הנדסת נתוני עתק - מטלה 1 - דו"ח מסכם

מאת: קבוצה 2 - אושרי מנדלאווי, אופק שהרבני ועידן כנת

1. במקרה של Olist, חברה המנהלת פלטפורמת מסחר אלקטרוני ומייצרת כמויות גדולות של נתונים ממקורות שונים (כגון הזמנות, פריטים, תשלומים, לקוחות, מוכרים וביקורות), הצורך המרכזי הוא **ניתוח מידע היסטורי והפקת תובנות עסקיות משמעותיות ממנו**. לכן, פתרון כמו **מחסן נתונים (Data Warehouse)** מהווה הבחירה המתאימה ביותר. בשונה ממסדי נתונים מבצעיים (OLTP), שמיועדים לניהול שוטף של פעולות כמו קליטת הזמנות ועדכון סטטוסים, מחסן נתונים תומך בתהליכי OLAP - כלומר, הוא מאפשר להריץ שאילתות חקירה מורכבות במהירות, על פני פרקי זמן ארוכים, תוך ניתוח לפי מימדים מגוונים (למשל לפי זמן, אזור גאוגרפי, קטגוריית מוצר, לקוח וכו').

מבנה ה־DW מבוסס לרוב על **סכמת כוכב (Star Schema)**, שבה קיימת טבלת עובדות מרכזית (כגון רכישות בפלטפורמה) המקושרת לטבלאות מימד כמו לקוחות, מוכרים, מוצרים וזמן. מבנה זה **מדמה את האופן שבו אנו שואלים שאלות עסקיות בפועל**: לדוגמה, "מה קטגוריות המוצרים הנמכרות ביותר באזור דרום-מזרח ב־2023?", או "אילו מוכרים מקבלים ציונים נמוכים באופן עקבי?". כדי לאפשר ביצוע מהיר של שאילתות מסוג זה, מחסן הנתונים נבנה בגישה של **דה-נורמליזציה חלקית** - כלומר, ויתור על פירוק לטבלאות קטנות ועצמאיות לטובת טבלאות רחבות יותר שמכילות מידע כברירת מחדל. כך ניתן **להימנע מ־joins כבדים ומיותרים** ולשפר משמעותית את הביצועים והפשטות של השאילתות.

בהקשר של הנתונים הנוכחיים, ה-DW מאפשר לקבץ את כל תשעת קובצי הנתונים שניתנו (מוכרים, לקוחות, מוצרים, תשלומים וכו') למבנה מאוחד ועקבי - שבו המידע זורם מתהליך ETL מסודר לתוך סכמת כוכב שמוכנה לשימוש אנליטי. זו תשתית קריטית לכל פעילות BI של החברה - מדוחות סטטיסטיים ועד ניתוחי נטישה, מיפוי צווארי בקבוק בזמני שילוח, או השוואת הכנסות לפי ערים. כל אלה מתבצעים ביעילות ובמהירות בזכות המבנה הייעודי והאופטימיזציה של מחסן הנתונים.

2. <u>דפוס העיצוב אותו בחרנו עבור טבלת העובדות:</u> הוא **Transaction Design**. דפוס זה מתאים למקרים בהם כל שורה בטבלה מייצגת **טרנזקציה בודדת** - כלומר, פעולה עסקית שמתרחשת בנקודת זמן מסוימת. במקרה שלפנינו, בחרנו להתייחס אל **המוצר בהזמנה כיחידת הטרנזקציה המרכזית**, מתוך הבנה ש**זוהי הרמה האנליטית הרלוונטית ביותר לניתוחים במטלה זו**, בהתאם לחומר הנלמד בהרצאה 3 (Design Patterns). רמת גרנולריות זו מאפשרת גמישות מספקת עם ניתוח הנתונים, יחד עם שמירה על **רמת פירוט גבוהה (grain)**, מבלי לוותר על דיוק כמו שקורה ברמות כלליות מדי (כגון לקוח, מוכר או אזור), ומאידך, תאפשר לנתח תובנות עסקיות לפי מוכרים ומוצרים, מה שלא היה מתאפשר כאשר יחידת הטרנזקציה המרכזית הייתה מיוצגת כהזמנה בודדת. בכך הוא מאפשר תמיכה ביכולת ניתוח גמישה ומדויקת יחסית של פעולות עסקיות.

מזהה ההזמנה - השילוב (order_id, order_item_id) שימש כמפתח הראשי בטבלת העובדות, מה שאיפשר שילוב של נתונים רלוונטיים ממקורות שונים - כגון פרטי פריטים, מוכרים, תשלומים וביקורות. הנתונים בטבלה זו כוללים, בין היתר: מזהי פריטים ומוצרים, מזהה מוכר, מזהה לקוח, מחירים, עלויות משלוח, תאריכי אישור ואספקה, מספר תשלומים, אמצעי תשלום וביקורת לקוח.

באמצעות מבנה זה ניתן לבצע מגוון ניתוחים ממוקדים בעלי ערך עסקי עבור חברת Olist. כמו למשל: בחינת דפוסי המכירות לפי קטגוריית מוצר (לצורך תעדוף מלאים), ניתוח הכנסות לפי מוכר (לזיהוי שותפים מובילים), מדידת זמני אספקה ממוצעים לפי אזור (לשיפור לוגיסטיקה), והשוואת שביעות רצון בין אמצעי תשלום שונים (באמצעות שילוב עם טבלת הביקורות, לצורך שיפור חוויית הלקוח). ניתוחים אלה נועדו להפיק תובנות מבוססות נתונים התומכות בקבלת החלטות עסקיות.

יתרה מזאת, הבחירה ב-Transaction Design משתלבת היטב עם הדרישה למימוש סכמת כוכב בתרגיל (Star Schema), כיוון ש**המוצר בהזמנה, שהוא יחידת הטרנזקציה בה בחרנו, משמש כמוקד טבעי ואינטואיטיבי לקישור בין כל הישויות המרכזיות**: לקוחות, מוצרים, מוכרים, תשלומים וביקורות. כך ניתן לבנות טבלת עובדות אחת סביב מזהה המוצר בהזמנה (שילוב מזהה ההזמנה עם מזהה המוצר בה), שמקושרת ישירות למימדי מפתח, באופן שמאפשר לבצע ניתוחים אנליטיים חוצי מימדים ביעילות ובבהירות.

לסיכום, הבחירה ב-Transaction Design, תוך הגדרת המוצר הבודד בהזמנה כיחידת הטרנזקציה והמפתח המרכזי בטבלת העובדות בסכמת הכוכב שלנו, היתה האופטימלית מבחינתנו - לדעתנו, בחירה זו שמרה על רמת גרנולריות מדויקת, שאיפשרה ניתוח עסקי מעמיק עבור חברת Olist, יחד עם מימוש אינטואיטיבי וברור של סכמת הכוכב בתרגיל זה.

3. א. <u>חסרונות אפשריים עיקריים של המבנה הנבחר - סכמת כוכב + Transaction Design לפי הזמנות:</u>

- **יתירות וכפילויות:** כדי לתמוך בניתוח תובנות עסקיות אנליטיות (OLAP), נדרוש מחסן נתונים המאפשר ביצוע מהיר ויעיל של שאילתות מורכבות – ובפרט כאלה שכוללות פעולות JOIN חוזרות בין טבלאות. לשם כך, כל הטבלאות במחסן הנתונים עוצבו בצורה לא מנורמלת. דה־נורמליזציה זו מפחיתה את הצורך בפעולות קישור יקרות בזמן ריצה, אך מייצרת כפילויות ויתירויות מכוונות - כמו מופעים חוזרים של מזהי מוכרים, סכומי תשלום או קטגוריות מוצרים - שעלולות להגדיל את נפח הנתונים ולדרוש תחזוקה שוטפת כדי להבטיח עקביות בין הטבלאות. הכפילויות הנ"ל גוררות חסרונות נוספים, שאליהם נתייחס בהמשך.
במסגרת המטלה, ראינו בכפילויות אלו פשרה סבירה שנובעת מהעיצוב המודע של הסכמה, כדי לאזן בין מהירות ויעילות של יישום השאילתות תוך הקרבה של המורכבות העולה בתחזוקה.
- **קושי בתחזוקה ועדכון:** הדה־נורמליזציה החלקית אשר ביצענו, שכללה איחוד נתונים מטבלאות שונות לתוך טבלת העובדות, אומנם שיפרה את הביצועים וייעלה את השאילתות האנליטיות, אך גם גררה מורכבות בתחזוקה. מאחר שחלק מהמידע מופיע ביותר ממקום אחד (למשל, מזהה לקוח או מוכר), מצב זה יוצר תלות גבוהה יותר בריצה תקינה של תהליך ה־ETL ובאחידות הלוגיקה העסקית לאורך זמן. ברמה הפרקטית, המשמעות היא צורך במעקב תחזוקתי שוטף ומדויק כדי למנוע אי־עקביות ולשמור על אמינות הנתונים.
- **תמיכה בקשרים פשוטים בלבד:** סכמת הכוכב שנבנתה תומכת בקשרים מסוג "יחיד לרבים" (N:1) בין טבלת העובדות לבין כל אחת מטבלאות המימד. מבנה זה מאפשר גישה פשוטה וישירה לניתוחים אנליטיים, אך מגביל את הגמישות במקרים שבהם נדרש לתאר קשרים מורכבים יותר - כמו למשל, קשרים בין מימדים שונים או בין ישויות שאינן מקושרות ישירות להזמנה. קשרים מסוג "רבים לרבים" (M:N), כמו ניתוח הקשר בין מוכרים ללקוחות, מחייבים פתרונות נוספים כמו טבלאות מקשרות, שאינם נתמכים ישירות במסגרת הסכמה הנוכחית.
- **חוסר יעילות בנוגע לנתונים בנפחים עצומים:** כאשר היקף הנתונים גדול במיוחד, למשל - מיליוני או עשרות מיליוני רשומות, טבלת העובדות עלולה להפוך לעמוסה מדי. הדה-נורמליזציה המודעת מבניית סכמת הכוכב תגרור כפילויות רבות, חוסר נרמול והרחבת הטבלה בעמודות שנגזרו בתהליך ה־ETL. השלכות אלו נעשות אקוטיות במיוחד כשמדובר במחסן נתונים גדול, מה שמקשה על ניתוח הנתונים ומעמיס על משאבי המערכת. במצבים כאלה, ייתכן שיידרשו פתרונות מתקדמים יותר כמו סכמת פתית שלג (Snowflake Schema) - שמטרתה לצמצם כפילויות ולשפר את ניהול הנתונים בטווח הארוך.

ב. <u>4 שלבי תכנון מחסן הנתונים, בהתאמה לנתוני Olist:</u>

1. **זיהוי התהליך העסקי:** מחסן הנתונים מתמקד בניתוח הפעילות העסקית של פלטפורמת המסחר האלקטרוני הברזילאית Olist. מטרת המחסן היא לאפשר הפקה של תובנות עסקיות מתוך נתוני הזמנות, מוצרים, לקוחות, תשלומים, משלוחים וביקורות. הנתונים עוסקים בין היתר בהכנסות, בזמן שילוח, ברמות שביעות רצון הלקוחות ובדפוסי רכישה, ומתפרסים על פני תקופה של מספר חודשים. ניתוח הנתונים מאפשר לקבל החלטות בנוגע למלאים, ביצועי מוכרים, העדפות לקוח ועוד.

2. **בחירת הגרעין (grain):** בחרנו להגדיר את המוצר הבודד בהזמנה כיחידת ה-grain בטבלת העובדות, כלומר: כל שורה מייצגת מוצר בודד בהזמנה אחת שהוזמן בפלטפורמה. רמת פירוט זו מאפשרת ניתוחים רחבים הכוללים השוואות בין לקוחות, מוכרים, אזורים, שיטות תשלום וקטגוריות מוצרים. ההחלטה התקבלה מתוך שיקול של איזון בין רזולוציה אנליטית לבין פשטות מבנית, תוך התבססות על הדרישות האנליטיות במטלה והשאילתות שנדרש לבנות.

3. **בחירת המימדים (dimensions):** בהתבסס על טבלאות הנתונים שסופקו, ולצורך מענה מיטבי של השאילתות הנדרשות, בחרנו את המימדים הבאים:

- Date Dimension: טבלת תאריכים אשר גזרנו מתוך כלל התאריכים בטבלאות ההזמנות, התשלומים והמשלוחים. הטבלה כוללת תאריך מלא (מזהה), יום בשבוע, חודש, רבעון ושנה.
- Customer Dimension: כוללת מידע דמוגרפי בסיסי על הלקוח המזמין, כולל מזהה לקוח, עיר, מדינה, שם החברה ותאריך רישום לפלטפורמה.
- Seller Dimension: מכילה פרטים על המוכר שמספק את המוצר, לרבות מזהה מוכר, עיר, מדינה ותאריך רישום לפלטפורמה.
- Product Dimension: כוללת מזהה מוצר, קטגוריית מוצר, מידות המוצר (אורך, גובה ורוחב בס"מ, משקל בגרמים), תאריך השקה ותיאור מילולי. קטגוריה זו חיונית לצורך ניתוח דפוסי רכישה לפי סוגי מוצרים.

4. **זיהוי העובדות (facts):** טבלת העובדות עוצבה סביב ההזמנות, וכללה את המדדים המרכזיים הבאים:

- order_id - מזהה ההזמנה.
- order_item_id - מס' סידורי של המוצר בהזמנה הנתונה. בשילוב order_id, מהווה מפתח ראשי לטבלה.
- price - עלות המוצר בהזמנה.
- freight – עלות המשלוח של המוצר בהזמנה.
- total_price - סכום מחירי המוצרים בהזמנה (מתוך order_items).
- total_freight - עלות המשלוח להזמנה.
- payment_value - סכום התשלום הכולל (מתוך טבלת התשלומים).
- average_review_score - ציון הביקורת (הממוצע - מיצוע נעשה לפי הביקורות מהזמנה ספציפית, על כך בהמשך..) שהוזן על ידי הלקוח (מתוך טבלת הביקורות).
- order_status - סטטוס ההזמנה.
- תאריכים רלוונטיים **ברמת ההזמנה:** תאריך רכישה (order_purchase_timestamp), תאריך אישור ההזמנה (order_approved_at), תאריך המשלוח (order_delivered_carrier_date), תאריך הגעת ההזמנה ללקוח (order_delivered_customer_date) ותאריך ההגעה המשוערך של ההזמנה ללקוח (order_estimated_delivery_date).

לצד שדות אלו כללנו גם את המפתחות הזרים לטבלאות המימד שנבחרו. חלק מהשדות נוספו לטבלת העובדות כתוצאה ממהלך דה־נורמליזציה חלקית, על מנת לאפשר ביצוע יעיל של שאילתות נפוצות, גם במחיר של יתירות מסוימת בנתונים. כללנו את עמודות אלו הרלוונטיות ברמת ההזמנה כיוון שהן כללו מידע שבעינינו ראוי היה לשמור לצורך ניתוחי המשך.

ג. <u>הנתונים בהם בחרנו להשתמש ולהשמיט:</u>

במהלך בניית מחסן הנתונים (בשלב ה-ETL בפרט), בחרנו להשתמש רק בטבלאות ובשדות שתורמים ישירות למבנה האנליטי של המערכת. זאת מתוך שיקולים של טיוב וניקוי הנתונים, שמירה על רלוונטיות עסקית, צמצום מידע אישי או טקסטואלי לא ניתוחי, והתמקדות בנתונים שיכולים לתרום ישירות לבניית מחסן נתונים איכותי, יעיל ובר־תחזוקה. ההחלטות נתקבלו לאחר בדיקה מעשית של איכות השדות, שיעור ערכים חסרים, מידת התרומה לניתוחים העסקיים, וחפיפות בין מקורות המידע. חלק מהטבלאות הושמטו לחלוטין, חלקן שולבו בצורה חלקית ומושכלת, וחלקן עברו עיבוד שכלל טרנספורמציות ואיחודים עם טבלאות אחרות.

- **טבלת המיקומים הגיאוגרפיים olist_geolocation_dataset**, שכללה פרטי מיקום לפי zip code, לא נכללה במחסן. לא היה צורך בה, שכן מידע על עיר ומדינה כבר מופיע בטבלאות הלקוחות והמוכרים, והקישור לפי קוד דואר לא תרם ערך מוסף.
- **בטבלת תרגומי הקטגוריות product_category_name_translation**, השתמשנו רק בעמודת התרגום לאנגלית. מיזגנו את המידע עם טבלת המוצרים, השמטנו את שם הקטגוריה בפורטוגזית (product_category_name), וביצענו ניקוי של ערכים חסרים וכפילויות.
- **טבלת הביקורות olist_order_reviews_dataset** עברה טרנספורמציה דומה: לאחר שניקינו את שדות הטקסט, חישבנו את הציון המספרי הממוצע (review_score) לכל הזמנה. מידע זה מוזג ישירות עם טבלת ההזמנות ושולב בטבלת העובדות.
- **טבלת ההזמנות olist_orders_dataset**, אף שהיא שימשה אותנו רבות בשלב הקדם־עיבוד, לא נשמרה כמימד נפרד במחסן. היא אוחדה ישירות עם טבלת פרטי ההזמנה (order_items) במטרה לרכז את המידע בתצורה גרנולרית אחת שתומכת בשאילתות מבוססות פרטי הזמנה.
- **גם טבלת התשלומים (olist_order_payments_dataset)** לא נשמרה כמימד עצמאי. הסרנו את העמודות: payment_sequential payment_type, payment_installments, payment_value. לא נדרשנו לנתח סוגי תשלום או אמצעי תשלום לצורך השאלות האנליטיות (למשל אשראי מול שובר וכו'). החלטנו לחשב מדדים מצומצמים מתוך הקובץ הזה, כגון num_payments_method - חישוב מספר אמצעי התשלום השונים עבור כל הזמנה. הכנסנו אותם כעמודת מידה בטבלת העובדות, במקום להכיל את כל העמודות כמימד נפרד. הממצאים הרלוונטיים אוחדו בתוך טבלת העובדות.

בנוסף לכך, **בתוך טבלאות שנכנסו למחסן**, השמטנו שדות מיותרים או פרטיים מדי:

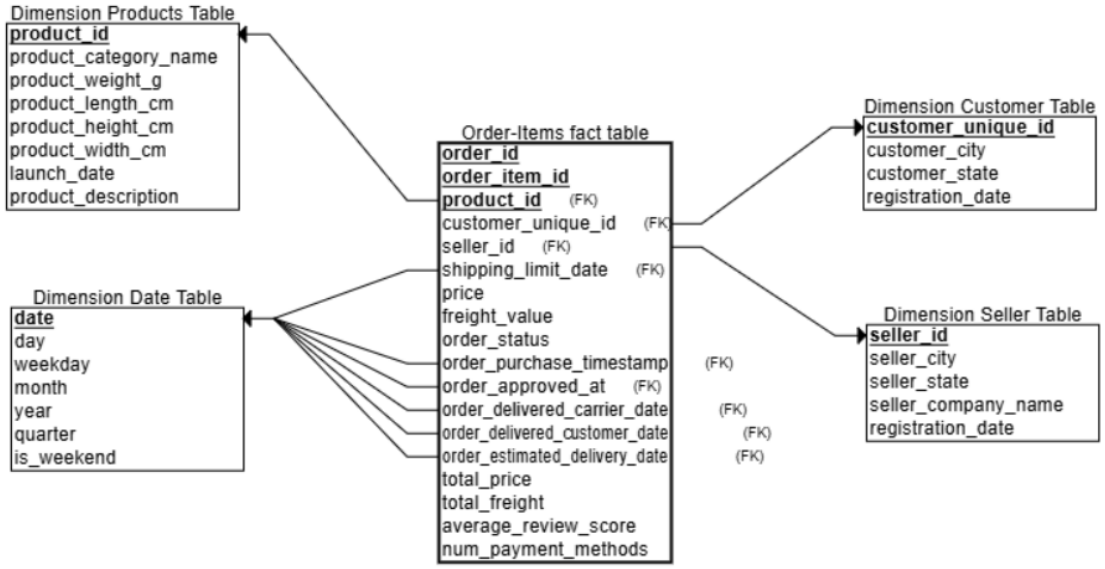
- **בטבלת הלקוחות** - הושמטו שדות כמו zip_code_prefix ו־customer_id, וביצענו טרנספורמציות על מנת לשמור רק customer_unique_id (מזהה הלקוח האמיתי) על חשבון customer_id (ששימש לקישור עם טבלת ההזמנות ולא לזיהוי הלקוח), יחד עם העיר והמדינה. בנוסף, השמטנו שדות הקשורים לפרטים אישיים של הלקוח - מייל, שם פרטי ושם משפחה משום

שמדובר במידע פרטי שאינו נחוץ למחסן הנתונים ומהווה עומס מיותר. העמודה customer_zip_code_prefix הוסרה מאחר והיא מספקת מידע גרנולרי שאינו נדרש לניתוח ברמה הלאומית או האזורית, ובמילא קיימים שדות customer_city ו־customer_state. כמו כן, בעקבות אחוז גבוה יחסית של ערכים חסרים וחוסר רלוונטיות לניתוחי המשך, החלטנו להסיר את עמודת region. הנחנו שלא רלוונטי לשמור אותו כל עוד שמרנו עמודות אחרות הרלוונטיות למיקום כמו מחוז ועיר.

- **בטבלת המוכרים** - לשם העקביות ובדומה לנעשה בטבלת לקוחות (וגם משיקולים דומים), לא השתמשנו ב־region, zip code prefix או כתובת מייל, אלא רק ב־seller_id, עיר, מדינה ותאריך הרשמה.
- **בטבלת המוצרים** - לא נעשה שימוש בשדות טקסטואליים כמו product_name_length, product_description_length, או product_photos_qty.
- **בטבלת התשלומים** – הושמטו payment_sequential ושדות טכניים נוספים, לאחר שנעשה שימוש בניתוחים מקדימים.

החלטות ההשמטה נועדו ליצור מחסן נתונים רזה, ממוקד ואפקטיבי יותר לניתוחים עסקיים, ללא עודף מידע טקסטואלי או מזהים שאינם מוסיפים תובנה. בכך הבטחנו שמירה על יעילות, פשטות ויכולת הרצת שאילתות מהירה במחסן.

ד. <u>תרשים סכמתי (ERD) של מחסן הנתונים:</u>



4. א. <u>שלב ה־ETL (Extract, Transform, Load):</u> שבוצע בפרויקט התמקד בטיוב הנתונים, סינונם והתאמתם למבנה של מחסן נתונים איכותי, יציב וניתן לתחזוקה. ההחלטות התקבלו לאחר ניתוח שיטתי של כל טבלה - כולל תיאור כללי (info), צפייה ב־50 שורות ראשונות, בדיקת שיעורי ערכים חסרים, זיהוי כפילויות, איתור

שורות עם מזהים חסרים, בדיקת פרופורציות של חריגים וזיהוי חוסר עקביות לוגית. כל התאריכים הומרו לפורמט datetime אחיד כדי לאפשר חישובי זמן ולוגיקה כרונולוגית מדויקת בהמשך. שלב זה מתחלק למספר תתי שלבים עיקריים:

1. חילוץ העמודות הרלוונטיות - צוין בסעיף 3ב'

2. הטרנספורמציות אשר עשינו לטיוב הנתונים: במהלך שלב הטרנספורמציות בוצעו טרנספורמציות רבות:

ניקוי כפילויות וערכים חסרים:

- הוסרו שורות עם מזהים חסרים בטבלאות.
- הוסרו כפילויות.
- הוסרו רשומות של לקוחות, מוכרים ומוצרים שמופיעים בהזמנות אך לא קיימים בטבלאות היסוד. פעולה זו נועדה לשמור על עקביות ייחוס בין טבלאות ולמנוע קישורים שגויים.
- בטבלת reviews, כאשר נמצאו מספר ביקורות להזמנה - חושב ממוצע review_score לכל הזמנה או נשמרה הביקורת המוקדמת ביותר, על מנת לייצר ייצוג עקבי ואחיד להזמנות.

סטנדרטיזציה של טקסטים:

- customer_state הומר לאותיות גדולות, בעוד customer_city עבר ניקוי והורדה לאותיות קטנות להסרת סימנים חריגים. השיטה יושמה גם בטבלת המוכרים - כולל על שמות ערים ופרובינציות - לשם אחידות במיזוגים וחיפושים.

טיפול בערכים בלתי הגיוניים:

- מוצרים בהזמנה עם price שלילי או freight שלילי סומנו כבלתי תקינים והוסרו - ערכים אלה מנוגדים להגיון עסקי.
- בטבלת products, הוסרו מוצרים עם ממדים שליליים - לדוגמה, גובה או משקל שלילי - מאחר שמדובר בנתונים שגויים.

בדיקות לוגיות בין תאריכים - בוצעו תיקונים סלקטיביים במקרים הבאים:

- הזמנות שסומנו כ־shipped אך אין להן תאריך משלוח - סומנו כ־NaT ונשמרו כחריגים, ללא השערה מחדש, כדי לשמור על שלמות הנתונים.
- הזמנות שסומנו כ־delivered אך חסרים להן תאריכים קריטיים (רכישה, אישור או משלוח) - עודכנו באופן ממוקד ל־NaT רק בשדות הבעייתיים.
- תוקנו מקרים של סדר כרונולוגי שגוי (רכישה אחרי אישור, אישור אחרי משלוח וכו') ע"י שינוי התאריכים הלא תקינים ל־NaT בלבד.
- סטטוס לא עקבי (למשל הזמנות שאושרו אך עדיין ב־processing) תוקן ל־approved.
- הזמנות עם תאריך משלוח אך ללא סטטוס מתאים תוקנו ל־shipped.
- הזמנות עם תאריך קבלה אצל הלקוח אך ללא סטטוס מתאים תוקנו ל־delivered.

אי התאמות ברישום משתמשים ומוכרים:

- מוכרים שביצעו מכירות לפני תאריך ההרשמה שלהם - במידה ויותר מ־50% מהמקרים היו כאלו, הסרנו את תאריך ההרשמה (כדי לא ליצור עיוותים כרונולוגיים).
- בוצעה בדיקה זהה על לקוחות עם דפוסים בעייתיים - תאריכי הרשמה הוסרו גם עבורם.

טרנסאקציות בזמן שגוי:

- תשלומים עם transaction_date המוקדם מתאריך הרכישה או האישור - עקב שכיחות גבוהה של מקרים אלה, הוחלט להסיר את עמודת transaction_date לחלוטין.

איחודים וטרנספורמציות נוספות:

- בטבלת order_items חושבו total_price ו־total_freight לפי order_id, ואוחדו לטבלת orders כדי לרכז מידע כספי.
- טבלת payments עברה צמצום - חושבה עמודת num_payment_methods לפי order_id, והממצאים אוחדו בטבלת orders.
- טבלת orders אוחדה עם טבלת reviews לאחר חישוב review_score ממוצע להזמנה.
- שם הקטגוריה באנגלית הוזן במקום המקור בפורטוגזית, בטבלת המוצרים, כחלק ממיזוג עם טבלת product_category_name_translation - פעולה זו בוצעה כדי להקל על ניתוח באנגלית ולמנוע כפילויות.
- נבנתה טבלת dim_date הכוללת את כל התאריכים הקיימים לאחר המרה לפורמט datetime.
- השדות הרלוונטיים (כולם) בטבלת ההזמנות (orders) אוחדו לתוך טבלת העובדות הסופית fact_order_items - מתוך הנחה שהפרטים הללו ברובם נומריים וקריטיים לניתוח עסקי, כך שבאמצעות הכנסתם לטבלה זו נוכל להקל על הניתוחים האנליטיים.

ב. דה-נורמליזציה ביישום ה-DW שלנו - מקרים בולטים:

- **ציון הביקורת (review_score):**

  בטבלת הביקורות הופיעו מספר ביקורות עבור חלק מההזמנות. כדי למנוע טבלת קשר נוספת בין הזמנות לביקורות, חושב ממוצע הציון לכל הזמנה, והמשתנה אוחד ישירות לטבלת ההזמנות ולאחר מכן הועבר לטבלת העובדות.

  בכך נשמר משתנה כמותי מסכם ברמת ההזמנה מבלי לשמר את הטקסטים הגולמיים או את מבנה היחס של קשר יחיד-רבים.
- **מספר אמצעי תשלום להזמנה (num_payment_methods):**

  במקום לשמר את טבלת התשלומים כטבלת מימד בפני עצמה, חושב סיכום מספרי עבור כל הזמנה לפי מספר השורות שלה בטבלת התשלומים, והמשתנה אוחד ישירות לטבלת העובדות.

  באופן זה נמנענו מהחזקת רשומות תשלום מפורטות, תוך שמירה על אינדיקטור בעל ערך ניתוחי.
- **איחוד נתוני הזמנה לכל שורת מוצר (fact_order_items):**

  בטבלת העובדות, שכללה את פרטי המוצרים שהוזמנו (order_items), מוזגו לתוכה עמודות מרמת

ההזמנה – כולל תאריך רכישה, סטטוס הזמנה, ציון הביקורת, מספר אמצעי תשלום, סך כל המחיר (total_price) ודמי המשלוח הכוללים (total_freight).
איחוד זה גרם לריבוי ערכים זהים בשורות רבות (למשל אותו review_score מופיע עבור כל מוצר באותה הזמנה), אך מאפשר ביצוע שאילתות גמישות ומיידיות לפי כל משתנה אפשרי, גם ברמת המוצר, גם ברמת הלקוח וגם ברמת ההזמנה – ללא צורך ב־JOIN-ים מורכבים.

יתרון מובהק שדה־נורמליזציה מאפשרת הוא **הרצת שאילתות מהירה יותר**, שכן כל המידע מרוכז בטבלה אחת ללא צורך בפעולות JOIN מרובות וכבדות, מה שיכול להקל במיוחד עבור ניתוח עסקי והשגת תובנות אנליטיות משאילתות, במיוחד כשמדובר ב-DW המיועד גם עבור מטרה זו.

א. <u>השאילתות:</u>
א. שאילתה 1:

```
# QUERY 1 - This query identifies sellers who had orders with at least 2 different products
# and more than 1 item in total, and shows their total number of such orders
# along with the date of their latest qualifying order.

SELECT
    s.seller_company_name,                                  # Name of the seller
    COUNT(DISTINCT f.order_id) AS qualifying_orders,        # Number of qualified orders per seller
    MAX(f.order_purchase_timestamp) AS last_order_date      # Date of the most recent qualifying order
FROM fact_order_items f
JOIN dim_sellers s ON f.seller_id = s.seller_id             # Join with seller dimension for names
WHERE f.order_id IN (
    # Filter to orders that:
    # 1. Contain at least 2 distinct products
    # 2. Contain more than 1 item (order_item_id > 1 means multi-item)
    SELECT order_id
    FROM fact_order_items
    GROUP BY order_id
    HAVING COUNT(DISTINCT product_id) >= 2
       AND MAX(order_item_id) > 1
)
GROUP BY s.seller_company_name;                             # Group results by seller name
```

ב. שאילתה 2:

```sql
# QUERY 2 - This query finds the top 5 cities by total payment amount,
# excluding orders where any single item exceeded a total of 2000 (price + freight).
# Only considers orders with a review score of at least 4.

SELECT
    c.customer_city,                                        # Customer's city
    SUM(f.total_price + f.total_freight) AS total_payment   # Sum of payments including freight
FROM fact_order_items f
JOIN dim_customers c ON f.customer_unique_id = c.customer_unique_id  # Join with customers for location
WHERE f.order_id NOT IN (
    # Exclude any order that had a single item whose cost (in total) over 2000
    SELECT order_id
    FROM fact_order_items
    WHERE price + freight_value > 2000
)
AND f.average_review_score >= 4                             # Include only well-reviewed orders
GROUP BY c.customer_city
ORDER BY total_payment DESC                                 # Sort from highest to lowest total payment
LIMIT 5;                                                    # Keep only top 5 cities
```

ב. <u>פלטי השאילתות:</u>

א. שאילתה 1 (בפועל יש יותר שורות, מדובר בצילום מסך מהפלט..)

seller_company_name	seller_id	qualifying_orders	last_order_date
Morris-Sullivan	1025f0e2d44d7041d6cf58b6550e0bfa	13	2018-08-09 21:26:00
Peterson Ltd	1f50f920176fa81dab994f9023523100	13	2018-05-19 14:35:00
Bryant Inc	1900267e848ceeba8fa32d80c1a5f5a8	12	2018-07-17 23:36:00
Ross-Gregory	da8622b14eb17ae2831f4ac5b9dab84a	10	2018-08-07 11:29:00
Estrada and Sons	d2374cbcbb3ca4ab1086534108cc3ab7	8	2018-07-14 00:04:00
Hill, Fox and Johnson	1835b56ce799e6a4dc4eddc053f04066	8	2018-08-07 16:33:00
Williams, Sullivan and Roberts	391fc6631aebcf3004804e51b40bcf1e	8	2018-02-22 11:33:00
Barber-Oliver	7c67e1448b00f6e969d365cea6b010ab	6	2018-04-22 18:56:00
Kelly-Liu	cca3071e3e9bb7d12640c9fbe2301306	5	2017-12-01 18:23:00
Rodgers and Sons	4a3ca9315b744ce9f8e9374361493884	5	2018-02-22 11:33:00
Chambers Inc	3d871de0142ce09b7081e2b9d1733cb1	4	2018-01-25 16:22:00

ב. שאילתה 2:

customer_city	total_payment
sao paulo	619352.4400000011
RIO DE JANEIRO	252545.04000000088
BELO HORIZONTE	108496.51000000002
brasilia	97298.52999999994
CURITIBA	70984.10999999999