Course Home
Gradeables
Notifications
Office Hours Queue
Submini Polls
Course Materials
Discussion Forum
My Late Days/Extensions
My Courses
My Profile
Collapse Sidebar
Logout Yihang

New submission for: Lecture 12 Exercise

Due: 10/17/2021 @ 04:00 PM EDT

Gradeable Time Remaining: 02 days 21 hours

You are given the baking database:

bakers(baker,fullname,age,occupation,hometown)

episodes(id,title,firstaired,viewers7day,signature,technical

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.

Please make sure you DO NOT use any non-ASCII characters in your queries. This includes all fancy quotes you might get if you copy code from a PDF or other rich format. Postgresql will throw an error if it encounters invalid characters, and you will garde a zero for that question.

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(e.viewers7day)::numeric(5, 2) as a
FROM episodes e;
```

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 e.id FROM episodes e
WHERE e.viewers7day > (SELECT avg(e.viewers7d
FROM episodes e)
ORDER BY id ASC;
```

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 b.fullname, COUNT(DISTINCT s.episodeid
FROM bakers b LEFT OUTER JOIN showstoppers s
ON b.baker = s.baker and lower(s.make) LIKE '
GROUP BY b.baker
ORDER BY b.fullname ASC, numtimes ASC;
```

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 b.baker, COUNT(DISTINCT t.episodeid) a
FROM bakers b LEFT OUTER JOIN technicals t ON
GROUP BY b.baker
ORDER BY b.baker ASC, numtimes ASC;
```

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 baker, COUNT(DISTINCT episodeid) as nu
FROM technicals
WHERE rank = 1
GROUP BY baker
ORDER BY baker ASC, numtimes ASC) AS x
ORDER BY numtimes DESC LIMIT 2;
```

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 #5 Score: 50 / 50 GRADE THIS VI

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 (0.14kb)	Download
l12ex_3.sql (0.21kb)	Download
l12ex_4.sql (0.18kb)	Download
l12ex_5.sql (0.19kb)	Download
Download all files:	

First access timestamp:	10/14/2021 @ 06:17:14 PM EDT
Submission timestamp:	10/14/2021 @ 06:07:36 PM EDT
Days late:	0 (before extensions)
Grading time:	23 seconds
Graded on machine:	worker4
Queue wait time:	1 seconds

50 / 50 Autograding Total

Show Details