

Lesson 1

Retrieving Data Using the SQL SELECT Statement

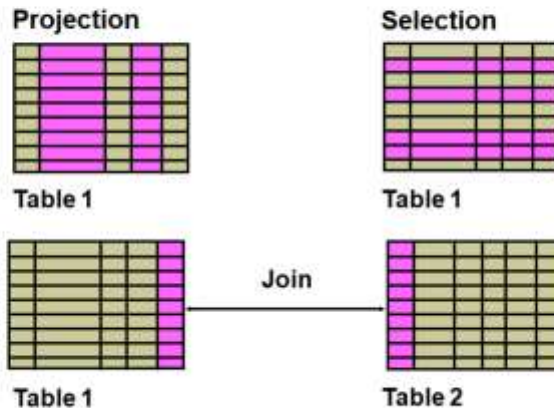
What You will Learn at the end of this Session ?



**1 List the capabilities of SQL
SELECT statements**

**2 Execute a basic SELECT
statement**

Capabilities of SQL SELECT Statements



1-3

Capabilities of SQL SELECT Statements

A `SELECT` statement retrieves information from the database. With a `SELECT` statement, you can do the following:

Projection: Select the columns in a table that are returned by a query. Select as few or as many of the columns as required.

Selection: Select the rows in a table that are returned by a query. Various criteria can be used to restrict the rows that are retrieved.

Joins: Bring together data that is stored in different tables by specifying the link between them. SQL joins are covered in more detail in the lesson titled "Displaying Data from Multiple Tables Using Joins."

Basic SELECT Statement

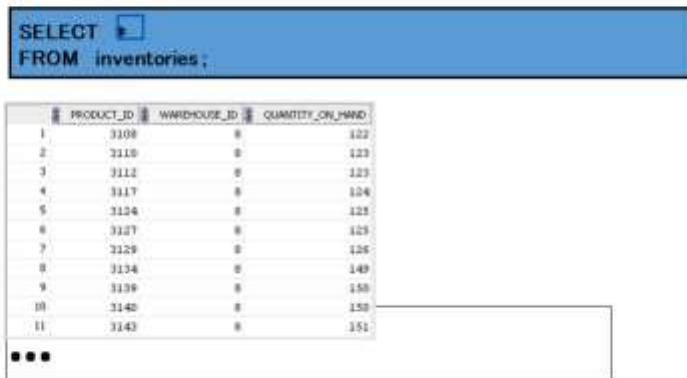
```
SELECT *|([DISTINCT] column|expression [alias],...)  
FROM    table;
```



SELECT identifies the columns to be displayed.

FROM identifies the table containing those columns.

Selecting All Columns



The screenshot shows a SQL Developer interface. At the top, a blue bar contains the SQL statement: `SELECT *
FROM inventories;`. Below this, a table displays the results of the query. The table has four columns: `PRODUCT_ID`, `WAREHOUSE_ID`, and `QUANTITY_ON_HAND`. The first 11 rows of data are visible, showing product IDs from 1 to 11, all with a warehouse ID of 0 and varying quantities. A scrollbar is visible on the right side of the table, and three dots at the bottom left indicate that more rows exist.

	PRODUCT_ID	WAREHOUSE_ID	QUANTITY_ON_HAND
1	3109	0	122
2	3110	0	123
3	3111	0	123
4	3117	0	124
5	3124	0	125
6	3127	0	125
7	3129	0	126
8	3134	0	149
9	3139	0	150
10	3140	0	150
11	3143	0	151

1-5

Selecting All Columns

You can display all columns of data in a table by following the `SELECT` keyword with an asterisk (*). In the example in the slide, the `INVENTORIES` table contains four columns: `PRODUCT_ID`, `WAREHOUSE_ID`, `QUANTITY_ON_HAND`.

You can also display all columns in the table by listing all the columns after the `SELECT` keyword. For example, the following SQL statement (like the example in the slide) displays all columns and all rows of the `INVENTORIES` table:

```
SELECT product_id, warehouse_id, quantity_on_hand  
FROM inventories;
```

Note: In SQL Developer, you can enter your SQL statement in a SQL Worksheet and click the “Execute Statement” icon or press [F9] to execute the statement. The output displayed on the Results tabbed page appears as shown in the slide.

Selecting Specific Columns

```
SELECT product_id, quantity_on_hand  
FROM inventories;
```

	PRODUCT_ID	QUANTITY_ON_HAND
1	3109	122
2	3110	123
3	3112	123
4	3117	124
5	3124	125
6	3127	125
7	3129	126
8	3134	149
9	3139	150
10	3140	150
11	3143	151

...

1-6

Selecting Specific Columns

You can use the `SELECT` statement to display specific columns of the table by specifying the column names, separated by commas. The example in the slide displays all the department numbers and location numbers from the `INVENTORIES` table.

In the `SELECT` clause, specify the columns that you want in the order in which you want them to appear in the output. For example, to display location before department number (from left to right), you use the following statement:

```
SELECT product_id, quantity_on_hand  
FROM inventories;
```

	LOCATION_ID	DEPARTMENT_ID
1	1700	10
2	1800	20
3	1500	50
4	1400	60

...

Writing SQL Statements

SQL statements are not case sensitive

SQL statements can be entered on one or more lines.

Keywords cannot be abbreviated or split across lines.



In SQL Developer, SQL statements can be optionally terminated by a semicolon (;). Semicolons are required when you execute multiple SQL statements

Clauses are usually placed on separate lines.

Indents are used to enhance readability.

In SQL*Plus, you are required to end each SQL statement with a semicolon (;).

Column Heading Defaults

A light blue 3D rectangular button with the text "SQL Developer" in black.

SQL Developer

Default heading alignment: Left-aligned
Default heading display: Uppercase

A light blue 3D rectangular button with the text "SQL*Plus" in black.

SQL*Plus

Character and Date column headings are left-aligned.
Number column headings are right-aligned.
Default heading display: Uppercase

Column Heading Defaults

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	100	Steven	King	SKING	\$15.123.4567
2	101	Neena	Kochhar	NKOCHHAR	\$15.123.4568
3	102	Lex	De Haan	LDEHAAN	\$15.123.4569
4	103	Alexander	Hunold	AHUNOLD	\$90.423.4567
5	104	Bruce	Ernst	BERNST	\$90.423.4568
6	105	David	Austin	DAUSTIN	\$90.423.4569
7	106	Valli	Pataballa	VPATABAL	\$90.423.4560
8	107	Diana	Lorentz	DLORENTZ	\$90.423.5567
9	108	Nancy	Greenberg	NGREENBE	\$15.124.4569
10	109	Daniel	Faviet	DFAVIET	\$15.124.4169

In SQL Developer, column headings are displayed in uppercase and are left-aligned.