**SQL Queries Quiz**

**Question 1**

Consider the Bicycle table

|  |  |  |  |
| --- | --- | --- | --- |
| Bicycle | | | |
| **brand** | **type** | **colour** | **shop** |
| Giant | Mountain | White | Matt’s Cycles |
| R&M | Trekking | Yellow | Kew Cycling |
| Bianchi | Road | Black | Abbotsford Bicycles |
| Gudereit | Trekking | Blue | 99Bikes |

How many rows are returned by this statement?

*SELECT \**

*FROM Bicycle*

*WHERE type like (‘%o%’);*

* 2

**Question 2**

Based on the Bicycle table

|  |  |  |  |
| --- | --- | --- | --- |
| Bicycle | | | |
| **brand** | **type** | **colour** | **shop** |
| Giant | Mountain | White | Matt’s Cycles |
| R&M | Trekking | Yellow | Kew Cycling |
| Bianchi | Road | Black | Abbotsford Bicycles |
| Gudereit | Trekking | Blue | 99Bikes |

Which of the following statements would display the following result?

|  |  |
| --- | --- |
| **BicycleBrand** | **BicycleType** |
| Giant | Mountain |
| R&M | Trekking |
| Bianchi | Road |
| Gudereit | Trekking |

* *SELECT brand as BicycleBrand, type as BicycleType*

*FROM Bicycle;*

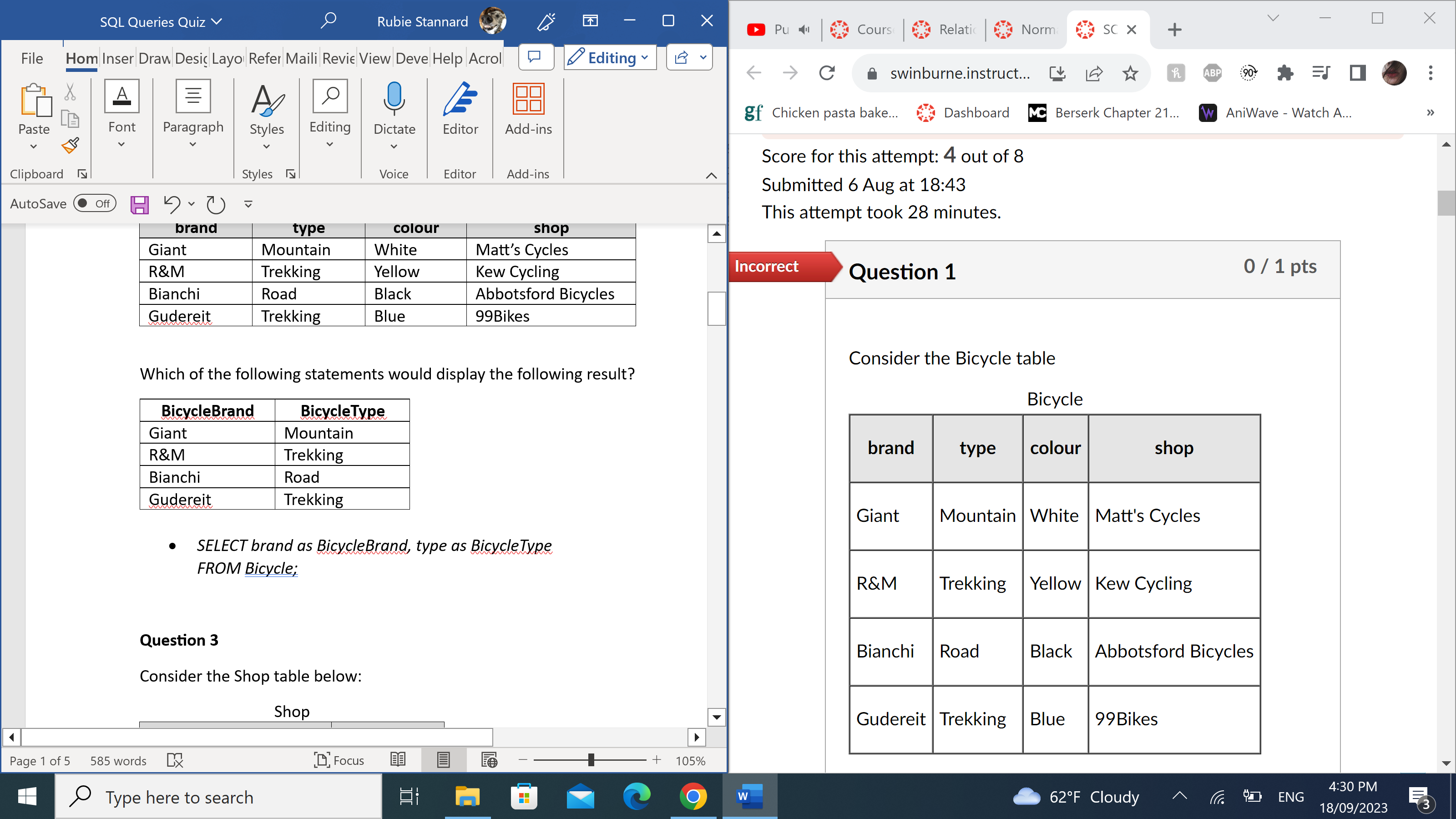
**Question 3**

Consider the Shop table below:

|  |  |
| --- | --- |
| Shop | |
| **ShopName** | **Phone** |
| Matt’s Cycle | 9111111 |
| Kew Cycling | 9555555 |
| Abbotsford Bicycles | 9876344 |
| 99Bicycles |  |

Which of the following is the correct output of this statement

*SELECT COUNT(Phone)*

*FROM Shop;*

|  |
| --- |
| **COUNT(PHONE)** |
| 3 |

**Question 4**

Consider the following ORDER table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ORDER | | | | |
| **ORDER\_ITEM\_ID** | **ORDER\_ID** | **PRODUCT\_ID** | **UNIT\_PRICE** | **QUANTITY** |
| 1 | 1 | 1 | 50 | 10 |
| 2 | 1 | 2 | 80 | 8 |
| 3 | 1 | 3 | 150 | 5 |
| 4 | 2 | 1 |  | 3 |
| 5 | 3 | 10 | 50 | 2 |
| 6 | 4 | 6 | 120 | 2 |
| 7 | 4 | 7 | 30 | 6 |
| 8 | 4 | 8 | 40 | 10 |
| 9 | 4 | 9 | 110 | 2 |

What is the result obtained by the following statement

*SELECT \* FROM ORDER*

A screenshot of a computer

Description automatically generated*WHERE UNIT\_PRICE >50 and QUANTITY = 2 or QUANTITY = 10;*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ORDER\_ITEM\_ID** | **ORDER\_ID** | **PRODUCT\_ID** | **UNIT\_PRICE** | **QUANTITY** |
| 1 | 1 | 1 | 50 | 10 |
| 6 | 4 | 6 | 120 | 2 |
| 9 | 4 | 9 | 110 | 2 |
| 8 | 4 | 8 | 40 | 10 |

**Question 5**

Consider the following ORDER table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ORDER | | | | |
| **ORDER\_ITEM\_ID** | **ORDER\_ID** | **PRODUCT\_ID** | **UNIT\_PRICE** | **QUANTITY** |
| 1 | 1 | 1 | 50 | 10 |
| 2 | 1 | 2 | 80 | 8 |
| 3 | 1 | 3 | 150 | 5 |
| 4 | 2 | 1 |  | 3 |
| 5 | 3 | 10 | 50 | 2 |
| 6 | 4 | 6 | 120 | 2 |
| 7 | 4 | 7 | 30 | 6 |
| 8 | 4 | 8 |  | 2 |
| 9 | 4 | 9 | 110 | 2 |

Which of the following statements would return this result:

|  |  |
| --- | --- |
| **ORDER\_ID** | **SUM(QUANTITY)** |
| 1 | 23 |
| 4 | 12 |

* *SELECT ORDER\_ID, SUM(QUANTITY)*

*FROM ORDER*

*GROUP BY ORDER\_ID*

*HAVING SUM(QUANTITY) >= AVG(QUANTITY);*

**Question 6**

Consider the following ORDER table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ORDER | | | | |
| **ORDER\_ITEM\_ID** | **ORDER\_ID** | **PRODUCT\_ID** | **UNIT\_PRICE** | **QUANTITY** |
| 1 | 1 | 1 | 50 | 10 |
| 2 | 1 | 2 | 80 | 8 |
| 3 | 1 | 3 | 150 | 5 |
| 4 | 2 | 1 |  | 3 |
| 5 | 3 | 10 | 50 | 2 |
| 6 | 4 | 6 | 120 | 2 |
| 7 | 4 | 7 | 30 | 6 |
| 8 | 4 | 8 |  | 2 |
| 9 | 4 | 9 | 110 | 2 |

Which of the following statements would return this result:

|  |  |
| --- | --- |
| **PRODUCT\_ID** | **TOTAL\_VALUE** |
| 1 | 500 |
| 2 | 640 |
| 3 | 750 |
| 6 | 240 |
| 7 | 180 |
| 8 |  |
| 9 | 220 |
| 10 | 110 |

* *SELECT PRODUCT\_ID, SUM(QUANTITY\*UNIT\_PRICE) AS TOTAL\_VALUE*

*FROM ORDER*

*GROUP BY PRODUCT\_ID*

*ORDER BY PRODUCT\_ID;*

**Question 7**

Consider the following ORDER table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ORDER | | | | |
| **ORDER\_ITEM\_ID** | **ORDER\_ID** | **PRODUCT\_ID** | **UNIT\_PRICE** | **QUANTITY** |
| 1 | 1 | 1 | 50 | 10 |
| 2 | 1 | 2 | 80 | 8 |
| 3 | 1 | 3 | 150 | 5 |
| 4 | 2 | 1 |  | 3 |
| 5 | 3 | 10 | 50 | 2 |
| 6 | 4 | 6 | 120 | 2 |
| 7 | 4 | 7 | 30 | 6 |
| 8 | 4 | 8 |  | 2 |
| 9 | 4 | 9 | 110 | 2 |

Are there any issues with this result:

|  |  |
| --- | --- |
| **PRODUCT\_ID** | **TOTAL\_VALUE** |
| 1 | 500 |
| 2 | 640 |
| 3 | 750 |
| 6 | 240 |
| 7 | 180 |
| 8 |  |
| 9 | 220 |
| 10 | 110 |

* Yes, the quantity of 500 for PRODUCT\_ID 1 hides the fact that PRODUCT\_ID value has a null value for UNIT\_PRICE in one of the entries.

**Question 8**

Consider the following ORDER table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ORDER | | | | |
| **ORDER\_ITEM\_ID** | **ORDER\_ID** | **PRODUCT\_ID** | **UNIT\_PRICE** | **QUANTITY** |
| 1 | 1 | 1 | 50 | 10 |
| 2 | 1 | 2 | 80 | 8 |
| 3 | 1 | 3 | 150 | 5 |
| 4 | 2 | 1 |  | 3 |
| 5 | 3 | 10 | 50 | 2 |
| 6 | 4 | 6 | 120 | 2 |
| 7 | 4 | 7 | 30 | 6 |
| 8 | 4 | 8 |  | 2 |
| 9 | 4 | 9 | 110 | 2 |

The following query:

*SELECT ORDER\_ID, SUM(QUANTITY) FROM ORDER;*

Gives the following result in MySQL:

|  |  |
| --- | --- |
| **ORDER\_ID** | **SUM(QUANTITY)** |
| 1 | 40 |

Which of the following comments is accurate?

* This behaviour is undesirable because the ORDER\_ID is an arbitrary choice.