

W2_case_study

March 31, 2025

```
[ ]: /*markdown
# Exercise 9.1: Preliminary Data Collection Using SQL Techniques
This exercise collects preliminary data. We will load the database, list_
↳ scooter product details, extract product IDs, and store the results in a new_
↳ table.
```

```
[ ]: */
```

```
[ ]: /*markdown
```

1 Step 1: Load the sqllda database

```
[ ]: */
```

```
[ ]: /*markdown
```

```
[ ]: psql sqllda
```

```
[ ]: */
```

```
[ ]: /*markdown
```

2 Step 2: List the model, base_msrp, and production_start_date for scooter products

```
[ ]: */
```

```
[ ]: SELECT model, base_msrp, production_start_date
FROM products
WHERE product_type = 'scooter';
```

```
[ ]:
```

```
[ ]: /*markdown
## Step 3: Extract the model name and product IDs for scooter products
```

```
[ ]: */
```

```
[ ]: SELECT model, product_id  
FROM products  
WHERE product_type = 'scooter';
```

```
[ ]:
```

```
[ ]: /*markdown  
## Step 4: Insert the above results into a new table called product_names
```

```
[ ]: */
```

```
[ ]: SELECT model, product_id  
INTO product_names  
FROM products  
WHERE product_type = 'scooter';
```

```
[ ]:
```

```
[ ]: /*markdown
```

```
[ ]: # Exercise 9.2: Extracting the Sales Information  
In this exercise we join sales data with the product names and then isolate BatU  
↪ Scooter sales.
```

```
[ ]: */
```

```
[ ]: /*markdown
```

3 Step 1: Load the sqlda database

```
[ ]: */
```

```
[ ]: /*markdown
```

```
[ ]: psql sqlda
```

```
[ ]: */
```

```
[ ]: /*markdown
```

4 Step 2: List the available fields in the database

```
[ ]: */
```

```
[ ]: /*markdown
```

5 Step 3: Create a new table (products_sales) by joining sales and product_names on product_id

```
[ ]: */
```

```
[ ]: SELECT model, customer_id, sales_transaction_date, sales_amount, channel,
      ↳ dealership_id
      INTO products_sales
      FROM sales
      INNER JOIN product_names
        ON sales.product_id = product_names.product_id;
```

```
[ ]:
```

```
[ ]: /*markdown
    ## Step 4: Display the first five rows of products_sales
```

```
[ ]: */
```

```
[ ]: SELECT *
      FROM products_sales
      LIMIT 5;
```

```
[ ]:
```

```
[ ]: /*markdown
    ## Step 5: Retrieve Bat Scooter sales ordered by sales_transaction_date
```

```
[ ]: */
```

```
[ ]: SELECT *
      FROM products_sales
      WHERE model = 'Bat'
      ORDER BY sales_transaction_date;
```

```
[ ]:
```

```
[ ]: /*markdown
    ## Step 6: Count the number of Bat Scooter sales records
```

```
[ ]: */
```

```
[ ]: SELECT COUNT(model)
      FROM products_sales
```

```
WHERE model = 'Bat';
```

```
[ ]:
```

```
[ ]: /*markdown  
## Step 7: Determine the last sale date for the Bat Scooter
```

```
[ ]: */
```

```
[ ]: SELECT MAX(sales_transaction_date)  
FROM products_sales  
WHERE model = 'Bat';
```

```
[ ]:
```

```
[ ]: /*markdown  
## Step 8: Insert Bat Scooter sales records into a new table (bat_sales)   
↳ ordered by date
```

```
[ ]: */
```

```
[ ]: SELECT *  
INTO bat_sales  
FROM products_sales  
WHERE model = 'Bat'  
ORDER BY sales_transaction_date;
```

```
[ ]:
```

```
[ ]: /*markdown  
## Step 9: Remove the time information in bat_sales (convert to date)
```

```
[ ]: */
```

```
[ ]: UPDATE bat_sales  
SET sales_transaction_date = DATE(sales_transaction_date);
```

```
[ ]:
```

```
[ ]: /*markdown  
## Step 10: Display the first five records of bat_sales ordered by date
```

```
[ ]: */
```

```
[ ]: SELECT *  
FROM bat_sales  
ORDER BY sales_transaction_date  
LIMIT 5;
```

```
[ ]:
[ ]: /*markdown
[ ]: ## Step 11: Create bat_sales_daily table with daily sales count
[ ]: */
[ ]: SELECT sales_transaction_date, COUNT(sales_transaction_date)
[ ]: INTO bat_sales_daily
[ ]: FROM bat_sales
[ ]: GROUP BY sales_transaction_date
[ ]: ORDER BY sales_transaction_date;
[ ]:
[ ]: /*markdown
[ ]: # Activity 9.1: Quantifying the Sales Drop
[ ]: Here we compute a cumulative sum of daily sales, apply a 7-day lag, and
[ ]: ↪ calculate the growth rate (volume).
[ ]: */
[ ]: /*markdown
```

6 Step 1: Load the sqlda database

```
[ ]: psql sqlda
[ ]: */
[ ]: /*markdown
[ ]: ## Step 2: Compute the daily cumulative sum of sales and insert into
[ ]: ↪ bat_sales_growth
[ ]: */
[ ]: SELECT *, sum(count) OVER (ORDER BY sales_transaction_date) AS cumulative_sum
[ ]: INTO bat_sales_growth
[ ]: FROM bat_sales_daily;
[ ]:
[ ]: /*markdown
[ ]: ## Step 3: Compute a 7-day lag of the cumulative sum and insert into
[ ]: ↪ bat_sales_daily_delay
[ ]: */
```

```
[ ]: SELECT *, lag(cumulative_sum, 7) OVER (ORDER BY sales_transaction_date) AS lag_value
      INTO bat_sales_daily_delay
      FROM bat_sales_growth;
```

```
[ ]:
```

```
[ ]: /*markdown
      ## Step 4: Inspect the first 15 rows of bat_sales_daily_delay
```

```
[ ]: */
```

```
[ ]: SELECT *
      FROM bat_sales_daily_delay
      LIMIT 15;
```

```
[ ]:
```

```
[ ]: /*markdown
      ## Step 5: Compute sales growth as a percentage and insert into bat_sales_delay_vol
```

```
[ ]: */
```

```
[ ]: SELECT *, (cumulative_sum - lag_value) / lag_value AS volume
      INTO bat_sales_delay_vol
      FROM bat_sales_daily_delay;
```

```
[ ]:
```

```
[ ]: /*markdown
      ## Step 6: Display the first 22 records of bat_sales_delay_vol
```

```
[ ]: */
```

```
[ ]: SELECT *
      FROM bat_sales_delay_vol
      LIMIT 22;
```

```
[ ]:
```

```
[ ]: /*markdown
      #### lag_value
      -----
      */
```