Basic SELECT Clauses

After completing this module, you will be able to:

- **Distinguish between 3 classes of queries.**
- **Select rows and columns from a table based upon equality.**
- **Use ORDER BY to sort result sets.**
- **Alias column names for providing new names.**
- **Use DISTINCT to project a distinct list of result rows.**
- **Apply WHERE constraints to conditionally return rows.**
- **Project character or numeric literal values.**
- **Write SQL in a way that is more structured and easier to read.**

SQL: Structured Query Language

- **A complete data access and maintenance language**
- **Designed for Relational Database Management Systems (RDBMS)**
- **An industry standard for relational databases**
- **A non-procedural language**
- **Three defined SQL standards:**
    - **SQL-89 (SQL 1)**
    - **SQL-92 (SQL 2)**
        - ✓ Entry Level
        - ✓ Intermediate Level
        - ✓ Full Level
    - **SQL-99 (SQL 3)**
        - ✓ Core
        - ✓ Enhanced

Three SQL Classifications

There are different classes of SQL requests.

Data Definition Language (DDL)

CREATE	Define a database object (table, view, macro, index, trigger or stored procedure).
DROP	Remove a table, view, macro, index, trigger or stored procedure.
ALTER	Change a database object.

Data Manipulation Language (DML)

SELECT	Select data from one or more tables.
INSERT	Place a new row into a table.
UPDATE	Change data values in one or more existing
DELETE	Remove one or more rows from a table.

Data Control Language (DCL)

GRANT	Give user privileges on database objects.
REVOKE	Remove user privileges on database objects.
GIVE	Transfer database ownership.

A Simple SQL SELECT

Obtain a list of all valid department names.

Two possible methods are:

```
SELECT     Department_Name
FROM       Department;

SELECT      Department.Department_Name
FROM        Department                ;
```

Recall that qualifications for SQL are:
databasename.tablename.columnname

Note that the order of the result appears to be random.

The default column heading is the column name.

```
department_name
------------------------------
education
None
software support
technical operations
president
product planning
research and development
marketing sales
customer support
```

Projecting All Columns and All Rows

Display all columns of information for all of the departments in the Department table.

```
SELECT  *  FROM Department;
```

```
department_number  department_name            budget_amount  manager_employee_number
-----------------  -------------------------  -------------  -----------------------
              403  education                      932000.00                     1005
              600  None                                NULL                     1099
              402  software support               308000.00                     1011
              100  president                      400000.00                      801
              302  product planning               226000.00                     1016
              301  research and development       465600.00                     1019
                ?  technical operations           293800.00                     1025
              401  customer support               982300.00                     1003
              501  marketing sales                308000.00                     1017
```

Aliasing a Column Using AS

You can provide an "alias" for a projected column using the optional "AS" keyword.

An alias is the assignment of a new name.

It may be thought of as renaming the column for the life of the query.

Show all column values for all rows of the department table renaming the columns names to something shorter.

```
SELECT      department_number            AS "Dept Nbr"
           ,department_name              AS DeptName
           ,budget_amount                AS Budget
           ,manager_employee_number AS Mgr#
FROM       department;
```

As new names, aliases now become the names for the column headings.

```
Dept Nbr  DeptName                               Budget          Mgr#
--------  ------------------------------  ------------  -----------
     403  education                          932000.00         1005
     600  None                                       ?         1099
     402  software support                   308000.00         1011
     201  technical operations               293800.00         1025
     100  president                          400000.00          801
     302  product planning                   226000.00         1016
     301  research and development           465600.00         1019
     501  marketing sales                    308000.00         1017
     401  customer support                   982300.00         1003
```

Note the Heading.

Aliasing Mistake?

Based on our discussion from the previous page, can you determine what is happening with this query and its result?

Show all column values for all rows of the department table without applying aliases.

```
SELECT      department_number,
            department_name
            budget_amount,
            manager_employee_number
FROM        department;
```

```
department_number  budget_amount                  manager_employee_number
-----------------  -----------------------------  -----------------------
              403  education                                         1005
              600  None                                              1099
              402  software support                                  1011
              201  technical operations                              1025
              100  president                                          801
              302  product planning                                  1016
              301  research and development                          1019
              501  marketing sales                                   1017
              401  customer support                                  1003
```

Ordering Rows Using ORDER BY

The ORDER BY clause can be used to order result rows.

Show all column values for all rows of the department table ordered by their department name.

```
SELECT    department_number           AS Dept#
         ,department_name             AS DeptName
         ,budget_amount               AS Budget
         ,manager_employee_number     AS Mgr#
FROM      department
ORDER BY DeptName;
```

The default ORDER BY is "ascending".

You could order explicitly doing either of these:

```
ORDER BY DeptName ASC;
ORDER BY DeptName DESC;
```

```
  Dept#  DeptName                            Budget        Mgr#
-------  ------------------------------  ------------  -----------
    401  customer support                   982300.00         1003
    403  education                          932000.00         1005
    501  marketing sales                    308000.00         1017
    600  None                                       ?         1099
    100  president                          400000.00          801
    302  product planning                   226000.00         1016
    301  research and development           465600.00         1019
    402  software support                   308000.00         1011
    201  technical operations               293800.00         1025
```

Other Ordering Options

There are many different techniques that may be used for ordering result rows.

Discuss what each option shown is attempting to do and if it is valid or not.

```
SELECT  department_number, budget_amount, manager_employee_number
FROM    department
ORDER BY manager_employee_number, department_number;

ORDER BY manager_employee_number DESC, department_number;

ORDER BY 3, 1;

ORDER BY 4;

ORDER BY 3 DESC, 1;

ORDER BY 3, department_number DESC;

ORDER BY department_name;
```

What about these two?

```
SELECT * FROM department ORDER BY 2;
SELECT * FROM department ORDER BY 10;
```

Projecting Literal Values

With SQL you can project literal values as well as column values.

Character literals are enclosed inside single quotes while numeric data is not.

```
SELECT     'Department number'    AS  Character_Literal,
           12345                  AS Numeric_Literal
FROM       Department
ORDER BY 1;
```

What effect did ordering by a literal have on the result?

Why are there 9 rows returned?

```
Character_Literal  Numeric_Literal
-----------------  ---------------
Department number            12345
Department number            12345
Department number            12345
Department number            12345
Department number            12345
Department number            12345
Department number            12345
Department number            12345
Department number            12345
```

Using WHERE to Eliminate Rows

Show name for department 401.

```
SELECT 'Department Number'          AS Literal1,
       Department_Number            AS D#,
       'Has the name of'            AS Literal2,
       Department_Name              AS DName
FROM Department
WHERE Department_Number = 401
ORDER BY 1;
```

```
Literal1                          D# Literal2        DName
----------------- ----------- --------------- -----------------------------
Department Number             401 Has the name of customer support
```

Show number for the customer support department.

```
SELECT 'The'                        AS Literal1,
       Department_Name              AS DName,
       'department is numbered'     AS Literal2,
       Department_Number            AS D#
FROM Department
WHERE Department_Name =  'customer support'
ORDER BY 1;
```

```
Literal1 DName                          Literal2                        D#
-------- ------------------------------ ---------------------- -----------
The      customer support               department is numbered         401
```

Basic Logical Operators

The chart shows options for WHERE constraints that require equality or inequality constraints.

These will discussed in more depth in the next module.

	ANSI Standard
Equal	=
Not Equal	<>
Less Than	<
Greater Than	>
Greater Than Equal To	>=
Less Than Equal To	<=

DISTINCT Option

Find all the different job codes assigned to employees.

Without DISTINCT option:

```
SELECT      department_number       AS D#
            ,job_code               AS J_Cd
FROM        employee
ORDER BY 1, 2;
```

```
         D#         J_Cd
-----------  -----------
          ?       211100
          ?       222101
        100       111100
        301       311100
        301       312101
        301       312102
        302       321100
        401       411100
        401       412101
        401       412101
        401       412101
        401       412102
        401       412102
        401       413201
        402       421100
        402       422101
        403       431100
        403       432101
        403       432101
        403       432101
        403       432101
        403       432101
        501       511100
        501       512101
        501       512101
        501       512101
```

With DISTINCT option:

```
SELECT  DISTINCT
        department_number  AS D#
       ,job_code           AS J_Cd
FROM    employee;
```

```
         D#         J_Cd
-----------  -----------
          ?       211100
          ?       222101
        100       111100
        301       311100
        301       312101
        301       312102
        302       321100
        401       411100
        401       412101
        401       412102
        401       413201
        402       421100
        402       422101
        403       431100
        403       432101
        501       511100
        501       512101
```

The following sort is performed by DISTINCT.

```
        ORDER BY 1, 2;
```

DISTINCT MUST APPEAR FIRST IN THE SELECT LIST!

Recommended Coding Conventions

Although SQL is considered a “free-form” language, the following represents a commonly used convention for SQL coding.

```
SELECT           last_name
                ,first_name
                ,hire_date
                ,salary_amount
FROM            employee
WHERE           department_number = 401
ORDER BY        last_name;
```

The convention below, often referred to as “paragraph-style”, can be difficult to debug. Identify two potential problems with the following query.

```
select last_name,first_name,hire_date salary_amount from
employee wheredepartment_number = 401 order by last_name;
```

Module 1: Summary

- SQL has 3 classes of queries.
  - ✓ DDL
  - ✓ DML
  - ✓ DCL
- The number of rows returned can be affected by condition applied via a WHERE clause.
- You can rearrange the order of the rows in the result set by using ORDER BY;
- You can alias a column name using AS.
- Operators like =, <>, <=, >=, <, > can be used as qualifiers.
- DISTINCT can be used to project a distinct list of result rows.
- You can project literal values as well as column values.
- Get into good habits of writing SQL early and avoid writing in paragraph form.

Module 1: Review Questions

True or False:

1. **“SELECT * FROM Employee ORDER BY 1;” is a valid SQL construct.**
2. **The SQL DELETE is considered a DDL request.**
3. **DISTINCT automatically performs a sort.**
4. **A WHERE clause can be used to eliminate columns from a result.**
5. **A character literal not enclosed in single quotes is interpreted as an object name.**
6. **Double quotes can also be used to display literal values.**

Module 1: Lab Exercise

1) Select all columns for all departments from the department table.

2) Request a report of employee last and first names and salary for all of manager 1019's employees. Order the report in last name ascending sequence.

3) What are the first names of people with a last name of “Brown”?

4) How many people have been assigned job codes greater than or equal to 510001?