← Project review - SQL1 Bootcamp. Day04

Type of project	Individual
Duration	30 min
Passed Peer Reviews	2/3

Git project

ssh://git@repos-ssh.21-school.ru:2289/students/SQL_beginner._Day04.ID_574092/brai...

Copy link Open

Student

brainych@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.

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 day04_ex00.sql
- The SQL script looks like below.

```
create view v_persons_female as
select *
from person
where gender= 'female';

create view v_persons_male as
select *
from person
where gender= 'male';
```

- The SQL script looks like below.

```
select *
from v_persons_female
order by 1
```

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

```
"1"    "Anna"    "16"        "female" "Moscow"
```

```
"3"   "Kate"   "33"     "female" "Kazan"
"5"   "Elvira" "45"     "female" "Kazan"
"6"   "Irina"  "21"     "female" "Saint-Petersburg"
"8"   "Nataly" "30"     "female" "Novosibirsk"
```

- The SQL script looks like below.

```
select *
from v_persons_male
order by 1
```

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

```
"2"   "Andrey" "21"     "male"   "Moscow"
"4"   "Denis"  "13"     "male"   "Kazan"
"7"   "Peter"  "24"     "male"   "Saint-Petersburg"
"9"   "Dmitriy"     "18"     "male"   "Samara"
```

No ✓ Yes

Exercise 01

Checks for the file day04_ex01.sql
- The SQL script looks like below.

```
select name
from v_persons_female
union
select name
from v_persons_male
order by 1
```

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

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

No ✓ Yes

Exercise 02

Checks for the file day04_ex02.sql
- The SQL script looks like below.

```
create view v_generated_dates as
select g::date as generated_date
from generate_series('2022-01-01','2022-01-31',interval '1 day') as g
order by 1
```

- The check script is below.

```
select count(*) =31 as check,
    min(generated_date) as check1,
    max(generated_date) as check2
from v_generated_dates;
```

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

```
true   2022-01-01    2022-01-31
```

No ✓ Yes

Exercise 03

Checks for the file day04_ex03.sql
- The SQL script looks like below.

```
select generated_date as missing_date
from v_generated_dates
except
select visit_date
from person_visits
order by 1
```

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

```
"2022-01-11"
"2022-01-12"
"2022-01-13"
"2022-01-14"
"2022-01-15"
"2022-01-16"
"2022-01-17"
"2022-01-18"
"2022-01-19"
"2022-01-20"
"2022-01-21"
"2022-01-22"
"2022-01-23"
"2022-01-24"
"2022-01-25"
"2022-01-26"
"2022-01-27"
```

```
    "2022-01-28"
    "2022-01-29"
    "2022-01-30"
    "2022-01-31"
```

No ✓ Yes

Exercise 04

Checks for the file day04_ex04.sql
- The SQL script looks like below.

```
    create view v_symmetric_union as
    (select person_id
    from person_visits
    where visit_date = '2022/01/02'
    except
    select person_id
    from person_visits
    where visit_date = '2022/01/06')
    union
    (select person_id
    from person_visits
    where visit_date = '2022/01/06'
    except
    select person_id
    from person_visits
    where visit_date = '2022/01/02')
    order by 1
```

- The SQL script looks like below.

```
    select *
    from v_symmetric_union
```

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

```
     "2"
     "8"
```

No ✓ Yes

Exercise 05

Checks for the file day04_ex05.sql
- The SQL script looks like below.

```
     create view v_price_with_discount as
     SELECT p.name,
          m.pizza_name,
          m.price,
          round(m.price - m.price * 0.1) AS discount_price
      FROM person_order
              JOIN person p ON p.id = person_order.person_id
              JOIN menu m ON m.id = person_order.menu_id
      ORDER BY p.name, m.pizza_name
```

- The SQL script looks like below.

```
    select *
    from v_price_with_discount
```

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

```
      "Andrey"  "cheese pizza"      "800"     "720"
      "Andrey"  "mushroom pizza"  "1100"    "990"
      "Anna"     "cheese pizza"      "900"     "810"
      "Anna"     "pepperoni pizza"  "1200"    "1080"
      "Denis"    "cheese pizza"      "700"     "630"
      "Denis"    "pepperoni pizza"  "800"     "720"
      "Denis"    "pepperoni pizza"  "800"     "720"
      "Denis"    "sausage pizza"     "1000"    "900"
      "Denis"    "sicilian pizza"     "900"     "810"
      "Denis"    "supreme pizza"     "850"     "765"
      "Dmitriy" "pepperoni pizza"  "800"     "720"
      "Dmitriy" "supreme pizza"     "850"     "765"
      "Elvira"   "pepperoni pizza"  "800"     "720"
      "Elvira"   "sausage pizza"     "1000"    "900"
      "Irina"     "mushroom pizza"  "950"     "855"
      "Irina"     "sicilian pizza"     "900"     "810"
      "Kate"      "cheese pizza"      "700"     "630"
      "Nataly"  "cheese pizza"      "800"     "720"
      "Nataly"  "pepperoni pizza"  "1000"    "900"
      "Peter"    "mushroom pizza"  "1100"    "990"
      "Peter"    "sausage pizza"     "1200"    "1080"
      "Peter"    "supreme pizza"     "1200"    "1080"
```

No ✓ Yes

Exercise 06

Checks for the file day04_ex06.sql
- The SQL script looks like below.

```
       create materialized view mv_dmitriy_visits_and_eats as
         select p.name as pizzeria_name
         from menu inner join pizzeria p on p.id = menu.pizzeria_id
         inner join person_visits pv on menu.pizzeria_id = pv.pizzeria_id
         inner join person p2 on p2.id = pv.person_id
         where price  < 800 and p2.name = 'Dmitriy' and visit_date = '2022-01-08'
```

- The SQL script looks like below.

```
select *
from mv_dmitriy_visits_and_eats
```

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

```
"Papa Johns"
```

No ✓ Yes

Exercise 07

Checks for the file day04_ex07.sql
- The SQL script looks like below.

```
 insert into person_visits(id, person_id, pizzeria_id, visit_date)
   values( (select max(id)+1 from person_visits),(select id from person where name= 'Dmitriy'),
       (select pizzeria.id from pizzeria  inner join menu m on pizzeria.id = m.pizzeria_id
          where price < 800 and m.pizza_name!='Papa Johns' order by 1 limit 1),'2022-01-08');

   refresh materialized view mv_dmitriy_visits_and_eats;
```

- The check script is below.

```
select *
from mv_dmitriy_visits_and_eats
```

- The result is below. Please ignore a value "DoDo Pizza", a value can be different from student

```
"DoDo Pizza"
"Papa Johns"
```

No ✓ Yes

Exercise 08

Checks for the file day04_ex08.sql
- The SQL script looks like below.

```
drop view v_generated_dates;
drop view v_persons_female;
drop view v_persons_male;
drop view v_price_with_discount;
drop view v_symmetric_union;
drop materialized view  mv_dmitriy_visits_and_eats;
```

- The SQL script looks like below.

```
 select count(*) =0 as check
from pg_class
where relname in ('v_generated_dates','v_persons_female','v_persons_male',
'v_price_with_discount','v_symmetric_union', 'mv_dmitriy_visits_and_eats')
```

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

```
"true"
```

No ✓ Yes

Feedback

Fails ⓘ

Cheat Empty work Forbidden functions

Comment

Leave a comment...

✓ Review