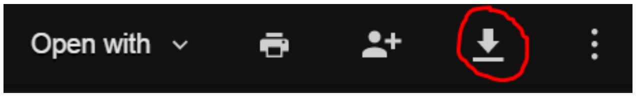
Northeastern University
**Khoury College of
Computer Sciences**

CS1100: Assignment 13

Creating and Querying Databases

To complete this assignment you must submit an electronic copy to BlackBoard by the due date. Download the starter files and save the files under the names *CS1100.LastName.A13* and where *LastName* is your last name.

To download the files from Google Drive, click the download button -- do not open the files in Google.



Knowledge Needed

This lab requires the following Access techniques:

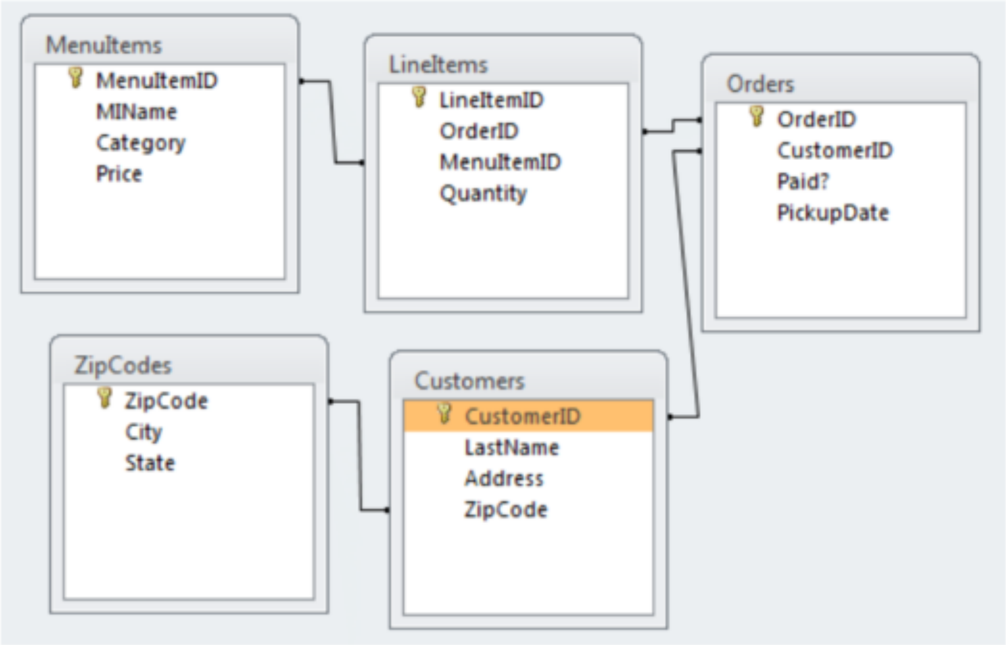
- Building queries using multiple tables
- Use GROUP BY to eliminate duplicates
- Apply aggregation functions to groups as well as tables

The Database

This database represents customer and order information for the *Milk & Honey Bakery*. We have been asked to help the owner understand his sales better and therefore have been given certain queries. For each of the problems assume that the data in the database might change. Your query should give the correct results for whatever data happens to be in the database. If the problem says that your query result should not have duplicate entries for a particular field, it is not sufficient that there are no duplicate entries for the particular data in the database. Your query must give unique values for that field whatever data happens to be in the database

Database Structure

To help you in designing queries, the database structure is shown below:



Queries (5 Points Each)

We use a systematic naming convention for queries and subqueries because we will have several subqueries that we can run independently.

For a query numbered X we might have a tree of subqueries. The final query we call QLastnameX-Optional where Optional is a name which reminds us what the query is about, e.g., QSmith01-OrderTotals.

Save your *queries* using the following naming pattern: QLastnameX-Optional where Lastname is your last name and X is the query number.

Save your *subqueries* using the following naming pattern: QLastnameXA-Optional, where X is the query number and A is a letter. For instance, if Schmidt wrote a subquery for Query 1 it should be named QSchmidt01a-Winners. If Schmidt needed a second subquery for Query 1, it would be called QSchmidt01b-Winners.

Optional is a name which reminds us what the query is about, e.g. QSchmidt01-OrderTotals or QSchmidt01a-ExtPrices.

For a subquery of X, we use a name QLastnameXY-Optional:

- Y is a,b,c, ... E.g., QSmith01a-OrderTotals is an example of a subquery for the main query QSmith01-OrderTotalsPerCustomer. (This is not an example based on the queries from the assignment.)
- Y is a,b,c, ... E.g., QSmith14a-ExtendedPrices is an example of a subquery for the main query QSmith14-OrderTotals. (This is not an example based on the queries from the assignment.)

The reason for the optional suffix is easier query reusability. It will be easier to find the query we want to reuse. You do not need to rename a subquery if it is reused as a subquery for a different query.

You may reuse subqueries in other questions without renaming them.

Follow the query design principle discussed in the lesson, abbreviated as WAFE (Widening-Aggregation-Filtering-Elimination of Duplicates).

1. Create a query to list the LastName, ZipCode, State and Customer ID of all customers regardless whether they placed an order or not.

2. Create a query to list the CustomerID and OrderID of order(s) placed by that customer.

3. Create a query to list the CustomerID, LastName and ZipCode of all Customers that placed one or more orders. Your query result **should not have duplicate entries** for the same Customer.

4. Write a query to find all the MenuItems that were ordered by some customer. List the MenuItemID and MIName. Your query result should not have duplicate entries for menu items.

5. Create a query to find all the Cities where at least one Customer from that city placed an order. Your query result should not have duplicate cities.

6. Create a query to find the dates on which paid orders will be picked up (*i.e.*, the Paid? field equals TRUE). Your query result should not have duplicate entries for a date. Only include the date as part of your result; do not include any additional fields.

7. Create a query to find all Orders that contain a Line Item with a Quantity of at least 3 for that MenuItem. List the Order ID. Your query result should not have duplicate entries for that order.

8. Create a query to find the Customers who placed one or more Orders containing a Line Item with a Quantity of at least 3 for a particular MenuItem. List the Customer's LastName and CustomerID. Your query result should not have duplicate entries for that customer.

9. Create a query to find the Customers who placed one or more orders that have been paid. Your query result should not have duplicate entries for that Customer. List the CustomerID and Last Name. If a customer placed multiple orders but only paid for some of them, then the customer should be included in the result.

10. Create a query to find the Customers who placed one or more Orders that contained a pie. List the Customer's Last Name and CustomerID. Your query result should not have duplicate entries for that customer.

11. How many Customers are there from each City? List the City and the number of customers from that city.

12. How many Orders did each Customer place? List the Customer ID, Last Name and the number of Orders placed.

13. Which Customers placed at least 3 orders? List the Customer ID and Last Name.
14. What is the total amount for each Order? List the Order ID and the amount for each order. Recall that the order total is the sum of all the extended prices of each line item that is part of the order. Extended price for a Line Item is [Quantity] * [Price].
15. How much did each Customer that placed an Order spend on all of their orders combined? List the Customer ID, Last Name and the amount spent.
16. What is the maximum extended price for each order? (Recall that an Extended Price for a Line Item is Quantity * Price) List the Order ID and the max extended price for each order.
17. What is the maximum extended price of all Orders? (Give a single number)
18. What is the maximum total of all Orders? (Give a single numbers)
19. Which Customers placed an Order that included a Line Item with an extended price of $50 or more? List the Customer IDs with no duplicates.
20. Which Orders have a total amount of at least $200? List the Order IDs and the totals.

After you have completed these tasks, your modified version of the database should contain a number of additional queries. *Close Access* and then submit this modified Access database to Blackboard.

Grading Rubric:

- -10pts for each incorrect query (partial credits will be given for each correct step)
- -3pts for each query that does not follow WASE/WAFE
- -5% if the naming convention isn't followed