**Midterm Exam**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Part 1 - Define the following in your own words and share a metaphor from like or software if possible:**  
Database: Imagine a giant digital filing cabinet, neatly organized with folders and labels.

Table: Each folder in the cabinet is a table. It holds related information, like a "Customer Table" with names, addresses, and phone numbers.

Field: Think of each label on a folder as a field. It tells you one specific thing about the information inside, like "Phone Number" or "Email Address."

Record: A whole folder full of information is a record. It's like a complete profile for one customer, including all their details in one place.

View: Imagine using a special lens on your cabinet. You can see specific things, like only customers in a certain city, without opening every folder. That's a view!

Stored Procedure: Think of a recipe for cooking data. You mix and match instructions (commands) to get the results you want, without repeating yourself each time.

CRUD Commands: These four actions are like the basic tools in your data kitchen:

Select: Grab the ingredients (data) you need.

Insert: Add new ingredients (data) to the mix.

Update: Change or fix the ingredients (data) you already have.

Delete: Throw away the ingredients (data) you don't need anymore.

Bonus: FROM, WHERE, ORDER BY:

FROM: Tells you which folder (table) to look in for your ingredients (data).

WHERE: Picks out the specific ingredients (data) you want based on their qualities.

ORDER BY: Arranges your chosen ingredients (data) neatly, like from A to Z or newest to oldest.

**Part 2- Building Tables and Populating them**

9) Create the Customers table with the following specifications, then paste your design below showing Identity Specification for the primary key field:

A screenshot of a computer

Description automatically generated

10) Populate the Customers table with the following sample data plus 5 new customers of your own. Please note that your CustomerID’s may not match mine and that is fine:

A screenshot of a computer

Description automatically generated

11) Create a new table named “Purchases”. This table will store the final costs, purchase date, payment type, status of the purchase, change given, and the ID of the customer who made the purchase. This is usually the data at the top and bottom of your receipt. Fields are given below:

PurchaseID – BigInt, Primary Key, Identity Specification Set to Yes

Purch\_Date - DateTime

Status – NvarChar(8) – Potential values: COMPLETE, FAILED, VOID

SubTotal – Float

Discount – Float

Taxes – Float (If you calculated this, it would be SubTotal \* (1 – Discount) \* 0.07 in RI)

Total – Float (If you calculated this, it would be SubTotal \* (1 – Discount) + Taxes)

Payment\_Type – NvarChar(5) – Potential values: CASH, CC, DEBIT, CHECK, GC

Payment\_Amount – Float

Change – Float

CustomerID – BigInt – Only place Customer ID’s that come from your Customers table…Connecting this table with the Customers table.

**Please post image of design, including primary key and showing Identity Spec for Primary Key. Do not shrink image.**

**A computer screen with a white screen

Description automatically generated**

12) Create 15 Purchase records, with at least 2 customers coming in twice, 1 customer coming in three times. Think of you going to stop and shop more than once, then you should have more than one purchase. **Please post image of records. Do not shrink image.**

A screenshot of a computer

Description automatically generated

13) Create a new table named “PurchasedProducts”. This table will store the ProductID’s, Quantity of this product, the Price Per Item, and the PurchaseID associated with this record. This is usually the data in the middle of your receipt listing the products, quantities, and prices. PLEASE use existing PurchaseID’s from the Purchases table as well as existing ProductID’s from the Products table. **Please post image of design, including primary key and showing Identity Spec for Primary Key. Do not shrink image.**

Fields are given below:

PurchProdID – BigInt, Primary Key, Identity Specification is set to Yes.

PurchaseID – BigInt (These will match those that exist in the Purchases table.

ProductID – BigInt (These will match those that exist in the Products table.

Qty – Int (This will be the quantity being bought of this particular item.)

PricePerItem – Float - Current cost, because there are sale prices sometimes.

A computer screen with a white screen

Description automatically generated

14) Create 15 PurchasedProducts records, with at least 1 purchase with 2 **different** products, 1 purchase with 3 **different** products, and one purchase with 5 **different** products. Think of you going to stop and shop and having to pay for each item individually…it does not make sense. In reality, you may one purchase payment for one or more products. You should also have records where the quantity of the **same** product is more than one. Use existing PurchaseID’s from the purchases table as well as ProductID’s from the Products table. **Please post image of records. Do not shrink image.**

**A screenshot of a computer

Description automatically generated**

**Part 3: Create the following SQL Queries** **(Show SQL Code and Results)**:

1) Show me the **two** ways that that you can show all fields with all records from the Purchases table.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

2) Show me a query that has all records, but does not include the Customer ID and Change given fields.

A screenshot of a computer

Description automatically generated

3)  Show all Purchases records that have a Total greater than $75.00

A screenshot of a computer

Description automatically generated

4)  Show all Purchases records that have a Total less than $10.00

A screenshot of a computer

Description automatically generated

5) Show me the two ways that you could get all Purchases records that are **between** $11 and $74

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

6) Show me all Purchases that were not paid with Gift Cards (GC).

A screenshot of a computer

Description automatically generated

7) Show me the **two** ways to get all Purchases that included all Cash, Check, Debit purchases.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

8) Create a View that Pulls up all PurchasedProducts, but has an Alias that calcs Qty \* PricePerItem.

A computer screen with a white screen

Description automatically generated

9) Create a Stored Procedure that receives PurchaseID and pulls up all Purchase Products for that purchase.

A screenshot of a computer

Description automatically generated

10 Create a Stored Procedure that receives a CustomerID and pulls up all Purchases for that customer.  Sort by purchase Date.

**A computer screen with a white screen

Description automatically generated**

**A student asked me what I meant by my term of having customer overlap. So Here is my explanation, just in case it is helpful to you folks. I have not looked at the midterms yet, so if there is a tweak, please do it before the due date/time.**

The purchases table will be using existing CustomerID's from the Customers table.  
Example: You are customer # 1234. When you make a purchase, it will store your Customer ID for that purchase

One or more customers will make more than one purchase in the Purchases table.  
Example: If you like a store, then you are likely to go there more than once. So, you should see your CustomerID (1234) in multiple purchases/records.

So....when you enter example purchases, use the Customer ID's that already exist in your other table AND maybe have one or two customers have more than one purchase record.

Let me make sure that this one-to-many relationship (one customer to many purchases) will be similar to your PurchasedProducts (or PurchasedItems) table. Each purchase can have one or more products. So...I noted that you folks should have customers buy a range of 1 to 5 items for each purchase you added to the purchases table.

Example: You (cust ID 1234) have made two purchases (purch ID 456 and 789).  
In purchase 456 you bought 2 products (# 321 and 432).  
In purchase 789 you bought 4 products (# 211, 311, 411, 511)

So your purchasedProducts/items table will have five records for you (one for each different item in a purchase)  
If you bought 2 of #211, you would just set its quantity field to 2.

Hope this helps,  
Scott