|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מס' קבוצה | אתר | | | תאריך הגשה |
| 16 | <seedsheets.com> | | | DD/03/2024 |
| חברי הצוות - מספרי ת.ז | | | | |
| 318334901 | | 211680509 | 318255882 | |

## פרויקט בסיסי נתונים – חלק ג'

**מטלת חובה מקדימה – תיקון ה-ERD והרחבת בסיס הנתונים**

**התרשים בהגשה הקודמת**

A diagram of a product

Description automatically generated

**המודל הטבלאי בהגשה הקודמת**

A screenshot of a computer

Description automatically generated

**ההערות שניתנו להגשה הקודמת**

1. בצורת המידול הזו, לא בא לידי ביטוי המוצר המעוצב הסופי ויותר מזה, אין התייחסות לאפשרות לחזרתיות- תהיה מחיקה של נתונים.  
   ניתן דוגמא: מוצר בשם A (נניח יריעה מסוג א באתר) מקושרת ל8 מוצרי seed שונים- b,c,d,e,f,g,h,i שכל קשר מכיל כמות 1.  
   ומחר בבוקר נקנה מוצר מעוצב נוסף שגם מורכב ממוצר A עם קשר לSeed בשם b כמות של 8 פעמים.  
   בטבלת הchosen יצטרך להימחק לנו הכמות הקודמת של מוצר A עם seed מסוג B בכמות 1.  
   ועכשיו, איך נדע איזה מוצרים מעוצבים סופיים יש לנו?
2. הפתרון כאן הוא שימוש בישות חלשה. כפי שכתבנו לכם במייל, להיעזר בשקופית 74 מההרצאה.  
   שימו לב לפני חלק ג` לשנות את המודל בהתאם.
3. יש להקפיד להיות תמציתיים, ולא לציין הנחות שמובנות מאליהן.  
   לא הורדו נקודות אבל הקפידו על כך לקראת חלק ג
4. על הדף זה נשמע נורא יפה, אבל זה לא בא לידי ביטוי בERD שלכם.  
   שימו לב שהקשר של הלקוח הוא למוצר מסוג garden שבפועל המזהה שלו הוא שם. כלומר אם מחר הוא יקנה יריעה מסוג A עם קשר chosen מסויים ומחרתיים הוא יקנה יריעה מסוג A עם קשר chosen אחר, עדיין היריעה תיהיה אותה יריעה.  
   הבעיה היא הקשר בין יריעה לSEED.
5. שימו לב שהתבקשתם להגיש סקריפט מלא שכולל את כל השלבים בסדר הנכון.  
   לא הורדו נקודות כי סידרתם את העבודה נכון כך שיכולנו לבחון את הסדר. בפעם הבאה ירדו נקודות.
6. חסר אילוץ בדיקה לסיסמא: עליכם להגדיר אורך סיסמא מינימלי (לא הגיוני שסיסמא תהיה באורך 1), תווים שהם חובה וכו`.  
   חסר טבלת חיפוש לשיטות השילוח.

## עיצוב קונספטואלי מתוקן (תרשים ישויות-קשרים)

A diagram of a product

Description automatically generated

## הנחות מתוקנות של מודל ה-ERD המתוקן

**הערה: השינויים שנדרשו לביצוע בהנחות נמצאים בסוף הטבלה החל מהקשר Planted.**

|  |  |  |
| --- | --- | --- |
| **מרכיב** | **הנחות** | **הערות נוספות** |
| Entity: USER   * email * password | * מפתח ראשי: email – לכל USER יש כתובת email אחת ויחידה. |  |
| Entity: SEARCH   * searchDT * IP\_address * Search\_text | * מפתח ראשי: צירוף השדות searchDT, IP\_address – כל חיפוש מסוים יכול להיעשות בתאריך וזמן מסוים במכשיר בעל כתובת IP מסוימת. * הקשר "Made by" – חיפוש יכול להיעשות על ידי משתמש רשום מסוים או משתמש מזדמן מסוים, בנוסף, כל משתמש (רשום או לא) יכול לעשות אינספור חיפושים או לא לעשות חיפושים כלל. לכן הקרדינליות של קשר זה בצד הישות SERACH היא 0:N ובצד הישות USER היא 0:1. |  |
| Entity: DETAILS   * name * address * company * phone# | * name – שדה זה מכיל את השם המלא של משתמש רשום/לא רשום. באתר שם הלקוח מיוצג כשתי שדות, אך היות ושתי השדות הללו מייצגות תוכן מאותו אופי ובעלות משמעות דומה נתייחס אליהם כאחד. * Address – שדה זה מכיל את הכתובת המלאה של משתמש רשום/לא רשום. באתר כתובת הלקוח מיוצג כחמישה שדות, אך היות וחמשת השדות הללו מייצגות תוכן מאותו אופי ובעלות משמעות דומה נתייחס אליהם כאחד. * מפתח ראשי: צירוף השדות name, address – כל צירוף הינו ייחודי באתר. לכל צירוף כזה צריך להיות שם חברה יחיד (company) ומספר טלפון יחיד (phone#). |  |
| הקשר "Details of" בין USER לDETAILS | * פרטי לקוח מסוימים יכולים להיות שמורים באופן זהה אצל אינספור משתמשים רשומים שונים או לאף משתמש רשום. לכן הקרדינליות של קשר זה בצד הישות USER היא 0:N. * למשתמש רשום מסוים יכולים להיות שמורים אינספור פרטי לקוח שונים או בכלל לא. לכן הקרדינליות של קשר זה בצד הישות DETAILS היא 0:N. |  |
| Entity: ORDER   * orderID * orderDate * shipping\_method * payment\_type | * מפתח ראשי: orderID – מספר סידורי ייחודי, מזהה של הזמנה מסוימת. * payment\_type – שדה המכיל את שם שירות התשלום החיצוני שבו השתמש הלקוח בעת התשלום על ההזמנה. * הקשר "Ordered by" – הזמנה מסוימת יכולה להיעשות ע"י משתמש רשום/לא רשום אחד. לכן הקרדינליות של קשר זה בצד הישות USER היא 0:1. משתמש רשום מסוים יכול לעשות אינספור הזמנות או בכלל לא. לכן הקרדינליות של קשר זה בצד הישות ORDER היא 0:N * הקשר "Addressed to" – בהזמנה מסוימת יכולים להירשם סט פרטי לקוח אחד בלבד. לכן הקרדינליות של קשר זה בצד הישות DETAILS היא 1:1. פרטי לקוח מסוימים יכולים להירשם באינספור הזמנות או בכלל לא. לכן הקרדינליות של קשר זה בצד הישות ORDER היא 0:N. | כפי שצוין בחלק א', כלל תהליך התשלום, על פני כל סוגי התשלום הזמינים באתר, מתקיים על בסיס שירותי תשלום חיצוניים אותם האתר צורך. שניים משירותי התשלום נצרכים מחברת Shopify ונקראים ShopPay, אחד שירות חצי-חיצוני והשני שירות חיצוני, והשלישי נצרך מחברת Klarna, שירות חיצוני. לכן במידת הצורך כדי להשיג את פרטי התשלום במועד מאוחר יותר נשמר שם שירות התשלום ממנו צריך לייבא את פרטי התשלום |
| Entity: PRODUCT   * name * price * discount | * מפתח ראשי: name – לכל מוצר קיים שם ייחודי * price – מחירו של מוצר מסוים יהיה קבוע וניתן לשינוי ע"י בעל העסק בלבד, לא שדה מחושב. שדה זה יכיל מספרים ממשיים גדולים מ0. * discount – הנחה על מוצר מסוים אשר הערך הדיפולטי שלה הוא 0, ערכה של ההנחה תוכל להיות קטנה או שווה לערכו של מחיר המוצר. שדה זה יכיל מספרים ממשיים גדולים או שווים ל0. * קיימים סוגים שונים של מוצרים בעלי תכונות שונות אחת מהשנייה וקיימים המוצרים בעלי התכונות הבסיסיות של מוצר. לכן, ישות PRODUCT היא ישות אב כחלק ממודל הורשה בדידה וחלקית. |  |
| הקשר "Resulted with" בין PRODUCT וSEARCH | * כל חיפוש מסוים יכול להוביל להצגת אינספור מוצרים או את אף אחד מהמוצרים. לכן הקרדינליות של קשר זה בצד הישות PRODUCT היא 0:N. * כל מוצר מסוים יכול להיות מוצג כתוצאה של אינספור חיפושים או של שום חיפוש. לכן הקרדינליות של קשר זה בצד הישות SEARCH היא 0:N. |  |
| הקשר הרקורסיבי "Related to" בין PRODUCT לעצמו | * כל מוצר מסוים יכול להיות מוצר נלווה לאינספור מוצרים או לאף אחד מהמוצרים, ולכל מוצר מסוים יכולים להיות אינספור מוצרים נלווים או אף לא אחד. לכן הקרדינליות של הקשר הרקורסיבי בשתי צדדי הקשר תהיה 0:N. |  |
| הקשר "Included" בין PRODUCT וORDER   * quantity | * כל מוצר מסוים יכול להיכלל באינספור הזמנות או באף אחת מההזמנות. לכן הקרדינליות של קשר זה בצד הישות ORDER היא 0:N. * בכל הזמנה מסוימת יכולים להיכלל אינספור מוצרים ולכל הפחות מוצר אחד. לכן הקרדינליות של קשר זה בצד הישות PRODUCT היא 1:N. * quantity – כמות של מוצר מסוים שנכלל בהזמנה מסוימת. שדה זה יכיל מספרים שלמים גדולים מ0. * גינה בעיצוב אישי איננה מוצר ככל המוצרים ולכן קשר זה לא מבטא את השיוך והכללה של גינה בעיצוב אישי בהזמנה מסויימת |  |
| Child Entity: SEED   * ]seed\_type[ * size * season * sun\_amount | * ישות בת במודל ההורשה של ישות מוצר. * זרעים מסוג מסוים יכולים להימכר באופן עצמאי ללא הכללתם בגינה מסוימת או בעסקת חבילה מסוימת. * [seed\_type] – קטגוריה שמתארת את סוג הצמח, זרע מסוים יכול להימצא במספר קטגוריות. * size – גודל החלקה שהזרע לוקח. יכול להיות או גדול או קטן. * season, sun\_amount – תכונות המכילות מידע אינפורמטיבי על הזרע המוצגות ללקוח באתר. |  |
| Child Entity: GARDEN   * small\_count * large\_count | * ישות בת במודל ההורשה של ישות מוצר. * באתר קיימות גינות בעיצוב אישי וגינות מוכנות מראש אשר. * small\_count, large\_count – תכונות המכילות מספרים ערך שלם לא שלילי. צירופי הערכים האפשריים בשדות אלו יהיו אחד מהשלושה הבאים: small\_count = 8 וגם large\_count = 0 small\_count = 5 וגם large\_count = 1 small\_count = 2 וגם large\_count = 2 |  |
| הקשר "Planted" בין SEED וGARDEN   * quantity | * כל זרע מסוים יכול להיכלל באינספור גינות מעוצבות מראש או באף אחת מהן. לכן הקרדינליות של קשר זה בצד הישות GARDEN היא 0:N. * בכל גינה מעוצבת מראש מסוימת ייכללו מראש בין 1 ל8 זרעים, ובכל גינה בעיצוב אישי לא ייכלל מראש אף זרע. לכן הקרדינליות של קשר זה בצד הישות SEED היא 0:N. * quantity – כמות יחידות של זרע מסוים הכלול בגינה מעוצבת מראש.  שדה זה יכיל מספרים שלמים גדולים מ0 וקטנים מ9. |  |
| Week Entity: DESIGN   * designID | * מפתח חלקי: designID – לכל עיצוב מסוים של גינה בעיצוב אישי מסוימת קיים מספר סידורי ייחודי. * לכל גינה בעיצוב אישי יכולים להתקיים אינספור עיצובים או אף לא אחת. לכן הקרדינליות של הקשר של הישותGARDEN עם הישות החלשה DESIGN אשר תחת חסותה בצד הישות DESIGN היא 0:N. |  |
| הקשר "Chosen" בין SEED וDESIGN   * quantity | * כל זרע מסוים יכול להיבחר להיכלל באינספור עיצובים או באף אחד מהם. לכן הקרדינליות של קשר זה בצד הישות DESIGN היא 0:N. * בכל עיצוב מסוים של גינה בעיצוב אישי מסוימת יכולים להיבחר בין 1 ל8 זרעים. לכן הקרדינליות של קשר זה בצד הישות SEED היא 1:8. * quantity – כמות יחידות של זרע מסוים שנבחר להיכלל בעיצוב מסוים. שדה זה יכיל מספרים שלמים גדולים מ0 וקטנים מ9. |  |
| הקשר "Designed for" בין DESIGN וORDER   * Quantity | * בכל הזמנה מסוימת יכולים להיכלל אינספור גינות בעיצוב אישי או אף לא אחת. לכן הקרדינליות של קשר זה בצד הישות DESIGN היא 0:N. * כל עיצוב מסוים של גינה בעיצוב אישי מסוימת יכול להיכלל בהזמנה אחת בלבד. לכן הקרדינליות של קשר זה בצד הישות ORDER היא 1:1. * quantity – כמות יחידות של גינה בעיצוב אישי מסוימת בעיצוב מסוים הנכללות בהזמנה מסוימת. שדה זה יכיל מספרים שלמים גדולים מ0. |  |

## מודל טבלאי מתוקן

* **PRODUCTS** ( **Name**, Price, Discount )
* **USERS** ( **Email**, Password)
* **SEARCHES** ( **SearchDT**, **IP\_address**, Email ( USERS ), Search\_text )
* **RELATIONS** ( **Product1** ( PRODUCTS ), **Product2** ( PRODUCTS ) )
* **RESULTS** ( **Name** ( PRODUCTS ), { **SearchDT**, **IP\_address** } ( SEARCHES ) )
* **GARDENS** ( **Name** ( PRODUCTS ), Small\_count, Large\_count )
* **SEEDS** ( **Name** ( PRODUCTS ), Size, Season, Sun\_amount )
* **SEED\_TYPES** ( **Name** ( SEEDS ), **Type** )
* **PLANTEDS** ( **Garden** ( GARDENS ), **Seed** ( SEEDS ), Quantity )
* **DETAILS** ( **Name**, **Address**, Company, Phone# )
* **DETAILS\_OF** ( **Email** ( USERS ), { **Name**, **Address** } ( DETAILS ) )
* **ORDERS** ( **OrderID**, Email ( USERS ), { Name, Address } ( DETAILS ), OrderDate,  
   Shipping\_method, Payment\_type )
* **DESIGNS** ( **Name** ( GARDENS ), **DesignID**, OrderID ( ORDERS ), Quantity )
* **CHOSENS** ( { **Garden**, **Design** } ( DESIGNS ), **Seed** ( SEEDS ), Quantity )
* **INCLUSIONS** ( **OrderID** ( ORDERS ), **Name** ( PRODUCTS ), Quantity )

**הנחות והסברים נוספים:**

* **ביטוי הקשר Ordered by שבין USER ל-ORDER** – קשר יחיד לרבים. על פי חוקי המעבר למודל הטבלאי כאן בהכרח הקשר יתבטא כשדה נוסף בצד הרבים של הקשר (ORDER) שמגלם מפתח זר לצד היחיד של הקשר (USER).  
  לציין כי האתר מאפשר לבצע הזמנה גם ללא משתמש רשום. הדבר בא לידי ביטוי במפתח הזר של USER בטבלת ORDERS, בכך שאם מבצע ההזמנה אינו משתמש רשום, הערך בשדה זה יהיה ריק.
* **ביטוי הקשר Made by שבין USER ל-SEARCH** – קשר יחיד לרבים. גם כאן באופן זהה ומאותם סיבות הקשר יתבטא כשדה נוסף בצד הרבים של הקשר (SEARCH) שמגלם מפתח זר לצד היחיד של הקשר (USER).
* **ביטוי הקשר Addressed to שבין ORDER ל-DETAILS** – קשר יחיד לרבים. בעקבות חוקי המעבר למודל הטבלאי, הקשר יתבטא כזוג שדות נוספים בצד הרבים של הקשר (ORDER) שיחד יהוו מפתח זר לצד היחיד של הקשר (DETAILS).  
  חשוב לציין ששדה זה בהכרח לא יכיל ערך ריק בעקבות קרדינליות הקשר בצד היחיד.
* **ביטוי הקשר Details of שבין USER ל-DETAILS** – קשר רבים לרבים. על פי חוקי המעבר למודל הטבלאי כאן נדרש ליצור טבלה נפרדת עם שתי מפתחות זרים, אחד שמגלם בשדה יחיד את המפתח הראשי של USER והשני שמגלם בשתי שדות את המפתח הראשי של DETAILS.  
  שתי המפתחות הזרים הללו יחד יהוו מפתח ראשי לטבלה החדשה, בטבלה זו לא יהיו שדות נוספים.
* **ביטוי הקשר Resulted with שבין SEARCH ל-PRODUCT** – קשר רבים לרבים. באופן זההלקשר הקודם, ניצור טבלה נפרדת בה שתי מפתחות זרים, הראשון בעל שדה אחד שמגלם את המפתח הראשי של PRODUCT, והשני שמגלם באמצעות שני שדות את המפתח הראשי של SEARCH.  
  שני המפתחות הזרים הללו יחד יהוו מפתח ראשי לטבלה החדשה ולא יהיו שדות נוספים בטבלה זו.
* **ביטוי הקשר הרקורסיבי Related to של PRODUCT** – קשר רקורסיבי רבים לרבים. בדומה לקשרים הקודמים, ניצור טבלה חדשה בה יהיו הפעם רק שני שדות, כל אחד יהווה מפתח זר נפרד וביחד יהוו מפתח ראשי לטבלה חדשה. שני המפתחות יצביעו על שני מופעים שונים של ישות מוצר.
* **ביטוי הקשרים Included שבין ORDER ל-PRODUCT ו-Designed for שבין DESIGN ל-ORDER** – הקשר Included הוא קשר רבים לרבים. בדומה לקשרים קודמים, ניצור טבלה חדשה כדי לייצג את הקשר. בנוסף למפתחות הזרים אשר ביחד יהוו מפתח ראשי, יהיה שדה עבור התכונה quantity המצוינת במודל הERD-.  
  בקשר Included השדה הראשון יהווה מפתח זר לPRODUCT, והשדה השני יהווה מפתח זר לORDER.  
  הקשר Designed for הוא קשר יחיד לרבים. לכן מבוטא ע"י שדה Designed\_for בטבלת DESIGNS אשר מהווה מפתח זר לORDER.  
  חשוב לציין כי על אף שגינה בעיצוב אישי היא מוצר המוגדר כמו כל שאר המוצרים DESIGN אינו מוצר אלא שירות שניתן ללקוח בעת ביצוע הזמנה מסוימת הכוללת בתוכה גינה בעיצוב אישי, זאת לעומת כל מוצר אחר אשר לא מקבל התייחסות אישית מהלקוח בעת הזמנתו.  
  בכך לשירות העיצוב של גינה מסוימת קיימת ההצדקה של קשר כמעט זהה לקשר קיים אחר.
* **ביטוי הקשרים Planted שבין GARDEN ל-SEED ו-Chosen שבין DESIGN ל-SEED** – קשרי רבים לרבים. הטבלה של הקשר Planted תכיל שלושה שדות: הראשון מפתח זר לGARDEN, השני מפתח זר לSEED כך ששניהם יחד יהיו המפתח הראשי, ושדה שלישי עבור התכונה quantity המצוינת במודל ה-ERD.הטבלה של הקשר Chosen תכיל 4 שדות: השניים הראשונים יהוו מפתח זר לDESIGN, השלישי מפתח זר לSEED כך ששלושתם יחד יהיו המפתח הראשי ושדה רביעי עבור התכונה quantity המצוינת במודל ה-ERD.  
  בדומה לדמיון בין הקשרים Included וDesigned for גם קשר זה מקבל את ההצדקה מעצם היות עיצוב מסוים של גינה בעיצוב אישי שירות.
* **ביטוי מודל ההורשה שבין PRODUCT ל-SEED ול-GARDEN** – מודל הורשה בדיד וחלקי. על פי חוקי המעבר למודל הטבלאי קיימת הבחירה בין טבלה אחת לכלל הישויות או טבלה לכל אחת מהישויות במודל ההורשה, במקרה שלנו באמצעות שלוש טבלאות.  
  הדרך שנבחרה לביטוי מודל ההורשה הינה באמצעות טבלה לכל ישות, כדי להימנע מערכי NULL רבים שבהכרח יהיו במימוש באמצעות טבלה אחת. זאת מכיוון שערכי ה-NULL הרבים יתפסו מקום בזכירון ויעמדו בדרכנו בעת תחקור מסד הנתונים.

## פירוט שדות בסיס הנתונים המתוקן והערכת גודלם

| **Table** | **Field** | | **Data Type** | **Size (Bytes)** |
| --- | --- | --- | --- | --- |
| **PRODUCTS** | Name | | Varchar(80) | 80 |
| Price | | Smallmoney | 4 |
| Discount | | Smallmoney | 4 |
| **Total Record Size (Bytes)** | | | **88** |
| **USERS** | Email | | Varchar(40) | 40 |
| Password | | Varchar(30) | 30 |
| **Total Record Size (Bytes)** | | | **70** |
| **SEARCHES** | SearchDT | | Datetime | 8 |
| IP\_address | | Varchar(15) | 15 |
| Email **(USERS)** | | Varchar(40) | 40 |
| Search\_text | | Varchar(80) | 80 |
| **Total Record Size (Bytes)** | | | **143** |
| **RESULTS** | Name **(PRODUCTS)** | | Varchar(80) | 80 |
| SearchDT | **(SEARCHES)** | Datetime | 8 |
| IP\_address | Varchar(15) | 15 |
| **Total Record Size (Bytes)** | | | **103** |
| **SEEDS** | Name **(PRODUCTS)** | | Varchar(80) | 80 |
| Size | | Char(5) | 5 |
| Season | | Varchar(6) | 6 |
| Sun\_amount | | Varchar(30) | 30 |
| **Total Record Size (Bytes)** | | | **121** |
| **SEED\_TYPES** | Name **(SEEDS)** | | Varchar(80) | 80 |
| Type | | Varchar(20) | 20 |
| **Total Record Size (Bytes)** | | | **100** |
| **GARDENS** | Name **(PRODUCTS)** | | Varchar(80) | 80 |
| Small\_count | | Tinyint | 1 |
| Large\_count | | Tinyint | 1 |
| **Total Record Size (Bytes)** | | | **82** |
| **PLANTEDS** | Garden **(GARDENS)** | | Varchar(80) | 80 |
| Seed **(SEEDS)** | | Varchar(80) | 80 |
| Quantity | | Tinyint | 1 |
| **Total Record Size (Bytes)** | | | **161** |
| **DETAILS** | Name | | Varchar(40) | 40 |
| Address | | Varchar(150) | 150 |
| Company | | Varchar(40) | 40 |
| Phone# | | Varchar(30) | 30 |
| **Total Record Size (Bytes)** | | | **260** |
| **DETAILS\_OF** | Email **(USERS)** | | Varchar(40) | 40 |
| Name | **(DETAILS)** | Varchar(40) | 40 |
| Address | Varchar(150) | 150 |
| **Total Record Size (Bytes)** | | | **230** |
| **ORDERS** | OrderID | | Int | 4 |
| Email **(USERS)** | | Varchar(40) | 40 |
| Name | **(DETAILS)** | Varchar(40) | 40 |
| Address | Varchar(150) | 150 |
| OrderDate | | Date | 3 |
| Shipping\_method | | Varchar(100) | 100 |
| Payment\_type | | Varchar(7) | 7 |
| **Total Record Size (Bytes)** | | | **344** |
| **RELATIONS** | Product1 **(PRODUCTS)** | | Varchar(80) | 80 |
| Product2 **(PRODUCTS)** | | Varchar(80) | 80 |
| **Total Record Size (Bytes)** | | | **160** |
| **INCLUSIONS** | OrderID **(ORDERS)** | | Int | 4 |
| Name **(PRODUCTS)** | | Varchar(80) | 80 |
| Quantity | | Tinyint | 1 |
| **Total Record Size (Bytes)** | | | **85** |
| **DESIGNS** | Garden **(GARDENS)** | | Varchar(80) | 80 |
| DesignID | | Int | 4 |
| Designed\_for **(ORDERS)** | | Int | 4 |
| **Total Record Size (Bytes)** | | | **88** |
| **CHOSENS** | Garden | **(DESIGNS)** | Varchar(80) | 80 |
| Design | Int | 4 |
| Seed **(SEEDS)** | | Varchar(80) | 80 |
| Quantity | | Tinyint | 1 |
| **Total Record Size (Bytes)** | | |  |

**הנחות והסברים נוספים:**

* בשדה המתאר כתובת יהיו ערכים בהם יש לכל היותר 150 תווים, זאת משום שבכל ערך יצוינו כלל הפרטים של הכתובת ברצף, מדינה, מחוז, עיר, רחוב, מס' בית ומיקוד.
* בשדה Size יוכלו להיות הערכים "Small" או "Large" בלבד.
* בשדה Payment\_type יוכלו להיות הערכים "Klarna" או "ShopPay" בלבד, זאת כך ש"Klarna" מייצגת את שיטת התשלום דרך שירות התשלום החיצוני של Klarna שהאתר צורך, וש"ShopPay" מייצגת את שתי שיטות התשלום דרך שירותי התשלום החיצוני והחצי-חיצוני של Shopify.
* מחיר של מוצר מסוים, ובהתאם גם הנחה, יהיה לכל היותר 214,748.3647.
* כל השדות שמהווים כמות כלשהי (Quantity) יוכלו להכיל ערך של 255 לכל היותר.
* בשדה Phone# יוכלו להיות ערכים מהפורמט של 10 ספרות ברצף בלבד.
* בשדה IP\_address יוכלו להיות ערכים מהפורמט x.x.x.x כאשר כל x יכול להיות מספר שלם בתחום שבין 0 ל255.
* יהיו 2,247,483,647 הזמנות לכל היותר.

# פרק ראשון (מטלות חובה)

המטלות בפרק זה הן בגדר חובה, והציון הכולל עליהן יקבל משקל של 10% מציון הקורס.

# מטלת חובה מקדימה – תיקון ה-ERD והרחבת בסיס הנתונים

תיקון מודל ה-ERD וסכמת הטבלאות, ואכלוס הטבלאות בנתונים בהיקף משמעותי. **ביצוע בלתי ראוי של מטלה מקדימה זו עלול להוביל לקנס של עד 20% לציון חלק ג' פרק ראשון.**

# מטלה 1 – שאילתות (70% מציון הפרק הראשון)

המחשת שאילתות מסוגים שונים.

# מטלה 2 – כלים להצגת נתונים (30% מציון הפרק הראשון)

מימוש כלים להצגת נתונים המיועדים לגורמים שונים בארגון ונותנים מענה לשאלות עסקיות.

# פרק שני (רשות)

המטלות בפרק השני הן בגדר רשות, והציון הכולל עליהן יהיה מהווה מגן מול ציון המבחן המסכם (כלומר, הציון הגבוה מבין השניים יילקח בחשבון), במשקל של 5% מציון הקורס.

# מטלה 3 – יישומים מתקדמים (צבירה עד משקל של 100%)

מימוש כלים מתקדמים, הממחישים תועלות שונות שניתן להפיק מבסיס הנתונים ומהכלים המתלווים אליו.

### בונוס תחרותי על יצירתיות ומקוריות (עד 10%)

צוותים שיפגינו חשיבה יצירתית ומקורית בפתרונותיהם יזכו לבונוס של 10 נק' לציון חלק ג'.

* מדובר בבונוס תחרותי, שיוכלו לזכות בו **לכל היותר 10% מהצוותים.**
* הבונוס יוענק על פי שיקול הדעת של צוות הבודקים, **ללא אפשרות ערעור.**
* כיוון שהשיקול העיקרי להענקת הבונוס הוא מידת היצירתיות והמקוריות - **לא בהכרח יזכו בו היישומים שיקבלו את הציון הגבוה ביותר** על יישום מרכיבי חלק ג'.

# בהצלחה!

## פרק ראשון – מטלות חובה (10% מציון הקורס)

המטלות בפרק זה הן בגדר חובה, והציון הכולל עליהן יקבל משקל של 10% מציון הקורס.

## מטלת חובה מקדימה – תיקון ה-ERD והעשרת בסיס נתונים

## (קנס של עד 20% לציון הפרק הראשון על ביצוע לא ראוי)

כמטלה מקדימה, יש לתקן את מודל ה-ERD ואת טבלאות בסיס הנתונים על פי ההערות שניתנו על הגשת חלק ב' של הפרויקט.

* עברו על הערות הבדיקה שניתנו לחלקים א' ו-ב', ותקנו בהתאם את ה-ERD ואת מודל הטבלאות – **הדרישה היא שתסתמכו על המודל המתוקן ליישום המטלות בהמשך.**
* יש להביא למפגש ההגנה תדפיס של תרשים ה-ERD ומבנה הטבלאות שהוגשו בחלק ב', עם ההערות שניתנו, וכן תדפיס של תרשים ה-ERD ומבנה הטבלאות המתוקן עליהם הסתמכתם בחלק ג'. במפגש, תידרשו להסביר את הטעויות בהגשה הקודמת, וכיצד הן תוקנו.

לאחר תיקון המודל הטבלאי, יש להזין לכל טבלה נתוני המחשה בהיקף מתאים סימולציה – לפחות כמה מאות רשומות בכל טבלה מרכזית.

* **להכנת נתוני סימולציה, מומלץ להיעזר ביכולות השונות הזמינות באינטרנט וב-EXCEL. לצורך כך, צירפנו במודל קובץ המלצות לחילול נתונים באמצעות תוכנת EXCEL – חשוב לעיין בקובץ זה, לפני שמתחילים בהכנת בסיס הנתונים המועשר.**
* הציפייה היא לסדר גודל של עשרות עד מאות רשומות בטבלאות המכילות נתוני ישות מרכזית (לקוח, מוצר, שירות, וכד'), ומאות רשומות בטבלאות המכילות נתוני קשרים או פעולות חזרתיות (פעולות חיפוש, הזמנות, משלוחים, חוות דעת, וכד').
* חשוב להכין נתונים שיש בהם הגיון עסקי משכנע - לא סביר למשל שבאתר המציע מאות מוצרים, יוזמנו בפועל רק שניים. כמו כן יש לשים לב לקשרים בין הנתונים השונים, ולתכנן בהתאמה את ערכי המפתחות ראשיים והזרים.

אי תיקון טעויות, תיקון בלתי מספק, ו/או אי הזנת נתונים בכמות ראויה, עלולים להוביל להורדת נקודות משמעותית. במקרים קיצוניים – ביצוע בלתי הולם של מטלה מקדימה זו עלול להוביל לפסילת חלק ג' ולדרישה לביצועו מחדש.

## מטלה 1 (%70 מציון הפרק הראשון) – שאילתות

על סמך הטבלאות והנתונים שיישמתם, עליכם להגדיר מספר שאילתות כמפורט להלן:

* לכל שאילתה – יש להציג את קוד ה-SQL, ולהמחיש את הפעלתו.
* להגשת המטלה יש לצרף הסבר קצר (משפט או שניים) להיגיון העסקי מאחורי כל שאילתה – על איזו שאלה היא עונה? עבור איזה תרחיש עסקי שאלה כגון זו עשויה להיות רלוונטית ?

# שתי שאילתות SELECT ללא קינון (8%)

עליכם ליישם ולהציג שתי שאילתות SELECT שאינן משתמשות בקינון.

* על כל שאילתה לבצע JOIN מוצדק ונכון לפחות בין שלוש טבלאות שונות
* על כל שאילתה להדגים את המרכיבים –SELECT, FROM, WHERE, ORDER BY
* לפחות אחת השאילתות צריכה להדגים גם מרכיבי GROUP BY ו- HAVING
* לפחות אחת השאילתות צריכה להדגים שימוש נכון בפונקציות (הכוונה לפונקציות מערכת קיימות, לא לפונקציות שתכתבו לבד בהמשך)

# שתי שאילתות SELECT מקוננות (15%)

עליכם ליישם שתי שאילתות מקוננות – כאשר הקינון יתבצע על ידי שימוש בשאילת SELECT בתוך שאילתת SELECT אחרת.

* כל שאילתה צריכה להדגים סגנון קינון שונה, מבין אלו שנלמדו בקורס – קינון שמחזיר ערך יחיד (Scalar), רשימת ערכים (List), או טבלה (Table). כמו כן – יש למקום כל קינון במרכיב שונה של השאילתה (ב-SELECT, ב-FROM, ב-WHERE, או ב- HAVING – בהתאם לסוג הקינון).
* הקינון צריך להתבצע בתוך השאילתה עצמה. בסעיף זה אין להשתמש ב-WITH, ב- VIEW או בפונקציה שנוספה על ידי חברי הצוות (ניתן להשתמש בפונקציות המערכת הקיימות בשרת ה-SQL).
* בשתי השאילתות חייבת להיות הצדקה לשימוש בקינון. אם ניתן באופן ברור ומובהק לממש את השאילתה ללא שימוש בקינון – הפתרון ייחשב כשגוי.

# שתי שאילתות מקוננות תוך שימוש במרכיבים נוספים (15%)

* **שאילתה 1:** יצירת שאילתה אחת של Update או Delete מקוננת.
* **שאילתה 2:** יצירת איחוד/חיתוך/הפרש בין 2 שאילתות, כך שלפחות אחת מהן מקוננת.
  + חייבת להיות הצדקה לשימוש בקינון. אם ניתן באופן ברור ומובהק לממש את השאילתה ללא שימוש בקינון – הפתרון ייחשב כשגוי.
  + חייבת להיות הצדקה לשימוש ב-איחוד/חיתוך/הפרש. אם ניתן להגיע לתוצאה דומה ע"י תנאי WHERE בלבד, ללא איחוד/חיתוך/הפרש - הפתרון ייחשב כשגוי.

# שאילתות עסקיות המשלבות Window Functions (%16)

פונקציות "חלון" (Window Functions) הוא אוסף פונקציות ייחודי המאפשר פעולות אגרגרציה על סמך אוסף רשומות המחזיר כפלט אוסף רשומות (בניגוד לפעולות אגרגציה סטנדרטיות, המחזירות כפלט מספר יחיד).

* למשל, פונקציות כגון: Lead (), Lag (), Rank (), Percent\_Rank ()
* לקריאה נוספת אודות פונקציות אלו: <https://www.sqlservertutorial.net/sql-server-window-functions>
* השימוש בפונקציות הוא על בסיסי לימוד עצמי. לא תינתן עליהן הדרכה על ידי צוות הקורס.

במטלה זו, יש להמחיש שימוש **בלפחות ארבע פונקציות "חלון" שונות**. ההמחשה תעשה על ידי שילוב הפונקציות **בלפחות שתי שאילתות שונות** (כלומר, ניתן לשלב שתיים או יותר פונקציות באותה שאילתה).

* על השאילתות לענות על צורך עסקי ממשי ומובהק לניתוח נתונים שעושה שימוש בפונקציות ה"חלון", ולהדגים רמת מורכבות גבוהה – למשל, אגרגציה לא שגרתית של נתונים, על סמך ערכי הפלט של פונקציות ה"חלון".
* אם ניתן לבצע את הפעולה הממומשת בטכניקות אחרות פשוטות יותר (למשל, שאילתה מקוננת או פונקציה פשוטה) – הכלי לא יחשב כעונה על דרישת המטלה למימוש כלי מורכב.
* יש לצרף צילומי מסך של פלטי השאילתות בהם השתמשתם.

# דו"ח המושתת על שימוש בפסקת WITH מורכבת (%16)

טכניקת WITH נועדה להקל כתיבה של שאילתות מקוננות, על ידי הגדרת תתי השאילתות שאליהן ניתן להפנות במספר מקומות בתוך השאילתה הראשית. יש להדגים יצירת דו"ח עסקי **(באמצעות קוד SQL)**, אשר עושה כולו שימוש בשאילתות הנכתבות בתוך בלוק WITH, ומדגים שימוש בטכניקות שאילתה מעניינות ומורכבות. על המימוש בשאילתה הראשית לשלב מספר טכניקות קינון שונות.

* על בלוק ה-WITH להכיל **לפחות** 4 תתי שאילתות.
* **לפחות** אחת מתתי השאילתות בבלוק נדרשת לשמש ביצירת 2 עמודות שונות בדו"ח העסקי.
* **לפחות** אחת מתתי השאילתות בבלוק נדרשת לעשות שימוש בתת-שאילתה אחרת מבלוק ה-WITH.
* על הדו"ח להיות בעל **הגיון ורציונל עסקי מובהק**, ולהפגין רמת מורכבות גבוהה. בעת הצגת הדו"ח, הצוות יידרש להבין ולהסביר את התפקיד של כל אחד מהמרכיבים בדו"ח הסופי.
* דו"ח בעל רמת מורכבות נמוכה מדי יקבל ציון חלקי בלבד. אם אינכם בטוחים שרמת המורכבות של הדו"ח מספיקה - יש להתייעץ עם צוות המתרגלים.
* עם זאת, יש להבהיר שהמורכבות אין משמעותה הוספת מרכיבים מיותרים. למשל – לא יתקבל פתרון שמבוסס על בניית תת-שאילתה או פונקציה נפרדת לכל עמודה, שלא לצורך. **יש לחשוב ולתכנן מראש כיצד להפיק את הדו"ח בצורה מיטבית – קינון מיותר והוספת תתי-שאילתות שלא לצורך עלולה לפגום בציון.**
* לצורך המחשה, בנספח בהמשך מובאת דוגמא לשאילתה היוצרות דו"ח עסקי באמצעות קינון מורכב של מספר שאילתות ביניים, שניתנה בעבר במבחן. יודגש כי מדובר בדוגמא להמחשה – הציפייה היא שצוות המתמודד עם מטלה זו **יציג פתרון שונה מהדוגמא,** המתאים לצרכים העסקיים של מנהלי האתר אותו הוא מנתח.
* יש לצרף צילומי מסך של פלטי בלוק ה-WITH.

## מטלה 2 (%30 מציון הפרק הראשון) – כלים להצגת נתונים

חלק זה ימחיש שימוש בנתונים למימוש כלים להצגת נתונים, העונים על שאלות עסקיות חשובות, ו/או תומכים בהחלטות הקשורות לתהליכים העסקיים בהם תומך בסיס הנתונים. על כל כלי לשקף אספקט עסקי שונה, ולשרת קהל משתמשים שונה.

**דו"ח עסקי (%15) -** עליכם להגדיר במפורש למי מוצג הדו"ח (למשל: לקוחות/מחלקת שיווק/סמנכ"ל תפעול וכו') ובהתאם לכך להשתמש באמצעים להצגת נתונים (תרשימים, או הצגה טבלאית של נתונים) המותאמים לקהל המשתמשים המיועד.

**לוח מחוונים (%15) -** לוח מחוונים דיגיטלי (Digital Dashboard) הוא צורת יישום מקובלת במערכות בינה עסקית (Business Intelligence), המיועדת בדרך כלל למנהלים בכירים בחברה. כנקודת מוצא ליישום זה, מומלץ לקרוא תחילה על המונח (ניתן למצוא בחיפוש Google מקורות נוספים העוסקים בנושא):

* <https://en.wikipedia.org/wiki/Dashboard_(business)>
* <https://www.klipfolio.com/resources/articles/what-is-digital-dashboard#gref>
* יישום כלים להצגת נתונים מצריך חשיבה ותכנון מקדים. בכל כלי חשוב לנסח שאלות עסקיות רלוונטיות, תוך התייחסות לקהל היעד שאליו הכלי מיועד. כמו כן יש לתת את הדעת לנתונים שכדאי להציג, ולצורה המועדפת להצגתם (למשל, שיקולים של הצגה טקסטואלית לעומת גרפית, סוגי הגרפים המתאימים, פילטרים, וכד').
* בעת מפגש ההגנה – תדרשו להדגים את השימוש בכל כלי, ולהסביר מה תפקידו, איזו מטרה עסקית הוא משרת, ובאילו החלטות הוא נועד לתמוך.

## הנחיות נוספות ליישום הכלים להצגת הנתונים:

* יש להוסיף לבסיס הנתונים VIEWS ו/או טבלאות סיכום שיסייעו בהפקת הכלים. במפגש ההגנה תידרשו להראות את ה-VIEWS ואת טבלאות הסיכום שמימשתם
* בכדי לקבל את מלוא הנקודות על סעיף זה על שני הכלים להיות מעניינים ורלוונטיים, מעוצבים בצורה מקצועית ונאה לעין, ולכלול מספר מרכיבי תצוגה שונים (גרפים, טבלאות, וכד') אשר יהיו **מקושרים** ביניהם. כמו כן, **על** **כל כלי לאפשר שימוש בפילטרים מגוונים**  (למשל – בחירת טווח תאריכים, קטגוריית מוצרים, מדינה, טווח מחירים ועוד), בהתאם למידע הנדרש.
* יש לוודא שהכלי מציג את המידע הנדרש בצורה מיטבית ונוחה להבנה ופרשנות. כלים שאינם מציגים את המידע בצורה מיטבית, יזכו לניקוד חלקי בלבד.
  + מצד אחד, יש לוודא שהכלים אינם דלים מדי מבחינת מספר מרכיבי התצוגה - לדוגמא, כאלו המציגים רק גרף אחד וטבלה אחת, /ואו פילטר יחיד.
  + מצד שני, גם עומס ויזואלי מוגזם אינו רצוי. שימוש במספר רב מדי של מרכיבי תצוגה עלול להפוך את המידע לבלתי מובן. יש לוודא שכל פריטי המידע המוצגים בכלים קריאים וברורים. פריט תצוגה עמוס ובלתי שימושי, עלול לגרור הורדת ניקוד.
* הכלי המומלץ למימוש חלק זה הוא ה-PowerBI של Microsoft. כלי זה מאפשר יישום יעיל ונוח של הדו"חות, וכן של מוצרי המידע הנוספים הנדרשים במטלות בהמשך. לכלי זה תינתן הדרכה בסיסית בלבד – כדאי להשקיע בלימוד עצמי של הכלי, כיוון שמדובר בכלי נפוץ ופופולארי, העשוי לסייע למטלות רבות בהמשך התואר ובפרויקט הגמר. בנוסף לחומר העזר שיפורסם, ניתן למצוא שפע של סרטוני הדרכה ב- YouTube ובאתרים אחרים.
  + ניתן להשתמש גם בכלי BI אחר (למשל QlikSense או Tableau), **על בסיס לימוד עצמי.** לפני בחירת כלי חלופי, **יש להתייעץ עם צוות הקורס ולקבל את אישורו.** צוות הבוחר בכלי חלופי, יידרש ליישם חלק זה של הפרויקט על מחשב נייד ולהביאו למפגש ההגנה.
* לוח המחוונים מציג נתונים בעזרת שילוב מרכיבי תצוגה טקסטואליים וויזואליים. קיימים עם זאת הבדלים חשובים, שיש להביאם לידי ביטוי ביישום:
  + לוח המחוונים משמש לקבלת תמונת מצב כוללת על הארגון ופעילותו. המנהלים הבכירים של החברה רוצים לראות היכן הם עומדים בהתאם ליעדים העסקיים שהם הציבו לעצמם ממגוון תחומים (כמו הגדלת קהל הלקוחות בX אחוזים, הגדלת הרווחים בX אחוזים, שימור לקוחות קבועים ומניעת נטישת לקוחות, הגדלת המחירות ממותג מסויים, עליית בריווחיות מהמבצים של החברה), לכן נצפה לראות הרבה שעונים שמראים את המצב הנוכחי ביחס ליעד שנקבע. לעומת זאת, דו"ח עוסק לרוב באלמנט עסקי ספציפי ועונה על שאלה עסקית ממוקדת.
  + Drill Down - במסך הראשי של לוח מחוונים מקובל להציג את הנתונים בצורה סיכומית וברמת גרעיניות נמוכה – אך לאפשר "קידוח" לחתכי נתונים פרטניים ותחקור של נתונים ספציפיים. לדוגמא – נניח שלוח המחוונים מראה גרף של היקף מכירות לפי מדינות. לחיצת עכבר על הגרף על מדינה מסוימת יוביל **למסך נוסף** ובו פילוח מכירות על פי ערים במדינה שנבחרה (רמת גרעיניות גדולה יותר), מה שיאפשר למשתמש להתמקד בחתך נתונים ספציפי יותר.
  + על לוח המחוונים להיות מעוצב בצורה מקצועית ונאה לעיין, תוך הבלטת פריטי המידע החשובים, וללא עומס ויזואלי מוגזם. אתרים וסרטוני הדרכה רבים מציעים הנחיות, המלצות ו"טיפים" לעיצוב מוצלח של לוח מחוונים, למשל:
* [www.cpoc.org/assets/Data/guide\_to\_dashboard\_design1.pdf](file:///C:\Users\Danna\Downloads\www.cpoc.org\assets\Data\guide_to_dashboard_design1.pdf)
* [www.klipfolio.com/blog/10-tips-for-better-dashboards](file:///C:\Users\Danna\Downloads\www.klipfolio.com\blog\10-tips-for-better-dashboards)
* <https://www.youtube.com/watch?v=V-QzuuraqV0>
* <https://www.youtube.com/watch?v=XEPJPidH4W0>
* <https://www.youtube.com/watch?v=aZKNkvi9ybI>
  + **דגש**: מצופה מכם ללמוד באופן עצמאי על מהותו של לוח המחוונים וליישמו בהתאם. לשם כך צירפנו קישורים, אך מומלץ לחקור ולצפות בתוצרים מעבר למצורף.
  + **יש לצרף צילומי מסך בדו"ח של המסכים הראשיים ומסכי הDrill Down שיצרתם**.

## פרק שני - מטלות רשות (5% מציון הקורס, מגן מול ציון המבחן)

המטלות בפרק השני הן בגדר רשות, והציון הכולל עליהן יהיה מהווה מגן מול ציון המבחן המסכם (כלומר, הציון הגבוה מבין השניים יילקח בחשבון), במשקל של 5% מציון הקורס.

## מטלה 3 – יישומי כלים מתקדמים

בחלק זה עליכם לממש כלים להעשרת בסיס הנתונים. הדגמה נאותה והצדקת השימוש בכל כלי חשובה לא פחות מהמימוש. **ניקוד מלא יינתן רק לכלי הפועל באופן מלא ומושלם, ובליווי הדגמה הולמת בעת מפגש ההגנה.**

* **אין לממש פונקציונאליות זהה בעזרת כלים שונים.** למשל, View ו- Function שמחזירים את אותה תוצאה, או Trigger ו- Stored Procedure שמבצעים פעולה דומה, וכד'.

# View (15%)

עליכם לממש View שעונה על צורך עסקי מסוים – למשל: שאילתה מורכבת שמבוצעת לעיתים קרובות, הסתרת נתונים ממשתמשים לא-מורשים וכדומה. בנוסף למימוש יהיה עליכם להצדיק את ה- View ולהמחיש את הפעלתו.

* יש לצרף צילום מסך של הדגמת הView -.
* **בעת מפגש ההגנה**- יש להדגים שימוש בטבלת ה- View שיצרתם, היערכו מראש עם שאילתה המדגימה שימוש מוצדק ב-View ( הימנעו מהצגת טבלת הView- כפי שהיא select \* From View).

# פונקציות (Functions) (30%)

יש לממש שתי פונקציות (15%, כ"א) - אחת המחזירה ערך יחיד (Scalar), ושנייה המחזירה טבלה.

* לכל פונקציה יש לכתוב שאילתת SELECTמתאימה המדגימה כראוי את הפעלתה.
* צריכה להיות הצדקה עניינית להוספת הפונקציות. שימו לב שאינכם כותבים פונקציות שמתאים יותר לממש אותן באמצעות פרוצדורה פשוטה.
* שימו לב כי שתי הפונקציות צריכות לבצע פעולות שונות במהותן. לא יתקבלו פתרונות המממשים שתי פונקציות שמשתמשות באותן הטבלאות ו/או מבצעות פעולת אגרגרציה דומה.
* יש לצרף צילום מסך של הפעלת הפונקציות.
* **בעת מפגש ההגנה**- יש להדגים את פעולת הפונקציות, היערכו מראש עם השאילתות להפעלות הפונקציות.

# Trigger פשוט (15%)

יש לממש Trigger פשוט - כזה המופעל בעקבות הוספת רשומות או עדכון נתונים בטבלה מסוימת ומבצע פעולת עדכון עוקבת בהתאם.

* על הטריגר להשתמש ביותר מטבלה אחת באמצעות שאילתה לא בסיסית, ולבצע עדכון על שדה אחד לפחות - באותה טבלה או בטבלה אחרת.
* דרך מומלצת להמחשת Trigger היא ליצור שדה מחושב באחת הטבלאות, ולהגדיר Triggers מתאימים כך שהשדה המחושב יעודכן כראוי. יש לקחת בחשבון גם מצבים בהם מעדכנים מספר רשומות במקביל, ו/או נתונים המשפיעים על הערך באופן עקיף.
* יש לצרף צילום מסך של הדגמת ה-Trigger.
* **בעת מפגש ההגנה**- יש להדגים את פעולת ה-Trigger ולהצדיק מילולית את נחיצותו. למשל, הציגו טבלה מסוימת לפני הפעלת הטריגר, בצעו את הפעולה שמפעילה את הטריגר והציגו כיצד הטבלה השתנתה.

# פרוצדורה שמורה ((Stored Procedure פשוטה (15%)

יש לממש פרוצדורה שמורה פשוטה, המקבלת פרמטרים ומבצעת בהתאם פעולה פשוטה כלשהי בטבלה מסוימת.

* חייבת להיות הצדקה עניינית ליישום הפרוצדורה, כלומר חייבת להיות סיבה לבצע את הפעולות באמצעות הפרוצדורה ולא על ידי כתיבת פונקציה או ישירות על ידי כתיבת קוד SQL. למשל, ניתן להצדיק פרוצדורה אם מדובר בפעולות מבוקרות שנעשות לרוב על ידי משתמשים שאינם מורשים לגשת ישירות לטבלה מסוימת.
* על הפרוצדורה לקבל כ- Input לפחות פרמטר אחד.
* יש לצרף צילום מסך של הדגמת הפרוצדורה.
* **בעת מפגש ההגנה-** יש להדגים את הפרוצדורה, היערכו מראש עם שורת ביצוע (Execute) המפעילה אותה. יש להמחיש גם את תוצאות הפעלת הפרוצדורה – למשל, על ידי הצגת הנתונים הרלוונטיים לפני ואחרי ההפעלה.

## שילוב מערכתי של מספר כלים (%25)

יש להדגים יישום מערכתי הדורש שילוב של מספר כלים שונים (שלושה לפחות מבין: פרוצדורה שמורה, פונקציה, טריגר, View), לצורך מתן מענה על פונקציה עסקית או תפעולית חשובה. על שלושת הכלים הללו לעבוד יחדיו (כלומר הכלי הראשון מפעיל את השני והשלישי, וכד') ולא רק להיות קשורים אחד לשני מבחינה רעיונית.

* לדוגמא, SP שמעדכנת טבלה באמצעות פונקציה, והעדכון גורם להפעלת Trigger.
* יש לדעת להסביר את תפקיד כל אחד מהמרכיבים ביישום הכולל, ולהדגים את היישום כך שהתפקיד של כל אחד מהמרכיבים יומחש. על כל אחד מהכלים להיות מוצדק באופן עצמאי, ללא קשר לשימוש בו בכלי המורכב.
* אם ניתן לבצע את השילוב הממומש באמצעות טכניקות חלופיות פשוטות יחסית – הפתרון לא ייחשב כעונה על הדרישה. אם אתם לא בטוחים לגבי מידת ההתאמה של השילוב המערכתי שאתם מתכוונים לממש – מומלץ להתייעץ עם צוות הקורס.
* יש לצרף צילומי מסך של הדגמת כל אחד מהשלבים.
* **בעת מפגש ההגנה**- היערכו מראש עם שאילתות להפעלת הכלים, יש להדגים את התהליך בצורה תהליכית (כיצד פעולה אחת מפעילה פעולה אחרת).

## נספח: דוגמא לשאילתת דו"ח, מסתמכת על שאלה שנכללה בעבר במבחן

חברה מפתחת יישום ניווט המאפשר למשתמשים (USERS) להגדיר פעולות חיפוש (SEARCHES), המובלות להמלצות על מסלולים (PATHS). הפעילות מתועדת בבסיס הנתונים הטבלאי הבא:

* **ROUTES (Route, Length, Duration)**
* **מקטעי דרך:** שם המקטע, אורך (בק"מ), משך זמן הנסיעה המשוערך (בשניות)
* **USERS (User, JoinDate, Fee)**
* **משתמשים:**  מזהה המשתמש, תאריך ההצטרפות לאתר, תשלום לכל פעולת חיפוש (בש"ח)
* **SEARCHES (User (USERS), SearchDT, ArrivalDT)**
* **פעולות חיפוש:** מזהה משתמש, מועד (תאריך/זמן) החיפוש, מועד ההגעה בפועל
* **PATHS (} User, SearchDT{ (SEARCHES), Number, Chosen)**
* **מסלולים:** מזהה משתמש, מועד חיפוש, סדר הצגה (1, 2, 3, ...), האם נבחר (1- כן, 0 – לא)
* **INCLUDES (}User, SearchDT, Number} (PATHS), Order ,Route (ROUTES))**
* **סדר המסלול:** שם משתמש, מועד חיפוש, סדר הצגה, המספר הסידורי של המקטע במסלול (1, 2, 3, ...) , שם מקטע

**לניתוח מאפייני נטישת לקוחות, התבקשת להכין דו"ח במבנה הבא (הרשומות להמחשה):**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Estimated Loss** | **Total**  **Late** | **Total**  **Dismissed** | **Search**  **Activities** | **Days Active** | **Last Activity** | **User** |
| 9000 | 10 | 150 | 500 | 20 | 1.8.2020 | Sarit Hadad |
| 8000 | 20 | 60 | 300 | 10 | 6.7.2019 | Eyal Golan |
| 5000 | 30 | 40 | 100 | 15 | 4.7.2019 | Ben Static |
| 3000 | 5 | 80 | 200 | 12 | 9.9.2018 | Omer Adam |
| **…** | **…** |  | **…** | **…** | **…** | **…** |

**הסבר השדות:**

* **User, Last Activity, Days Active –** שם המשתמש, התאריך בו ערך את החיפוש האחרון (המאוחר ביותר), מספר הימים בהם היה פעיל (ההפרש בימים בין תאריכי ההצטרפות והנטישה)
* יש להציג את כל המשתמשים שלא היו פעילים במהלך שלושת החודשים שלפני התאריך בו הופק הדו"ח
* **Search Activities, Total-Dismissed, Total-Late -** מס' פעולות החיפוש הכולל באתר, מס' פעולות החיפוש בהן לא בחר בסופו של דבר אף מסלול, מס' פעולות החיפוש בהם משך הנסיעה בפועל היה ארוך מהמשוערך
* **משך נסיעה בפועל:** ההפרש (בשניות) בין מועד החיפוש לבין מועד ההגעה בפועל
* **משך נסיעה משוערך:** על סמך שיערוך משך הזמן הכולל (בשניות) של המקטעים המרכיבים את המסלול שנבחר, (ניתן להניח שנבחר רק מסלול אחד)
* **Estimated Loss-** הערכת הפסד ההכנסה בש"ח כתוצאה מנטישת המשתמש
* **שיערוך ההפסד:** כמה כסף החברה הייתה גובה מהלקוח, לו היה ממשיך להשתמש ביישום באותה תדירות חיפוש יומית ממוצעת מאז שנטש (תאריך הפעילות האחרונה) ועד התאריך בו הופק הדו"ח.
* **כמומחש:** יש לסדר את הרשומות **בסדר יורד** של הנזק המשוערך

**פתרון**

בלוק ה-WITH הבא מייצר בסופו תת-שאילתה בשם SEARCH\_DETAILS המציגה לכל חיפוש, מידע על בחירה (1- נבחר מסלול, 0 – לא) ואיחור (1 – היה איחור, 0 - לא), על ידי שימוש בתתי שאילתות קודמות . תת-השאילתה מציגה גם מידע נוסף על מספר חודשים עברו בין תאריך החיפוש לבין תאריך הפקת הדו"ח:

WITH

FIRST AS (SELECT User, SearchDT, Dismissed = 1 – MAX(Chosen),

Months = DATEDIFF (mm, SearchDT, GETDATE())

FROM PATHS

GROUP BY User, SearchDT),

DURATION AS (SELECT P.User, P.SearchDT, Duration = SUM (R.Duration)

FROM PATHS as P JOIN INCLUDES AS I

ON P.User = I.User AND P.SearchDT = I.SearchDT

AND P.Number = I.Number

JOIN ROUTES as R ON I.Route = R.Route

WHERE P.Chosen = 1

GROUP BY P.User, P.SearchDT),

LATENESS AS (SELECT S.User, S.SearchDT,

Late = CASE WHEN DATEDIFF (ss, S.SearchDT, S.ArrivalDT ) >

D. Duration THEN 1 ELSE 0 END

FROM SEARCHES as S JOIN DURATION as D ON S.User = D.User AND

S.SearchDT = D.SearchDT),

SEARCH\_DETAILS AS (SELECT F.User, F.SearchDT, F.Dismissed, L.Late, F.Months

FROM FIRST as F JOIN LATENESS as L)

השאילתה הבאה מייצרת את הדו"ח המורכב תוך שימוש בבלוק:

SELECT User = U.User,

Last\_Activity = MAX (SD.SearchDT),

Days\_Active = MAX ( DATEDIFF (dd, U.JoinDate, SD.SearchDT ),

Search\_Activities = COUNT (\*),

Total\_Dismissed = SUM ( SD.Dismissed ),

Total\_Late = SUM ( SD.Late ),

Estimated\_Loss = U.Fee \* DATEDIFF (dd, Last\_Activity, GETDATE()) \*

(Search\_Activities / Days\_Active)

FROM USERS AS U JOIN SEARCH\_DETAILS AS SD ON U.User = SD.User

GROUP BY U.User, U.JoinDate, U.Fee

HAVING MIN (SD.Months ) >= 3

ORDER BY Estimated\_Loss DESC