Introduction to Data Management
CSE 344

Lecture 3: SQL Basics

Announcements

- WQ1 due on Tuesday
  - Any issues?
- WQ2 will be out tomorrow
  - SQL basics and aggregates
  - Due in 1 week
- HW1 due on Wednesday

Review

- Relational data model
- SQL for manipulating relational data
  - Create tables
  - Retrieve records from tables
  - Declare keys and foreign keys

Review

- Tables are NOT ordered
  - they are sets or multisets (bags)
  - arity: # of attributes in a relation
  - cardinality: # of records in a relation
- Tables are FLAT
  - No nested attributes
- Tables DO NOT prescribe how they are implemented / stored on disk
  - This is called **physical data independence**

Today

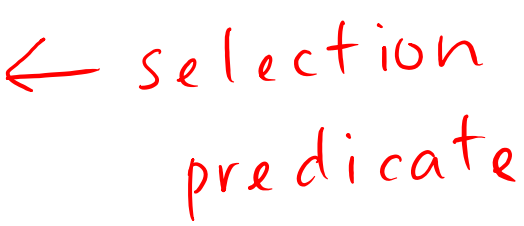
- SQL Basics
  - Selection
  - Projection
  - Ordering and distinct
  - Joins

SQL

- SQL
  - **S**tructured **Q**uery **L**anguage
  - Most widely used language to query relational data
  - One of the many languages for querying relational data
  - A **declarative** programming language

Selections in SQL

```
SELECT *
FROM   Product
WHERE  price > 100.0
```



Demo

Joins in SQL

```
SELECT pname, price
FROM   Product, Company
WHERE  manufacturer=cname AND
       country='Japan' AND price < 150
```

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

What does this query do?

Joins in SQL

```
SELECT pname, price
FROM   Product, Company
WHERE  manufacturer=cname AND
       country='Japan' AND price < 150
```

join predicate

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

Retrieve all Japanese products that cost < $150

Joins in SQL

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

pname	price	manufacturer
MultiTouch	199.99	Canon
SingleTouch	49.99	Canon
SuperGizmo	250.00	GizmoWorks

cname	country
GizmoWorks	USA
Canon	Japan

```
SELECT pname, price
FROM   Product, Company
WHERE  manufacturer=cname AND
       country='Japan' AND price < 150
```

Joins in SQL

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

pname	price	manufacturer
MultiTouch	199.99	Canon
SingleTouch	49.99	Canon
SuperGizmo	250.00	GizmoWorks

cname	country
GizmoWorks	USA
Canon	Japan

Retrieve all American companies that manufacture “gadget” products

Joins in SQL

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

pname	price	manufacturer
MultiTouch	199.99	Canon
SingleTouch	49.99	Canon
SuperGizmo	250.00	GizmoWorks

cname	country
GizmoWorks	USA
Canon	Japan

```
SELECT DISTINCT cname
FROM   Product, Company
WHERE  country='USA' AND category = 'gadget'
       AND manufacturer = cname
```

Joins in SQL

- This query is called an inner join
  - Each row in the result **must come from both tables in the join**
  - In our example, notice that companies that didn't make any "gadgets" did not show up
  - What if we want to retain those in the results as well?

Outer Joins

```
Employee(id, name)
Sales(employeeID, productID)
```

id	name
1	'Joe'
2	'Jack
3	'Jill'

employeeID	productID
1	344
1	355
2	544

Retrieve employees and their sales

```
SELECT *
FROM   Employee E, Sales S
WHERE  E.id = S.employeeID
```

Outer Joins

```
Employee(id, name)
Sales(employeeID, productID)
```

id	name
1	'Joe'
2	'Jack
3	'Jill'

employeeID	productID
1	344
1	355
2	544

Retrieve employees and their sales

```
SELECT *
FROM   Employee E INNER JOIN Sales S
ON     E.id = S.employeeID
```

Outer Joins

```
Employee(id, name)
Sales(employeeID, productID)
```

id	name
1	'Joe'
2	'Jack
3	'Jill'

employeeID	productID
1	344
1	355
2	544

Retrieve employees and their sales

```
SELECT *
FROM   Employee E LEFT OUTER JOIN Sales S
ON     E.id = S.employeeID
```