



DATABASE SYSTEMS

Dr. Noha Nagy


Lecture 9

Subqueries and set operations

SQL SYNTAX

2

```
SELECT <Column list>  
FROM <table names>  
[WHERE <Condition>]  
[GROUP BY <Column list>]  
[HAVING <Condition>]  
[ORDER BY <Column list>]
```



Single table
Multiple tables
(Join)

Aliases

3

- Aliases rename columns or tables to
 - ▣ Make names more meaningful
 - ▣ Make names shorter and easier to type
 - ▣ Resolve ambiguous names

- Two forms:
- Column alias
**SELECT column
AS newName...**
- Table alias
**SELECT ...
FROM table AS
newName**

Aliases and 'Self-Joins'

4

Aliases can be used to copy a table, so that it can be combined with itself:

Get the names of all employees who work in the same department as Andy.

Employee

Name	Dept
John	Marketing
Mary	Sales
Peter	Sales
Andy	Marketing
Anne	Marketing

Aliases and Self-Joins

5

Employee A

A

Name	Dept
John	Marketing
Mary	Sales
Peter	Sales
Andy	Marketing
Anne	Marketing

Employee B

B

Name	Dept
John	Marketing
Mary	Sales
Peter	Sales
Andy	Marketing
Anne	Marketing

Aliases and Self-Joins

6

SELECT ... FROM Employee A, Employee B ...

A.Name	A.Dept	B.Name	B.Dept
John	Marketing	John	Marketing
Mary	Sales	John	Marketing
Peter	Sales	John	Marketing
Andy	Marketing	John	Marketing
Anne	Marketing	John	Marketing
John	Marketing	Mary	Sales
Mary	Sales	Mary	Sales
Peter	Sales	Mary	Sales
Andy	Marketing	Mary	Sales
Andy	Marketing	Mary	Sales

Aliases and Self-Joins

```
SELECT ... FROM Employee A, Employee B
WHERE A.Dept = B.Dept
```

A.Name	A.Dept	B.Name	B.Dept
John	Marketing	John	Marketing
Andy	Marketing	John	Marketing
Anne	Marketing	John	Marketing
Mary	Sales	Mary	Sales
Peter	Sales	Mary	Sales
Mary	Sales	Peter	Sales
Peter	Sales	Peter	Sales
John	Marketing	Andy	Marketing
Andy	Marketing	Andy	Marketing
Andy	Marketing	Andy	Marketing

Aliases and Self-Joins

8

```
SELECT ... FROM Employee A, Employee B
WHERE A.Dept = B.Dept AND B.Name = 'Andy'
```

A.Name	A.Dept	B.Name	B.Dept
John	Marketing	Andy	Marketing
Andy	Marketing	Andy	Marketing
Anne	Marketing	Andy	Marketing

A.Name
John
Andy
Anne

Aliases and Self-Joins

9

```
SELECT A.Name FROM Employee A, Employee B  
WHERE A.Dept = B.Dept AND B.Name = 'Andy'
```

A.Name
John
Andy
Anne

The result is the names of all employees who work in the same department as Andy.

Set operator

10

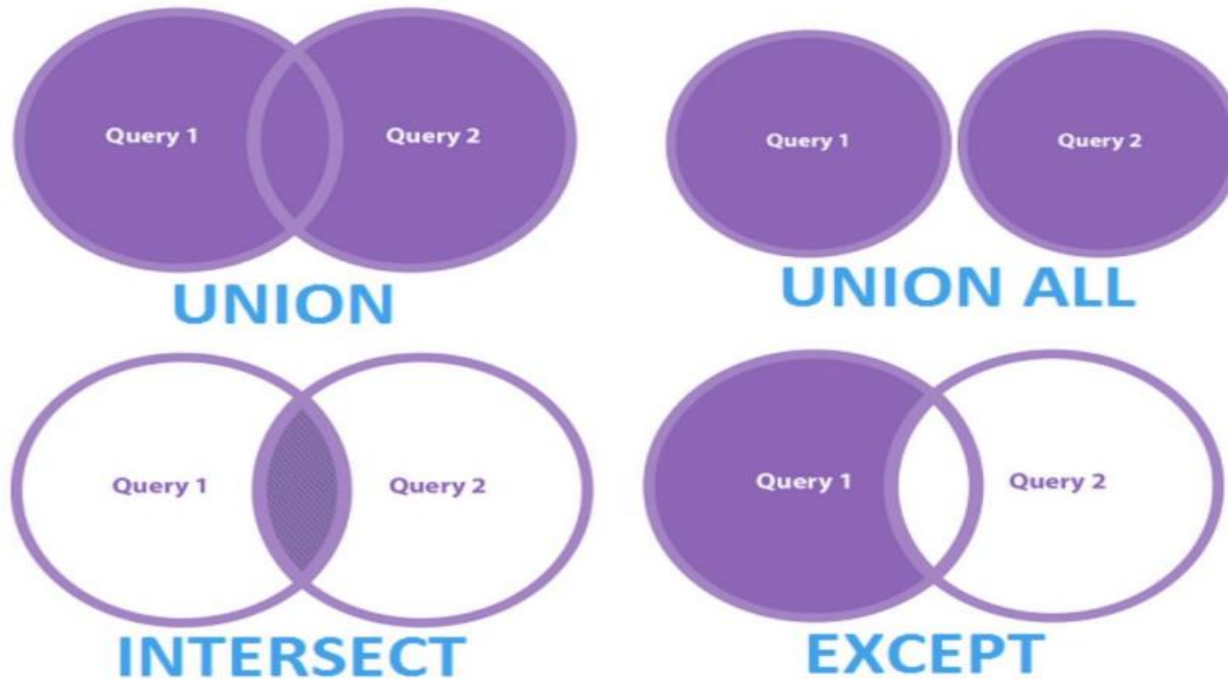
- Union \cup
- Intersection \cap
- Difference(Except) $-$

The two relation
must be union
compatible

Difference between Union, Union All, Intersect and Except

11

The results of multiple queries can be combined into a single result set.



Union Operator

12

- The UNION operator is used to combine the result-set of two or more SELECT statements.
- The relations need to be **union compatible**
 - ▣ Each SELECT statement within UNION must have the **same number of columns**
 - ▣ The columns must also have **similar data types**
 - ▣ The columns in each SELECT statement must also be in the **same order**

Union

13

Customers(cid,cname,city)

Suppliers(sid,sname,city)

- Get the cities of all customers and suppliers in ascending order

```
SELECT City FROM Customers
```

```
UNION
```

```
SELECT City FROM Suppliers
```

```
ORDER BY City;
```

Can I have where condition?

Union

14

Customers(cid,cname,city, country)

Suppliers(sid,sname,city, country)

- If some customers or suppliers have the same city, each city will only be listed once, because UNION selects only **distinct values**.

```
SELECT City FROM Customers
UNION
SELECT City FROM Suppliers
ORDER BY City;
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country
1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	London	EC1 4SD	UK
2	New Orleans Cajun Delights	Shelley Burke	P.O. Box 78934	New Orleans	70117	USA
3	Grandma Kelly's Homestead	Regina Murphy	707 Oxford Rd.	Ann Arbor	48104	USA

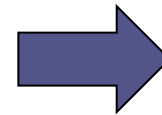
SELECT City FROM Customers
 UNION
 SELECT City FROM Suppliers
 ORDER BY City;

City
Ann Arbor
Berlin
London
Mexico D.F.
New Orleans

To select duplicate values

16

```
SELECT City FROM Customers
UNION ALL
SELECT City FROM Suppliers
ORDER BY City;
```



City
Ann Arbor
Berlin
London
Mexico D.F.
Mexico D.F.
New Orleans

Exercise

17

- selects all the different German cities (only distinct values) from "Customers" and "Suppliers"
- Customers(CID,Cname,City,Country)
- Supplier(SID,Sname, City,Country)

```
SELECT City, Country FROM Customers
WHERE Country='Germany'
UNION
SELECT City, Country FROM Suppliers
WHERE Country='Germany'
ORDER BY City;
```

Borrower(ID, Customer_name, amount)

Depositor(ID, customer_name, balance)

Set Operations

18

- Find all customers who have both a loan, an account or both.
(select customer_name from depositor)
union
(select customer_name from borrower)
- Find all customers who have both a loan and an account.
(select customer_name from depositor)
intersect
(select customer_name from borrower)
- Find all customers who have an account but no loan.
(select customer_name from depositor)
except
(select customer_name from borrower)



Subqueries

Processing Multiple Tables -- Subqueries

20

- Subquery = placing an inner query (SELECT statement) inside an outer query
- Options:
 - ▣ In a condition of the WHERE clause
 - ▣ As a “table” of the FROM clause
 - ▣ Within the HAVING clause

Subqueries Types

21

1- Subqueries that return a single, scalar value

- ▣ operator (=, <, >, <>)

2- Subqueries that operate on lists but the values must be from a single column of a table.

- ▣ IN
- ▣ ANY
- ▣ ALL

3- Subqueries that use the EXISTS operator to test the *existence* of data rows satisfying specified criteria.

Subquery Example

22

Customer(Cid, Cname, Salary)
Order(Oid,pid,Odate,CID)

- Show all customers who have placed an order

SELECT Distinct Cname
From Customer
WHERE Cid **IN**

The IN operator will test to see if the CUSTOMER_ID value of a row is included in the list returned from the subquery

(SELECT DISTINCT CID FROM ORDERS);

Subquery is embedded in parentheses. In this case it returns a list that will be used in the WHERE clause of the outer query

Subqueries

23

□ Show all customers who have placed an order

Customer

CName	CID	Salary
Nancy	1	1000
Mark	2	1500
Ali	3	2000

Orders

OID	PID	CID	Odate
10308	1024	1	18/9/2016
10857	1055	2	3/5/2017
10698	1022	1	5/1/2017

CName
Nancy
Mark

- Get all the products names that have been sold at least once.

Product

Pid	Name	Description
1	TV	52 inch
2	Camera	High resolution
3	Laptop	Very thin black color

ProductSales

ID	Pid	Unit price	Qty
1	1	450	7
2	3	1200	8
3	3	1200	20

- Get all the products id and names that have been sold at least once.

Outer
Query

```
Select Pid, Name  
From Product  
Where Pid in (select distinct pid from ProductSales)
```

Product

Pid	Name	Description
1	TV	52 inch
2	Camera	High resolution
3	Laptop	Very thin black color

ProductSales

ID	Pid	Unit price	Qty
1	1	450	7
2	3	1200	8
3	3	1200	20

Inner
Query

- Get all the products that have been sold at least once.


26

Subquery

```
Select Pid, Name  
From Product  
Where Pid in (select distinct pid from ProductSales)
```


```
Select Pid, Name  
From Product ,ProductSales  
Where product.Pid= ProductSales.Pid)
```

Product



Pid	Name	Description
1	TV	52 inch
2	Camera	High resolution
3	Laptop	Very thin black color

ProductSales



ID	Pid	Unit price	Qty
1	1	450	7
2	3	1200	8
3	3	1200	20

Join

- Get all the products that not sold.

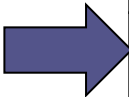
Write this query in two different ways

Select Pid, Name

From Product

Where Pid not in (select distinct pid from ProductSales)

Product



Pid	Name	Description
1	TV	52 inch
2	Camera	High resolution
3	Laptop	Very thin black color

ProductSales

ID	Pid	Unit price	Qty
1	1	450	7
2	3	1200	8
3	3	1200	20

Write an equivalent query

28

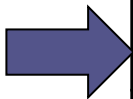
- Get all the products that not sold at least once.

```
Select Pid, Name  
From Product  
Where Pid not in (select distinct pid from ProductSales)
```

```
Select product.Pid, product.Name  
From Product left join ProductSales  
On product.Pid=ProductSales.Pid  
Where ProductSales.Pid is null
```

Product

Pid	Name	Description
1	TV	52 inch
2	Camera	High resolution
3	Laptop	Very thin black color



ProductSales

ID	Pid	Unit price	Qty
1	1	450	7
2	3	1200	8
3	3	1200	20

Is there another way?

Product

name	category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

name	store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

```
SELECT Product.name  
FROM Product  
Except Select purchase.name  
From Purchase
```



name
oneclick

Exercise

30

Company (Name, city)
Product(PID, Pname, maker)
Purchase(PID, Pname, buyername)

Get the cities of the company which make product purchased by Tom Cruise without repetition

Exercise

31

```
SELECT DISTINCT Company.city
FROM    Company
WHERE   Company.name IN
        (SELECT Product.maker
         FROM   Purchase , Product
         WHERE  Product.pname=Purchase.pname
         AND   Purchase .buyer = 'Tom Cruise');
```

```
SELECT DISTINCT Company.city
FROM    Company, Product, Purchase
WHERE   Company.name= Product.maker
        AND Product.pname = Purchase.pname
        AND Purchase.buyer = 'Tom Cruise'
```

Some Queries Cont.

explicit (enumerated) set of values

32

It is also possible to use an **explicit (enumerated) set of values** in the WHERE-clause rather than a nested query

Query 6: Retrieve the social security numbers of all employees who work on project number 1, 2, or 3.

```
SELECT DISTINCT ESSN  
FROM WORKS_ON  
WHERE PNO IN (1, 2, 3)
```


Exists and Not Exists

33

- ❑ EXISTS condition is considered "to be met" if the sub query returns at least one row.
- ❑ NOT EXISTS is satisfied if no rows are returned by the sub query.
- ❑ EXISTS always returns TRUE or FALSE and it will return TRUE as soon as it finds only a single matching row in the sub query, or FALSE, if it find none. Therefore, NOT EXISTS will return TRUE only if no row is returned from the sub query.

Some Queries Cont. EXISTS

34

EXISTS is used to check whether the result of a nested query is empty (contains no tuples) or not

```
SELECT FNAME, LNAME
FROM EMPLOYEE
WHERE EXISTS (SELECT *
               FROM DEPENDENT
               WHERE Employee.SSN=Dependent.SSN
               AND  FNAME=NAME)
```

Writing Queries

35

- When writing queries
 - ▣ There are often many ways to write the query
 - ▣ You should worry about being correct, clear, and concise in that order
 - ▣ Don't worry about being clever or efficient
- Most DBMSs have query optimisers
 - ▣ These take a user's query and figure out how to efficiently execute it
 - ▣ A simple query is easier to optimise
 - ▣ We'll look at some ways to improve efficiency later

Join or Subqueries?

36

- Some cases can not be done except with subqueries and is easier to be created using subqueries.
- It is usually better in terms of response time to use Joins.