	4	Primary Key, Foreign Key
	Health_Condition	Varchar(50)	50	
	<b>Total Record Size (Bytes)</b>		<b>62</b>	

## Insurance

Table	Field	Data Type	Size (Bytes)	Key
-------	-------	-----------	--------------	-----



<b>Insurance</b>	<u>Insur_Type</u>	Varchar(50)	50	Primary Key
	Price_Per_Day	Decimal(10,2)	9	
	<b>Total Record Size (Bytes)</b>		<b>59</b>	

### Insur\_Date

Table	Field	Data Type	Size (Bytes)	Key
<b>Insur_Date</b>	Insur_Type (Insurance)	Varchar(50)	50	Foreign Key
	Insur_ID(Passengers)	Integer	4	Foreign Key
	Cus_ID (Passengers)	Integer	4	Foreign Key
	Start_Date	DateTime	8	
	End_Date	DateTime	8	
	<b>Total Record Size (Bytes)</b>		<b>74</b>	

### Insur\_Price

Table	Field	Data Type	Size (Bytes)	Key
<b>Insur_Price</b>	Insur_Type (Insurance)	Varchar(50)	50	Foreign Key
	Order_ID (Orders)	Integer	4	Foreign Key
	Price	Decimal(10,2)	9	
	<b>Total Record Size (Bytes)</b>		<b>63</b>	

### Extra\_Ins

Table	Field	Data Type	Size (Bytes)	Key
<b>Extra_Ins</b>	<u>Extra Insur_Type (Insurance)</u>	Varchar(50)	50	Primary Key, Foreign Key



	Start_Date	DateTime	8	
	End_Date	DateTime	8	
	Type	Varchar(50)	50	
	<b>Total Record Size (Bytes)</b>		<b>116</b>	

### Country\_Price

Table	Field	Data Type	Size (Bytes)	Key
<b>Country_Price</b>	Order_ID (Orders)	Integer	4	Foreign Key
	Country_ID (Price_per_country)	Varchar(20)	20	Foreign Key
	Price	Decimal(10,2)	9	
	<b>Total Record Size (Bytes)</b>		<b>33</b>	

### הנחות והסברים נוספים:

#### הנחות על שדות בסיסי הנתונים:

- בישות customer הגדרנו את מספר הטלפון כvarchar מכיוון שיכולים להיות תווים שאינם מספרים ובint מספר לא יכול להתחיל באפס.
- בישות passenger השדה is\_main מוגדר כbit כלומר הוא ערך בינארי של 0 ו-1.
- בישות payments השדה credit\_card\_number מוגדר כvarchar מאותן סיבות כמו בטבלת customer.
- בישות health\_status השדה health\_condition מוגדר מסוג varchar(50) כיוון שנצפה שיכיל רצף של תשובות לשאלות הרפואיות.





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

30  
(3)

יפה!

### אילוצי מרחב ערכים

אילוצי בדיקה (CHECK):



#### טבלת Insurance

- אילוץ על השדה Price\_Per\_Day

אילוץ זה מוודא שהמחיר ליום חייב להיות חיובי, מכיוון שמחיר שלילי אינו הגיוני בעסקה כלשהי.

**CONSTRAINT**

CHECK (Price\_Per\_Day > 0)

#### טבלת Payment

- אילוץ על השדה Credit\_Card\_Number

מוודא שמספר כרטיס האשראי מכיל בדיוק 16 ספרות. זהו אילוץ קריטי למניעת הזנת נתונים שגויים המשויכים לכרטיסי אשראי.

**CONSTRAINT**

CHECK (LEN(Credit\_Card\_Number) = 16 AND Credit\_Card\_Number LIKE '[0-9]%')

- אילוץ על השדה Exp\_Date

תאריך תפוגת הכרטיס חייב להיות בעתיד, כדי למנוע שימוש בכרטיס שפג תוקפו.

**CONSTRAINT**

CHECK (Exp\_Date > GETDATE())

- אילוץ על השדה CVV

מוודא שקוד האבטחה (CVV) מורכב מ-3 ספרות בדיוק.

**CONSTRAINT**

CHECK (LEN(CVV) = 3 AND CVV LIKE '[0-9]%')

- אילוץ על השדה Payments

מספר התשלומים חייב להיות שווה ל-0 או חיובי, כי לא ניתן לבצע מספר תשלומים שלילי.

**CONSTRAINT**

CHECK (Payments >= 0)

#### טבלת Customers

- אילוץ על השדה Phone

מספר הטלפון חייב להתחיל בסימן ( + למקרה של מספר בינלאומי) או במספר (למספר מקומי), ואורכו חייב להיות בין 10 ל-20 תווים.

**CONSTRAINT**

CHECK ((Phone LIKE '+%' OR Phone LIKE '[0-9]%') AND LEN(Phone) BETWEEN 10 AND 20)

- אילוץ על השדה Email

ידוע שהשדה מכיל את הסימנים @. - כדי להבטיח תקינות פורמט המייל.



### CONSTRAINT

CHECK (Email LIKE '%@%.%')

### טבלת Gender\_Lookup ו Extra\_Ins\_Type\_Lookup

- אילוץ ייחודיות (UNIQUE) על הערכים מונע כפילות של סוגי ביטוח נוספים ומגדרים.

### טבלת Price\_Per\_Country

- אילוץ על השדה Price\_Per\_Day מוודא שמחיר הביטוח לפי יום חייב להיות חיובי, כדי להבטיח נתונים ריאליים.

### CONSTRAINT

CHECK (Price\_Per\_Day > 0)

### טבלת Passengers

- אילוץ על השדה Birthdate תאריך הלידה חייב להיות שווה או קטן מהתאריך הנוכחי, כדי למנוע הזנת תאריכים עתידיים שגויים.

### CONSTRAINT

CHECK (Birthdate <= GETDATE())

- אילוץ על השדה Is\_Main שדה זה מוודא שסטטוס המבוטח הוא ראשי (1) או לא ראשי (0).

### CONSTRAINT

CHECK (Is\_Main IN (0, 1))

### טבלת Orders

- אילוץ על השדה Dep\_Date מוודא שתאריך היציאה אינו לפני התאריך הנוכחי, כדי למנוע הזנת תאריכים שכבר חלפו.

### CONSTRAINT

CHECK (Dep\_Date >= GETDATE())

- אילוץ על השדה Total\_Price המחיר הכולל חייב להיות חיובי.

### CONSTRAINT

CHECK (Total\_Price > 0)



### טבלת Health\_Status

- **אילוץ על השדה Health\_Condition**  
מוודא שהשדה אינו ריק (מכיל טקסט עם אורך גדול מ-0).  
**CONSTRAINT**  
CHECK (LEN(Health\_Condition) > 0)

### טבלת Insur\_Date

- **אילוץ על השדה Start\_Date**  
מוודא שתאריך התחלת הביטוח שווה או מאוחר לתאריך הנוכחי.  
**CONSTRAINT**  
CHECK (Start\_Date >= GETDATE())

### טבלת Insur\_Price

- **אילוץ על השדה Price**  
מוודא שמחיר הביטוח חייב להיות חיובי.  
**CONSTRAINT**  
CHECK (Price > 0)

### טבלת Country\_Price

- **אילוץ על השדה Price**  
כמו בשדות מחיר אחרים, וידוא שהערך חייב להיות גדול מאפס.  
**CONSTRAINT**  
CHECK (Price > 0)



### טבלאות חיפוש (Lookup Tables):

#### 1. טבלת Extra\_Ins\_Type\_Lookup

מטרה: טבלה זו משמשת כטבלת חיפוש עבור סוגי ביטוחים נוספים (Extra Insurance Types) שניתן להציע ללקוחות.  
פרטים:  
מזהה (Type\_ID) לכל סוג ביטוח.  
שם הסוג (Type\_Value) עם ערך ייחודי.  
הצדקה:  
מאפשרת לשמור רשימת ערכים מוגדרת מראש ולמנוע טעויות בהכנסת נתונים.  
מאפשרת גמישות לעדכן או להוסיף סוגי ביטוחים חדשים ללא שינוי משמעותי בשאר הטבלאות.



## 2. טבלת Gender\_Lookup

מטרה: טבלת חיפוש עבור מגדרים אפשריים ללקוחות.

פרטים:

מזהה (Gender\_ID) עבור כל ערך מגדר.

ערך מגדר – (Gender\_Value) לדוגמה: Male, Female, Other.

הצדקה:

מבטיחה תקינות של נתונים ושומרת על קונסיסטנטיות בין ערכי מגדר.

ניתנת להרחבה במקרה של הוספת מגדרים נוספים

## 3. טבלת Country\_Lookup

מטרה: טבלת חיפוש של מדינות זמינות.

פרטים:

קוד מדינה – (Country\_ID) לדוגמה US, עבור ארצות הברית.

שם מדינה. (Country\_Name)

הצדקה:

מבטיחה שכל המדינות המוזנות יהיו בעלות פורמט תקין וקוד מוגדר.

מאפשרת לנהל בקלות שינויים ברשימת המדינות מבלי להשפיע על טבלאות התלויות.

### -- Step 1: Independent Tables

CREATE TABLE Insurance (

Insur\_Type VARCHAR(50) PRIMARY KEY NOT NULL,

Price\_Per\_Day DECIMAL(10,2) NOT NULL CHECK (Price\_Per\_Day > 0)

);

CREATE TABLE Payment (

Credit\_Card\_Number VARCHAR(16) PRIMARY KEY NOT NULL CHECK

(LEN(Credit\_Card\_Number) = 16 AND Credit\_Card\_Number LIKE '[0-9]%' ),

Exp\_Date DATETIME NOT NULL CHECK (Exp\_Date > GETDATE()),

CVV VARCHAR(3) NOT NULL CHECK (LEN(CVV) = 3 AND CVV LIKE '[0-9]%' ),

Payments INT NOT NULL CHECK (Payments >= 0),

First\_Name VARCHAR(20) NOT NULL,

Last\_Name VARCHAR(20) NOT NULL,

ID INT NOT NULL

);

CREATE TABLE Customers (

Cus\_ID INT PRIMARY KEY NOT NULL,

Phone VARCHAR(20) NOT NULL CHECK ((Phone LIKE '+%' OR Phone LIKE '[0-9]%' ) AND  
LEN(Phone) BETWEEN 10 AND 20),

Email VARCHAR(50) NOT NULL CHECK (Email LIKE '%@%.%' ),

Gender VARCHAR(10) NOT NULL,

Adress\_City VARCHAR(20) NOT NULL,





```
Address_Street VARCHAR(20) NOT NULL,
Address_Number VARCHAR(10) NOT NULL
);
```

### -- Step 2: Lookup Tables (Created First)

```
CREATE TABLE Extra_Ins_Type_Lookup (
    Type_ID INT PRIMARY KEY IDENTITY(1,1),
    Type_Value VARCHAR(50) NOT NULL UNIQUE
);
```

```
INSERT INTO Extra_Ins_Type_Lookup (Type_Value)
VALUES ('Search and Rescue'),
      ('Trip Cancellation/Shortening'),
      ('Baggage'),
      ('Mobile Phone'),
      ('Laptop/Tablet'),
      ('Sports'),
      ('Event or Experience Tickets'),
      ('Rental Car'),
      ('Camera'),
      ('Medical Expenses in Israel');
```

```
CREATE TABLE Gender_Lookup (
    Gender_ID INT PRIMARY KEY IDENTITY(1,1),
    Gender_Value VARCHAR(10) NOT NULL UNIQUE
);
```

```
INSERT INTO Gender_Lookup (Gender_Value)
VALUES ('Male'), ('Female'), ('Other');
```

```
CREATE TABLE Country_Lookup (
    Country_ID VARCHAR(20) PRIMARY KEY,
    Country_Name VARCHAR(50) NOT NULL
);
```

```
INSERT INTO Country_Lookup (Country_ID, Country_Name)
VALUES
('US', 'United States'), ('GB', 'United Kingdom'), ('FR', 'France'), ('DE', 'Germany'),
('IT', 'Italy'), ('ES', 'Spain'), ('GR', 'Greece'), ('CY', 'Cyprus'),
('TH', 'Thailand'), ('TR', 'Turkey'), ('IL', 'Israel'), ('RU', 'Russia'),
('UA', 'Ukraine'), ('CH', 'Switzerland'), ('NL', 'Netherlands'), ('AT', 'Austria'),
('HU', 'Hungary'), ('CZ', 'Czech Republic'), ('PL', 'Poland'), ('RO', 'Romania'),
```



('BG', 'Bulgaria'), ('PT', 'Portugal'), ('CA', 'Canada'), ('AU', 'Australia'),  
('JP', 'Japan'), ('CN', 'China'), ('IN', 'India'), ('BR', 'Brazil');



### -- Step 3: Dependent Tables

```
CREATE TABLE Price_Per_Country (
    Country_ID VARCHAR(20) PRIMARY KEY NOT NULL,
    Price_Per_Day DECIMAL(10,2) NOT NULL CHECK (Price_Per_Day > 0),
    FOREIGN KEY (Country_ID) REFERENCES Country_Lookup(Country_ID)
);
```

```
CREATE TABLE Passengers (
    Cus_ID INT NOT NULL,
    Insured_ID INT NOT NULL,
    First_Name VARCHAR(20) NOT NULL,
    Last_Name VARCHAR(20) NOT NULL,
    Birthdate DATE NOT NULL CHECK (Birthdate <= GETDATE()),
    Gender VARCHAR(10) NOT NULL,
    Is_Main BIT NOT NULL CHECK (Is_Main IN (0, 1)),
    PRIMARY KEY (Cus_ID, Insured_ID),
    FOREIGN KEY (Cus_ID) REFERENCES Customers(Cus_ID),
    FOREIGN KEY (Gender) REFERENCES Gender_Lookup(Gender_Value)
);
```

```
CREATE TABLE Orders (
    Order_ID INT PRIMARY KEY NOT NULL,
    Dep_Date DATETIME NOT NULL CHECK (Dep_Date >= GETDATE()),
    Arriv_Date DATETIME NOT NULL,
    Total_Price DECIMAL(10,2) NOT NULL CHECK (Total_Price > 0),
    Cus_ID INT NOT NULL,
    Credit_Card_Num VARCHAR(16) NOT NULL,
    FOREIGN KEY (Cus_ID) REFERENCES Customers(Cus_ID)
);
```

```
CREATE TABLE Extra_Ins (
    Extra_Insur_Type VARCHAR(50) NOT NULL,
    Start_Date DATETIME NOT NULL,
    End_Date DATETIME NOT NULL,
    Type VARCHAR(50) NULL,
    PRIMARY KEY (Extra_Insur_Type),
    FOREIGN KEY (Extra_Insur_Type) REFERENCES Insurance(Insur_Type),
    FOREIGN KEY (Type) REFERENCES Extra_Ins_Type_Lookup(Type_Value)
);
```

```
CREATE TABLE Country_Price (
    Order_ID INT NOT NULL,
    Country_ID VARCHAR(20) NOT NULL,
    Price DECIMAL(10,2) NOT NULL CHECK (Price > 0),
    FOREIGN KEY (Order_ID) REFERENCES Orders(Order_ID),
    FOREIGN KEY (Country_ID) REFERENCES Country_Lookup(Country_ID)
);

CREATE TABLE Health_Status (
    Order_ID INT NOT NULL,
    Insur_ID INT NOT NULL,
    Cus_ID INT NOT NULL,
    Health_Condition VARCHAR(50) NOT NULL CHECK (LEN(Health_Condition) > 0),
    PRIMARY KEY (Order_ID, Insur_ID, Cus_ID),
    FOREIGN KEY (Order_ID) REFERENCES Orders(Order_ID),
    FOREIGN KEY (Cus_ID, Insur_ID) REFERENCES Passengers(Cus_ID, Insured_ID)
);

CREATE TABLE Insur_Date (
    Insur_Type VARCHAR(50) NOT NULL,
    Insur_ID INT NOT NULL,
    Cus_ID INT NOT NULL,
    Start_Date DATETIME NOT NULL CHECK (Start_Date >= GETDATE()),
    End_Date DATETIME NOT NULL,
    PRIMARY KEY (Insur_Type, Insur_ID, Cus_ID),
    FOREIGN KEY (Insur_Type) REFERENCES Insurance(Insur_Type),
    FOREIGN KEY (Cus_ID, Insur_ID) REFERENCES Passengers(Cus_ID, Insured_ID)
);

CREATE TABLE Insur_Price (
    Insur_Type VARCHAR(50) NOT NULL,
    Order_ID INT NOT NULL,
    Price DECIMAL(10,2) NOT NULL CHECK (Price > 0),
    PRIMARY KEY (Insur_Type, Order_ID),
    FOREIGN KEY (Insur_Type) REFERENCES Insurance(Insur_Type),
    FOREIGN KEY (Order_ID) REFERENCES Orders(Order_ID)
);

-- Insert data into Insurance
INSERT INTO Insurance (Insur_Type, Price_Per_Day)
VALUES
```



('Search and Rescue', 10.00),  
(('Trip Cancellation/Shortening', 20.00),  
(('Baggage', 15.00),  
(('Mobile Phone', 25.00),  
(('Laptop/Tablet', 30.00),  
(('Sports', 35.00),  
(('Event or Experience Tickets', 40.00),  
(('Rental Car', 50.00),  
(('Camera', 45.00),  
(('Medical Expenses in Israel', 60.00));

#### -- Insert data into Extra\_Ins

```
INSERT INTO Extra_Ins (Extra_Insur_Type, Start_Date, End_Date, Type)
VALUES
('Search and Rescue', '2025-12-25', '2026-01-05', 'Search and Rescue'),
('Baggage', '2025-11-15', '2025-11-30', 'Baggage'),
('Sports', '2025-10-01', '2025-10-10', 'Sports'),
('Mobile Phone', '2025-12-01', '2025-12-20', 'Mobile Phone'),
('Laptop/Tablet', '2025-08-15', '2025-08-25', 'Laptop/Tablet'),
('Camera', '2026-01-10', '2026-01-20', 'Camera'),
('Rental Car', '2025-09-05', '2025-09-15', 'Rental Car'),
('Medical Expenses in Israel', '2025-11-20', '2025-11-27', 'Medical Expenses in Israel');
```

```
INSERT INTO Customers (Cus_ID, Phone, Email, Gender, Adress_City, Adress_Street,
Adress_Number)
VALUES
(1, '+972543210987', 'john.doe@example.com', 'Male', 'Tel Aviv', 'Herzl', '10'),
(2, '+972543210123', 'jane.smith@example.com', 'Female', 'Jerusalem', 'Jaffa', '5A'),
(3, '+972501234567', 'alice.brown@example.com', 'Female', 'Haifa', 'HaNassi', '12'),
(4, '+972507654321', 'bob.williams@example.com', 'Male', 'Beer Sheva', 'Rager', '7B'),
(5, '+972525678910', 'charlie.miller@example.com', 'Male', 'Eilat', 'HaTmarim', '3C'),
(6, '+972507898765', 'david.taylor@example.com', 'Male', 'Ashdod', 'Ben Gurion', '8'),
(7, '+972525123456', 'eva.johnson@example.com', 'Female', 'Netanya', 'Herzel', '1'),
(8, '+972545678901', 'frank.moore@example.com', 'Male', 'Rishon LeZion', 'HaRishonim', '6');
```

```
INSERT INTO Passengers (Cus_ID, Insured_ID, First_Name, Last_Name, Birthdate, Gender,
Is_Main)
VALUES
(1, 100, 'John', 'Doe', '1990-01-01', 'Male', 1),
(2, 101, 'Jane', 'Smith', '1985-03-15', 'Female', 1),
(3, 102, 'Alice', 'Brown', '1992-06-10', 'Female', 1),
(4, 103, 'Bob', 'Williams', '1988-09-23', 'Male', 1),
```



```
(5, 104, 'Charlie', 'Miller', '1994-12-01', 'Male', 1),
(6, 105, 'David', 'Taylor', '2000-05-18', 'Male', 1),
(7, 106, 'Eva', 'Johnson', '1996-04-25', 'Female', 1),
(8, 107, 'Frank', 'Moore', '1998-07-13', 'Male', 1);
```

```
INSERT INTO Orders (Order_ID, Dep_Date, Arriv_Date, Total_Price, Cus_ID, Credit_Card_Num)
VALUES
```

```
(1, '2025-12-25', '2026-01-05', 1500.00, 1, '1234567890123456'),
(2, '2025-11-15', '2025-11-30', 2500.00, 2, '1111222233334444'),
(3, '2025-10-01', '2025-10-10', 1000.00, 3, '5555444433332222'),
(4, '2025-12-01', '2025-12-20', 2000.00, 4, '4444555566667777'),
(5, '2025-08-15', '2025-08-25', 1800.00, 5, '2222333344445555'),
(6, '2026-01-10', '2026-01-20', 3000.00, 6, '6666777788889999'),
(7, '2025-09-05', '2025-09-15', 1200.00, 7, '9999888877776666'),
(8, '2025-11-20', '2025-11-27', 1700.00, 8, '3333444455556666');
```

```
INSERT INTO Price_Per_Country (Country_ID, Price_Per_Day)
```

```
VALUES
```

```
('US', 150.00),
('GB', 120.00),
('FR', 100.00),
('DE', 90.00),
('GR', 80.00),
('CY', 70.00);
```

```
INSERT INTO Country_Price (Order_ID, Country_ID, Price)
```

```
VALUES
```

```
(1, 'US', 750.00),
(1, 'GB', 600.00),
(2, 'FR', 1250.00),
(3, 'DE', 800.00),
(4, 'GR', 950.00),
(5, 'CY', 700.00),
(6, 'US', 1500.00),
(7, 'GB', 900.00);
```

```
INSERT INTO Health_Status (Order_ID, Insur_ID, Cus_ID, Health_Condition)
```

```
VALUES
```

```
(1, 100, 1, 'No No No No'),
(2, 101, 2, '3 No, Yes: Pregnancy'),
(3, 102, 3, '2 No, Yes: Heart Disease'),
(4, 103, 4, 'No No No No'),
```



```
(5, 104, 5, '1 No, Yes: Asthma'),
(6, 105, 6, '2 No, Yes: Diabetes'),
(7, 106, 7, '4 No'),
(8, 107, 8, '3 No, Yes: Allergy');
```

```
INSERT INTO Insur_Date (Insur_Type, Insur_ID, Cus_ID, Start_Date, End_Date)
VALUES
('Search and Rescue', 100, 1, '2025-12-25', '2026-01-05'),
('Baggage', 101, 2, '2025-11-15', '2025-11-30'),
('Sports', 102, 3, '2025-10-01', '2025-10-10'),
('Mobile Phone', 103, 4, '2025-12-01', '2025-12-20'),
('Laptop/Tablet', 104, 5, '2025-08-15', '2025-08-25'),
('Camera', 105, 6, '2026-01-10', '2026-01-20'),
('Rental Car', 106, 7, '2025-09-05', '2025-09-15'),
('Medical Expenses in Israel', 107, 8, '2025-11-20', '2025-11-27');
```

```
INSERT INTO Insur_Price (Insur_Type, Order_ID, Price)
VALUES
('Search and Rescue', 1, 150.00),
('Baggage', 2, 120.00),
('Sports', 3, 100.00),
('Mobile Phone', 4, 250.00),
('Laptop/Tablet', 5, 300.00),
('Camera', 6, 450.00),
('Rental Car', 7, 500.00),
('Medical Expenses in Israel', 8, 600.00);
```

```
INSERT INTO Payment (Credit_Card_Number, Exp_Date, CVV, Payments, First_Name,
Last_Name, ID)
VALUES
('1234567890123456', '2026-01-01', '123', 2, 'John', 'Doe', 1001),
('1111222233334444', '2026-02-15', '456', 3, 'Jane', 'Smith', 1002),
('5555444433332222', '2025-12-30', '789', 1, 'Alice', 'Brown', 1003),
('4444555566667777', '2025-11-20', '321', 4, 'Bob', 'Williams', 1004),
('2222333344445555', '2026-04-10', '654', 2, 'Charlie', 'Miller', 1005),
('6666777788889999', '2026-03-25', '987', 5, 'David', 'Taylor', 1006),
('9999888877776666', '2025-09-15', '111', 3, 'Eva', 'Johnson', 1007),
('3333444455556666', '2025-10-01', '222', 2, 'Frank', 'Moore', 1008);
```



```
IF OBJECT_ID('Insur_Price', 'U') IS NOT NULL DROP TABLE Insur_Price;
IF OBJECT_ID('Insur_Date', 'U') IS NOT NULL DROP TABLE Insur_Date;
IF OBJECT_ID('Health_Status', 'U') IS NOT NULL DROP TABLE Health_Status;
IF OBJECT_ID('Extra_Ins', 'U') IS NOT NULL DROP TABLE Extra_Ins;
IF OBJECT_ID('Country_Price', 'U') IS NOT NULL DROP TABLE Country_Price;

IF OBJECT_ID('Orders', 'U') IS NOT NULL DROP TABLE Orders;
IF OBJECT_ID('Passengers', 'U') IS NOT NULL DROP TABLE Passengers;
IF OBJECT_ID('Price_Per_Country', 'U') IS NOT NULL DROP TABLE Price_Per_Country;
IF OBJECT_ID('Customers', 'U') IS NOT NULL DROP TABLE Customers;

IF OBJECT_ID('Payment', 'U') IS NOT NULL DROP TABLE Payment;
IF OBJECT_ID('Insurance', 'U') IS NOT NULL DROP TABLE Insurance;

-- Lookup Tables (Drop Last)
IF OBJECT_ID('Country_Lookup', 'U') IS NOT NULL DROP TABLE Country_Lookup;
IF OBJECT_ID('Extra_Ins_Type_Lookup', 'U') IS NOT NULL DROP TABLE
Extra_Ins_Type_Lookup;
IF OBJECT_ID('Gender_Lookup', 'U') IS NOT NULL DROP TABLE Gender_Lookup;
```



בנוס- קוד מסודר וקריא