← **Project review - SQL beginner. Day05**

📄 Type of project	Individual
🕐 Duration	30 min
☑ Passed Peer Reviews	0/2

Git project   ⌃

ssh://git@repos-ssh.21-school.ru:2289/students/SQL_beginner._Day05.ID_574093/hcolumb...

Copy link   **Open**

Student   ⌃

👤 **hcolumbu-sql-tester**   level 1
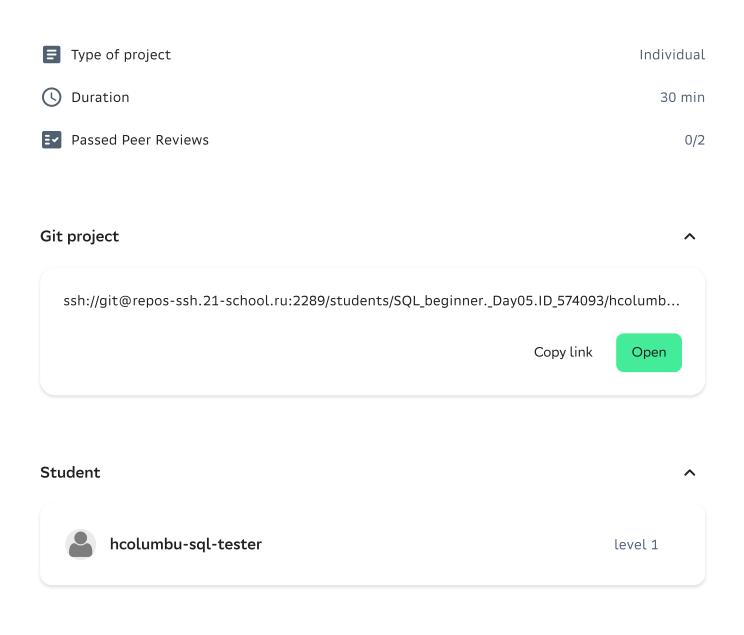
About   ⌃

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 day05_ex00.sql
- The SQL script looks like below.

```
create index idx_menu_pizzeria_id on menu(pizzeria_id);
create index idx_person_order_person_id on person_order(person_id);
create index idx_person_order_menu_id on person_order(menu_id);
create index idx_person_visits_person_id on person_visits(person_id);
create index idx_person_visits_pizzeria_id on person_visits(pizzeria_id);
```

- The SQL script looks like below.

```
select count(*) = 5 as check
```

```
from pg_indexes
where indexname in ('idx_menu_pizzeria_id','idx_person_order_person_id',
'idx_person_order_menu_id','idx_person_visits_person_id','idx_person_visits_pizzeria_id')
```

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

```
"true"
```

No   ✓ Yes

Exercise 01

Checks for the file day05_ex01.sql
- The SQL script looks like below.

```
set enable_seqscan =off;
explain analyse
select
    m.pizza_name as pizza_name,
    pz.name as pizzeria_name
from  menu m
inner join pizzeria pz on m.pizzeria_id = pz.id;
```

- The result should contain a part of text below.

```
"Index Scan using idx_menu_pizzeria_id on menu m  ..."
```

No   ✓ Yes

Exercise 02

Checks for the file day05_ex02.sql
- The SQL script looks like below.

```
create index idx_person_name on person(upper(name));

set enable_seqscan =off;
explain analyse
select *
from person
where upper(name) = 'Kate';
```

- The result should contain a part of text below.

"Index Scan using idx_person_name on person ..."

No   ✓   Yes

Exercise 03

Checks for the file day05_ex03.sql
- The SQL script looks like below.

    create index idx_person_order_multi on person_order(person_id, menu_id,order_date);

    set enable_seqscan =off;
    explain analyse
    SELECT person_id, menu_id,order_date
    FROM person_order
    WHERE person_id = 8 AND
       menu_id = 19;

- The result should contain a part of text below.

    "Index Only Scan using idx_person_order_multi on person_order ..."

No   ✓   Yes

Exercise 04

Checks for the file day05_ex04.sql
- The SQL script looks like below.

    create unique index idx_menu_unique on menu(pizzeria_id,pizza_name);

    set enable_seqscan =off;
    explain analyse
    select *
    from menu
    where pizzeria_id = 1 and pizza_name = 'cheese pizza';

- The result should contain a part of text below.

    "Index Scan using idx_menu_unique on menu ..."

No   ✓   Yes

Exercise 05

Checks for the file day05_ex05.sql
- The SQL script looks like below.

```
create unique index idx_person_order_order_date on person_order(person_id, menu_id) where order_date = '2022-01-01';

set enable_seqscan =off;
explain analyse
select menu_id
from person_order
where order_date = '2022-01-01' and  person_id =2;
```

- The result should contain a part of text below.

```
"Index Only Scan using idx_person_order_order_date on person_order  ..."
```

No ✓ Yes

Exercise 06

Checks for the file day05_ex06.sql
- The SQL script looks like below.

```
create index idx_1 on pizzeria(rating);

set enable_seqscan =off;
explain analyse
SELECT
    m.pizza_name AS pizza_name,
    max(rating) OVER (PARTITION BY rating ORDER BY rating ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS k
    FROM  menu m
    INNER JOIN pizzeria pz ON m.pizzeria_id = pz.id
    ORDER BY 1,2;
```

- The result should contain a part of text below.

```
"Index Scan using idx_1 on pizzeria pz  ..."
```

No ✓ Yes

Feedback ⌃

Fails ⓘ

Forbidden functions   Empty work   Cheat

Comment

Leave a comment...

✓ Review