# Lab 2 – Week 2 (SELECT, RANGE, ORDER)

## DUE: Thursday, Week 3, midnight

This week’s lab continues using the SELECT command and learning the GUI iNavigator.

## IMPORTANT

*When following these examples, it is often not wise to copy and paste from this document. MS Word and PDF treat certain quotes differently than plain text and this results in errors in your SQL code because the wrong type of quote is used. Honestly, it is a better learning practice to type it in yourself.*

## Background

### Introduction to SQL

SQL – Structured Query Language

SQL is a standardized universal language used to access relational databases. There are some other types of databases such as network and hierarchical, but these mostly become obsolete as relational databases started being introduced through the 1990s.

In this course we are going to learn SQL. With small variations, the SQL dialect used on the iSeries will be similar to that used by Oracle, Microsoft SQL Server, MySQL and other database management systems.

Simplistically speaking there are just a few things you can do with data. You can insert data or store data, edit data, retrieve and manipulate it for viewing, or you can delete data. We are going to start the SQL by looking how we retrieve data and manipulate the data that we retrieve. In later lessons we will learn how to create tables to store the data into the tables, change and delete that data and change and delete tables.

### Retrieve data from the database tables

SELECT statements are used to extract data from tables in the database. The select statement has three actions that it can do on tables.

**3 ACTIONS ON TABLES**

1. PROJECTION
2. SELECTION
3. JOIN

Looking at simple tables, the 3 actions are as follows

#### Projection

The SELECT statement, selects columns of data to retrieve and display

|  |  |  |  |
| --- | --- | --- | --- |
| 11111 | Last name1 | First name1 | Other data about 1 |
| 2222 | Last name2 | First name2 | Other data about 2 |
| 3333 | Last name3 | First name3 | Other data about 3 |
| 4444 | Last name4 | First name4 | Other data about 4 |
| 5555 | Bass name5 | First name5 | Other data about 5 |
| etc |  |  |  |

#### Selection

The SELECT statement retrieves rows of data.

|  |  |  |  |
| --- | --- | --- | --- |
| 11111 | Last name1 | First name1 | Other data about 1 |
| 2222 | Last name2 | First name2 | Other data about 2 |
| 3333 | Last name3 | First name3 | Other data about 3 |
| 4444 | Last name4 | First name4 | Other data about 4 |
| 5555 | Bass name5 | First name5 | Other data about 5 |

#### Join

The select statement will join columns from two different tables based on a common data type in a column

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11111 | Last name1 | First name1 | Other data about 1 | 177 |
| 2222 | Last name2 | First name2 | Other data about 2 | 234 |
| 3333 | Last name3 | First name3 | Other data about 3 | 177 |
| 4444 | Last name4 | First name4 | Other data about 4 | 341 |
| 5555 | Bass name5 | First name5 | Other data about 5 | 657 |

|  |  |  |
| --- | --- | --- |
| 177 |  |  |
| 190 |  |  |
| 234 |  |  |

### Exercises using SELECT

Login to iNavigator and try out commands:   
(NOTE: There is no submission required for this part, this is just practice)

#### Basic Format

SELECT *<columns names comma separated>*

FROM *<tablename>*;

#### SELECT TYPES

* ALL columns (fields) 🡪 \*
* Specific columns (comma separated)
* Any order of columns
* Column Headings (Aliases) -- AS
* Arithmetic
* Arithmetic and Heading (Alias)
  + operator precedence
  + use of parenthesis 12 \* Salary + 100 vs 12 \* ( salary + 100)   
    (Order of Operations)
* Null values - null is NOT the same as zero or blank space
  + Arithmetic expressions containing NULL values (hmmmm….)
* Concatenation - example LAST || ‘, ‘ || FNAME watch spacing
* Literal character strings - use of single quotes   
  LNAME || ‘ is an employee in ‘ || DEPARTMENT
* Distinct

### RESTRICTING (Filtering) AND SORTING

When retrieving data from the database you may need to do the following:

* Restricts the rows of data that are displayed (only display rows that match a given criteria)
* Specify the order which the rows are displayed (alphabetical, numeric, chonological).

#### Limiting Rows of Filtering using SELECT

Example:

SELECT lastname, salary

FROM student

WHERE salary > 10000;

## Lab Submission

***Your submission will be a single text-based SQL file with the solutions provided (text based answers are to be written in commented code.***

**Late submission** will be accepted and reviewed, but will always receive a mark of zero (0). Solutions are posted shortly after the due date and therefore late submissions are compromised.

### Example Submission

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

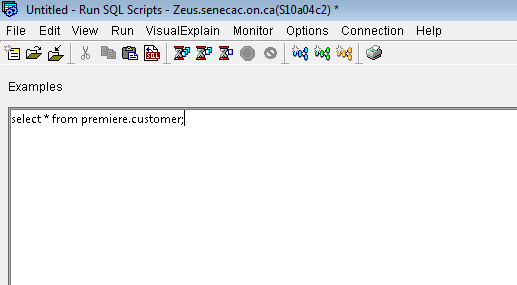
-- Name: Your Name

-- ID: #########  
-- Date: The current date  
-- Purpose: Lab 2 DBS201  
-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
-- Question 1 – write a brief note about what the question is asking  
-- Q1 SOLUTION --  
  
SELECT \* FROM table;  
  
-- Question 2 – blah blah blah

## Deliverable Items

Using iNavigator, initiate a session to enter SQL commands using the login and password provided to you. i.e. login and open the Run an SQL script window.

Make sure you have an open window for running an SQL script. You should see a screen with something like this in the left top area but with no words in the example area.



1. Create a “comment header” as per the example submission above customized with your own information.
2. True of False, this connection can only be done from the lab computers   
   (Answer in a comment line clearly labeled for Q#).

Using the PREMIERE schema window:

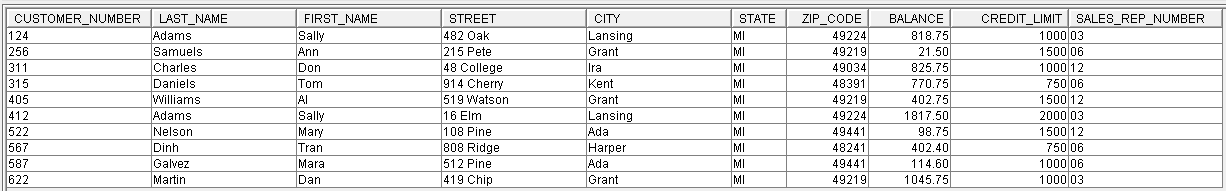
1. How many tables do you see in the PREMIERE collection that do NOT start with Q (NOTE: Tables that start with Q are system tables that should not be touched)
2. In the examples window, type the following:

-- Q4

SELECT \*

FROM premiere.customer;

*(It could have been put all on 1 line and for something short like this one line is fine. However just as C programming had a better and more readable layout, the above method that splits the line on key words is better to read.)*

In the bottom pane (you may have to expand the pane to see it) will be a graphic display of the content of the CUSTOMER table that looks something like the following: 

Do the same for the other tables. Look at the data and get an idea what is contained in these very small tables.

SELECT \* FROM part;

SELECT \* FROM orderline;

SELECT \* FROM orders;

SELECT \* FROM salesrep;

(one line for each is okay here as they are really short).

1. To demonstrate how to do a better more user friendly title (otherwise known as an “Alias”, execute the following

SELECT last\_name AS ”Last Name”

FROM premiere.customer;

1. Do the same as number 4 above, but select Customer Number, First Name, and Balance.  
   Note that the request is for Customer Number (2 words) but you must use the name as stored in the database table. Make the output titles look like Customer Title and First Name. See below.

SELECT customer\_number "Customer Number",   
 first\_name "First Name", balance

FROM premiere.customer;

1. How many rows are in the salesrep table?
2. What is the name of salesrep 14?
3. What is the credit limit for Mary Nelson?
4. Enter the following SQL command:

SELECT COUNT(\*) “Total Rows” FROM customer;

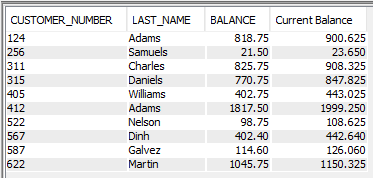
Did anything unexpected happen?  
 comment out the above line and create the correct code and execute it.

1. Enter the following SQL:

SELECT COUNT(\*) AS “Total Rows”

FROM premiere.customer;

What did the “AS” do to the output? (answer in a comment under the above code)

1. Write the SQL Code that will show the customer number, last name, their balance and the result of increasing the balance by 10%. Give the balance column a new alias of “Current Balance”. Your output should look something like:   
   Note the capitalization of the column headings….add a comment explaining the difference and why this is the case.
2. Write the SQL code that will list customer number, order number and date of order.
3. List the orders in the orders table. Put the columns in the order of date, order number, and then customer number. The output should display the date from newest to oldest.
4. Produce a list showing Part\_Number, Part\_Description, On\_Hand, and Price sorted by Warehouse and Class.
5. Produce a list of part numbers, descriptions and the quantity in stock of the part in the following format:  
    Part Number AX12 is an Iron and there is 3 in stock  
    Part Number AZ52 is a Dartboard
6. Using the orderline table show what part numbers were ordered. (*Hint*: do not repeat the number if it was ordered more than once.
7. List all salesrep data from all sales people that earn more $3000 commission.
8. List the part number, description, number ordered and quoted price. (note that this is a tricky one as information comes from multiple tables)
9. List the Customer’s last name and first name followed by the order number, part description and number ordered for all parts that start with the letter C.

## Submit your completed .sql file to Lab 2 submission box in blackboard