# **Chapter 4 Lecture Terms / Questions**

## **General SQL**

1. Native table- any table that is part of a database(not virtual)
2. Virtual table (virtual data views)- obtained by querying the database to form any view of the data we desire
3. SQL- Structured Query Language
   1. DML- Data Manipulation Language
   2. DDL- Data Definition Language
4. ANSI- American National Standards Institute, means there is an established set of SQL statements that every database management system recognizes
5. What can SQL statements do?
   1. Sort records
   2. Choose fields
   3. Choose records
   4. Cross reference tables
   5. Perform calculations
   6. Provide data for database reports
   7. Modify data
6. SQL Wildcards
   1. MS Access: \*
   2. ANSI Standard: %
7. What do we do when a table name has a space in it?
   1. [Place it in Brackets]
8. How do we refer to a particular field in a table in a SQL statement?
   1. TableName.FieldName
9. On page 4-5 what will be put in the MySQL field, and what statement is that field used in?
   1. A command object, the data adapter that fills the table
10. Still on page 4-5 what statement can you use to determine how many records are returned by a SQL query?
    1. myTable.Rows.Count
11. What is a DataGridView control (what can it do)?
    1. The DataGridView control allows us to view and edit an entire database table by setting just one property (DataSource)
12. On page 4-10 what is the purpose of the try-catch structure used in the code?
    1. Without it, if we make a small error in a SQL statement, the program will stop. With it, we get a message indicating our mistake and are allowed to continue

## **SELECT Statement**

1. On page 4-11 what happens when we type **SELECT \* FROM Titles** into the text box? (How much of the table does it show?
   1. It displays everything from the titles table
2. How would you change the SQL statement (**SELECT \* FROM Titles**) to see all the records in the Authors table? Change Titles to Authors
3. How do you change the SELECT statement to only look at specific fields?
   1. You change the \* to be the specific fields you want
4. In the statement (**SELECT Title,Year\_Published FROM Titles**), why don’t you have to say **Title.Titles** & **Title.Year\_Published**?
   1. You are only selecting fields from the titles table
5. What does the **DISTINCT** keyword do?
   1. Can be used with SELECT to restrict the returned records to one per unique entry for the field. That is, there are no duplicate entries

## **ORDER BY Clause**

1. What does **ORDER BY** do and where does it go in the SQL SELECT statement?
   1. Sorts the returned records in some other order
2. How do you specify descending order?
   1. If you want the sort to be in descending order, the FieldSort name is followed by the keyword DESC.
3. How do you order by multiple fields?
   1. You place commas in between the sort fields

## **WHERE Clause**

1. What does **TOP** do and how is it included?
   1. If you want to restrict the number of records returned by a SQL statement that orders the returned records, you can use the TOP keyword with SELECT. TOP n returns the first n records
2. What does **WHERE** do and how is it included?
   1. This clause specifies some criteria that must be met in forming the data table
3. In a WHERE clause, what do **Between**, **In**, and **Like** do? Give examples of their use.
   1. Within a specified range
   2. Specify a list of values
   3. Wild card matching

## **Multiple Tables**

1. Write the example of the 2 table query on page 4-29
   1. SELECT Titles.Title,Publishers.Company\_Name

FROM Titles,Publishers

WHERE Titles.PubID = Publishers.PubID

AND Publishers.Company\_Name = 'QUE CORP'

ORDER BY Titles.Title

1. Write the 3 table query on page 4-30
   1. SELECT Authors.Author,Titles.Title,Publishers.Company\_Name
   2. FROM Authors,Titles,Publishers,Title\_Author
   3. WHERE Titles.ISBN = Title\_Author.ISBN
   4. AND Authors.Au\_ID = Title\_Author.Au\_ID
   5. AND Titles.PubID = Publishers.PubID
   6. ORDER BY Authors.Author
2. How do you “alias” the name, and why would you do it? (page 4-31)
   1. you can alias the name, or change it to something more meaningful using the AS clause.
3. What does the INNER JOIN clause do, and what part of the SQL statement does it go in? Give an example of it from the code.
   1. When combining tables, the SQL INNER JOIN clause does the same work as the WHERE clause. It belongs with the FROM statement
   2. SELECT Titles.Title, Publishers.Company\_Name

FROM Titles INNER JOIN Publishers

ON Titles.PubID = Publishers.PubID

WHERE Publishers.Company\_Name = 'QUE CORP'

ORDER BY Titles.Title

1. How does a nested INNER JOIN create and use a temporary table?
   1. You place the inner join statement within parantheses
2. Why and how are parentheses used in a nested INNER JOIN? Give an example of a SQL statement with a nested INNER JOIN.
   1. They are used to do the above action.
   2. FROM (Publishers INNER JOIN Titles ON Publishers.PubID=Titles.PubID)
3. How is an INNER JOIN different from an OUTER JOIN?
   1. The INNER JOIN only retrieves records that have a match on both sides of the JOIN
   2. If you want all records returned, whether there is a match or not, you need to use what is called an OUTER JOIN
4. What does a **RIGHT OUTER JOIN** do? Give an example and tell which table it uses all the records from.
   1. includes all records from the second-named table (the right-most table), even if there are no matching values for records in the first-named (left-most table)
5. What does a **LEFT** **OUTER** **JOIN** do? Give an example and tell which table it uses all the records from.
   1. includes all records from the first-named table (the left-most table), even if there are not matching values for records in the second-named (right-most table)

## **SQL Functions**

1. What does the UCase function do? Give an example.
   1. Changes text to all in Upper Case
   2. SELECT UCase(Titles.Title) AS Title FROM Titles
2. What does the Left function do? Give an example.
   1. It selects an n numbers of leftmost characters of data to display
   2. SELECT Left(UCase(Titles.Title), 10) AS Title FROM Titles
3. Give the example of math being done on Authors.Year\_Born on page 4-39. What is the alias used for the result?
   1. It subtracts the year, 2006, from the year the author was born to calculate the age
   2. The alias is Age
4. What does NULL mean?
   1. A special value meaning there is nothing there - this is not the same as an empty string or blank space
5. Give the example of IS NOT NULL being used in the SQL statement.
   1. SELECT Authors.Author,(2006-Authors.Year\_Born) AS Age

FROM Authors

WHERE Authors.Year\_Born IS NOT NULL

## **Aggregate Functions**

1. For each of the functions, tell what is does and give an example.
   1. AVG- Average value of the field
   2. COUNT- Number of entries for the field
   3. FIRST- First value of the field
   4. LAST- Last value of the field
   5. MAX- Maximum value of the field
   6. MIN- Minimum value of the field
   7. SUM- Sum of the field values
2. How are aliases used with these functions in the SQL code?
   1. They are used to label the returned values from each function
3. How does the GROUP BY clause work? Give an example.
   1. Lets you determine records with duplicate field values
   2. GROUP BY Publishers.State
4. How does HAVING work with GROUP BY? Give an example.
   1. HAVING qualifier to further reduce the grouping obtained with a GROUP BY clause. Say you only want to display states starting with the letter M. This SQL statement will do the trick
   2. HAVING Publishers.State LIKE 'M%'
5. How can you use Access to help you construct SQL statements?
   1. You can build your queries in access, and it builds the SQL statements for you as you create the query in access
6. Know how to build a query in Access and view the SQL code for it.
7. Also know how to do this with the Data Wizard.