




UNIVERSITAS
INDONESIA
Visitas, Profektas, Inovitas

FAKULTAS
ILMU
KOMPUTER

Slide 4

Basic SQL Query

CSF2600700 - BASIS DATA
SEMESTER GENAP 2016/2017



These slides are a modification to the supplementary slide of
“Database System”, 7th edition, Elmasri/Navathe, 2015:
**Chapter 7 More SQL: Complex Queries, Triggers, Views, and
Schema Modification**

Additional materials from:
<http://www.postgresql.org/docs/current/static/index.html>

Basic Retrieval Queries

- SQL allows a table (relation) to have two or more tuples that are identical in all their attribute values
 - Different with formal relational model!
 - Use constraints on DDL

Basic Retrieval Queries

Basic statement for retrieving information from a database: **SELECT** statement

```
SELECT    <attribute list>
FROM      <table list>
WHERE     <condition>;
```

where

- <attribute list> is a list of attribute names whose values are to be retrieved by the query.
- <table list> is a list of the relation names required to process the query.
- <condition> is a conditional (Boolean) expression that identifies the tuples to be retrieved by the query.

◦ Logical comparison operators:

◦ =, <, <=, >, >=, and <>

CARTESIAN PRODUCT (CROSS PRODUCT)

Denoted by 'X' operator

Operation that applied on two relations

Produces a new element by combining every tuple from one relation with every tuple from the other relation

For example:

- $R(A_1, A_2, \dots, A_n)$
- $S(B_1, B_2, \dots, B_m)$
- $R \times S = (A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$

If R has $|R|$ tuples and S has $|S|$ tuples, then

- $|R \times S| = |R| * |S|$ tuples

CARTESIAN PRODUCT EXAMPLE

R1

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

R2

X	Y
x1	y1
x2	y2

$R3 = R1 \times R2$

A	B	C	X	Y
a1	b1	c1	x1	y1
a1	b1	c1	x2	y2
a2	b2	c2	x1	y1
a2	b2	c2	x2	y2
a3	b3	c3	x1	y1
a3	b3	c3	x2	y2

JOIN

Combine related tuples from two relations into single “longer” tuples

- General join condition of the form:
 <condition> **AND** <condition> **AND...AND**
 <condition>
- Tuples that **satisfy** the join condition will be combined
- Join condition(s) is defined in the WHERE clause

JOIN EXAMPLE

R1

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

R2

X	Y
a1	y1
a1	y2
a2	y1

```
SELECT *  
FROM R1, R2  
WHERE A = X;
```

A	B	C	X	Y
a1	b1	c1	a1	Y1
a1	b1	c1	a1	y2
a2	b2	c2	a2	y1

Same Data!

Figure 4.3

Results of SQL queries when applied to the COMPANY database state shown in Figure 3.6. (a) Q0. (b) Q1. (c) Q2. (d) Q8. (e) Q9. (f) Q10. (g) Q1C.

(a)

<u>Bdate</u>	<u>Address</u>
1965-01-09	731 Fondren, Houston, TX

(b)

<u>Fname</u>	<u>Lname</u>	<u>Address</u>
John	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, Houston, TX
Ramesh	Narayan	975 Fire Oak, Humble, TX
Joyce	English	5631 Rice, Houston, TX

Query 0. Retrieve the birth date and address of the employee(s) whose name is 'John B. Smith'.

Q0: **SELECT** Bdate, Address
 FROM EMPLOYEE
 WHERE Fname='John' **AND** Minit='B' **AND** Lname='Smith';

Query 1. Retrieve the name and address of all employees who work for the 'Research' department.

Q1: **SELECT** Fname, Lname, Address
 FROM EMPLOYEE, DEPARTMENT
 WHERE Dname='Research' **AND** Dnumber=Dno;

	emp_no	emp_fname	emp_lname	dept_no
1	1	Matthew	Smith	d3
2	2	Ann	Jones	d3
3	3	John	Borrimore	d1
4	4	Andrew	James	d2
5	5	Elisa	Bertoni	d2
6	6	Elke	Hansel	d2
7	7	Sybill	Moser	d1

Tabel department

	dept_no	dept_name	location
1	d1	Developer	Dallas
2	d2	Tester	Seattle
3	d3	Marketing	Dallas

- Siapakah pegawai yang bekerja sebagai *Developer* ?
- Di departemen manakah pegawai bernama depan *Matthew* bekerja?

Figure 4.3

Results of SQL queries when applied to the COMPANY database state shown in Figure 3.6. (a) Q0. (b) Q1. (c) Q2. (d) Q8. (e) Q9. (f) Q10. (g) Q1C.

(c)

<u>Pnumber</u>	<u>Dnum</u>	<u>Lname</u>	<u>Address</u>	<u>Bdate</u>
10	4	Wallace	291Berry, Bellaire, TX	1941-06-20
30	4	Wallace	291Berry, Bellaire, TX	1941-06-20

Query 2. For every project located in ‘Stafford’, list the project number, the controlling department number, and the department manager’s last name, address, and birth date.

Q2: **SELECT** Pnumber, Dnum, Lname, Address, Bdate
 FROM PROJECT, DEPARTMENT, EMPLOYEE
 WHERE Dnum=Dnumber **AND** Mgr_ssn=Ssn **AND**
 Plocation=‘Stafford’;

Ambiguous Attribute Names

Same name can be used for two (or more) attributes

- As long as the attributes are in *different relations*
- Must **qualify** the attribute name with the relation name to prevent ambiguity

```
Q1A:  SELECT  Fname, EMPLOYEE.Name, Address
        FROM    EMPLOYEE, DEPARTMENT
        WHERE   DEPARTMENT.Name='Research' AND
                DEPARTMENT.Dnumber=EMPLOYEE.Dnumber;
```

ALIASES

Some queries need to refer to the same relation twice
In this case, *aliases* are given to the relation name

Query 8: For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.

Q8:

```
SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME  
FROM EMPLOYEE E S  
WHERE E.SUPERSSN = S.SSN
```

E and S are called aliases, or alternate copies of EMPLOYEE

ALIASES

Can also use the **AS** keyword:

Q8:

```
SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME  
FROM EMPLOYEE AS E, EMPLOYEE AS S  
WHERE E.SUPERSSN=S.SSN
```

UNSPECIFIED WHERE-clause

A missing WHERE-clause indicates no condition; hence, all tuples of the relations in the FROM-clause are selected

This is equivalent to the condition **WHERE TRUE**
CROSS PRODUCT: all possible tuple combination

Queries 9 and 10. Select all EMPLOYEE Ssns (Q9) and all combinations of EMPLOYEE Ssn and DEPARTMENT Dname (Q10) in the database.

Q9: **SELECT** Ssn
 FROM EMPLOYEE;

Q10: **SELECT** Ssn, Dname
 FROM EMPLOYEE, DEPARTMENT;

Use of Asterisk

Specify an asterisk (*)

- Retrieve all the attribute values of the selected tuples

Q1C: **SELECT** *

FROM EMPLOYEE

WHERE Dno=5;

Q1D: **SELECT** *

FROM EMPLOYEE, DEPARTMENT

WHERE Dname='Research' **AND** Dno=Dnumber;

Q10A: **SELECT** *

FROM EMPLOYEE, DEPARTMENT;

Tables as Sets in SQL

SQL does not automatically eliminate duplicate tuples in query results

Use the keyword **DISTINCT** in the SELECT clause

- Only distinct tuples should remain in the result
- Query 11. Retrieve the salary of every employee (Q11) and all distinct salary values (Q11A).

Q11: **SELECT** **ALL** Salary
 FROM EMPLOYEE;

Q11A: **SELECT** **DISTINCT** Salary
 FROM EMPLOYEE;

Tables as Sets in SQL

Set operations

- UNION, **EXCEPT** (difference), **INTERSECT**
 - Duplicate tuples are **eliminated**
 - Corresponding multiset operations: UNION ALL, EXCEPT ALL, INTERSECT ALL – duplicate tuples are **not** eliminated
- Query 4.** 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.

```
Q4A: ( 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 and Arithmetic Operators

LIKE comparison operator

- Used for string **pattern matching**
- % replaces an arbitrary number of zero or more characters
- underscore (_) replaces a single character

Standard arithmetic operators:

- Addition (+), subtraction (-), multiplication (*), and division (/)

BETWEEN comparison operator

Substring Pattern Matching and Arithmetic Operators

Example:

```
SELECT Fname, Lname  
FROM EMPLOYEE  
WHERE Address LIKE '%Houston,TX%' ;
```

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

Equivalent with condition

```
((Salary) >= 30000) AND (Salary <= 40000))
```

Ordering of Query Results

Use **ORDER BY** clause

- Keyword **DESC** to see result in a descending order of values
- Keyword **ASC** to specify ascending order explicitly
ORDER BY D.Dname DESC, E.Lname ASC, E.Fname ASC

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

Exercise

- Retrieve the names of all employees in dept. 5 who work more than 10 hours per week on the ProductX project
- List the names of all employees who have a dependent with the same first name as themselves
- Find the names of all employees who directly supervised by 'Franklin Wong'