**Midterm Exam**

**Name: Sebastian Oquendo**

**Part 1 - Define the following in your own words and share a metaphor from like or software if possible:**  
1) Database: a collection of tables, used to store various types of data  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2) Table: a database object; used to store its data in rows and columns  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3) Field: the area used to store value in a column, within a table  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4) Record: the area used to store value in a row  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5) View: a way to view a table’s contents based upon the parameters of a query; essentially viewing the table’s data through a filter.  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6) Stored Procedure: a set of sql code that can be saved to later be used on command; either with a set of predefined parameters or by user input

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7) What are the four core commands for finding and manipulating data?: Hint they start with S,I,U,D

Select, Insert, Update, Delete   
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

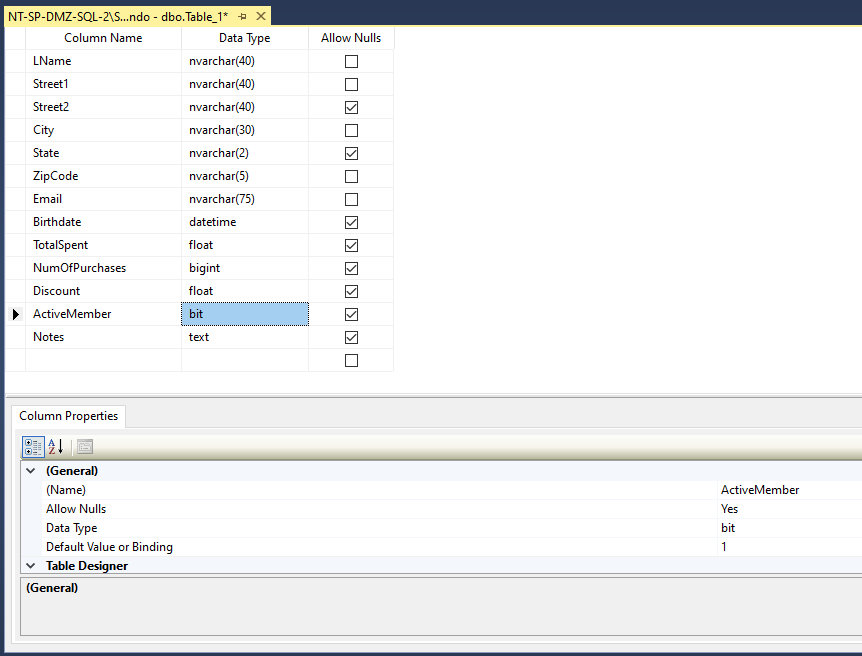
8) Explain FROM, WHERE, ORDER BY:   
  
“From” is identifying which table within the database the data is to be called from. “Where” is used to determine the specific instance or circumstance in which the user desires to see part of a table’s data, can be used in conjunction with other commands to yield even more specific results. “Order By” is used to determine which column the search results will be displayed in order by. Sorted in ascending order by default, users need to specify if they desire descending order instead.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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

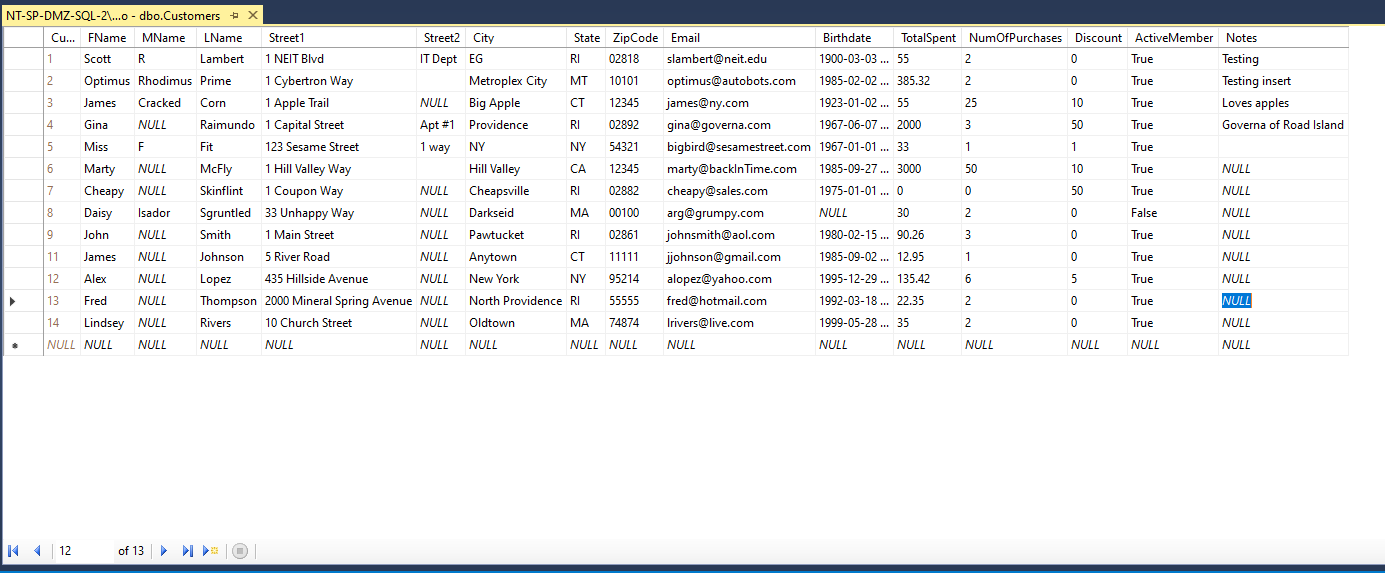
Description automatically generated



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)

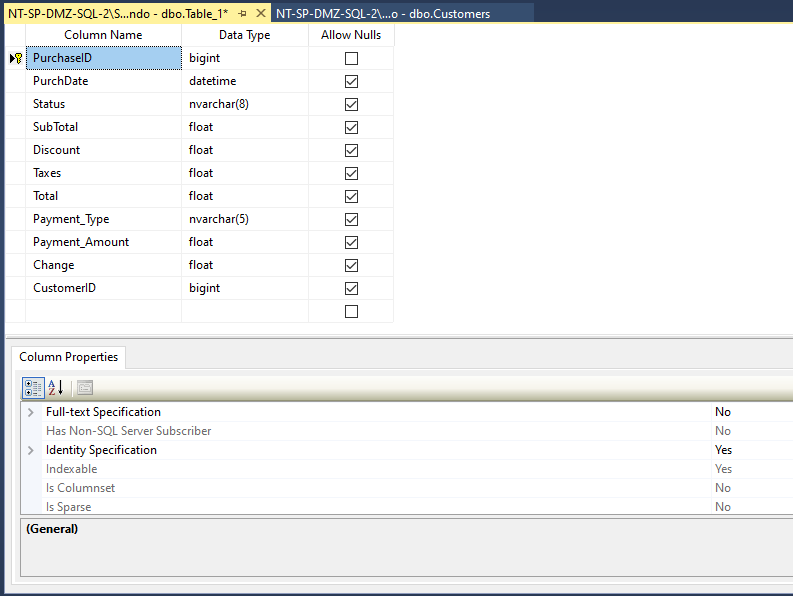
Payment\_Type – NvarChar(5) – Potential values: CASH, CC, DEBIT, CHECK, GC

Payment\_Amount – Float

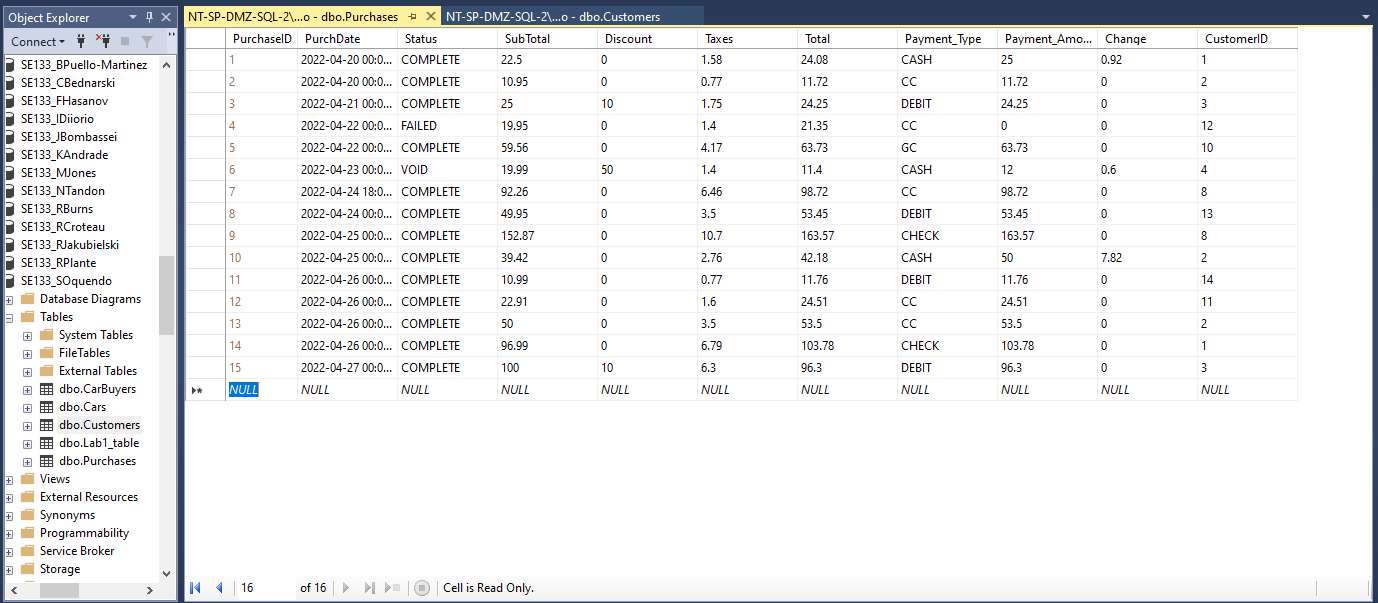
Change – Float

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.**

****

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.**



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.

14) Create 15 PurchasedProducts records, with at least 1 purchase with 2 items, 1 purchase with 3 items, and one purchase with 5 items. 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. 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.**

**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.

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

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

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

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

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

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

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

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

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

**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