

## Database Management System – 25 (Retrieval Queries in SQL 2)

Ajay James  
Asst. Prof in CSE  
Government Engineering College Thrissur

### Outline

- Tables as sets
- Eliminate duplicates
- Set Operations
- Substring pattern matching
- Arithmetic operators
- Ordering of tuples

## Tables as Sets in SQL

- SQL usually treats a table not as a set but rather as a **multiset**
  - duplicate tuples can appear more than once in a table, and in the result of a query
- SQL does not automatically eliminate duplicate tuples in the results
  - Duplicate elimination is an expensive operation
  - User may want to see duplicate tuples in the result of a query

## Eliminate duplicates

- If we **do want** to eliminate duplicate tuples from the result of an SQL query, we use the keyword DISTINCT

- Retrieve the salary of every employee

**SELECT ALL Salary**  
**FROM EMPLOYEE;**

- Retrieve all distinct salary value

**SELECT DISTINCT Salary**  
**FROM EMPLOYEE;**

Salary
30000
40000
25000
43000
38000
25000
25000
55000

Salary
30000
40000
25000
43000
38000
55000

## Set Operations

- Union(UNION), set difference (EXCEPT), and set intersection (INTERSECT) operations
- Duplicate tuples are eliminated from the result

## UNION Example

- Make a list of all project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department that controls the project

```
( SELECT DISTINCT Pnumber
  FROM PROJECT, DEPARTMENT, EMPLOYEE
 WHERE Dnum = Dnumber AND Mgr_ssn = Ssn
 AND Lname = 'Smith' )
```

**UNION**

```
( SELECT DISTINCT Pnumber
  FROM PROJECT, WORKS_ON, EMPLOYEE
 WHERE Pnumber = Pno AND Essn = Ssn
 AND Lname = 'Smith' );
```

## Substring Pattern Matching

- LIKE comparison operator - string **pattern matching**
- % replaces an arbitrary number of zero or more characters
- underscore (\_) replaces a single character
- Retrieve all employees whose address is in Houston, Texas.

```
SELECT Fname, Lname
```

```
FROM EMPLOYEE
```

```
WHERE Address LIKE '%Houston,TX%';
```

## LIKE example

- Find all employees who were born during the 1950s.

```
SELECT Fname, Lname
```

```
FROM EMPLOYEE
```

```
WHERE Bdate LIKE '__ 7 _____';
```

## Arithmetic Operators

- Show the resulting salaries if every employee working on the 'ProductX' project is given a 10% raise

```
SELECT E.Fname, E.Lname, 1.1 * E.Salary AS  
    Increased_sal  
FROM EMPLOYEE AS E, WORKS_ON AS W, PROJECT AS P  
WHERE E.Ssn = W.Essn AND W.Pno = P.Pnumber AND  
    P.Pname = 'ProductX';
```

## BETWEEN Operator

- Retrieve all employees in department 5 whose salary is between \$30,000 and \$40,000

```
SELECT * FROM EMPLOYEE  
WHERE (Salary BETWEEN 30000 AND 40000) AND  
    Dno = 5;  
((Salary >= 30000) AND (Salary <= 40000))
```

## Ordering of Query Results

- Retrieve a list of employees and the projects they are working on, ordered by department and, within each department, ordered alphabetically by last name, then first name

```
SELECT D.Dname, E.Lname, E.Fname, P.Pname
FROM DEPARTMENT AS D, EMPLOYEE AS E, WORKS_ON AS W,
      PROJECT AS P
WHERE D.Dnumber = E.Dno AND E.Ssn = W.Essn AND W.Pno =
      P.Pnumber
ORDER BY D.Dname, E.Lname, E.Fname;
```

***ORDER BY** D.Dname **DESC**, E.Lname **ASC**, E.Fname **ASC***

## Summary

```
SELECT <attribute list>
FROM <table list>
[ WHERE <condition> ]
[ ORDER BY <attribute list> ];
```

## Reference

- Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education 6<sup>th</sup> edition and 7<sup>th</sup> edition

Thank you