CS 327E Lecture 10

Shirley Cohen

October 26, 2016

Homework for Today

- Chapter 8 from the Learning SQL book
- Exercises at the end of Chapter 8

Quiz Question 1

```
mysql> select * from Employee;
+--------+----------+-----------+-----------------+
| emp_id | fname    | lname     | superior_emp_id |
+--------+----------+-----------+-----------------+
|      1 | Michael  | Smith     |            NULL |
|      2 | Susan    | Barker    |               1 |
|      3 | Robert   | Tyler     |               1 |
|      4 | Susan    | Hawthorne |               3 |
|      5 | John     | Gooding   |               4 |
+--------+----------+-----------+-----------------+
```

What **value** does the following query return?

```
SELECT COUNT(*) FROM Employee;
```

A. 1 B. 3 C. 5 D. 4

Quiz Question 2

```
mysql> select * from Employee;
+--------+----------+----------+-----------------+
| emp_id | fname    | lname    | superior_emp_id |
+--------+----------+----------+-----------------+
|      1 | Michael  | Smith    |            NULL |
|      2 | Susan    | Barker   |               1 |
|      3 | Robert   | Tyler    |               1 |
|      4 | Susan    | Hawthorne|               3 |
|      5 | John     | Gooding  |               4 |
+--------+----------+----------+-----------------+
```

What **value** does the following query?

```
SELECT COUNT(superior_emp_id)
FROM Employee;
```

A. 1 B. 2 C. 3 D. 4

Quiz Question 3

```
mysql> select * from Employee;
+--------+----------+----------+-----------------+
| emp_id | fname    | lname    | superior_emp_id |
+--------+----------+----------+-----------------+
|      1 | Michael  | Smith    |            NULL |
|      2 | Susan    | Barker   |               1 |
|      3 | Robert   | Tyler    |               1 |
|      4 | Susan    | Hawthorne|               3 |
|      5 | John     | Gooding  |               4 |
+--------+----------+----------+-----------------+
```

How many **groups** does the following query return?

```
SELECT superior_emp_id, COUNT(*)
FROM Employee
GROUP BY superior_emp_id;
```

A. 0 B. 3 C. 4 D. 5

Quiz Question 4

```
mysql> select * from Employee;
+--------+----------+----------+-----------------+
| emp_id | fname    | lname    | superior_emp_id |
+--------+----------+----------+-----------------+
|      1 | Michael  | Smith    |            NULL |
|      2 | Susan    | Barker   |               1 |
|      3 | Robert   | Tyler    |               1 |
|      4 | Susan    | Hawthorne|               3 |
|      5 | John     | Gooding  |               4 |
+--------+----------+----------+-----------------+
```

What many **groups** does the following query return?

```
SELECT superior_emp_id, COUNT(*)
FROM Employee
WHERE COUNT(*) > 1
GROUP BY superior_emp_id;
```

A. 1 B. 3 C. 4 D. N/A. The query is syntactically incorrect

Quiz Question 5

Aggregate functions like `max()`, `min()`, etc. **cannot** be used without a `group by` clause.

A. True

B. False

Standard Aggregation Functions

- count
- sum
- avg
- max
- min

Count Examples

```
SELECT COUNT(*)

FROM Customer
```

****

```
SELECT COUNT(city)

FROM Customer
```

****

```
SELECT COUNT(DISTINCT city)

FROM Customer
```

cust_id	name	city	status
1	Paul Gore	Austin	Active
2	Chris Burns	Austin	Active
3	Jerry Hargrove	Dallas	Active
4	Lara Wells	Dallas	Active
5	Nick O'Neil	Austin	Inactive
6	Brad Fisher	NYC	Active
7	Kevin O'Connor	NYC	Active
8	Brian Zender		Inactive

Concept Question 1

Suppose we have a `Product` table as shown below. How can we find out the number of unique departments that have products or SKUs in this table?

A.
```
SELECT COUNT(Department)
FROM Product
```

B.
```
SELECT SUM(Department)
FROM Product
```

C.
```
SELECT COUNT(*)
FROM Product
```

D.
```
SELECT COUNT(DISTINCT Department)
FROM Product
```

E. None of the above

Product (SKU, SKU_Description, Department)

SELECT * FROM Product

SKU	SKU_Description	Department
100100	Std. Scuba Tank, Yellow	Water Sports
100200	Std. Scuba Tank, Magenta	Water Sports
101100	Dive Mask, Small Clear	Water Sports
101200	Dive Mask, Med Clear	NULL
201000	Half-dome Tent	Camping
202000	Half-dome Tent Vestibule	Camping
301000	Light Fly Climbing Harness	Climbing
302000	Locking carabiner, Oval	NULL

Group By Example

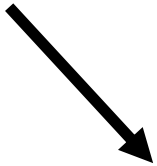
```
SELECT c.city, SUM(o.total_amount) AS total_sales
FROM Customer c INNER JOIN Orders o USING (cust_id)
GROUP BY c.city
ORDER BY c.city
```

cust_id	name	city
1	Paul Gore	Austin
2	Chris Burns	Austin
3	Jerry Hargrove	Dallas
4	Lara Wells	Dallas
5	Nick O'Neil	Austin
6	Brad Fisher	NYC
7	Kevin O'Connor	NYC
8	Brian Zender	

Customer

order_id	cust_id	order_date	total_amount
100	1	2016-10-01	5
200	1	2016-10-01	5
300	1	2016-10-19	5
400	2	2016-10-19	10
500	8	2016-10-20	10
600	7	2016-10-21	20
700	2	2016-10-22	20
800	8	2016-10-22	5

Orders



city	total_sales
NULL	15
Austin	45
NYC	20

Results

Note: Lost the Dallas group. Why?

Group By Example

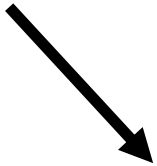
```
SELECT c.city, SUM(o.total_amount) AS total_sales
FROM Customer c LEFT OUTER JOIN Orders o USING (cust_id)
GROUP BY c.city
ORDER BY c.city
```

cust_id	name	city
1	Paul Gore	Austin
2	Chris Burns	Austin
3	Jerry Hargrove	Dallas
4	Lara Wells	Dallas
5	Nick O'Neil	Austin
6	Brad Fisher	NYC
7	Kevin O'Connor	NYC
8	Brian Zender	

Customer

order_id	cust_id	order_date	total_amount
100	1	2016-10-01	5
200	1	2016-10-01	5
300	1	2016-10-19	5
400	2	2016-10-19	10
500	8	2016-10-20	10
600	7	2016-10-21	20
700	2	2016-10-22	20
800	8	2016-10-22	5

Orders



city	total_sales
NULL	15
Austin	45
Dallas	NULL
NYC	20

Results

Note: Preserved the Dallas group

Group By Example

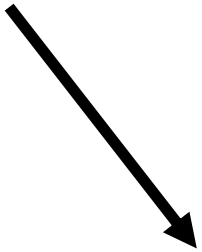
```
SELECT c.city, SUM(o.total_amount) AS total_sales
FROM Customer c LEFT OUTER JOIN Orders o USING (cust_id)
GROUP BY c.city
HAVING SUM(o.total_amount) > 20
ORDER BY c.city
```

cust_id	name	city
1	Paul Gore	Austin
2	Chris Burns	Austin
3	Jerry Hargrove	Dallas
4	Lara Wells	Dallas
5	Nick O'Neil	Austin
6	Brad Fisher	NYC
7	Kevin O'Connor	NYC
8	Brian Zender	

Customer

order_id	cust_id	order_date	total_amount
100	1	2016-10-01	5
200	1	2016-10-01	5
300	1	2016-10-19	5
400	2	2016-10-19	10
500	8	2016-10-20	10
600	7	2016-10-21	20
700	2	2016-10-22	20
800	8	2016-10-22	5

Orders



city	total_sales
Austin	45

Results

Note: HAVING filters out Dallas, NYC, NULL

Concept Question 2

Continuing with the same `Product` table, how can we retrieve the names of all the departments along with the number of unique products or SKUs that they sell? Assume that we want to includes null departments

A.
```
SELECT Department, COUNT(*)
FROM Product
GROUP BY Department
```

B.
```
SELECT Department, COUNT(*)
FROM Product
```

C.
```
SELECT Department,
COUNT(Department)
FROM Product
```

D.
```
SELECT Department,
COUNT(Department)
FROM Product
GROUP BY Department
```

E. None of the above

Product (SKU, SKU_Description, Department)

SELECT * FROM Product

SKU	SKU_Description	Department
100100	Std. Scuba Tank, Yellow	Water Sports
100200	Std. Scuba Tank, Magenta	Water Sports
101100	Dive Mask, Small Clear	Water Sports
101200	Dive Mask, Med Clear	NULL
201000	Half-dome Tent	Camping
202000	Half-dome Tent Vestibule	Camping
301000	Light Fly Climbing Harness	Climbing
302000	Locking carabiner, Oval	NULL

Concept Question 3

What's wrong with this query?

A.	SELECT c.city
B.	o.order_date
C.	SUM(o.total_amount)
D.	GROUP BY c.city
E.	None of the above

```
SELECT c.city, o.order_date, SUM(o.total_amount) AS total_sales
FROM Customer c LEFT OUTER JOIN Orders o USING (cust_id)
GROUP BY c.city
```

cust_id	name	city
1	Paul Gore	Austin
2	Chris Burns	Austin
3	Jerry Hargrove	Dallas
4	Lara Wells	Dallas
5	Nick O'Neil	Austin
6	Brad Fisher	NYC
7	Kevin O'Connor	NYC
8	Brian Zender	

Customer

order_id	cust_id	order_date	total_amount
100	1	2016-10-01	5
200	1	2016-10-01	5
300	1	2016-10-19	5
400	2	2016-10-19	10
500	8	2016-10-20	10
600	7	2016-10-21	20
700	2	2016-10-22	20
800	8	2016-10-22	5

Orders

Concept Question 4

We have a table of test results. A test has one or more steps and a step represents a record in the table. A test is considered complete when all its steps have a non-null `completion_date`. How can we find only the tests that are **complete**?

A.
```
select test_name
from Test_Results
group by test_name
having count(*) =
count(completion_date)
```

B.
```
select test_name
from Test_Results
where completion_date
is not null
group by test_name
```

C.
```
select test_step
from Test_Results
group by test_step
having count(*) =
count(completion_date)
```

D.
```
select test_step
from Test_Results
where completion_date
is not null
group by test_step
```

E. None of the above

Table definition:

```
create table Test_Results
(
   test_name CHAR(20) NOT NULL,
   test_step INTEGER NOT NULL,
   completion_date DATE,
   PRIMARY KEY (test_name, test_step)
)
```

Sample dataset:

test_name	test_step	completion_date
Math Skills	1	2016-02-01
Math Skills	2	2016-02-02
Math Skills	3	2016-02-03
Math Skills	4	NULL
Math Skills	5	NULL
Reading Skills	1	2016-02-07
Reading Skills	2	2016-02-08
Reading Skills	3	2016-02-08

Homework for Next Time

- Read chapter 14 from the Learning SQL book
- Exercises at the end of the assigned chapter