```sql
CREATE DATABASE olist;

USE olist;

DROP DATABASE olist;

CREATE DATABASE olist;

USE olist;


CREATE TABLE customers (

  customer_id VARCHAR(50) PRIMARY KEY,

  customer_unique_id VARCHAR(50),

  customer_zip_code_prefix INT,

  customer_city VARCHAR(100),

  customer_state CHAR(2)

);

CREATE TABLE orders (

order_id VARCHAR(50) PRIMARY KEY,

customer_id VARCHAR(50), order_status VARCHAR(20), order_purchase_timestamp DATETIME,
order_approved_at DATETIME,

order_delivered_carrier_date DATETIME, order_delivered_customer_date DATETIME,
order_estimated_delivery_date DATETIME,

FOREIGN KEY (customer_id) REFERENCES customers(customer_id)

);


CREATE TABLE products (

product_id VARCHAR(50) PRIMARY KEY, product_category_name VARCHAR(100),
product_name_length INT, product_description_length INT, product_photos_qty INT,

product_weight_g INT, product_length_cm INT, product_height_cm INT, product_width_cm INT

);


CREATE TABLE sellers (

seller_id VARCHAR(50) PRIMARY KEY,

seller_zip_code_prefix INT, seller_city VARCHAR(100), seller_state CHAR(2)

);
```

```sql
CREATE TABLE order_items ( order_id VARCHAR(50), order_item_id INT, product_id VARCHAR(50),
seller_id VARCHAR(50),

shipping_limit_date DATETIME, price DECIMAL(10,2),

freight_value DECIMAL(10,2), PRIMARY KEY (order_id, order_item_id),

FOREIGN KEY (order_id) REFERENCES orders(order_id), FOREIGN KEY (product_id) REFERENCES
products(product_id), FOREIGN KEY (seller_id) REFERENCES sellers(seller_id)

);


CREATE TABLE order_payments ( order_id VARCHAR(50), payment_sequential INT, payment_type
VARCHAR(20), payment_installments INT, payment_value DECIMAL(10,2),

PRIMARY KEY (order_id, payment_sequential), FOREIGN KEY (order_id) REFERENCES
orders(order_id)

);


CREATE TABLE order_reviews (

review_id VARCHAR(50) PRIMARY KEY,

order_id VARCHAR(50), review_score INT, review_comment_title TEXT, review_comment_message
TEXT, review_creation_date DATETIME,

review_answer_timestamp DATETIME,

FOREIGN KEY (order_id) REFERENCES orders(order_id)

);


CREATE TABLE geolocation ( geolocation_zip_code_prefix INT, geolocation_lat DECIMAL(10,8),
geolocation_lng DECIMAL(10,8), geolocation_city VARCHAR(100), geolocation_state CHAR(2)

);


CREATE TABLE product_category_name_translation ( product_category_name VARCHAR(100),
product_category_name_english VARCHAR(100)

);


LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/olist_customers_dataset.csv'

INTO TABLE customers FIELDS TERMINATED BY ',' ENCLOSED BY '"'
```

```sql
LINES TERMINATED BY '\n'

IGNORE 1 ROWS

(customer_id, customer_unique_id, customer_zip_code_prefix, customer_city, customer_state);


LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/olist_orders_dataset.csv'

INTO TABLE orders

FIELDS TERMINATED BY ',' ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS

(order_id, customer_id, order_status, @order_purchase_timestamp, @order_approved_at,
@order_delivered_carrier_date, @order_delivered_customer_date,
@order_estimated_delivery_date)

SET

order_purchase_timestamp = NULLIF(@order_purchase_timestamp,''), order_approved_at =
NULLIF(@order_approved_at,''), order_delivered_carrier_date =
NULLIF(@order_delivered_carrier_date,''), order_delivered_customer_date =
NULLIF(@order_delivered_customer_date,''), order_estimated_delivery_date =
NULLIF(@order_estimated_delivery_date,'');


-- Products

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/olist_products_dataset.csv'

INTO TABLE products FIELDS TERMINATED BY ',' ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS

(product_id, product_category_name, @product_name_length, @product_description_length,
@product_photos_qty, @product_weight_g, @product_length_cm, @product_height_cm,
@product_width_cm)

SET

product_name_length = NULLIF(@product_name_length,''), product_description_length =
NULLIF(@product_description_length,''), product_photos_qty = NULLIF(@product_photos_qty,''),

product_weight_g = NULLIF(@product_weight_g,''), product_length_cm =
NULLIF(@product_length_cm,''), product_height_cm = NULLIF(@product_height_cm,''),
product_width_cm = NULLIF(@product_width_cm,'');
```

```sql
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/olist_sellers_dataset.csv'

INTO TABLE sellers

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS

(seller_id, seller_zip_code_prefix, seller_city, seller_state);


LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/olist_order_items_dataset.csv'

IGNORE INTO TABLE order_items

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS

(order_id, order_item_id, product_id, seller_id, @shipping_limit_date, @price, @freight_value)

SET

  shipping_limit_date = NULLIF(@shipping_limit_date,''),

  price = NULLIF(@price,''),

  freight_value = NULLIF(@freight_value,'');


USE olist;


LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/olist_order_items_dataset.csv'

REPLACE

INTO TABLE order_items

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS

(order_id, order_item_id, product_id, seller_id, @shipping_limit_date, @price, @freight_value)
```

```sql
SET

  shipping_limit_date = NULLIF(@shipping_limit_date,''),

  price = NULLIF(@price,''),

  freight_value = NULLIF(@freight_value,'');


  USE olist;


-- Step 1: Temporarily change datetime columns to VARCHAR

ALTER TABLE order_reviews

MODIFY review_creation_date VARCHAR(50),

MODIFY review_answer_timestamp VARCHAR(50);


-- Step 2: Load the CSV safely

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/olist_order_reviews_dataset.csv'

REPLACE

INTO TABLE order_reviews

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS

(review_id, order_id, review_score, review_comment_title, review_comment_message,
@review_creation_date, @review_answer_timestamp)

SET

  review_creation_date = NULLIF(@review_creation_date, ''),

  review_answer_timestamp = NULLIF(@review_answer_timestamp, '');


-- Step 3: Convert strings to proper DATETIME, invalid values become NULL

UPDATE order_reviews

SET review_creation_date = STR_TO_DATE(review_creation_date, '%Y-%m-%d %H:%i:%s')

WHERE STR_TO_DATE(review_creation_date, '%Y-%m-%d %H:%i:%s') IS NOT NULL;
```

```sql
UPDATE order_reviews

SET review_answer_timestamp = STR_TO_DATE(review_answer_timestamp, '%Y-%m-%d %H:%i:%s')

WHERE STR_TO_DATE(review_answer_timestamp, '%Y-%m-%d %H:%i:%s') IS NOT NULL;


-- Step 4: Convert columns back to DATETIME

ALTER TABLE order_reviews

MODIFY review_creation_date DATETIME,

MODIFY review_answer_timestamp DATETIME;


LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/olist_geolocation_dataset.csv'

INTO TABLE geolocation

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS

(geolocation_zip_code_prefix, @lat, @lng, geolocation_city, geolocation_state)

SET

    geolocation_lat = CASE

            WHEN @lat REGEXP '^-?[0-9]+(\.[0-9]+)?$' THEN CAST(@lat AS DECIMAL(10,8))

            ELSE NULL

        END,

    geolocation_lng = CASE

            WHEN @lng REGEXP '^-?[0-9]+(\.[0-9]+)?$' THEN CAST(@lng AS DECIMAL(11,8))

            ELSE NULL

        END;
```

```sql
-- ==============================
-- Task 6: Monthly Sales Trend Analysis
-- ==============================


-- Step 1: Aggregate revenue and order volume per month
SELECT
    YEAR(o.order_purchase_timestamp) AS order_year,
    MONTH(o.order_purchase_timestamp) AS order_month,
    COUNT(DISTINCT o.order_id) AS total_orders,
    SUM(oi.price) AS total_revenue
FROM
    orders o
JOIN
    order_items oi
    ON o.order_id = oi.order_id
WHERE
    o.order_purchase_timestamp IS NOT NULL
GROUP BY
    YEAR(o.order_purchase_timestamp),
    MONTH(o.order_purchase_timestamp)
ORDER BY
    YEAR(o.order_purchase_timestamp),
    MONTH(o.order_purchase_timestamp);
```