SQL Course

The Simple Select Statement

- Demonstrate the use of the simple **SELECT** statement.
- Highlight the capabilities of the **SELECT** and **FROM** clauses.
- Illustrate the simple **SELECT** statement with a few examples.

In many ways, queries are at the heart of the SQL language. The **SELECT** statement, which is used to express SQL queries, is the most powerful and complex of the SQL statements. Despite the many options afforded by the **SELECT** statement, it's possible to start simply and work up to more complex queries.
This module discusses the syntax of the **SELECT** statement and its capabilities before entering into more detail in later modules.
The simple select statement consists solely of using the two main clauses:
SELECT and **FROM**.
In the simple case covered in this module we will only ever be querying from one table. It is possible to select from more than one table and this will be covered in more detail in later modules.

The **SELECT** statement in its simplest form is as follows :

SELECT [**ALL**|**DISTINCT**] *{select_list | *}* **FROM** *table_name*;

The **SELECT** clause specifies the data items to be retrieved by the query. These items are usually specified by a select list. This is a list of select items separated by a comma. Each item in the list generates a single column of query results, in left to right order. Each item can be one of the following :
Column name
This must identify a column from the table in the **FROM** clause. In such situations SQL simply takes the value of that column from each row of the database table and places it in the corresponding row of query results.
Constant
This means the same constant value will appear in every row of the query results.
SQL expression
Indicating that SQL must calculate the value to be placed into the query results, in the style specified by the expression.

When selecting a column name we can use its full (qualified) name which has the form :

table_name.column_name

Columns can be renamed for display purposes, for example :

SELECT *customer_name "Customer Name"* .....

This will print the customer names as before but this time the column heading will be *Customer Name* instead of *customer_name*. As in this case, double quotes are only needed when the new column name contains a space or other special characters.

The **FROM** clause consists of the keyword **FROM**, followed by a list of table specifications separated by commas. Each table specification identifies a table containing data to be

retrieved by the query. This module will only cover examples where only one table is ever specified in the **FROM** clause. Multi table queries are covered in later modules.
Sometimes it's convenient to display all the contents of all the columns of a table. This can be particularly useful when you encounter a new database and you want to get familiar with its structure and data. As a convenience, SQL lets you use an asterix (*) as an abbreviation for "all columns". The ANSI/ISO standard specifies that a **SELECT** statement can either have an all column selection or a column list, but not both. However many implementation treat the * as just another element of the select list.
If a query result contains the primary key of a table in its column list then every row of the query will be unique. If the primary key is not included in the query results, duplicate rows can occur. You can eliminate duplicate rows of query results by inserting the keyword **DISTINCT** in the **SELECT** statement just before the select list. You may also specify the keyword **ALL** to indicate that duplicate rows are to be retained.

The simplest SQL queries request columns of data from a single table in the database.
The first example shows how the SQL **SELECT** is used to implement the relational algebra **PROJECT** operator, and will retrieve the name of all aircrafts and the number of club seats on each plane :

SELECT *aircraft_name, no_club_seats* **FROM** *aircraft*;

aircraft name	*model*	*club seats*	*econ seats*	*call sign*
Eagle Flyer	ATR42	22	40	N410C
(NULL)	Boeing 707-320C	50	102	9J-AEB
(NULL)	Boeing 727-200	34	100	N7255U
Finians Dream	Boeing 737-200	22	96	DQ-FDM
(NULL)	Boeing 737-200	8	120	N301SW

Table returned from the Projection.

aircraft name	*club seats*
Eagle Flyer	22
(NULL)	50
(NULL)	34
Finians Dream	22
(NULL)	8

The following is an example of a **PROJECT**ion of columns from the customer table :

SELECT *customer_name, telephone* **FROM** *customer*;

The following **SELECT** just displays all the rows and columns in one table :

SELECT * **FROM** *airport*;

Each row in a table is unique, but by **SELECT**ing only some of the columns, duplicate rows can be produced. The keyword **DISTINCT** is used to qualify the column name so that no duplicate values will be displayed.
The following example returns the list of customers (by customer number) who have made bookings. Duplicate customer numbers are eliminated.

SELECT DISTINCT *customer_number* **FROM** *booking*;

Booking Reference	Booking Date	Customer Number	Flight Number	Daparture Date	Seat Class
b222	03/04/99	c422	EI082	01/08/99	Econ
b555	21/02/99	c102	BD303	11/06/99	Econ
b675	13/06/99	c102	EI124	08/07/99	First Class
b342	01/03/99	c340	SK537	21/05/99	Econ
b212	29/04/99	c280	EI989	12/06/99	First Class

Table returned from Select statement using Distinct.

Customer Number
c422
c102
c340
c280