Data Science

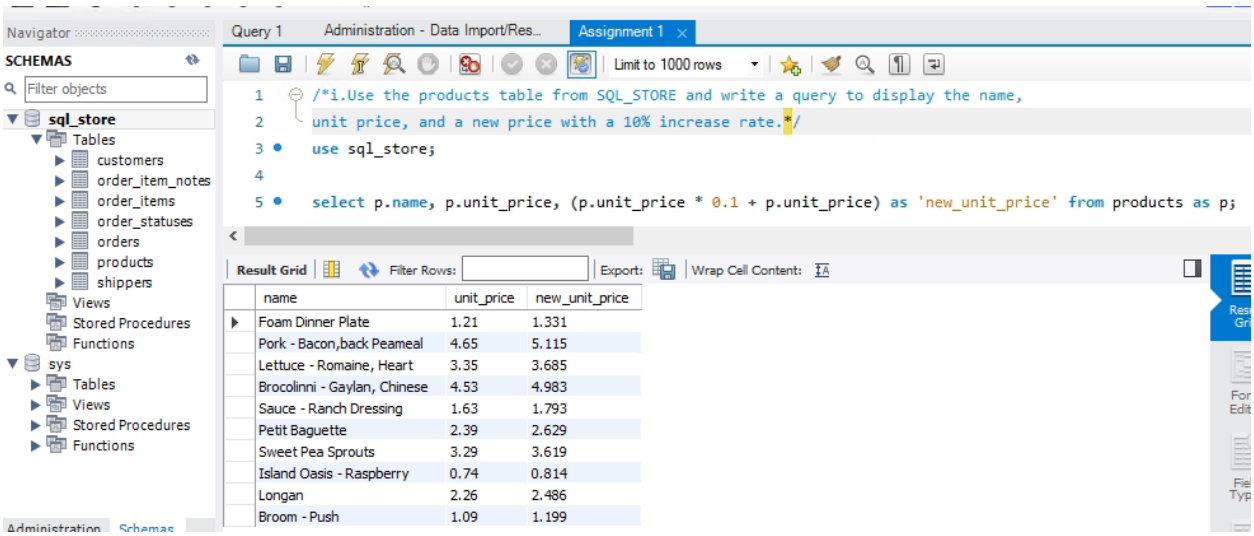
Data Mining Techniques

DQL Queries

Task 1:

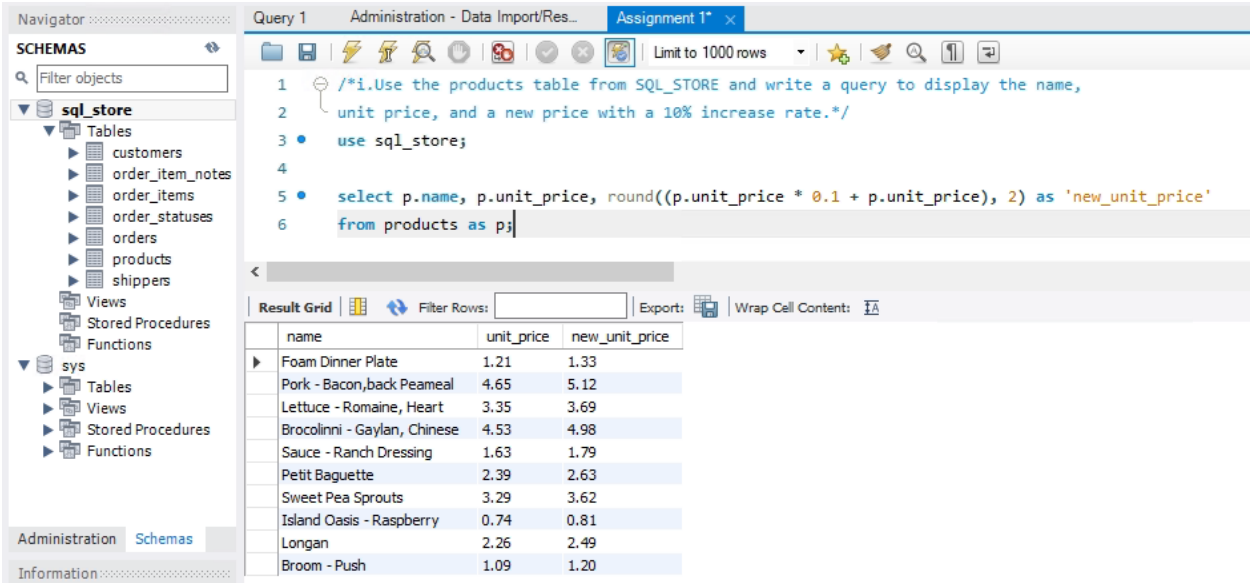
DQL Queries

i. Use the products table from SQL_STORE and write a query to display the name, unit price, and a new price with a 10% increase rate.

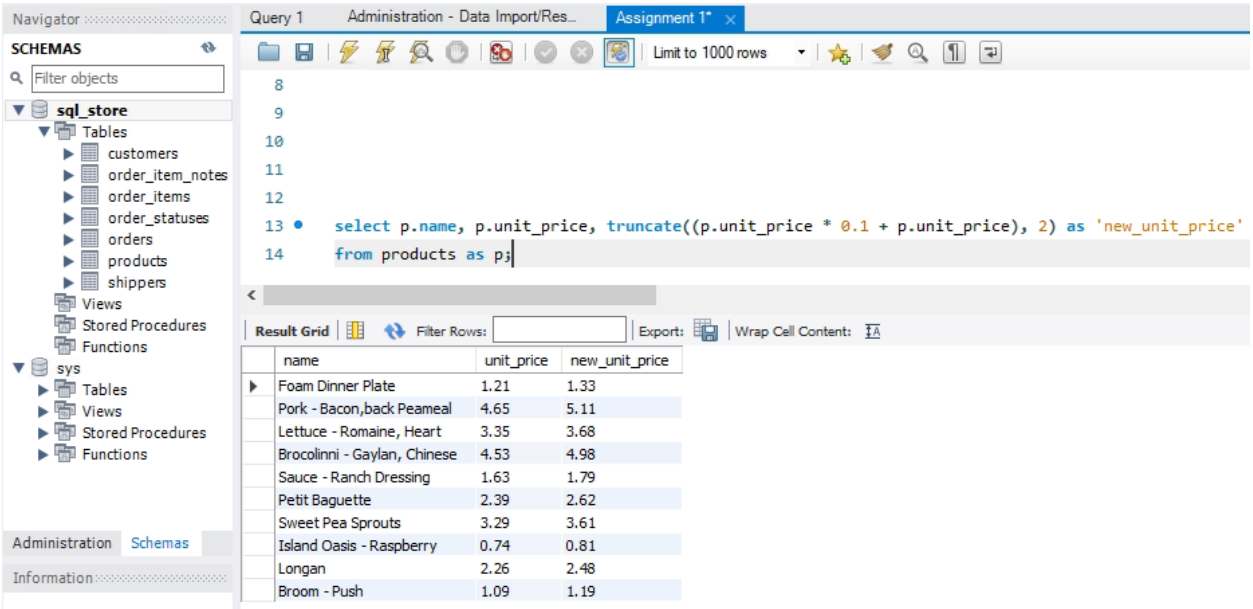


The round() function can be used to create a unit price with two decimal places. Additionally, this will round decimal places. For instance, the round() function changed the ‘Petit Baguette's unit

price from 2.629 to 2.63.

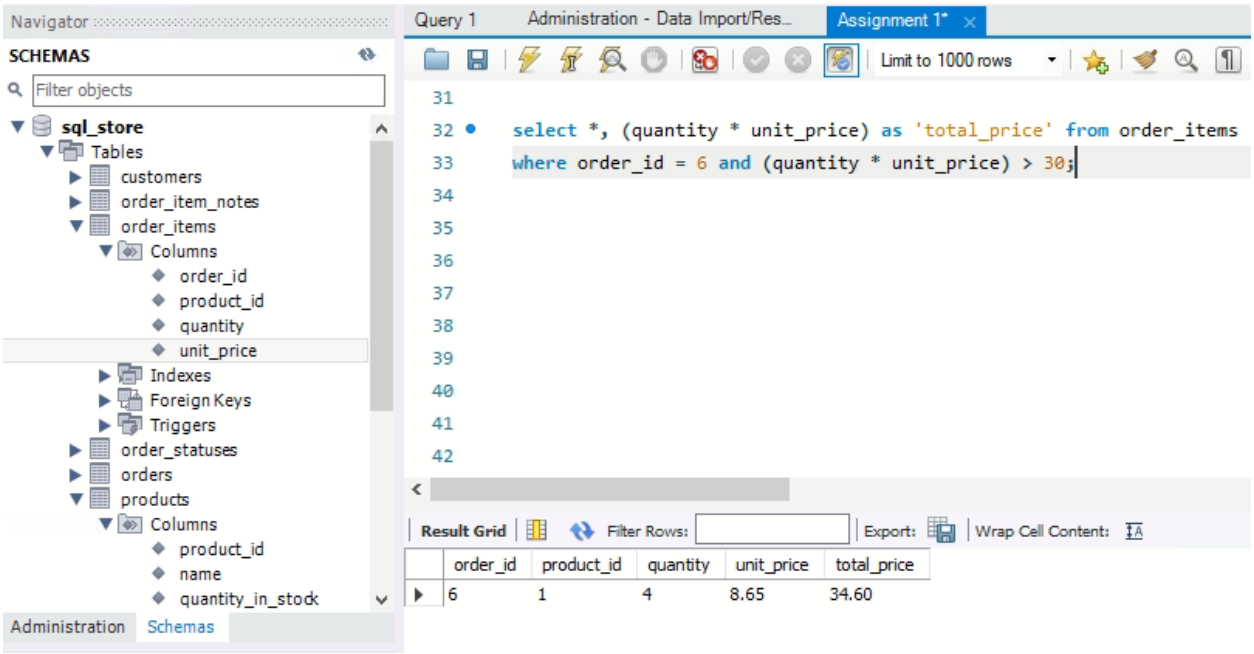


It is even feasible to obtain the true two decimal places without rounding by using the truncate() function. The 'Petit Baguette' product had a unit price of 2.629; however, after utilizing the truncate() function, it changed to 2.62.

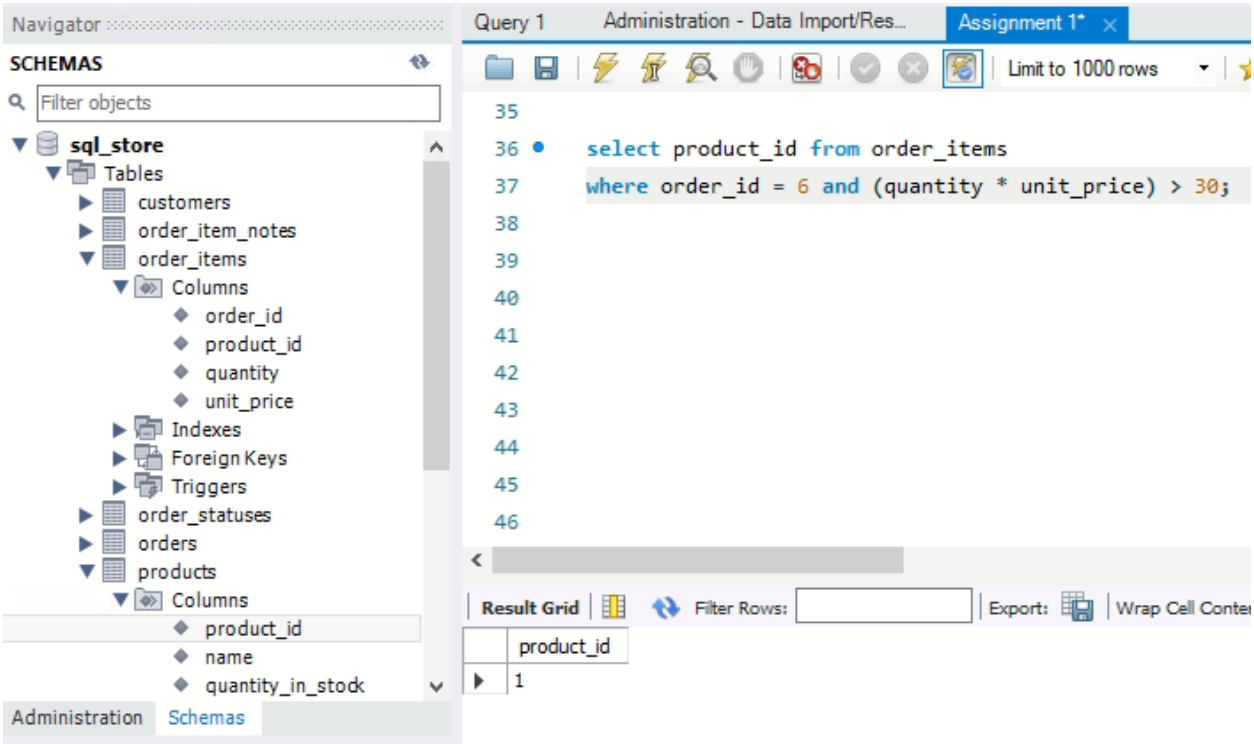


ii. Write a query to get the items for order #6 with a total price of over 30. Use the ORDER_ITEMS table from SQL_STORE database.

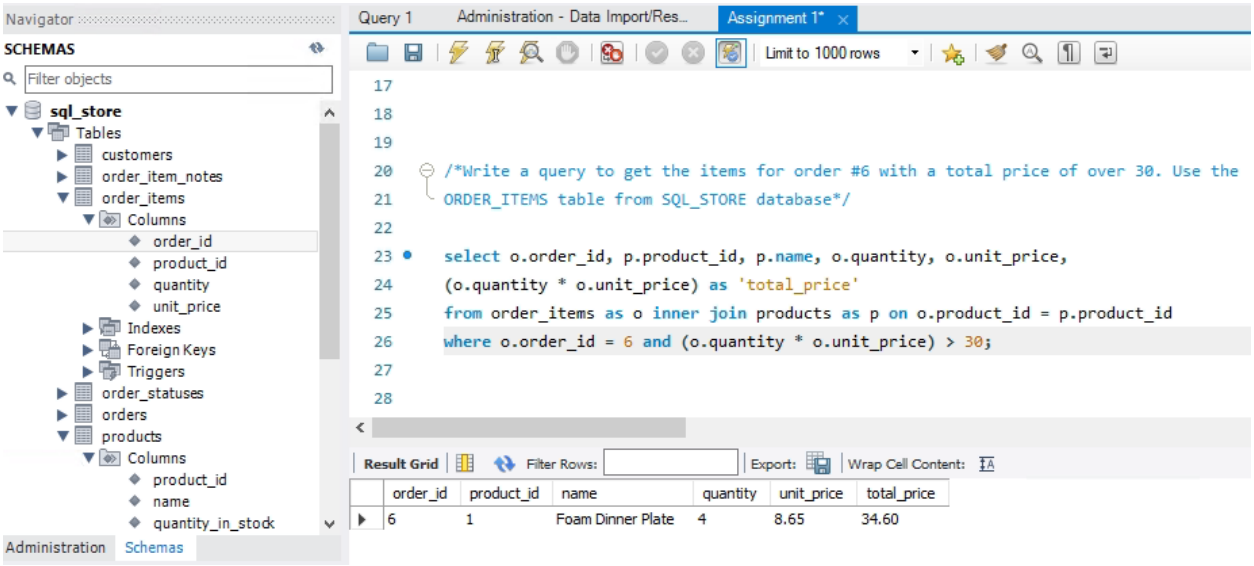
The order_items table returns four default columns and one additional custom column called total_price when the query below is run.



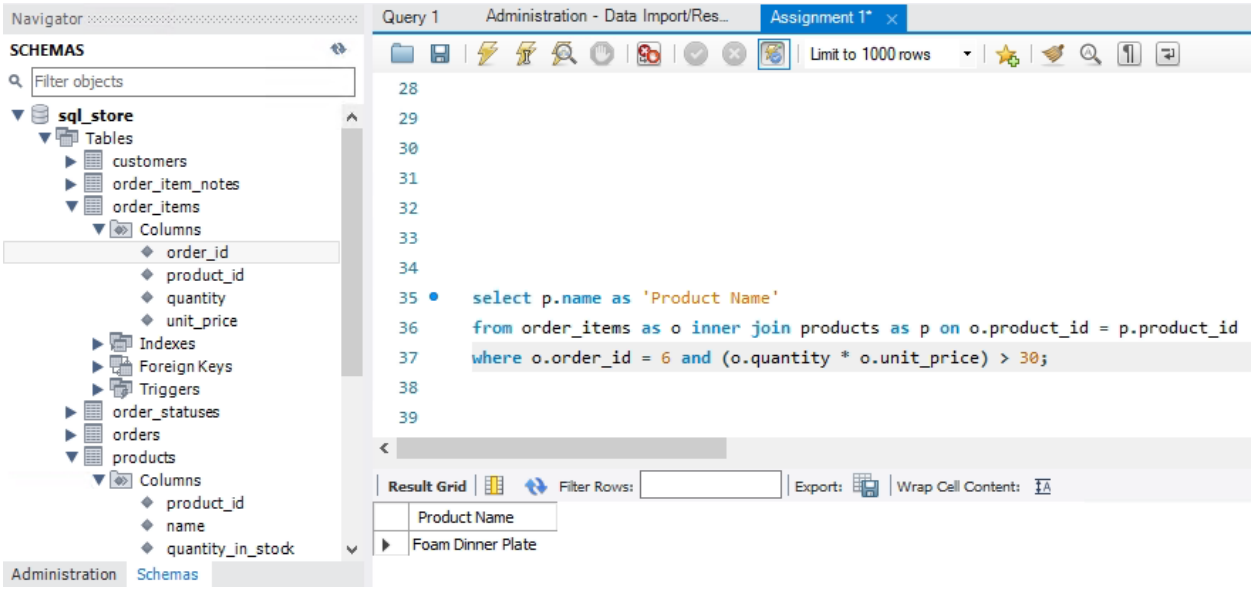
The query below simply creates and shows the product ID field, which is when the order equals six and the total price is more than thirty.



Due to an inner join between the order_items and products tables, the query below creates six columns. The column names are order id, product id, product name, order quantity, product unit price, and total order price.



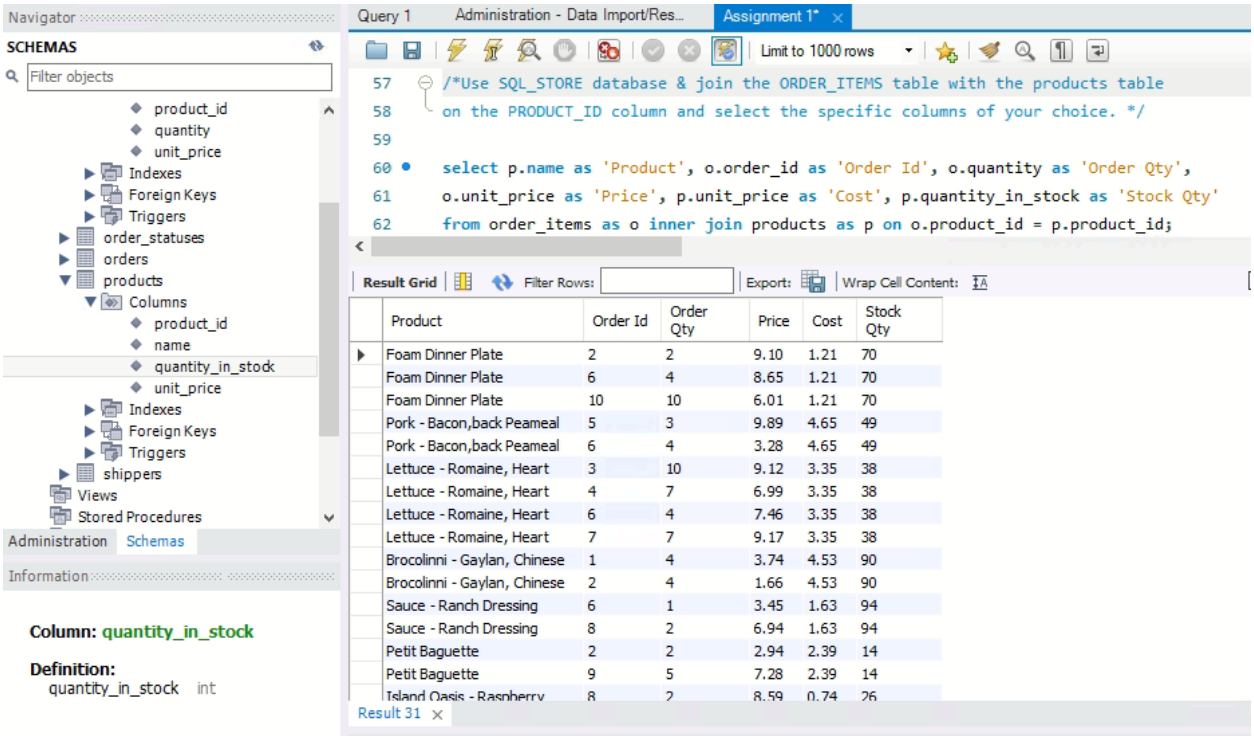
For the criteria mentioned above, the query below just returns the product name.



Inner joins

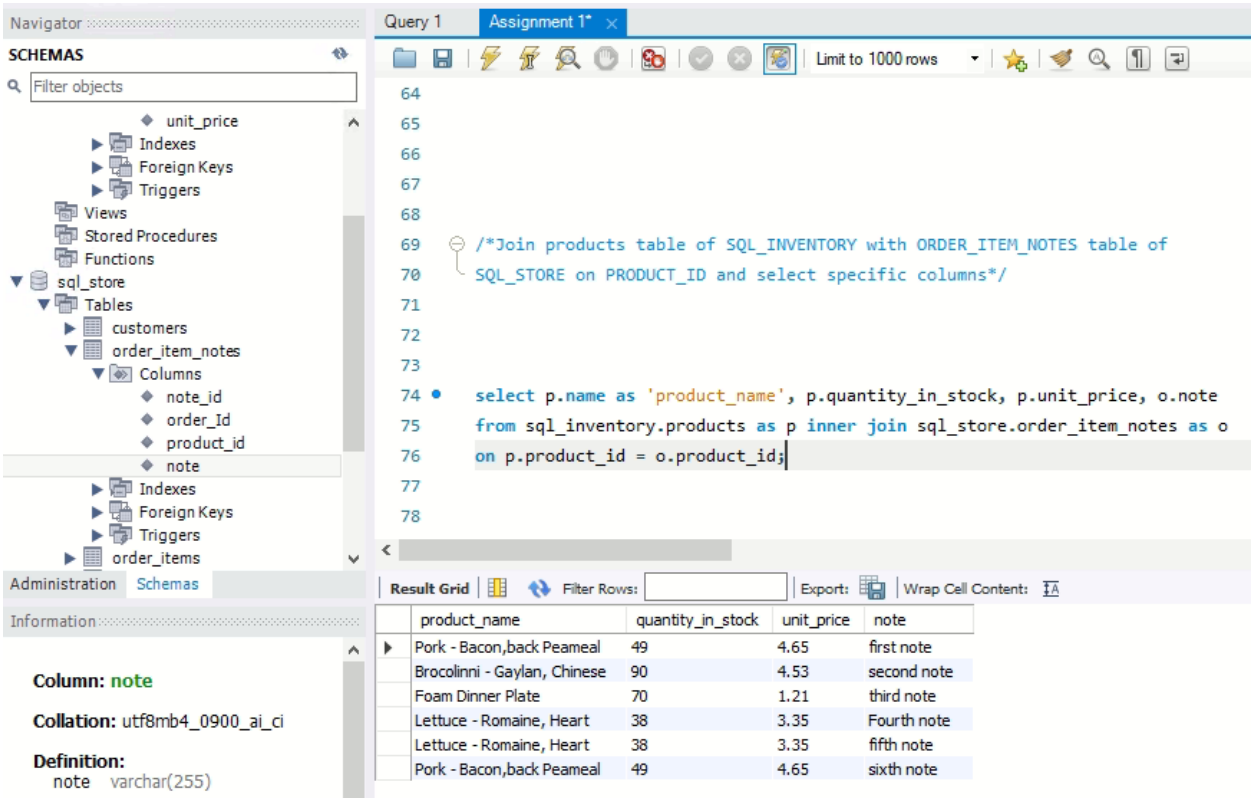
iii. Use SQL_STORE database & join the ORDER_ITEMS table with the products table on the PRODUCT_ID column and select the specific columns of your choice.

The query below inner joins the order_items and products tables using the product_id column. It also makes use of aliases to make the column names easier to comprehend.



iv. Join products table of SQL_INVENTORY with ORDER_ITEM_NOTES table of SQL_STORE on PRODUCT_ID and select specific columns.

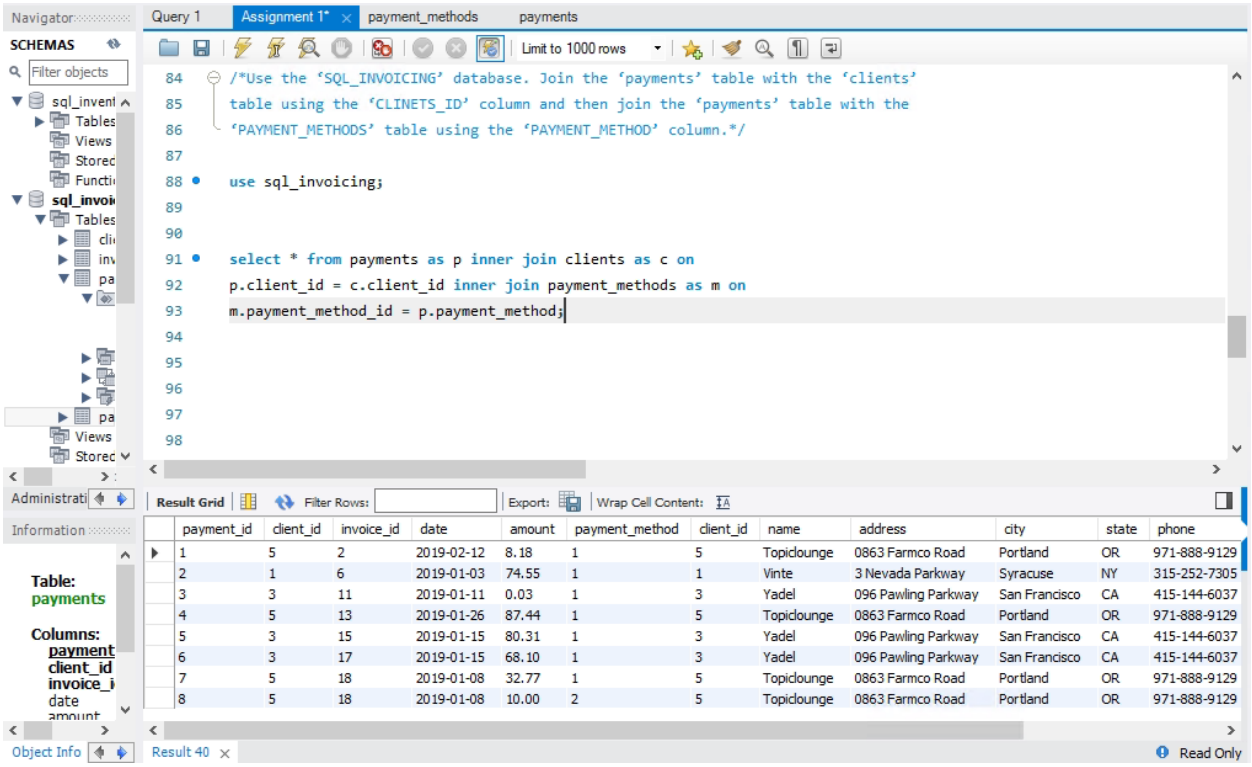
The following query pulls data from the order_item_notes table in the sql_store database and the products table in the sql_inventory database, producing columns for each product that include its name, on-hand quantity, unit price, and notes.



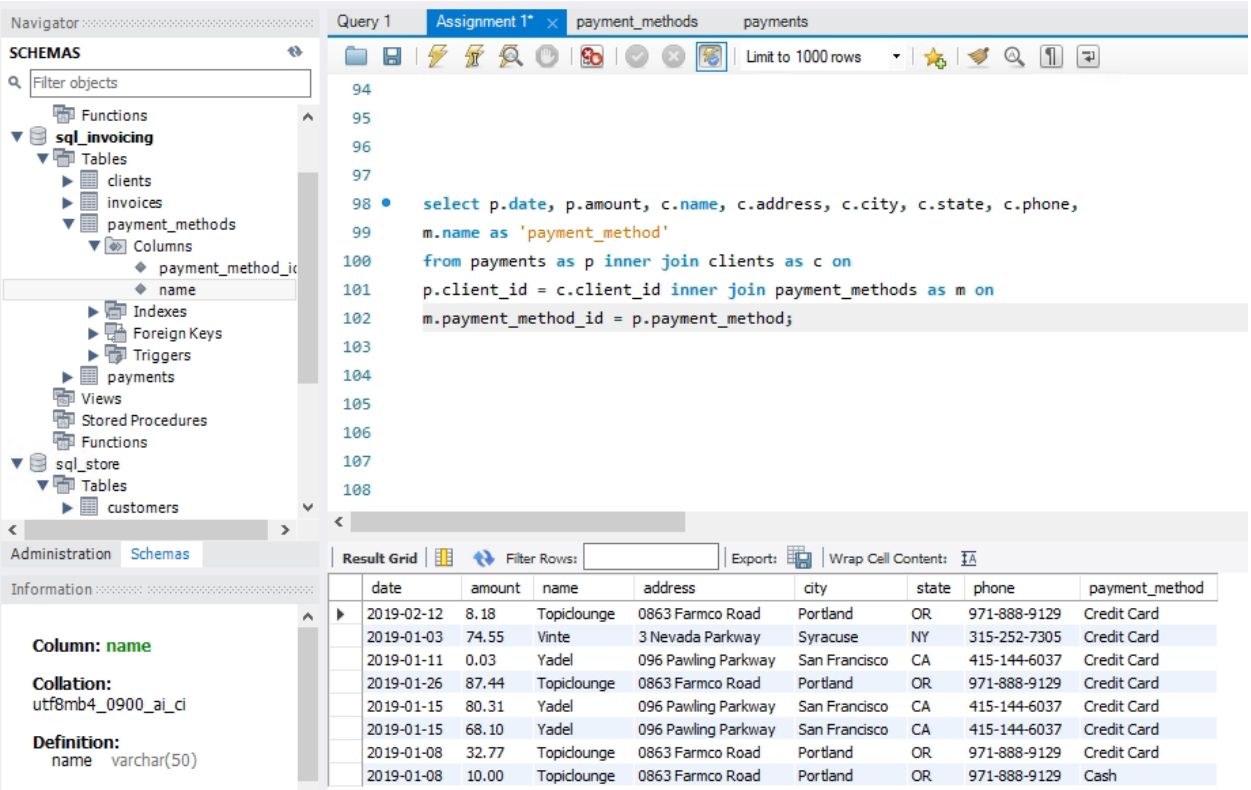
Joining multiple tables

v. Use the ‘SQL_INVOICING’ database. Join the ‘payments’ table with the ‘clients’ table using the ‘CLINETS_ID’ column and then join the ‘payments’ table with the ‘PAYMENT_METHODS’ table using the ‘PAYMENT_METHOD’ column.

The below query shows every column that is part of the payments, clients, and payment_methods tables. The SQL query statement utilizes the "use <database name>" sql query execute the database selection instead of including the database name in the query.



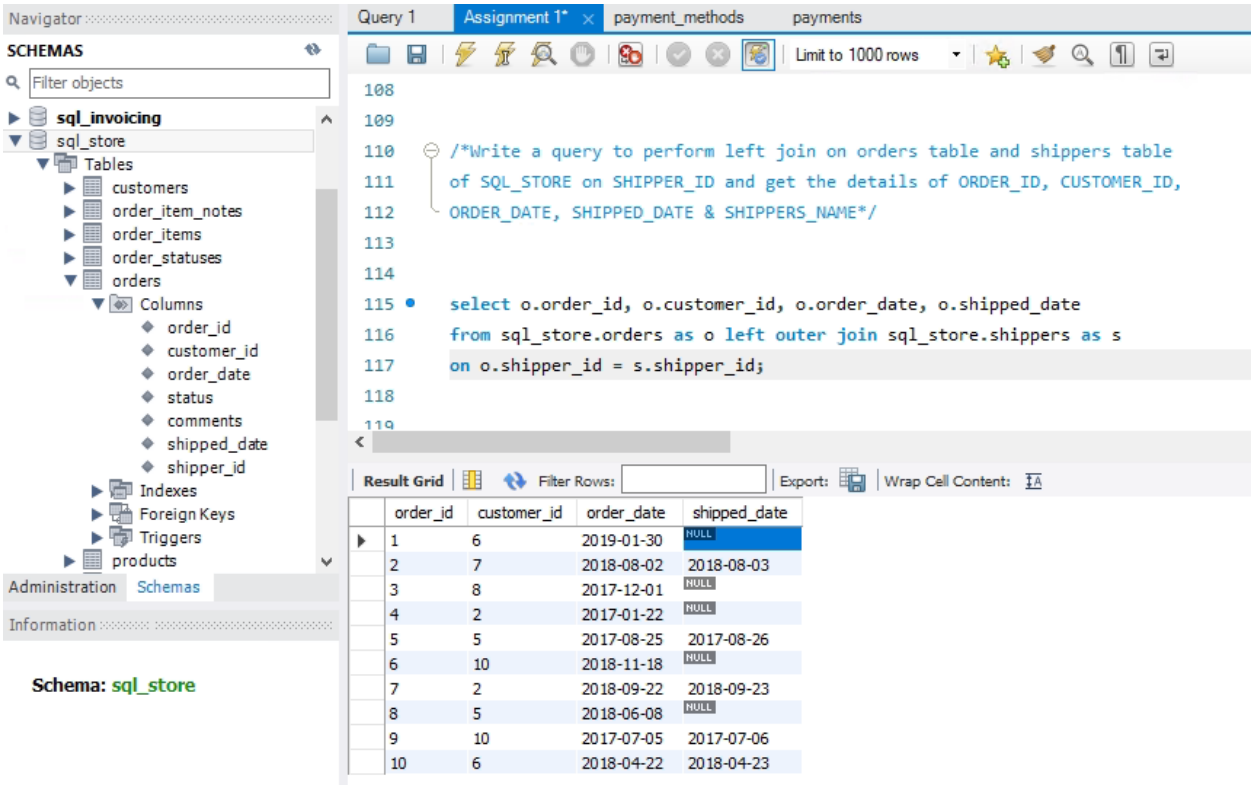
The above same three tables with a few chosen columns are displayed in the query results below.



Outer joins

vi. Write a query to perform left join on orders table and shippers table of SQL_STORE on SHIPPER_ID and get the details of ORDER_ID, CUSTOMER_ID, ORDER_DATE, SHIPPED_DATE & SHIPPERS_NAME.

The orders table and the shippers table are linked using a left outer join in the query below. As a result, all of the rows from the orders table appear in the results table, and the NULL values in the shipped_date column allows to identify the unshipped orders. In this SQL query, the table names have been directly linked to the database name. (db name.table name)



References

Abld Ali Awan. (2023, Sep). *What is Named Entity Recognition (NER)? Methods, Use Cases, and Challenges*. Retrieved from datacamp: https://www.datacamp.com/blog/what-is-named-entity-recognition-ner

Akash Bokade. (2023, Aug). *NLP workbook + Amazon Review Sentiment analysis*. Retrieved from kaggle: https://www.kaggle.com/code/akashbokade/nlp-workbook-amazon-review-sentiment-analysis

Chetna Khanna. (2021, Feb 10). *Text pre-processing: Stop words removal using different libraries*. Retrieved from Medium: https://towardsdatascience.com/text-pre-processing-stop-words-removal-using-different-libraries-f20bac19929a

Kevin Arvai. (2020, Jul 20). *K-Means Clustering in Python: A Practical Guide*. Retrieved from realpython: https://realpython.com/k-means-clustering-python/

Kevin Arvai. (n.d.). *K-Means Clustering in Python: A Practical Guide*. Retrieved from realpython: https://realpython.com/k-means-clustering-python/

Martin Porter. (2006, Jan). *The Porter Stemming Algorithm*. Retrieved from tartarus: https://www.tartarus.org/~martin/PorterStemmer/

Neri. (2023, Nov 03). *How To Implement Intent Classification In NLP [7 ML & DL Models] With Python Example*. Retrieved from spotintelligence: https://spotintelligence.com/2023/11/03/intent-classification-nlp/

Prateek Majumder. (2023, Sep 13). *Named Entity Recognition (NER) in Python with Spacy*. Retrieved from analyticsvidhya: https://www.analyticsvidhya.com/blog/2021/06/nlp-application-named-entity-recognition-ner-in-python-with-spacy/

Sadrach Pierre & Matthew Urwin. (2017, Oct 14). *builtin*. Retrieved from How to Form Clusters in Python: Data Clustering Methods: https://builtin.com/data-science/data-clustering-python

What is named entity recognition? (n.d.). Retrieved from ibm: https://www.ibm.com/topics/named-entity-recognition