Introduction to Database Systems
CSE 444

Lecture 02: SQL

September 28, 2007

1

Administrivia

- Homework 1 is out. Due: Fri., Oct. 5
- Did you login on IISQLSRV?
- Did you change your password?
- Did you read today's reading assignment?
  - (Do you remember what it was?)
- Project 0: Who's your partner?
  - Due Wednesday; posted by this weekend

2

Outline

- Data in SQL
- Simple Queries in SQL (6.1)
- Queries with more than one relation (6.2)

3

SQL Introduction

Standard language for querying and manipulating data

Structured **Q**uery **L**anguage

Many standards out there:
- ANSI SQL, SQL92 (a.k.a. SQL2), SQL99 (a.k.a. SQL3), ….
- Vendors support various subsets: watch for fun discussions in class!

4

SQL

- Data Definition Language (DDL)
  - Create/alter/delete tables and their attributes
  - Following lectures...
- Data Manipulation Language (DML)
  - Query one or more tables – discussed next !
  - Insert/delete/modify tuples in tables

5

Tables in SQL

Table name

Attribute names

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Tuples or rows

6

1

Tables Explained

- The *schema* of a table is the table name and its attributes:

Product(PName, Price, Category, Manfacturer)

- A *key* is an attribute whose values are unique; we underline a key

Product(PName, Price, Category, Manfacturer)

7

Data Types in SQL

- Atomic types:
  - Characters: CHAR(20), VARCHAR(50)
  - Numbers: INT, BIGINT, SMALLINT, FLOAT
  - Others: MONEY, DATETIME, …

- Every attribute must have an atomic type
  - Hence tables are flat
  - Why ?

8

Tables Explained

- A tuple = a record
  - Restriction: all attributes are of atomic type

- A table = a set of tuples
  - Like a list…
  - …but it is unorderd:
    no **first()**, no **next()**, no **last()**.

9

SQL Query

Basic form: (plus many many more bells and whistles)

SELECT <attributes>
FROM   <one or more relations>
WHERE   <conditions>

10

Simple SQL Query

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

SELECT *
FROM   Product
WHERE   category='Gadgets'

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks

"selection"

11

Simple SQL Query

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

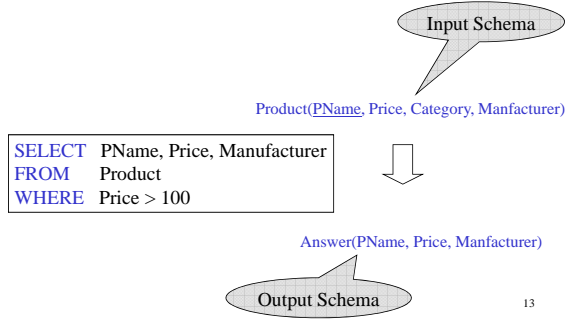SELECT PName, Price, Manufacturer
FROM   Product
WHERE   Price > 100

PName	Price	Manufacturer
SingleTouch	$149.99	Canon
MultiTouch	$203.99	Hitachi

"selection" and "projection"

12

2

Notation

Input Schema

Product(PName, Price, Category, Manfacturer)

```
SELECT    PName, Price, Manufacturer
FROM      Product
WHERE     Price > 100
```

Answer(PName, Price, Manfacturer)

Output Schema

13

Details

- Case insensitive:
  - Same: SELECT Select select
  - Same: Product   product
  - Different: 'Seattle' 'seattle'

- Constants:
  - 'abc' - yes
  - "abc" - no

14

The **LIKE** operator

```
SELECT    *
FROM      Products
WHERE     PName LIKE '%gizmo%'
```

- s **LIKE** p: pattern matching on strings
- p may contain two special symbols:
  - % = any sequence of characters
  - _ = any single character

15

Eliminating Duplicates

```
SELECT  DISTINCT category
FROM    Product
```

Category
Gadgets
Photography
Household

Compare to:

```
SELECT   category
FROM     Product
```

Category
Gadgets
Gadgets
Photography
Household

16

Ordering the Results

```
SELECT   pname, price, manufacturer
FROM     Product
WHERE    category='gizmo' AND price > 50
ORDER BY price, pname
```

Ties are broken by the second attribute on the ORDER BY list, etc.

Ordering is ascending, unless you specify the DESC keyword.

17

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

```
SELECT  DISTINCT category
FROM    Product
ORDER BY category
```
?

```
SELECT  Category
FROM    Product
ORDER BY  PName
```
?

```
SELECT  DISTINCT category
FROM    Product
ORDER BY PName
```
?

18

3

Keys and Foreign Keys

Company

CName	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

Key

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Foreign key

19

Joins

Product (pname, price, category, manufacturer)
Company (cname, stockPrice, country)

Find all products under $200 manufactured in Japan;
return their names and prices.

Join between Product and Company

```
SELECT   PName, Price
FROM     Product, Company
WHERE    Manufacturer=CName AND Country='Japan'
         AND Price <= 200
```

20

Joins

Product

PName	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

Cname	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
SELECT   PName, Price
FROM     Product, Company
WHERE    Manufacturer=CName AND Country='Japan'
         AND Price <= 200
```

PName	Price
SingleTouch	$149.99

21

More Joins

Product (pname, price, category, manufacturer)
Company (cname, stockPrice, country)

Find all Chinese companies that manufacture products
both in the 'electronic' and 'toy' categories

```
SELECT   cname

FROM

WHERE
```

22

A Subtlety about Joins

Product (pname, price, category, manufacturer)
Company (cname, stockPrice, country)

Find all countries that manufacture some product in the
'Gadgets' category.

```
SELECT   Country
FROM     Product, Company
WHERE    Manufacturer=CName AND Category='Gadgets'
```

23

A Subtlety about Joins

Product

Name	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

Cname	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
SELECT   Country
FROM     Product, Company
WHERE    Manufacturer=CName AND Category='Gadgets'
```

What is the problem ?

Country
??
??

24

A Subtlety about Joins

Product

Name	Price	Category	Manufacturer
Gizmo	$19.99	Gadgets	GizmoWorks
Powergizmo	$29.99	Gadgets	GizmoWorks
SingleTouch	$149.99	Photography	Canon
MultiTouch	$203.99	Household	Hitachi

Company

Cname	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
SELECT   Country
FROM     Product, Company
WHERE    Manufacturer=CName AND Category='Gadgets'
```

Country
USA
USA

Duplicates !
What's the
solution ?

25

Tuple Variables

Person(pname, address, worksfor)
Company(cname, address)

```
SELECT   DISTINCT pname, address
FROM     Person, Company
WHERE    worksfor = cname
```

Which
address ?

```
SELECT   DISTINCT Person.pname, Company.address
FROM     Person, Company
WHERE    Person.worksfor = Company.cname
```

```
SELECT   DISTINCT x.pname, y.address
FROM     Person AS x, Company AS y
WHERE    x.worksfor = y.cname
```

26

Meaning (Semantics) of SQL Queries

```
SELECT a₁, a₂, …, aₖ
FROM    R₁ AS x₁, R₂ AS x₂, …, Rₙ AS xₙ
WHERE  Conditions
```

```
Answer = {}
for x₁ in R₁ do
    for x₂ in R₂ do
        …..
            for xₙ in Rₙ do
                if Conditions
                    then Answer = Answer ∪ {(a₁,…,aₖ)}
return Answer
```

27

An Unintuitive Query

```
SELECT  DISTINCT R.A
FROM    R, S, T
WHERE   R.A=S.A   OR   R.A=T.A
```

What does it compute ?

Computes R ∩ (S ∪ T)   But what if S = φ ?

28

Subqueries Returning Relations

Company(name, city)
Product(pname, maker)
Purchase(id, product, buyer)

Return cities where one can find companies that manufacture products bought by Joe Blow

```
SELECT  Company.city
FROM    Company
WHERE  Company.name  IN
            (SELECT Product.maker
             FROM   Purchase , Product
             WHERE Product.pname=Purchase.product
                AND Purchase .buyer = 'Joe Blow');
```

29

Subqueries Returning Relations

Is it equivalent to this ?

```
SELECT  Company.city
FROM    Company, Product, Purchase
WHERE   Company.name= Product.maker
        AND  Product.pname  = Purchase.product
        AND  Purchase.buyer = 'Joe Blow'
```

Beware of duplicates !

30

Removing Duplicates

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

```
SELECT DISTINCT Company.city
FROM      Company, Product, Purchase
WHERE   Company.name= Product.maker
        AND  Product.pname  = Purchase.product
        AND  Purchase.buyer = 'Joe Blow'
```
Now they are equivalent

31

Subqueries Returning Relations

You can also use: s > ALL R
                    s > ANY R
                    EXISTS R

Product (pname, price, category, maker)
Find products that are more expensive than all those produced
By "Gizmo-Works"

```
SELECT  name
FROM     Product
WHERE  price > ALL (SELECT price
                    FROM    Product
                    WHERE  maker='Gizmo-Works')
```
32

Question for Database Fans and their Friends

- Can we express this query as a single SELECT-FROM-WHERE query, without subqueries ?

33

Monotone Queries

- A query Q is monotone if:
  - Whenever we add tuples to one or more of the tables…
  - … the answer to the query cannot contain fewer tuples

- Fact: all SFW (select-from-where) queries are monotone

- Fact: A query with ALL is not monotone

- Consequence: we cannot rewrite an ALL query into a SFW

34

Correlated Queries

Movie (title, year, director, length)
Find movies whose title appears more than once.

correlation

```
SELECT DISTINCT title
FROM   Movie AS x
WHERE  year <> ANY
              (SELECT  year
               FROM    Movie
               WHERE  title = x.title);
```

Note (1) scope of variables (2) this can still be expressed as single SFW

35

Complex Correlated Query

Product (pname, price, category, maker, year)
- Find products (and their manufacturers) that are more expensive than all products made by the same manufacturer before 1972

```
SELECT DISTINCT  pname, maker
FROM     Product AS x
WHERE  price > ALL  (SELECT  price
                     FROM    Product AS y
                     WHERE  x.maker = y.maker AND y.year < 1972);
```

Very powerful ! Also much harder to optimize.

36