Session-2
Retrieving data using SQL SELECT statements

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Limiting rows in SELECT statement

- Restrict the rows that are returned by using WHERE clause.
- The WHERE clause follows the FROM clause.

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE department_id = 90;
```


Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Character Strings and Dates

- Character strings and date values are enclosed in single quotation marks(‘ ‘).
- Character values are case sensitive , and date values are format sensitive.
- The default date format is DD-MON-RR.

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE last_name = ‘King’;
```

No rows will return:

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE last_name = ‘KING’;
```


Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Comparison **Operators**

Operator	Purpose	Example
=	Equality test.	SELECT * FROM employees WHERE salary = 1500;
!= ^= <> ¬=	Inequality test. Some forms of the inequality operator may be unavailable on some platforms.	SELECT * FROM employees WHERE salary != 1500;
> <	"Greater than" and "less than" tests.	SELECT * FROM employees WHERE salary > 1500; SELECT * FROM employees WHERE salary < 1500;
>= <=	"Greater than or equal to" and "less than or equal to" tests.	SELECT * FROM employees WHERE salary >= 1500; SELECT * FROM employees WHERE salary <= 1500;

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Comparison Operators

IN	**"Equal to any member of" test. Equivalent to "= ANY".**	**SELECT * FROM employees WHERE job IN ('CLERK','ANALYST'); SELECT * FROM employees WHERE salary IN (SELECT salary FROM employees WHERE department_id = 30);**
NOT IN	Equivalent to "!=ALL". Evaluates to FALSE if any member of the set is NULL.	SELECT * FROM employees WHERE salary NOT IN (SELECT salary FROM employees WHERE department_id = 30); SELECT * FROM employees WHERE job NOT IN ('CLERK', ANALYST');
ANY SOME	Compares a value to each value in a list or returned by a query. Must be preceded by =, !=, >, <, <=, >=. Evaluates to FALSE if the query returns no rows.	SELECT * FROM employees WHERE salary = ANY (SELECT salary FROM employees WHERE department_id = 30);
ALL	Compares a value to every value in a list or returned by a query. Must be preceded by =, !=, >, <, <=, >=. Evaluates to TRUE if the query returns no rows.	SELECT * FROM employees WHERE salary >= ALL (1400, 3000);

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Comparison Operators

EXISTS	**TRUE if a subquery returns at least one row.**	**SELECT ename, department_id FROM dept WHERE EXISTS (SELECT * FROM employees WHERE dept.deptno = emp.deptno);**
x [NOT] LIKE y [ESCAPE 'z']	TRUE if x does [not] match the pattern y. Within y, the character "%" matches any string of zero or more characters except null. The character "_" matches any single character. Any character, excepting percent (%) and underbar (_) may follow ESCAPE; a wildcard character is treated as a literal if preceded by the character designated as the escape character.	SELECT salary FROM employees WHERE ename LIKE 'SM%'; SELECT ename FROM employees WHERE ename LIKE '%A_B%' ESCAPE '\';
IS [NOT] NULL	Tests for nulls. This is the only operator that you should use to test for nulls.	SELECT ename, department_id FROM employees WHERE comm IS NULL;
[NOT] BETWEEN x AND y	[Not] greater than or equal to *x* and less than or equal to *y*.	`SELECT * FROM employees WHERE salary BETWEEN 2000 AND 3000;`

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Logical Operators

Operator	Function	Example
NOT	Returns TRUE if the following condition is FALSE. Returns FALSE if it is TRUE. If it is UNKNOWN, it remains UNKNOWN.	SELECT * FROM employees WHERE NOT (job IS NULL); SELECT * FROM employees WHERE NOT (salary BETWEEN 1000 AND 2000);
AND	Returns TRUE if both component conditions are TRUE. Returns FALSE if either is FALSE. Otherwise returns UNKNOWN.	SELECT * FROM employees WHERE job = 'CLERK' AND department_id = 10;
OR	Returns TRUE if either component condition is TRUE. Returns FALSE if both are FALSE. Otherwise returns UNKNOWN.	SELECT * FROM employees WHERE job = 'CLERK' OR department_id = 10;

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Sorting Rows

- Sort retrieved rows with the **ORDER BY** clause:
  - ASC: ascending order ; default
  - DESC: descending order
- The ORDER BY clause comes last in SELECT statement

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date;


SELECT employee_id, last_name, salary*12 annsal
FROM employees
ORDER BY annsal DESC;
```


Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Substitution Variables

- Use substitution variables to:
  - Temporarily store values by single-ampersand(&) and double-ampersand (&&).
  - In WHERE conditions
  - In ORDER BY clauses
  - In column expressions

```
SELECT employee_id, last_name , salary
FROM employees
WHERE employee_id = &employee_num;
```


Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com

Session-2
END

Author: Mohammad Kaysar Iqbal Talukdar
adnankaysar@hotmail.com