# Database Design and SQL

Single Table Queries

CSAM - 2021S

Sagara Samarawickrama

Section 1 & 2

# Chapter Objectives

- Use SQL commands to retrieve data from a database table
- Use literals in the selected list
- Use computed columns in the selected list.
- Use the AS clause
- Use Simple and Compound conditions in WHERE clauses
- Use the DISTINCT keyword
- Use the CONCAT, BETWEEN, LIKE and IN operations
- Sort the results of a SELECT statement using the ORDER BY clause

### **Basic Format of the SELECT Statement:**

The SQL SELECT statement is used to query a database and return a result set containing rows from one or more tables or views.

SELECT column\_names
FROM table or view\_name
WHERE search\_condition
GROUP BY column\_names
HAVING search\_condition
ORDER BY column\_name;

The SELECT and FROM keywords must be specified; the other keywords are optional.

### The parts of a SELECT statement serve the following purposes:

Portion of select statement	Description
SELECT column-names	Specifies the columns in the SELECT statement's result set
FROM table-list	Specifies tables and/or views from which the result set data is returned
WHERE search-condition	Specifies a logical condition that must be true for a row to be included in the result set
GROUP BY grouping-column-list	Specifies the column(s) whose values are used to group the rows
HAVING search-condition	Specifies a logical condition that must be true for a group to be included in the result set
ORDER BY order-by-column-list	Specifies a list of columns with ascending or descending (with the DESC keyword) sequence

Example 6-1: Return all row and column from DEPARTMENTS table.			
SELECT * FROM departme	ents;		
Results			
DEPARTMENT_CODE	DEPARTMENT_NAME	MANAGER_ID	
AD	Administration	NULL	Light of the Control
AC	Accounting	NULL	.02 7108
MK	Marketing	NULL	Alpa Steel Int.

```
Example 6-2: Return customer_name for all rows in the CUSTOMERS table.

SELECT customer_name
FROM customers;

Results

CUSTOMER_NAME

Smith Mfg.
Bolt Co.
Ajax Steel Inc.
```

### Select Multiple Columns From a TABLE

Example 6-3: Return customer_id, customer_name, and discount for all rows in the CUSTOMERS table.			
SELECT customer_id, customer_name, FROM customers;	discount		
Results			
CUSTOMER_ID CUSTOMER_NAME	DISCOUNT		
133568 Smith Mfg. 246900 Bolt Co. 275978 Ajax Steel Inc.	0.050 0.020 NULL		
499320 Bluewater Inc. 499921 Bell Bldg. 518980 London Inc.	0.015 0.010 0.050	on Colorea co uccess	SATURATE AND THE SATURA

### **Computed Columns**

A computed column is a column in the result set that does not exist in the table. Instead a computed column is calculated using a data from existing columns in the table.

Arithmetic operator	Description
* Anti-consular Third	Addition
a rath tartus	Subtraction
*	Multiplication
1	Division

```
Example 6-4: Return order_id, customer_id, order_total, and discount for all rows
 in the ORDERS table. Discount is a computed value calculated as order_total *
SELECT order_id, customer_id, order_total,
      (order_total * .05) AS discount
  FROM orders;
Results
ORDER_ID CUSTOMER_ID ORDER_TOTAL DISCOUNT
 234112
            499320
                        35.00 1.7500
 234113
            888402
                       278.75 13.9375
 234114
            499320
                       78.90 3.9450
 234115
            290000
SELECT customer_id, customer_name,
        (credit_limit - balance) AS "Available Credit"
  FROM customers;
Results
CUSTOMER_ID CUSTOMER_NAME
                                                Available Credit
     133568 Smith Mfg.
                                                        148002.00
   246900 Bolt Co.
                                                        224896.45
     275978 Ajax Steel Inc.
```

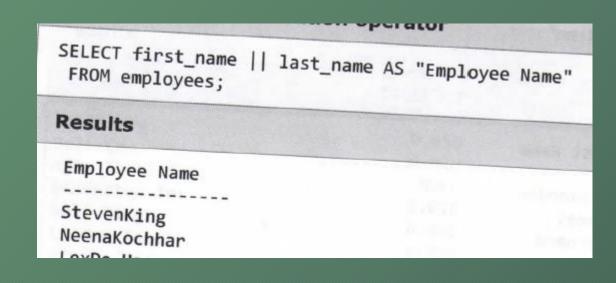
### **Column Aliases**

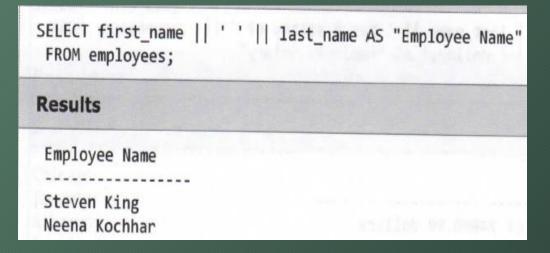
An alias is a way to rename column heading in the output. Aliases can be used to better describe the contents of a column in the result set.

```
SELECT first name
                    AS "First Name",
      middle initial
                        Initial.
      last name
                       "Last Name"
  FROM employees:
Results
First Name
                   INITIAL Last Name
Lauren
                          Alexander
Lisa
                          James
Dave
                          Bernard
Steve
                          Carr
Marg
```

### **Concatenation Operator**

Concatenation means to connect two items together.





### **DISTINCT** keyword

When the SELECT statement does not include the primary key columns, the result set may contain duplicate rows. To eliminate duplicate rows from the result set, the SELECT keyword is followed with the DISTINCT keyword.

Example 6-12: Return ship\_city for all rows in the CUSTOMERS table. Use the DISTINCT operator to list each city only once.

SELECT DISTINCT ship\_city FROM customers;

Results

SHIP\_CITY

Chicago Toronto Albany Portland

### Using the WHERE Clause to limit row selection

Sometimes, a request is made to retrieve only one row or set of rows that meet a search condition

```
SELECT customer_id, customer_name
FROM customers
WHERE ship_city = 'Boston';

Results

CUSTOMER_ID CUSTOMER_NAME

518980 London Inc.
663456 Alpine Inc.
```

The Where condition can return zero, one or more rows. The Where clause specifies which columns or rows will be returned, based on the criteria described after the WHERE keyword. A simple condition compares two values, using one of the comparison operators listed below

Comparison operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
<>	Not equal

Example 6-16: Test for a NULL cond	lition
	and discount for all rows in the CUSTOMERS table
SELECT customer_id, customer_name, FROM customers WHERE discount IS NULL;	
Results	
CUSTOMER_ID CUSTOMER_NAME	DISCOUNT
275978 Ajax Steel Inc. 78 <b>1010</b> Bluewater Mfg.	NULL NULL

Example 6-17: Test for a NOT NULL cond	dition
	discount for all rows in the CUSTOMERS table
SELECT customer_id, customer_name, disc FROM customers	
Pocula	
CUSTOMER_ID CUSTOMER_NAME  133568 Smith Mfg. 246900 Bolt Co. 499320 Bluewater Inc.	DISCOUNT  0.050 0.020

### Using a Computed Value with a Where Clause

A computed value can be used in WHERE clause.

### **Using Compound Conditions**

The WHERE clause also may contain a compound condition that specifies two or more conditions with AND or OR connectors.

```
Return customer_name, ship_city, and discount for all rows in the CUSTOMERS table where ship_city = Albany AND discount > 0.

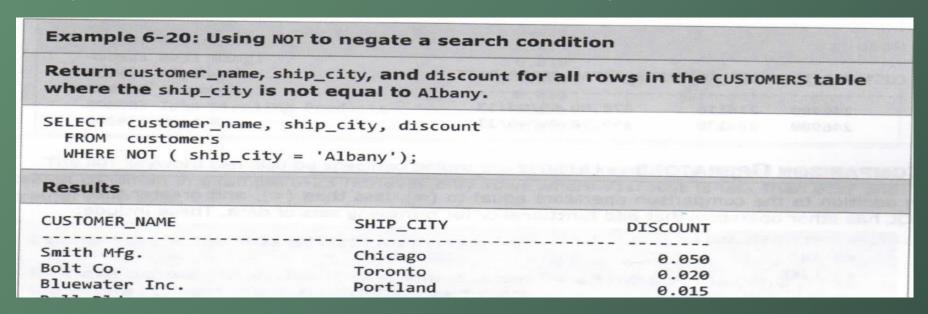
SELECT customer_name, ship_city, discount
FROM customers
WHERE ship_city = 'Albany'
AND discount > 0;

Results

CUSTOMER_NAME SHIP_CITY DISCOUNT
Seaworthy Albany 0.010
```

### **Negating a Conditions**

To negate a condition, the NOT logical operator can be specified at the beginning of any condition or before conditions connected by AND or OR.



### **Comparison Operators**

In addition to the comparison operators equal to (=), less than (<), and greater than (>), SQL has other operators that add functionality for retrieving sets of data. These include:

- BETWEEN ... AND
- IN
- LIKE

### **Comparison Operators**

In addition to the comparison operators equal to (=), less than (<), and greater than (>), SQL has other operators that add functionality for retrieving sets of data. These include:

- BETWEEN ... AND
- IN
- LIKE

### **BETWEEN ...AND Operators**

### **Example 6-22 The BETWEEN operator**

Retrieve customer\_id, customer\_name, and discount for all rows in the CUSTOMERS table where discount is between 0.01 and 0.02.

SELECT customer\_id, customer\_name, discount FROM customers
WHERE discount BETWEEN 0.01 AND 0.02;

### Results

CUSTOMER_ID	CUSTOMER_NAME	DISCOUNT
246900	Bolt Co.	0.020
499320	Bluewater Inc.	0.015

### **NOT BETWEEN Operators**

518980 London Inc.

# Retrieve customer\_id, customer\_name, and discount for all rows in the CUSTOMERS table where discount is NOT BETWEEN 0.01 and 0.02. SELECT customer\_id, customer\_name, discount FROM customers WHERE discount NOT BETWEEN 0.01 AND 0.02; Results CUSTOMER\_ID CUSTOMER\_NAME DISCOUNT 133568 Smith Mfg. 0.050

0.050

### **IN Operators**

```
Example 6-24: Using the IN operator with a WHERE clause
Retrieve customer_id, customer_name, and ship_city for all rows in the CUSTOMERS
table where ship_city is either Detroit, Portland, or Boston.
       customer_id, customer_name, ship_city
  FROM customers
  WHERE ship_city IN ( 'Detroit', 'Portland', 'Boston' );
Results
CUSTOMER_ID CUSTOMER_NAME
                                         SHIP_CITY
    499320 Bluewater Inc.
                                     Portland
    499921 Bell Bldg.
                                         Detroit
```

### **LIKE Operator**

The percent sign(%) specifies a match to a string of any length Underscore(\_) specifies a match on a single character

```
SELECT employee_id, first_name
FROM employees
WHERE first_name LIKE 'R%';

Results

EMPLOYEE_ID FIRST_NAME

116 Robert
139 Rick
```

```
SELECT employee_id, first_name
FROM employees
WHERE first_name LIKE '_ick%';

Results

EMPLOYEE_ID FIRST_NAME

139 Rick
```

### **Logical Operator**

A logical operator combines the result of two or more conditions to produce a single result.

- \* AND Operator: Returns TRUE if both conditions are true
- ❖ OR Operator: Returns TRUE if either condition is true
- ❖ NOT Operator: Returns TRUE if the condition is false

### **AND Operator**

```
SELECT first_name, last_name, department_code, salary
FROM employees
WHERE department_code = 'IT' AND salary > 23500;

Results

FIRST_NAME LAST_NAME DEPARTMENT_CODE SALARY

Greg Zimmerman IT 31500.00
Dave Bernard IT 24000.00
Rick Peters IT 28750.00
```

### **OR Operator**

```
SELECT first_name, last_name, department_code, salary
 FROM employees
 WHERE department_code = 'IT' OR salary > 23500;
Results
FIRST_NAME LAST_NAME DEPARTMENT_CODE SALARY
Lauren
          Alexander TR
                                  45000.00
Lisa
          James
                                  65000.00
Dave
          Bernard
                                  60000.00
Steve
          Carr
                   VG
                                  55000.00
Marg
          Horner
                                  45000.00
```

### **NOT Operator**

```
SELECT first_name, last_name, department_code, salary
FROM employees
WHERE department_code NOT IN ('MA', 'VG', 'HT');

Results

FIRST_NAME LAST_NAME DEPARTMENT_CODE SALARY

Lauren Alexander TR 45000.00
Jim Best SA 24000.00
Greg Zimmerman IT 31500.00
```

# Conclusion

