Practice and Applications of Data Management
CMPSCI 345

Lecture 03: Basic SQL
Today: basic SQL

- Chapters 2.3, 6.1, 6.2, 6.5

- Quiz 2 is up on Gradiance
  - Due before next lecture

- In-class practice next Monday
  - More self-driven
Structured Query Language (SQL)

- Data Definition Language (DDL)
  - Create/alter/delete tables and their attributes

- Data Manipulation Language (DML)
  - Query one or more tables
  - Insert/delete/modify tuples in tables
### Tables in SQL

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
<th>Category</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>$19.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>Powergizmo</td>
<td>$29.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>SingleTouch</td>
<td>$149.99</td>
<td>Photography</td>
<td>Canon</td>
</tr>
<tr>
<td>MultiTouch</td>
<td>$203.99</td>
<td>Household</td>
<td>Hitachi</td>
</tr>
</tbody>
</table>
Simple SQL query

```
SELECT * FROM Product
```

<table>
<thead>
<tr>
<th>PName</th>
<th>Price</th>
<th>Category</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>$19.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>Powergizmo</td>
<td>$29.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>SingleTouch</td>
<td>$149.99</td>
<td>Photography</td>
<td>Canon</td>
</tr>
<tr>
<td>MultiTouch</td>
<td>$203.99</td>
<td>Household</td>
<td>Hitachi</td>
</tr>
</tbody>
</table>
Simple SQL query

```
SELECT * FROM Product
WHERE category='Gadgets'
```

### Product

<table>
<thead>
<tr>
<th>PName</th>
<th>Price</th>
<th>Category</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>$19.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>Powergizmo</td>
<td>$29.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>SingleTouch</td>
<td>$149.99</td>
<td>Photography</td>
<td>Canon</td>
</tr>
<tr>
<td>MultiTouch</td>
<td>$203.99</td>
<td>Household</td>
<td>Hitachi</td>
</tr>
</tbody>
</table>

Selection predicate
All gadgets cheaper than $20?

<table>
<thead>
<tr>
<th>PName</th>
<th>Price</th>
<th>Category</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>$19.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>Powergizmo</td>
<td>$29.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>SingleTouch</td>
<td>$149.99</td>
<td>Photography</td>
<td>Canon</td>
</tr>
<tr>
<td>MultiTouch</td>
<td>$203.99</td>
<td>Household</td>
<td>Hitachi</td>
</tr>
</tbody>
</table>

**SELECT** * FROM **Product** WHERE category='Gadgets' AND price < 20

<table>
<thead>
<tr>
<th>PName</th>
<th>Price</th>
<th>Category</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>$19.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
</tbody>
</table>
**Simple SQL query**

```
SELECT PName, price, manufacturer
FROM Product
WHERE price > 100
```

<table>
<thead>
<tr>
<th>PName</th>
<th>Price</th>
<th>Category</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>SingleTouch</td>
<td>$149.99</td>
<td>Photography</td>
<td>Canon</td>
</tr>
<tr>
<td>MultiTouch</td>
<td>$203.99</td>
<td>Household</td>
<td>Hitachi</td>
</tr>
</tbody>
</table>

**Projection attributes**

**Selection predicate**
Basic SQL query structure

```
SELECT attribute 1, attribute 2, ...
FROM table1, table 2, ...
WHERE selection predicates
```
## Example (1, 2)

### World

<table>
<thead>
<tr>
<th>name</th>
<th>continent</th>
<th>area</th>
<th>population</th>
<th>gdp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>652230</td>
<td>25500100</td>
<td>203430000000</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>28748</td>
<td>2831741</td>
<td>129600000000</td>
</tr>
<tr>
<td>Algeria</td>
<td>Africa</td>
<td>2381741</td>
<td>37100000</td>
<td>188681000000</td>
</tr>
<tr>
<td>Andorra</td>
<td>Europe</td>
<td>468</td>
<td>78115</td>
<td>3712000000</td>
</tr>
<tr>
<td>Angola</td>
<td>Africa</td>
<td>1246700</td>
<td>20609294</td>
<td>100990000000</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Example

All countries in Asia and their population

```
SELECT name, population
FROM world
WHERE continent = 'Asia'
```

All European countries with population less than 15 million

```
SELECT name
FROM world
WHERE continent = 'Europe'
AND population < 15000000
```
Boolean operators

- AND, OR, NOT

All countries not in Europe with population less than 15 million

```sql
SELECT name
FROM world
WHERE NOT (continent = 'Europe')
AND population < 15000000
```

```sql
SELECT name
FROM world
WHERE continent <> 'Europe'
AND population < 15000000
```
Example

Find the population of all countries whose name starts with the letter ‘F’

```
SELECT name, population
FROM world
WHERE name LIKE 'F%'
```
Predicate comparisons

- Equal to: =
- Not equal to: <>
- Less than: <
- Less than or equal to: <=
- Greater than: >
- Greater than or equal to: >=
- In inclusive range: BETWEEN
- Search for pattern: LIKE
- Multiple possible values: IN
Example

All countries in Asia or Europe and their population

```
SELECT name, population
FROM world
WHERE continent IN ('Asia', 'Europe')
```

All countries with population between 15 and 30 million

```
SELECT name
FROM world
WHERE population BETWEEN 15000000 AND 30000000
```
## Example (3)

### World

<table>
<thead>
<tr>
<th>name</th>
<th>continent</th>
<th>area</th>
<th>population</th>
<th>gdp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>652230</td>
<td>25500100</td>
<td>203430000000</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>28748</td>
<td>2831741</td>
<td>129600000000</td>
</tr>
<tr>
<td>Algeria</td>
<td>Africa</td>
<td>2381741</td>
<td>37100000</td>
<td>188681000000</td>
</tr>
<tr>
<td>Andorra</td>
<td>Europe</td>
<td>468</td>
<td>78115</td>
<td>37120000000</td>
</tr>
<tr>
<td>Angola</td>
<td>Africa</td>
<td>1246700</td>
<td>20609294</td>
<td>1009900000000</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Example

Display the names of all countries with density more than 150 people per km$^2$

```
SELECT  name
FROM    world
WHERE   population/area > 150
```
## Example (4)

### World

<table>
<thead>
<tr>
<th>name</th>
<th>continent</th>
<th>area</th>
<th>population</th>
<th>gdp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afghanistan</td>
<td>Asia</td>
<td>652230</td>
<td>25500100</td>
<td>203430000000</td>
</tr>
<tr>
<td>Albania</td>
<td>Europe</td>
<td>28748</td>
<td>2831741</td>
<td>129600000000</td>
</tr>
<tr>
<td>Algeria</td>
<td>Africa</td>
<td>2381741</td>
<td>37100000</td>
<td>188681000000</td>
</tr>
<tr>
<td>Andorra</td>
<td>Europe</td>
<td>468</td>
<td>78115</td>
<td>3712000000</td>
</tr>
<tr>
<td>Angola</td>
<td>Africa</td>
<td>1246700</td>
<td>20609294</td>
<td>100990000000</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Example

Find the population density of each country in the world

```
SELECT name, population/area
FROM world
```

What will be the attribute name in the result?

```
SELECT name, population/area AS density
FROM world
```

```
SELECT name AS country,
    population/area AS density
FROM world
```
Eliminating Duplicates

### Product

<table>
<thead>
<tr>
<th>PName</th>
<th>Price</th>
<th>Category</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>$19.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>PowerGizmo</td>
<td>$29.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>SingleTouch</td>
<td>$149.99</td>
<td>Photography</td>
<td>Canon</td>
</tr>
<tr>
<td>MultiTouch</td>
<td>$203.99</td>
<td>Household</td>
<td>Hitachi</td>
</tr>
</tbody>
</table>

SELECT \textbf{category} FROM Product

### Set vs. Bag semantics

SELECT \textbf{DISTINCT} category FROM Product
Ordering the Results

```sql
SELECT pName, price, manufacturer
FROM Product
WHERE category='Gadgets'
    and price > 10
ORDER BY price, pName
```

- Ties in price attribute broken by pname attribute
- Ordering is ascending by default. Descending:

```sql
... ORDER BY price, pName DESC
```
<table>
<thead>
<tr>
<th>PName</th>
<th>Price</th>
<th>Category</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>$19.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>Powergizmo</td>
<td>$29.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>SingleTouch</td>
<td>$149.99</td>
<td>Photography</td>
<td>Canon</td>
</tr>
<tr>
<td>MultiTouch</td>
<td>$203.99</td>
<td>Household</td>
<td>Hitachi</td>
</tr>
</tbody>
</table>

**SELECT** DISTINCT category  
**FROM** Product  
**ORDER BY** category

**SELECT** category  
**FROM** Product  
**ORDER BY** pName

**SELECT** DISTINCT category  
**FROM** Product  
**ORDER BY** pName
SELECT category FROM Product ORDER BY category

Syntax error*

SELECT DISTINCT category FROM Product ORDER BY pName

SELECT category FROM Product ORDER BY pName

Product

<table>
<thead>
<tr>
<th>PName</th>
<th>Price</th>
<th>Category</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>$19.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>Powergizmo</td>
<td>$29.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>SingleTouch</td>
<td>$149.99</td>
<td>Photography</td>
<td>Canon</td>
</tr>
<tr>
<td>MultiTouch</td>
<td>$203.99</td>
<td>Household</td>
<td>Hitachi</td>
</tr>
</tbody>
</table>
NULLS in SQL

Whenever we don’t have a value, we can put a NULL

Can mean many things:
- Value does not exist
- Value exists but is unknown
- Value not applicable
- Etc.

The schema specifies for each attribute if it can be NULL or not

How does SQL cope with tables that have NULLs?
Null Values

- If \( x = \text{NULL} \) then
  - Arithmetic operations produce \text{NULL}. E.g: \( 4 * (3-x)/7 \)
  - Boolean conditions are also \text{NULL}. E.g: \( x='Joe' \)

- In SQL there are three boolean values:
  FALSE, TRUE, UNKNOWN

- Reasoning:

  \[
  \begin{align*}
  \text{FALSE} &= 0 \\
  \text{TRUE} &= 1 \\
  \text{UNKNOWN} &= 0.5 \\
  \text{x AND y} &= \min(x,y) \\
  \text{x OR y} &= \max(x,y) \\
  \text{NOT x} &= (1 - x)
  \end{align*}
  \]
SELECT * 
FROM Person 
WHERE (age < 25) and 
     (height > 6 or weight > 190)

<table>
<thead>
<tr>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>NULL</td>
<td>200</td>
</tr>
<tr>
<td>NULL</td>
<td>6.5</td>
<td>170</td>
</tr>
</tbody>
</table>

Rule in SQL: include only tuples that yield TRUE

SELECT * 
FROM Person 
WHERE age < 25 or age >= 25

Unexpected behavior

SELECT * 
FROM Person 
WHERE age < 25 or age >= 25 or age IS NULL

Test NULL explicitly
Example: NULL

For what values of x, y, and z, including NULL, does the boolean expression

\[ x \leq 3 \text{ AND } (y \geq 1 \text{ OR } z = 5) \]

have the truth value TRUE? Identify one of those values from the list below.

- x = 2, y = NULL, z = 5  ✔
- x = NULL, y = 2, z = 4  ✖
- x = 3, y = NULL, z = 4  ✖
- x = 3, y = NULL, z = NULL  ✔
- x = 3, y = 2, z = NULL  ✔
Joins

Product (pName, price, category, manufacturer)
Company (cName, stockPrice, country)

Q: Find all products under $200 manufactured in Japan; return their names and prices!

```
SELECT pName, price
FROM Product, Company
WHERE manufacturer=cName
  and country='Japan'
  and price <= 200
```
## Joins

### Product

<table>
<thead>
<tr>
<th>PName</th>
<th>Price</th>
<th>Category</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>$19.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>Powergizmo</td>
<td>$29.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>SingleTouch</td>
<td>$149.99</td>
<td>Photography</td>
<td>Canon</td>
</tr>
<tr>
<td>MultiTouch</td>
<td>$203.99</td>
<td>Household</td>
<td>Hitachi</td>
</tr>
</tbody>
</table>

### Company

<table>
<thead>
<tr>
<th>CName</th>
<th>StockPrice</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>GizmoWorks</td>
<td>25</td>
<td>USA</td>
</tr>
<tr>
<td>Canon</td>
<td>65</td>
<td>Japan</td>
</tr>
<tr>
<td>Hitachi</td>
<td>15</td>
<td>Japan</td>
</tr>
</tbody>
</table>

### SQL Query

```sql
SELECT pName, price
FROM Product, Company
WHERE manufacturer=cName
    AND country='Japan'
    AND price <= 200
```

<table>
<thead>
<tr>
<th>PName</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>SingleTouch</td>
<td>$149.99</td>
</tr>
</tbody>
</table>
Tuple Variables

Person (pName, address, works_for)
Company (cName, address)

SELECT DISTINCT pName, address
FROM Person, Company
WHERE works_for = cName

SELECT DISTINCT Person.pName, Company.address
FROM Person, Company
WHERE Person.works_for = Company.cName

SELECT DISTINCT X.pName, Y.address
FROM Person as X, Company as Y
WHERE X.works_for = Y.cName

"as" is optional
### Example (5)

**Employees**

<table>
<thead>
<tr>
<th>empID</th>
<th>name</th>
<th>phone</th>
<th>managerID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John</td>
<td>555-1234</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Alice</td>
<td>555-4321</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Peter</td>
<td>555-2314</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Cecilia</td>
<td>555-3241</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>James</td>
<td>555-4231</td>
<td>NULL</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Projects**

<table>
<thead>
<tr>
<th>empID</th>
<th>project</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Web archive</td>
</tr>
<tr>
<td>1</td>
<td>Phone app</td>
</tr>
<tr>
<td>3</td>
<td>Web archive</td>
</tr>
<tr>
<td>2</td>
<td>Tools</td>
</tr>
<tr>
<td>3</td>
<td>Game design</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Example

List all employee names and the projects that each is involved in

```
SELECT  e.name, p.project
FROM    Employees as e, Projects as p
WHERE   e.empID = p.empID
```