



## ← Project review - SQL1 Bootcamp. Day06

 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.\_Day06.ID\_574095/me...

Copy link

Open

### Student



mediumcl@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 day06\_ex00.sql

- The SQL script looks like below.

```
create table person_discounts
(id bigint primary key ,
 person_id bigint,
 pizzeria_id bigint,
 discount numeric,
 constraint fk_person_discounts_person_id foreign key (person_id) references person(id),
 constraint fk_person_discounts_pizzeria_id foreign key (pizzeria_id) references pizzeria(id)
);
```

- The SQL script looks like below.

```
select count(*) = 1 as check
from pg_tables
where tablename = 'person_discounts'
```

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

"true"

### Exercise 01

Checks for the file day06\_ex01.sql

- The SQL script looks like below.

```
insert into person_discounts (id, person_id, pizzeria_id, discount)
select
    row_number() over () as id,
    person_id,
    m.pizzeria_id,
    case
        when count(*) = 1 then 10.5
        when count(*) = 2 then 22
        else 30
    end as discount
from person_order inner join menu m on m.id = person_order.menu_id
group by person_id, m.pizzeria_id;
```

- The SQL script looks like below.

```
select count(*) > 0 as check
from person_discounts
```

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

"true"

## Exercise 02

Checks for the file day06\_ex02.sql

- The SQL script looks like below.

```
select p.name, m.pizza_name, m.price, (m.price - (m.price * pd.discount/100) ) as discount_price, p2.name as pizzeria_name
from person_order inner join menu m on m.id = person_order.menu_id
inner join person p on p.id = person_order.person_id
inner join person_discounts pd on p.id = pd.person_id and pd.pizzeria_id = m.pizzeria_id
inner join pizzeria p2 on m.pizzeria_id = p2.id
order by 1,2
```

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

"Andrey"	"cheese pizza"	"800"	"624"	"Dominos"
"Andrey"	"mushroom pizza"	"1100"	"858"	"Dominos"
"Anna"	"cheese pizza"	"900"	"702"	"Pizza Hut"
"Anna"	"pepperoni pizza"	"1200"	"936"	"Pizza Hut"
"Denis"	"cheese pizza"	"700"	"490"	"Best Pizza"
"Denis"	"pepperoni pizza"	"800"	"624"	"DinoPizza"
"Denis"	"pepperoni pizza"	"800"	"560"	"Best Pizza"
"Denis"	"sausage pizza"	"1000"	"780"	"DinoPizza"
"Denis"	"sicilian pizza"	"900"	"805.5"	"Dominos"
"Denis"	"supreme pizza"	"850"	"595"	"Best Pizza"
"Dmitriy"	"pepperoni pizza"	"800"	"716"	"DinoPizza"
"Dmitriy"	"supreme pizza"	"850"	"760.75"	"Best Pizza"
"Elvira"	"pepperoni pizza"	"800"	"624"	"DinoPizza"
"Elvira"	"sausage pizza"	"1000"	"780"	"DinoPizza"
"Irina"	"mushroom pizza"	"950"	"850.25"	"Papa Johns"
"Irina"	"sicilian pizza"	"900"	"805.5"	"Dominos"
"Kate"	"cheese pizza"	"700"	"626.5"	"Best Pizza"
"Nataly"	"cheese pizza"	"800"	"716"	"Dominos"
"Nataly"	"pepperoni pizza"	"1000"	"895"	"Papa Johns"
"Peter"	"mushroom pizza"	"1100"	"984.5"	"Dominos"
"Peter"	"sausage pizza"	"1200"	"936"	"Pizza Hut"
"Peter"	"supreme pizza"	"1200"	"936"	"Pizza Hut"

No

✓ Yes

## Exercise 03

Checks for the file day06\_ex03.sql

- The SQL script looks like below.

```
create unique index idx_person_discounts_unique on person_discounts(person_id,pizzeria_
```

id);

```
set enable_seqscan =off;
explain analyse
select *
from person_discounts
where person_id = 1 and pizzeria_id=12;
```

- The result should contain a part of text below

"Index Scan using idx\_person\_discounts\_unique on person\_discounts ..."

No

✓ Yes

## Exercise 04

Checks for the file day06\_ex04.sql

- The SQL script looks like below.

```
alter table person_discounts add constraint ch_nn_person_id check ( person_id is not null);
alter table person_discounts add constraint ch_nn_pizzeria_id check ( pizzeria_id is not null);
alter table person_discounts add constraint ch_nn_discount check ( discount is not null);
alter table person_discounts alter column discount set default 0;
alter table person_discounts add constraint ch_range_discount check ( discount between 0 and 100);
```

- The SQL script looks like below.

```
select count(*) = 4 as check
from pg_constraint
where conname in ('ch_nn_person_id','ch_nn_pizzeria_id','ch_nn_discount','ch_range_discount')
```

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

"true"

- The SQL script looks like below.

```
select column_default::integer = 0 as check
from information_schema.columns
where column_name = 'discount' and table_name = 'person_discounts'
```

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

"true"

No

✓ Yes

### Exercise 05

Checks for the file day06\_ex05.sql

- The SQL script looks like below.

```
comment on column person_discounts.id is 'primary key identifier';
comment on column person_discounts.person_id is 'person identifier';
comment on column person_discounts.pizzeria_id is 'pizzeria identifier';
comment on column person_discounts.discount is 'discount for pair person and pizzeria';
comment on table person_discounts is 'Discounts for persons';
```

- The SQL script looks like below.

```
SELECT count(*) = 5 as check
FROM pg_description
WHERE objoid = 'person_discounts'::regclass
```

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

"true"

No

✓ Yes

### Exercise 06

Checks for the file day06\_ex06.sql

- The SQL script looks like below.

```
create sequence seq_person_discounts start 1;
alter table person_discounts alter column id set default nextval('seq_person_discounts');
select setval('seq_person_discounts', (select count(*)+1 from person_discounts));
```

- The SQL script looks like below.

```
select count(*)=1 as check1,
       max("start_value") = 1 as check2,
       max("last_value") > 5 as check3
from pg_sequences
```

where sequencename = 'seq\_person\_discounts'

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

"true" "true" "true"

No

✓ Yes

## Feedback



Fails 

Forbidden functions

Empty work

Cheat

Comment

Leave a comment...

✓ Review