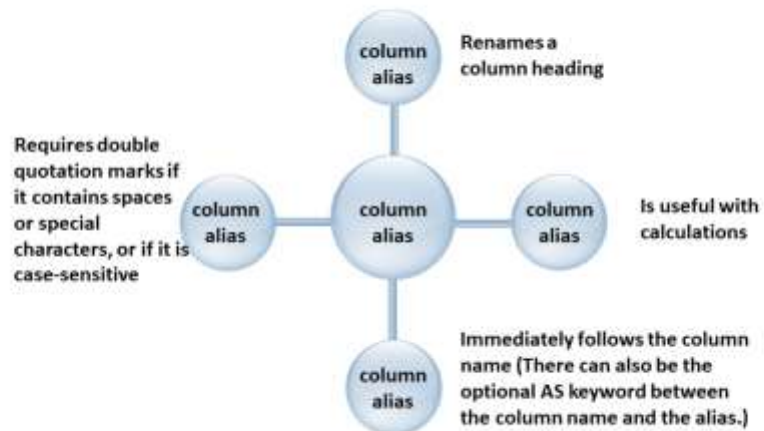


Defining a Column Alias



Using Column Aliases

```
SELECT product_id AS Product, quantity_on_hand Quantity  
FROM inventories;
```

	PRODUCT	QUANTITY
1	3108	122
2	3110	123
3	3112	125
4	3117	124

■ ■ ■

```
SELECT order_id "Order", ROUND(order_date) "Date of Order"  
FROM orders;
```

	Order	Date of Order
1	2458 17-AUG-99	
2	2397 20-NOV-99	
3	2454 03-OCT-99	
4	2354 15-JUL-00	

■ ■ ■

1-2

Using Column Aliases

The first example displays the products and the quantities of all the products. Note that the optional `AS` keyword has been used before the column alias name. The result of the query is the same whether the `AS` keyword is used or not. Also, note that the SQL statement has the column aliases, `product` and `quantity`, in lowercase, whereas the result of the query displays the column headings in uppercase. As mentioned in the preceding slide, column headings appear in uppercase by default.

The second example displays the products and the quantities of all the products. Because `Date of Order` contains spaces, it has been enclosed in double quotation marks. Note that the column heading in the output is exactly the same as the column alias.

Concatenation Operator

•A concatenation operator:

- Links columns or character strings to other columns
- Is represented by two vertical bars (||)
- Creates a resultant column that is a character expression

```
SELECT first_name || last_name AS "NAME"  
FROM customers;
```

	NAME
1	KellenAbel
2	Sunderland
3	MosheAtkinson
4	DavidAustin
5	HermannBaer
6	ShelliBaida
7	AmitBanda
8	ElizabethBates

13

Concatenation Operator

You can link columns to other columns, arithmetic expressions, or constant values to create a character expression by using the concatenation operator (||). Columns on either side of the operator are combined to make a single output column.

In the example, `ORDER_ID` and `ORDER_MODE` are concatenated, and given the alias `ORDERS`. Note that the ID of the order and mode of the order are combined to make a single output column.

The `AS` keyword before the alias name makes the `SELECT` clause easier to read.

Null Values with the Concatenation Operator

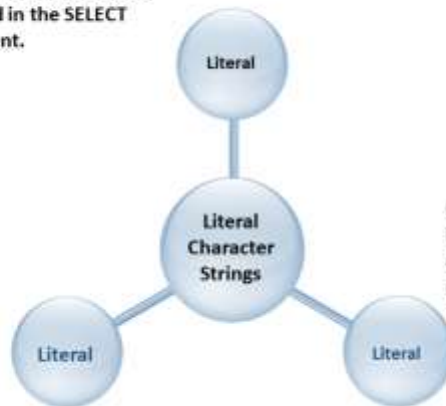
If you concatenate a null value with a character string, the result is a character string. `ORDER_ID || NULL` results in `ORDER_ID`.

Note: You can also concatenate date expressions with other expressions or columns.

Literal Character Strings

A literal is a character, a number, or a date that is included in the SELECT statement.

Each character string is output once for each row returned



Date and character literal values must be enclosed within single quotation marks.

Using Literal Character Strings

```
SELECT product_id || ' is in Warehouse ' || warehouse_id  
       AS "Product-Warehouse"  
FROM   inventories;
```

	Product-Warehouse
1	1733 is in Warehouse 1
2	1734 is in Warehouse 1
3	1737 is in Warehouse 1
4	1738 is in Warehouse 1
5	1745 is in Warehouse 1
6	1748 is in Warehouse 1
7	2278 is in Warehouse 1

...

1-5

Using Literal Character Strings

The example in the slide displays the product id and Warehouse numbers of all products. The column has the heading Product-Warehouse. Note the spaces between the single quotation marks in the `SELECT` statement. The spaces improve the readability of the output.

In the following example, the last name and salary for each employee are concatenated with a literal, to give the returned rows more meaning:

```
SELECT last_name || ': 1 Month salary = ' || salary Monthly  
FROM   employees;
```

	MONTHLY
1	Whalen: 1 Month salary = 4400
2	Hartstein: 1 Month salary = 13000
3	Fay: 1 Month salary = 6000
4	Higgins: 1 Month salary = 12000
5	Gietz: 1 Month salary = 8300
6	King: 1 Month salary = 24000
7	Kochhar: 1 Month salary = 17000
8	De Haan: 1 Month salary = 17000

Alternative Quote (q) Operator

- Specify your own quotation mark delimiter.
- Select any delimiter.
- Increase readability and usability.

```
SELECT department_name || q '[ Department's Manager Id: ]'
      || manager.id
      AS "Department and Manager"
FROM departments;
```

	Department and Manager
1	Administration Department's Manager Id: 200
2	Marketing Department's Manager Id: 201
3	Shipping Department's Manager Id: 124
4	IT Department's Manager Id: 103
5	Sales Department's Manager Id: 149
6	Executive Department's Manager Id: 100
7	Accounting Department's Manager Id: 205
8	Contracting Department's Manager Id:

1-8

Alternative Quote (q) Operator

Many SQL statements use character literals in expressions or conditions. If the literal itself contains a single quotation mark, you can use the quote (q) operator and select your own quotation mark delimiter.

You can choose any convenient delimiter, single-byte or multibyte, or any of the following character pairs: [], { }, (), or < >.

In the example shown, the string contains a single quotation mark, which is normally interpreted as a delimiter of a character string. By using the q operator, however, brackets [] are used as the quotation mark delimiters. The string between the brackets delimiters is interpreted as a literal character string.