

Two groups of SQL

DDL – creating tables, columns

DML – Querying and modifying data

Where did DML SQL come from?

The Relational Algebra !

# *The Relational Algebra*

---

- Write a query in SQL, submit it
- The DBMS engine “parses” the SQL, checks syntax
- Builds an execution plan (based on Relational Algebra)
- Executes the SQL
- Returns an Answer Set (a table)
- Stores the SQL in cache (Oracle Shared Pool)

# The Relational Algebra

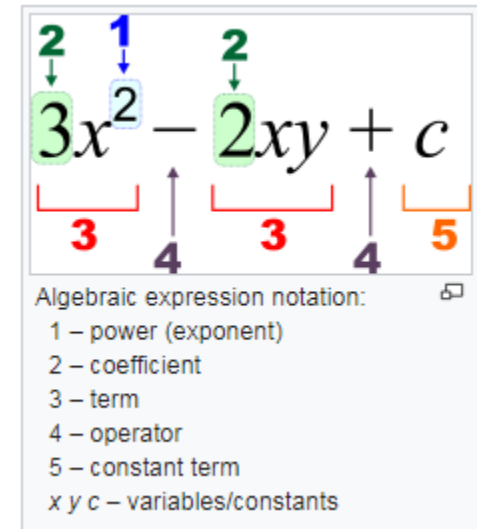
- Preparing the query for execution

- It is like algebra

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

- Variables and Operations

- The “variables” are relations
- The operations are
  - Union
  - Intersection
  - Difference
  - Selection
  - Projection
  - Cartesian Product
  - Join



Some Operations are “Set” Operations

- **Think of each table as entire “set” of data**
- **Consider two tables: R and S**
  - Same attributes, same domains
- I can do the following set operations in R.A.
  - R <union> S                      all elements in R or S or both
  - R <intersection> S                the set of elements in R and S
  - R <difference> S                  the set of elements in R but not in S

# Example Union

## Table C

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

## Table S

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

# *Example Union*

---

C <union> S

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.	5021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	5023	Mexico
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

# Projection

Produce a new relation that is a subset of columns

Table C

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

<projection> CustomerID, CustomerName, City (C)

1	Alfreds Futterkiste	Berlin
2	Ana Trujillo Emparedados y helados	México D.F.
3	Antonio Moreno Taquería	México D.F.

Produce a new relation that is a subset of tuples  
– Based on a condition

Table C

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

<selection> <Country=Mexico> (C)

2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	5021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	5023	Mexico



# *Cartesian Product*

---

Produce a new relation that is the combination of every tuple in one table combined with every tuple in the other.

Table C

CustomerID	CustomerName
1	Howard Snyder
2	Yoshi Lattimer
3	John Steel
4	Jaime Yorres
5	Fran Wilson
6	Rene Phillips

Table O

OrderID	CustomerID	OrderTotal
10262	1	10556.22
10269	7	124.56
10278	8	105938.44
10304	12	1034.34
10307	2	15679.02
10322	17	9305.18

CustomerID	CustomerName	OrderID	CustomerID	OrderTotal
1	Howard Snyder	10262	1	10556.22
1	Howard Snyder	10269	7	124.56
1	Howard Snyder	10278	8	105938.44
1	Howard Snyder	10304	12	1034.34
1	Howard Snyder	10307	2	15679.02
1	Howard Snyder	10322	17	9305.18
2	Yoshi Lattimer	10262	1	10556.22
2	Yoshi Lattimer	10269	7	124.56
2	Yoshi Lattimer	10278	8	105938.44
2	Yoshi Lattimer	10304	12	1034.34
2	Yoshi Lattimer	10307	2	15679.02
2	Yoshi Lattimer	10322	17	9305.18
3	John Steel	10262	1	10556.22
3	John Steel	10269	7	124.56
3	John Steel	10278	8	105938.44
3	John Steel	10304	12	1034.34
3	John Steel	10307	2	15679.02
3	John Steel	10322	17	9305.18
4	Jaime Yorres	10262	1	10556.22
4	Jaime Yorres	10269	7	124.56
4	Jaime Yorres	10278	8	105938.44
4	Jaime Yorres	10304	12	1034.34
4	Jaime Yorres	10307	2	15679.02
4	Jaime Yorres	10322	17	9305.18
5	Fran Wilson	10262	1	10556.22
5	Fran Wilson	10269	7	124.56
5	Fran Wilson	10278	8	105938.44
5	Fran Wilson	10304	12	1034.34
5	Fran Wilson	10307	2	15679.02
5	Fran Wilson	10322	17	9305.18
6	Rene Phillips	10262	1	10556.22
6	Rene Phillips	10269	7	124.56
6	Rene Phillips	10278	8	105938.44
6	Rene Phillips	10304	12	1034.34
6	Rene Phillips	10307	2	15679.02
6	Rene Phillips	10322	17	9305.18

# *Cartesian Product*

← Product

CustomerID	CustomerName	OrderID	CustomerID	OrderTotal
1	Howard Snyder	10262	1	10556.22
1	Howard Snyder	10269	7	124.56
1	Howard Snyder	10278	8	105938.44
1	Howard Snyder	10304	12	1034.34
1	Howard Snyder	10307	2	15679.02
1	Howard Snyder	10322	17	9305.18
2	Yoshi Lattimer	10262	1	10556.22
2	Yoshi Lattimer	10269	7	124.56
2	Yoshi Lattimer	10278	8	105938.44
2	Yoshi Lattimer	10304	12	1034.34
2	Yoshi Lattimer	10307	2	15679.02
2	Yoshi Lattimer	10322	17	9305.18
3	John Steel	10262	1	10556.22
3	John Steel	10269	7	124.56
3	John Steel	10278	8	105938.44
3	John Steel	10304	12	1034.34
3	John Steel	10307	2	15679.02
3	John Steel	10322	17	9305.18
4	Jaime Yorres	10262	1	10556.22
4	Jaime Yorres	10269	7	124.56
4	Jaime Yorres	10278	8	105938.44
4	Jaime Yorres	10304	12	1034.34
4	Jaime Yorres	10307	2	15679.02
4	Jaime Yorres	10322	17	9305.18
5	Fran Wilson	10262	1	10556.22
5	Fran Wilson	10269	7	124.56
5	Fran Wilson	10278	8	105938.44
5	Fran Wilson	10304	12	1034.34
5	Fran Wilson	10307	2	15679.02
5	Fran Wilson	10322	17	9305.18
6	Rene Phillips	10262	1	10556.22
6	Rene Phillips	10269	7	124.56
6	Rene Phillips	10278	8	105938.44
6	Rene Phillips	10304	12	1034.34
6	Rene Phillips	10307	2	15679.02
6	Rene Phillips	10322	17	9305.18

## Natural Joins

Useless?

# Natural Join

---

Produce a new relation that combines matching tuples on a common attribute.

Table C

CustomerID	CustomerName
1	Howard Snyder
2	Yoshi Lattimer
3	John Steel
4	Jaime Yorres
5	Fran Wilson
6	Rene Phillips

Table O

OrderID	CustomerID	OrderTotal
10262	1	10556.22
10269	7	124.56
10278	8	105938.44
10304	12	1034.34
10307	2	15679.02
10322	17	9305.18

# *Natural Join*

---

C <join> O

CustomerID	CustomerName	OrderID	CustomerID	OrderTotal
1	Howard Snyder	10262	1	10556.22
2	Yoshi Lattimer	10307	2	15679.02

Produce a new relation that combines matching tuples on a condition.

(quite uncommon)

If  $R$  is a relational algebra expression,  
 $R = 0$  is a constraint that says  $R$  must be empty

If  $R$  and  $S$  are relational algebra expressions,  
 $R <\text{constrains}> S$  says that every tuple in  $R$  is also in  $S$