

## Discussion #5

*Name:*

## Writing SQL Queries

Given the tables,

```
Clowns(cid integer, cname text, booth text)
```

```
Balloons(bid integer, bshape text, bcolor text)
```

```
Catalog(cid integer, bid integer, cost float)
```

Note: The Catalog table contains prices for Balloons sold by different Clowns standing at certain booths in a fair.

1. How may we query for the top 3 most expensive shapes sold by Whompers LeFou?

**Solution:**

```
SELECT bshape, cost
FROM Clowns, Balloons, Catalog
WHERE Clowns.cid=Catalog.cid
      AND Balloons.bid=Catalog.bid
      AND cname='Whompers LeFou'
ORDER BY cost DESC
LIMIT 3;
```

2. How many different colors are available at each booth?

**Solution:**

```
SELECT booth, COUNT(DISTINCT bcolor)
FROM Clowns, Balloons, Catalog
WHERE Clowns.cid=Catalog.cid
      AND Balloons.bid=Catalog.bid
GROUP BY booth
```

3. What is the average cost of a red balloon at booths that offer more than 3 red shapes per clown?  
Note that each clown at the booth does not necessarily have to be selling more than 3 shapes.

**Solution:**

```
SELECT booth, avg(cost)
FROM Clowns, Balloons, Catalog
WHERE Clowns.cid=Catalog.cid
      AND Balloons.cid=Catalog.cid
      AND bcolor='red'
GROUP BY booth
HAVING COUNT(DISTINCT bshape)/COUNT(DISTINCT Clowns.cid) > 3
```

You can play with a toy version of this schema at:

<https://tinyurl.com/ds100-clowns>

4. The following relational schema represents a large database describing Olympic medalists.

```
medalist(name, country, birthday);
games(year, city, country);
medals(name, year, category, medaltype);
```

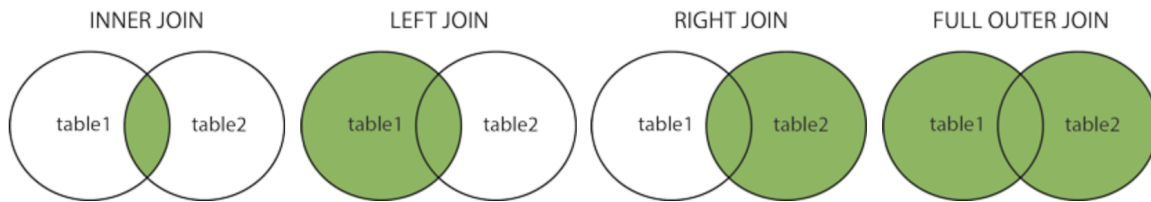
Which of the following queries returns the total number of medals broken down by type (gold, silver, and bronze) for each country in the 'vault' competition. (Select all that apply.)

- A. `SELECT medalists.country,  
medals.medaltype,  
COUNT(*) AS medal_count  
FROM medals, medalists  
WHERE medalists.name = medals.name  
AND medals.category = 'vault'  
GROUP BY medalists.country, medals.medaltype`
- B. `SELECT games.country,  
medals.medaltype,  
COUNT(medals.medaltype) AS medal_count  
FROM medals, games  
AND games.year = medals.year  
HAVING medals.category = 'vault'  
GROUP BY games.country, medals.medaltype`
- C. `SELECT medalists.country,  
medals.medaltype,  
COUNT(*) AS medal_count  
FROM medals, medalists  
WHERE medalists.name = medals.name  
GROUP BY medalists.country, medals.medaltype, medals.category  
HAVING category = 'vault'`
- D. `FROM medals, games  
SELECT games.country,  
medals.medaltype,  
COUNT(medals.medaltype) AS medal_count  
AND games.year = medals.year  
AND medals.category = 'vault'  
GROUP BY games.country, medals.medaltype`

**Solution:** Choice 1 and 3 are correct.

Choice 2 uses incorrect syntax and incorrect tables. Choice 4 is incorrect syntax.

## SQL Joins



Note: You do not always have to use the JOIN keyword to join sql tables. The following are equivalent:

```
SELECT column1, column2
FROM table1, table2
WHERE table1.id = table2.id;
```

```
SELECT column1, column2
FROM table1 JOIN table2
ON table1.id = table2.id;
```

5. Describe which records are returned from each type of join.

**Solution:**

(INNER) JOIN: Returns records that have matching values in both tables

LEFT (OUTER) JOIN: Return all records from the left table, and the matched records from the right table

RIGHT (OUTER) JOIN: Return all records from the right table, and the matched records from the left table

FULL (OUTER) JOIN: Return all records when there is a match in either left or right table

## SQL

6. Circle TRUE or FALSE.

- (a) **TRUE**    False    SQL is a declarative language that specifies what to produce but not how to compute it.

**Solution:** SQL is declarative programming language which specifies what the user wants to accomplish allowing the system to determine how to accomplish it.

- (b) **TRUE** False The primary key of a relation is the column or set of columns that determine the values of the remaining column.
- (c) True **FALSE** The schema of a table consists of the data stored in the table.

**Solution:** The schema of a table consists of the column names, their types, and any constraints on those columns. The instance of a database is the data stored in the database.

- (d) True **FALSE** The WHERE and HAVING clause can be used interchangeably as they perform the same operation.

**Solution:** The having clause is used to filter out groups, while the where clause operates on individual rows.