## **ST1501 Data Engineering**

## **CA1 Assignment Test**

|  |  |
| --- | --- |
| **Name** | Silviana |
| **Student ID** | P1939213 |
| **Module Class** | DIT/FT/2A/14 |

## Important Notes:

If your database design cannot support the said query scenario, please explain the reason. Marks can be awarded if you explain the reason clearly and provide an improvement to your design.

## SECTION A

**You should prepare the following diagram/screen shots of your database before the test start**

1. **Insert your Database Diagram here**

|  |
| --- |
| **Database diagram** |

1. **Take screen shorts all the records available. (The following is example of one table)**

**product Table**

**A screenshot of a computer

Description automatically generated**

**product\_brand Table**

**A screenshot of text

Description automatically generated**

**product\_category Table**

**A screenshot of text

Description automatically generated**

**product\_pricing Table**

**A screenshot of a cell phone

Description automatically generated**

**product\_promotion Table**

**A screenshot of a computer

Description automatically generated**

**promotion Table**

**A picture containing white, table, plate

Description automatically generated**

## SECTION B

Give your answer in the order of the questions show in your test items.

**Question 1**

|  |
| --- |
| Copy your question here (optional)  The business owner informs that a product can have more than one classification. For example, ‘Farmhouse UHT Milk’.  The ‘Farmhouse UHT Milk’ can be classified in the following two ways.  Drinks – UHT Milk -> UHT Milk -> Farm House UHT Milk  Dairy, Chilled & Eggs -> UHT Milk -> Farm House UHT Mile  If your database supports the above requirement, write an SQL statement to list all products that have more than one classification.  If your database could not support the above requirement, explain the reason and how would you improve the design to support the above requirement. |
| Provide the answer or SQL statement  My database does not support the above requirement because my **product** and **product\_category** table has a one to many relationship. Therefore, my current system only stores the level 3 category on the **product** table and uses that as a foreign key that references the **product\_category**.  In order to improve my design, I should turn the **product** and **product\_category** relationship into many to many relationship. |
| Paste the screen shot of query result if applicable |

**Question 2**

|  |
| --- |
| Copy your question here (optional)  Write a SQL statement to list all products classified in the ‘Detergent’ category. You can assume ‘Detergent’ is a level two classification.  The query result should show the product description, the product classification (all the three levels) and brand name. The display should be sorted by ascending order of brand name. |
| Provide the answer or SQL statement  SELECT  p.product\_desc,  pc3.category\_name AS category\_name,  pc2.category\_name AS sub\_category\_name,  pc1.category\_name AS sub\_sub\_category\_name,  pb.brand\_name  FROM product as p  JOIN product\_brand AS pb ON pb.brand\_id = p.brand\_id  JOIN product\_category AS pc1 ON pc1.category\_id = p.category\_id  LEFT OUTER JOIN product\_category AS pc2 ON pc1.parent\_id = pc2.category\_id  LEFT OUTER JOIN product\_category AS pc3 ON pc2.parent\_id = pc3.category\_id  WHERE pc2.category\_name = 'Detergents'  ; |

|  |
| --- |
| Paste the screen shot of query result if applicable |

**Question 3**

|  |
| --- |
| Copy your question here (optional)  Write a SQL statement to list the product description and promotion price (discounted unit price) for all items on promotion on 15 June 2020. The promotion includes direct discount and quantity-based discount.  The query result should show the product description, the promotion type, the discounted price, the start date, and end date of the promotion. |
| Provide the answer or SQL statement  GO  CREATE FUNCTION dbo.discounted\_price  (  @base\_price DECIMAL(10,2),  @discount\_value DECIMAL(10,2),  @discount\_unit VARCHAR(10),  @min\_order\_qty INT,  @quantity INT = 1  )  RETURNS DECIMAL(10, 2)  AS  BEGIN  DECLARE @discounted\_price DECIMAL(10,2);  DECLARE @multiplier INT;    IF (ROUND(@quantity/@min\_order\_qty, 0) = @quantity/@min\_order\_qty) SET @multiplier = @quantity/@min\_order\_qty  ELSE SET @multiplier = FLOOR(@quantity/@min\_order\_qty)  IF (@discount\_unit = '$') SET @discounted\_price = @base\_price \* @quantity - @discount\_value \* @multiplier  IF (@discount\_unit = '%') SET @discounted\_price = @base\_price \* @quantity - @discount\_value \* @multiplier/100 \* @base\_price  RETURN @discounted\_price  END  GO  DECLARE @quantity INT;  SET @quantity = 3;  SELECT  p.product\_desc,  promo.promotion\_name,  @quantity AS quantity,  promo.min\_order\_qty,  pp.base\_price,  dbo.discounted\_price(pp.base\_price, promo.discount\_value, promo.discount\_unit, promo.min\_order\_qty, @quantity) AS current\_price,  promo.valid\_from,  promo.valid\_to  FROM  product AS p  JOIN product\_pricing AS pp ON pp.product\_id = p.product\_id  RIGHT JOIN product\_promotion AS ppro ON ppro.product\_id = p.product\_id  RIGHT JOIN promotion AS promo ON promo.promotion\_id = ppro.promotion\_id  WHERE pp.is\_active = 1 AND p.is\_in\_stock = 1 AND is\_available = 1 AND '2020-06-15 00:00:00.000' >= promo.valid\_from AND '2020-06-15 00:00:00.000' <= promo.valid\_to  ORDER BY promo.promotion\_id |
| Paste the screen shot of query result if applicable |

**Question 4**

|  |
| --- |
| Copy your question here (optional)  Write a SQL statement to list the product name, product description and regular price of all products that are available on the online store. The regular price should be the effective regular price on 15 June 2020.  The query result should show the product name (or the primary key of the product table), product description and the regular price. |
| Provide the answer or SQL statement  GO  CREATE FUNCTION dbo.current\_price  (  @base\_price DECIMAL(10,2),  @discount\_value DECIMAL(10,2),  @discount\_unit VARCHAR(10),  @min\_order\_qty INT,  @quantity INT = 1  )  RETURNS DECIMAL(10, 2)  AS  BEGIN  DECLARE @discounted\_price DECIMAL(10,2);  DECLARE @multiplier INT;    IF (ROUND(@quantity/@min\_order\_qty, 0) = @quantity/@min\_order\_qty) SET @multiplier = @quantity/@min\_order\_qty  ELSE SET @multiplier = FLOOR(@quantity/@min\_order\_qty)  IF (@discount\_unit = '$') SET @discounted\_price = @base\_price \* @quantity - @discount\_value \* @multiplier  IF (@discount\_unit = '%') SET @discounted\_price = @base\_price \* @quantity - @discount\_value \* @multiplier/100 \* @base\_price  ELSE SET @discounted\_price = @base\_price  RETURN @discounted\_price  END  GO  DECLARE @quantity INT;  SET @quantity = 1;  SELECT  p.product\_name,  p.product\_desc,  pp.base\_price,  dbo.current\_price(pp.base\_price, promo.discount\_value, promo.discount\_unit, promo.min\_order\_qty, @quantity) AS current\_price  FROM  product AS p  FULL OUTER JOIN product\_pricing AS pp ON pp.product\_id = p.product\_id  FULL OUTER JOIN product\_promotion AS ppro ON ppro.product\_id = p.product\_id  FULL OUTER JOIN promotion AS promo ON promo.promotion\_id = ppro.promotion\_id  WHERE pp.is\_active = 1 AND p.is\_in\_stock = 1 AND is\_available = 1  ORDER BY promo.promotion\_id DESC  I did not have the time to check the price on 15 June 2020 and I am not sure if the effective regular price refers to the base price or the discounted price. Hence, I am showing the discounted\_price. |
| Paste the screen shot of query result if applicable |