New submission for: Lecture 12 Exercise

Due: 10/18/2020 @ 11:59 PM EDT

You are given the baking database:
bakers(baker,fullname,age,occupation,hometown)
episodes(id,title,firstaired,viewers7day,signature,technical,showstopper)
results(episodeid,baker,result) technicals(episodeid,baker,rank)
signatures(episodeid,baker,make)
showstoppers(episodeid,baker,make) favorites(episodeid,baker)

For each question, enter your SQL query directly in the box. Do not forget to put a semicolon at the end.

1. Return the average viewers (viewers7day) for episodes. Name the returned attribute avgviewers and cast the result as numeric(5,2) using :: for casting.

Press TAB to indent. Press ESC to advance from answer area.

```
SELECT
    AVG(viewers7day)::NUMERIC(5,2) as avgviewers
FROM
    episodes;
```

Clear | Use Most Recent Submission

2. Using your query from above, return the id of all episodes with more than average viewers. Order results by id ascending.

Press TAB to indent. Press ESC to advance from answer area.

```
SELECT
    id
FROM
    episodes
WHERE
    viewers7day > (SELECT AVG(viewers7day)::NUMERIC(5,2) FROM episodes)
ORDER BY
    id;
```

Clear | Use Most Recent Submission

3. For each baker, return their fullname and the number of times they used 'chocolate' in their showstopper bakes (numtimes). Order results by name and numtimes ascending.

Press TAB to indent. Press ESC to advance from answer area.

```
SELECT
    fullname,
    COALESCE(numtimes, 0) as numtimes
FROM
    bakers b
    LEFT JOIN (SELECT COUNT(*) as numtimes, baker FROM showstoppers st WHERE lower(st.make) LIKE
'%chocolate%' GROUP BY baker) AS t
    ON t.baker = b.baker
ORDER BY
    fullname,
    numtimes;
```

Clear | Use Most Recent Submission

4. For each baker in the technicals table, return baker and the number of times they were rank 1 (numtimes). Order by baker and numtimes ascending.

Press TAB to indent. Press ESC to advance from answer area.

```
SELECT
    t1.baker,
    t2.numtimes
FROM
    (SELECT DISTINCT baker FROM technicals) as t1
    LEFT JOIN (SELECT COUNT(*) as numtimes, baker FROM technicals t1 WHERE t1.rank = 1 GROUP BY
t1.baker) AS t2
    ON t1.baker = t2.baker
WHERE
    t2.numtimes IS NOT NULL
ORDER BY
    baker,
    numtimes;
```

Clear | Use Most Recent Submission

5. Using your query from above, find the two largest numtimes values in descending order (i.e. the two highest number of times someone ranked 1 in technicals)!

Press TAB to indent. Press ESC to advance from answer area.

```
SELECT
    numtimes
FROM
    (SELECT
        t1.baker,
        t2.numtimes
    FROM
        (SELECT DISTINCT baker FROM technicals) as t1
        LEFT JOIN (SELECT COUNT(*) as numtimes, baker FROM technicals t1 WHERE t1.rank = 1
GROUP BY t1.baker) AS t2
        ON t1.baker = t2.baker
    WHERE
        t2.numtimes IS NOT NULL
    ORDER BY
        baker,
        numtimes) AS r
ORDER BY
    numtimes DESC
```

Clear | Use Most Recent Submission

By clicking "Submit" you are confirming that you have read, understand, and agree to follow the Academic Integrity Policy.

Submit

Select Submission Version: Version #3 Score: 50 / 50 GRADE THIS VERSION | Do Not Grade This Assignment

Note: This version of your assignment will be graded by the instructor/TAs and the score recorded in the gradebook.

Submitted Files

l12ex_1.sql (0.07kb)	Download
l12ex_2.sql (0.11kb)	Download
l12ex_3.sql (0.26kb)	Download
l12ex_4.sql (0.29kb)	Download
l12ex_5.sql (0.41kb)	Download
Download all files:	

First access timestamp:	12/20/2020 @ 04:42:56 PM EST
Submission timestamp:	10/16/2020 @ 01:25:21 AM EDT
Days late:	0 (before extensions)
Grading time:	75 seconds
Queue wait time:	0 seconds

50 / 50 Autograding Total

50 / 50 Test 1 l12ex Show Details