CODING NINJAS

Formatting Practices

Outline

- Why Formatting?
- Alignment & Indentation
- UpperCase/LowerCase
- Naming objects
- Commenting
- Using Aliases

Formatting makes:

CODING NINJAS

1. Queries will be easier to read.
2. Queries will be easier to correct.
3. Queries will be earlier to compare with other written code.
4. Programmers avoid errors.

Why Formatting

Unformatted Query:

```
SELECT  SUM(orders.Sales) FROM orders LEFT JOIN Manager ON orders.Segment=Manager.Segment GROUP BY orders.Segment;
```

Formatted Query:

```
    SELECT  SUM(orders.Sales)
      FROM orders
 LEFT JOIN Manager
        ON orders.Segment=Manager.Segment
 GROUP BY orders.Segment;
```

Common and popular rules to format SQL queries:

Alignment & Indentation

01

```
  SELECT  SUM(orders.Sales)
     FROM orders
LEFT JOIN Manager
       ON orders.Segment=Manager.Segment
 GROUP BY orders.Segment;
```

- Writing keywords on a new line to the left and the rest of the code to the right.
- Use a new line for each separate query.
- Use spaces to surround the equals operator.
- Use spaces before or after apostrophes.

Uppercase/ Lowercase

02

```
INSERT INTO Manager VALUES
('Consumer','Gaganjit Singh'),
('Corporate','Aman Jain'),
('Home Office','Kush Arora');
```

- Use uppercase for the SQL keywords (like INSERT, INTO, VALUES).
- Use uppercase for the SQL functions (like SUM(), AVG()).
- Use lowercase for your tables and columns (Manager here).

Naming objects

03

```
CREATE TABLE orders(
    Order_ID VARCHAR(100),
    Order_Date VARCHAR(100),
    Ship_Date VARCHAR(100),
    Customer_ID VARCHAR(200),
    Product_Name VARCHAR(500),
    Sales DOUBLE,
    Profit DOUBLE
);
```

- Use uppercase for the SQL keywords.
- Avoid the name of a table/column in the plural.
- If the name of the table or column must consist of more than one word, use an underscore to connect them.
- Avoid giving the same name to both a table and a column.
- Avoid special characters in the name like $, &, *.
- Avoid abbreviations.
- Don't start the name with an underscore.

Commenting

04

```
SELECT PersonId,
       FirstName,
       LastName,
       /* cname column is the name of the city: */
       cName,
  FROM Person
 WHERE cName = 'New York';
```

- Comments might be useful in situations where code needs to be explained or re-examined.
- Better to use multiple-line comments which are indicated by /* opening and */ closing characters.
- Comment at the start of a new line,
- The comment should be written above the relevant SQL code line, using the same indentation.
- Avoid commenting too much.

Using Aliases

05

```
SELECT PersonId,
       FirstName,
       LastName,
       /* cname column is the name of the city: */
       cName AS cityName
  FROM Person
 WHERE cName = 'New York';
```

- Aliases are a convenient way to rename tables or columns which doesn't make sense.
- Include the AS keyword for creating aliases
- Give an alias to tables and columns when their names aren't meaningful.

CODING NINJAS

To read more about formatting practices, refer to the following links:

https://www.sqlstyle.guide/

https://towardsdatascience.com/10-best-practices-to-write-readable-and-maintainable-sql-code-427f6bb98208

https://learnsql.com/blog/24-rules-sql-code-formatting-standard/