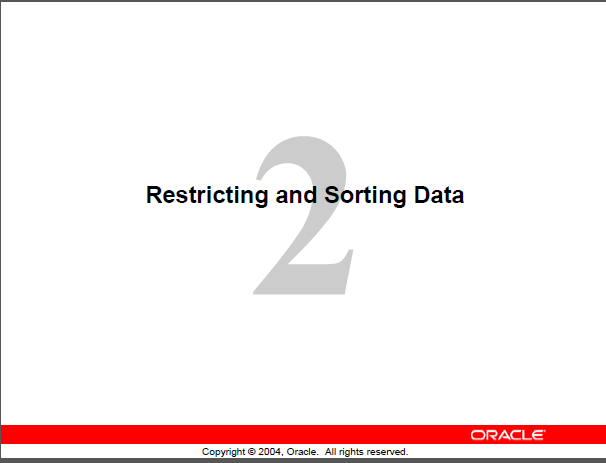
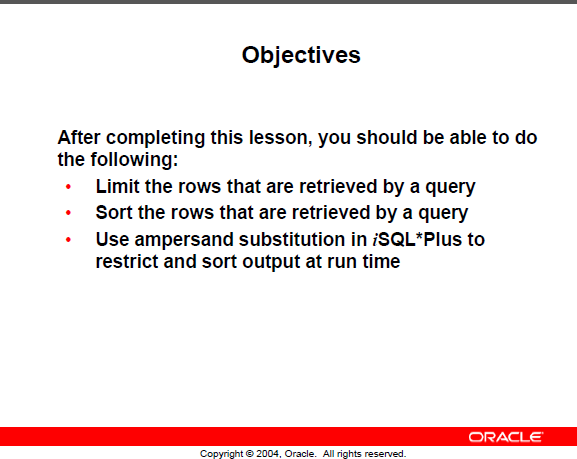
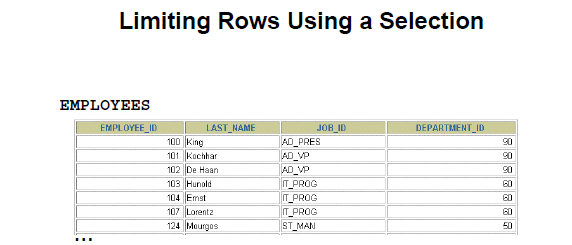
Les02-WHERE Restricting Sorting





Limiting rows --- WHERE

Here are 7 of the 20 employees in the EMPLOYEE table



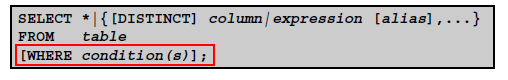
**PROBLEM: 🡺 Show those in department 90 only**

(do it)

Result should be

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE\_ID** | **LAST\_NAME** | **FIRST\_NAME** | **DEPARTMENT\_ID** |
| 100 | King | Steven | 90 |
| 101 | Kochhar | Neena | 90 |
| 102 | De Haan | Lex | 90 |

General Format of WHERE



WHERE 🡺 follows the FROM

Solving the problem.

**PROBLEM: 🡺 Show those in department 90 only**

**SELECT employee\_id, last\_name, job\_id, department\_id**

**FROM employees**

**WHERE department\_id = 90;**

***Another style -- when the list is long and harder to read***

**SELECT employee\_id,**

**last\_name,**

**job\_id,**

**department\_id**

**FROM employees**

**WHERE department\_id = 90;**

WHERE with Character strings and Dates







**SELECT last\_name, job\_id, department\_id**

**FROM employees**

**WHERE last\_name = ‘Whalen’; 🡸 single quotes – exact case**

|  |  |  |
| --- | --- | --- |
| **LAST\_NAME** | **JOB\_ID** | **DEPARTMENT\_ID** |
| Whalen | AD\_ASST | 10 |

WHERE clause

**Compares** – values in a column,

Literal

Arithmetic expressions

Functions (more later)

Date format can be

1 customized – later chapter

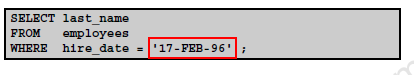
2 Default changed

**NEEDS** 1 Column name

2 Comparison condition

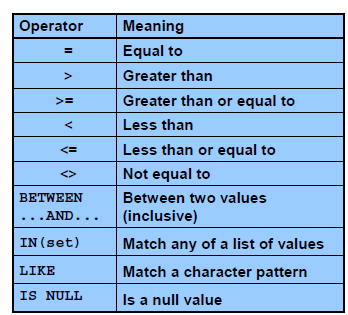
3 Column Name, constant or list of values

**Date Sample**



'02/17/1996'🡸 might be the default

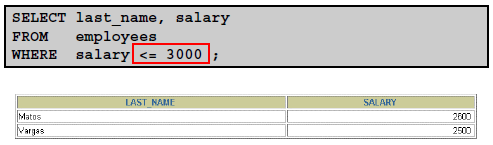
Comparison Operators and Conditions



NOTE:🡺 != and ^= can also represent not equal

Samples:

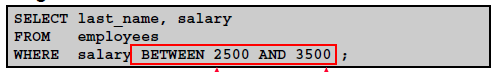
**Salary less than 3000 or a salary equal to 3000**

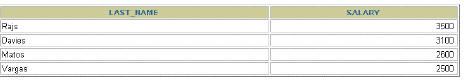


**BETWEEN**

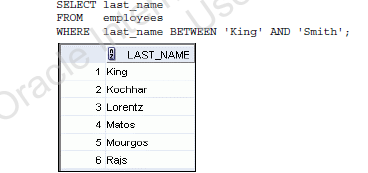
**RANGE of SALARIES 🡺 BETWEEN**

The values are INCLUSIVE of the start and end value limits

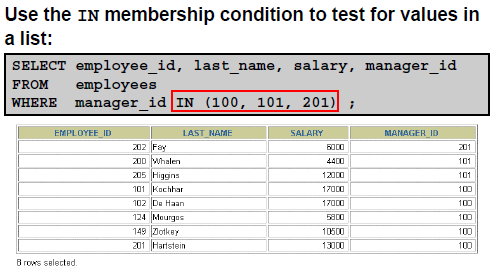




**RANGE of Character values such as name**



**IN**



Show rows that match on any manager ID that are in the list shown.

Later with subqueries this becomes more useful

Also character data

SELECT last\_name, hire\_date, job\_id, department\_id

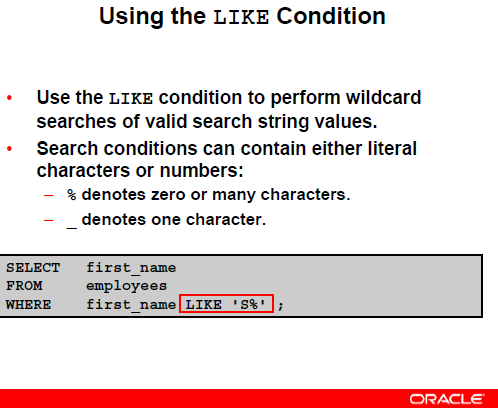
FROM employees

WHERE last\_name in('King', 'Whalen');

**PROBLEM: You don’t know the exact spelling of a name**

**OR**

**You are seeking a search condition**

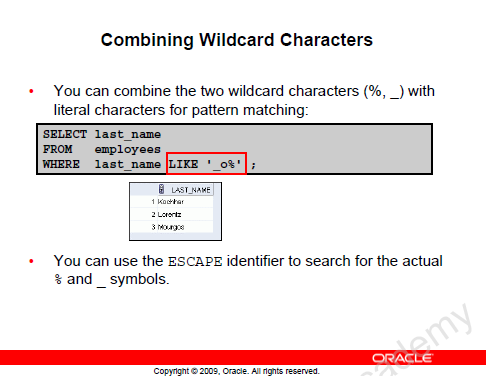


WILD CARD values

**Sample with a year**



**What is the result ?**



**Looking for job IDs that have an underscore in the name.**

**Helps find an EXACT match**

**SELECT last\_name, job\_id**

**FROM employees**

**WHERE job\_id LIKE '%SA\\_%'**

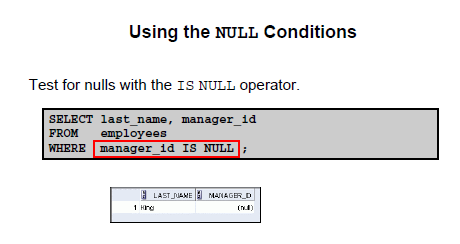
**ESCAPE '\'**

**🡺 Defined the \ as an escape identifier**

**Means looking for 1 anything or nothing in front**

**2 SA followed by \_ the underscore**

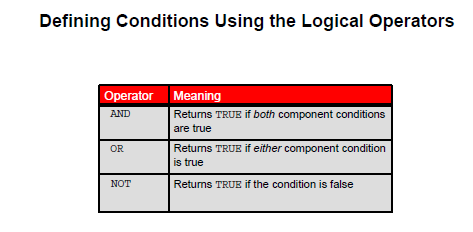
**3 Followed by anything or nothing**



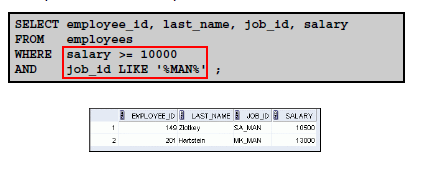
**You cannot be equal to a non-existent value**

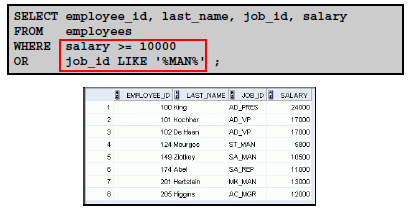
**MUST use the word 🡺 IS**

**Try a SELECT for those who do not have a commission\_pct value**

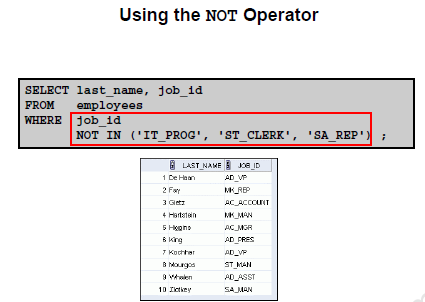


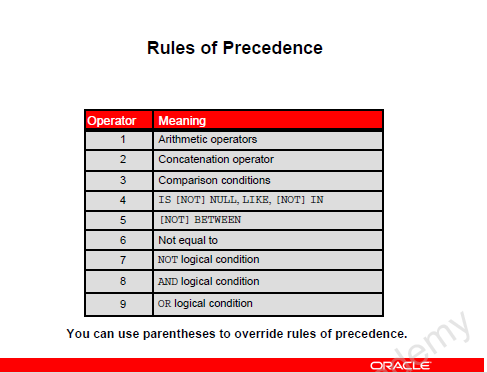
**EXAMPLES:**



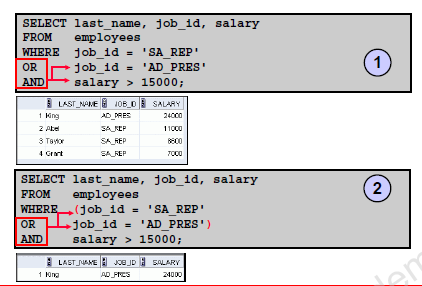


**More complicated example:**





**EXAMPLE:**



By adding the bracketing, it changes the resulting output.

Should use bracketing often

Asks for AD\_PRES and a salary above 15,000

OR

A job\_id of SA\_REP

**SORTING**

**PROBLEM: List all employees in order of their department ID**

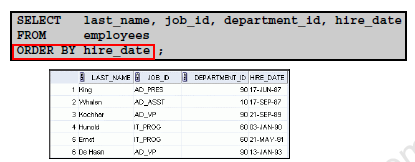
**SELECT \***

**FROM employees**

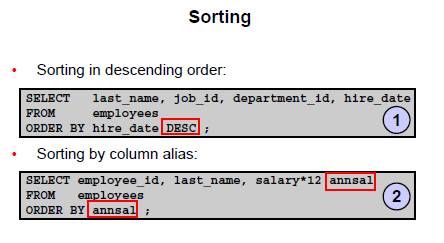
**ORDER BY department\_id;**

**ORDER BY 🡸 is the LAST statement always**

**Another:**



**Sorting Descending order**



**DEFAULT is ASC**

**You can use it but most don’t bother**

**2 You can sort on a column that is not being displayed**

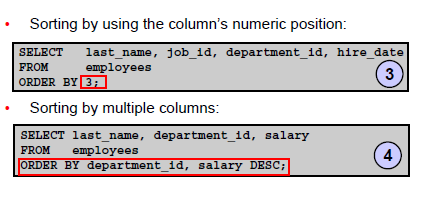
**SELECT employee\_name**

**FROM employees**

**ORDER BY department\_id;**

**NOTE: You can order by with an alias**

**ANOTHER SORTING EXAMPLE**



3 Specifies the numeric position in the SELECT🡪 - in this case the 3rd column or department\_id

4 Shows sorting on two different fields in different directions

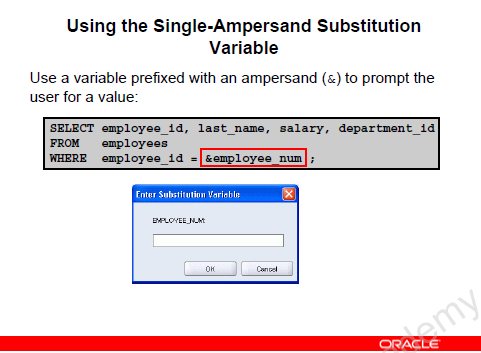
By default the department\_id is sorted lowest value to highest (ASC)

The salary is sorted WITHIN each department in DESC order (highest to lowest)

**Substitution Variables**

**Up until now if you want to query department 90 you specified it in the code**

**What if you want to do several departments but not rewrite the code each time?**



**Allows the user to specify a value at run time and that value is not stored**

**SELECT last\_name AS LAST, department\_id AS DEPT**

**FROM employees**

**where department\_id = &deptno;**

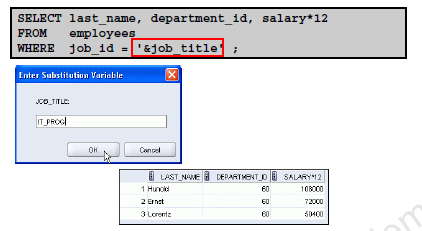
**SELECT last\_name AS LAST, department\_id AS DEPT**

**FROM employees**

**where department\_id in (&deptno1, &deptno2);**

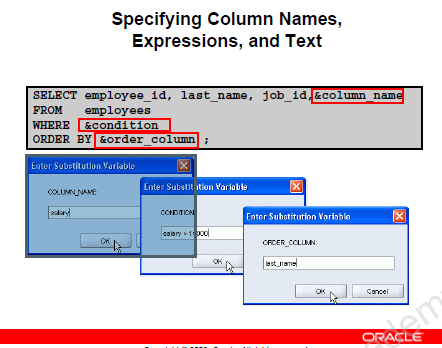
**Note how to use it in a character string**

**It is inside the character string quotes.**

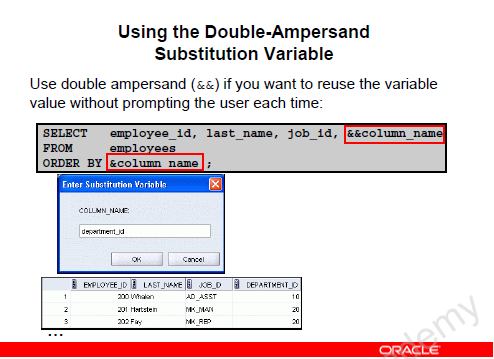


**How to specify a column name to retrieve**

**More flexibility**



**Maintaining the value**

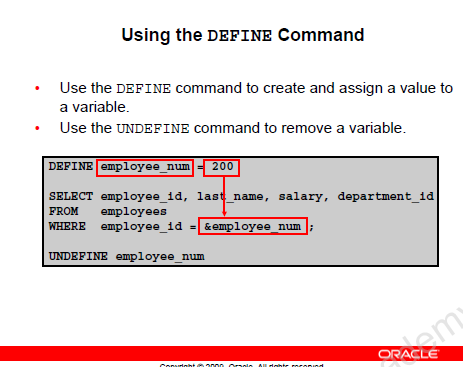


**Prompted first time**

**Value is stored**

**To remove the value from future use**

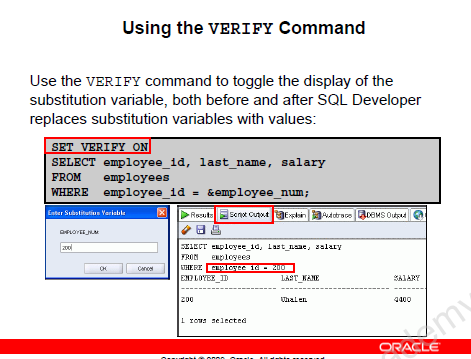
**UNDEFINE column\_name**



Release or unstore the value

User NOT prompted as the value exists already

Store 200



**Sequence of events**

**1 SQL code has no value specified**

**2 It prompts you for the value**

**3 It displays the substitution**

**(show the verify effect)**