DATABASE SYSTEMS

Dr. Noha Nagy

Lecture 9

Inserting from Another Table

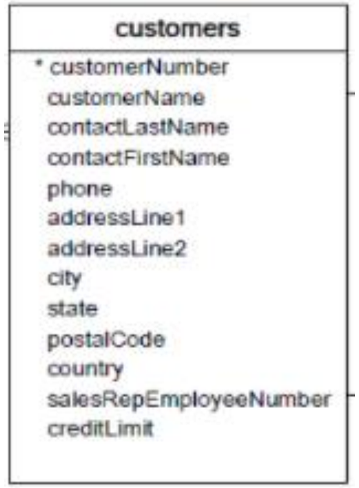
MyCustomers(id, name, city)

Insert all customers from the table “Customers” to table “MyCustomers”

insert into myCustomers

Select customerNumber, customerName, city

from customers;



customers

* customerNumber
customerName
contactLastName
contactFirstName
phone
addressLine1
addressLine2
city
state
postalCode
country
salesRepEmployeeNumber
creditLimit

SQL SYNTAX

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

Single table
Multiple tables
(Join)

Left Join

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

```
select *
   from one left join two
      on one.x = two.x;
```

X	A	X	B
1	a	.	
2	b	2	x
4	d	.	

Right Join

Table Two

X	B
2	x
3	y
5	v

Table One

X	A
1	a
4	d
2	b

```
select *
    from two right join one
       on one.x = two.x;
```

X	B	X	A
.		1	a
2	x	2	b
.		4	d

Full Join

Table One

X	A
1	a
4	d
2	b

Table Two

X	B
2	x
3	y
5	v

```
select *
   from one full join two
      on one.x = two.x;
```

X	A	X	B
1	a	.	
2	b	2	x
.		3	y
4	d	.	
.		5	v

Aliases and ‘Self-Joins’

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

```
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

Aliases and Self-Joins

```
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

- Union U
- Intersection ∩
- Difference -

The two relation must be union compatible

Union Operator

- 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

Customers(cid,cname,city)
Suppliers(sid,sname,city)

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

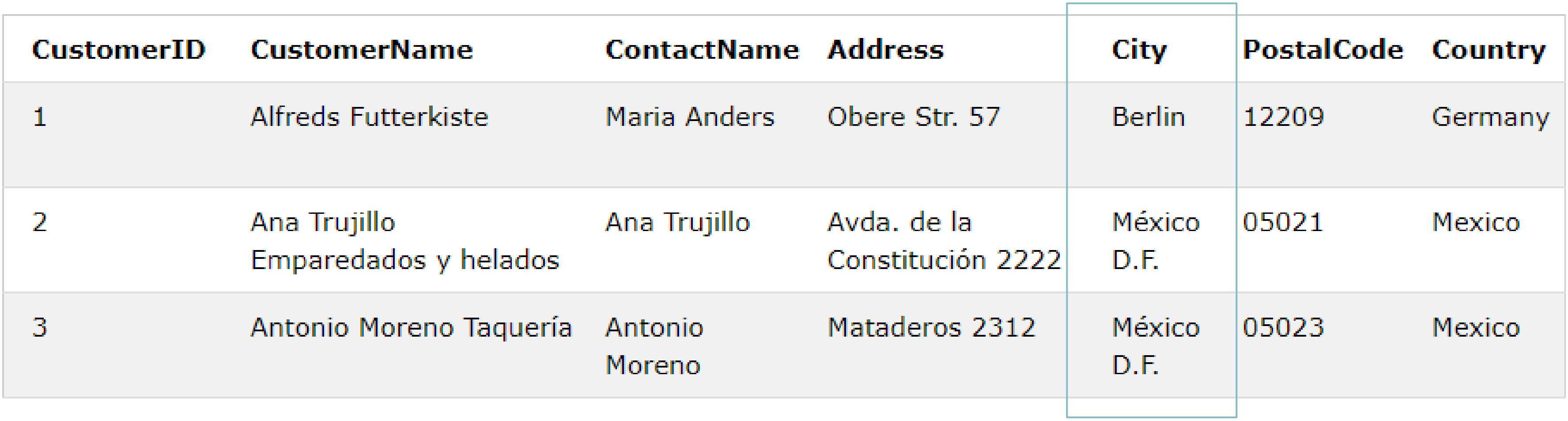
```
SELECT City FROM Customers
UNION
SELECT City FROM Suppliers
ORDER BY City;
```

Union

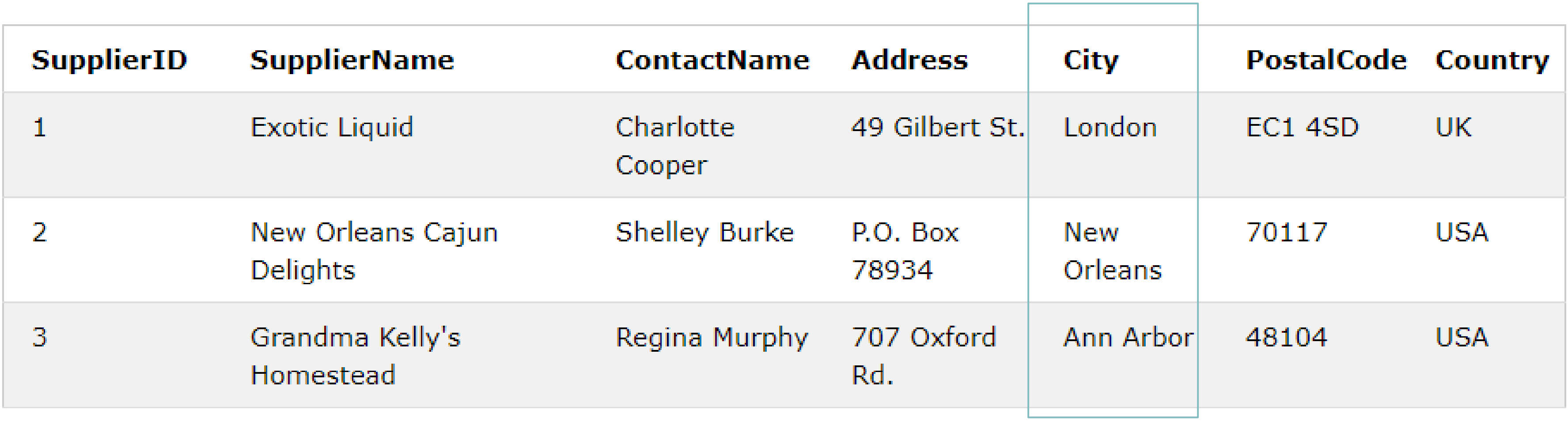
Customers(cid,cname,city, country)
Suppliers(sid,sname,city, country)

- 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

```
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

- □ selects all the different German cities (only distinct values) and the country 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

- 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)
```

Processing Multiple Tables -- Subqueries

- 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

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.

Example

Customer(Cid, Cname, Address, Phone)
Order(Oid,Date,customer_ID)

- Show all customers who have placed an order

```
SELECT Cname
From Customer
WHERE Cid IN
          (SELECT DISTINCT CUSTOMER_ID FROM ORDER);
```

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

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

Example

- 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 that have been sold at least once. □

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

Outer Query

Inner Query

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 that have been sold at least once.

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)
```

Join

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

Exercise 1

- □ Get all the product’s names that not sold.
- □ Get all the product’s id that not sold

Write these queries in two different ways

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

Exercise

- Get all the products that not sold.

```
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

Exercise

- Get all the products id that not sold.

```
Select Pid
From Product
Except
Select 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

Exercise 2

Employee(SSN, Fname, Lname, Salary, DepId)
Dependent(DSSN, Name, SSN)
Department(DepId,Depname, Location)

Retrieve the name of each employee who has a dependent with the same first name as the employee.[write in two different ways]

Exercise 2

Employee(SSN, Fname, Lname, Salary, DepId)
Dependent(DSSN, Name, SSN)
Department(DepId,Depname, Location)

Retrieve the name of each employee who has a dependent with the same first name as the employee.[write in two different ways]

```
SELECT E.FNAME, E.LNAME
FROM EMPLOYEE AS E
WHERE E.SSN IN (SELECT SSN
            FROM DEPENDENT As D
             WHERE   E.FNAME=D.NAME)
```

```
SELECT E.FNAME, E.LNAME
 FROM EMPLOYEE E, DEPENDENT D
 WHERE E.SSN=D.SSN AND
             E.FNAME=D.NAME
```