1. Suppose you're given the following tables called 'orders' and ‘order\_info'. The table 'orders' shows revenue values for unique orders along with the associated channel ('online' or 'in\_store') while the table 'order\_info' shows the order's ID along with its location. Table: Orders Table: Order\_info Using these tables, write a SQL query to return the top 3 'online' orders and their associated locations based on revenue generated.

SELECT order\_info.location, orders.revenue

FROM orders

JOIN order\_info ON orders.order\_id = order\_info.order\_id

WHERE orders.channel = 'online'

ORDER BY orders.revenue DESC

LIMIT 3;

1. Consider the following table, annual\_sale, shown below: year total\_sale 2015 23000 2016 25000 2017 34000 2018 32000 2019 33000 Use lag() and lead() function to compare annual sale amounts across years.

SELECT year,

total\_sale,

LAG(total\_sale) OVER (ORDER BY year) AS prev\_year\_sale,

LEAD(total\_sale) OVER (ORDER BY year) AS next\_year\_sale

FROM annual\_sale;

1. What is the difference between Stored Procedures and UDFs.

Stored procedures and user-defined functions (UDFs) are both ways of encapsulating SQL code that can be reused and executed multiple times. However, they have some differences in terms of their purpose, usage, and behavior. Here are some of the main differences between them:

Stored procedures can perform data manipulation (DML) and data definition (DDL) operations, such as inserting, updating, deleting, creating, or altering data and objects. [UDFs can only perform data retrieval (SELECT) operations and cannot modify any data or objects1](https://stackoverflow.com/questions/2039936/difference-between-stored-procedures-and-user-defined-functions" \t "https://www.bing.com/_blank)[2](https://docs.snowflake.com/en/developer-guide/stored-procedures-vs-udfs" \t "https://www.bing.com/_blank).

Stored procedures can accept input parameters and return output parameters, but they do not have a return value. [UDFs can accept input parameters and must have a return value, either scalar or table-valued1](https://stackoverflow.com/questions/2039936/difference-between-stored-procedures-and-user-defined-functions" \t "https://www.bing.com/_blank)[2](https://docs.snowflake.com/en/developer-guide/stored-procedures-vs-udfs" \t "https://www.bing.com/_blank)[3](https://askanydifference.com/difference-between-udf-and-stored-procedure-in-sql/" \t "https://www.bing.com/_blank).

Stored procedures can be executed using the EXEC or EXECUTE commands, or by using the CALL command in some databases. [UDFs can be invoked as part of a SQL statement, such as in the SELECT, WHERE, HAVING, or JOIN clauses1](https://stackoverflow.com/questions/2039936/difference-between-stored-procedures-and-user-defined-functions" \t "https://www.bing.com/_blank)[2](https://docs.snowflake.com/en/developer-guide/stored-procedures-vs-udfs" \t "https://www.bing.com/_blank)[4](https://differencebetweenz.com/difference-between-udf-and-stored-procedure-in-sql/" \t "https://www.bing.com/_blank).

Stored procedures can use temporary tables and table variables to store intermediate results. [UDFs can only use table variables and cannot access temporary tables1](https://stackoverflow.com/questions/2039936/difference-between-stored-procedures-and-user-defined-functions" \t "https://www.bing.com/_blank)[5](http://www.differencebetween.net/technology/difference-between-udf-and-stored-procedure-in-sql/" \t "https://www.bing.com/_blank).

Stored procedures can call other stored procedures or UDFs within their code. [UDFs cannot call stored procedures, but they can call other UDFs1](https://stackoverflow.com/questions/2039936/difference-between-stored-procedures-and-user-defined-functions" \t "https://www.bing.com/_blank)[2](https://docs.snowflake.com/en/developer-guide/stored-procedures-vs-udfs" \t "https://www.bing.com/_blank).

Stored procedures can handle exceptions using the TRY-CATCH block or other error handling mechanisms. [UDFs cannot use the TRY-CATCH block or raise errors](https://stackoverflow.com/questions/2039936/difference-between-stored-procedures-and-user-defined-functions)[1](https://stackoverflow.com/questions/2039936/difference-between-stored-procedures-and-user-defined-functions" \t "https://www.bing.com/_blank)[2](https://docs.snowflake.com/en/developer-guide/stored-procedures-vs-udfs" \t "https://www.bing.com/_blank).

Stored procedures can use transactions to ensure data consistency and rollback changes in case of errors. [UDFs cannot use transactions or rollback changes](https://stackoverflow.com/questions/2039936/difference-between-stored-procedures-and-user-defined-functions)[1](https://stackoverflow.com/questions/2039936/difference-between-stored-procedures-and-user-defined-functions" \t "https://www.bing.com/_blank)[2](https://docs.snowflake.com/en/developer-guide/stored-procedures-vs-udfs" \t "https://www.bing.com/_blank).

In general, stored procedures are more suitable for complex tasks that involve multiple steps, data manipulation, error handling, and transaction control. UDFs are more suitable for simple calculations that return a single value or a table of values that can be used in a SQL statement.

or

Stored Procedures and User Defined Functions (UDFs) are both database objects in SQL that help in encapsulating a series of SQL statements into a single entity. They are different in the following ways:

1. **Return Values**: Stored Procedures can return zero, one, or multiple values. This is done through the use of output parameters. On the other hand, User Defined Functions (UDFs) must return a single value, and this is not optional.
2. **Parameters**: Stored Procedures can have both input and output parameters. This allows them to both receive data and send data back to the caller. UDFs, however, can only have input parameters. They can take data in, but they can only send data back through their return value.
3. **SQL Statements**: Stored Procedures allow for a variety of SQL statements to be used, including SELECT, INSERT, UPDATE, and DELETE. UDFs are more limited in this regard, as they only allow for the use of the SELECT statement.
4. **Calling**: UDFs can be called from within Stored Procedures. This allows for code reuse and modular programming. However, the reverse is not true - Stored Procedures cannot be called from UDFs.
5. **Exception Handling**: Exception handling is possible in Stored Procedures through the use of try-catch blocks. This allows for robust error handling and debugging. UDFs do not support try-catch blocks, which can make error handling more difficult.
6. **Transaction Management**: Transaction management, which is crucial for maintaining database integrity, is possible in Stored Procedures. UDFs do not support transaction management, which can limit their use in certain scenarios.
7. **Usage in SQL Statements**: UDFs have the unique ability to be used directly within SQL statements. This includes being used in JOIN operations with other tables if the UDF returns a table. Stored Procedures cannot be used in this way.
8. **Administrative Operations**: Stored Procedures are often used to perform administrative tasks on the database. These tasks can include executing SQL statements that modify the database schema or other objects.
9. **Return Value Requirement**: A function always returns a value explicitly by specifying an expression. This means that every function will always produce some output when called.
10. Take any dataset of your choice and perform outlier analysis using boxplots. Write the interpretation of the graph.

In Jupyter Notebook named Ca2

5. Using any dataset of your choice perform bivariate analysis and interpret each graph – a) Line Charts b) Bar Graph c) Box plots

In Jupyter Notebook named Ca2

1. Briefly discuss about Stacked Bar Graphs with an example

In Jupyter Notebook named Ca2