Practice and Applications of Data Management

CMPSCI 345

Lecture 06: Aggregates
Reminders

- Quiz 3 due on Wednesday
- Assignment 1 is due in 9 days
- You will start working on the group project next week.
  - Group assignments on Moodle
Postgres

- Useful commands:
  - \h : help on SQL commands
  - \? : help on psql commands
  - \i : execute commands from file
  - \l : show databases
  - \d : show tables
  - \c : connect to a database

\pset null NULL
Practice

- Compute the average price of all products
  
  ```sql
  SELECT avg(price) 
  FROM Product
  ```

- Compute the average product price for each category
  
  ```sql
  SELECT category, avg(price) 
  FROM Product 
  GROUP BY category
  ```
Practice

Compute the average price of products cheaper than $150, for each category

```
SELECT category, avg(price)
FROM Product
WHERE price < 150
GROUP BY category
```
Practice

- Compute the average price of products cheaper than $150, for each category, but only keep the categories where the maximum price is greater than $80.

```
SELECT category, avg(price) 
FROM Product 
WHERE price < 150 
GROUP BY category 
HAVING max(price) > 80
```
Practice

- Compute the average price of products cheaper than $150, for each category, but only keep the categories that start with the letter ‘G’.

```sql
SELECT category, avg(price) 
FROM Product 
WHERE price < 150 
GROUP BY category 
HAVING category LIKE 'G%' 
```

```sql
SELECT category, avg(price) 
FROM Product 
WHERE price < 150 
AND category LIKE 'G%' 
GROUP BY category 
```
HAVING vs WHERE

- **WHERE**
  - Filters each input row based on the condition
  - May include any field

- **HAVING**
  - Filters the output based on the condition
  - May only include aggregate or group by attributes
Practice

What does this query compute?

```
SELECT E.name, count(*)
FROM Employees AS E, Projects AS P
WHERE E.empID = P.empID
GROUP BY E.name
```

What’s wrong?
Practice

What about this query?

```
SELECT  E.name, count(*)
FROM    Employees AS E
        LEFT OUTER JOIN Projects AS P
        ON  E.empID = P.empID
GROUP BY E.name
```

What’s wrong?

Explanation:

```
SELECT  *
FROM    Employees AS E
        LEFT OUTER JOIN Projects AS P
        ON  E.empID = P.empID
```
Practice

How do we fix it?

```
SELECT E.name, count(*)
FROM Employees AS E
  LEFT OUTER JOIN Projects AS P
  ON E.empID = P.empID
GROUP BY E.name
```
Practice

How do we fix it?

```sql
SELECT E.name, count(project) 
FROM Employees AS E 
LEFT OUTER JOIN Projects AS P 
ON E.empID = P.empID 
GROUP BY E.name
```
Example (from last week)

Product(name, category)
Purchase(prodName, month, store)

- Compute, for each product in the ‘Gadgets’ category, the total number of sales

```
SELECT Product.name, count(*)
FROM Product, Purchase
WHERE Product.name = Purchase.prodName
    and Product.category = 'Gadgets'
GROUP BY Product.name
```

What’s wrong?
Example

Product(name, category)
Purchase(prodName, month, store)

- Compute, for each product in the ‘Gadgets’ category, the total number of sales

```
SELECT Product.name, count(*)
FROM Product LEFT OUTER JOIN Purchase ON Product.name = Purchase.prodName
WHERE Product.category = 'Gadgets'
GROUP BY Product.name
```

What’s wrong?
Example

Compute, for each product in the ‘Gadgets’ category, the total number of sales

We need to use the attribute to get the correct 0 count.

```
SELECT Product.name, count(month)
FROM Product LEFT OUTER JOIN Purchase ON Product.name = Purchase.prodName
WHERE Product.category = 'Gadgets'
GROUP BY Product.name
```
Witnesses

- What is the price of the cheapest product?

```
SELECT min(price) 
FROM Product
```

- What is the cheapest product?

```
SELECT pname
FROM Product
WHERE price = (SELECT min(price) 
               FROM Product)
```
Witnesses

- Which company makes the most products?

```sql
SELECT manufacturer
FROM (SELECT manufacturer, count(*) AS total
      FROM product
      GROUP BY manufacturer) AS A
WHERE total = (SELECT max(allProd)
               FROM (SELECT manufacturer,
                       count(*) AS total
                       FROM product
                       GROUP BY manufacturer) AS B)
```