UNIVERSITAS INDONESIA

FAKULTAS ILMU KOMPUTER

8

Basic SQL Query

CSF2600700 - BASIS DATA

Acknowledgements

This slide is a modification to supplementary slide of
“Database System”, 7th edition, Elmasri/Navathe, 2015: **Chapter 7: More SQL** used in “Basis Data” course in academic years 2018/2019 in the Faculty of Computer Science, Universitas Indonesia.

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

Outline

1. `SELECT` Statement
2. Aliases and
3. `WHERE` Clause

Basic Retrieval Queries

SQL allows a table (relation) to have **two or more tuples that are identical** in all their attribute values

```
majumundur=# create table staff (id varchar(10), nama varchar (100));
CREATE TABLE
majumundur=# insert into staff (id, nama) values ('1', 'Chiaki Morisawa');
INSERT 0 1
majumundur=# insert into staff (id, nama) values ('1', 'Chiaki Morisawa');
INSERT 0 1
majumundur=# select * from staff;
 id |       nama
----+-----------------
 1  | Chiaki Morisawa
 1  | Chiaki Morisawa
(2 rows)
```

- ➔ Different with formal relational model!
- ➔ Use **constraints on DDL**

```
majumundur=# delete from staff where id = '1';
DELETE 2
majumundur=# alter table staff add primary key (id);
ALTER TABLE
majumundur=# \d+ staff
                                    Table "public.staff"
 Column |          Type          | Collation | Nullable | Default | Storage  | Stats target | Description
--------+------------------------+-----------+----------+---------+----------+--------------+-------------
 id     | character varying(10)  |           | not null |         | extended |              |
 nama   | character varying(100) |           |          |         | extended |              |
Indexes:
    "staff_pkey" PRIMARY KEY, btree (id)
Access method: heap
```

Basic Retrieval Queries (Cntd.)

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:

=	equal to	>	more than
<	less than	>=	more than or equal to
<=	less than or equal to	<>	not equal tp

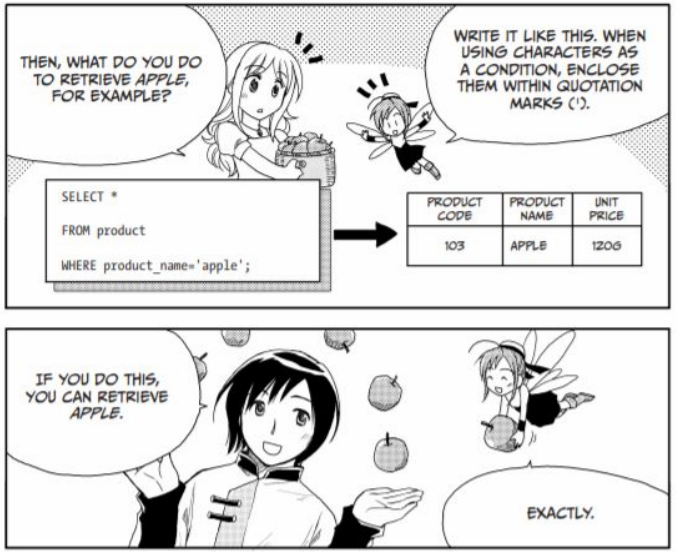


Illustration: Takahashi & Azuma (2014)

Cartesian Product (Cross Product)

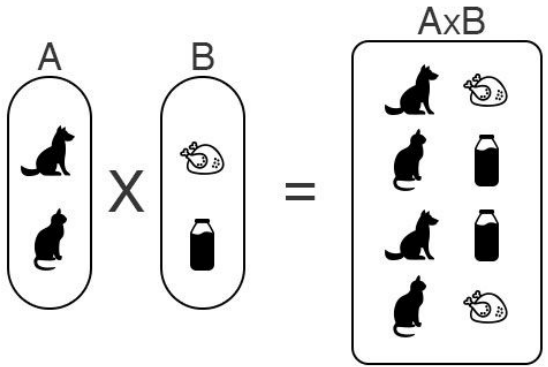
Denoted by '$\times$' operator

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:

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

If R has |R| tuples and S has |S| tuples, then |R x S| =|R| * |S| tuples



Cartesian Product of Two Sets.

Illustration: Mathtopia

Cartesian Product Example

R1

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

R2

X	Y
x1	y1
x2	y2

R3 = R2 x R1

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

```
majumundur=# select * from R1, R2;
 a  | b  | c  | x  | y
----+----+----+----+----
 a1 | b1 | c1 | x1 | y1
 a2 | b2 | c2 | x1 | y1
 a3 | b3 | c3 | x1 | y1
 a1 | b1 | c1 | x2 | y2
 a2 | b2 | c2 | x2 | y2
 a3 | b3 | c3 | x2 | y2
(6 rows)
```

Join

Combine related tuples from two relations into single "longer" tuples

General join condition of the form:

```
<condition> AND <condition> AND
<condition> AND … AND <condition>
```

- Tuples that satisfy the join condition will be combined
- Join condition(s) is defined in the WHERE clause

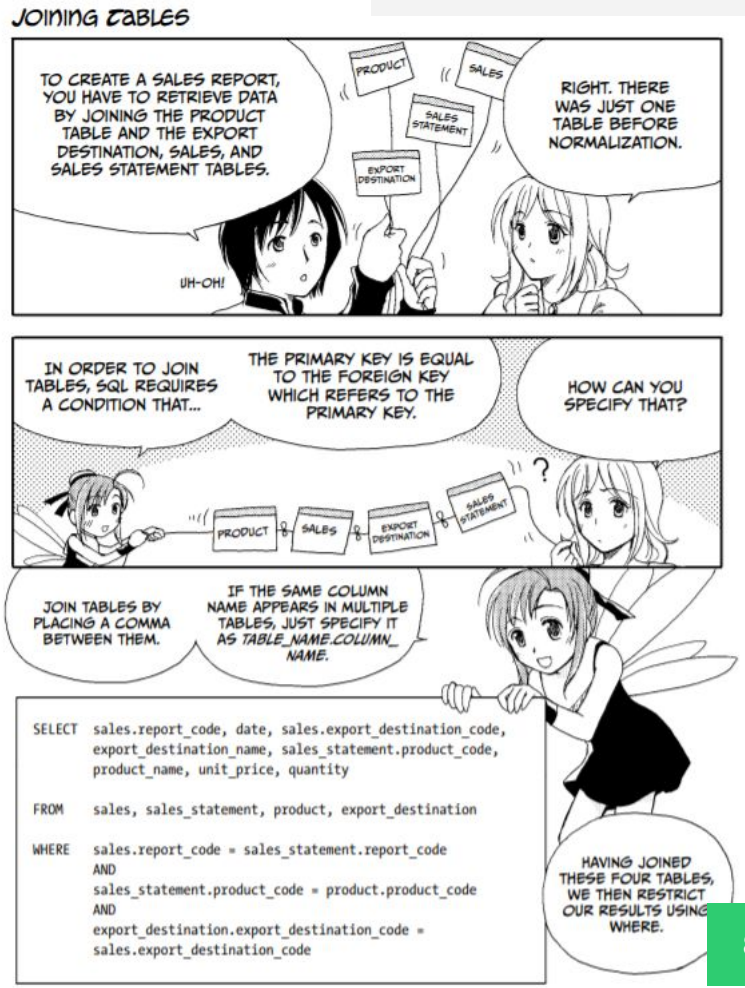


Illustration: Takahashi & Azuma (2014)

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

Join Example

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)

Bdate	Address
1965-01-09	731Fondren, Houston, TX

(b)

Fname	Lname	Address
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;
```

Latihan

EMPLOYEE

	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

DEPARTMENT

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

Buatlah SQL statement untuk menjawab pertanyaan berikut ini.

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

Example

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)

Pnumber	Dnum	Lname	Address	Bdate
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';
```

Outline

1. `SELECT` Statement
2. Aliases
3. `WHERE` Clause

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 AS E, EMPLOYEE AS S
      WHERE    E.Super_ssn = S.Ssn;
```

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

Aliases (Cntd.)

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.Super_ssn = S.Ssn;
```

Illustration: https://comicbook.com/gaming/news/marvels-spider-man-ps4-memes/

Outline

1. `SELECT` Statement
2. Aliases
3. `WHERE` Clause

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 posible 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;
```

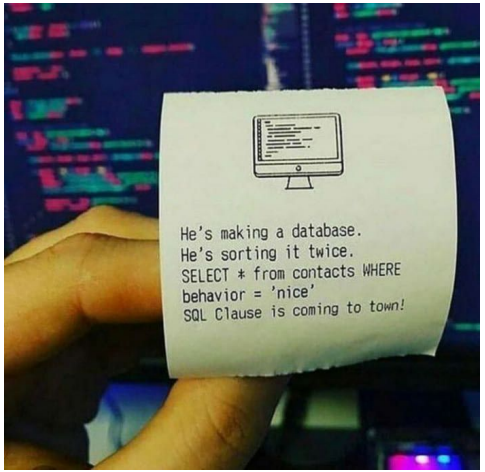


Illustration: https://www.pinterest.com/pin/502221795945959865/

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;
```

Figure 6.4
Results of additional SQL queries when applied to the COMPANY database state shown in Figure 5.6. (a) Q11. (b) Q11A.

(a)

Salary
30000
40000
25000
43000
38000
25000
25000
55000

(b)

Salary
30000
40000
25000
43000
38000
55000

Tables as Sets in SQL (Cntd.)

Set operations

`UNION`, `EXCEPT` (difference), `INTERSECT`
Duplicate tuples are eliminated

Corresponding multiset operations: `UNION ALL`, `EXCEPT ALL`, `INTERSECT ALL` – duplicate tuples are not eliminated

(a)

R
A
a1
a2
a2
a3

S
A
a1
a2
a4
a5

(b)

T
A
a1
a1
a2
a2
a2
a3
a4
a5

(c)

T
A
a2
a3

(d)

T
A
a1
a2

Figure 6.5
The results of SQL multiset operations. (a) Two tables, R(A) and S(A).
(b) R(A)UNION ALL S(A).
(c) R(A) EXCEPT ALL S(A).
(d) R(A) INTERSECT ALL S(A).

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

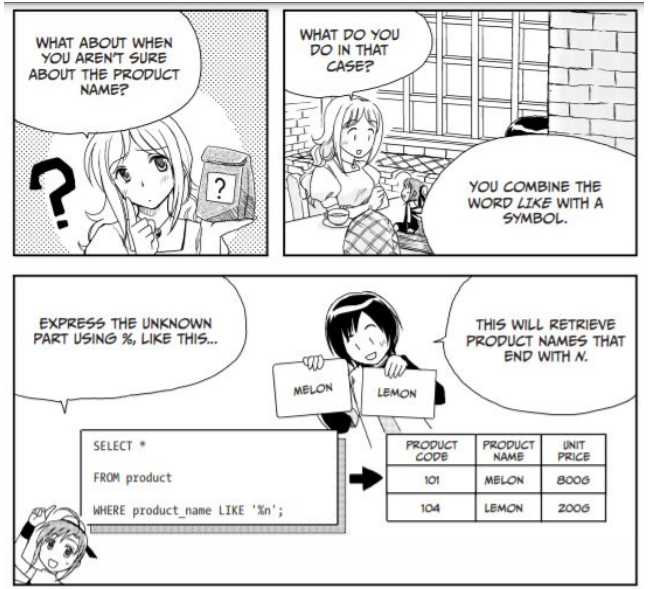


Illustration: Takahashi & Azuma (2014)

Substring Pattern Matching and Arithmetic Operators (Cntd.)

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> ];
```

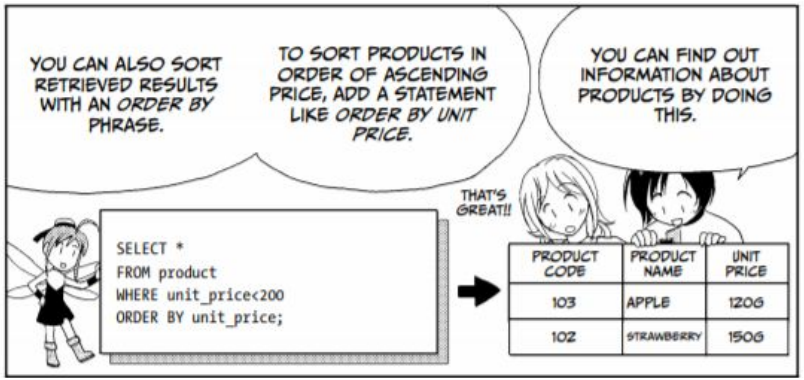


Illustration: Takahashi & Azuma (2014)

Exercise

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

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

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

Q&A