Aggregate Queries and Views

CS 327E
October 11, 2017
Announcements:
• Midterm: 2 weeks from today in **ETC 2.108**
• Next week: Lab 3
1) The SQL command for adding a new record to an existing table is:

A) ADD
B) PUT
C) INSERT
D) CREATE
2) Given the input tables shown, what will the following statement do?

```
INSERT INTO Employee2
SELECT * FROM Employee;
```

A) Copy 6 records from Employee to Employee2.

B) Copy 6 records from Employee2 to Employee.

C) Copy 0 records.
3) The SQL command for updating a record in a table is:

A) UPDATE
B) ALTER
C) MODIFY
D) CHANGE
4) The SQL command for deleting a record from a table is:

A) TRUNCATE
B) PURGE
C) DROP
D) DELETE
5) Given the input tables shown and the following statement, how many columns will the resulting view have?

CREATE VIEW Emp_Dept AS
SELECT e.*, d.depname
FROM Employee e
JOIN Department d
ON e.depid = d.depid;

A) 5
B) 6
C) 7
Semantics of COUNT

```
dev=> select count(*) from Employee;
  count
-----
   6
(1 row)
```

```
dev=> select count(depid) from Employee;
  count
-----
   5
(1 row)
```

```
dev=> select count(distinct depid) from Employee;
  count
-----
   3
(1 row)
```
What’s wrong with this query?

dev=> select depname, d.depid, count(*)
dev=> from Employee e full outer join Department d
dev=> on e.depid = d.depid
dev=> group by depname, d.depid;

depname | depid | count
-------- |------ |------
Sales    |  7   |  1   
Engineering |  8   |  1   
Executive |  5   |  2   
Research  |  6   |  2   
(5 rows)
This is what we want:

```sql
dev=> select depname, d.depid, count(empid)
    from Employee e full outer join Department d
    on e.depid = d.depid
    group by depname, d.depid;

+----------------+-----+-----|
| depname  | depid | count|
|----------+-------+-------|
| Sales    | 7     | 0     |
| Engineering | 8    | 1     |
| Executive | 5     | 2     |
| Research  | 6     | 2     |
+-----------+-------+-------|
| (5 rows)  |       |       |
```

### Employee

<table>
<thead>
<tr>
<th>empid</th>
<th>firstname</th>
<th>lastname</th>
<th>salary</th>
<th>depid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Michael</td>
<td>Dell</td>
<td>100</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Betty</td>
<td>Jennings</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Bill</td>
<td>Gates</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Kay</td>
<td>McNulty</td>
<td>300</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>Jim</td>
<td>Gray</td>
<td>500</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>Gordon</td>
<td>Moore</td>
<td>400</td>
<td>6</td>
</tr>
</tbody>
</table>

### Department

<table>
<thead>
<tr>
<th>depid</th>
<th>depname</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Executive</td>
</tr>
<tr>
<td>6</td>
<td>Research</td>
</tr>
<tr>
<td>7</td>
<td>Sales</td>
</tr>
<tr>
<td>8</td>
<td>Engineering</td>
</tr>
</tbody>
</table>
Practice Problem 1

Find Instacart’s core customers:

- customers who have placed at least 5 orders
- orders are no more than 7 days apart (use `days_since_prior_order`)
- orders contain at least 10 products (use `add_to_cart_order`)

Return `user_id` and number of orders placed by user.

Sort results by number of orders from highest to lowest.
Practice Problem 1

Find Instacart’s core customers:
• customers who have placed at least 5 orders
• orders are no more than 7 days apart
• orders contain at least 10 products

Return user_id and number of orders placed by user.

Sort results by number of orders, from highest to lowest.

Aside from SELECT, FROM, and ORDER BY, what clauses are needed to compute the answer?
A) GROUP BY, HAVING, WHERE, JOIN
B) GROUP BY, HAVING, JOIN
C) GROUP BY, JOIN
D) GROUP BY, WHERE, JOIN

Lab 2: Dimensional Schema

Notes:
- appalling titles: \( \leq 2.0 \)
- average titles: 2.1 - 7.9
- outstanding titles: \( \geq 8.0 \)