




← Project review - SQL1 Bootcamp. Day07

 Type of project	Individual
 Duration	30 min
 Passed Peer Reviews	0/3

Git project



ssh://git@repos-ssh.21-school.ru:2289/students/SQL_beginner._Day07.ID_574096/zoo...

Copy link

Open

Student



zoomdeni@student.21-school.ru

level 8

About



Introduction

The methodology of School 21 makes sense only if peer-to-peer reviews are done seriously. Please read all guidelines carefully before starting the review.

- Please, stay courteous, polite, respectful and constructive in all communications during this review.
- Highlight possible malfunctions of the work done by the person and take the time to discuss and debate it.
- Keep in mind that sometimes there can be differences in interpretation of the tasks and the scope of features. Please, stay open-minded to the vision of the other.
- If you have not finished the project yet, it is compulsory to read the entire instruction before starting the review.

Guidelines

- Evaluate only the files that are in src folder on the GIT repository of the student or group.
- Ensure to start reviewing a group project only when the team is present in full.
- Use special flags in the checklist to report, for example, an “empty work” if repository does not contain the work of the student (or group) in the src folder of the develop branch, or “cheat” in case of cheating or if the student (or group) are unable to explain their work at any time during review as well as if one of the points below is not met. However, except for cheating cases, you are encouraged to continue reviewing the project to identify the problems that caused the situation in order to avoid them at the next review.
- Doublecheck that the GIT repository is the one corresponding to the student or the group.
- Meticulously check that nothing malicious has been used to mislead you.
- In controversial cases, remember that the checklist determines only the general order of the check. The final decision on project evaluation remains with the reviewer.

Main part



Exercise 00

Checks for the file day07_ex00.sql

- The SQL script looks like below.

```
select person_id, count(*) as "count_of_visits"
from person_visits
group by person_id
order by 2 desc,1 asc;
```

- The result is below (raw ordering should be the same like on a screen below)

```
"9"  "4"
```

"4" "3"
"6" "3"
"8" "3"
"2" "2"
"3" "2"
"5" "2"
"7" "2"
"1" "1"

Exercise 01

Checks for the file day07_ex01.sql

- The SQL script looks like below.

```
select p.name, count(*) as "count_of_visits"  
from person_visits inner join person p on p.id = person_visits.person_id  
group by p.name  
order by 2 desc,1 asc  
limit 4;
```

- The result is below (raw ordering should be the same like below)

"Dmitriy" "4"
"Denis" "3"
"Irina" "3"
"Nataly" "3"

Exercise 02

Checks for the file day07_ex02.sql

- The SQL script looks like below.

```
(select p.name, count(*) as "count", 'visit' as action_type  
from person_visits inner join pizzeria p on p.id = person_visits.pizzeria_id  
group by p.name  
order by 2 desc  
limit 3)  
union  
(select p.name, count(*) as "count", 'order' as action_type  
from person_order inner join menu m on person_order.menu_id = m.id
```

```

inner join pizzeria p on m.pizzeria_id = p.id
group by p.name
order by 2 desc
limit 3)
order by 3,2 desc

```

- The result is below (raw ordering should be the same like below)

"Dominos"	"6"	"order"
"Best Pizza"	"5"	"order"
"DinoPizza"	"5"	"order"
"Dominos"	"7"	"visit"
"DinoPizza"	"4"	"visit"
"Pizza Hut"	"4"	"visit"

Exercise 03

Checks for the file day07_ex03.sql

- The SQL script looks like below.

```

select t1.name, coalesce(t1.count,0) + coalesce(t2.count,0) as total_count
from
(select p.name, count(*) as "count"
from person_visits inner join pizzeria p on p.id = person_visits.pizzeria_id
group by p.name) as t1 full join
(select p.name, count(*) as "count"
from person_order inner join menu m on person_order.menu_id = m.id
inner join pizzeria p on m.pizzeria_id = p.id
group by p.name) as t2 on t1.name = t2.name
order by 2 desc,1 asc;

```

- The result is below (raw ordering should be the same like below)

"Dominos"	"13"
"DinoPizza"	"9"
"Best Pizza"	"8"
"Pizza Hut"	"8"
"Papa Johns"	"5"
"DoDo Pizza"	"1"

Exercise 04

Checks for the file day07_ex04.sql

- The SQL script looks like below.

```
select p.name, count(*) as "count_of_visits"  
from person_visits inner join person p on p.id = person_visits.person_id  
group by p.name  
having count(*) > 3;
```

- The result is below (raw ordering should be the same like below)

"Dmitriy" "4"

Exercise 05

Checks for the file day07_ex05.sql

- The SQL script looks like below.

```
select distinct p.name  
from person_order inner join person p on p.id = person_order.person_id  
order by 1;
```

- The result is below (raw ordering should be the same like below)

"Andrey"
"Anna"
"Denis"
"Dmitriy"
"Elvira"
"Irina"
"Kate"
"Nataly"
"Peter"

Exercise 06

Checks for the file day07_ex06.sql

- The SQL script looks like below.

```

select p.name, count(*) as count_of_orders, round(avg(m.price),2) as average_price, max(m.price) as max_price, min(m.price) as min_price
  from person_order inner join menu m on person_order.menu_id = m.id
    inner join pizzeria p on m.pizzeria_id = p.id
 group by p.name
 order by 1;

```

- The result is below (raw ordering should be the same like below)

"Best Pizza"	"5"	"780"	"850"	"700"
"DinoPizza"	"5"	"880"	"1000"	"800"
"Dominos"	"6"	"933.33"	"1100"	"800"
"Papa Johns"	"2"	"975"	"1000"	"950"
"Pizza Hut"	"4"	"1125"	"1200"	"900"

Exercise 07

Checks for the file day07_ex07.sql

- The SQL script looks like below.

```

select round(avg(rating),4) as global_rating
  from pizzeria;

```

- The result is below (raw ordering should be the same like below)

"3.9167"

Exercise 08

Checks for the file day07_ex08.sql

- The SQL script looks like below.

```

select address, p.name, count(*)
  from person_order inner join menu m on person_order.menu_id = m.id
    inner join pizzeria p on m.pizzeria_id = p.id
    inner join person p1 on p1.id = person_order.person_id
 group by address, p.name
 order by 1,2;

```

- The result is below (raw ordering should be the same like below)

```

"Kazan"    "Best Pizza"  "4"
"Kazan"    "DinoPizza"   "4"
"Kazan"    "Dominos"    "1"
"Moscow"   "Dominos"    "2"
"Moscow"   "Pizza Hut"   "2"
"Novosibirsk" "Dominos"  "1"
"Novosibirsk" "Papa Johns" "1"
"Saint-Petersburg" "Dominos" "2"
"Saint-Petersburg" "Papa Johns" "1"
"Saint-Petersburg" "Pizza Hut" "2"
"Samara"    "Best Pizza"  "1"
"Samara"    "DinoPizza"  "1"

```

No

✓ Yes

Exercise 09

Checks for the file day07_ex09.sql

- The SQL script looks like below.

```

select address,round((max(age)- min(age) /max(age::numeric)),2) as formula,round(avg
(age),2) as "average",
        round((max(age)- min(age) /max(age::numeric)),2) > round(avg(age),2) as compariso
n
from person
group by address
order by 1;

```

- The result is below (raw ordering should be the same like below)

```

"Kazan"    "44.71"  "30.33"  "true"
"Moscow"   "20.24"  "18.5"   "true"
"Novosibirsk" "29"    "30"    "false"
"Saint-Petersburg" "23.13" "22.5"  "true"
"Samara"    "17"    "18"    "false"

```

No

✓ Yes

Feedback



fails 

Cheat

Forbidden functions

Empty work

Comment

Leave a comment...

✓ Review