Lab – 04

# Restricting and Sorting data and Type Casting Functions

**Name: Saad Ahmad**

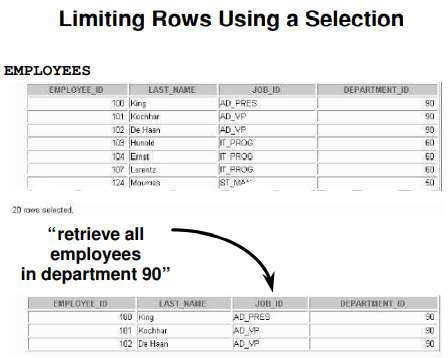
**Enrollment: 01-134222-130**

## Objective:

* Students will be able to limit the row retrieved by the query and sort the rows retrieved by a query.

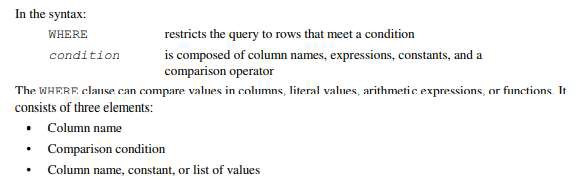
# Limiting Rows Using a Selection

In the example in the slide, assume that you want to display all the employees in department 90. The rows with a value of 90 in the DEPARTMENT\_ID column are the only ones returned. This method of restriction is the basis of the WHERE clause in SQL



# Limiting the Rows Selected

You can restrict the rows returned from the query by using the WHERE clause. A WHERE clause contains a condition that must be met, and it directly follows the FROM clause. If the condition is true, the row meeting the condition is returned.

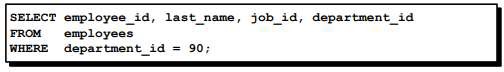




# Using the WHERE Clause

In the example, the SELECT statement retrieves the name, job ID, and department number of all employees whose job ID is SA\_REP.

Note that the job title SA\_REP has been specified in uppercase to ensure that it matches the job ID column in the EMPLOYEES table. Character strings are case sensitive

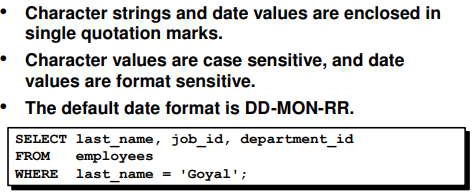


# Character Strings and Dates

Character strings and dates in the WHERE clause must be enclosed in single quotation marks (''). Number constants, however, should not be enclosed in single quotation marks.

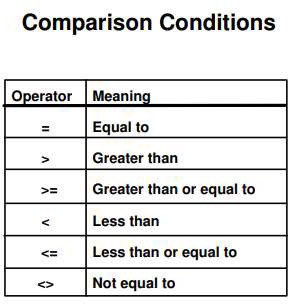
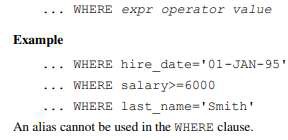
All character searches are case sensitive. In the following example, no rows are returned because the EMPLOYEES table stores all the last names in the proper case:

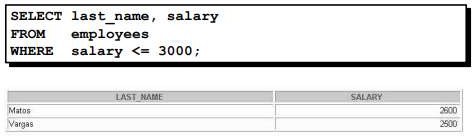


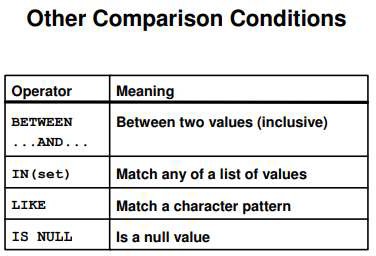


# Comparison Condition

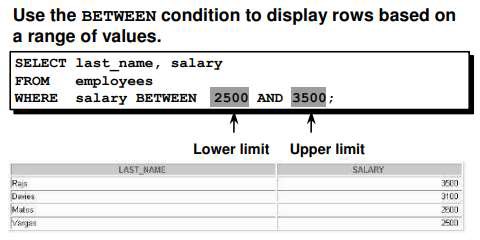
Comparison conditions are used in conditions that compare one expression to another value or expression. They are used in the WHERE clause in the following format:





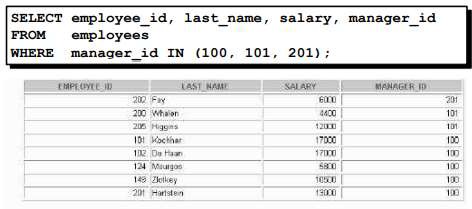


# The BETWEEN Condition

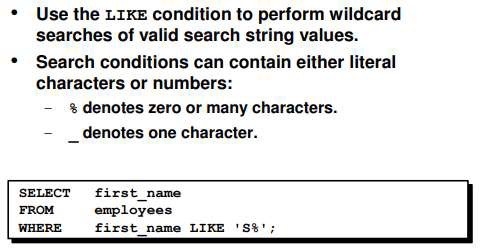


**The IN Condition**

To test for values in a specified set of values, use the IN condition. The IN condition is also known as the membership condition.



# The LIKE Condition

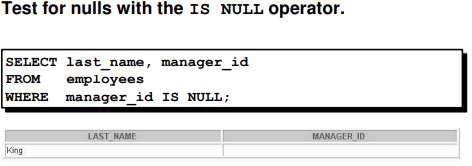


## The NULL Conditions

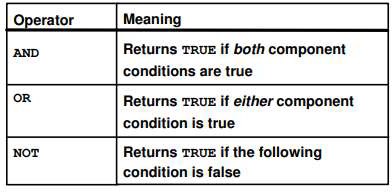
The NULL conditions include the IS NULL condition and the IS NOT NULL condition.

The IS NULL condition tests for nulls. A null value means the value is unavailable, unassigned, unknown, or inapplicable. Therefore, you cannot test with = because a null cannot be equal or unequal to any value. The slide example retrieves the last names and managers of all employees who do not have a manager.

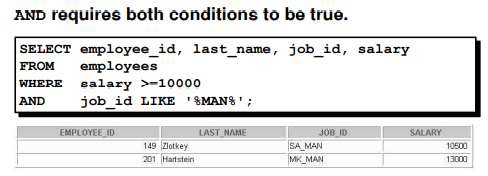
For another example, to display last name, job ID, and commission for all employees who are not entitled to get a commission, use the following SQL statement:



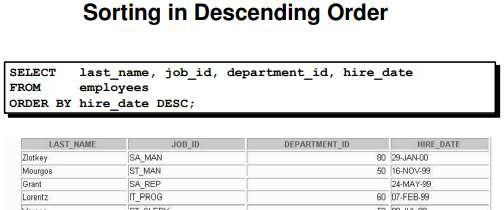
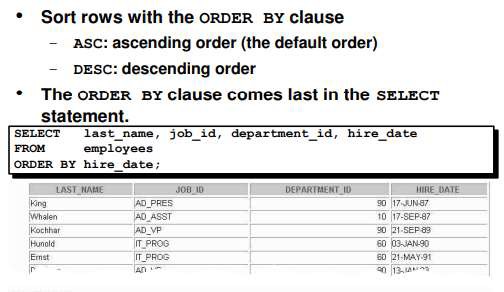
**Logical Conditions**



**The AND Operator**



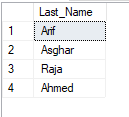
**The ORDER BY Clause**



## Lab Tasks:

1. Create a query to display the last name and salary of employees earning more than $12,000. Place your SQL statement in a text file named lab2\_1.sql. Run your query.

SELECT Last\_Name FROM Employee WHERE Salary > 12000;



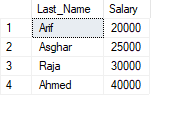
1. Create a query to display the employee last name and department number for employee number 176

SELECT Last\_Name,Dept\_ID FROM Employee WHERE Employee\_ID = 176;



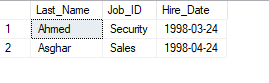
1. Modify lab2\_1.sql to display the last name and salary for all employees whose salary is not in the range of $5,000 and $12,000. Place your SQL statement in a text file named lab2\_3.sql.

SELECT Last\_Name,Salary FROM Employee WHERE NOT Salary BETWEEN 5000 AND 12000;



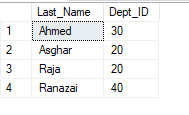
1. Display the employee last name, job ID, and start date of employees hired between February 20, 1998, and May 1, 1998. Order the query in ascending order by start date.

SELECT Last\_Name,Job\_ID,Hire\_Date FROM Employee WHERE Hire\_Date BETWEEN '1998-02-20' AND '1998-05-01' ORDER BY Hire\_Date;



1. Display the last name and department number of all employees in departments 20 and 50 in alphabetical order by name.

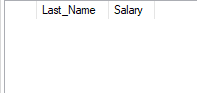
SELECT Last\_Name,Dept\_ID FROM Employee WHERE Dept\_ID BETWEEN 20 AND 50 ORDER BY Last\_Name;



1. Modify lab2\_3.sql to list the last name and salary of employees who earn between $5,000 and

$12,000, and are in department 20 or 50. Label the columns Employee and Monthly Salary, respectively. Resave lab2\_3.sql as lab2\_6.sql. Run the statement in lab2\_6.sql

SELECT Last\_Name,Salary FROM Employee WHERE Salary BETWEEN 5000 AND 12000 AND Dept\_ID =20 OR Dept\_ID =50;



1. Display the last name and hire date of every employee who was hired in 1994.

SELECT Last\_Name,Hire\_Date FROM Employee WHERE Hire\_Date BETWEEN '1994-01-01' AND '1994-12-31';



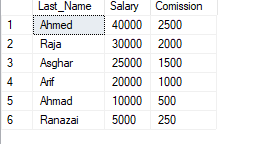
1. Display the last name and job title of all employees who do not have a manager.

SELECT Last\_Name,Job\_ID FROM Employee WHERE Manager\_ID IS NULL;



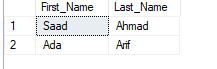
1. Display the last name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.

SELECT Last\_Name,Salary,Comission FROM Employee WHERE Comission IS NOT NULL ORDER BY Salary DESC,Comission DESC;



1. Display the last names of all employees where the third letter of the name is an a.

SELECT First\_Name,Last\_Name FROM Employee WHERE First\_Name LIKE '\_\_a%';



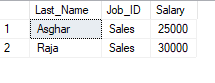
1. Display the last name of all employees who have an a and an e in their last name,

SELECT Last\_Name FROM Employee WHERE Last\_Name LIKE '%a%e%';



1. Display the last name, job, and salary for all employees whose job is sales representative or stock clerk and whose salary is not equal to $2,500, $3,500, or $7,000.

SELECT Last\_Name,Job\_ID,Salary FROM Employee WHERE Job\_ID = 'Sales' OR Job\_ID ='Stock' AND Not Salary IN(2500,3500,7000);



1. Modify lab2\_6.sql to display the last name, salary, and commission for all employees whose commission amount is 20%. Resave lab2\_6.sql as lab2\_13.sql. Rerun the statement in lab2\_13.sql.

SELECT Last\_Name,Salary,Comission FROM Employee WHERE Comission = (Salary\*20)/100;

