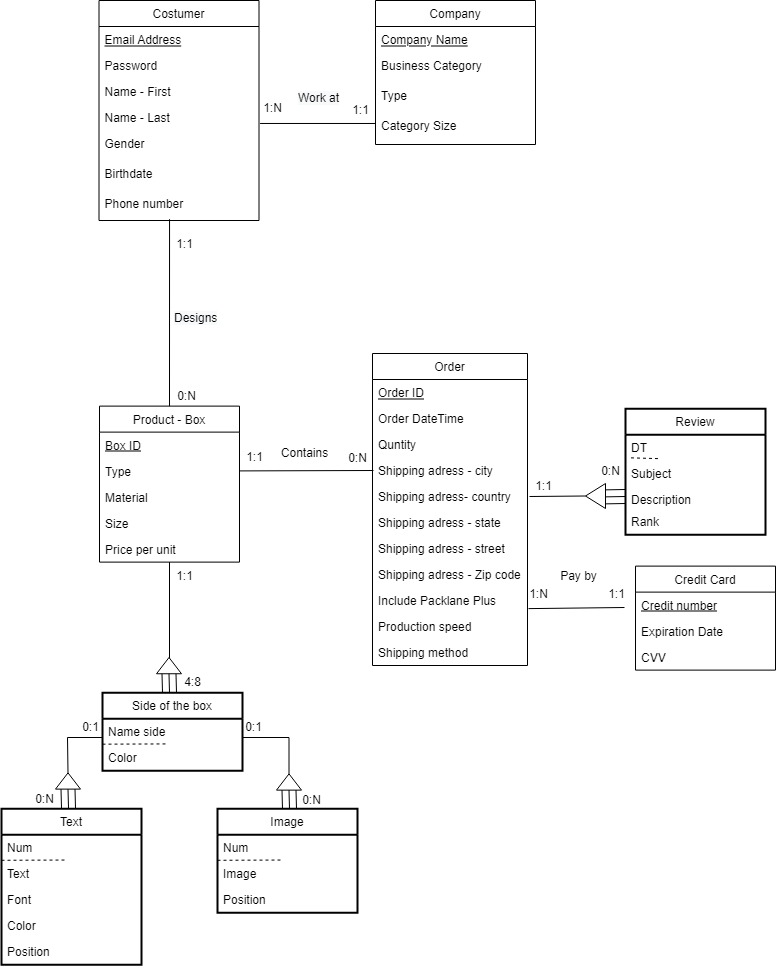
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מס' קבוצה | אתר | | | תאריך הגשה |
| 4 | https://packlane.com/ | | | 19/11/2021 |
| חברי הצוות - מספרי ת.ז | | | | |
| 316444587 | | 205524515 | 315374900 | |

**חלק ב' – הכנת תשתית הנתונים**

# יישום מקורות הנתונים ומחסן הנתונים הטבלאי

## מקור הנתונים הראשי



**הנחות מודל ה-ERD**

|  |  |  |
| --- | --- | --- |
| *מרכיב* | *הנחות* | *הערות נוספות* |
| Entity: Credit Card | פרטי כרטיס האשראי לא נשמרים תחת הלקוח אלא תחת הזמנה ספציפית בלבד. |  |
| Weak Entity: Side of the box | כל צד מקוטלג בהתאם לשם המופיע באתר, וזה כולל צדדים פנימיים. | למשל: Interior left, top. |
| Entity: Review Weak | לקוח יכול לתת מספר ביקורות על הזמנה שביצע. | למשל: לקוח קיבל את המוצר וחשב שנראה טוב, לאחר מספר ימים הבחין שהצבע דוהה מהקופסה. |
| Entity: Review Weak | המזהים של הישות הם השדות הבאים: מזהה ההזמנה ותאריך ושעה. בנוסף, ביקורת על הזמנה יכולה להיכתב על-ידי מבצע ההזמנה בלבד. | לא יתכן שלקוח יבקר הזמנה ספציפית באותם תאריך ושעה. |
| Entity: Review Weak | השדה Rank – מדרג מספרי שהלקוח נותן בין 1-5 |  |
| Attribute: Size | גודל הקופסה ישמר כטקסט, בצורה הבאה:  length x width x height |  |
| Weak Entities: Text, Image | השדה Num הינו מספר רץ המציג את המספר שניתן לאלמנט באותו צד של הקופסה. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Field** | **Data Type** | **Description** |
| **Customers**  List of Employees | Email Address | Varchar (40) | Customer's Email |
| Password | Varchar (20) | Customer's Password |
| First name | Varchar (20) | Customer's First name |
| Last name | Varchar (20) | Customer's Last name |
| Gender | Char (1) | Customer's Gender (M/F) |
| Birthdate | Date | Customer's Birthdate |
| Phone number | Varchar (15) | Customer's Phone number |
| Company name (Companies) | Varchar (20) | Company that orders the product |
| **Companies**  List of Companies | Company name | Varchar (20) | Company that orders the product |
| Business Category | Varchar (20) | The factory's field of manufacturing, such as: Cosmetics. |
| Category Size (Business Size Scale) | Char (1) | Category size of the Business: A, B, C |
| **Products**  List of products details designed by customers | Box ID | Int | Box unique ID |
| Type | Varchar (30) | Box's Type |
| Material | Varchar (20) | Box's Material |
| Size | Varchar (20) | Box's Size |
| Price per unit | Decimal (5,2) | Box's Price per unit |
| Email Address (Customers) | Varchar (40) | FK – Customer's Email |
| **Sides of the box**  List of Box Side details | Box ID (Products) | Int | FK – Box ID |
| Name side | Varchar (20) | Name of the side of the box |
| Color | Varchar (30) | Side's Background color |
| **Images**  List of Images details | Box ID (Sides of the box) | Int | FK – Box ID |
| Name side (Sides of the box) | Varchar (20) | FK – Name side |
| Num | Int | Image unique ID |
| Image | Varchar (40) | URL image's source |
| Position | Varchar (30) | Position of the image on the side of the box |
| **Texts**  List of Text details | Box ID (Sides of the box) | Int | FK – Box ID |
| Name side (Sides of the box) | Varchar (20) | FK – Name side |
| Num | Int | Text Box unique ID |
| Text | Varchar (40) | The written text |
| Font | Varchar (20) | Text's Font |
| Color | Varchar (30) | Text's Color |
| Position | Varchar (30) | Position of the Text Box on the side of the box |
| **Orders**  List of orders made by customers | Order ID | Int | Order unique ID |
| Order Datetime | Date | Order's Datetime |
| Quantity | Int | Quantity of unites ordered |
| Street | Varchar (30) | Shipping Address - Street |
| City | Varchar (20) | Shipping Address - City |
| Country | Varchar (20) | Shipping Address - Country |
| State | Char (2) | Shipping Address - State |
| Zip code | Varchar (10) | Shipping Address – Zip code |
| Include Packlane Plus | Bit | 1. Not purchased the service 2. Purchased the service |
| Production speed (Production speeds) | Varchar (30) | FK - Production Speed |
| Shipping method (Shipping methods) | Varchar (30) | FK - Shipping method |
| Box ID (Products) | Int | FK – Box ID |
| Credit number (Credit Cards) | Char (16) | FK - Credit number (for payment order) |
| **Production speeds**  List of types of production speeds | Production speed | Varchar (30) | Box Production Speed |
| Production speed cost | Decimal (7,2) | The cost of the selected Production Speed |
| **Shipping methods**  List of types of shipping methods | Shipping method | Varchar (30) | Box Shipping method |
| Shipping method cost | Decimal (7,2) | The cost of the selected Shipping Method |
| **Business Size Scale**  List of categories size | Category Size | Char (1) | Category size of the Business: A, B, C. The number of the employees at the business |
| Size | Varchar (10) | A - <5  B - <20  C - <500 |
| **Reviews**  List of reviews written by customers | Order ID (Orders) | Int | FK – Order ID |
| DateTime | DateTime | Review's DateTime |
| Subject | Varchar (20) | Review's Subject |
| Description | Varchar (500) | Review's Description |
| Rank | Int | Between 1-5 |
| **Credit** **Cards**  List of Credit Cards details | Credit number | Char (16) | Credit Card’s Number |
| Expiration Date | Char (5) | Credit Card’s Expiration |
| CVV | Char (3) | Credit Card’s CVV |

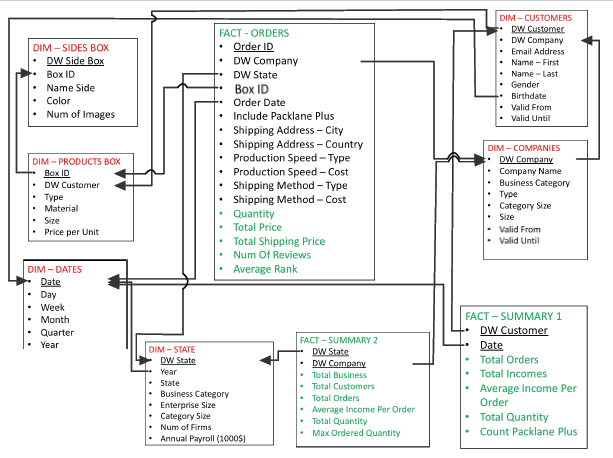
מקור הנתונים המשני

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Field** | **Data Type** | **Description** |
| **Establishment Industry**  Information about businesses and their size in the United States | EstablishmentIndustryID | Int | ID of the record |
| Year | Int | Year of the data valid |
| State | Varchar (30) | State in US |
| Business Category | Varchar (40) | The factory's field of manufacturing, such as: Cosmetics. |
| Enterprise Size | Varchar (30) | The number of employees in the company |
| Num of Firms | Int | Number of firms in the state |
| Annual Payroll (1000$) | Decimal (11,2) | Annual profit of the business |

סכימת כוכב



|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Field** | **Data Type** | **Description** |
| **DIM - Customers**  List of Employees  טבלת מימד משתנה מסוג 2. שומרת את גרסאות נתוני העובדים שהזמינו קופסאות עבור החברה שבה הם עובדים.  ההצדקה באה לידי ביטוי בשדות:  last name. | DW Customer | Int | מזהה העובד ,מפתח עזר, כל ערך משקף גרסת לקוח בעלת טווח תאריכים מוגדר |
| DW Company (DIM-Companies) | Int | שם החברה שעבורה העובד הזמין את המוצר |
| Email Address | Varchar (40) | כתובת מייל |
| First Name | Varchar (20) | שם פרטי |
| Last Name | Varchar (20) | שם משפחה |
| Gender | Char (1) | מגדר העובד (M/F) |
| Birthdate | Date (DIM – Dates) | תאריך הלידה של העובד |
| Valid from | Date | תאריך תחילת תוקף הגרסה |
| Valid until | Date | תאריך סיום תוקף הגרסה |
| **DIM - Companies**  List of Companies  טבלת מימד משתנה מסוג 2. שומרת את גרסאות נתוני החברות שעבורן הלקוחות הזמינו.  ההצדקה באה לידי ביטוי בשדות: company name, category Size , Size. | DW Company | Int | שם החברה שהזמינה את המוצר |
| Company name | Varchar (20) | שם החברה שהזמינה את המוצר |
| Business Category | Varchar (20) | תחום הייצור, למשל: קוסמטיקה |
| Category Size | Char (1) | קטגוריה של גודל העסק: A/B/C |
| Size | Varchar (10) | A - <5  B - <20  C- <500 |
| Type | Varchar (20) | סוג הלקוח שהחברה מהווה:  Premium/regular  (נקבע בהתאם לכמות היחידות שהוזמנו) |
| Valid from | Date | תאריך תחילת תוקף הגרסה |
| Valid until | Date | תאריך סיום תוקף הגרסה |
| **DIM - Products Box**  List of products details designed by customers  טבלת מימד משתנה מסוג 1. נרצה לשמור את הגרסה העדכנית ביותר של הקופסה שהלקוח עיצב (המחיר מתעדכן בהתאם לשינויים בגרסה). | Box ID | Int | מזהה ייחודי לקופסה |
| Type | Varchar (30) | סוג הקופסה |
| Material | Varchar (20) | חומר הגלם ממנו עשויה הקופסה |
| Size | Varchar (20) | מידות הקופסה |
| Price per unit | Decimal (5,2) | מחיר הקופסה ליחידה |
| DW Customer (Customers) | Int | הלקוח שהקופסה שייכת אליו |
| **DIM - Sides box**  List of Box Side details  טבלת מימד משתנה מסוג 1. נרצה לשמור את הגרסה העדכנית ביותר של צד הקופסה שהלקוח עיצב. טבלת מימד זו מחוברת כפתית שלג לטבלת הקופסה, מאחר ויש היררכיה בין קופסה לצד של קופסה. | DW Side Box | Int | מזהה צד של קופסה, מפתח עזר. השימוש במפתח העזר הוא בגלל שהמפתח המקורי מורכב ממספר שדות. |
| Box ID (DIM - Products Box) | Int | מזהה הקופסה שאליה שייך הצד. |
| Name side | Varchar (20) | שם הצד (לדוג': עליון/תחתון..) |
| Color | Varchar (30) | צבע הרקע של הצד |
| Num of Images | Int | מספר התמונות הכולל על גבי אותו צד של הקופסה. בוצעה אגרגציה מהישות החלשה .Image |
| **DIM - States**  Information about businesses and their size in the United States  טבלת מימד משתנה מסוג 1 – נרצה לשמור את הגרסה העדכנית ביותר של נתונים אלו, ללא צורך בשמירה של נתוני עבר שאינם רלוונטיים עוד. | DW State | Int | מזהה ייחודי - מפתח עזר. השימוש במפתח העזר הוא בגלל שהמפתח המקורי מורכב ממספר שדות. |
| Year (DIM-Dates) | Int | שנה שבה נאסף המידע |
| State | Varchar (30) | שם המדינה בארה"ב |
| Business Category | Varchar (40) | תחום הייצור, למשל: קוסמטיקה |
| Enterprise Size | Varchar (30) | The number of employees in the company |
| Category Size | Char (1) | Category size of the Business: A, B, C |
| Num of Firms | Int | מספר החברות במדינה |
| Annual Payroll (1000$) | Decimal (11,2) | גיליון השכר השנתי של החברה (מעיד על מצב הרווחיות של החברה). |
| **DIM - Time**  טבלת מימד זמן, המכילה מאפייני תאריך | Time | Time | מזהה זמן |
| Minute | Int | דקה |
| Hour | Int | שעה |
| **DIM - Dates**  טבלת מימד זמן, המכילה מאפייני תאריך | Date | Date | מזהה תאריך |
| Day | Int | יום בחודש |
| Week | Int | שבוע |
| Month | Int | חודש |
| Quarter | Int | רבעון |
| Year | Int | שנה |
| **FACT - Orders**  List of orders made by customers  טבלת עובדה (טרנזקציה), המכילה את הזמנות של הקופסאות שלקוחות החברה ביצעו. מסכמת את פרטי ההזמנות לרמת גרעיניות של הזמנה בודדת. | Order ID | Int | מזהה ההזמנה |
| DW Company  (DIM - Company) | Int | הלקוח שביצע את ההזמנה (מפתח עזר) |
| State  (DIM - State) | Int | המדינה שאליה בוצעה ההזמנה (מפתח עזר) |
| DW Box  (DIM - Products Box) | Int | הקופסה המוזמנת (מפתח עזר) |
| Order Date (Date) | Date | התאריך בו בוצעה ההזמנה (מפתח עזר) |
| Order Time (Time) | Time | הזמן בו בוצעה ההזמנה (מפתח עזר) |
| Include Packlane Plus | Bit | האם הלקוח רכש את השירות הנוסף: כן – 1, לא – 0. |
| City | Varchar (20) | פרטי משלוח - עיר |
| Country | Varchar (20) | פרטי משלוח - ארץ |
| Production speed | Varchar (30) | מהירות הייצור (Economy/Standard) |
| Production speed cost | Decimal (7,2) | עלות מהירות היצור שנבחרה |
| Shipping method | Varchar (30) | שיטת המשלוח (Economy/Priority) |
| Shipping method cost | Decimal (7,2) | עלות שיטת המשלוח שנבחרה |
| Quantity | Int | כמות כוללת של היחידות שהוזמנו. |
| Total Price | Decimal (11,2) | שדה מחושב – המחיר הכולל של ההזמנה (מחיר ליחידה כפול הכמות) |
| Total Shipping price | Decimal (11,2) | שדה מחושב – המחיר הכולל של שיטת המשלוח |
| Num Of Reviews | Int | שדה מחושב – מספר הביקורות שניתנו על ההזמנה |
| Average Rank | Decimal (3,2) | שדה מחושב - הדירוג הממוצע של כלל הביקורות הניתנות על ההזמנה לפי Rank |
| **FACT - Summary 1**  טבלה סיכומית ממקור הנתונים הראשי. מציגה את הנתונים ברמת גרעיניות לפי לקוח ותאריך (לפי חודש/שנה). כלל השדות החישוביים יתבצעו לפי שילוב של לקוח מסוים ותאריך מסוים. | DW Company (DIM - Companies) | Int | מזהה הלקוח (מפתח עזר) |
| Date (DIM – Date) | Date | מזהה התאריך (מפתח עזר) |
| Total Orders | Int | שדה מחושב - מספר ההזמנות |
| Total Incomes | Decimal (11,2) | שדה מחושב - סכום ההכנסות |
| Average Income Per Order | Decimal (11,2) | שדה מחושב - ממוצע הכנסות להזמנה: סיכום כלל המחירים של כלל ההזמנות/מספר ההזמנות שבוצעו |
| Total Quantity | Int | שדה מחושב – הכמות הכוללת של הקופסאות שהלקוח הזמין |
| Count Packlane Plus | Int | מספר ההזמנות שעבורן רכש הלקוח את השירות הנוסף. |
| **FACT - Summary 2**  טבלה סיכומית המשלבת בין מקור הנתונים הראשי למשני. מציגה את הנתונים ברמת גרעיניות לפי סוג עסק (הלקוח מטבלת הלקוחות) ומדינה. כלל השדות החישוביים יתבצעו לפי שילוב של סוג עסק מסוים במדינה מסוימת. | DW State (DIM - STA | Int | מזהה מדינה (מפתח עזר) |
| DW Company (DIM - Companies) | Int | מזהה הלקוח (מפתח עזר) |
| Total business | Int | שדה מחושב - מספר העסקים הכולל במדינה באותו סוג עסק. |
| Total Customers | Int | שדה מחושב - מספר הלקוחות של החברה לפי המדינה וסוג העסק. |
| Total Orders | Int | שדה מחושב - כמות ההזמנות הכוללת לפי מדינה וסוג עסק. |
| Average Income Per Order | Decimal (11,2) | שדה מחושב - ממוצע הכנסות להזמנה: סיכום כלל ההכנסות מההזמנות לפי סוג מדינה ועסק/מספר ההזמנות שבוצעו |
| Total Quantity | Int | שדה מחושב – הכמות הכוללת של היחידות המוזמנות לפי מדינה וסוג עסק. |
| Max Ordered Quantity | Int | כמות הזמנה מקסימלית לפי מדינה וסוג עסק. |

תיקונים מרכזיים שבוצעו:

* הוספת ישות Companies המייצגת את החברה שעבורה הלקוח מזמין.

# אפיון תהליכי ה ETL

תהליך ה-ETL מורכב ממספר שלבים ופעולות עיקריות:

1. :Data Mirroring בשלב זה מבצעים העתקה של מחסן הנתונים הראשי, וזאת בכדי לא לפגוע בפעילות השוטפת שמתבצעת בו. מבצעים טעינה אינקרמנטלית של טבלאות העובדה וטעינה מלאה של טבלאות המימד.
2. Dimensions Staging: בשלב השני, יוצרים את טבלאות המימד למבנה הטבלאי המתאים למבנה הכוכב.
3. Dimensions Warehousing: בשלב זה, מתבצע עדכון של טבלאות המימד בעזרת פעולת Upsert (עדכון והכנסת נתונים ביחד). פעולה זו עשויה לחסוך זמן ריצה משום שטעינה מלאה עלולה לקחת זמן רב.
4. Fact Staging: יצירת טבלאות העובדה במבנה הטבלאי המתאים למבנה הכוכב. מבצעים טעינה אינקרמנטלית רק של הרשומות העדכניות (נוספו, עודכנו או נמחקו). בשלב זה מבצעים גם מניפולציות ואגרגציה על המידע וכן, מהלכי ביניים לסכימה והעמסה.
5. Referential Integrity: בשלב זה מתבצעת בדיקת תקינות טבלאות העובדה, בה בודקים התאמה של נתוני הממדים לנתוני העובדות על-פי המפתחות הזרים. אם קיימת רשומה בטבלת העובדות שלא נמצאת בטבלת הממד יש להקפיץ הודעת שגיאה.
6. Fact Warehousing: עדכון המידע של טבלאות העובדה בצורה אינקרמנטלית ב-DW.
7. לאחר כל join בתהליך, נבצע פעולתclean שהיא פעולה של ניקוי נתונים – מהלך של בדיקת איכות לנתונים.
8. התלויות הקיימות בין כל שלבי התהליך

* יש לוודא שכל שלב מתבצע לאחר קודמו, כדי שתהליך ה-ETL יעבוד בצורה טובה.
* לפני שמעדכנים את טבלאות העובדה יש להשלים את עדכון טבלאות הממדים ב-DW, מכיוון שקיימים בטבלאות העובדה מפתחות זרים המקושרים לטבלאות המימד.

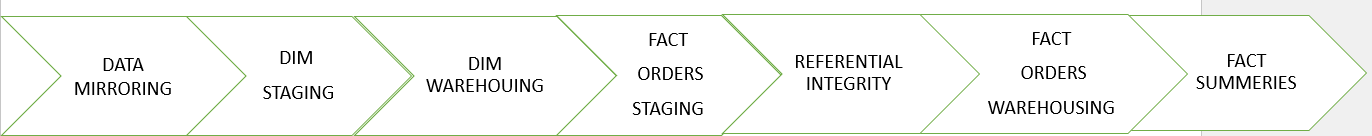
1. בדיקת תקינות (שלב 5) צריכה להתבצע לפני עדכון הטבלאות ב-DW משום שב-DW אין מפתחות זרים (constraint).

במערכת התפעולית, שלבים המבצעים שליפה או טעינה עלולים לקחת זמן רב במיוחד:

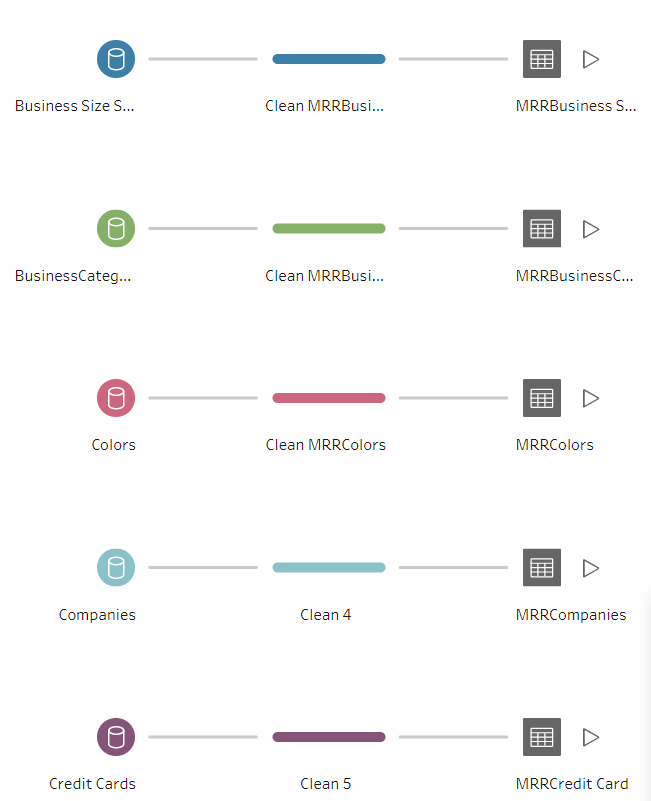
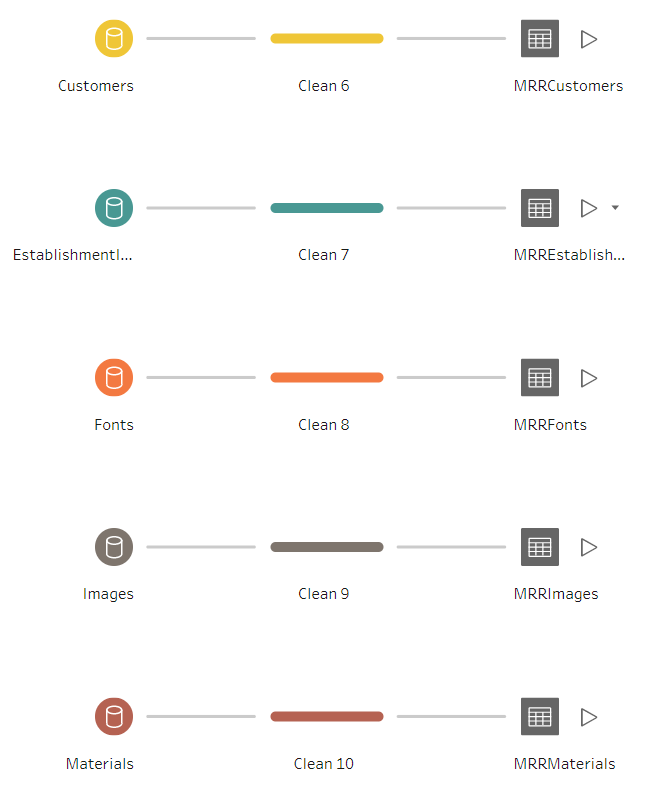
* שלב 2 (Dimensions Staging) עשוי לקחת זמן רב בשל טעינה של כמות מידע גדולה וביצוע Join בין מספר רב של טבלאות.
* טבלאות עובדה מכילות מידע רב יותר מטבלאות מימד, ולכן, שלב 3 (Fact Staging) עשוי לקחת זמן רב במיוחד בשל פעולות טעינה ואגרגציה.
* השלב האחרון (Fact Warehousing) שבו מתבצעת טעינת טבלאות העובדה ל-DW עשוי לקחת זמן רב בשל כמות המידע הגדולה במיוחד.

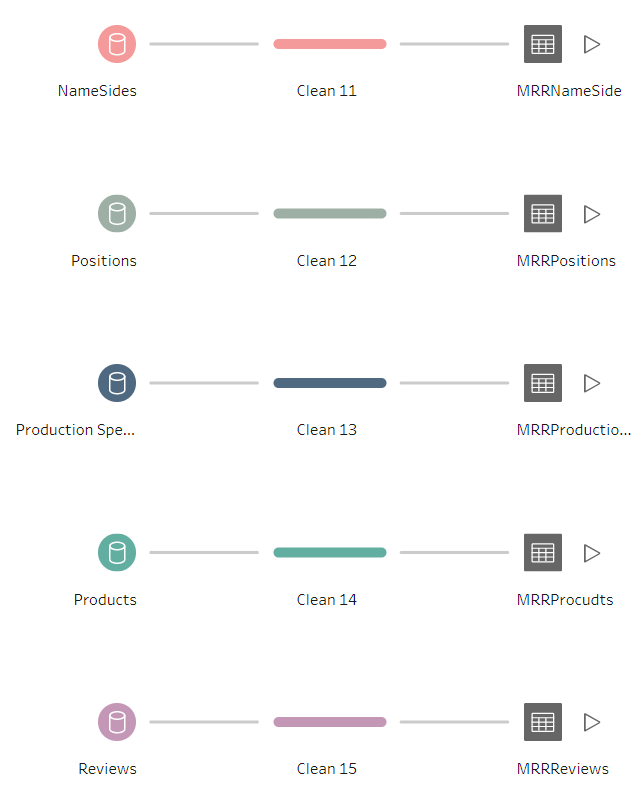
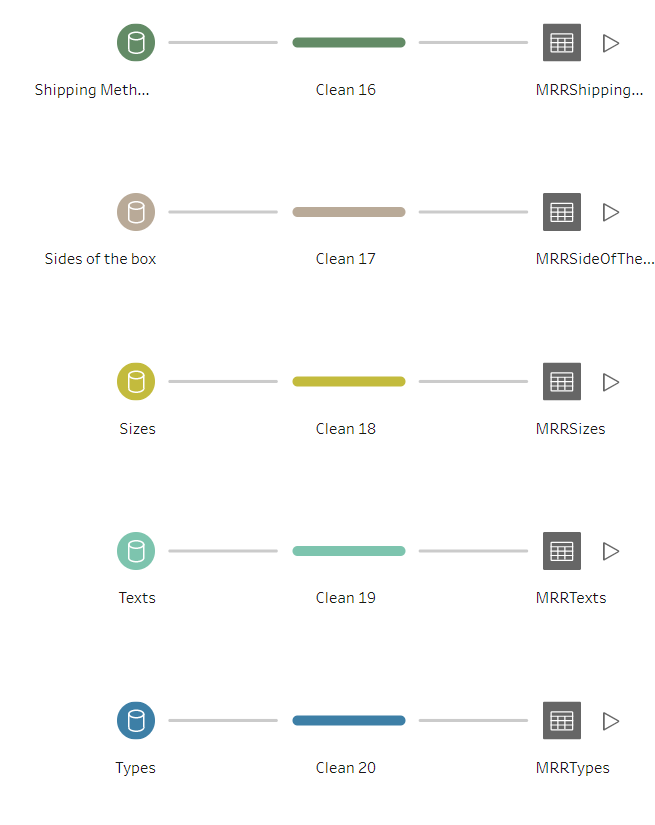
1. שלבים שעשויים להיכשל ובעיות שעשויות להתרחש:

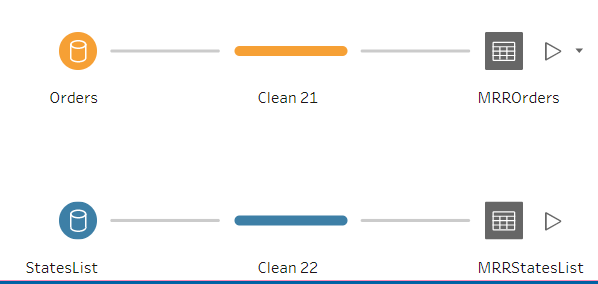
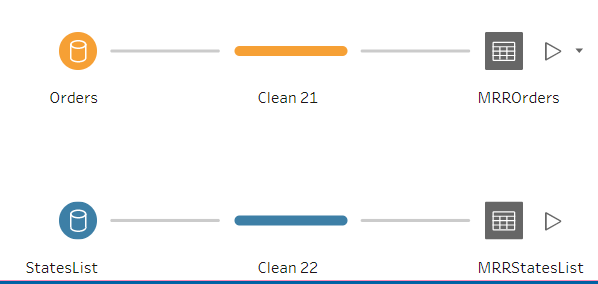
* חוסר תאימות של מפתחות: אי שמירה על תהליך הטעינה בסדר הנכון עלול לגרור בעיה בתאימות המפתחות, בעקבות טעינת טבלאות עובדה לפני טבלאות המימד, לכן יש להיצמד לתהליך ה-ETL הנכון.
* הפסקת שאילתות ועומס על השרת: בעקבות טעינה של כמויות גדולות מאוד של מידע, עשויים להיווצר כשלים טכניים. לכן, כדאי מבעוד מועד להשקיע בתשתית חזקה ויציבה, לנטר את כל התהליך באמצעות לוגים ולהבין את מקור הבעיה, אם קיימת.
* במימד משתנה לאט מסוג 2: כל פעם שטוענים נתונים יש צורך בבדיקה האם קיימת גרסה קודמת, לסגור אותה ולפתוח גרסה חדשה – תהליך שעלול לקחת זמן.

תרשים העל ETL:

# מימוש תהליכי ה-ETL – יישומי הבסיס

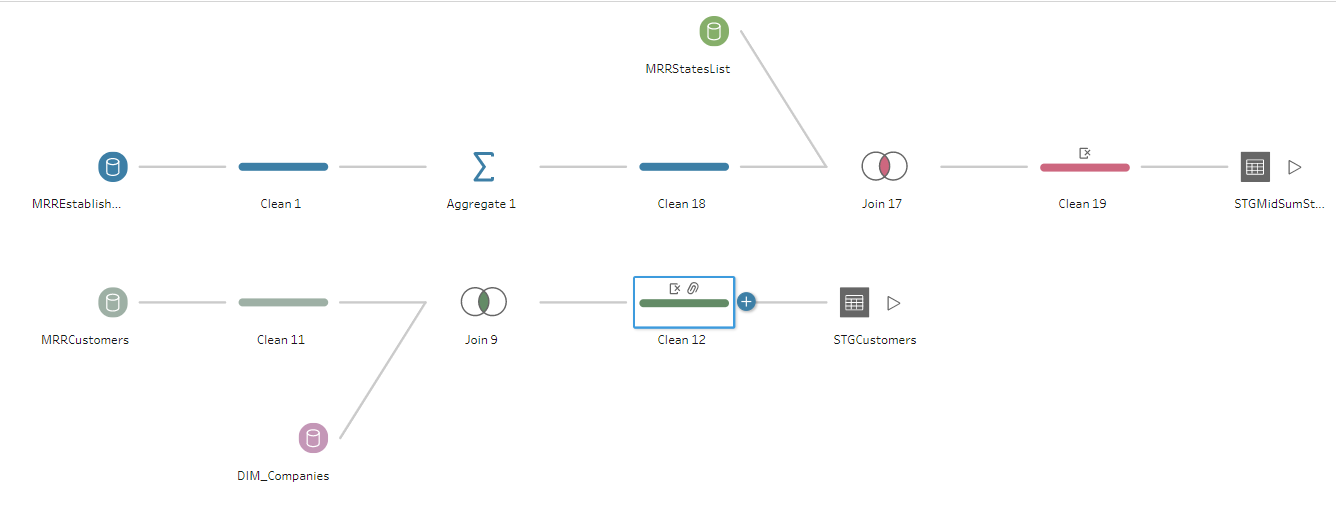
MRR – שיקוף בסיס הנתונים הראשי

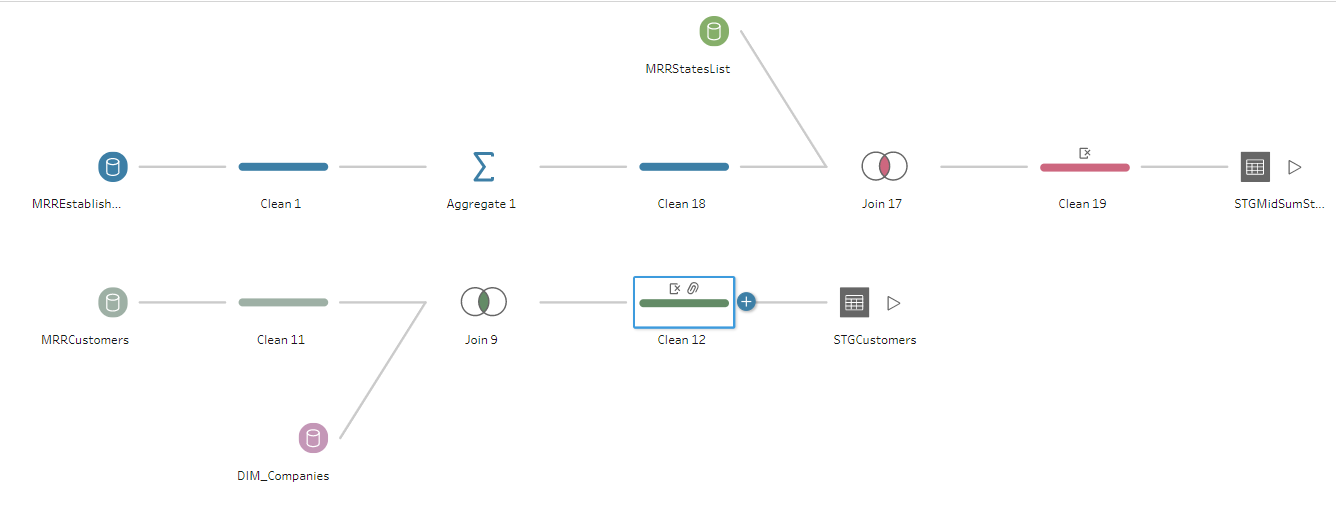




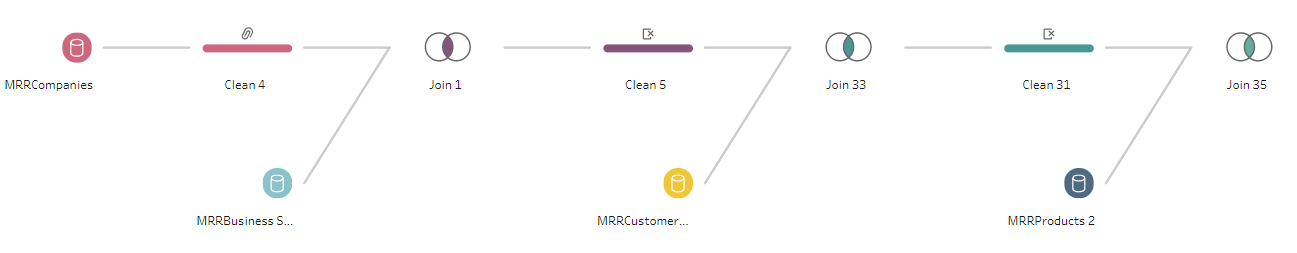
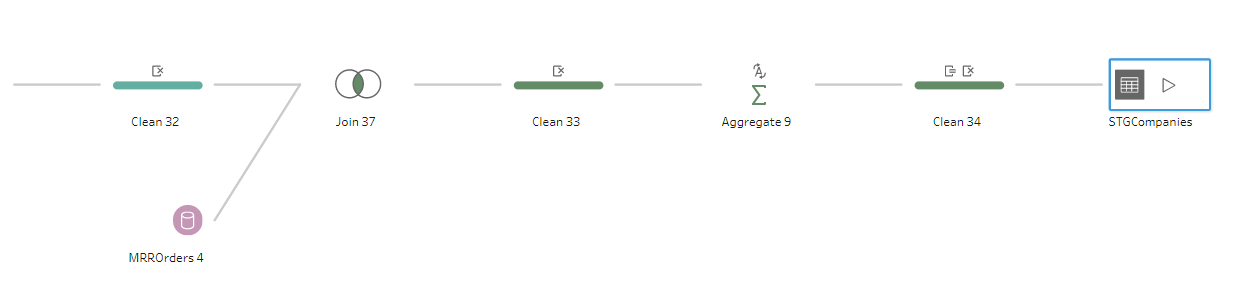
STG

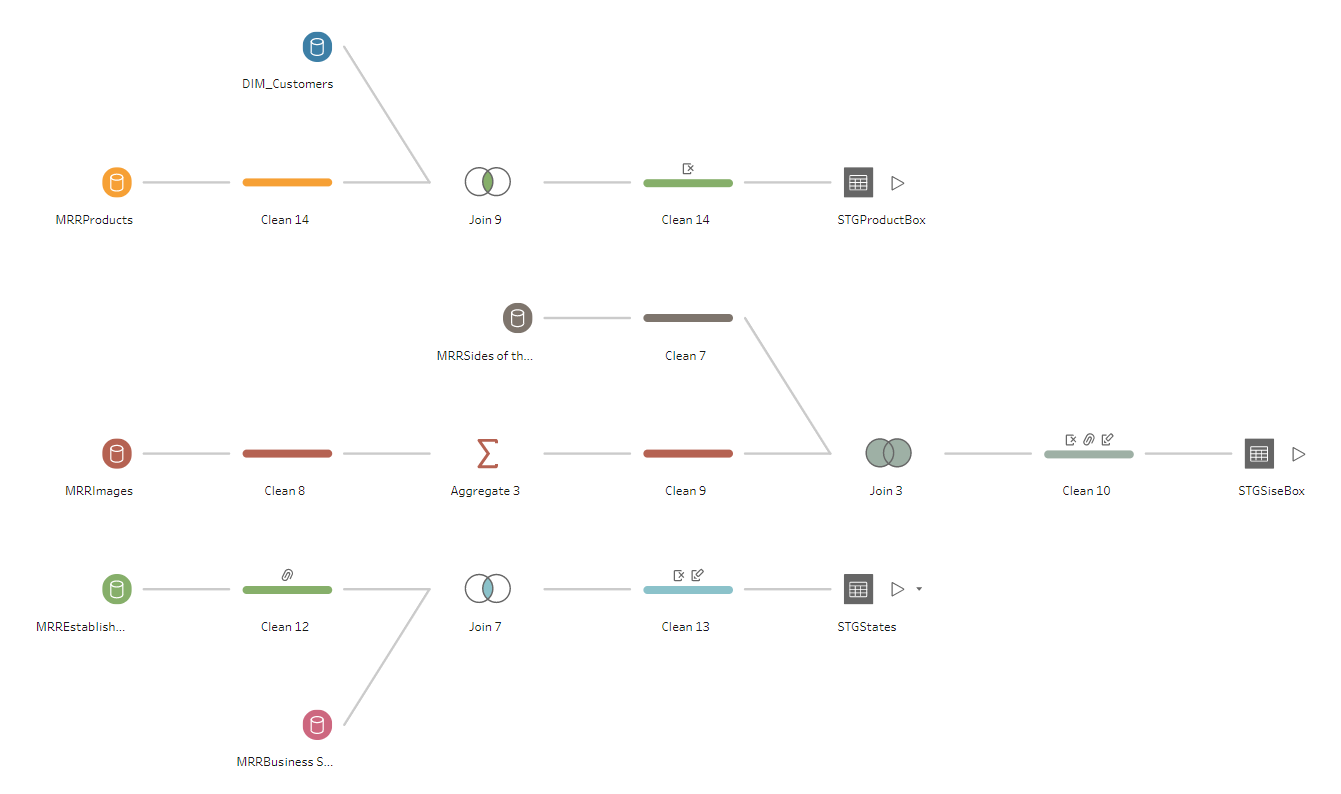
**STGMidSumState -** על מנת לבצע Join בין DIM\_State לבין FACT\_Order, נדרשה ההכנה הבאה: אגרגציה של טבלת MRREstablishmentIndustry לפי מדינה, על מנת ליצור שורה לכל מדינה. בנוסף נעשה Join עם טבלת StateList – רשימת מדינות המציגה את שם המדינה המלא ואת הקיצור. שלב זה נעשה על מנת להתאים בין הפורמט בו רשומה מדינה במימד לבין העובדה.



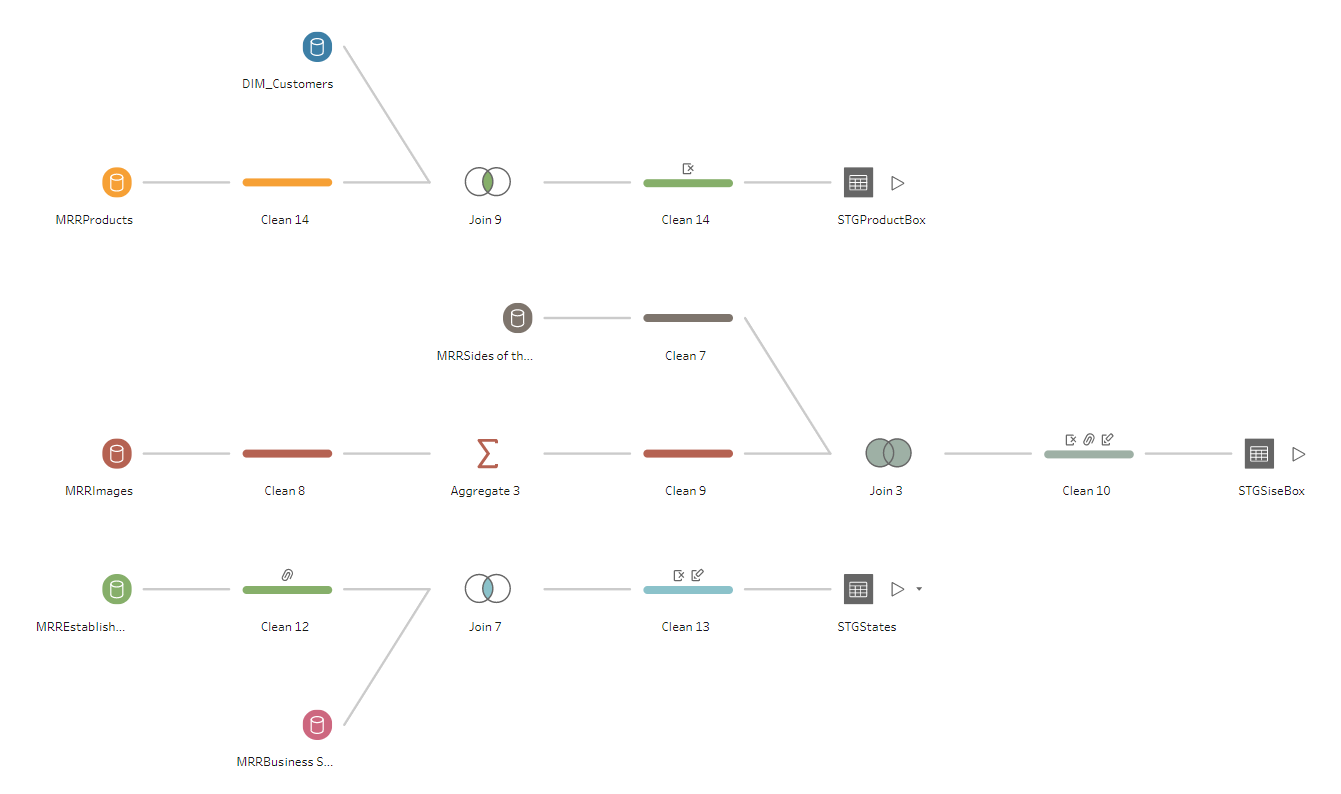
**STGCustomers** – Join בין טבלת Customer ל-Company – לכל לקוח יש את החברה שאליה הוא שייך.

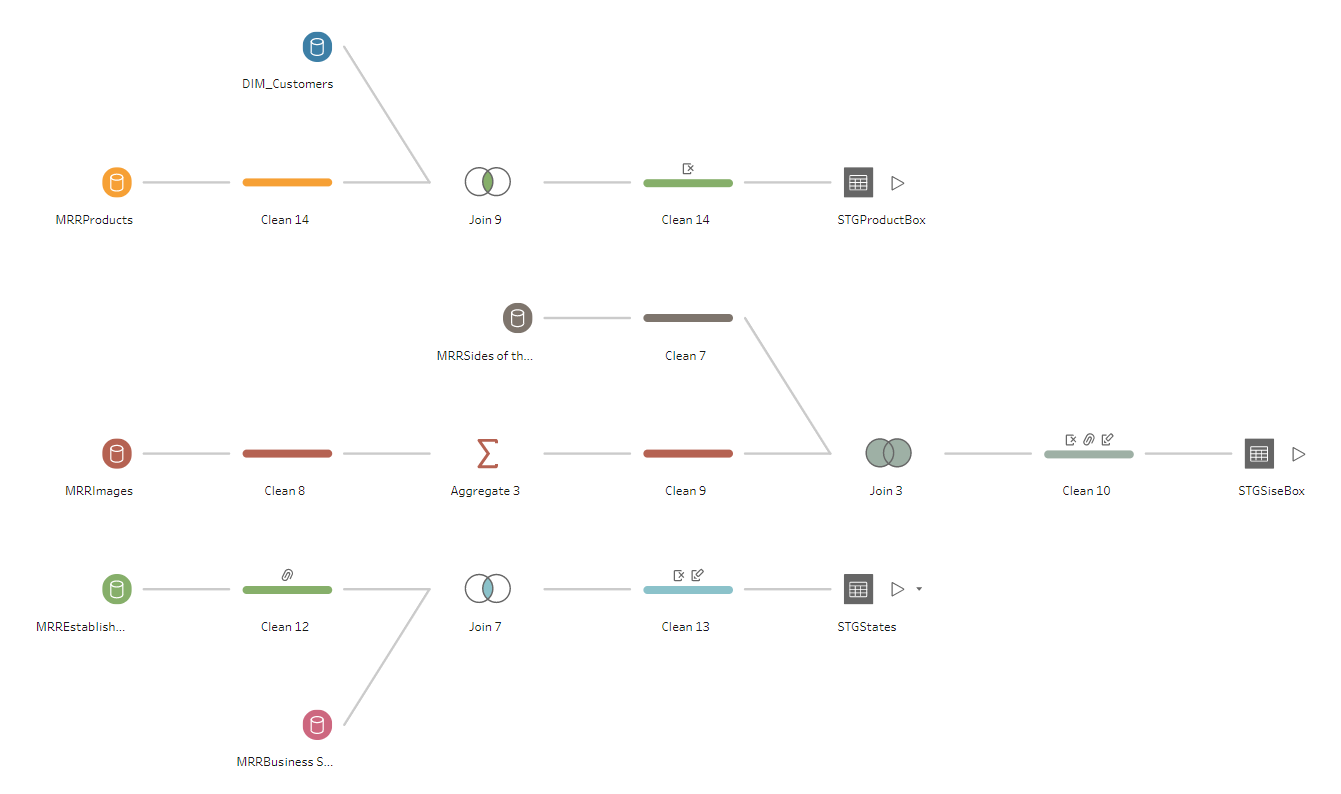
**STGCompanies –** join בין ה-MRRCompanies ל-MRRBusinessScale על מנת להתאים לכל גודל של עסק את המדרג הראוי לו.על מנת ליצור שדה מחושב מסוג Type בטבלת המימד, המתאר את סוג החברה (primum/regular), היה צורך לחשב את כמות הכמויות שהוזמנו על ידי החברה. לכן, נעשו פעולות מסוג join על מנת לחשב את השדה המבוקש- לטבלת חברות נעשה חיבור עם טבלת לקוחות על מנת לזהות את המוצר שהוזמן. לאחר מכן בוצעו אגרגיציות והמידע הוכנס.

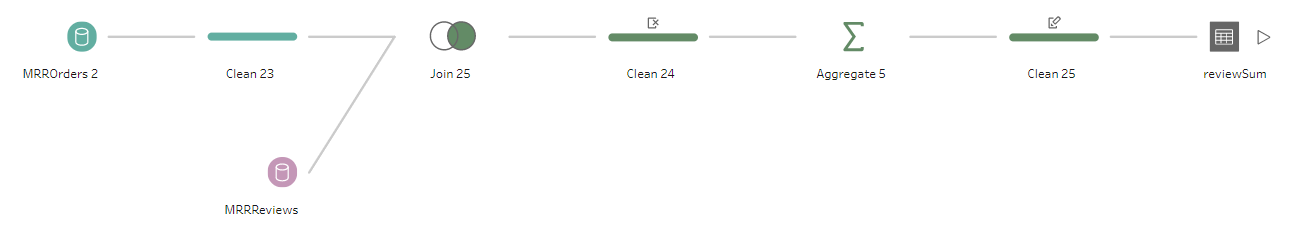


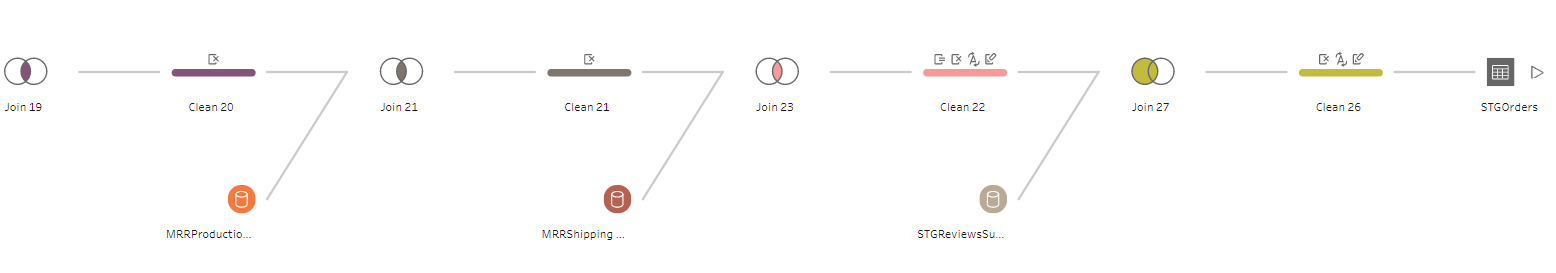
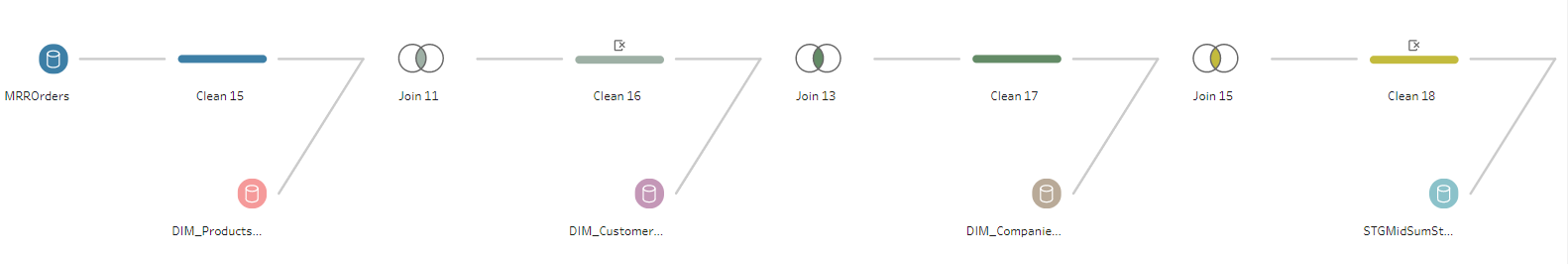
**STGProductBox –** Join בין מימד הלקוח לבין מימד הקופסה (פתית שלג – הלקוח והקופסה ששייכת לו).

**STGSideBox –** על מנת להכניס שדה סיכומי ל- STGSideBox המתאר את מספר התמונות על אותו צד של קופסה, בוצעה אגרגציה ו-Join עם ה-MRR הרלוונטי.



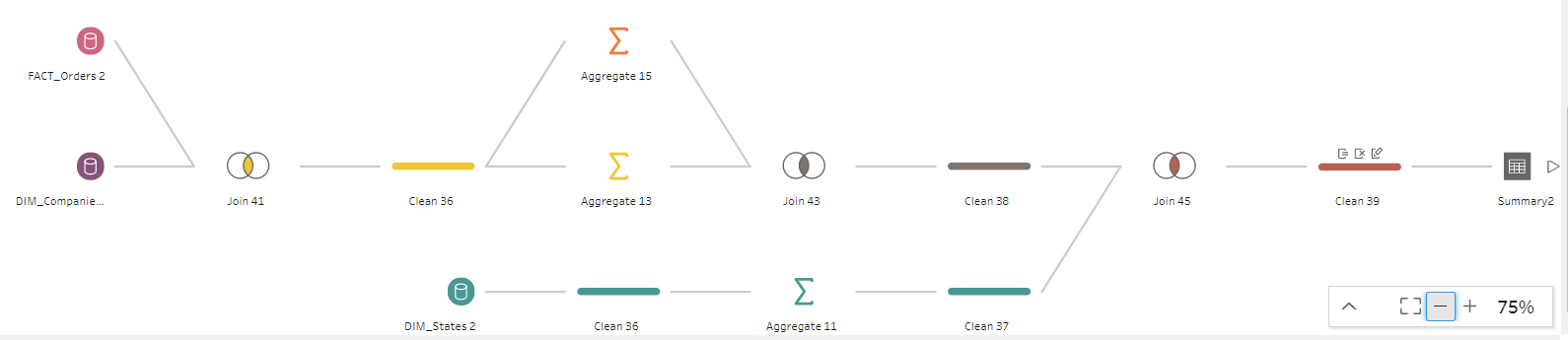
**STGStates –** יצירת Join בין טבלת ה-MRREstablishmentIndustry לבין MRRBusinessSizeScale על מנת להתאים את פורמט יחידת המידה של גדלי העסקים.

**STGReviewSummary –** יצירת טבלת סיכום ביניים על מנת להתאים את רמת הקרדינליות בין ביקורת להזמנה (על כל הזמנה ייתכנו מספר ביקורות) ובכדי ליצור נתונים סיכומיים משדות בטבלת הביקורת שיהיו בטבלת ההזמנות.

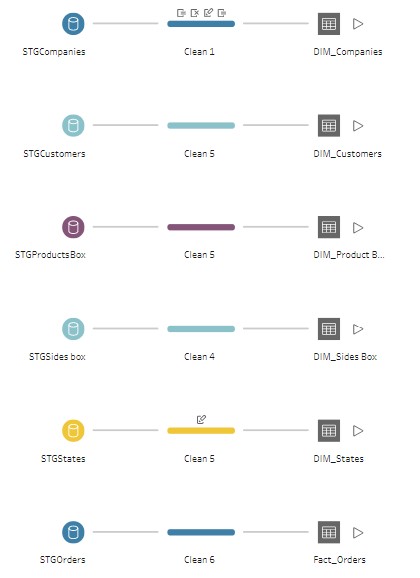
**STGOrders –** יצירת ה-Join המתאימים על מנת לקבל את מבנה הטבלה בסכימת הכוכב וביצוע אגריגציות בהתאם.

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיDW**\_**Summary1 –** יצירת טבלה סיכומית וביצוע אגרגציה על פי השדות חברה ושנה.

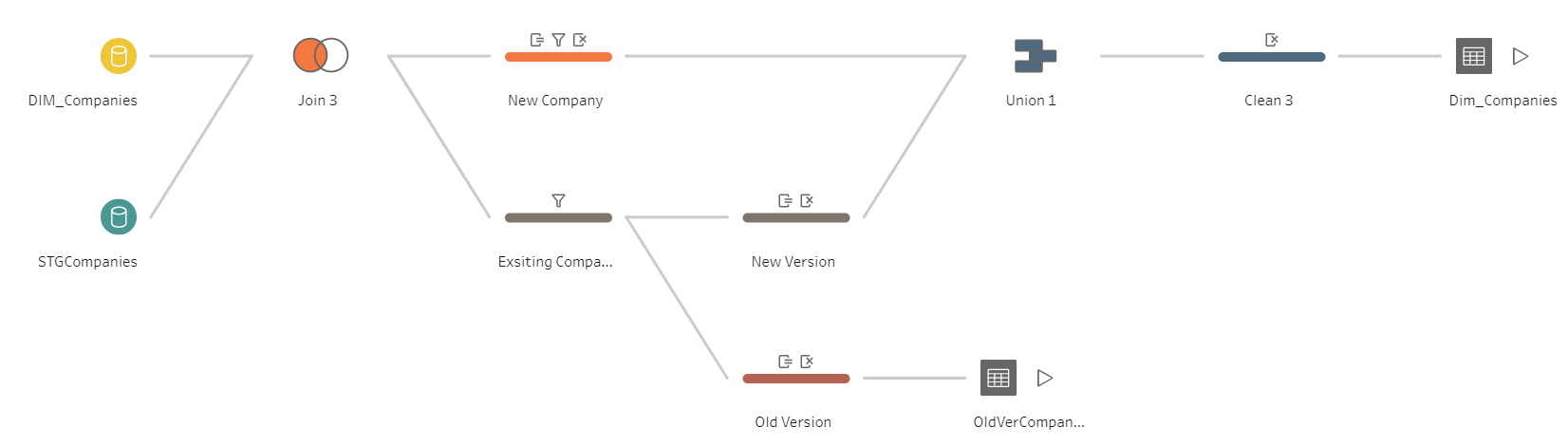
**DW\_Summary2 –** יצירת טבלה סיכומית וביצוע אגרגציות על פי השדות מדינה וסוג עסק.

יצירת DW– העברה מה-STG למבנה הכוכב הסופי



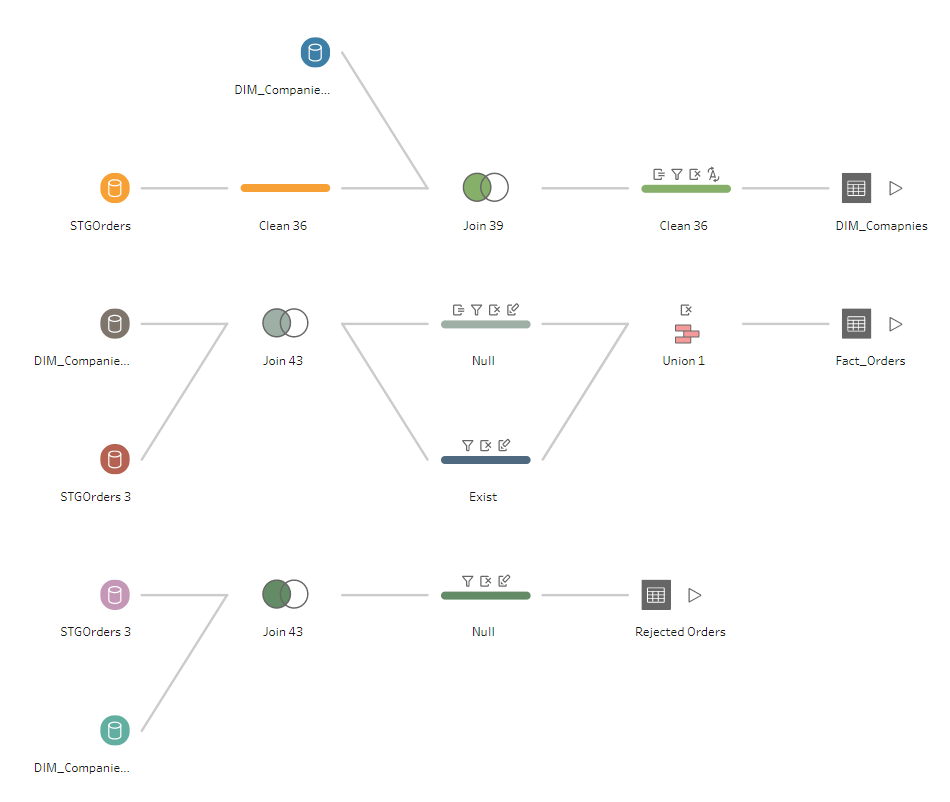
# מימוש תהליכי ה-ETL – יישומים מתקדמים

מימוש משתנה לאט מסוג 2

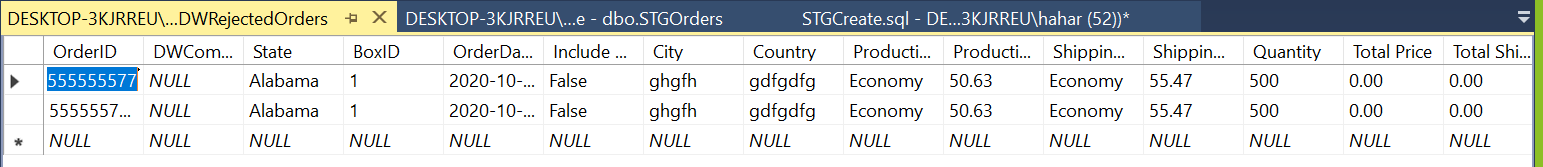
המימוש נעשה על מימד ה-Companies, בוצע Join בין ה-MRR ל-STG על בסיס שם החברה (ייחודי). נעשה Join שמאלי על מנת לבדוק לאילו חברות אין מפתח עזר וכך להבין מי לקוח חדש. הגרסאות הישנות יותר נשמרות בטבלת OldVerCompanies. נוצרת גרסה חדשה במידה ואחד מהשדות הבאים משתנה: Type, Category Size.

מימוש RI

בדיקת ה-RI התבצעה על שדה Companies בטבלת הזמנות. בחלק העליון של התרשים ניתן לראות את מימוש הגישה הראשונה – הכנסת רשומה פקטיבית ל-DIM\_Companies והכנסת הרשומה גם ל-FACT. בחלק התחתון של התמונה ניתן לראות מימוש של הגישה השניה – הכנסת הרשומות הבעייתיות לטבלה נפרדת.



צילום של טבלת ה-Rejected מה-SQL:



**נספחים - SQL**

Create table - Packlane

-- Lookup table

Create table dbo.Fonts (

Font Varchar(20) primary key not null

)

-- Lookup table

create table dbo.BusinessCategory(

[Business Category] Varchar (20) primary key not null ,

)

-- Lookup table

create table dbo.[Types](

[Type] Varchar (30) primary key not null,

)

-- Lookup table

Create table dbo.Sizes(

Size Varchar (20) primary key not null,

)

-- Lookup table

Create table dbo.Materials(

Material Varchar (20) primary key not null,

)

-- Lookup table

Create table dbo.NameSides(

NameSide Varchar(20) primary key not null

)

-- Lookup table

Create table dbo.Colors(

Color Varchar (30) primary key not null,

)

-- Lookup table

Create table dbo.Positions(

Position Varchar (30) primary key not null,

)

Create table dbo.[Business Size Scale](

[Category Size] Char (1) Not null,

Size Varchar (10) Not null,

Constraint PK\_BusinessSizeScale primary key([Category Size]),

)

Create table dbo.Companies(

[Company name] Varchar (40) Not null,

[Business Category] Varchar (20) ,

[Category Size] Char (1),

Constraint PK\_Customers primary key([Company name]),

)

ALTER TABLE dbo.Companies ADD CONSTRAINT FK\_BusinessCategory

FOREIGN KEY ([Business Category]) REFERENCES dbo.BusinessCategory ([Business Category])

ALTER TABLE dbo.Companies ADD CONSTRAINT FK\_CategorySize

FOREIGN KEY ([Category Size]) REFERENCES dbo.[Business Size Scale] ([Category Size])

Create table dbo.Customers(

[Email Address] Varchar (40) Not null,

[Password] Varchar (20) Not null,

[First name] Varchar (20)Not null,

[Last name] Varchar(20)Not null,

Gender Char (1),

Birthdate Date,

[Phone number] Varchar (15),

[Company name] Varchar (40) Not null,

ValidFrom Date,

ValidUntil Date

Constraint PK\_Customers2 primary key([Email Address]),

Constraint FK\_Companyname

FOREIGN KEY ([Company name]) REFERENCES dbo.Companies ([Company name]),

Constraint CK\_EmailAddress check ([Email Address] LIKE '%@%.%'),

Constraint Ck\_Password check (LEN(Password) >= 8),

Constraint CK\_Gender check (Gender IN('M','F')),

)

ALTER TABLE dbo.Customers drop column ValidFrom

ALTER TABLE dbo.Customers drop column ValidUntil

Create table dbo.Products(

BoxID Int Not null,

[Type] Varchar (30) Not null,

Material Varchar (20) Not null,

Size Varchar(20) Not null,

[Price per unit] Decimal (5,2) Not null,

[Email Address] Varchar (40) Not null,

Constraint PK\_Products primary key(BoxID),

Constraint FK\_EmailAddress

FOREIGN KEY ([Email Address]) REFERENCES dbo.Customers ([Email Address]),

Constraint FK\_Type

FOREIGN KEY (Type) REFERENCES dbo.[Types] ([Type]),

Constraint FK\_Material

FOREIGN KEY (Material) REFERENCES dbo.Materials (Material),

Constraint FK\_Sizes

FOREIGN KEY (Size) REFERENCES dbo.Sizes (Size),

Constraint Ck\_PPU check ([Price per unit] >= 0),

)

Create table dbo.[Sides of the box](

BoxID Int Not null,

[Name side] Varchar (20) Not null,

Color Varchar (30) Not null Default 'White'

Constraint PK\_SidesOfTheBox primary key(BoxID,[Name side]),

Constraint FK\_BoxID

FOREIGN KEY (BoxID) REFERENCES dbo.Products (BoxID),

Constraint FK\_NameSide

FOREIGN KEY ([Name side]) REFERENCES dbo.NameSides (NameSide),

Constraint FK\_Color

FOREIGN KEY (Color) REFERENCES dbo.Colors (Color)

)

Create table dbo.Images(

BoxID Int Not null,

[Name side] Varchar (20) Not null,

Num Int Not null,

Image Varchar (40) Not null,

Position Varchar (30) Not null Default 'Center'

Constraint PK\_Images primary key(BoxID,[Name side],Num),

Constraint FK\_SideBox

FOREIGN KEY (BoxID,[Name side]) REFERENCES dbo.[Sides of the box] (BoxID,[Name side]),

Constraint FK\_ImagePos

FOREIGN KEY (Position) REFERENCES dbo.Positions (Position)

)

Create table dbo.Texts(

BoxID Int Not null,

[Name side] Varchar (20) Not null,

Num Int Not null,

[Text] Varchar (40) Not null,

Font Varchar (20) Not null Default 'Proxima Nova',

Color Varchar (30) Not null Default 'Black',

Position Varchar (30) Not null Default 'Center',

Constraint PK\_Texts primary key(BoxID,[Name side], Num),

Constraint FK\_SideBoxText

FOREIGN KEY (BoxID,[Name side]) REFERENCES dbo.[Sides of the box] (BoxID,[Name side]),

Constraint FK\_Fonts

FOREIGN KEY (Font) REFERENCES dbo.Fonts (Font),

Constraint FK\_TextPosition

FOREIGN KEY (Position) REFERENCES dbo.Positions (Position),

Constraint FK\_TextColor

FOREIGN KEY (Color) REFERENCES dbo.Colors (Color)

)

Create table dbo.[Production Speeds](

[Production speed] Varchar (30) Not null,

[Production speed cost] Decimal (7,2) Not null,

Constraint PK\_ProductionSpeeds primary key([Production speed]),

Constraint Ck\_PSC check ([Production speed cost] >= 0),

)

Create table dbo.[Shipping Methods](

[Shipping method] Varchar (30) Not null,

[Shipping method cost] Decimal (7,2) Not null,

Constraint PK\_ShippingMethod primary key([Shipping method]),

Constraint Ck\_SMC check ([Shipping method cost] >= 0),

)

Create table dbo.[Credit Cards](

[Credit number] char (16) Not null,

[Expiration Date] Char (5) Not null,

CVV Char (3) Not null,

Constraint PK\_CreditCards primary key([Credit number]),

Constraint CK\_CreditNum check ([Credit number] Like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

Constraint CK\_ExpirationDate check ([Expiration Date] Like '[0-1][0-9]/[0-9][0-9]'),

Constraint CK\_CVV check (CVV Like '[0-9][0-9][0-9]')

)

Create table dbo.Orders(

OrderID Int Not null,

[Datetime] Datetime Not null,

Quantity int Not null,

Street Varchar (30) Not null,

City Varchar (20) Not null,

Country Varchar (20) Not null,

[State] Char(2) Not null,

[Zip code] Varchar (10) Not null,

[Packlane Plus] Bit,

[Production speed] Varchar (30) Not null,

[Shipping method] Varchar (30) Not null,

BoxID Int Not null,

[Credit number] Char (16),

Constraint PK\_Orders primary key(OrderID),

Constraint Ck\_Quantity check (Quantity > 0),

Constraint FK\_PS

FOREIGN KEY ([Production speed]) REFERENCES dbo.[Production Speeds] ([Production speed]),

Constraint FK\_SM

FOREIGN KEY ([Shipping method]) REFERENCES dbo.[Shipping Methods] ([Shipping method]),

Constraint FK\_Products

FOREIGN KEY (BoxID) REFERENCES dbo.Products (BoxID),

Constraint FK\_CN

FOREIGN KEY ([Credit number]) REFERENCES dbo.[Credit Cards] ([Credit number])

)

Create table dbo.Reviews(

OrderID Int Not null,

[Datetime] Datetime Not null,

[Subject] Varchar (20) Not null,

[Description] Varchar (500),

[Rank] Int Not null,

Constraint PK\_Reviews primary key(OrderID,[Datetime]),

Constraint FK\_OrderID

FOREIGN KEY (OrderID) REFERENCES dbo.Orders (OrderID),

Constraint Ck\_Rank check ([Rank]>= 1 and [Rank]<= 5),

)

Create table dbo.EstablishmentIndustry(

EstablishmentIndustryID Int Not null,

[Year] Int not null,

[State] Varchar (30) not null,

[Business Category] Varchar (20) not null,

[Enterprise Size] varchar(30) not null,

[Num of Firms] Int,

[Annual Payroll (1000$)] Decimal (11,2),

Constraint PK\_EstablishmentIndustry primary key(EstablishmentIndustryID),

Constraint FK\_BusinessCategory2

FOREIGN KEY ([Business Category]) REFERENCES dbo.BusinessCategory ([Business Category])

)

Create table dbo.StatesList(

AGState Char(2) Not null,

[State] Varchar (30) not null,

Constraint PK\_StatesList primary key(AGState),

)

**MRR**

-- Lookup table

Create table dbo.MRRFonts (

Font Varchar(20) primary key not null

)

-- Lookup table

create table dbo.MRRBusinessCategory(

[Business Category] Varchar (20) primary key not null ,

)

-- Lookup table

create table dbo.[MRRTypes](

[Type] Varchar (30) primary key not null,

)

-- Lookup table

Create table dbo.MRRSizes(

Size Varchar (20) primary key not null,

)

-- Lookup table

Create table dbo.MRRMaterials(

Material Varchar (20) primary key not null,

)

-- Lookup table

Create table dbo.MRRNameSides(

NameSide Varchar(20) primary key not null

)

-- Lookup table

Create table dbo.MRRColors(

Color Varchar (30) primary key not null,

)

-- Lookup table

Create table dbo.MRRPositions(

Position Varchar (30) primary key not null,

)

Create table dbo.[MRRBusiness Size Scale](

[Category Size] Char (1) Not null,

Size Varchar (10) Not null,

Constraint PK\_BusinessSizeScale primary key([Category Size]),

)

Create table dbo.MRRCompanies(

[Company name] Varchar (40) Not null,

[Business Category] Varchar (20) ,

[Category Size] Char (1),

Constraint PK\_Customers primary key([Company name]),

)

Create table dbo.MRRCustomers(

[Email Address] Varchar (40) Not null,

[Password] Varchar (20) Not null,

[First name] Varchar (20) Not null,

[Last name] Varchar(20) Not null,

Gender Char (1),

Birthdate Date,

[Phone number] Varchar (15),

[Company name] Varchar (40) Not null,

Constraint PK\_Customers2 primary key([Email Address]),

)

Create table dbo.MRRProducts(

BoxID Int Not null,

[Type] Varchar (30) Not null,

Material Varchar (20) Not null,

Size Varchar (20) Not null,

[Price per unit] Decimal (5,2) Not null,

[Email Address] Varchar (40) Not null,

Constraint PK\_Products primary key(BoxID),

)

Create table dbo.[MRRSides of the box](

BoxID Int Not null,

[Name side] Varchar (20) Not null,

Color Varchar (30) Not null

Constraint PK\_SidesOfTheBox primary key(BoxID,[Name side]),

)

Create table dbo.MRRImages(

BoxID Int Not null,

[Name side] Varchar (20) Not null,

Num Int Not null,

[Image] Varchar (40) Not null,

Position Varchar (30) Not null

Constraint PK\_Images primary key(BoxID,[Name side],Num),

)

Create table dbo.MRRTexts(

BoxID Int Not null,

[Name side] Varchar (20) Not null,

Num Int Not null,

[Text] Varchar (40) Not null,

Font Varchar (20) Not null,

Color Varchar (30) Not null,

Position Varchar (30) Not null,

Constraint PK\_Texts primary key(BoxID,[Name side], Num),

)

Create table dbo.[MRRProduction Speeds](

[Production speed] Varchar (30) Not null,

[Production speed cost] Decimal (7,2) Not null,

Constraint PK\_ProductionSpeeds primary key([Production speed]),

)

Create table dbo.[MRRShipping Methods](

[Shipping method] Varchar (30) Not null,

[Shipping method cost] Decimal (7,2) Not null,

Constraint PK\_ShippingMethod primary key([Shipping method]),

)

Create table dbo.[MRRCredit Cards](

[Credit number] char (16) Not null,

[Expiration Date] Char (5) Not null,

CVV Char (3) Not null,

Constraint PK\_CreditCards primary key([Credit number]),

)

Create table dbo.MRROrders(

OrderID Int Not null,

[Datetime] Datetime Not null,

Quantity int Not null,

Street Varchar (30) Not null,

City Varchar (20) Not null,

Country Varchar (20) Not null,

[State] Char(2) Not null,

[Zip code] Varchar (10) Not null,

[Packlane Plus] Bit,

[Production speed] Varchar (30) Not null,

[Shipping method] Varchar (30) Not null,

BoxID Int Not null,

[Credit number] Char (16),

Constraint PK\_Orders primary key(OrderID),

)

Create table dbo.MRRReviews(

OrderID Int Not null,

[Datetime] Datetime Not null,

[Subject] Varchar (20) Not null,

[Description] Varchar (500),

[Rank] Int Not null,

Constraint PK\_Reviews primary key(OrderID,[Datetime]),

)

Create table dbo.MRREstablishmentIndustry(

EstablishmentIndustryID Int Not null,

[Year] Int not null,

[State] Varchar (30) not null,

[Business Category] Varchar (20) not null,

[Enterprise Size] varchar(30) not null,

[Num of Firms] Int,

[Annual Payroll (1000$)] Decimal (11,2),

Constraint PK\_EstablishmentIndustry primary key(EstablishmentIndustryID),

)

Create table dbo.MRRStatesList(

AGState Char(2) Not null,

[State] Varchar (30) not null,

Constraint PK\_StatesList primary key(AGState),

)

**STG**

Create table dbo.STGCustomers(

[Email Address] Varchar (40) Not null,

[DW Companies] Int Not null,

[First name] Varchar (20) Not null,

[Last name] Varchar(20) Not null,

Gender Char (1) Not null,

Birthdate Date Not null,

Constraint PK\_Customers primary key([Email Address]),

)

Create table dbo.STGCompanies(

[Company name] Varchar (40) Not null,

[Business Category] Varchar (20) Not null,

[Category Size] Char (1) Not null,

Size Varchar(10) Not null,

[Type] Varchar (20) ,

Constraint PK\_Companies primary key([Company name]),

)

Create table dbo.STGProductsBox(

BoxID Int Not null,

[Type] Varchar (30) Not null,

Material Varchar (20) Not null,

Size Varchar(20) Not null,

[Price per unit] Decimal (5,2) Not null,

DWCustomers Int Not null,

Constraint PK\_ProductsBox primary key(BoxID),

)

Create table dbo.[STGSides box](

BoxID Int Not null,

[Name side] Varchar (20) Not null,

Color Varchar (30) Not null ,

NumOfImages Int,

Constraint PK\_SidesOfTheBox primary key(BoxID,[Name side]),

)

Create table dbo.STGStates(

[DW State] Int Not null,

[Year] Int not null,

[State] Varchar (30) not null,

[Business Category] Varchar (20) not null,

[Enterprise Size] varchar(30) not null,

[Category Size] Char(1) not null,

[Num of Firms] Int not null,

[Annual Payroll (1000$)] Decimal (11,2) not null,

Constraint PK\_EstablishmentIndustry primary key([DW State]),

)

Create table dbo.STGOrders(

OrderID Int Not null,

DWCompany Int Not null,

[State] Varchar (30) Not null,

BoxID Int Not null,

OrderDate Date Not null,

[Include Packlane Plus] Bit Not null,

City Varchar (20) Not null,

Country Varchar (20) Not null,

[Production speed] Varchar (30) Not null,

[Production speed Cost] Decimal (7,2) Not null,

[Shipping method] Varchar (30) Not null,

[Shipping method Cost] Decimal (7,2) Not null,

Quantity int Not null,

[Total Price] Decimal (11,2) Not null,

[Total Shipping Price] Decimal (11,2) Not null,

[Num Of Reviews] int ,

[Average Rank] Decimal (3,2) ,

Constraint PK\_Orders primary key(OrderID),

)

Create table dbo.STGMidSumState(

AGState Char(2) Not null,

[State] Varchar (30) not null,

[Num of Firms] Int not null,

[Annual Payroll (1000$)] Decimal (11,2) not null,

Constraint PK\_StatesList primary key(AGState),

)

Create table dbo.STGReviewsSummary(

OrderID Int Not null,

NumOfReviews int Not null,

AverageRank Decimal (3,2) Not null,

Constraint PK\_ReviewsSummary primary key (OrderID),

)

Insert Into dbo.STGOrders (OrderID,DWCompany, [State], BoxID,OrderDate, [Include Packlane Plus],City,Country, [Production speed],

[Production speed Cost],[Shipping method],[Shipping method Cost],Quantity,[Total Price], [Total Shipping Price], [Num Of Reviews],[Average Rank])

Values ('555555577','700000','Alabama','1','2020-10-22','0','ghgfh','gdfgdfg','Economy','50.63','Economy','55.47','500','0','0','0','0')

Insert Into dbo.STGOrders (OrderID,DWCompany, [State], BoxID,OrderDate, [Include Packlane Plus],City,Country, [Production speed],

[Production speed Cost],[Shipping method],[Shipping method Cost],Quantity,[Total Price], [Total Shipping Price], [Num Of Reviews],[Average Rank])

Values ('557888877','','Alabama','1','2020-10-22','0','ghgfh','gdfgdfg','Economy','50.63','Economy','55.47','500','0','0','0','0')

**DW**

Create table dbo.DWSummary1(

DWCompany Int Not null,

[Date] Int Not null,

[Total Orders] Int Not null,

[Total Incomes] Decimal (11,2) Not null,

[Average Income Per Order] Decimal (11,2) Not null,

[Total Quantity] Int Not null,

[Count Packlane Plus] Int Not null,

Constraint PK\_DWSummary1 primary key(DWCompany,[Date]),

)

Create table dbo.DWSummary2(

[State] Varchar (30) Not null,

[Business Category] Varchar (20) Not null,

[Total Business] Int Not null,

[Total Companies] Int Not null,

[Total Orders] Int Not null,

[Average Income Per Order] Decimal (11,2) Not null,

[Total Quantity] Int Not null,

[Max Ordered Quantity] Int Not null,

Constraint PK\_DWSummary2 primary key([State],[Business Category]),

)

Create table dbo.DIM\_Customers(

DWCustomers Int IDENTITY(1,1) Not null,

[DW Companies] Int Not null,

[Email Address] Varchar (40) Not null,

[First name] Varchar (20) Not null,

[Last name] Varchar(20) Not null,

Gender Char (1) Not null,

Birthdate Date Not null,

ValidFrom Date,

ValidUntil Date

Constraint PK\_Customers primary key(DWCustomers),

)

Create table dbo.DIM\_Companies(

[DW Companies] Int IDENTITY(1,1) ,

[Company name] Varchar (40) ,

[Business Category] Varchar (20) ,

[Category Size] Char (1) ,

Size Varchar(10) ,

[Type] Varchar (20) ,

ValidFrom Date,

ValidUntil Date

Constraint PK\_Companies primary key([DW Companies]),

)

Create table dbo.DIM\_ProductsBox(

BoxID Int Not null,

[Type] Varchar (30) Not null,

Material Varchar (20) Not null,

Size Varchar(20) Not null,

[Price per unit] Decimal (5,2) Not null,

DWCustomers Int Not null,

Constraint PK\_DIM\_ProductsBox primary key(BoxID),

)

Create table dbo.[DIM\_Sides box](

[DW Sides box] Int IDENTITY(1,1) Not null,

BoxID Int Not null,

[Name side] Varchar (20) Not null,

Color Varchar (30) Not null ,

NumOfImages Int,

Constraint PK\_SidesOfTheBox primary key([DW Sides box]),

)

Create table dbo.DIM\_States(

DWState Int not null,

[Year] Int not null,

[State] Varchar (30) not null,

[Business Category] Varchar (20) not null,

[Enterprise Size] varchar(30) not null,

[Category Size] Char(1) not null,

[Num of Firms] Int not null,

[Annual Payroll (1000$)] Decimal (11,2) not null,

Constraint PK\_EstablishmentIndustry primary key(DWState),

)

Create table dbo.FACT\_Orders(

OrderID Int Not null,

DWCompany Int ,

[State] Varchar (30) Not null,

BoxID Int Not null,

OrderDate Date Not null,

[Include Packlane Plus] Bit Not null,

City Varchar (20) Not null,

Country Varchar (20) Not null,

[Production speed] Varchar (30) Not null,

[Production speed Cost] Decimal (7,2) Not null,

[Shipping method] Varchar (30) Not null,

[Shipping method Cost] Decimal (7,2) Not null,

Quantity int Not null,

[Total Price] Decimal (11,2) Not null,

[Total Shipping Price] Decimal (11,2) Not null,

[Num Of Reviews] int ,

[Average Rank] Decimal (3,2) ,

Constraint PK\_Orders primary key(OrderID),

)

Create table dbo.DWRejectedOrders(

OrderID Int ,

DWCompany Int ,

[State] Varchar (30) Not null,

BoxID Int Not null,

OrderDate Date Not null,

[Include Packlane Plus] Bit Not null,

City Varchar (20) Not null,

Country Varchar (20) Not null,

[Production speed] Varchar (30) Not null,

[Production speed Cost] Decimal (7,2) Not null,

[Shipping method] Varchar (30) Not null,

[Shipping method Cost] Decimal (7,2) Not null,

Quantity int Not null,

[Total Price] Decimal (11,2) Not null,

[Total Shipping Price] Decimal (11,2) Not null,

[Num Of Reviews] int ,

[Average Rank] Decimal (3,2) ,

Constraint PK\_DWRejectedOrders primary key(OrderID),

)

Create table dbo.OldVerCompanies(

[DW Companies] Int IDENTITY(1,1) Not null,

[Company name] Varchar (40) ,

[Business Category] Varchar (20) Not null,

[Category Size] Char (1) Not null,

Size Varchar(10) Not null,

[Type] Varchar (20) ,

ValidFrom Date,

ValidUntil Date,

Constraint PK1\_Companies primary key([DW Companies]),

)

Create procedure dbo.[updetOldVersions]

as Begin

update DIM\_Companies set ValidUntil=GETDATE()-1 where [DW Companies] in (select [DW Companies] from OldVerCompanies)

truncate table OldVerCompanies

end

create table DWPacklane.dbo.DIM\_DATE

(

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 FandangoDW.dbo.DIM\_DATE

SELECT \* FROM dim

ORDER BY TheDate

OPTION (MAXRECURSION 0);