



**MYSQL END OF MODULE CHALLENGE ANSWER SHEET**

| **Assessor** | Luvuyo |
| --- | --- |
| **Moderator** | Joel / Ryan / Oslin |
| **Due Date** | 3 Feb 2023 |
| **PASS MARK** | 50% |
| **Student Name** |  |

**INSTRUCTIONS**

1. Please adhere to the due dates.

2. Coping/Plagiarism is not accepted.

3. Make sure you write the code in the space provided for each question

4. Upload the Answer Sheet together with the database dump (backup file)

**1. CREATING DATABASE AND MANIPULATING DATA**

A local shop owner has asked you to design a database which they will use to store information about their suppliers and the products they supply. They sell fresh produce to the public. They provided you with a list of their suppliers and the products they supply to help you design the database. The database consists of TWO tables (*Suppliers* and *Products*).

1.1 Database Creation. Paste your code in the space provided. **[3]**

| CREATE DATABASE FruitMarket;  USE FruitMarket; |
| --- |

1.2 Table Suppliers. Paste your code in the space provided. **[8]**

| CREATE TABLE Suppliers(SupplierID VARCHAR(10) NOT NULL PRIMARY KEY,  CompanyName VARCHAR(30) NOT NULL,  ContactPerson VARCHAR(30) NOT NULL,  ContactNo VARCHAR(13) NOT NULL DEFAULT '000-000-0000',  ProductCategory VARCHAR(55) NOT NULL  ); |
| --- |

3 marks for correct SQL statements

5 Marks for correct field-names and keys as stated above

1.3 Products table using details provided below.

**Solution**

Write your code on the space provided in the space provided below. [4]

| CREATE TABLE Products(ProductID INT NOT NULL PRIMARY KEY,  ProductName VARCHAR(30),  Price DECIMAL(10, 2),  Weight VARCHAR(10),  Stock INT,  SupplierID VARCHAR(10), FOREIGN KEY(SupplierID) REFERENCES Suppliers(SupplierID)  ); |
| --- |

1 Mark for correct SQL and table

2 Marks for correct field-names and sizes, deduct 1 mark for wrong fields sizes and   
 1 mark for data types

1 mark for correct linking of the foreign key

1.4 Relationship [1]

| A Many to Many Relationship. |
| --- |

1.5 Foreign key [1]

| SupplierID |
| --- |

1.6 Adding records to the database. [1]

Write your answer in the space provided.

| INSERT INTO Suppliers VALUES('SUPP0001', 'Fruit City', 'Themba', '0115062089', 'Fruit'),  ('SUPP0002', 'Vegan Veggie Xpress', 'Chinyere', '0137228936', 'Vegetables'),  ('SUPP0003', 'The Nut House', 'Sam', '0216965133', 'Nuts'),  ('SUPP0004', 'The Lazy Cow', 'Angelo', '0216964157', 'Dairy')  ; |
| --- |

1 Marks for each correct record

Deduct 1 Mark if a single insert is not used

1.7 Add records into the Products table. Write your answer in the space provided. [4]

| INSERT INTO Products(ProductID, ProductName, Price, Weight, Stock, SupplierID)  VALUES  -- Fruit City  (1001, 'Lady Finger Bananas', 17.95, '750 g', 45, 'SUPP0001'),  (1002, 'Pink Lady Apples', 18.95, '1,5 kg', 15, 'SUPP0001'),  (1003, 'Red Anjou Pears', 22.99, '1 kg', 24, 'SUPP0001'),  (1004, 'Cavendish Bananas', 12.65, '900 kg', 18, 'SUPP0001'),  -- Vegan Veggie Xpress  (2001, 'Tenderstem Broccoli', 35.90, '400 g', 8, 'SUPP0002'),  (2002, 'Portabellini Mushrooms', 18.99, '250 g', 16, 'SUPP0002'),  -- The Nut House  (3001, 'Raw Almonds', 99, '1 kg', 6, 'SUPP0003'),  (3002, 'Macaroon Butter', 32.95, '410 g', 9, 'SUPP0003'),  (3003, 'Organic Coconut Oil', 89.95, '500 ml', 15, 'SUPP0003'),  -- The Lazy Cow  (4001, 'Ayshire Milk', 33.95, '3 l', 23, 'SUPP0004'),  (4002, 'Simonzola Blue Cheese', 27.65, '270 g', 4, 'SUPP0004')  ;  SELECT ProductID, ProductName, CONCAT('R', Price) AS Price, Weight, Stock FROM Products; |
| --- |

4 Marks for adding all the records correctly

Deduct 1 mark if a record is missing. Maximum of 2 marks

1.8 SQL query to extract the following ProductId, ProductName, Price, Weight, Stock, ProductCategory [6]

| SELECT p.ProductID, p.ProductName, p.Price, p.Weight, p.Stock,  s.ProductCategory  FROM Products p  INNER JOIN Suppliers s  USING(SupplierID)  WHERE p.Stock < 20  ORDER BY p.Price DESC; |
| --- |

2 marks for correct SQL statement

Maximum of 2 marks for correct fields.

1 mark for the order

1 mark for permanently saving into the database

1.9 View Q1.9 [6]

Your Output in the space below:

| CREATE VIEW Q9 AS  SELECT p.ProductID, p.ProductName, p.Price, p.Weight, p.Stock,  s.ProductCategory, round((p.Price \* p.Stock \* 1.15), 4) AS TotalPrice  FROM Products p  INNER JOIN Suppliers s  USING(SupplierID)  WHERE p.Stock < 20  ORDER BY p.Price DESC;  SELECT \* FROM Q9; |
| --- |

1 Mark for the view

1 Mark for the correct fields

3 marks for calculation of total price and ceiling

1 mark for correct records

Deduct a mark for incorrect tax calculation, total calculation

1.10 User called ‘yourname\_initialofyoursurname’ with INSERT privileges on to the current database and Supplier Table. Write your answer in the space provided [3]

| CREATE USER 'Deno\_r'@'localhost' IDENTIFIED BY '12345';  GRANT INSERT ON FruitMarket.Suppliers TO 'Deno\_r'@'localhost';  FLUSH PRIVILEGES; |
| --- |

1 mark for the correct user created

1 mark for the privileges

1 mark for correct database and table

1.11 Number of tables accessible. [1]

| mysql -u Deno\_r -p  USE FruitMarket;  SHOW TABLES;  ===Answer===  Only one table is accessible: Suppliers |
| --- |

1.12 Add a record with the following details to the Suppliers table

('SUPP006', 'Fruit&Veg', 'Abdu', '0216965111', 'Nuts'); [1]

| INSERT INTO Suppliers VALUES('SUPP006', 'Fruit&Veg', 'Abdu', '0216965111', 'Nuts'); |
| --- |

1 mark for correct SQL and record details

No mark if the output is wrong

1.13 Display all records in the Suppliers table. [2]

| The Records can’t be displayed because the ‘SELECT’ privilege wasn’t given in the ‘GRANT’ statement. |
| --- |

1 Mark for correct answer

1 Mark for reason

1.14 View called Q1.14 which will display ONLY the new user you created and the root. [2]

| CREATE VIEW Q14 AS  SELECT User, Host  FROM mysql.user  WHERE User IN ('Deno\_R', 'root');  SELECT \* FROM Q14; |
| --- |

1 Mark for correct view

1 Mark for correct condition

**1.15** SQL statement which produces the given records and permanently saves them. Paste the code in the space provided below [3]

| CREATE VIEW Qproducts AS  SELECT ProductID, ProductName, Price, Weight, Stock, SupplierID  FROM Products  WHERE ProductID IN (1002, 3003, 2002, 1004)  ORDER BY Stock ASC;  SELECT \*  FROM Qproducts; |
| --- |

1 view created

1 mark for correct Select statements

2 mark for correct condition

**1.16** SQL statement which produces the following output.

Paste the code in the space provided. [3]

| CREATE VIEW Q1\_16 AS  SELECT s.CompanyName, s.ContactNo, p.ProductName, p.Price  FROM Suppliers s  RIGHT JOIN Products p  USING(SupplierID);  SELECT \* FROM Q1\_16; |
| --- |

1 Mark for creating view

1 Mark for correct fields [no mark for incorrect fields]

1 mark for the correct statement and correct records.

1.17 SQL code with output correctly formatted. [6]

| CREATE VIEW Q1\_17 AS  SELECT SUM(Price) AS Total\_unit\_price,  ROUND(AVG(Price),2) AS Average\_price,  COUNT(ProductID) AS NumbeR\_of\_products  FROM Products;  SELECT \* FROM Q1\_17; |
| --- |

1 Mark for permanent SQL statement

3 marks for the correct functions.

1 marks for the headings

1 mark for formatting output

1.18 SQL statement to display the number of distinct supplierid from the products table.   
 Add the SQL statement in the space provided. [3]

| CREATE VIEW Q1\_18 AS  SELECT DISTINCT SupplierID  FROM Products;  SELECT \* FROM Q1\_18; |
| --- |

1 Mark for creating a permanent sql statement

2 Mark for correct SQL statement producing correct output

[lecturer to award a mark if the statement is partially correct]

1.19 SQL statement to produce the following output as shown below.   
 Make sure you save the output permanently as Q1.19.  
 Add the SQL statement in the space provided. [4]

| CREATE VIEW Q1\_18 AS  SELECT COUNT(ProductID) AS 'Count(ProductID)', SupplierID  FROM Products  GROUP BY SupplierID;  SELECT \* FROM Q1\_18; |
| --- |

1 Mark for permanently saving the solution on to the databases

3 Marks for correct SQL producing correct output

1.20SQL Statement to produce a given output. Add the code in the space provided.

[5]

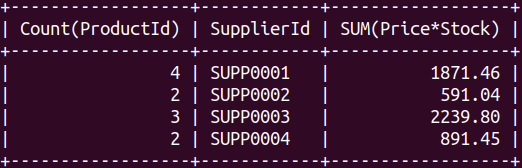
| CREATE VIEW Q1\_20 AS  SELECT COUNT(ProductID) AS 'Count(ProductID)', SupplierID,  SUM(Price\*Stock) AS 'SUM(Price\*Stock)'  FROM Products  GROUP BY SupplierID;  SELECT \* FROM Q1\_20; |
| --- |

1 Mark for saving output permanently

1 Mark for correct headings

2 Marks for correct functions

1 Mark for correct output



Make sure the result is visible in your database.   
 Add the SQL statement in the space provided in the LMS.

1.21 SQL Statement to produce a given output.   
 Add the code in the space provided. [2]

| UPDATE Products  SET Price = 15.95,  Weight = '1 kg',  Stock = 18  WHERE ProductID IN (1004);  SELECT ProductID, ProductName, Price, Weight, Stock, SupplierID  FROM Products  WHERE ProductID IN (1004); |
| --- |

1 mark for correct SQL command

1 Mark for correct output

1.22Create a database dump called FruitMarketYourname.sql and upload it into the   
 LMS. [1]

| mysqldump -u root -p FruitMarket > FruitMarketDeno.sql |
| --- |

1 Mark for database dump backup

1.23 What is the purpose of applying ON Delete cascade in a foreign key constraint?   
 [2]

| It means that when/if content is deleted from a parent table that the data from the child table will also be deleted automatically in the child table. |
| --- |

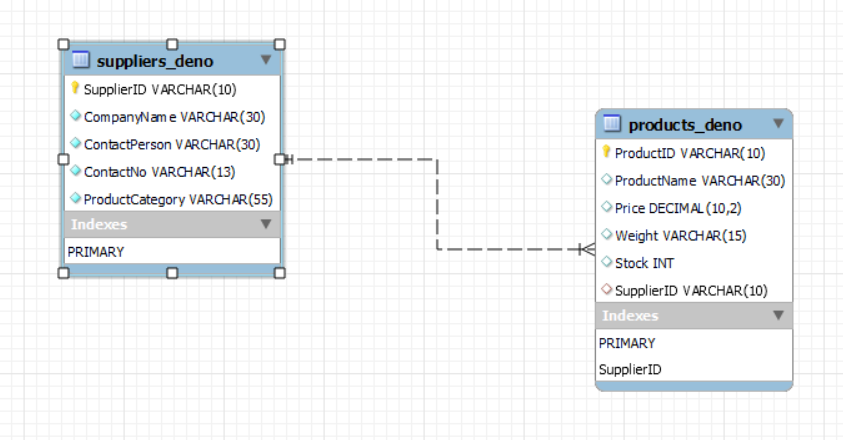
1.24 What is the difference between composite key and candidate key? [2]

| A composite key allows us to uniquely identify two or more columns that have been joined.  A candidate key also allows for unique identification, but by a cluster of multiple or single keys that uniquely identify rows in a table. |
| --- |

1.25 What is the difference between SQL and DBMS [2]

| SQL-Structured Query Language is just used to talk/communicate with the Database.  DBMS-Database Management System is a way of keeping data on a computer and lets the users define, create, maintain and control access to the database. |
| --- |

1.26 EER Diagram [3]



1.27 Write a MySQL statement to extract the following records in that order. [3]

| UPDATE Products\_Deno  SET Price = 12.36  WHERE ProductID = 1004;  SELECT ProductID, ProductName, Price, Weight, Stock, SupplierID  FROM Products\_Deno  WHERE ProductID IN (4001, 1004, 2002, 1002)  ORDER BY Stock DESC; |
| --- |

1.28 Write an SQL statement to for products table to produce the following records: [5]

| CREATE VIEW Q1\_28 AS  SELECT SupplierID AS 'supplierid', SUM(Prices) AS 'prices',  GROUP\_CONCAT(ProductName) AS 'group\_concat(productName)'  FROM Products\_Deno  GROUP BY SupplierID;  SELECT \* FROM Q1\_28; |
| --- |

**QUESTION 2 NORMALISATION (30 Marks)**

Consider the following unnormalised table.

| **User\_ID** | **User\_Name** | **MSE\_ID** | **Rec\_Date** | **Subject** | **Text** | **Srvr\_ID** | **Server\_Name** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2301 | Smith | 54101 | 05/07 | Meeting Today | There is... | 3786 | IMAP05 |
| 2301 | Smith | 54098 | 07/12 | Promotions | I like to... | 3786 | IMAP05 |
| 2301 | Smith | 54445 | 10/06 | Next Assignment | Your next... | 3786 | IMAP05 |
| 5607 | Jones | 54101 | 05/07 | Meeting Today | There is... | 6001 | IMAP08 |
| 5607 | Jones | 54512 | 06/07 | Lunch? | Can you... | 6001 | IMAP08 |
| 5607 | Jones | 54660 | 12/01 | Jogging Today? | Can you... | 6001 | IMAP08 |
| 7773 | Walsh | 54101 | 05/07 | Meeting Today | There is... | 9988 | EMEA01 |
| 7773 | Walsh | 54554 | 03/17 | Stock Quote | The latest... | 9988 | EMEA01 |
| 0022 | Patel | 54101 | 05/07 | Meeting Today | There is... | 2201 | EMEA09 |
| 0022 | Patel | 54512 | 06/07 | Lunch? | Can we... | 2201 | EMEA09 |

Normalize the above table to Third Normal Form (3NF). Show all the tables

First Normal Form 2 tables (9 marks) 1 mark for each correct field.

Second Normal Form 3 Tables (10 Marks) 1 mark for each correct field.

Third Normal Form 4 Tables (11 Marks) 1 mark for each correct field.