**SITUATION**

You are a business analyst working at Deloitte. You already know SQL, and you have both a more junior analyst and a more senior analyst working on your team. In this exercise, you will help the junior analyst make their SQL commands work correctly and help the more senior analyst finish off some of their harder queries.

Your client, Adventure Works Cycles, has some urgent questions - questions that need answering. They are looking to drive sales of products but need some help navigating their data.

The friendly DBA (database administrator) team at Adventure Works supplied you with the [database schema](https://drive.google.com/file/d/10OMrWes-ygcP5VUUox1CrKVDYtnHkaI6/view?usp=sharing) and [documentation of data types](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008/ms124438(v=sql.100)) to help you navigate the database if needed.

**EXERCISES**

**QUESTION 1:**

The junior analyst wants to get the item numbers and descriptions of products with a list price of at least $100 and safety stock level of at least 250. She writes:

SELECT p.productid , p.name

FROM production.product p

WHERE p.listprice <= 100

WHERE p.safetystocklevel <= 250;

Why isn’t this query working? [,] AND Where after Where

**SELECT** p.productid , p.**name**

**FROM** production.product p

**WHERE** p.listprice **<=** 100 **OR** p.safetystocklevel **<=** 250;

**QUESTION 2:**

She now wants to get a list of all combinations of “class” and “style” in the product data. She knows that she can select these two columns using:

SELECT p.class, p.style FROM production.product p;

What else does she need to add to accomplish her goal?

**SELECT** p.**class**, p.**style**

**FROM** production.product p

**WHERE** p.**style** !**=** 'NULL'

**AND** p.**class** !**=** 'NULL'

**QUESTION 3:**

Her query to get all the products where list price is greater than $3,000 seems right, but it’s generating an error:

SELECT p.productid, p.name

FROM production.product p

WHERE p.listprice > '$3000';

How could you fix it? ( $ )

**SELECT** p.productid, p.**name**

**FROM** production.product p

**WHERE** p.listprice **>** '3000';

**QUESTION 4:**

Deloitte has a few products identified that they’d like to promote in their next marketing campaign:

‘Spokes’

‘Stem’

‘Sport-100 Helmet, Red’

Your colleague’s SQL has improved based on your feedback, so she correctly writes:

SELECT p.productid, p.name, p.productnumber

FROM production.product p

WHERE p.name = 'Spokes' OR

p.name = 'Stem' OR

p.name = 'Sport-100 Helmet, Red';

How could you make this query shorter and more consistent with SQL best practices?

**SELECT** p.productid,

p.**NAME**,

p.productnumber

**FROM** production.product p

**WHERE** p.**NAME** **=** 'Spokes'

**OR** p.**NAME** **=** 'Sport-100 Helmet, Red'

**OR** p.**NAME** **=** 'Stem'

**QUESTION 5:**

Now you are given an area to analyze independently: your Deloitte contact wants a list of name of all the bicycle frames they have in their product line.

First, generate a list of names of all the bicycle frames.

**SELECT** id, **name**

**FROM** productline.production

**QUESTION 6:**

Now, Deloitte wants the average list price of all mountain bike frames, grouped by size and weight, and sorted ascending by size.

**SELECT** id, **name** , **size** , weight ,

**CAST**(**AVG**(price) **AS** MONY)

**FROM** productline.production

**GROUP** **BY** **size** , weight

**ORDER** **BY** **size** **ASC**;

**QUESTION 7:**

Using the **sales.salesorderdetail** table, make a list of the top 10 sales orders - the sales orders with the highest dollar amount of goods sold. Report the sales order id and the total dollar value of each of these top 10 sales orders.

top 10 sales orders:

**SELECT** salesorderid,orderqty,

**CAST**(unitprice **AS** MONEY)

**FROM** sales.salesorderdetail

**ORDER** **BY** orderqty **DESC** **LIMIT** 10

highest dollar amount of goods sold:

**SELECT** salesorderid,orderqty,

**CAST**(unitprice **AS** MONEY)

**FROM** sales.salesorderdetail

**ORDER** **BY** unitprice **DESC** **LIMIT** 10

**QUESTION 8:**

Using the **sales.salesorderheader** table and the results from the previous question, what is the freight cost associated with the highest dollar value sales order? Write a query that can achieve this result dynamically (don’t view the results from the first query and manually code them into the second query)!

**STRETCH QUESTION:**

You are proud of this work and you want to share it with the rest of the team. However, Question 8 results in a complicated query that would be tough for someone who wasn’t an advanced SQL user to pick up and understand.

Take the query above and add some comments explaining how it works, using the correct syntax for commenting in SQL. Aim to write a couple of full sentences at the beginning of the query explaining how it works, and add at least 3 shorter, 1 line comments throughout the code.