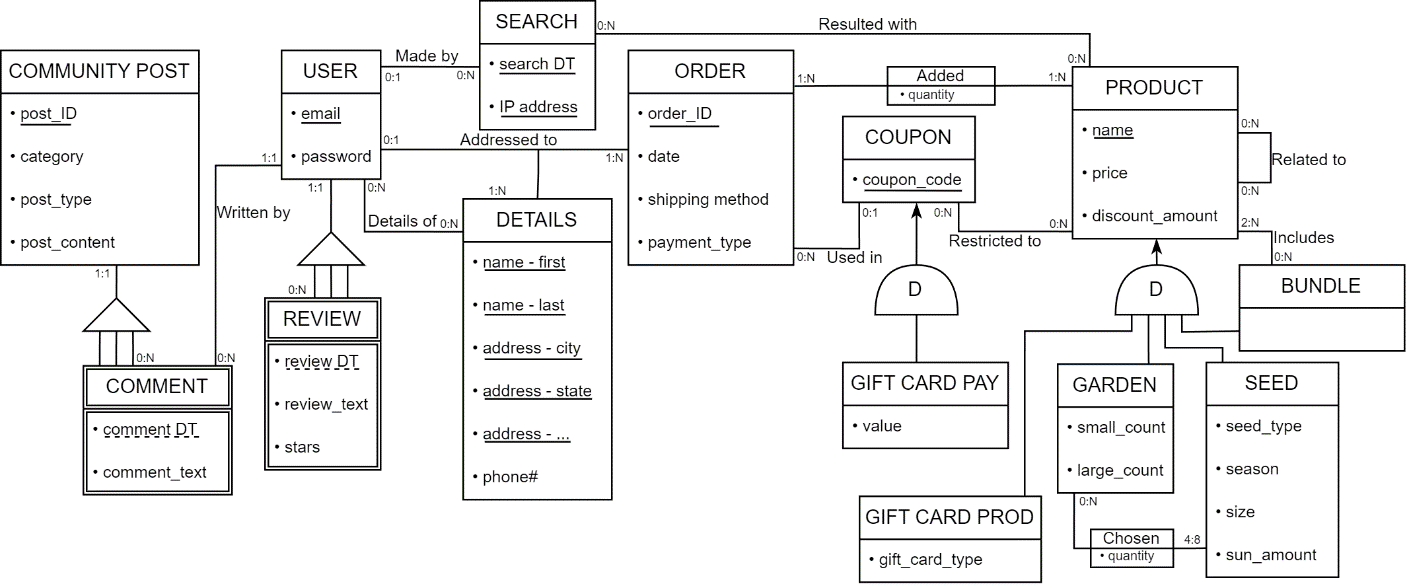
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מס' קבוצה | אתר | | | תאריך הגשה |
| 16 | [seedsheets.com](http://seedsheets.com) | | | 19/02/2024 |
| חברי הצוות – מספרי ת.ז | | | | |
| 318334901 | | 211680509 | 318255882 | |

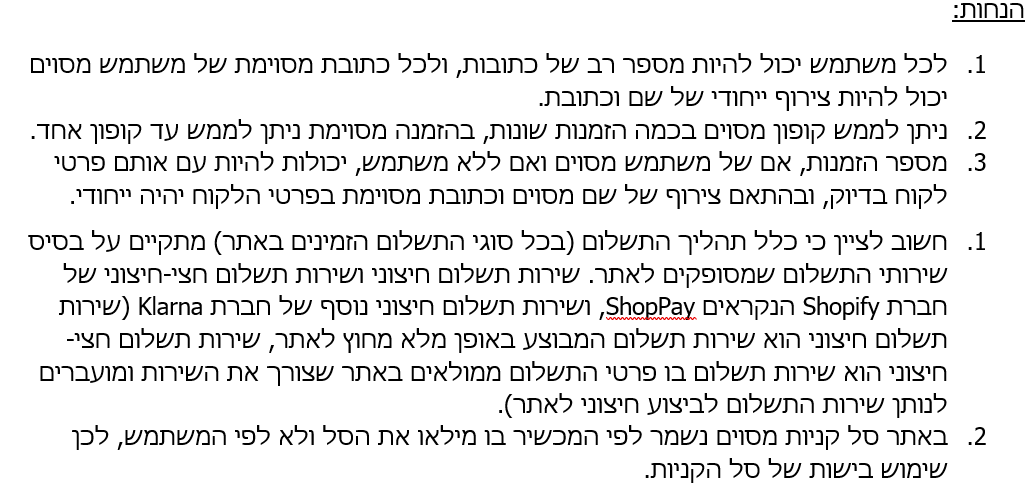
## פרויקט בסיסי נתונים

## מטלה מקדימה – ה-ERD שהוגש בחלק א'

**התרשים בהגשה הקודמת**



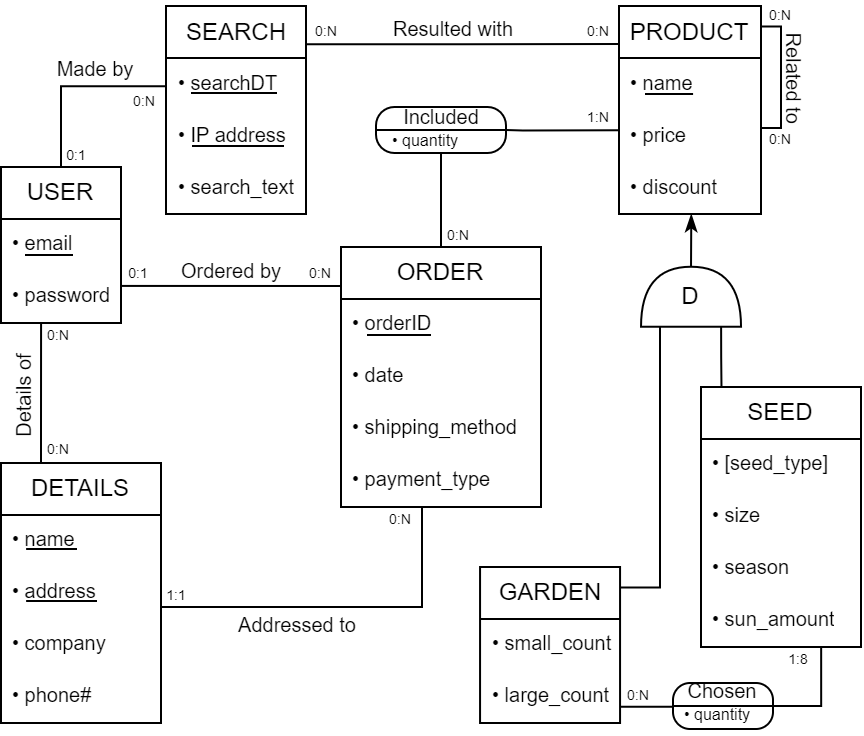
**ההנחות שנרשמו בהגשה הקודמת**



**ההערות שניתנו להגשה הקודמת**

1. התבקשתם ליצור מודל פשוט ככל הניתן המכיל בין 5-8 ישויות ומתייחס לתהליך העסקי המרכזי שלכם. יש לכם 14 ישויות. ישויות ללא הצדקה.
2. סימונים לא נכונים של קשרים- תכונה של קשר יסומן בתאך צורה אובלית ולא בתוך מלבן. מלבן מייצג ישות בלבד.
3. איך מתחשבת ההנחה? היא קבועה? אם לא, זה שדה מחושב והוא לא ייכנס לERD
4. איך נקבע מחיר של גינה מעוצבת? האם היא קבועה או תלויה בעיצוב?  
   אם תלויה בעיצוב, שדה זה יהיה מחושב ולא נכניס אותו לERD. נסו להבין מה באמת קובע את המחיר.
5. מה משמעות הקשר הרקורסיבי?
6. האם באתר אפשר לקנות רק SEED? יש לו מחיר משל עצמו? הוא ממש מוצר בפני עצמו או שהוא רק חלק מגינה מעוצבת?
7. מה המשמעות של השדות של גינה? איזה ערכים הם מקבלים? התכוונתם לכמות הplants? יש להשתמש בשמות אינפורמטיבים. בנוסף, אם באמת התכוונתם לplants שימו לב שאין הרבה אופציות והקומבינתיות בין הקטן לגדול קבועות. במקרה כזה הייתי משתמשת במקום שדות אלו בtype על פי שמות האופציות באתר
8. לא יכול להיות מוצר שלא הופיע באף הזמנה? קרדינליות 0:N יותר הגיונית פה
9. במודל הרשה שיש רק בן אחד אין משמעות לייצוג D או O, נשאיר ריק
10. האם קופונים וגיפט קארדים קלוונטי לתהליך העסקי שלכם?
11. כשנשמור חיפושים נרצה לדעת את תוכן החיפוש.
12. משתמש חייב לעשות הזמנה? קרדינליות 0:N יותר הגיוני פה.
13. מה מטרת הישות הזאת? האם כדי לבצע חיפוש באתר אני צריכה להכניס פרטי כתובת?  
    למה שלאותו משתמש יהיו כמה שמות?  
    מה זה כתובת שלוש נקודות?  
    ולמה אין קשר להזמנה? איפה הכתובת באה לידי ביטוי בהזמנה?
14. על מה הלקוח נותן ביקורת? על מוצא כלשהו או על האתר?  
    אם על מוצר, חסר קשר למוצר ואם על האתר זה לא קשור לתהליך העסקי ולכן לא רלוונטי
15. גם כאן, על מה נכתבת תגובה? האם יש לזה קשר לתהליך העסקי שלכם?

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



## הנחות מודל ה- ERD

|  |  |  |
| --- | --- | --- |
| **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, משתמש רשום מסוים או משתמש מזדמן מסוים 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 כאחת. * 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 כאחת. * מפתח ראשי: צירוף השדות name, address – כל צירוף של שם מלא מסוים וכתובת מסוימת באופן ייחודי באתר כך שלכל צירוף של שם מלא מסוים וכתובת מסוימת יהיה שם חברה אחד ומספר טלפון אחד שיכול להירשם בשדות המתאימים * phone# - בפרטי לקוח מסוימים יצוין מספר הטלפון של הלקוח. | כפי שמצוין בהנחות הקשר Details of למשתמש רשום מסוים יכולות להיות שמורות אינספור פרטי לקוח הנשמרים ברשימת הכתובות של המשתמש, בכל כתובת ברשימה זו ניתן לשים צירוף שונה לחלוטין של הכתובת והשם של הלקוח |
| 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 * date * shipping\_method * payment\_type | * מפתח ראשי: orderID – מספר סידורי מזהה של הזמנה מסוימת אשר לא יחזור על עצמו. * 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 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. |  |
| 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. * קיימות באתר גינות בעיצוב אישי וגינות מוכנות מראש אשר כמות החלקות מהגדלים השונים זהה ואף הזרעים שנכללים בגינה זהים אך מחירם שונה, זאת כתוצאה מעיצוב אישי של גינה באופן שזהה לגינה מוכנה מראש. * small\_count, large\_count – תכונות המכילות מספרים ערך שלם לא שלילי, באופן ספציפי התכונה small\_count גדולה או שווה ל2, כלומר, בגינה מסוימת יכולים להימצא מינימום של 2 זרעים אשר דורשים גודל חלקה קטן. תכונות אלו מייצגות את סוגי החלקות בגינה מבחינת סדר הגודל שלהן. |  |
| הקשר "Chosen" בין SEED ו GARDEN   * quantity | * כל זרע מסוים יכול להיכלל באינספור גינות או באף אחת מהן. לכן הקרדינליות של קשר זה בצד הישות GARDEN היא 0:N. * בכל גינה מסוימת יכולים להיכלל בין 1 ל8 זרעים. לכן הקרדינליות של קשר זה בצד הישות SEED היא 1:N. * quantity – כמות של מוצר מסוים שנכלל בעסקת חבילה מסוימת. שדה זה יכיל מספרים שלמים גדולים מ0. * בגינה מסוימת יכללו בדיוק כמות הזרעים שהיא יכולה להכיל וזה לפי הכמויות וסוגי החלקות המצוינים בישות הגינה המסוימת. |  |

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

* **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 ), Season, Size, Sun\_amount )
* **SEED\_TYPES** ( **Name** ( SEEDS ), **Type** )
* **CHOSENS** ( **Garden** ( GARDENS ), **Seed** ( SEEDS ), Quantity )
* **DETAILS** ( **Name**, **Address**, Company, Phone# )
* **DETAILS\_OF** ( **Email** ( USERS ), { **Name**, **Address** } ( DETAILS ) )
* **ORDERS** ( **OrderID**, Email ( USERS ), { Name, Address } ( DETAILS ), Date,  
   Shipping\_method, Payment\_type )
* **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).  
  חשוב בעינינו לציין שקיימת חשיבות נוספת בכך שהמפתח זר נמצא בORDER והוא מפתח זר לUSER והיא כי לא כל הזמנה בהכרח נעשית על ידי משתמש רשום ולכן במצב שהיא מבוצעת על ידי משתמש לא רשום השדה הזה יהיה בעל ערך ריק.  
  במילים אחרות היתרון שמתבטא פה הוא שבזכות כך שזהו מפתח זר מתאפשר לו הגדרתית להכיל ערך ריק, להבדיל ממפתח ראשי שלא יכול להכיל מפתח ראשי.
* **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. בדומה לשתי הקשרים הקודמים, אך הפעם רק בעקבות חוקי המעבר למודל הטבלאי, כאן בהכרח יתבטא הקשר כזוג שדות נוספים בצד הרבים של הקשר (ORDER) שיחד יהוו מפתח זר לצד היחיד של הקשר (DETAILS).  
  כאן 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 שתי שדות את המפתח הראשי של SEARCH.  
  גם כאן שתי 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 לשניים הקודמים גם כאן ניצור טבלה חדשה בה יהיו הפעם רק שתי שדות, כל אחד יהווה מפתח זר נפרד וביחד יהוו מפתח ראשי לטבלה חדשה אך הפעם שתי המפתחות הזרים הללו יהוו את אותה ישות (עם רשומות שיכולות להיות שונות).
* **ביטוי הקשר Included שבין ORDER ל-PRODUCT** – קשר רבים לרבים. בדומה לשלוש הקשרים הקודמים גם כאן ניצור טבלה חדשה כדי לייצג את הקשר, אך יתבטא הבדל בכך שכאן בנוסף למפתחות הזרים אשר יהיו גם מפתח זר, יהיה שדה נוסף שיכיל את הערך הרצוי של תכונה quantity המצוינת במודל הERD.  
  מפתח זר ראשון יגלם את המפתח הראשי של PRODUCT והמפתח הזה יכיל שדה אחד, מפתח זר שני יגלם את המפתח הראשי של ORDER וגם הוא יכיל שדה אחד.
* **ביטוי הקשר Chosen שבין GARDEN ל-SEED** – קשר רבים לרבים. באופן זהה לקשר הקודם יכיל שלוש שדות, הראשון מפתח זר לGARDEN, השני מפתח זר לSEED, שניהם יחד יהיו המפתח הראשי, ושדה שלישי שיכיל את הערך הרצוי של תכונה quantity המצוינת גם כאן במודל הERD.
* **ביטוי מודל ההורשה שבין PRODUCT ל-SEED ול-GARDEN** – מודל הורשה בדיד וחלקי. כאן, על פי חוקי המעבר למודל הטבלאי קיימת הבחירה בין לבטא את המודל ההורשה באמצעות טבלה אחת או טבלה לכל אחת מהישויות במודל ההורשה, במקרה שלנו באמצעות שלושה טבלאות.  
  בחרנו לבטא את מודל ההורשה באמצעות שלושה טבלאות כדי להימנע מערכי NULL רבים שבהכרח יהיו במימוש באמצעות טבלה אחת, זאת מכיוון שהחלטנו כי ערכי ה-NULL הרבים יעמדו בדרכנו בעת תחקור מסד הנתונים.

# פירוט שדות בסיס הנתונים והערכת גודלם

| **Table** | **Field** | | **Data Type** | **Size (Bytes)** |
| --- | --- | --- | --- | --- |
| **PRODUCTS** | Name | | Varchar(40) | 40 |
| Price | | Smallmoney | 4 |
| Discount | | Smallmoney | 4 |
| **Total Record Size (Bytes)** | | | **48** |
| **USERS** | Email | | Varchar(40) | 40 |
| Password | | Varchar(20) | 20 |
| **Total Record Size (Bytes)** | | | **60** |
| **SEARCHES** | SearchDT | | Datetime | 8 |
| IP\_address | | Varchar(15) | 15 |
| Email **(USERS)** | | Varchar(40) | 40 |
| Search\_text | | Varchar(20) | 20 |
| **Total Record Size (Bytes)** | | | **83** |
| **RESULTS** | Name **(PRODUCTS)** | | Varchar(40) | 40 |
| SearchDT | **(SEARCHES)** | Datetime | 8 |
| IP\_address | Varchar(15) | 15 |
| **Total Record Size (Bytes)** | | | **63** |
| **SEEDS** | Name **(PRODUCTS)** | | Varchar(40) | 40 |
| Season | |  |  |
| Size | | Char(5) | 5 |
| Sun\_amount | | Varchar(30) | 30 |
| **Total Record Size (Bytes)** | | |  |
| **SEED\_TYPES** | Name **(SEEDS)** | | Varchar(40) | 40 |
| Type | | Varchar(20) | 20 |
| **Total Record Size (Bytes)** | | | **60** |
| **GARDENS** | Name **(PRODUCTS)** | | Varchar(40) | 40 |
| Small\_count | | Tinyint | 1 |
| Large\_count | | Tinyint | 1 |
| **Total Record Size (Bytes)** | | | **42** |
| **CHOSENS** | Garden **(GARDENS)** | | Varchar(40) | 40 |
| Seed **(SEEDS)** | | Varchar(40) | 40 |
| Quantity | | Tinyint | 1 |
| **Total Record Size (Bytes)** | | | **81** |
| **DETAILS** | Name | | Varchar(40) | 40 |
| Address | | Varchar(150) | 150 |
| Company | | Varchar(25) | 25 |
| Phone# | | Varchar(17) | 17 |
| **Total Record Size (Bytes)** | | | **232** |
| **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 |
| Date | | Date | 3 |
| Shipping\_method | | Varchar(100) | 100 |
| Payment\_type | | Varchar(7) | 7 |
| **Total Record Size (Bytes)** | | | **344** |
| **RELATIONS** | Product1 **(PRODUCTS)** | | Varchar(40) | 40 |
| Product2 **(PRODUCTS)** | | Varchar(40) | 40 |
| **Total Record Size (Bytes)** | | | **80** |
| **INCLUSIONS** | OrderID **(ORDERS)** | | Int | 4 |
| Name **(PRODUCTS)** | | Varchar(40) | 40 |
| Quantity | | Tinyint | 1 |
| **Total Record Size (Bytes)** | | | **45** |

**\*\*זמני כאן --- מתחיל כאן --- זמני כאן\*\***

סוגי נתונים וגבולות הערכים שיכולים להכיל

Varchar(n) – string of maximal length of n characters

Smallmoney – Real number with up to 4 decimal places from -214,748.3648 to 214,748.3647

Datetime – any variation of the following variables

* YYYY – four digits from 1753 to 9999
* MM – two digits from 01 to 12
* DD – two digits from 01 to 31
* hh – two digits from 00 to 23
* mm – two digits from 00 to 59
* ss – two digits from 00 to 59
* n\* – zero to three digits from 0 to 999

Char(n) – string of exact length of n characters

Tinyint – integer in the range of 0 to 255

Int – integer in the range of -2,147,483,648 to 2,147,483,647

**\*\*זמני כאן --- עד לכאן --- זמני כאן\*\***

**הנחות והסברים נוספים:**

**< כאן מוצג פירוט חלקי. במידת הצורך יש לכלול בהגשה הסברים רלוונטיים להבנת תכנון השדות >**

* שדות התאריכים בטבלה SEARCHES נבחרו כתאריך ולא כתאריך/זמן, כיוון שהאתר אינו מאפשר חיפוש על פי שעות ספציפיות.

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

# אילוצי מרחב ערכים

**אילוצי בדיקה (CHECK):**

**< כאן מוצגת דוגמא אחת. יש לכלול בהגשה את כלל האילוצים בהתאם לנדרש במטלה, בליווי הקוד>**

* **אילוץ על EMAIL**: הוספנו אילוץ על השדה Email לטבלת בהן מופיע שדה מסוג זה. זהו שדה חשוב מכיוון שהלקוח צריך לקבל את כל פרטי ההזמנה לכתובת המייל שהזין. כמו כן, דרך המייל החברה מזהה את הלקוח באופן חד ערכי ומתקשרת עימו. שדה זה ייחודי עבור כל לקוח ולכן עלינו לבדוק את תקינותו.

CONSTRAINT CK\_Email CHECK (Email like '%@%.%')

**טבלאות חיפוש (Lookup Tables):**

**< כאן מוצגת דוגמא אחת. יש לכלול בהגשה את כלל טבלאות החיפוש בהתאם לנדרש במטלה, בליווי הקוד>**

* **טבלת חיפוש** **ל-Payments:** הוספנו טבלת חיפוש עבור סוג כרטיס האשראי (Type) מתוך הנחה כי לקוח יכול לשלם רק בכרטיסי אשראי מורשים אשר איתם עובדת החברה.

CREATE TABLE CreditTypes (

Type Varchar (20) PRIMARY KEY

)

INSERT INTO CreditLOOKUP VALUES

('VISA'), ('MasterCard'), ('JCB'), ('AmericanExpress'), ('UniunPay-CreditCard')

ALTER TABLE Payments

ADD CONSTRAINT FK\_credittype FOREIGN KEY (type) REFERENCRS CreditTypes (type)

# SCRIPT לבניית הטבלאות

**< כאן מוצגת דוגמא אחת. יש לכלול בהגשה את הקוד של כלל פקודות יצירת הטבלאות>**

CREATE TABLE Registered\_customers (

Email Varchar (40) not null,

First\_Name Varchar (20) not null,

Last\_Name Varchar (20) not null,

Age Int null,

Gender Char(1) null,

CONSTRAINT PK\_customers PRIMARY KEY (Email),

CONSTRAINT ck\_email CHECK (Email like '%@%.%'),

CONSTRAINT ck\_gender CHECK (Gender in ('M','F'))

)

# SCRIPT להזנת רשומות

**< כאן מוצגת דוגמא אחת. יש לכלול בהגשה את הקוד של כלל פקודות ההזנה הרלוונטיות>**

INSERT INTO Registered\_customers (Email, First\_Name, Last\_Name, Age, Gender)

VALUES

('abc@gmail.com', 'Guy', 'Fried', 23, 'M'),

('david\_levi@yahoo.co.il', 'David', 'Levi', 45, 'M'),

('eyal.golan@walla.co.il', 'Eyal', 'Golan', 50, 'M'),

('israelisrali@gmail.com', 'Israel', 'Israeli', 30, 'M'),

('noa\_kile@noakile.domain.com', 'Noa', 'Kirl', 18, 'F'),

# SCRIPT להסרת טבלאות ואילוצים

**< כאן מוצגת דוגמא אחת. יש לכלול בהגשה את הקוד של כלל פעולות ההסרה הרלוונטיות>**

ALTER TABLE Registered\_customers

DROP CONSTRAINT PK\_customers, CK\_email, CK\_gender

DROP TABLE Registered\_customers