

מערכות קבצים ומסדי נתונים

פרויקט הגשה – פרויקט תכנות

**מגישים**

שם מלא סטודנט 1: יהונתן כובאני ת.ז: 204397244

שם מלא סטודנט 2: דניאל מודילבסקי ת.ז: 312493000

**תוכן עניינים**

1. **מידול מסד הנתונים במודל ER** ..........................................................................03

* 1.1- תרשים ER נוכחי ........................................................................................03
* 1.2- מקרא לדיאגרמה ........................................................................................03
* 1.3 - תרשים דיאגרמת מחלקות נוכחי .................................................................04
* 1.4 - מפתחות ...................................................................................................05
* 1.5 - הסברים על הדיאגרמה ...............................................................................06
* 1.5.1 –ישויות שהם PK ................................................................................06
* 1.5.2 – קבוצות ישויות חלשות ......................................................................06
* 1.5.3 - סוגי קשרים ......................................................................................07

1. **כלל ההנחות לצורך יצירת תרשים ה- ERD** ........................................................08
2. **רשימת תלויות פונקציונליות** ..............................................................................06
3. **סכמה ראשונית** ................................................................................................10

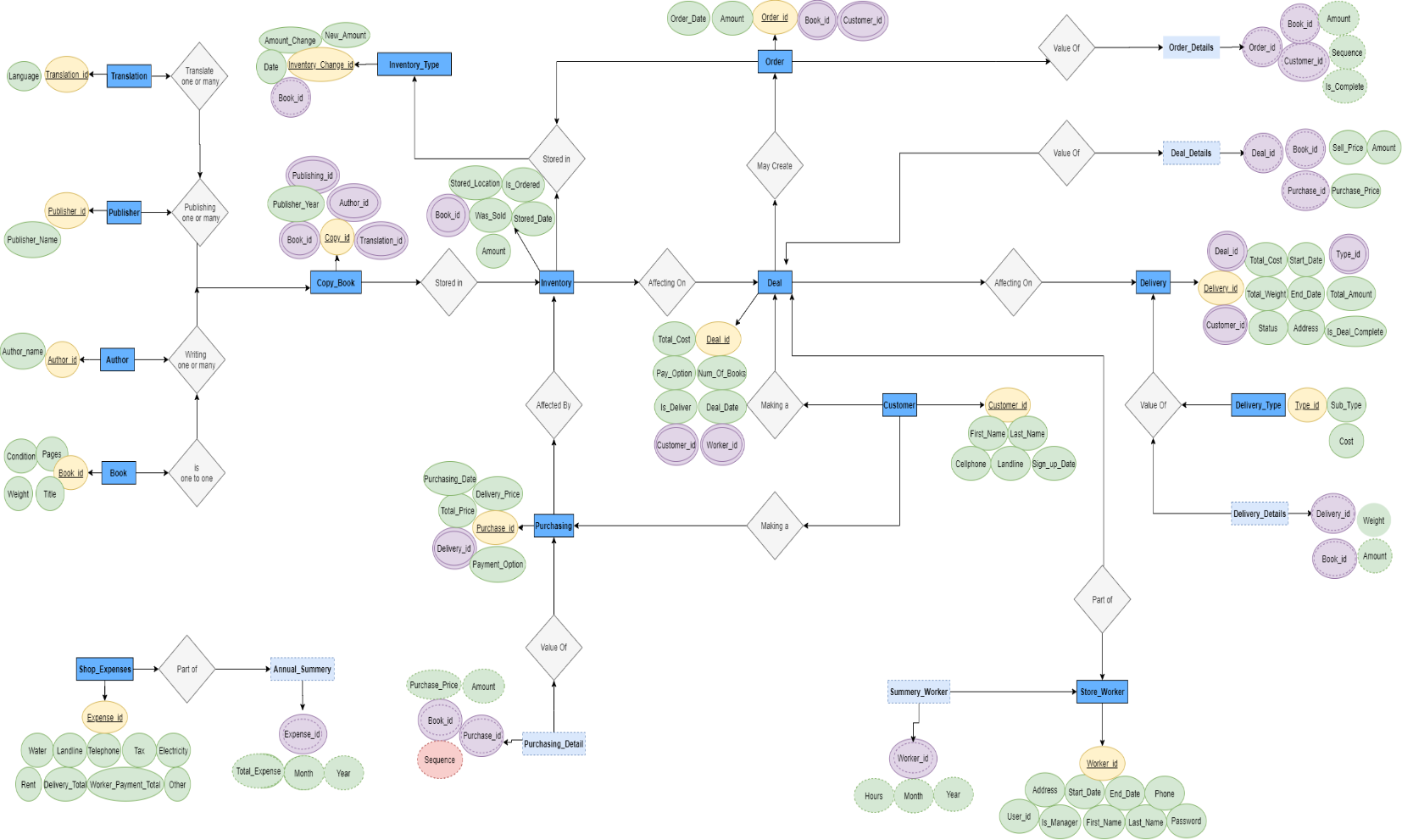
* 4.1 - שמות טבלאות ...........................................................................................10
* 4.2 – מבנה כל טבלה – שמות עמודות, טיפוסים ...................................................11
* 4.3 – מפתחות ..................................................................................................12

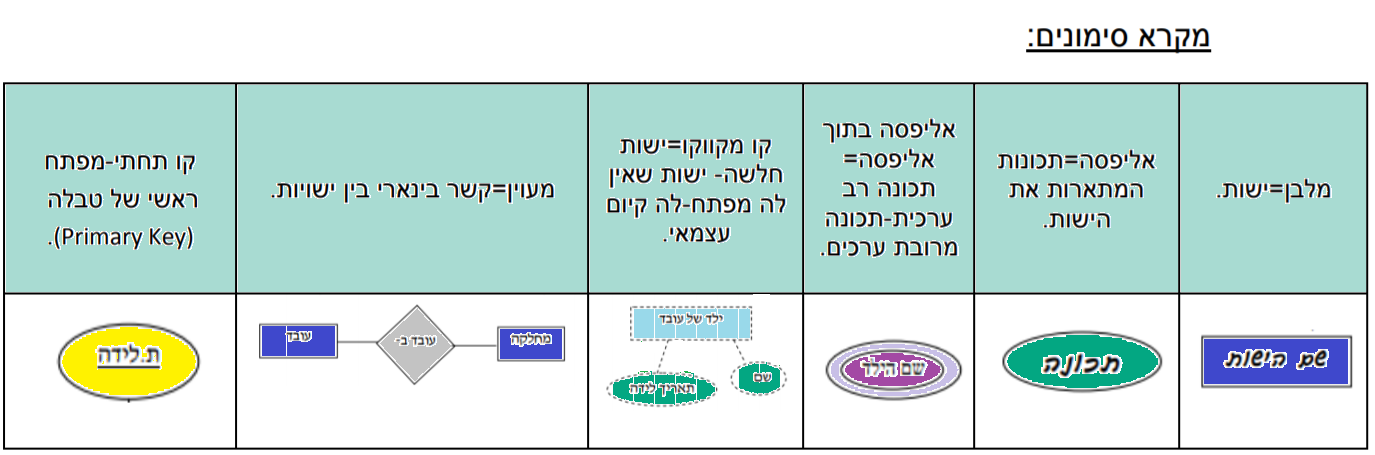
1. **סכמה של בסיס הנתונים לאחר נירמול ל- 3NF** ...................................................13

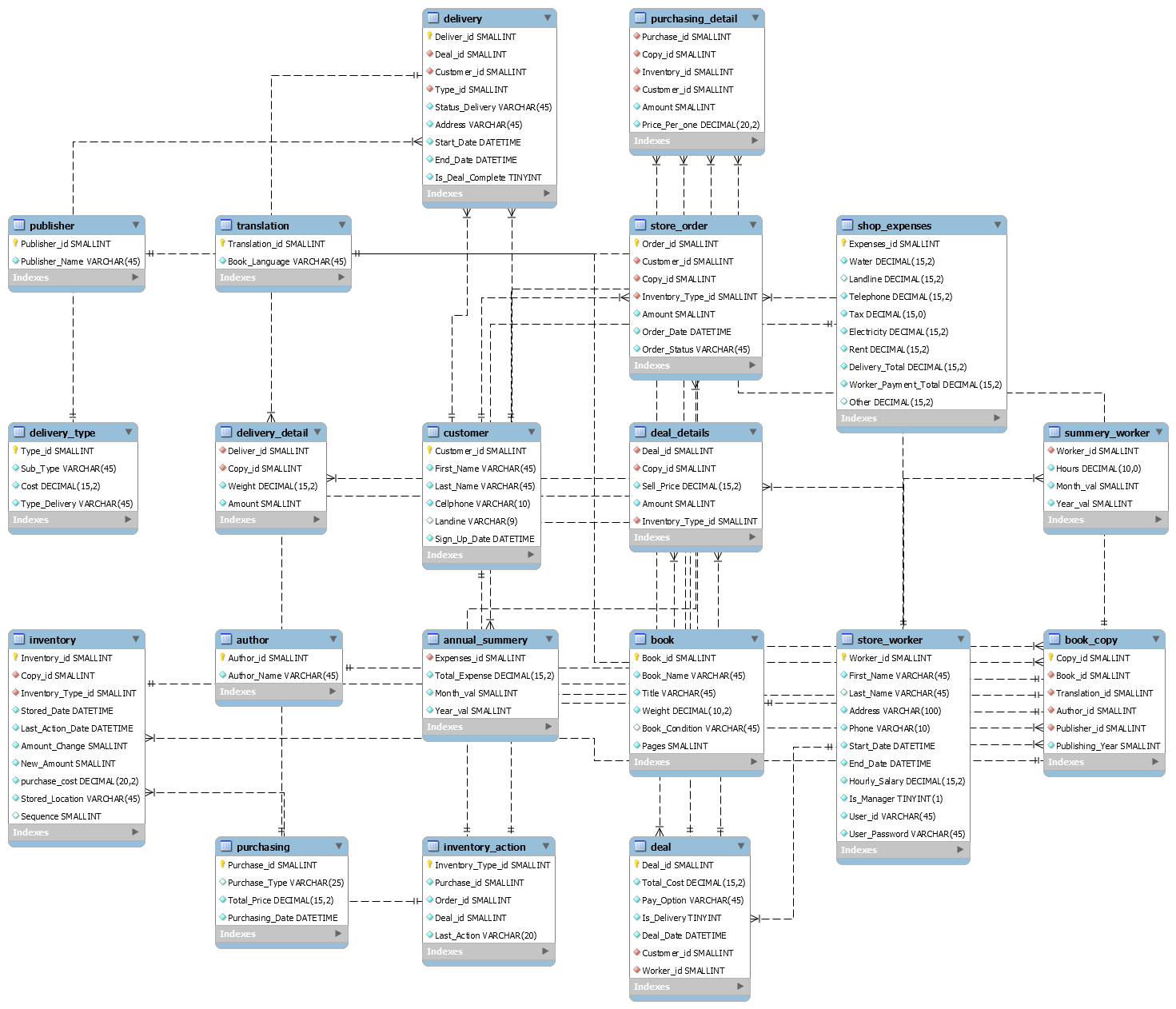
* 5.1 – הסבר כללי ...............................................................................................13
* 5.2 - נירמול טבלאות תוכן .................................................................................14
* 5.3 - נירמול טבלאות סוג ...................................................................................14
* 5.4 – נירמול מתאמים .......................................................................................15
* 5.5 – נירמול אובייקטים עצמאיים ......................................................................16
* 5.6 – סכמה ......................................................................................................17

1. **תוכן כל הטבלאות** .............................................................................................19
2. **מסמך שאילתות** ...............................................................................................26
3. **מסמך אלגברת יחסים** .......................................................................................33
4. **מידול מסד הנתונים במודל ER**

* 1.1 – **תרשים דיאגרמת ER נוכחי :**
* **הערה** – מצורף בזאת קובץ נלווה – "Diagram-ERD.PNG"



* 1.2 – **מקרא סימונים**
* 1.2 – **תרשים דיאגרמת מחלקות נוכחי :**
* **הערה** – מצורף בזאת קובץ נלווה – "Class-Diagram.PDF"



* 1.4 – **מפתחות**
* **הערה** – כל הקשרים שהם foreign key הינם קשרי CASCADE גם על Update וגם על Delete.
* 1.5.1 – **קבוצת ישויות שהם PK**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **סיבה** | **סוג קשר** | **שדה** | **יעד** | **סוג מפתח** | **מקור** |
| אובייקט עצמאי -  ניצור ממנו רשימת אובייקטים ללא תלות באחרים | / | / | / | Primary Key | סופר |
| אובייקט עצמאי -  ניצור ממנו רשימת אובייקטים ללא תלות באחרים | / | / | / | Primary Key | תרגום |
| אובייקט עצמאי -  ניצור ממנו רשימת אובייקטים ללא תלות באחרים | / | / | / | Primary Key | הוצאה לאור |
| אובייקט עצמאי -  ניצור ממנו רשימת אובייקטים ללא תלות באחרים | / | / | / | Primary Key | ספר |
| אובייקט ייחודי - כל שורה היא חח"ע | / | / | / | Primary Key | עותק ספר |
| אובייקט עצמאי -  ניצור ממנו רשימת אובייקטים ללא תלות באחרים | / | / | / | Primary Key | עובד |
| אובייקט עצמאי -  ניצור ממנו רשימת אובייקטים ללא תלות באחרים | / | / | / | Primary Key | הוצאות |
| אובייקט ייחודי - כל שורה היא חח"ע | / | / | / | Primary Key | רכש |
| אובייקט ייחודי - כל שורה היא חח"ע | / | / | / | Primary Key | הזמנה |
| אובייקט ייחודי - כל שורה היא חח"ע | / | / | / | Primary Key | עסקה |
| אובייקט ייחודי - כל שורה היא חח"ע | / | / | / | Primary Key | מלאי |
| טבלה קבועה - ללא שינויים בערכים | / | / | / | Primary Key | תנועות במלאי |
| אובייקט ייחודי - כל שורה היא חח"ע | / | / | / | Primary Key | משלוח |
| טבלה קבועה - ללא שינויים בערכים | / | / | / | Primary Key | סוג משלוח |

* 1.5.2 – **קבוצת ישויות חלשות**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **סיבה** | **סוג קשר** | **שדה** | **יעד** | **סוג מפתח** | **מקור** |
| קבוצת ישויות חלשה - ללא PK יכול להכיל שדות כפולים ועובד ביחס רבים ליחיד | / | / | / | / | רכש פרטים |
| קבוצת ישויות חלשה - ללא PK יכול להכיל שדות כפולים ועובד ביחס רבים ליחיד | / | / | / | / | הזמנה פרטים |
| קבוצת ישויות חלשה - ללא PK יכול להכיל שדות כפולים ועובד ביחס רבים ליחיד | / | / | / | / | משלוח פריטים |
| קבוצת ישויות חלשה - ללא PK יכול להכיל שדות כפולים ועובד ביחס רבים ליחיד | / | / | / | / | עסקה פריטים |
| קבוצת ישויות חלשה - ללא PK יכול להכיל שדות כפולים ועובד ביחס רבים ליחיד | / | / | / | / | סיכום חודשי עובד |
| קבוצת ישויות חלשה - ללא PK יכול להכיל שדות כפולים ועובד ביחס רבים ליחיד | / | / | / | / | סיכום חודשי חנות |

* 1.5.3 – **קבוצת ישויות חלשות**
* 1.4 – **מפתחות**
* 3 – **רשימת תלויות פונקציונליות**
* **הערה** – טבלת פתרון זאת רלוונטית לשלושת הסעיפים הבאים:

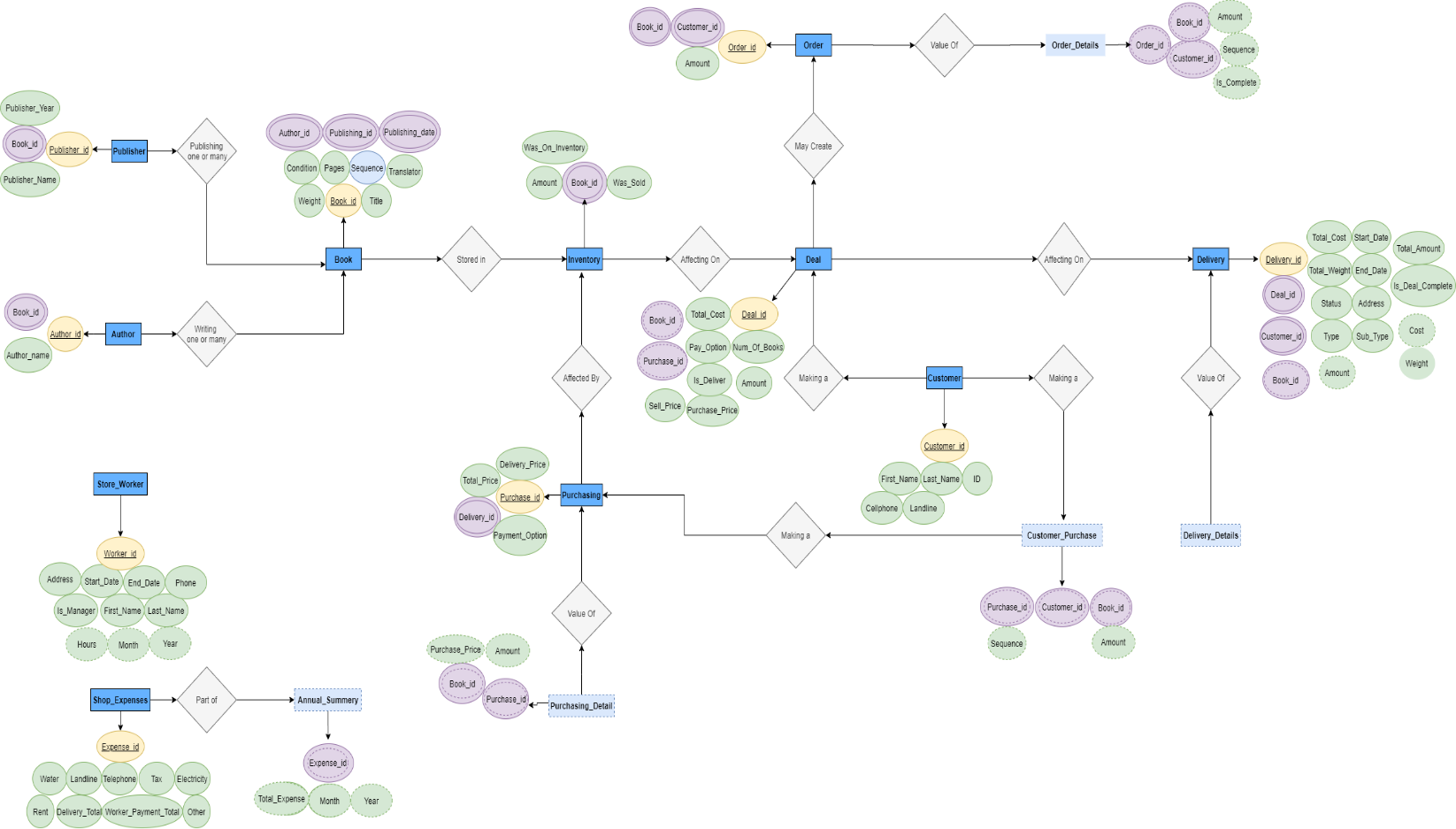
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **סיבה** | **סוג קשר** | **שדה** | **יעד** | **סוג מפתח** | **מקור** |
| לעותק ספר יש סופר שכתב אותו | One to One | Author\_id | עותק ספר | Foreign Key | סופר |
| לעותק ספר יש תרגום שבו הוא כתוב | One to One | Translation\_id | עותק ספר | Foreign Key | תרגום |
| לעותק ספר יש הוצאה לאור שמוציאה אותו לאור | One to One | Publisher\_id | עותק ספר | Foreign Key | הוצאה לאור |
| יצירת ערך לספר שיכיל פרטים של אחד מ-4 הנתונים הנ"ל, וכל שינוי של נתון (ויש המון מבחינה עסקית) יקבל עותק ספר משל עצמו עם id ייחודי | One to One | Book\_id | עותק ספר | Foreign Key | ספר |
| מניעת כפילויות של נתוני עובד על ידי הפרדה בין  הנתונים שלו שלא משתנים בתדירות לשעות העבודה  שלו שכן משתנים בתדירות | One to One | Customer\_id | סיכום חודשי עובד | Foreign Key | עובד |
| מניעת כפילויות של נתוני עובד על ידי הפרדה בין  הנתונים שלה שלא משתנים בתדירות להוצאות  החודשיות שלה שכן משתנות בתדירות | One to One | Expenses\_id | סיכום חודשי חנות | Foreign Key | הוצאות חנות |
| רכישה של החנות מתבצעת כשהספק הוא הלקוח | One to One | Customer\_id | רכש פרטים | Foreign Key | לקוח |
| הרכישה של החנות נכנסת למלאי החנות | One to One | Inventory\_id | רכש פרטים | Foreign Key | מלאי |
| על רכישה אחת יכולים להיות כמה פריטים | One to many | Purchase\_id | רכש פרטים | Foreign Key | רכש |
| רכישה מכילה עותקי ספרים | One to One | Copy\_id | רכש פרטים | Foreign Key | עותק ספר |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **סיבה** | **סוג קשר** | **שדה** | **יעד** | **סוג מפתח** | **מקור** |
| הזמנה היא לספרים | One to One | Copy\_id | הזמנה | Foreign Key | עותק ספר |
| הזמנה מושפעת מהאם הספר חזר למלאי | One to One | Inventory\_Type\_id | הזמנה | Foreign Key | תנועות במלאי |
| הזמנה מבוצעת על ידי לקוח | One to One | Customer\_id | הזמנה | Foreign Key | לקוח |
| על הזמנה אחת יכולים להיות כמה פריטים | One to many | Order\_id | הזמנה פרטים | Foreign Key | הזמנה |
| לקוח מבצע עסקה בחנות | One to One | Customer\_id | עסקה | Foreign Key | לקוח |
| עובד מטפל בעסקת הלקוח | One to One | Worker\_id | עסקה | Foreign Key | עובד |
| על עסקה אחת יכולים להיות כמה פריטים | One to many | Deal\_id | עסקה פרטים | Foreign Key | עסקה |
| עסקה מושפעת מהאם הספר קיים במלאי | One to One | Inventory\_Type\_id | עסקה פרטים | Foreign Key | תנועות במלאי |
| עסקה מורכבת מעותקים של ספרים | One to One | Copy\_id | עסקה פרטים | Foreign Key | עותק ספר |
| סיווג איזו פעולה אחרונה בוצעה על מלאי זה | One to One | Inventory\_Type\_id | מלאי | Foreign Key | תנועות במלאי |
| מלאי מורכב מעותקים של ספרים | One to One | Copy\_id | מלאי | Foreign Key | עותק ספר |
| משלוח יכול להיות אחד מ-5 סוגים שונים בתעריפים שונים | One to One | Type\_id | משלוח | Foreign Key | סוג משלוח |
| על עסקה אחת יכולים לצאת כמה משלוחים | One to many | Deal\_id | משלוח | Foreign Key | עסקה |
| משלוח מגיע לכתובת שהזין הלקוח | One to One | Customer\_id | משלוח | Foreign Key | לקוח |
| על משלוח אחד יכולים להיות מספר פריטים | One to many | Deliver\_id | משלוח פריטים | Foreign Key | משלוח |
| משלוח מורכב מעותקים של ספרים | One to One | Copy\_id | משלוח פריטים | Foreign Key | עותק ספר |

1. **כלל ההנחות לצורך יצירת תרשים ה- ERD**

* **הערה כללית** – כל אובייקט נבנה על פי דרישות הפרויקט – בא לידי ביטוי בשדות כל פרויקט ובדרישות העסקיות הנובעות ממנו.
* **מימד זמן** – מחייב הוספת מימד זמן כלשהו לרוב המתאמים והפעולות על מנת להבדיל ביניהם ברמת הביצוע.
* **רשימות עצמאיות** – נפריד גופים אשר הם ללא תלות אחד בשני, וניצור להם אובייקט שיהיה בסיס גם להרחבה של עוד שדות בהמשך במידה והחנות תרצה בכך. זאת על מנת לא ליצור עומס ממקוד בטבלאות מסוימות.
* **עותק ספר**
* יכולים להיות המון עותקים של ספרים שההבדל בינהם הוא לכל הפחות אחד.
* לספר יכולים להיות מספר מחברים או מספר תרגומים או מספר הוצאות לאור לכן נחזיק עותק ספר לכל אפשרות שכזאת.
* **מלאי**
* מחזיק כמויות של ספרים.
* מקשר בין פעילויות החנות הנוגעות בכמויות הספרים והימצאותן.
* תיעוד פעולה אחרונה – מלאי מתעדכן בכל פעולת "תנועת מלאי" שמתבצעת ותמיד ישמור את ההשפעה של הפעולה האחרונה (קיים ערך sequence) שמתעד מספר פעולות שנעשו על עותק מסויים במלאי.
* יכול להחזיק מלאי 0 מעותק של ספר , אך לא יכול להיות מלאי שלילי – מראה האם ספר מסוים היה אי פעם במלאי.
* עותק שלא קיים במלאי יקבל שורה חדשה, עותק שקיים יתעדכן בכמות ובסוג הפעולה.
* מושפע מרכש , עסקה, הזמנה.
* חלק מהספרים הם בחנות וחלק במחסן.
* **רכש**
* נעשה או מחנות שנסגרת או מלקוח , יבוא לידי ביטוי בשדה של סוג רכש.
* **תנועות במלאי**
* הצגה של איזו פעולה מתבצעת עכשיו על המלאי – טבלה קבועה של ערכים.
* **רכישה**
* פעולה עצמאית – תחזיק id אחד
* **הזמנה**
* פעולה עצמאית – תחזיק id אחד
* מותר להזמין פריט פר הזמנה.
* סטאטוס – בעת עדכון שחזר ערך מסוים למלאי , החנות מבצעת עדכון (update) לסעיף סטאטוס של ההזמנה – דבר המציג את התהליך מאחרוי הקלעים שעושה החנות כדי לסגור מעגל ולטפל בהזמנת לקוח.
* **עסקה**
* פעולה עצמאית – תחזיק id אחד
* **משלוח**
* סוג - יקבע לפי טבלה קבועה של סוג משלוח ועלות.W
* עלות – קיימת עלות כוללת ובנוסף יחושב בנפרד עלות משלוח המושפעת מסוג משלוח ומשקל המשלוח.
* כמה משלוחים יכולים להחזיק עסקה אחת – מראה שאפשר על עסקה אחת להוציא יותר ממשלוח אחד (פיצול משלוחים).
* יוצא רק על עסקאות שסומן עליהם כן על משלוח, יש אפשרות לקנות במקום.
* **לקוח**
* גורם עצמאי – יחזיק id אחד
* טלפון פרטי - ייחודי
* **עובד**
* גורם עצמאי – יחזיק id אחד
* עוהד יכול להיות מנהל.
* שם משתמש – ייחודי
* טלפון פרטי – ייחודי
* תיעוד של עובדים שכבר לא עובדים בחנות על ידי מימד זמן – תאריך התחלה וסוף.

**4. סכמה ראשונית**

* 4.1 – **סכמה - שמות טבלאות** 
* 4.2 – **מבנה כל טבלה – שמות עמודות, טיפוסים**

1 . Author – סופר

|  |  |  |
| --- | --- | --- |
| Author\_id | Author\_Name | Book\_id |

2 . Publisher – הוצאה לאור

|  |  |  |  |
| --- | --- | --- | --- |
| Publisher\_id | Publisher\_Name | Book\_id | Publisher\_Year |

3 . Translation– תרגום

|  |  |  |
| --- | --- | --- |
| Translation\_id | Translation\_Language | Book\_id |

4 . Book– ספר

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Book\_id | Book\_Name | Title | Weight | Book\_Condition | Pages |
| Translation\_id | Author\_id | Publisher\_id | Publishing\_date | sequence |  |

5 . Customer– לקוח

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Customer\_id | First\_Name | Last\_Name | Cellphone | Landline | Sign\_Up\_Date |

6 . Purchasing– רכש

|  |  |  |  |
| --- | --- | --- | --- |
| Purchase\_id | Payment\_Option | Total\_Price | Purchasing\_Date |

7 . Purchasing\_Detail – רכש פריטים

|  |  |  |  |
| --- | --- | --- | --- |
| Purchase\_id | Book\_id | Amount | Price\_Per\_one |

8 . Inventory– מלאי

|  |  |  |  |
| --- | --- | --- | --- |
| Book\_id | Amount | Was\_On\_Inventory | Was\_Sold |

9 . Store\_Worker – עובד

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Worker\_id | First\_Name | Last\_Name | Address | Phone | -- |
| Start\_Date | End\_Date | Hours | Is\_Manager | Month | Year |

10 . Shop\_Expenses – הוצאות

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Expenses\_id | Water | Landline | Telephone | Tax |
| Electricity | Rent | Delivery\_Total | Worker\_Payment\_Total | Other |

11 . Annual\_Summery – סיכום חנות חודשי

|  |  |  |  |
| --- | --- | --- | --- |
| Expenses\_id | Total\_Expense | Month\_val | Year\_val |

12 . Deal – עסקה

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Deal\_id | Sell\_Price | Pay\_Option | Is\_Delivery | Deal\_Date |
| Book\_id | Purchas\_id | Num\_Books | Worker\_id | Customer\_id |

12 . Store\_Order – הזמנה

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Order\_id | Customer\_id | Book\_id |  | Amount | Order\_Date | Order\_Status |

13 . Delivery – משלוח

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Deliver\_id | Deal\_id | Customer\_id | Type | Sub\_Type |
| Status\_Delivery | Address | Start\_Date | End\_Date | Is\_Deal\_Complete |
| Book\_id | Total\_Cost | Cost | Weight | Amount |

14 . Customer\_Purchase – רכישת לקוח

|  |  |  |  |
| --- | --- | --- | --- |
| Book\_id | Customer\_id | Purchase\_id | Amount |

* 4.3 – **מפתחות**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **סוג קשר** | **שדה** | **יעד** | **סוג מפתח** | **מקור** |
| / | / | / | Primary Key | סופר |
| / | / | / | Primary Key | תרגום |
| / | / | / | Primary Key | הוצאה לאור |
| / | / | / | Primary Key | ספר |
| / | / | / | Primary Key | עובד |
| / | / | / | Primary Key | הוצאות |
| / | / | / | Primary Key | רכש |
| / | / | / | Primary Key | הזמנה |
| / | / | / | Primary Key | עסקה |
| / | / | / | Primary Key | משלוח |
| / | / | / | / | רכש פרטים |
| One to One | Author\_id | ספר | Foreign Key | סופר |
| One to One | Translation\_id | ספר | Foreign Key | תרגום |
| One to One | Publisher\_id | ספר | Foreign Key | הוצאה לאור |
| One to One | Book\_id | סופר | Foreign Key | ספר |
| One to One | Book\_id | תרגום | Foreign Key | ספר |
| One to One | Book\_id | הוצאה לאור | Foreign Key | ספר |
| One to One | Customer\_id | רכש פרטים | Foreign Key | לקוח |
| One to One | Book\_id | רכש פרטים | Foreign Key | ספר |
| One to One | Book\_id | הזמנה | Foreign Key | ספר |
| One to One | Customer\_id | הזמנה | Foreign Key | לקוח |
| One to One | Customer\_id | עסקה | Foreign Key | לקוח |
| One to One | Worker\_id | עסקה | Foreign Key | עובד |
| One to One | Book\_id | מלאי | Foreign Key | ספר |
| One to many | Deal\_id | משלוח | Foreign Key | עסקה |
| One to One | Customer\_id | משלוח | Foreign Key | לקוח |
| One to One | Book\_id | משלוח | Foreign Key | ספר |

5. **סכמה של בסיס הנתונים לאחר נירמול ל- 3NF**

* 5.1 – **הסבר כללי** – תבנית R עם קבוצת תלויות F החלה עליה היא בצורה נורמאלית 3NF אם כל תלות X->A בסגור של F מקיים לפחות את אחד משלושת התנאים הבאים:
* 5.1.1 - X->A – היא טרוויאלית – A מוכל ב-X .
* 5.1.2 – X – מפתח על.
* 5.1.3 - A – הוא חלק ממפתח של R.
* 5.2 – **נירמול טבלאות תוכן**
* **צורך** – מניעת כפילות של נתונים , יצירת מציאות של הכלת נתונים כך שלאובייקט המכיל יש שדה מפתח יחיד ולו יכולים להיות מספר פריטים שונים שמשוייכים אליו.
* **עסקה** – **מצב לפני נירמול** – על אובייקט אחד של עסקה היו המון שורות של פריטים (במידה ולקוח קנה מעל פריט אחד) , כל פריט קיבל ID משלו , נוצר קושי בניהול עסקאות, שיוך פריט לעסקה עצמה ובניהול מסד הנתונים.
* **עסקה** – **מצב אחרי נירמול** – נוצרה הפרדה בין עסקה שהיא אובייקט X לבין "פרטי עסקה" שהם אובייקט A (סעיף 5.1.1) . הפרדה זאת נותנת יכולת של הכלה בה לעסקה אחת עם ID אחד יש מספר פריטים שונים המשוייכם לאותה עסקה, ללא כפל בנתונים, ללא סתירה ונותן יכולת קלה יותר בניהול מסד הנתונים.
* **רכש** – **מצב לפני נירמול** – על אובייקט אחד של רכש היו המון שורות של פריטים (במידה וחנות רוכשת מעל פריט אחד) , כל פריט קיבל ID משלו , נוצר קושי בניהול רכישות, שיוך פריט לרכישה עצמה ובניהול מסד הנתונים.
* **רכש** – **מצב אחרי נירמול** – נוצרה הפרדה בין רכישה שהיא אובייקט X לבין "פרטי רכישה " שהם אובייקט A (סעיף 5.1.1) . הפרדה זאת נותנת יכולת של הכלה בה ל רכישה אחת עם ID אחד יש מספר פריטים שונים המשוייכם לאותה רכישה, ללא כפל בנתונים, ללא סתירה ונותן יכולת קלה יותר בניהול מסד הנתונים.
* **משלוח** – **מצב לפני נירמול** – על אובייקט אחד של משלוח היו המון שורות של פריטים (במידה ולקוח מזמין מעל פריט אחד או מוציא מעל משלוח אחד לאותה עסקה) , כל פריט קיבל ID משלו , נוצר קושי בניהול משלוחים, שיוך פריט למשלוח ובניהול מסד הנתונים.
* **משלוח** – **מצב אחרי נירמול** – נוצרה הפרדה בין משלוח שהוא אובייקט X לבין "פרטי משלוח" שהם אובייקט A (סעיף 5.1.1) . הפרדה זאת נותנת יכולת של הכלה בה למשלוח אחד עם ID אחד יש מספר פריטים שונים המשוייכם לאותו משלוח, ללא כפל בנתונים, ללא סתירה ונותן יכולת קלה יותר בניהול מסד הנתונים.
* 5.3 – **נירמול טבלאות סוג**
* **צורך** – מניעת בזבוז נפח מסד הנתונים , ומניעת סרבול עדכון של נתונים שלא מתעדכנים והינם קבועים.
* **משלוח** – **מצב לפני נירמול** – לכל משלוח היום 3 עמודות נוספות של סוג משלוח תת סוג משלוח ועלות שהינם נתוהים קבועים ויצאו סירבול ועומס ל-DB ככל שנוספו שורות תוכן.
* **משלוח** – **מצב אחרי נירמול** – הוצאת 3 הנתונים הקבועים לטבלה אחת והקישור בינה לבין משלוח על ידי מפתח זר מה שמצמם חזרתיות של נתונים ובזבזוז זכרון מקון.
* **תנועות מלאי** – **מצב לפני נירמול** – לכל פעולה שקשורה למלאי הייתה נוספת שורה חדשה עם אותו מספר עותק ספר סוג הפעולה והשינוי. מה שיצר עומס רב על טבלת המלאי, אי סדר בכמויות המלאי ובהשפעות עליו**.**
* **תנועות מלאי** – **מצב אחרי נירמול** – קיטלוג אפשרויות ההשפעה על המלאי ל-3 אופציות והוצאתן החוצה מהמלאי כולל ההשפעה על הכמות. מצב שבו אין כפילות יותר בשורות בטבלת המלאי אלה עדכון הכמות הנוכחית + סוג הפעולה האחרונה שבוצעה. בשיטה זאת מונעים סירבול מסד הנתונים ומקבלים תמונה עדכנית תמיד של כמות המלאי לכל עותק ספר ומה הפעולה האחרונה שהושפע ממנה.
* 5.4 – **נירמול מתאמים**
* **צורך** – מניעת כפילות של נתונים וסתירה של אמינות הטבלה מאי התאמות בין טבלאות שונות, הקלה בהליכי עדכון ומחיקה.
* **עותק** – **מצב לפני נירמול** – לא היה קיים , הייתה כפילות של ספרים, וחוסר סדר בחיבור בין ספר , עורך , הוצאה לאור, ותרגום ביחסים של אחד לרבים לדוגמה ספר אחד עם מספר עורכים.
* **עותק** – **מצב אחרי נירמול** – ייחדוי ובעל מפתח, מתאם בין ארבעת האובייקטים ויוצר הפרדה בין כל סוג וסוג תוך אפשרות לייצר המון סוגי עותקים שונים.
* **מלאי** – **מצב לפני נירמול** – לא היה קיים, היה עומס בשדות שונים עבור ספר, שגם ככה היה עמוס מחזרתיות וכפילות של שורות משילוב עם אובייקטים אחרים כמו עורך.
* **מלאי** – **מצב אחרי נירמול** – ייחדוי ובעל מפתח, מתאם בין סוגי עותקים לכל פעולה המשפיעה על המלאי ויוצר הפרדה בין כל סוג וסוג תוך אפשרות לייצר המון סוגי עותקים שונים ולנהל אותם מבחינה כמותית.
* 5.5 – **רשימה עצמאית**
* **צורך** – מניעת בעיות כפילות נתונים,מניעת נפח זיכרון מיותר ויצירת ערכים עצמאיים ללא תלות שתדירות השינוי שלהם מאוד נמוכה.
* **עובד** – **מצב לפני נירמול** – עובד היה מכיל פרטי עובד ובנוסף פרטי שעות חודשיות (נתון שהיה מתעדכן בתדירות גדולה) ויצר שורות רבות ומיותרות של נתונים קבועים של עובד מסוים.
* **עובד** – **מצב אחרי נירמול** – יצירת רשימת עובדים ללא תלות באף גורם אחר, ועליהן נבנה מתאמים על מנת לקשר בינן לבין אובייקטים אחרים כמו שעות.
* **לקוח** – **מצב לפני נירמול** – לקוח היה מכיל פרטי לקוח ובנוסף פרטי רכישות שביצע (נתון שהיה מתעדכן בתדירות לא קטנה) ויצר שורות רבות ומיותרות של נתונים קבועים של לקוח מסוים.
* **לקוח** – **מצב אחרי נירמול** – יצירת רשימת לקוחות ללא תלות באף גורם אחר, ועליהן נבנה מתאמים על מנת לקשר בינן לבין אובייקטים אחרים כמו ספר.
* **עורך** – **מצב לפני נירמול** – עורך היה מכיל פרטי עורך ובנוסף פרטי ספרים שאותם כתב (נתון שהיה מתעדכן בתדירות גדולה) ויצר שורות רבות ומיותרות של נתונים קבועים של עורך מסוים.
* **עורך** – **מצב אחרי נירמול** – יצירת רשימת עורכים ללא תלות באף גורם אחר, ועליהן נבנה מתאמים על מנת לקשר בינן לבין אובייקטים אחרים כמו ספר.
* **תרגום** – **מצב לפני נירמול** – תרגום היה מכיל פרטי תרגום ובנוסף פרטי ספרים שמתורגמים לשפה זו (נתון שהיה מתעדכן בתדירות גדולה) ויצר שורות רבות ומיותרות של נתונים קבועים של עורך מסוים.
* **תרגום** – **מצב אחרי נירמול** – יצירת רשימת תרגומים ללא תלות באף גורם אחר, ועליהן נבנה מתאמים על מנת לקשר בינן לבין אובייקטים אחרים כמו ספר.
* **הוצאה לאור** – **מצב לפני נירמול** – הוצאה לאור הייתה מכילה פרטי הוצאה לאור ובנוסף פרטי ספרים שהוציאה (נתון שהיה מתעדכן בתדירות גדולה) מה שיצר שורות רבות ומיותרות של נתונים קבועים של הוצאה לאור מסוימת.
* **הוצאה לאור** – **מצב אחרי נירמול** – יצירת רשימת הוצאות לאור ללא תלות באף גורם אחר, ועליהן נבנה מתאמים על מנת לקשר בינן לבין אובייקטים אחרים כמו ספר.
* **ספר** – **מצב לפני נירמול** – ספר היה מכיל פרטי ספר ובנוסף פרטי עורך, תרגום, הוצאה לאור שקשורים לספר (נתון שהיה קיים ביחס אחד לרבים) ויצר שורות רבות ומיותרות של נתונים קבועים של ספר מסוים.
* **ספר** – **מצב אחרי נירמול** – יצירת רשימת ספרים ללא תלות באף גורם אחר, ועליהן נבנה מתאמים על מנת לקשר בינן לבין אובייקטים אחרים כמו עותק של ספר.
* **הוצאות חנות** – **מצב לפני נירמול** – הוצאות חנות היו מכילות פרטי חשבונות ובנוסף הוצאות לפי חודש (נתון שהיה קיים ביחס אחד לרבים) ויצר שורות רבות ומיותרות של נתונים קבועים של הוצאות החנות .
* **הוצאות חנות** – **מצב אחרי נירמול** – יצירת רשימת הוצאות חודשית ללא תלות באף גורם אחר, ועליהן נבנה מתאמים על מנת לקשר בינן לבין אובייקטים אחרים כמו עותק של סוג ההוצאות.
* 5.6 – **סכמה לאחר נירמול 3NF**

1 . Author – סופר

|  |  |
| --- | --- |
| Author\_id | Author\_Name |

2 . Publisher – הוצאה לאור

|  |  |
| --- | --- |
| Publisher\_id | Publisher\_Name |

3 . Translation– תרגום

|  |  |
| --- | --- |
| Translation\_id | Translation\_Language |

4 . Book– ספר

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Book\_id | Book\_Name | Title | Weight | Book\_Condition | Pages |

5 . Book\_Copy– עותק מספר

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Copy\_id | Book\_id | Translation\_id | Author\_id | Publisher\_id | Publishing\_Year |

6 . Customer– לקוח

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Customer\_id | First\_Name | Last\_Name | Cellphone | Landline | Sign\_Up\_Date |

7 . Purchasing– רכש

|  |  |  |  |
| --- | --- | --- | --- |
| Purchase\_id | Purchase\_Type | Total\_Price | Purchasing\_Date |

8 . Purchasing\_Detail – רכש פריטים

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Purchase\_id | Copy\_id | Inventory\_id | Customer\_id | Amount | Price\_Per\_one |

9 . Inventory– מלאי

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inventory\_id | purchase\_cost | Inventory\_Type\_id | Stored\_Date | Copy\_id |
| Amount\_Change | New\_Amount | Last\_Action\_Date | Stored\_Location | -- |

10 . Inventory\_Action – תנועות במלאי

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inventory\_Type\_id | Purchase\_id | Order\_id | Deal\_id | Last\_Action |

11 . Store\_Worker – עובד

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Worker\_id | First\_Name | Last\_Name | Address | Phone | -- |
| Start\_Date | End\_Date | Hourly\_Salary | Is\_Manager | User\_id | User\_Password |

12 . Summery\_Worker – סיכום חודשי עובד

|  |  |  |  |
| --- | --- | --- | --- |
| Worker\_id | Hours | Month\_val | Year\_val |

13 . Shop\_Expenses – הוצאות

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Expenses\_id | Water | Landline | Telephone | Tax |
| Electricity | Rent | Delivery\_Total | Worker\_Payment\_Total | Other |

14 . Annual\_Summery – סיכום חנות חודשי

|  |  |  |  |
| --- | --- | --- | --- |
| Expenses\_id | Total\_Expense | Month\_val | Year\_val |

15 . Deal – עסקה

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Deal\_id | Total\_Cost | Pay\_Option | Is\_Delivery | Deal\_Date | Customer\_id | Worker\_id |

16 . Deal\_Details – עסקה פרטים

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Deal\_id | Copy\_id | Sell\_Price | Amount | Inventory\_Type\_id |

17 . Store\_Order – הזמנה

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Order\_id | Customer\_id | Copy\_id | Inventory\_Type\_id | Amount | Order\_Date | Order\_Status |

18 . Delivery\_Type – סוג משלוח

|  |  |  |  |
| --- | --- | --- | --- |
| Type\_id | Sub\_Type | Cost | Type\_Delivery |

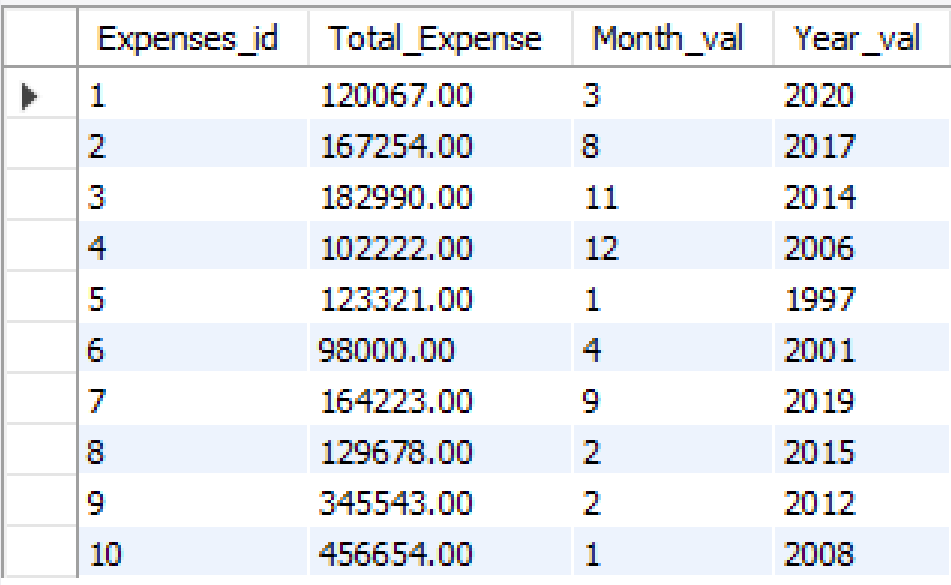
19 . Delivery – משלוח

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Deliver\_id | Deal\_id | Customer\_id | Type\_id | -- |
| Status\_Delivery | Address | Start\_Date | End\_Date | Is\_Deal\_Complete |

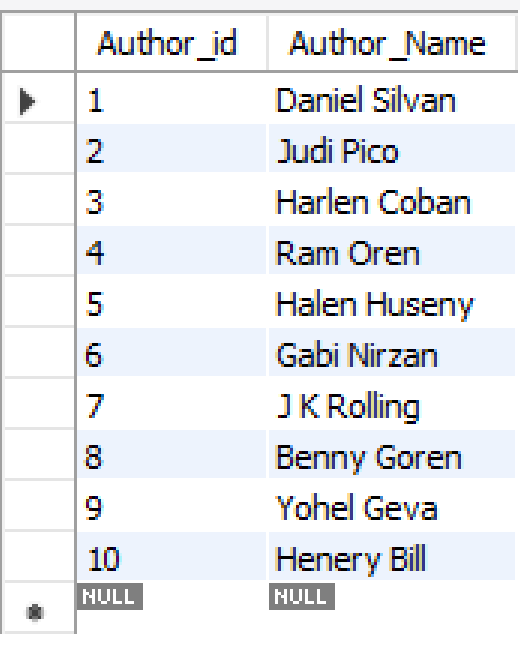
20 . Delivery\_Detail – משלוח פרטים

|  |  |  |  |
| --- | --- | --- | --- |
| Deliver\_id | Copy\_id | Weight | Amount |

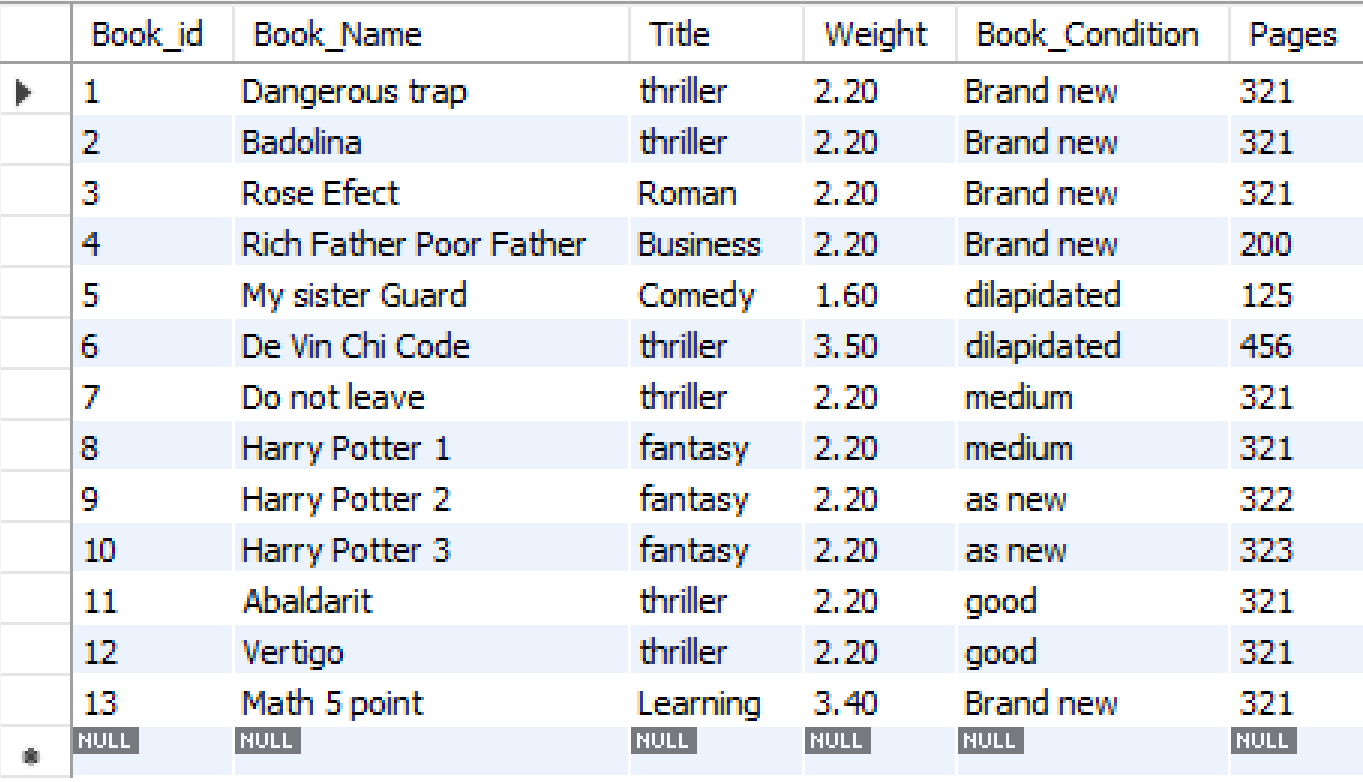
**6. תוכן כל הטבלאות**

**annual\_summery**

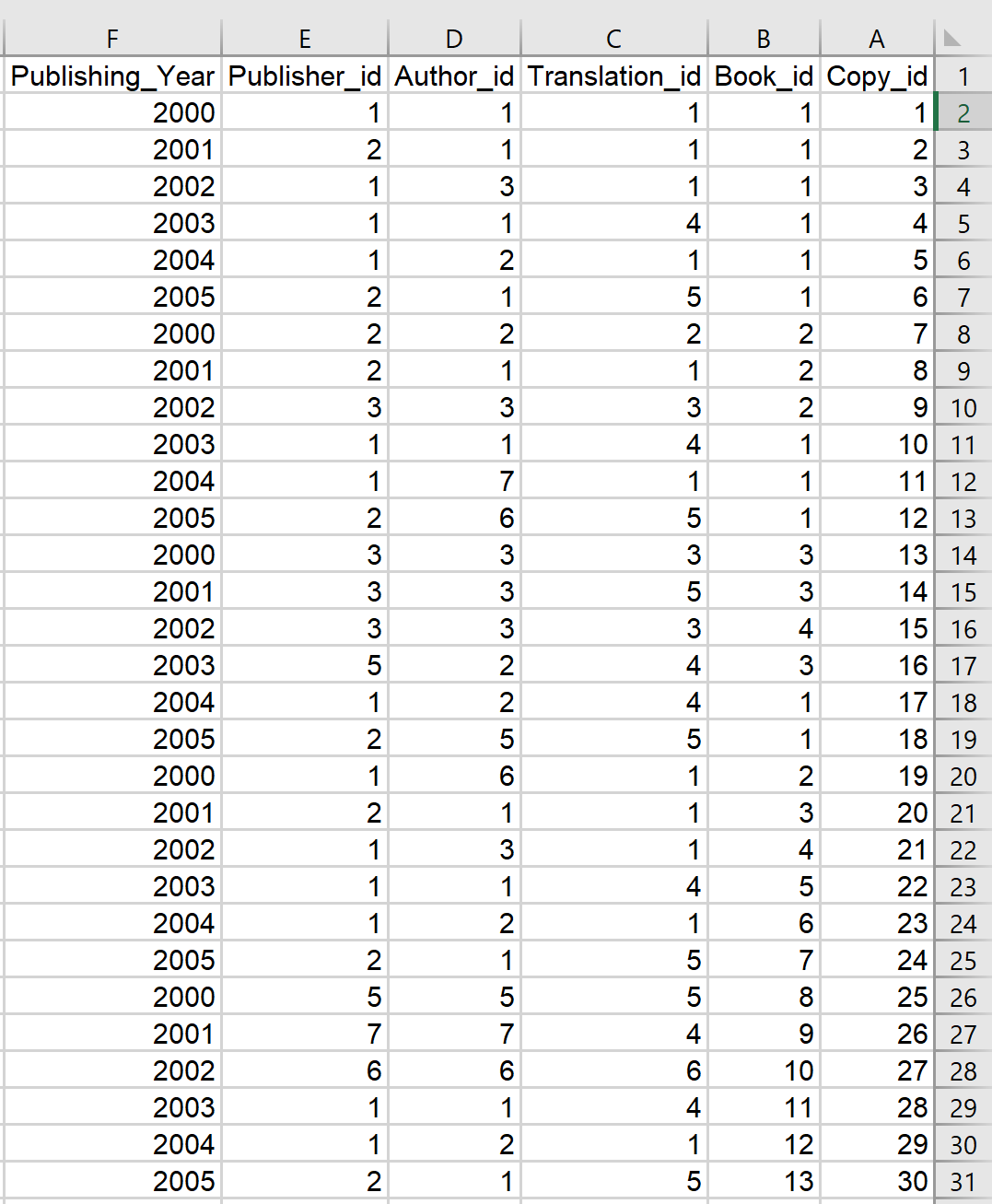
**author**

****

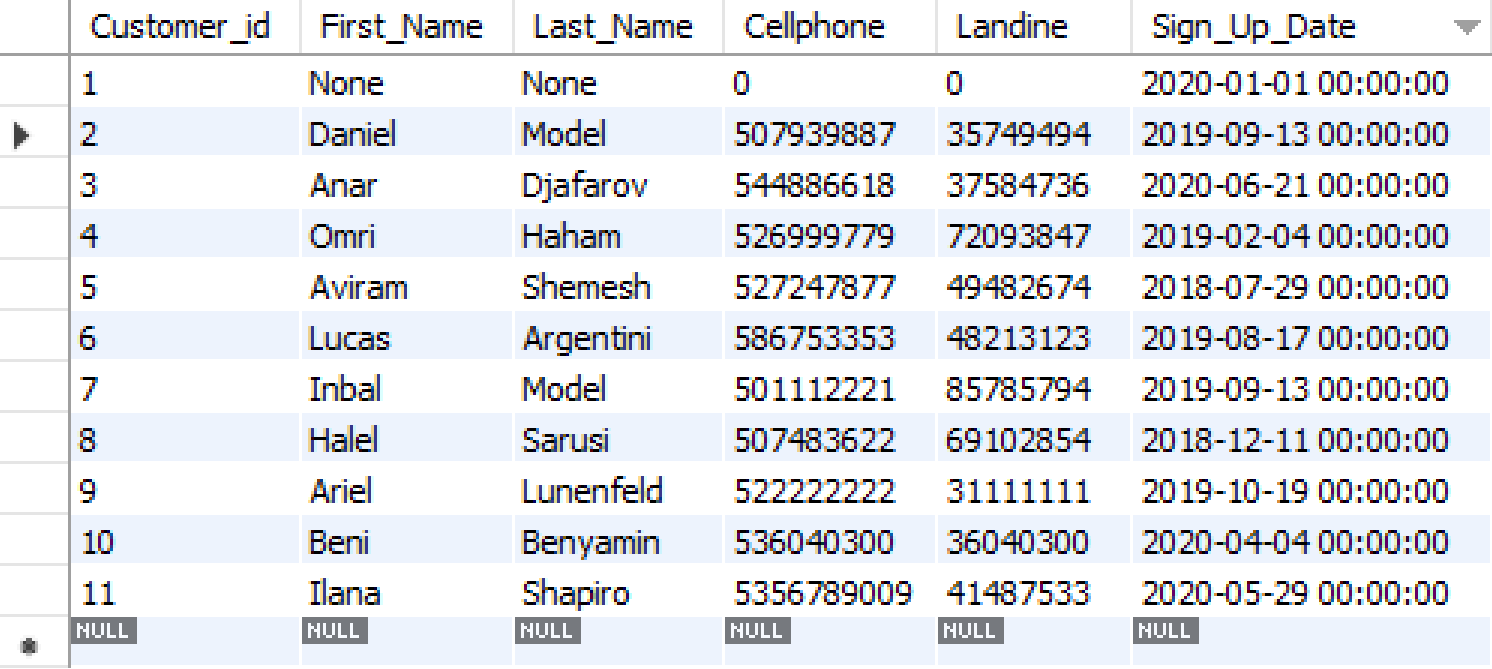
**Book**

****

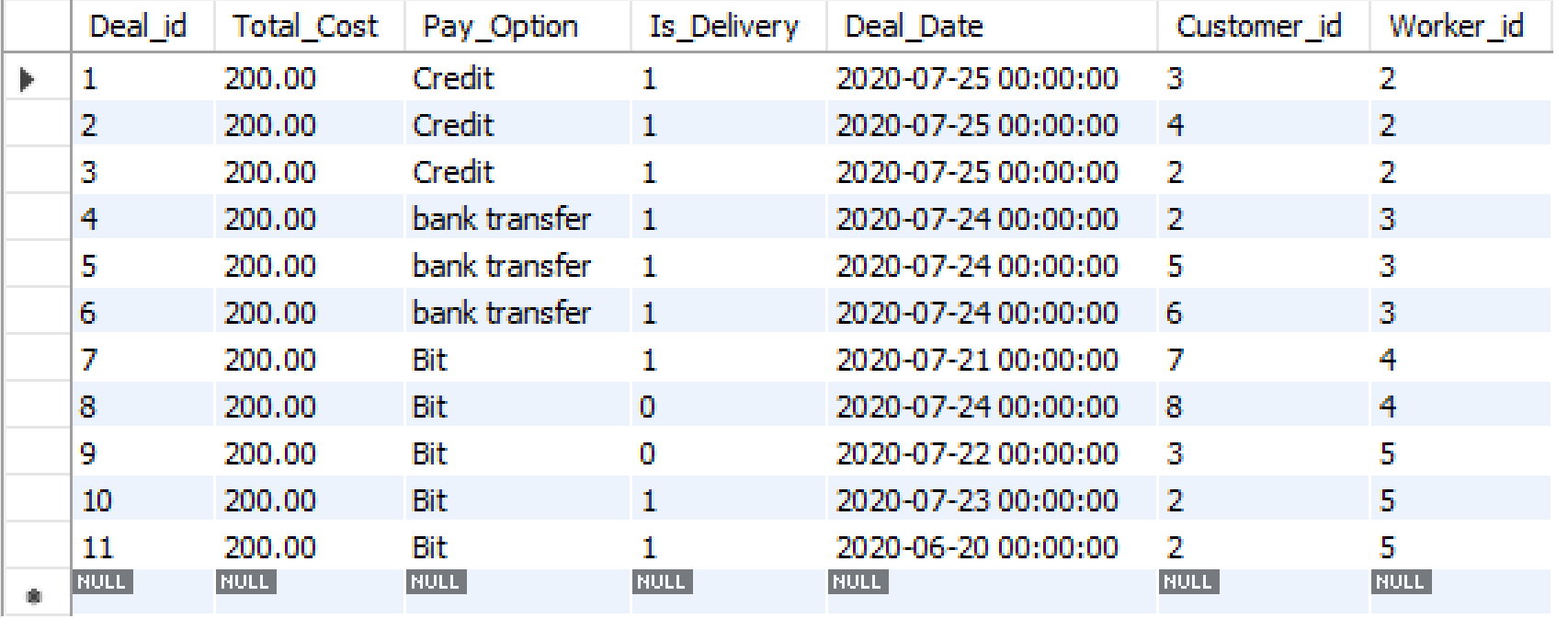
**book\_copy**

****

**Customer**

****

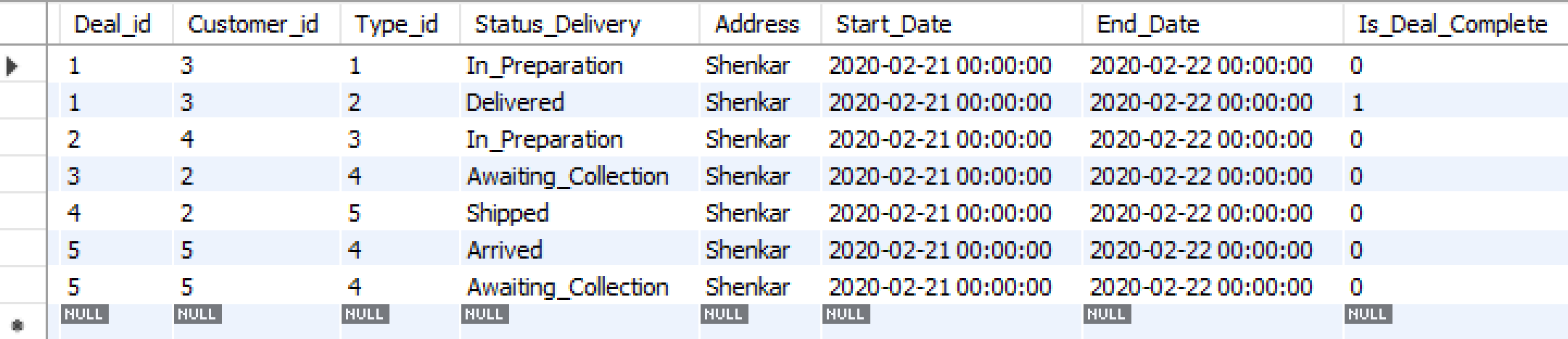
**Deal**

****

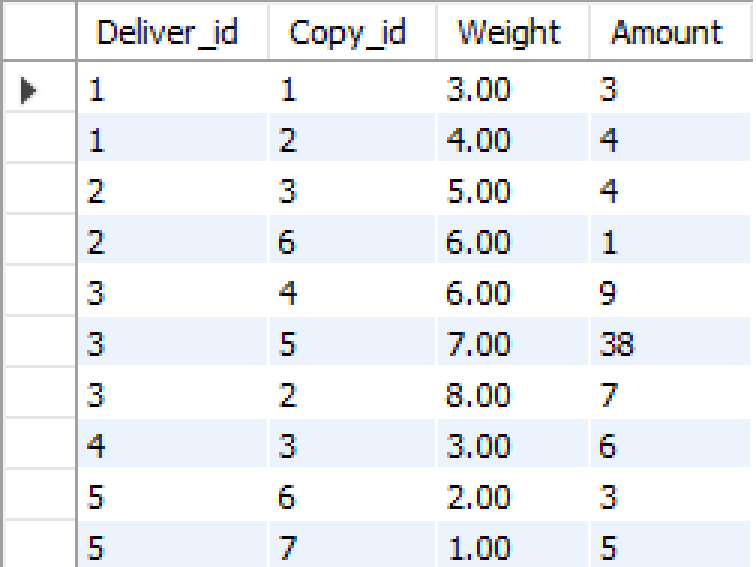
**deal\_details**

****

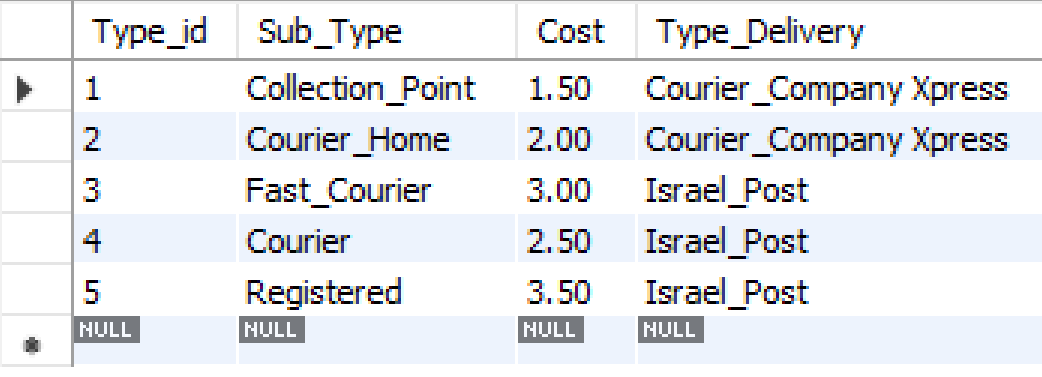
**Delivery**

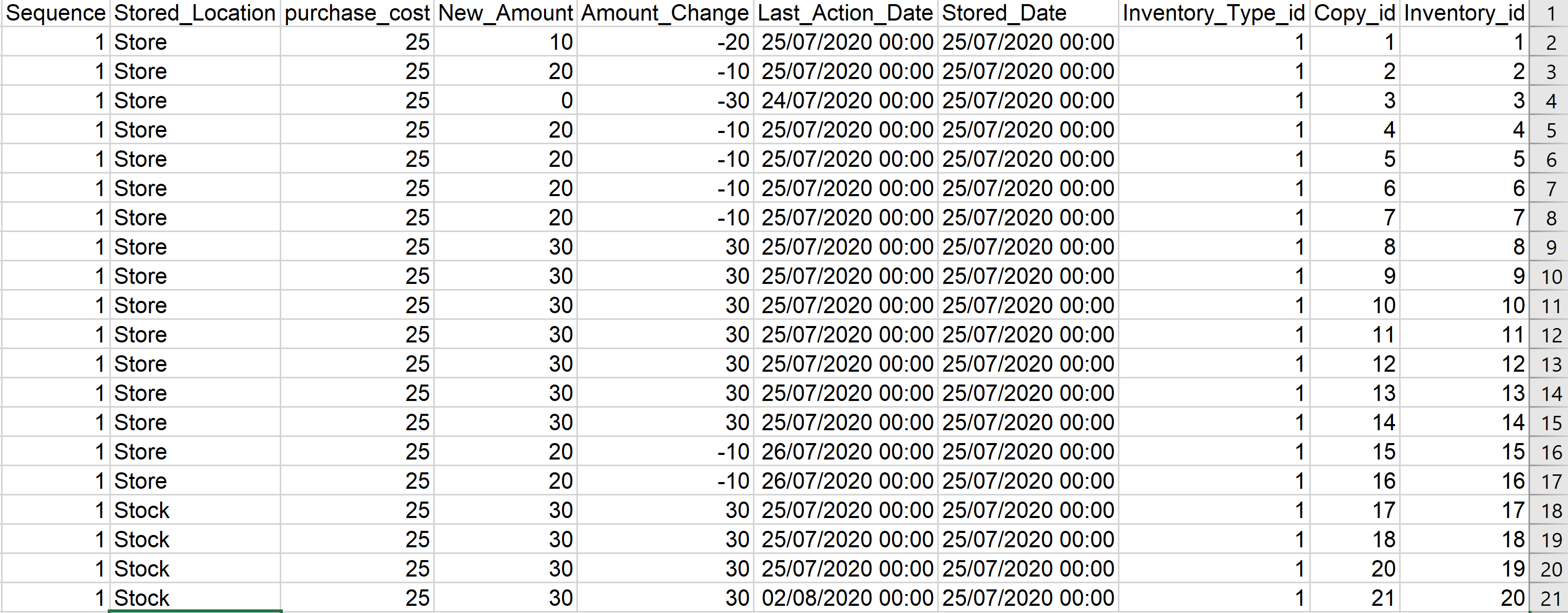
****

**delivery\_detail**

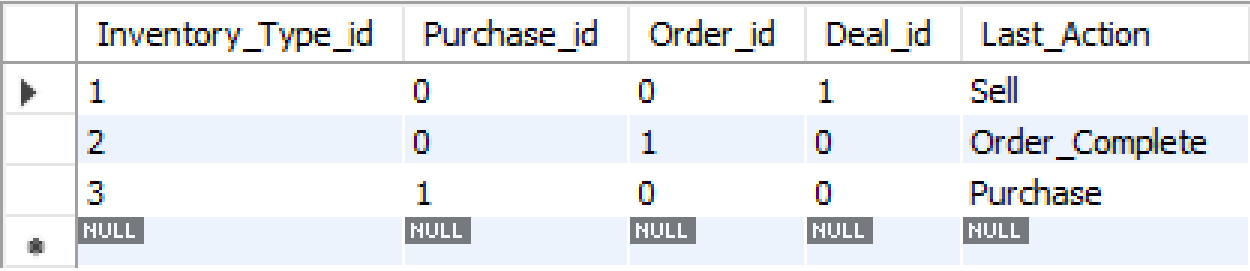
****

**delivery\_type**

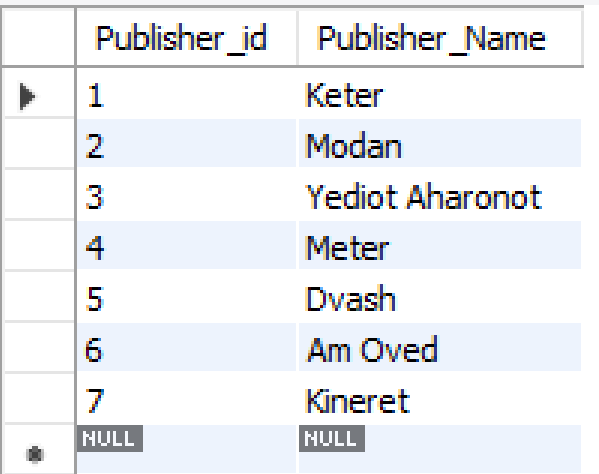
****

**inventory**

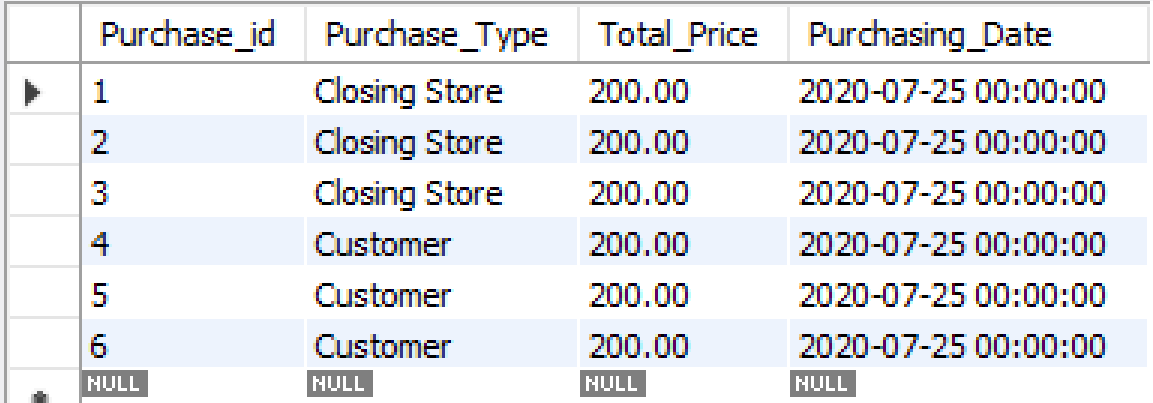
**inventory\_action**

****

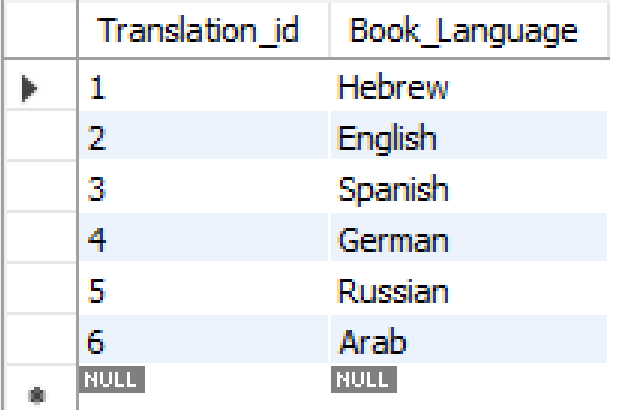
**Publisher**

****

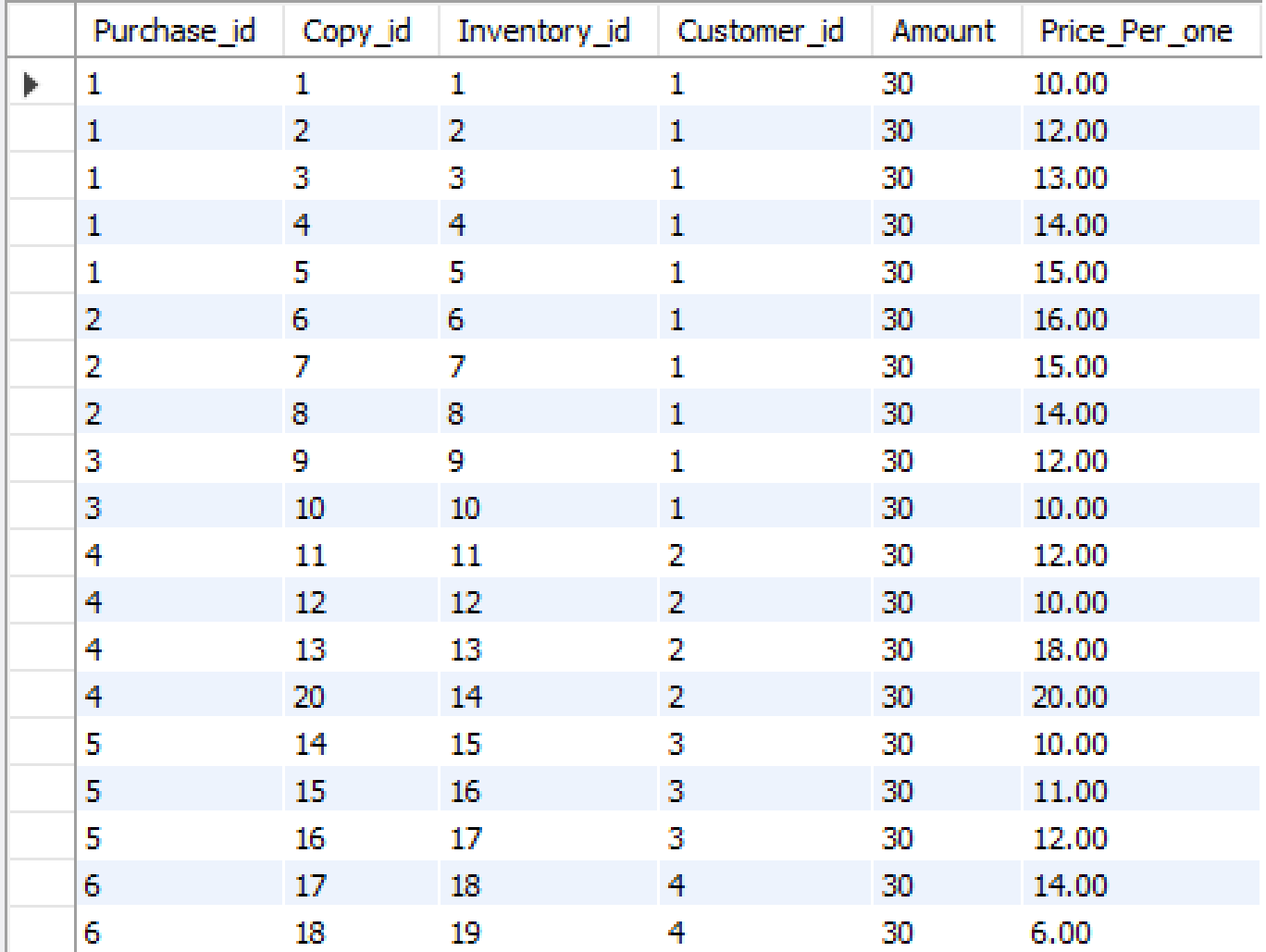
**Purchasing**

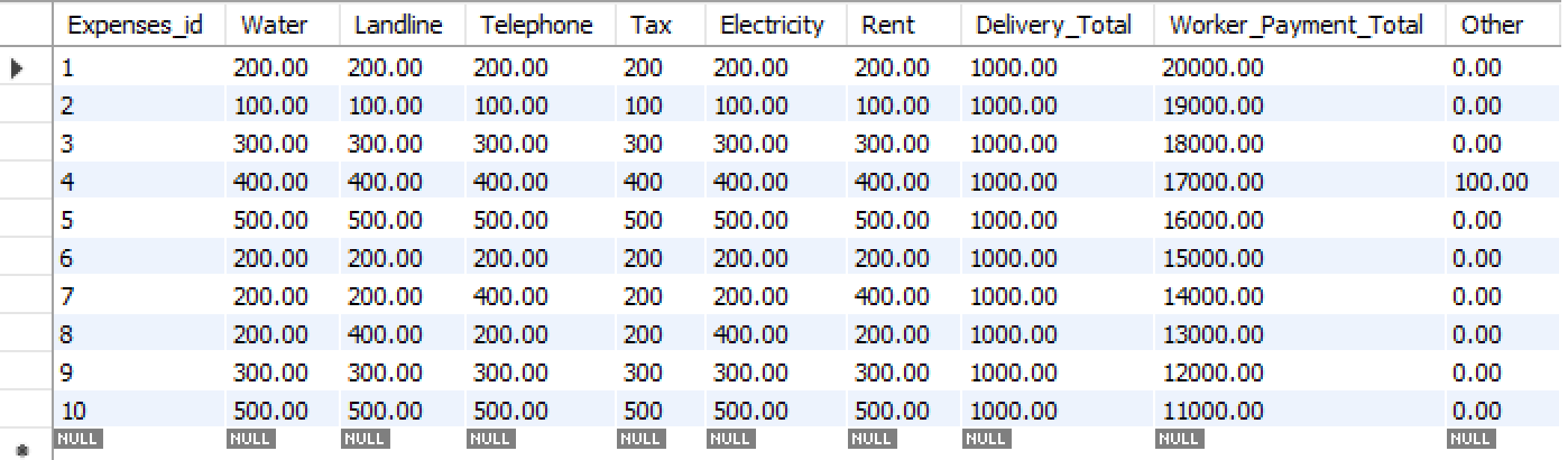
****

**Translation**

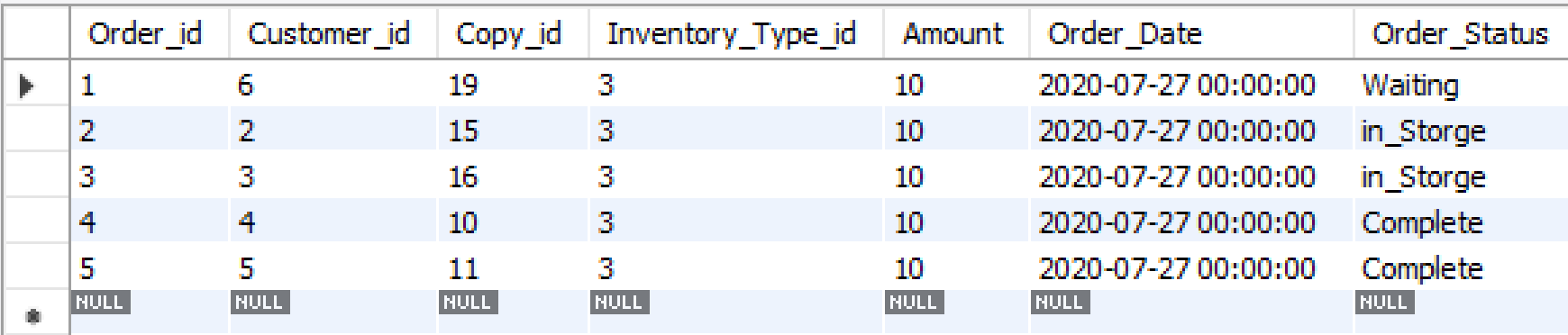
****

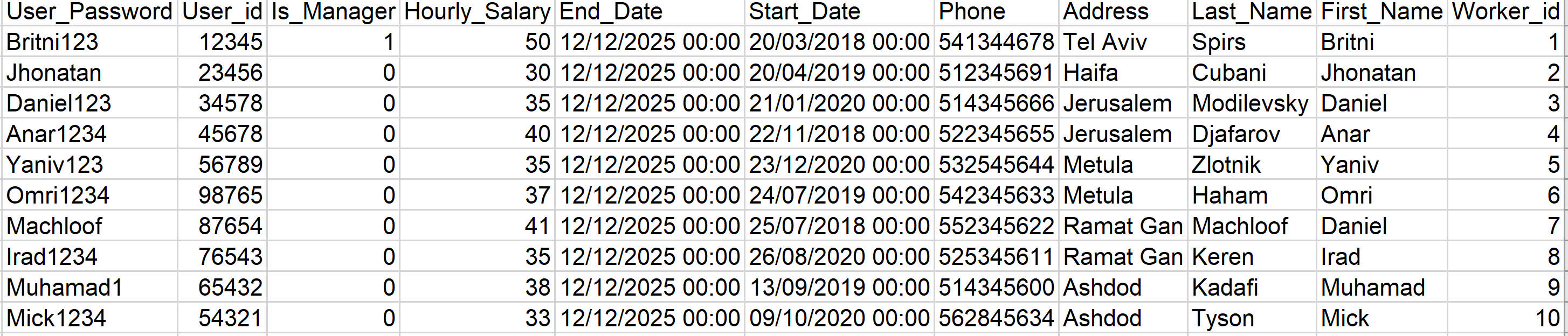
**purchasing\_detail**

****

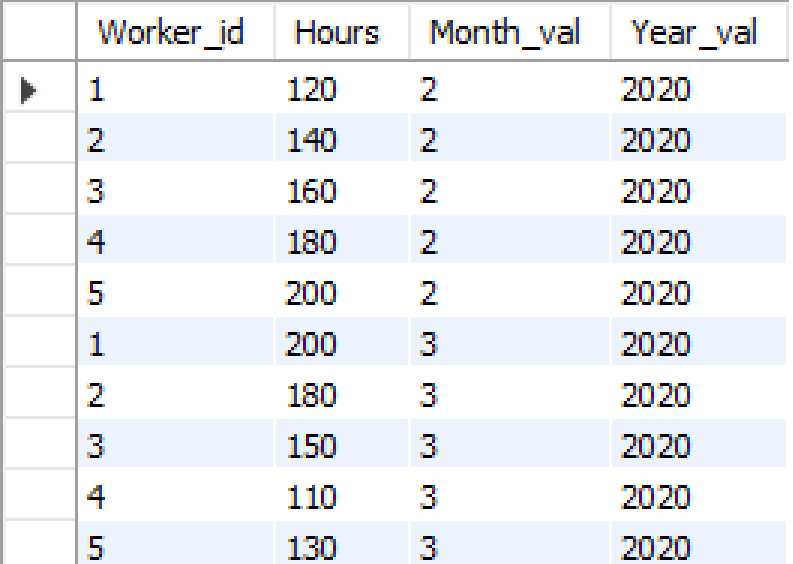
**shop\_expenses**

**store\_order**

****

**store\_worker**

**summery\_worker**

****

**7. מסמך שאילתות**

-- 1 --

Select b.Book\_Name , i.New\_Amount as Amount, i.Stored\_Location from Inventory i

inner join Book\_Copy BC on BC.Book\_id = i.Copy\_id

inner join book b on b.Book\_id = BC.Book\_id

where b.Book\_Name = "Badolina" group by i.Sequence;

-- 2 --

Select \* from Customer c order by c.Sign\_Up\_Date limit 1;

-- 3 --

select i.Copy\_id, i.Stored\_Date, b.Book\_Name from Inventory i

inner join Book\_Copy BC on BC.Book\_id = i.Copy\_id

inner join book b on b.Book\_id = BC.Book\_id order by i.Stored\_Date

limit 1;

-- 4 --

select \* from Store\_Order SO where Order\_Status != "Complete"

order by SO.Order\_Date;

-- 5 --

select sum(dd.Amount) as NUm\_Of\_Books from Deal d

inner join Deal\_Details dd on d.Deal\_id = dd.Deal\_id

inner join Book\_Copy BC on BC.Book\_id = dd.Copy\_id

inner join book b on b.Book\_id = BC.Book\_id

where b.Book\_Name = "Badolina"

group by dd.Amount;

-- 6 --

select sum(dd.Amount) as Selling\_Books , a.Author\_Name from Deal d

inner join Deal\_Details dd on d.Deal\_id = dd.Deal\_id

inner join Book\_Copy BC on dd.Copy\_id = BC.Copy\_id

inner join book b on b.Book\_id = BC.Book\_id

inner join author a on a.Author\_id = BC.Author\_id

where d.Deal\_Date >= "2000-01-01" AND d.Deal\_Date <= "2021-01-01"

group by A.Author\_Name order by sum(dd.Amount) DESC LIMIT 1;

-- 7 --

select sum(dd.Amount) as Buing\_Books, CONCAT(c.First\_Name, " ", c.Last\_Name ) as Customer\_Name from Deal d

inner join Deal\_Details dd on d.Deal\_id = dd.Deal\_id

inner join Book\_Copy BC on dd.Copy\_id = BC.Copy\_id

inner join book b on b.Book\_id = BC.Book\_id

inner join customer c on d.customer\_id = c.customer\_id

group by d.Customer\_id order by count(d.customer\_id) DESC limit 3;

-- 8 --

select count(distinct(BC.Translation\_id)) Translation\_Num, b.Book\_Name from Inventory i

inner join Book\_Copy BC on i.Copy\_id = BC.Copy\_id

inner join book b on b.Book\_id = BC.Book\_id

inner join translation t on t.Translation\_id = BC.Translation\_id

group by b.Book\_id order by count(distinct(BC.Translation\_id)) DESC limit 1;

-- 9 --

select CONCAT(c.First\_Name, " ", c.Last\_Name ) as Customer\_Name , b.book\_name, dd.sell\_price, d.Deal\_Date from Deal d

inner join Deal\_Details dd on d.Deal\_id = dd.Deal\_id

inner join Book\_Copy BC on dd.Copy\_id = BC.Copy\_id

inner join book b on b.Book\_id = BC.Book\_id

inner join customer c on d.customer\_id = c.customer\_id

where c.First\_Name = "Daniel" And c.Last\_Name = "Model"

order by d.Deal\_Date;

-- 10 --

select CONCAT(c.First\_Name, " ", c.Last\_Name ) as Customer\_Name, SO.Order\_Date, b.Book\_Name, b.Title, b.Book\_Condition, SO.Amount, So.Order\_Status from Store\_Order SO

inner join Book\_Copy BC on SO.Copy\_id = BC.Copy\_id

inner join book b on b.Book\_id = BC.Book\_id

inner join customer c on SO.customer\_id = c.customer\_id

where c.First\_Name = "Daniel" And c.Last\_Name = "Model"

order by SO.Order\_Date;

-- 11 --

select d.Deliver\_id, sum(Amount \* dd.Weight \* dt.Cost) as Delivery\_Price, CONCAT(c.First\_Name, " ", c.Last\_Name ) as Customer\_Name from delivery d

inner join Delivery\_Detail dd on d.Deliver\_id = dd.Deliver\_id

inner join Delivery\_Type dt on d.Type\_id = dt.type\_id

inner join customer c on d.customer\_id = c.customer\_id

group by d.Deliver\_id;

-- 12 --

select \* from delivery d

inner join Delivery\_Detail dd on d.Deliver\_id = dd.Deliver\_id

inner join Delivery\_Type dt on d.Type\_id = dt.type\_id

inner join customer c on d.customer\_id = c.customer\_id

where d.Deal\_id = c.First\_Name = "Anar" and c.Last\_Name = "Djafarov";

-- 13 --

select d.Deliver\_id, d.Status\_Delivery from delivery d

inner join Delivery\_Type dt on d.Type\_id = dt.type\_id

inner join customer c on d.customer\_id = c.customer\_id

where d.Deliver\_id = 1;

-- 14 --

select sum(Amount \* dd.Weight \* dt.Cost) as Delivery\_Price , dt.Type\_Delivery from delivery d

inner join Delivery\_Detail dd on d.Deliver\_id = dd.Deliver\_id

inner join Delivery\_Type dt on d.Type\_id = dt.type\_id

inner join customer c on d.customer\_id = c.customer\_id

where d.Start\_Date >= "2020-02-01" AND d.Start\_Date <= "2020-03-01"

and dt.Type\_Delivery = "Courier\_Company xpress"

group by dt.Type\_Delivery;

-- 15 --

select Sum(d.Total\_Cost) as Total\_Cost , d.Pay\_Option from Deal d

where d.Pay\_Option = "bit"

and d.Deal\_Date >= "2020-07-01" AND d.Deal\_Date <= "2020-08-01"

group by d.Pay\_Option;

-- 16 --

SELECT d.deal\_id, sum((dd.Sell\_Price - pd.price\_per\_one) \* dd.Amount) AS book\_profit FROM Deal d

inner join Deal\_Details dd on d.Deal\_id = dd.Deal\_id

inner join inventory i on dd.Copy\_id = i.Copy\_id

inner join Purchasing\_Detail pd on dd.Copy\_id = pd.Copy\_id

where d.Deal\_Date >= 12-MONTH(NOW()) AND d.Deal\_Date <= NOW()

GROUP BY d.deal\_id

HAVING (

SELECT AVG(Profit) AS average FROM

(select sum((dd.Sell\_Price - pd.price\_per\_one) \* dd.Amount)

as Profit from Deal d

inner join Deal\_Details dd on d.Deal\_id = dd.Deal\_id

inner join inventory i on dd.Copy\_id = i.Copy\_id

inner join Purchasing\_Detail pd on dd.Copy\_id =

pd.Copy\_id

inner join purchasing p on p.Purchase\_id = pd.Purchase\_id

where d.Deal\_Date >= 12-MONTH(NOW()) AND

d.Deal\_Date <= NOW()

group by d.deal\_id) AS Average\_Profit

) < book\_profit ;

-- 17 --

select count(d.Deliver\_id) Num\_Of\_Delivers, dt.Type\_Delivery from delivery d

inner join Delivery\_Detail dd on d.Deliver\_id = dd.Deliver\_id

inner join Delivery\_Type dt on d.Type\_id = dt.type\_id

where d.Start\_Date >= 12-MONTH(NOW()) AND d.Start\_Date <= NOW()

group by dt.Type\_Delivery;

-- 18 --

select count(distinct(b.book\_id)) Num\_Of\_Copys , d.Deliver\_id, b.Book\_Name , bc.Copy\_id from delivery d

inner join Delivery\_Detail dd on d.Deliver\_id = dd.Deliver\_id

inner join Delivery\_Type dt on d.Type\_id = dt.type\_id

inner join Book\_Copy BC on dd.Copy\_id = BC.Copy\_id

inner join book b on b.Book\_id = BC.Book\_id

group by d.Deliver\_id having Num\_Of\_Copys > 1

-- 19 --

select \* from Customer;

-- 20 --

select SO.Order\_id, SO.Amount, SO.Order\_Date, SO.Order\_Status, CONCAT(c.First\_Name, " ", c.Last\_Name ) as Customer\_Name from Store\_Order SO

inner join Book\_Copy BC on SO.Copy\_id = BC.Copy\_id

inner join book b on b.Book\_id = BC.Book\_id

inner join customer c on SO.customer\_id = c.customer\_id

where SO.Order\_Status = "in\_Storge"

and SO.Order\_Date >= 14-Day(NOW()) AND SO.Order\_Date <= NOW();

-- 21 --

select Sum(i.New\_Amount) as Amount, YEAR(i.Stored\_Date) AS 'year', MONTH(i.Stored\_Date) AS 'month' from inventory i

where i.Stored\_Location = "Stock"

and i.Stored\_Date >= "2020-07-01" AND i.Stored\_Date <= "2020-08-01";

-- 22 - A --

select Sum(pd.Amount) Books\_Total\_Amount, Sum(pd.Amount \* pd.Price\_Per\_One) Books\_Total\_Price from purchasing p

inner join Purchasing\_Detail pd on p.Purchase\_id = pd.Purchase\_id

where p.Purchasing\_Date >= "2020-07-01" AND p.Purchasing\_Date <= "2020-08-01";

-- 22- B--

select sum((dd.Sell\_Price - pd.price\_per\_one) \* dd.Amount) as Profit , YEAR(d.Deal\_Date) AS 'year', MONTH(d.Deal\_Date) AS 'month' from Deal d

inner join Deal\_Details dd on d.Deal\_id = dd.Deal\_id

inner join inventory i on dd.Copy\_id = i.Copy\_id

inner join Purchasing\_Detail pd on dd.Copy\_id = pd.Copy\_id

inner join purchasing p on p.Purchase\_id = pd.Purchase\_id

where d.Deal\_Date >= "2020-07-01" AND d.Deal\_Date <= "2020-08-01";

-- 23 --

select avg(d.Total\_Cost) Total\_Year\_Cost, Month(d.Deal\_Date) as 'Month', Year(d.Deal\_Date)-1 as 'From\_Year', Year(d.Deal\_Date) as 'To\_Year' from Deal d

where d.Deal\_Date >= "2019-08-01" AND d.Deal\_Date <= "2020-08-01"

GROUP BY MONTH(d.Deal\_Date);

-- 24 --

select CONCAT(W.First\_Name, " ", W.Last\_Name ) as Worker\_Name, (W.Hourly\_Salary \* SW.Hours) as Bruto\_Salary, SW.Month\_Val As Month, SW.Year\_Val As Year from store\_worker W

inner join Summery\_Worker SW on w.Worker\_id = SW.Worker\_id

where W.First\_Name = "Jhonatan" and W.Last\_Name ="Cubani"

and SW.Month\_Val = 2 and SW.Year\_Val = 2020 ;

-- 25 --

select count(d.deal\_id) Deal\_Number, CONCAT(SW.First\_Name, " ", SW.Last\_Name ) as Worker\_Name from Deal d

inner join store\_worker SW on SW.Worker\_id = d.Worker\_id

where d.Deal\_Date >= "2020-07-01" AND d.Deal\_Date <= "2020-08-01"

group by SW.Worker\_id limit 1;

1. **מסמך אלגברת יחסים**

Inventory ((Book\_Name New\_Amount as Amount) Stored\_Location) ((Inventory Book\_Copy BC on = Copy\_id) book b on Book\_id = BC.Book\_id)) ( Book\_Name = "X") p (i.Sequence))

2. coustumer ( Customer order by Sign\_Up\_Date limit 1 )

3. Inventory ((Copy\_id Stored\_Date) Book\_Name) ( (Inventory Book\_Copy BC on = Book\_id = i.Copy\_id ) (book on Book\_id = BC.Book\_id)) order by i.Stored\_Date

limit 1;

4. Store\_Order ( (Store\_Order Status != "Complete"

order by SO.Order\_Date)

5. Deal ((( (dd.Amount) as Num\_Of\_Books ) (Deal\_Details dd on d.Deal\_id =dd.Deal\_id))  
((Book\_Copy BC on BC.Book\_id = dd.Copy\_id)

(b on b.Book\_id = BC.Book\_id( b.Book\_Name = "x" (p( dd.Amount)))

6. Deal ((( (dd.Amount) as (Selling\_Books Author\_Name) ) (Deal\_Details dd on d.Deal\_id = dd.Deal\_id

)( Book\_Copy BC on dd.Copy\_id = BC.Copy\_id))(( book b on b.Book\_id = BC.Book\_id)( author a on a.Author\_id = BC.Author\_id))

( (Deal\_Date >= x ^ Deal\_Date <= y)( p(A.Author\_Name) ( ORDER BY ^ (dd.Amount) DESC

LIMIT 1;

7.. Deal ((( (dd.Amount) as (Buing\_Books CONCAT(c.First\_Name, " ", c.Last\_Name ) as Customer\_Name ) ) (Deal\_Details dd on d.Deal\_id = dd.Deal\_id)( Book\_Copy BC on dd.Copy\_id = BC.Copy\_id))(( book b on b.Book\_id = BC.Book\_id)( customer c on d.customer\_id = c.customer\_id))  
p(d.Customer\_id) ( ORDER BY ^ (d.customer\_id) DESC

LIMIT 3;

8. Inventory (COUNT(distinct(BC.Translation\_id) (( Translation\_Num b.Book\_Name) )) ((Book\_Copy BC on i.Copy\_id = BC.Copy\_id)   
(book b on b.Book\_id = BC.Book\_id) (t on t.Translation\_id = BC.Translation\_id))  
(p(b.book\_id) order by count(distinct(BC.Translation\_id)) DESC

limit 1;

9. Deal(((CONCAT(c.First\_Name, " ", c.Last\_Name ) as Customer\_Name) (b.book\_name) (dd.sell\_price) (d.Deal\_Date ))   
(Deal\_Details dd on d.Deal\_id = dd.Deal\_id)  
(Book\_Copy BC on dd.Copy\_id = BC.Copy\_id)

(book b on b.Book\_id = BC.Book\_id)  
(customer c on d.customer\_id = c.customer\_id)  
(c.First\_Name = "X" c.Last\_Name = "Y") order by d.Deal\_Date)

10. Store\_Order (((CONCAT(c.First\_Name, " ", c.Last\_Name ) as Customer\_Name) (SO.Order\_Date) (b.Book\_Name) (b.Title) (b.Book\_Condition) (SO.Amount) (So.Order\_Status))  
(Book\_Copy BC on dd.Copy\_id = BC.Copy\_id)

(book b on b.Book\_id = BC.Book\_id)  
(customer c on d.customer\_id = c.customer\_id)  
(c.First\_Name = "X" c.Last\_Name = "Y") order by SO.Order\_Date)

11. Delivery (((CONCAT(c.First\_Name, " ", c.Last\_Name ) as Customer\_Name) (sum(Amount \* dd.Weight \* dt.Cost) as Delivery\_Price)  
(Delivery\_Detail dd on d.Deliver\_id = dd.Deliver\_id)

(Delivery\_Type dt on d.Type\_id = dt.type\_id)  
(customer c on d.customer\_id = c.customer\_id)  
(c.First\_Name = "X" c.Last\_Name = "Y") order by SO.Order\_Date)

12. delivery ((Delivery\_Detail dd on d.Deliver\_id = dd.Deliver\_id   
(Delivery\_Type dt on d.Type\_id = dt.type\_id)  
(customer c on d.customer\_id = c.customer\_id))  
(c.First\_Name = "X"( c. First\_Name = "Y" c.Last\_Name = "Z")))

13. Delivery (((d.Deliver\_id) (d.Status\_Delivery))  
(Delivery\_Type dt on d.Type\_id = dt.type\_id)

(customer c on d.customer\_id = c.customer\_id)  
(d.Deliver\_id = 1))

14. . Delivery ((( (, dt.Type\_Delivery ) (sum(Amount \* dd.Weight \* dt.Cost) as Delivery\_Price)  
(Delivery\_Detail dd on d.Deliver\_id = dd.Deliver\_id)

(Delivery\_Type dt on d.Type\_id = dt.type\_id)  
(customer c on d.customer\_id = c.customer\_id)  
(d.Start\_Date >= "X" d.Start\_Date <= "Y") ((dt.Type\_Delivery = "Courier\_Company xpress") p(dt.Type\_Delivery))

15. Deal((((sum(d.Total\_Cost) as Total\_Cost) ( (d.Pay\_Option))   
(d.Pay\_Option = "bit")( d.Deal\_Date >= "X")( d.Deal\_Date <= "Y")   
p(d.Pay\_Option)

17. Delivery ((( (d.Deliver\_id) Num\_Of\_Delivers) (dt.Type\_Delivery )  
(Delivery\_Detail dd on d.Deliver\_id = dd.Deliver\_id)

(Delivery\_Type dt on d.Type\_id = dt.type\_id)  
(d.Start\_Date >= 12-MONTH(NOW()) d.Start\_Date <= NOW())  
p(dt.Type\_Delivery))

18. Delivery ((( count(distinct(b.book\_id)) Num\_Of\_Copys) (d.Deliver\_id) (b.Book\_Name) (bc.Copy\_id))  
(Delivery\_Detail dd on d.Deliver\_id = dd.Deliver\_id)

(Delivery\_Type dt on d.Type\_id = dt.type\_id)  
(Book\_Copy BC on dd.Copy\_id = BC.Copy\_id)  
(book b on b.Book\_id = BC.Book\_id)  
(d.Deliver\_id having Num\_Of\_Copys > 1))

19. Customer

20. Store\_Order (((CONCAT(c.First\_Name, " ", c.Last\_Name ) as Customer\_Name) (SO.Order\_id) (SO.Amount) (SO.Order\_Date) (SO.Order\_Status)  
(Book\_Copy BC on dd.Copy\_id = BC.Copy\_id)

(book b on b.Book\_id = BC.Book\_id)  
(customer c on d.customer\_id = c.customer\_id)  
(SO.Order\_Date >= 14-Day(NOW()) )  
( SO.Order\_Date <= NOW()))

21. Inventory ((Sum(i.New\_Amount) as Amount) (YEAR(i.Stored\_Date) AS 'year'   
(MONTH(i.Stored\_Date) AS 'month')   
 ((Book\_Copy BC on i.Copy\_id = BC.Copy\_id)   
(i.Stored\_Date >= "X" )  
( i.Stored\_Date <="Y"))

22A. Purchasing ((Sum(pd.Amount) Books\_Total\_Amount) (Sum(pd.Amount \* pd.Price\_Per\_One Books\_Total\_Price)   
 ((Purchasing\_Detail pd on p.Purchase\_id = pd.Purchase\_id)   
(p.Purchasing\_Date <= "Y" )

22b. Deal((((sum(dd.Sell\_Price - pd.price\_per\_one) \* dd.Amount) as Profit) (YEAR(d.Deal\_Date) AS 'year') ( MONTH(d.Deal\_Date) AS 'month')  
((Deal\_Details dd on d.Deal\_id = dd.Deal\_id)

(inventory i on dd.Copy\_id = i.Copy\_id)  
(Purchasing\_Detail pd on dd.Copy\_id = pd.Copy\_id)  
(purchasing p on p.Purchase\_id = pd.Purchase\_id)  
(( d.Deal\_Date >= "X")( d.Deal\_Date <= "Y")

23. Delivery ((avg(d.Total\_Cost) Total\_Year\_Cost) (Month(d.Deal\_Date) as 'Month')  
(Year(d.Deal\_Date)-1 as 'From\_Year')( Year(d.Deal\_Date) as 'To\_Year'))  
(d.Deal\_Date >="X") (d.Start\_Date <= "Y")  
p(MONTH(d.Deal\_Date))

24. store\_worker (((CONCAT(c.First\_Name, " ", c.Last\_Name ) as Worker\_Name) (W.Hourly\_Salary \* SW.Hours) as Bruto\_Salary) (SW.Month\_Val As Month) (SW.Year\_Val As Year)  
(Summery\_Worker SW on w.Worker\_id = SW.Worker\_id)  
(W.Last\_Name ="Y")  
( SW.Month\_Val = 2) SW.Year\_Val = 2020))

25. Deal (((CONCAT(SW.First\_Name, " ", SW.Last\_Name ) as Worker\_Name) (count(d.deal\_id) Deal\_Number))  
(store\_worker SW on SW.Worker\_id = d.Worker\_id)

(d.Deal\_Date >= "X" d.Deal\_Date <= "Y") p(SW.Worker\_id

limit 1))