

# Data Manipulation Language (DML)

Mirunalini.P

SSNCE

January 3, 2020

# Session Objective

- To learn about DML statments
- To learn about the Column Alias
- To learn about comparision and logical operations
- to learn about order by clause

# Table of Contents

- 1 Data Manipulation Language
- 2 Comparison & Logical Conditions
- 3 Order by Clause
- 4 Reference

# Data Manipulation Language (DML)

Includes commands to retrieve, store, modify, delete, insert and update data in database.

- **SELECT**
- **INSERT**
- **UPDATE**
- **DELETE**

# Basic SQL Structure

A typical SQL query has 3 clauses: select, from and where

```
select  A1, A2, . . . . ., An
from    r1, r2, . . . . ., rn
where P
```

- $A_i$  represents an attribute and the select clause contains list of attribute names whose values are to be retrieved by the query
- $R_i$  represents a relation and the from clause contains list of the relation names required to process the query
- P is a predicate and is a Boolean expression that identifies the tuples to be retrieved by the query

The result of an SQL query is a relation.

It is equivalent to the relational algebra query

$$\pi_{(A1.....An)}(\sigma_{P(r1*r2*.....*rm)}) \quad (1)$$

# Basic Select statement

- **SELECT** : is a list of one or more attributes
- **\*** : selects all columns
- **DISTINCT** : suppress duplicates
- **Column — expression** : selects the named column or the expression
- **Alias**: gives selected columns different headings
- **FROM**:table specifies the table containing the columns

# Company Database

**Figure 5.6**

One possible database state for the COMPANY relational database schema.

**EMPLOYEE**

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

# Examples - Select statement (column filtering)

- **To Select all attributes**

`Select * from employee`

- **To select specific columns of all rows**

`SELECT BDATE, ADDRESS,FNAME,LNAME from employee`

- **To remove the duplicates**

`Select distinct(salary) from employee.`



# Examples - Select statement ( tuple filtering)

- 1 Retrieve the birthdate and address of the employee whose name is 'John Smith'.

```
SELECT BDATE, ADDRESS from employee where FNAME='John'  
AND LNAME='Smith'
```

- 2 Retrieve the lname,sex,salary of the male employees whose salary greater than 30000 .

```
SELECT lname,sex,salary from employee where sex='M'  
and salary>=30,000
```

- 3 Retrieve the details of the employee whose salary either greater than 50,000 or should belong to department no 5 .

```
SELECT * from employee where salary>50,000 or dno=5
```

- 4 Retrieve the details of the employee whose do not belong to department no 5 .

```
SELECT * from employee where dno!=5
```

# Using Arithmetic operators

- Arithmetic operators are used in any clause of a SQL statement.
- Select clause may contain arithmetic operations like  $\{+, -, *, \}$

## ① Increment was added to all employees

Select LNAME,SALARY,SALARY+300 from employee.

## ② Increment the salary of the employee by 20% as the original salary and display the details along with their name

Select LNAME,SALARY\*.2 from employee

# Column Alias

A column alias:

- Rename a column heading.
- It is useful with calculations

```
Select Lname as name, SUPERSSN as MGR_ID from employee
```

```
Select last_Name NAME, 12*(salary) Annual Salary  
from employees.
```

```
Select LNAME || FNAME as Employees from EMPLOYEE
```

# Null Value

- A null is a value that is unavailable, unassigned, unknown, or inapplicable.
- A null is not the same as zero or a blank space
- ❶ **To select the employee who have not been allotted with manager**

```
SELECT lname,fname,super_ssn FROM employee  
where  super_ssn is Null
```

- ❷ **To select the employees who have manager**

```
SELECT lname,fname,super_ssn FROM employee  
where  super_ssn is not Null
```

# Comparison Conditions

=	Equal to
>	Greater than
≥	Greater than or equal to
<	Less than
≤	Less than or equal to
≠	Not equal to

- 1 Retrieve the lname of employee whose salary less than equal to 3000

```
SELECT last_name, salary FROM employees WHERE  
salary <= 3000;
```

- 2 Retrieve the bdate and address whose fname is john.

```
SELECT BDATE, ADDRESS FROM EMPLOYEE  
WHERE FNAME='John'
```

# Other Comparison Conditions

<i>BETWEEN...AND..</i>	Between two values (inclusive)
<i>IN(set)</i>	Match any of a list of values
<i>LIKE</i>	Match a character pattern
<i>ISNULL</i>	Is a null value

- 1 Retrieve all employees whose Fname starts with 'J'  
Select Fname,Lname from employee where fname like 'J%'
- 2 Select all employees who were born during the 1950s  
Select fname,lname from employee where BDATE like '%5-'

# Other Comparison Conditions

- 1 Retrieve all employees in dept 5 whose salary between 30,000 and 40,000

```
Select * from employee where(salary between 30,000  
and 40,000) and dno=5;
```

- 2 Display the details of all employee whose superssn no is 5 or 4 or 1

```
Select lname,fname,salary,dno from employee  
where dno in (5,4,1);
```

# Logical Conditions

Logical conditions combines the result of two components to produce single result based on them.

The logical operators are : AND,OR,NOT.

```
Select SSN,LNAME,SALARY FROM EMPLOYEE where SALARY>10,000  
and FNAME like '_o'
```

```
Select SSN,LNAME,SALARY FROM EMPLOYEE where SALARY>10,000  
or FNAME like '%o%'.
```



# ORDER BY Clause

Sort rows with the ORDER BY clause

- ASC: ascending order (the default order)
- DESC: descending order
- The ORDER BY clause comes last in the SELECT statement.

**To select the employees based on their birth\_date**

```
SELECT lname, fname, dno, bdate  
FROM employees  
ORDER BY bdate;
```

# ORDER BY Clause

- To sort by DESC order

```
SELECT lname, fname, dno, bdate  
FROM employees  
ORDER BY bdate DESC;
```

- To sort by Column Alias

```
SELECT ssn, lname, salary*12 annsal  
FROM employee ORDER BY annsal;
```

- To sort by Multiple Columns

```
SELECT ssn, lname, dno, salary  
FROM employees  
ORDER BY dno, salary DESC;
```

# Reference



Fundamentals of Database systems 7<sup>th</sup> Edition by Ramez Elmasri.



Oracle 9i SQL