



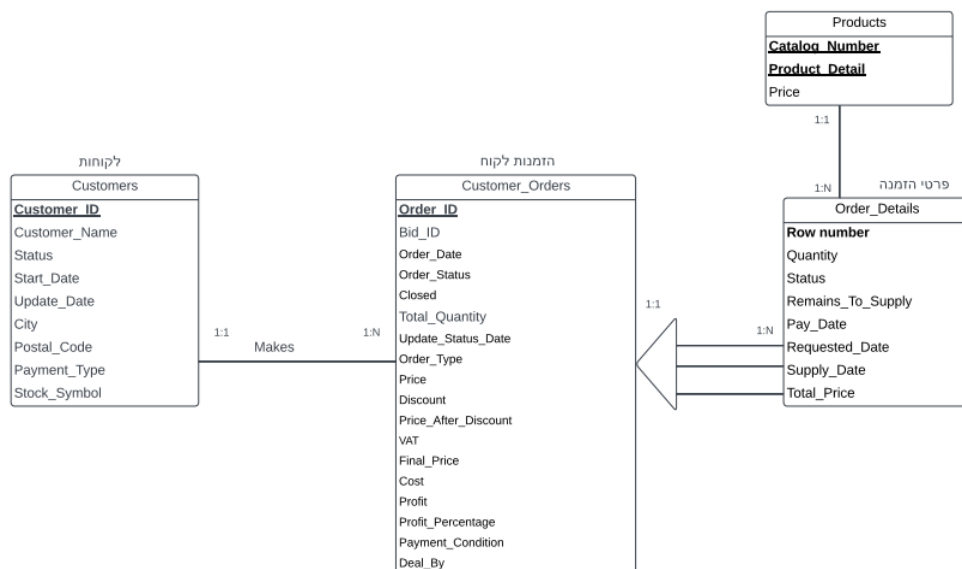
## חלק ב' – הכנת תשתית הנתונים

### 1. (20%) יישום מקורות הנתונים ומחסן הנתונים הטבלאי:

#### תיקון תרשים ERD

התיקונים שביצענו:

- מחיקת השדה Update\_Date מטבלת Order\_Details.
- הוספת הסברים למפתחות זרים בטבלה.
- הוספת קטגוריות לשדה STATUS.
- הוספת שדה למפתח ראשי PRODUCTS



#### תיקון בסיס נתונים ראשי: (הודגשו התיקונים בצהוב).

הטבלה	שם השדה	סוג הנתונים	סולם המדידה	תיאור השדה
CUSTOMERS רשימת לקוחות- רשימת הלקוחות אשר רושמים במפעל	<u>Customer_ID</u>	Varchar(20)	שמי	מזהה הלקוח
	Customer_Name	Varchar(60)	שמי	שם הלקוח
	Status	Varchar(20)	שמי	סטאטוס הלקוח
	Start_Date	Date	מרווח	תאריך רישום הלקוח במערכת
	Update_Date	Date	מרווח	תאריך הזנה/ עדכון
	City	Varchar(20)	שמי	עיר הלקוח
	Postal_Code	Varchar(20)	שמי	מיקוד הלקוח
	Payment_Type	Varchar(40)	שמי	אופן תשלום
CUSTOMER_ORDERS הזמנות	<u>Order_ID</u>	Varchar(20)	שמי	מזהה ההזמנה
	Bid_ID	Varchar(30)	שמי	מזהה הצעת המחיר
	Customer_ID (CUSTOMERS)	Varchar(20)	שמי	מזהה הלקוח. מפתח זר- מצביע על מפתח ראשי בטבלת CUSTOMERS. מראה את הקישוריות בין לקוח להזמנות שביצע.
	Order_Date	Date	מרווח	תאריך ההזמנה
	Order_Status	Varchar(40)	שמי	סטטוס ההזמנה
	Closed	Bit	שמי	אינדיקטור האם ההזמנה נסגרה
	Total_Quantity	Int	יחס	כמות בהזמנה



תאריך הזנה/ עדכון	מרווח	Date	Update_Status_Date	
סוג הזמנה (TK/KBM)	שמי	Varchar(20)	Order_Type	
מחיר ההזמנה ללא הנחה וללא מע"מ	יחס	Money	Price	
אחוז הנחה	יחס	Float	Discount	
מחיר לאחר הנחה	יחס	Money	Price_After_Discount	
מע"מ	יחס	Money	VAT	
מחיר סופי	יחס	Money	Final_Price	
עלות ההזמנה למפעל	יחס	Money	Cost	
רווח המפעל מההזמנה	יחס	Money	Profit	
אחוז רווח מההזמנה	יחס	Float	Profit_Percentage	
אופן התשלום (שוטף/60, שוטף/90)	שמי	Varchar(20)	Payment_Condition	
מחלקה מטפלת	שמי	Varchar(20)	Deal_By	
מזהה ההזמנה. מפתח זר- מצביע על מפתח ראשי בטבלת CUSTOMER_ORDERS. מראה את הקישוריות בין פרטי ההזמנה להזמנה עצמה.	שמי	Varchar(20)	Order_ID (CUSTOMER_ORDERS)	ORDER_DETAILS פרטי ההזמנה
שורת פריט	סדר	Int	Row_Number -----	
מק"ט הרכיב. מפתח זר- מצביע על מפתח ראשי בטבלת PRODUCTS. מראה את הקישוריות בין פרטי ההזמנה למוצר עצמו.	שמי	Varchar(60)	Catalog_Number (PRODUCTS)	
מק"ט הרכיב. מפתח זר- מצביע על מפתח ראשי בטבלת PRODUCTS. מראה את הקישוריות בין פרטי ההזמנה למוצר עצמו.	שמי	Varchar(100)	Product_Detail(PRODUCTS)	
כמות הרכיבים הזחים	יחס	Int	Quantity	
סטטוס הרכיב. קטגוריות: שולמה, מאושרת לביצוע, ביצוע רכש, בביצוע, לאישור הנדסה, טיוטא, לאישור תפי.	שמי	Varchar(60)	Status	
כמות יחידות שנשארו לספק	יחס	Int	Remains_To_Supply	
תאריך התשלום	מרווח	Date	Pay_Date	
תאריך המבוקש על ידי הלקוח	מרווח	Date	Requested_Date	
תאריך אספקה בפועל	מרווח	Date	Supply_Date	
תאריך הזנה/ עדכון	מרווח	Date	Update_Date	
מק"ט הרכיב	שמי	Varchar(60)	Catalog_Number	Products טבלת מוצרים
תיאור המוצר	שמי	Varchar(100)	Product_Detail	
מחיר יחידה	יחס	Money	Price	

תיקון מחסן נתונים: (הודגשו התיקונים בצהוב).

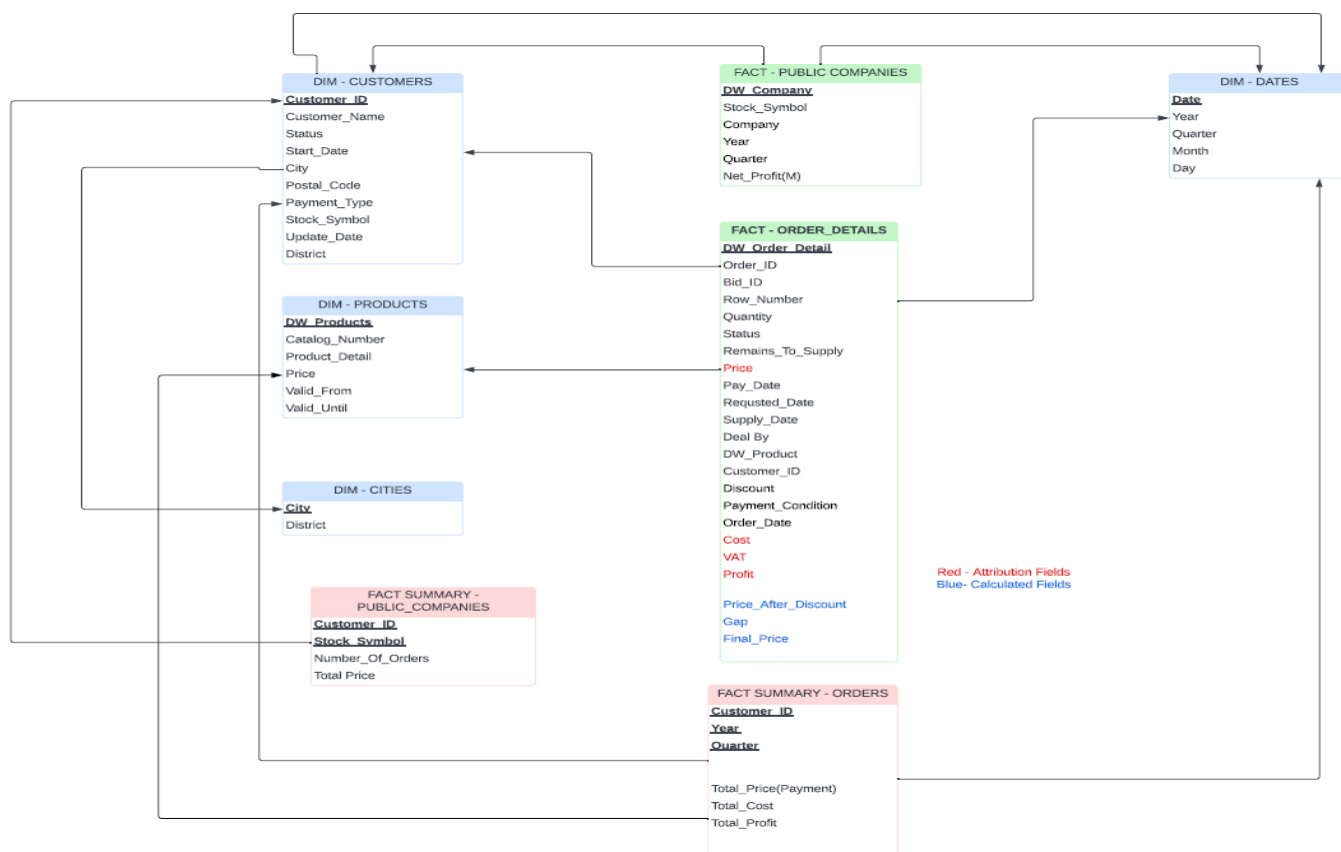


הטבלה	שם השדה	סוג הנתונים	תיאור השדה
<b>DIM - CUSTOMERS</b> טבלת ממד משתנה מסוג 1, המכילה את רשימות הלקוחות, אף שדה לא עתיד להשתנות בקצב תדירות יחסית.	DW_Customer	Varchar(30)	מזהה לקוח מפתח עזר, כל ערך משקף גרסת לקוח בעלת טווח תאריכים מוגדר
	Customer_ID	Varchar(20)	מזהה הלקוח
	Customer_Name	Varchar(60)	שם הלקוח
	Status	Varchar(20)	מציג את פעילות הלקוח בחברה (פעיל/לא פעיל/זמני)
	Start_Date(DIM-DATES)	Date	תאריך רישום הלקוח במערכת
	City	Varchar(20)	עיר הלקוח
	Postal_Code	Varchar(20)	מיקוד
	Payment_Type	Varchar(20)	אופן ביצוע התשלום
	Stock_Symbol	Varchar(20)	סימול החברה בשוק ההון
	City	Varchar(20)	מזהה עיר
	District	Varchar(20)	משתנה קטגוריאלי מסוג איזור בארץ <ul style="list-style-type: none"> <li>צפון</li> <li>מרכז</li> <li>דרום</li> </ul>
<b>FACT - ORDER DETAILS</b> פריט בהזמנה (שנינו את טבלת העובדה שלנו ולכן הוספנו את כולה)	DW_Order_Detail	Int	מפתח עזר לזיהוי פריט בהזמנה
	Order_ID	Varchar(20)	מזהה הזמנה
	Bid_ID	Varchar(30)	מזהה הצעת מחיר
	Row_Number	Int	מזהה סוג פריט בהזמנה (בהזמנה יכולים להיות מספר פריטים)
	Quantity	Int	מספר פריטים בהזמנה
	Status	Varchar(20)	סטטוס הרכיב. קטגוריות: <ul style="list-style-type: none"> <li>שולמה</li> <li>מאושרת לביצוע</li> <li>ביצוע רכש</li> <li>בביצוע</li> <li>לאישור הנדסה</li> <li>טיוטא</li> <li>לאישור תפ"</li> </ul>
	Remains_To_Supply	Int	כמות פריטים שנשאר לספק
	Price	Money	עלות פריט יחיד
	Pay_Date	Date	תאריך תשלום
	Requested_Date	Date	תאריך מבוקש לאספקה על ידי הלקוח
	Supply_Date	Date	תאריך אספקה בפועל
	Deal_By	Varchar(20)	מטופל על ידי
	DW_Product	Int	מפתח עזר לזיהוי הפריט
	Customer_ID	Varchar(20)	מזהה לקוח (אשר ביצע את ההזמנה)
	Discount	Float	אחוז הנחה לפריט
	Payment_Condition	Varchar(20)	אופן התשלום (ש60, ש45, ש30)
	VAT	Money	מע"מ לפריט
	Profit	Money	כמות הרווח מהפריט



עלות תשלום למפעל על הפריט	Money	Cost
מחיר הפריט לאחר הנחה	Money	Price_After_Discount
שיעור המרווח בין זמן האספקה הרצוי של הלקוח לזמן אספקה בפועל	Int	Gap
מחיר הפריט לאחר הנחה ולאחר תוספת מע"מ	Money	Final_Price

תרשים כוכב מתוקן:



התיקונים שביצענו:

- הפיכת הממד CUSTOMERS למשתנה מסוג לאט 1.
- סידור החיצים בתרשים.
- בהמלצת אדיר, נעבוד עם לקוחות מהארץ, לכן החלטנו לשנות את השדה Country לשדה District - (איזור צפון, מרכז, דרום).
- החלפת טבלת העובדה לטבלת עובדה של פריט בהזמנה (Order\_Details).

## 2. (20%) אפיון תהליכי ETL



סדר הרצת השלבים:

1. Data Mirroring (MRR) - שכפול נתוני המקור אל אזור השיקוף



מטרתו של שלב זה הינה שכפול מקור הנתונים לאזור השיקוף לצורך שמירת תקינות נתוני המקור. שלב מקדים לשלב זה הוא יצירת בסיסי נתונים המכילים את נתוני המקור.

## 2. Dimension Staging (STG-DIMS) - הכנת נתוני המימד

מטרת שלב זה הינה ביצוע טרנספורמציות לטבלאות המימד לצורך העברת הנתונים ממבנה טבלאי לסכמת כוכב. כצעד מקדים נבצע השוואה ומחיקה של כל הנתונים שכבר מוזנים לתוך ה-DW, כך שהנתונים אשר נמצאים בטבלאות באותו הרגע הינם אך ורק נתונים חדשים או כאלה שהשתנו.

## 3. Dimension Warehousing (DW-DIMS) - עדכון טבלאות המימד ב-DW

מטרתו של שלב זה הינו העברת הממדים אל ה-DW מתוך שלב ה-STG, נבצע זאת באמצעות פעולת Upsert אשר תבצע עדכון רשומות קיימות והוספת רשומות שאינן קיימות עבור ממד משתנה בזמן מסוג 1. עבור משתנה ממד מסוג 2, נבצע עדכון של תאריך סיום שינוי גרסה והוספת רשומה חדשה. נשתמש בפעולה זו על מנת לחסוך זמן טעינה. נדגיש כי לא נבצע מחיקת נתונים מה-DW מכיוון שאנו מעוניינים בשמירת רשומות היסטוריות.

## 4. Fact Staging (STG – FACT) - הכנת נתוני העובדה

מטרת שלב זה הינה ביצוע טרנספורמציות על טבלאות העובדה לצורך העברת הנתונים ממבנה טבלאי לסכמת כוכב. שלב זה מתבצע באמצעות טבלאות השיקוף מהשלב הראשון ובאמצעות טבלאות המימד מהשלב השלישי.

## 5. Referential Integrity (RI) - בדיקת יושרת הנתונים

מטרת שלב זה הינה בדיקת תקינות הנתונים בטבלאות העובדה שיצרנו בשלב הקודם אל מול הנתונים בטבלאות המימד שהוכנסו ל-DW בשלב השלישי. שלב זה חשוב לבדיקת אמינות המקורות והנתונים.

## 6. Fact Warehousing (DW – FACT) - עדכון טבלאות העובדה ב-DW

מטרת שלב זה הינה העברת טבלאות העובדה אל ה-DW משלב ה-STG לאחר התיקונים שבוצעו בשלב הקודם. שלב זה הינו השלב הסופי לתהליך ה-ETL

### תלויות הקיימות בין שלבי התהליך:

- טבלאות המימד צריכות להיטען לפני טבלאות העובדה בשל אילוצי מפתח זר לטבלאות המימד. שלב אימות המפתחות הזרים (RI) חייב להתבצע אחרי שלב ה-Fact Staging ולפי Fact Warehousing מכיוון שלא נרצה להזמין ל-DW נתונים אשר לא נבדקו או שלא עברו את שלב ה-Staging כראוי.
- התלות הראשונה הינה בסדר טעינת הטבלאות, ראשית נטען את טבלאות המימד ורק לאחר מכן את טבלאות העובדה, וזאת בשל אילוצי מפתחות זרים שיוכלו להיטען רק לאחר טבלאות המימד השונות.

בנוסף, קיימת תלות בשלב האימות, כאשר לפי ההזנה אל ה-DW נרצה לוודא ששלב ה-Staging עבר בצורה חלקה.

### שלבי ה-ETL שצפויים לקחת זמן רב במיוחד:

1. שלב השיקוף (MRR) - שלב זה ייקח זמן רב מאחר והוא מתבצע על כלל הנתונים אשר נמצאים במקורות הנתונים (בעיקר טבלאות עובדה).
2. שלב הכנת נתוני העובדה (STG-FACT) - שלב זה צפוי לקחת זמן רב עקבות פעולות JOIN ותחשיבי אגרגציה בין הטבלאות השונות.
3. שלב בדיקת יושרת הנתונים (RI) - השלב בו נבדוק את יושרת הנתונים צפוי לקחת זמן רב מאחר והתהליך יכלול מעבר על כלל המפתחות הזרים בנתוני טבלאות העובדה בהשוואה לטבלאות המימד.
4. בשלב עדכון טבלאות העובדה במחסן הנתונים - נטען את על הטבלאות בנוסף לנתונים החדשים שנוספו,

שלבים בתהליך אשר עלולים להיכשל וכיצד נתמודד איתם:

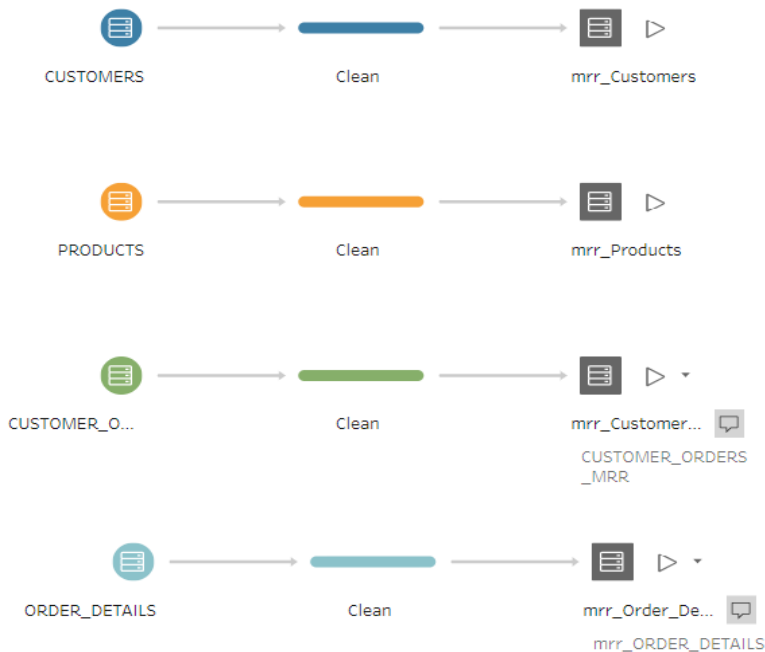


1. שלב ההכנת נתוני המימד (Dimension Staging) – בשלב זה ישנה חשיבות כיצד תתבצע ההיררכיה. כשלים בשלב זה יכולים להופיע בתצורה של ספירה כפולה לטובת שדות חישוביים או שרומות לא מתאימות אשר יתווספו לתהליך.
2. שלב בדיקת יושרת הנתונים (RI) - כחלק מתהליך זה עשויות להופיע שגיאות עקב רשומות שעבורן לא תמצא התאמה של מפתח בין טבלאות העבודה לטבלאות המימד. על מנת להימנע משגיאות אלו ניתן לבצע מספר פעולות:
  - סימון הרשומות הלא תקינות על ידי ביטוי מוסכם, או יצירת טבלה אשר אליה נשלח את הרשומות הלא תקינות.
  - מחיקת הנתונים שמייצרים חוסר תאימות בין הטבלאות השונות. (אך יש לקחת בחשבון את פוטנציאל ההטיה בשלב ניתוח הנתונים).

### 3. (40%) מימוש תהליכי ה-ETL – יישומי הבסיס

#### שלב 1 – Data Mirroring

#### דוגמה ל TRUNCATE לממדים

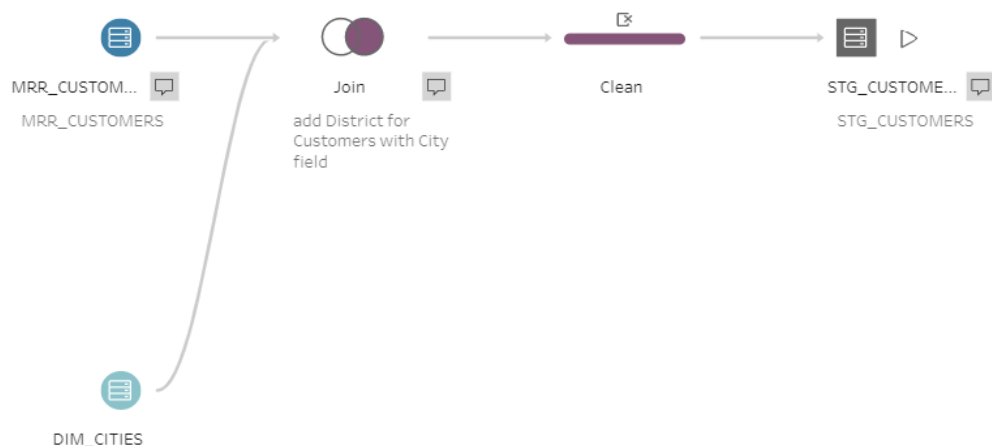


Settings	Custom SQL
Before	After
Optional: Execute this SQL before writing flow data to database table.	
<pre>TRUNCATE TABLE MAS_mrr.dbo.MRR_PRODUCTS</pre>	
Run Flow	

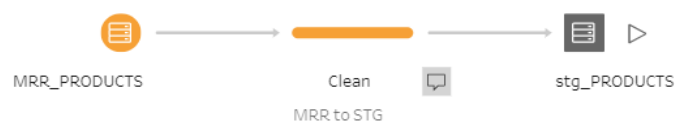
טעינה אינקרימנטלית  
לטרנזקציות



## שלב 2 – Dimensions Staging

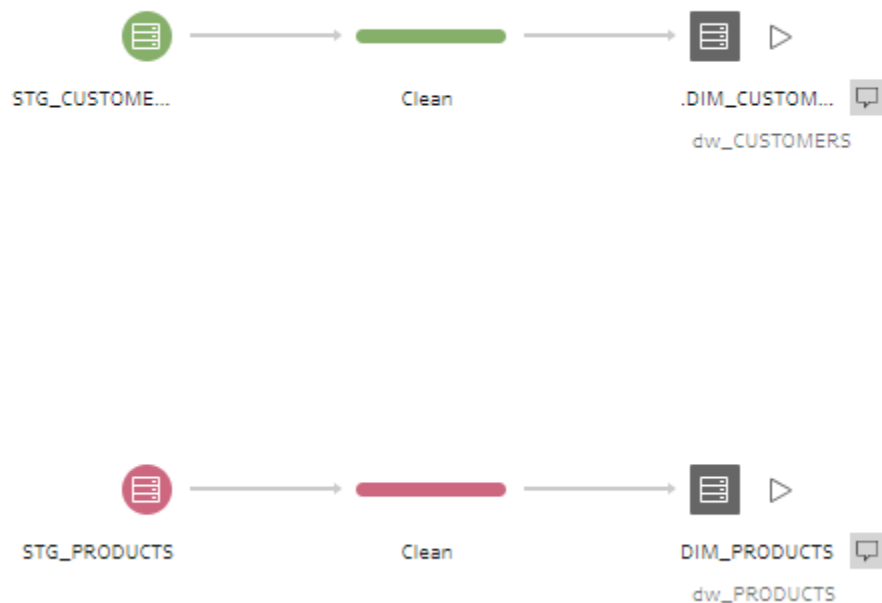


Settings	Custom SQL
Before	After
Optional: Execute this SQL before writing flow data to database table.	
<pre>TRUNCATE TABLE MAS_etg.dbo.STG_CUSTOMERS</pre>	
<button>Run Flow</button>	

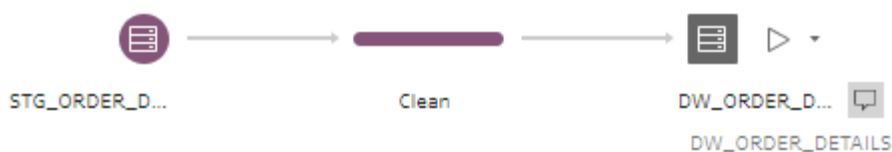


## שלב 3 – Dimensions Loading

### דוגמה לTRUNCATE לממדים



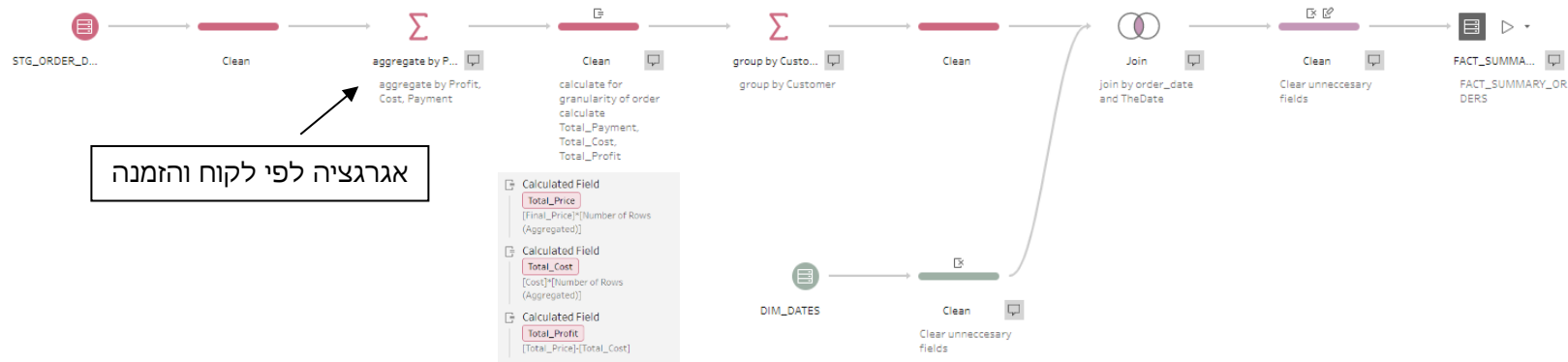
Settings	Custom SQL
Before	After
Optional: Execute this SQL before writing flow data to database table.	
<pre>TRUNCATE TABLE MAS_dw.dbo.DIM_PRODUCTS</pre>	
<button>Run Flow</button>	







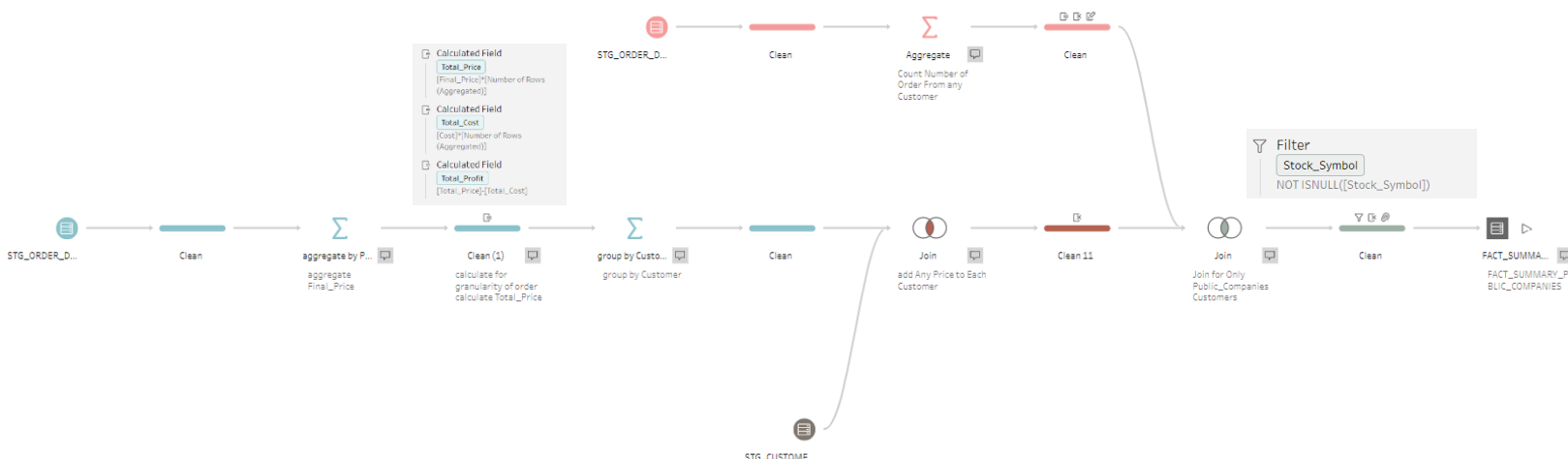
טבלת סיכומית **FACT SUMMARY ORDERS** - מסכמת עבור כל לקוח בכל שנה ורבעון את הכמות הוצאות, הכנסות והרווח הכולל של החברה.



אגרגציה לפי לקוח והזמנה

טבלת סיכומית **FACT SUMMARY PUBLIC COMPANIES**

מסכמת עבור כל לקוח שהינו חברה ציבורית את מספר ההזמנות ואת כמות ההכנסות של הארגון שלנו לאורך כל זמן העבודה מול אותו הלקוח.



\*\* את הבסיס נתונים המשני שלנו יצרנו ישירות ב-DW מכיוון שלא ביצענו עליו שום טרנספורמציה.  
\*\* את הממדים DATES ו-CITIES גם בחרנו להכניס ישירות ל-DW.

#### 4. (צבירה של עד 25%) מימוש תהליכי ה-ETL – יישומים מתקדמים

##### א. (15%) מימד משתנה לאט מסוג 2

ביצענו המרה של משתנה המימד DIM\_PRODUCTS למשתנה מימד מסוג 2, אשר שומר את כלל גרסאות המוצר. בחרנו ליישם זאת משום שאנחנו סבורים שמחירי המוצרים עשויים להשתנות עם הזמן, נרצה לבצע ניתוחים בזמנים שונים ולכן החשיבות מבחינתנו לשמור את כלל הגרסאות.  
**צעד 1:** Left-join בין טבלאות ה-Products במחסן ובשלב ה-STG, כאשר הטבלה של ה-STG בצד שמאל. ביצענו זאת כדי לבדוק האם יש שינוי ברשומות ו/או רשומות חדשות.  
ישנם שני מקרים:

מקרה א' – המוצרים הינם מוצרים חדשים לגמרי ונרצה להוסיף אותם לרשימת המוצרים במחסן.  
נבדוק זאת על ידי הוספת סינון [1] שמטרתו לסנן את הרשומות החדשות שקיימות ב-STG ולא ב-DW, תנאי הסינון יהיה  $DW\_Product = NULL$  (של המחסן). בנוסף נעדכן את השדה Valid\_From להיות מעודכן לתאריך של היום.

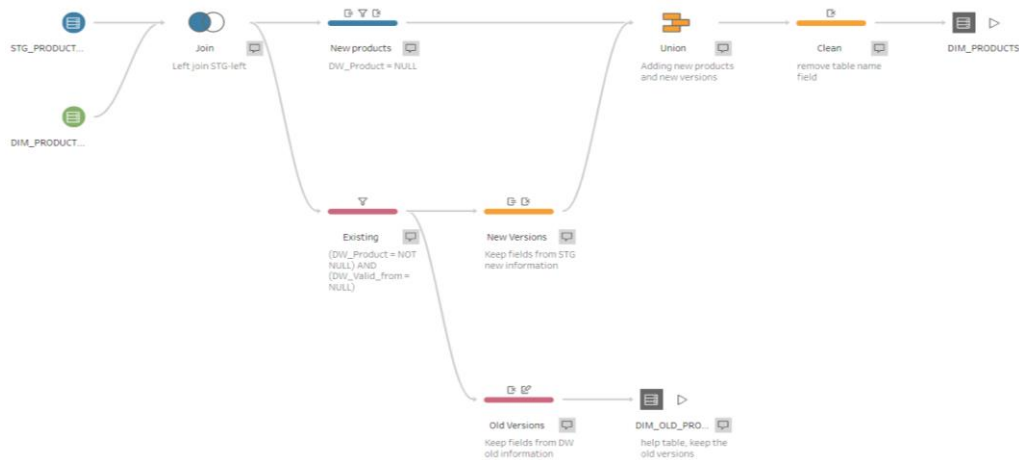


מקרה ב' – נרצה את יתר הרשומות של אופציה א'. נמצא אותם על ידי תנאי הפוך (NOT NULL) ובנוסף נרצה גם רק את הרשומות שבוצע שינוי במחיר שלהם על ידי בדיקת תנאי [2]. סינון זה יפיק עבורנו את המוצרים שקיימים כרגע במערכת ועדכנו את מחירם.

נרצה לבצע שני תהליכים:

1. לשמור את הגרסה הישנה (עם המחיר הישן) במערכת ולעדכן עבורו את שדה ה-Valid\_Until להיות שווה לתאריך של היום. כדי לעשות זאת, נשתמש בטבלת עזר DIM\_OLD\_PRODUCTS ופרוצדורה שמורה [3] שמטרתה לעדכן את התאריך של היום בשדה Valid\_Until עבור המוצרים שבוצע עבורם שינוי במחיר בטבלה המקורית במחסן.

2. עבור הרשומה עם המחיר החדש – נעדכן את השדה Valid\_From לתאריך של היום. לבסוף, נבצע איחוד בין האופציות (מוצר חדש לגמרי או מוצר קיים עם מחיר חדש) ונקניס לטבלה DIM\_PRODUCTS במחסן הנתונים.

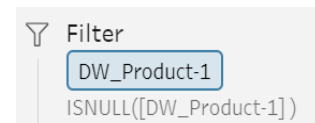


-[3]

-[2]

-[1]

```
CREATE PROCEDURE dbo.Update_Old_Products
AS BEGIN
UPDATE MAS_dw.dbo.DIM_PRODUCTS
SET MAS_dw.dbo.DIM_PRODUCTS.Valid_Until=GETDATE()
WHERE DW_Product IN (SELECT DW_Product
FROM MAS_dw.dbo.DIM_OLD_PRODUCTS)
TRUNCATE TABLE MAS_dw.dbo.DIM_OLD_PRODUCTS
END
```





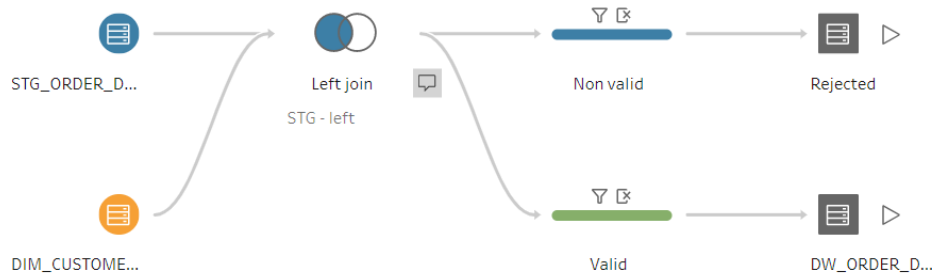
**צעד 2** – עדכון טבלת העובדה הפרטנית – כדי שהנתונים בטבלת העובדה הפרטנית DW\_ORDER\_DETAILS יישקפו נכון את מחירי המוצרים הרלוונטיים לאותו הזמן, הוספנו פילטר בשלב STG\_FACT. מטרת הפילטר היא סינון רשומות המוצר הרלוונטיות לאחר ביצוע ה-join עם טבלת המוצרים מהמחסן.



### ג. (10%) בדיקת "ישרת היחס" (RI - Referential Integrity) של נתוני טבלאות העובדה

בחרנו לעשות את בדיקת "ישרת הנתונים" עבור הכנסת רשומה של הזמנה שאינה תקינה. הגדרנו שאם ההזמנה מבוצעת על ידי לקוח שאינו קיים אזי ההזמנה אינה תקינה.

**מימוש גישה א':** רשומות לא תקינות לא יכנסו לטבלת העובדה DW\_ORDER\_DETAILS אלא יכנסו לטבלת REJECTED. תחילה נבצע Left join בין הטבלאות STG\_ORDER\_DETAILS ו-IM\_CUSTOMERS. כאשר טבלת ה-STG מכילת שדות NULL (לא קיים לקוח במערכת בלי שם חברה). הלקוחות, נוכל לזהות זאת על ידי כך ששםם יהיה NULL (לא קיים לקוח במערכת בלי שם חברה). אם השם קיים ההזמנה תוגדר כ"תקינה" ותכנס לטבלת DW\_ORDER\_DETAILS, במידה ולא תכנס לטבלת REJECTED.



### טבלת Rejected כפי שמופיעה בTableau:

Rejected 23 fields

Save output to: Database

Settings: Custom SQL

Table: LAPTOP-JFC9FF60\SQLEXPRESS, Database: MAS\_dw, Table: DW\_REJECTED

Write Options: Select an option to create or update

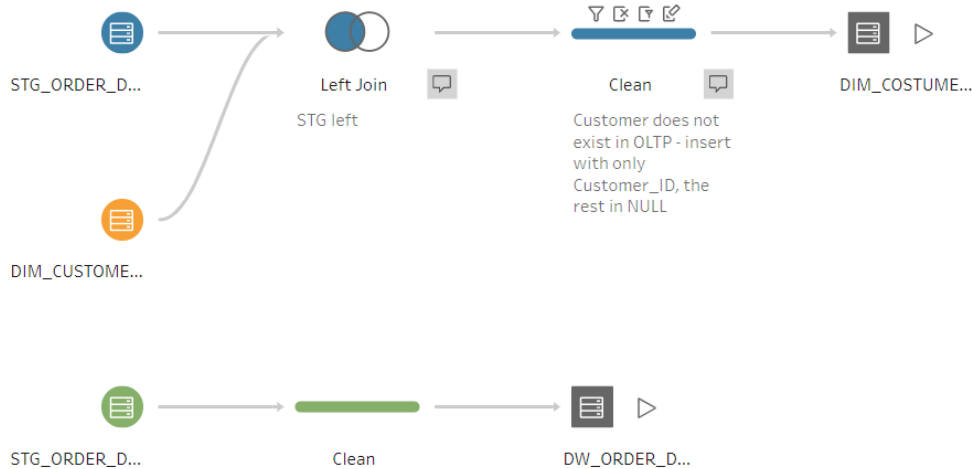
Run Flow

Write 23 of 23 fields to the table "DW\_REJECTED" in LAPTOP-JFC9FF60\SQLEXPRESS

w_Number	Quantity	Status	Remains_To_Supply	Price	Pay_Date	Requested_Date	Supply_Date	Deal_By	DW_Product	Customer_ID
200		מאושרת לביצוע	6	41,550	02/29/2024	06/01/2023	12/30/2023	basilj	1.039	98808



**מימוש גישה ב':** רשומות לא תקינות יכנסו לטבלת DW\_ORDER\_DETAILS, נוכל לזהות אותן על ידי תנאי של join עם טבלת הלקוחות. תחילה נבצע Left Join בין הטבלאות STG\_ORDER\_DETAILS ו-IM\_CUSTOMERS. כאשר טבלת ה- STG מציג שמאל. כאשר גם בגישה זו נבדוק עבור איזה רשומות שם הלקוח הוא NULL, עבור אותן רשומות נכניס את הלקוח כשם NULL פרט למזהה Customer\_ID. את כלל הרשומות בטבלת STG\_ORDER\_DETAILS נכניס לטבלת DW\_ORDER\_DETAILS. כאשר נרצה לתחקר את הרשומות שאינן תקינות נבצע join עם תנאי שכלל השדות הינן NULL, תוצאת הפלט תביא לנו את הרשומות שאינן תקינות.



על מנת לתחקר את הרשומות הלא תקינות, נבצע JOIN טבלת הממד הלקוחות מהמחסן וטבלת פריטי ההזמנות. לאחר מכן נסנן רק את הרשומות עבורן כלל השדות של אותו לקוח הינן NULL.

### צילום מסך מתוכנת Tableau:

The screenshot shows the Tableau interface. At the top, there's a 'Clean' step with 33 fields and 3 rows. Below it, a table is displayed with columns: Customer\_ID-1, Customer\_Name, Status-1, Start\_Date, and Up. The table contains three rows of data, all with null values for Customer\_Name, Status-1, and Start\_Date. A filter is applied to Customer\_Name, showing the expression ISNULL([Customer\_Name]).

Customer_ID-1	Customer_Name	Status-1	Start_Date	Up
9999	null	null	null	null
9999	null	null	null	null
88888	null	null	null	null



## נספחים

קוד SQL עבור בסיס נתונים תפעולי:

-----OLTP-----

```
CREATE TABLE MAS.dbo.CUSTOMERS (  
Customer_ID Varchar(20) not null,  
Customer_Name Varchar(60),  
[Status] Varchar(20),  
[Start_Date] Date,  
[Update_Date] Date,  
City Varchar(20),  
Postal_Code Varchar(20),  
Payment_Type Varchar(40),  
Stock_Symbol Varchar(20),  
CONSTRAINT PK_CUSTOMERS PRIMARY KEY (Customer_ID)  
)
```

```
CREATE TABLE MAS.dbo.CUSTOMER_ORDERS (  
Order_ID Varchar(20) not null,  
Bid_ID Varchar(30),  
Customer_ID Varchar(20),  
[Order_Date] Date,  
Order_Status Varchar(40),  
Closed Bit,  
Total_Quantity Int,  
Update_Status_Date Date,  
Order_Type Varchar(20),  
Price Money,  
Discount Float,  
Price_After_Discount Money,  
VAT Money,  
Final_Price Money,  
Cost Money,  
Profit Money,  
Profit_Percentage Float,  
Payment_Condition Varchar(20),  
Deal_By Varchar(20),  
CONSTRAINT PK_CUSTOMER_ORDERS PRIMARY KEY (Order_ID),  
CONSTRAINT FK_CUSTOMER_ORDERS FOREIGN KEY (Customer_ID)  
REFERENCES CUSTOMERS (Customer_ID)  
)
```

```
CREATE TABLE MAS.dbo.PRODUCTS (  
Catalog_Number Varchar(60) not null,  
Product_Detail Varchar(100) not null,  
Price Money,  
CONSTRAINT PK_PRODUCTS PRIMARY KEY (Catalog_Number,Product_Detail)  
)
```

```
CREATE TABLE MAS.dbo.ORDER_DETAILS (  
Order_ID Varchar(20) not null,  
[Row_Number] Int,  
Catalog_Number Varchar(60),  
Product_Detail Varchar(100),  
Quantity Int,  
[Status] Varchar(60),
```



```
Remains_To_Supply Int,  
Pay_Date Date,  
Requested_Date Date,  
Supply_Date Date,  
Total_Price Money,  
CONSTRAINT PK_ORDER_DETAILS PRIMARY KEY (Order_ID, [Row_Number]),  
CONSTRAINT FK_ORDER_DETAILS1 FOREIGN KEY (Order_ID)  
REFERENCES CUSTOMER_ORDERS (Order_ID),  
CONSTRAINT FK_ORDER_DETAILS2 FOREIGN KEY (Catalog_Number, Product_Detail)  
REFERENCES PRODUCTS (Catalog_Number, Product_Detail)  
)
```

-----DROPS-----

```
DROP TABLE MAS.dbo.ORDER_DETAILS  
DROP TABLE MAS.dbo.PRODUCTS  
DROP TABLE MAS.dbo.CUSTOMER_ORDERS  
DROP TABLE MAS.dbo.CUSTOMERS
```

קוד עבור שלב Mirroring:

-----MRR-----

```
--TRUNCATE TABLE MAS_mrr.dbo.MRR_CUSTOMERS  
CREATE TABLE MAS_mrr.dbo.MRR_CUSTOMERS (  
Customer_ID Varchar(20) not null,  
Customer_Name Varchar(60),  
[Status] Varchar(20),  
[Start_Date] Date,  
[Update_Date] Date,  
City Varchar(20),  
Postal_Code Varchar(20),  
Payment_Type Varchar(40),  
Stock_Symbol Varchar(20),  
CONSTRAINT PK_CUSTOMERS PRIMARY KEY (Customer_ID)  
)
```

```
--TRUNCATE TABLE MAS_mrr.dbo.MRR_CUSTOMER_ORDERS  
CREATE TABLE MAS_mrr.dbo.MRR_CUSTOMER_ORDERS (  
Order_ID Varchar(20) not null,  
Bid_ID Varchar(30),  
Customer_ID Varchar(20),  
[Order_Date] Date,  
Order_Status Varchar(40),  
Closed Bit,  
Total_Quantity Int,  
Update_Status_Date Date,  
Order_Type Varchar(20),  
Price Money,  
Discount Float,  
Price_After_Discount Money,  
VAT Money,  
Final_Price Money,  
Cost Money,  
Profit Money,  
Profit_Percentage Float,  
Payment_Condition Varchar(20),  
Deal_By Varchar(20),  
CONSTRAINT PK_CUSTOMER_ORDERS PRIMARY KEY (Order_ID)  
)
```



```
--TRUNCATE TABLE MAS_mrr.dbo.MRR_PRODUCTS
CREATE TABLE MAS_mrr.dbo.MRR_PRODUCTS (
  Catalog_Number Varchar(60) not null,
  Product_Detail Varchar(100) not null,
  Price Money,
  CONSTRAINT PK_PRODUCTS PRIMARY KEY (Catalog_Number,Product_Detail)
)
```

```
--TRUNCATE TABLE MAS_mrr.dbo.MRR_ORDER_DETAILS
CREATE TABLE MAS_mrr.dbo.MRR_ORDER_DETAILS (
  Order_ID Varchar(20) not null,
  [Row_Number] Int,
  Catalog_Number Varchar(60),
  Product_Detail Varchar(100),
  Quantity Int,
  [Status] Varchar(60),
  Remains_To_Supply Int,
  Pay_Date Date,
  Requested_Date Date,
  Supply_Date Date,
  Total_Price Money,
  CONSTRAINT PK_ORDER_DETAILS PRIMARY KEY (Order_ID, [Row_Number])
)
```

```
-----DROPS-----
DROP TABLE MAS_mrr.dbo.MRR_ORDER_DETAILS
DROP TABLE MAS_mrr.dbo.MRR_PRODUCTS
DROP TABLE MAS_mrr.dbo.MRR_CUSTOMER_ORDERS
DROP TABLE MAS_mrr.dbo.MRR_CUSTOMERS
```

קוד עבור שלב Staging:

```
-----STG-----
```

```
--TRUNCATE TABLE MAS_stg.dbo.STG_CUSTOMERS
CREATE TABLE MAS_stg.dbo.STG_CUSTOMERS (
  Customer_ID Varchar(20),
  Customer_Name Varchar(60),
  [Status] Varchar(20),
  [Start_Date] Date,
  [Update_Date] Date,
  City Varchar(20),
  Postal_Code Varchar(20),
  Payment_Type Varchar(40),
  Stock_Symbol Varchar(20),
  District Varchar(20)
)
```

```
--TRUNCATE TABLE MAS_stg.dbo.STG_PRODUCTS
CREATE TABLE MAS_stg.dbo.STG_PRODUCTS (
  DW_Product Int identity(1,1),
  Catalog_Number Varchar(60),
  Product_Detail Varchar(100),
  Price Money,
)
```

```
--TRUNCATE TABLE MAS_stg.dbo.STG_ORDER_DETAILS
CREATE TABLE MAS_stg.dbo.STG_ORDER_DETAILS ( /*פרטני למוצר*/
```



```
DW_Order_Detail Int identity(1,1),
Order_ID Varchar(20),
Bid_ID Varchar(30),
[Row_Number] Int,
Quantity Int,
[Status] Varchar(60),
Remains_To_Supply Int,
Price Money,
Pay_Date Date,
Requested_Date Date,
Supply_Date Date,
Deal_By Varchar(20),
DW_Product Int,
Customer_ID Varchar(20),
Discount Float,

Payment_Condition Varchar(20),
VAT Money,
Profit Money,
Cost Money,

Price_After_Discount Money,
Gap Int, /*(Requested_Date-Supply_Date)*/
Final_Price Money
)
```

```
-----DROPS-----
DROP TABLE MAS_stg.dbo.STG_ORDER_DETAILS
DROP TABLE MAS_stg.dbo.STG_PRODUCTS
DROP TABLE MAS_stg.dbo.STG_CUSTOMERS
```

קוד עבור מחסן הנתונים:

```
-----DW-----

--DROP          TABLE MAS_dw.dbo.DIM_CUSTOMERS
--TRUNCATE TABLE MAS_dw.dbo.DIM_CUSTOMERS
CREATE TABLE MAS_dw.dbo.DIM_CUSTOMERS (
    Customer_ID Varchar(20),
    Customer_Name Varchar(60),
    [Status] Varchar(20),
    [Start_Date] Date,
    [Update_Date] Date,
    City Varchar(20),
    Postal_Code Varchar(20),
    Payment_Type Varchar(40),
    Stock_Symbol Varchar(20),
    District Varchar(20)
)

--DROP          TABLE MAS_dw.dbo.DIM_PRODUCTS
--TRUNCATE TABLE MAS_dw.dbo.DIM_PRODUCTS
CREATE TABLE MAS_dw.dbo.DIM_PRODUCTS (
    DW_Product Int,
    Catalog_Number Varchar(60),
    Product_Detail Varchar(100),
    Price Money
)
```





```
--DROP          TABLE MAS_dw.dbo.DIM_CITIES
--TRUNCATE TABLE MAS_dw.dbo.DIM_CITIES
CREATE TABLE MAS_dw.dbo.DIM_CITIES (
    City Varchar(20),
    District Varchar(20)
)

--DROP          TABLE MAS_dw.dbo.DIM_DATES
CREATE TABLE MAS_dw.dbo.DIM_DATES
(
    TheDate date,
    TheDay int,
    TheDayName varchar(20),
    TheDayOfWeek int,
    IsWeekend int,
    TheWeek int,
    TheWeekOfMonth int,
    TheMonth int,
    TheMonthName varchar(20),
    TheQuarter int,
    TheFirstOfQuarter date,
    TheLastOfQuarter date,
    TheYear int,
    IsLeapYear int
)

--DROP          TABLE MAS_dw.dbo.DW_PUBLIC_COMPANIES
--TRUNCATE TABLE MAS_dw.dbo.DW_PUBLIC_COMPANIES
-----Secondary DB-----
CREATE TABLE MAS_dw.dbo.DW_PUBLIC_COMPANIES (
    Stock_Symbol Varchar(20),
    Company Varchar(100),
    [Year] Int,
    [Quarter] Int,
    Net_Profit Money,
    DW_Company Int identity(1,1)
)

--DROP          TABLE MAS_dw.dbo.DW_ORDER_DETAILS
--TRUNCATE TABLE MAS_dw.dbo.DW_ORDER_DETAILS
CREATE TABLE MAS_dw.dbo.DW_ORDER_DETAILS ( /*פרטני למוצר*/
    DW_Order_Detail Int,
    Order_ID Varchar(20),
    Bid_ID Varchar(30),
    [Row_Number] Int,
    Quantity Int,
    [Status] Varchar(60),
    Remains_To_Supply Int,
    Price Money,
    Pay_Date Date,
    Requested_Date Date,
    Supply_Date Date,
    Deal_By Varchar(20),
    DW_Product Int,
    Customer_ID Varchar(20),
    Discount Float,
```



```
Payment_Condition Varchar(20),
VAT Money,
Profit Money,
Cost Money,

Price_After_Discount Money,
Gap Int, /*(Requested_Date-Supply_Date)*/
Final_Price Money,
Order_Date Date
)

CREATE TABLE MAS_dw.dbo.FACT_SUMMARY_ORDERS (
Customer_ID Varchar(20),
[Year] Int,
[Quarter] Int,
Total_Profit Money,
Total_Cost Money,
Total_Payment Money
)

CREATE TABLE MAS_dw.dbo.FACT_SUMMARY_PUBLIC_COMPANIES (
Customer_ID Varchar(20),
Stock_Symbol Varchar(20),
Number_Of_Orders Int,
Total_Price Money
)

create table MAS_dw.dbo.DIM_DATES
(
TheDate date,
TheDay int,
TheDayName varchar(20),
TheDayOfWeek int,
IsWeekend int,
TheWeek int,
TheWeekOfMonth int,
TheMonth int,
TheMonthName varchar(20),
TheQuarter int,
TheFirstOfQuarter date,
TheLastOfQuarter date,
TheYear int,
IsLeapYear int
)

DECLARE @StartDate date = '20100101';
DECLARE @CutoffDate date = DATEADD(DAY, -1, DATEADD(YEAR, 30, @StartDate));
;WITH seq(n) AS
(
SELECT 0 UNION ALL SELECT n + 1 FROM seq
WHERE n < DATEDIFF(DAY, @StartDate, @CutoffDate)
),
d(d) AS
(
SELECT DATEADD(DAY, n, @StartDate) FROM seq
),
```



```
src AS
(
    SELECT
        TheDate          = CONVERT(date, d),
        TheDay           = DATEPART(DAY, d),
        TheDayName       = DATENAME(WEEKDAY, d),
        TheWeek          = DATEPART(WEEK, d),
        TheISOWeek       = DATEPART(ISO_WEEK, d),
        TheDayOfWeek     = DATEPART(WEEKDAY, d),
        TheMonth         = DATEPART(MONTH, d),
        TheMonthName     = DATENAME(MONTH, d),
        TheQuarter       = DATEPART(Quarter, d),
        TheYear          = DATEPART(YEAR, d),
        TheFirstOfMonth  = DATEFROMPARTS(YEAR(d), MONTH(d), 1),
        TheLastOfYear    = DATEFROMPARTS(YEAR(d), 12, 31),
        TheDayOfYear     = DATEPART(DAYOFYEAR, d)
    FROM d
),

dim AS
(
    SELECT
        TheDate,
        TheDay,
        TheDayName,
        TheDayOfWeek,
        IsWeekend        = CASE WHEN TheDayOfWeek IN (CASE @@DATEFIRST WHEN 1 THEN 6 WHEN 7
        THEN 1 END,7)
        THEN 1 ELSE 0 END,

        TheWeek,
        TheWeekOfMonth   = CONVERT(tinyint, DENSE_RANK() OVER
        (PARTITION BY TheYear, TheMonth ORDER BY TheWeek)),

        TheMonth,
        TheMonthName,
        TheQuarter,
        TheFirstOfQuarter = MIN(TheDate) OVER (PARTITION BY TheYear, TheQuarter),
        TheLastOfQuarter  = MAX(TheDate) OVER (PARTITION BY TheYear, TheQuarter),
        TheYear,
        IsLeapYear        = CONVERT(bit, CASE WHEN (TheYear % 400 = 0)
        OR (TheYear % 4 = 0 AND TheYear % 100 <> 0)
        THEN 1 ELSE 0 END)

    FROM src
)
insert into MAS_dw.dbo.DIM_DATES
SELECT * FROM dim
ORDER BY TheDate

OPTION (MAXRECURSION 0);

-----DROPS-----
DROP TABLE MAS_dw.dbo.DIM_CUSTOMERS
DROP TABLE MAS_dw.dbo.DIM_PRODUCTS
DROP TABLE MAS_dw.dbo.DIM_CITIES
DROP TABLE MAS_dw.dbo.DW_PUBLIC_COMPANIES
DROP TABLE MAS_dw.dbo.DW_ORDER_DETAILS
DROP TABLE MAS_dw.dbo.FACT_SUMMARY_ORDERS
DROP TABLE MAS_dw.dbo.FACT_SUMMARY_PUBLIC_COMPANIES
DROP TABLE MAS_dw.dbo.DIM_DATES
```