

## ← Project review - SQL1 Bootcamp. Day08

 Type of project	Individual
 Duration	15 min
 Passed Peer Reviews	0/2

### Git project

ssh://git@repos-ssh.21-school.ru:2289/students/SQL\_be...

[Open](#)

### Student



craydlec@student.21-school.ru

level 8

### About

#### Introduction

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

- Please remain courteous, polite, respectful, and constructive in all communications during this review.
- Point out possible flaws in the person's work and take the time to discuss and debate them.

- Keep in mind that sometimes there may be differences in interpretation of tasks and scope of functionality. Please remain open to each other's vision.

## Guidelines

- Evaluate only the files located in the src folder of the student's or group's GIT repository.
- If you have not finished the project yet, it is compulsory to read the entire instruction before starting the review.
- 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, "empty work" if the repository does not contain the student's (or group's) work in the src folder of the develop branch, or "cheat" in the case of cheating, or if the student (or group) is unable to explain their work at any time during the review, or if any of the items below are not met. However, except in cases of cheating, you are encouraged to continue reviewing the project to identify the problems that caused the situation so that they can be avoided in the next review.
- Double check that the GIT repository is the one corresponding to the student or group.
- Carefully check that nothing malicious has been used to fool 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 day08\_ex00.sql

- The SQL scripts look like below.

```
-- Session #1
BEGIN;

update pizzeria
set rating = 5
where name = 'Pizza Hut';

select *
from pizzeria
where name = 'Pizza Hut';

COMMIT;

--Session #2
select *
from pizzeria
where name = 'Pizza Hut';
```

- The result of Session #1

```
=> begin;
BEGIN
=*> update pizzeria set rating = 5 where name = 'Pizza Hut';
UPDATE 1
=*> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----+-----
1 | Pizza Hut | 5
(1 row)
=*> commit;
COMMIT
```

- The result of Session #2

```
=> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----+-----
1 | Pizza Hut | 4.6
(1 row)
=> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----+-----
1 | Pizza Hut | 5
(1 row)
```

No

Yes

## Exercise 01

Checks for the file day08\_ex01.sql

- The SQL script looks like below.

```
-- Session #1
begin transaction isolation level read committed;
select * from pizzeria where name = 'Pizza Hut';
update pizzeria set rating = 4 where name = 'Pizza Hut';
commit;
select * from pizzeria where name = 'Pizza Hut';

--Session #2
begin transaction isolation level read committed;
select * from pizzeria where name = 'Pizza Hut';
update pizzeria set rating = 3.6 where name = 'Pizza Hut';
commit;
select * from pizzeria where name = 'Pizza Hut';
```

- The result of Session #1

```
=> begin transaction isolation level read committed;
BEGIN
=*> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
```

```

-----+-----
1 |Pizza Hut | 5
(1 row)
=> update pizzeria set rating = 4 where name = 'Pizza Hut';
UPDATE 1
=> commit;
COMMIT
=> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----
1 |Pizza Hut | 3.6
(1 row)

```

- The result of Session #2

```

=> begin transaction isolation level read committed;
BEGIN
=> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----
1 |Pizza Hut | 5
(1 row)
=> update pizzeria set rating = 3.6 where name = 'Pizza Hut';
UPDATE 1
=> commit;
COMMIT
=> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----
1 |Pizza Hut | 3.6
(1 row)

```

No

Yes

## Exercise 02

Checks for the file day08\_ex02.sql

- The SQL script looks like below.

```

-- Session #1
begin transaction isolation level repeatable read;
select * from pizzeria where name = 'Pizza Hut';
update pizzeria set rating = 4 where name = 'Pizza Hut';
commit;
select * from pizzeria where name = 'Pizza Hut';

--Session #2
begin transaction isolation level repeatable read;
select * from pizzeria where name = 'Pizza Hut';
update pizzeria set rating = 3.6 where name = 'Pizza Hut';
commit;
select * from pizzeria where name = 'Pizza Hut';

```

- The result of Session #1

```
=> begin transaction isolation level repeatable read ;
BEGIN
=> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----+-----
1 | Pizza Hut | 3.6
(1 row)
=> update pizzeria set rating = 4 where name = 'Pizza Hut';
UPDATE 1
=> commit;
COMMIT
=> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----+-----
1 | Pizza Hut | 4
(1 row)
```

- The result of Session #2

```
=> begin transaction isolation level repeatable read ;
BEGIN
=> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----+-----
1 | Pizza Hut | 3.6
(1 row)
=> update pizzeria set rating = 3.6 where name = 'Pizza Hut';
ERROR: could not serialize access due to concurrent update
!=> commit;
ROLLBACK
=> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----+-----
1 | Pizza Hut | 4
(1 row)
```

No

Yes

### Exercise 03

Checks for the file day08\_ex03.sql

- The SQL script looks like below.

```
-- Session #1
begin transaction isolation level read committed;
select * from pizzeria where name = 'Pizza Hut';
select * from pizzeria where name = 'Pizza Hut';
commit;
select * from pizzeria where name = 'Pizza Hut';
```

```
--Session #2
begin transaction isolation level read committed;
update pizzeria set rating = 3.6 where name = 'Pizza Hut';
commit;
select * from pizzeria where name = 'Pizza Hut';
```

- The result of Session #1

```
=> begin transaction isolation level read committed ;
BEGIN
=*> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----+-----
1 | Pizza Hut | 4
(1 row)
=*> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----+-----
1 | Pizza Hut | 3.6
(1 row)
=*> commit;
COMMIT
=> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----+-----
1 | Pizza Hut | 3.6
(1 row)
```

- The result of Session #2

```
=> begin transaction isolation level read committed ;
BEGIN
=*> update pizzeria set rating = 3.6 where name = 'Pizza Hut';
UPDATE 1
=*> commit;
COMMIT
=> select * from pizzeria where name = 'Pizza Hut';
id | name | rating
---+-----+-----
1 | Pizza Hut | 3.6
(1 row)
```

No

Yes

## Exercise 04

Checks for the file day08\_ex04.sql

- The SQL script looks like below.

```
-- Session #1
begin transaction isolation level serializable;
```

```

select * from pizzeria where name = 'Pizza Hut';
select * from pizzeria where name = 'Pizza Hut';
commit;
select * from pizzeria where name = 'Pizza Hut';

--Session #2
begin transaction isolation level serializable;
update pizzeria set rating = 3.0 where name = 'Pizza Hut';
commit;
select * from pizzeria where name = 'Pizza Hut';

```

- The result of Session #1

```

=> begin transaction isolation level serializable ;
BEGIN
=> select * from pizzeria where name = 'Pizza Hut';
id | name  | rating
---+-----+-----
1 | Pizza Hut | 3.6
(1 row)
=> select * from pizzeria where name = 'Pizza Hut';
id | name  | rating
---+-----+-----
1 | Pizza Hut | 3.6
(1 row)
=> commit;
COMMIT
=> select * from pizzeria where name = 'Pizza Hut';
id | name  | rating
---+-----+-----
1 | Pizza Hut | 3.0
(1 row)

```

- The result of Session #2

```

=*> begin transaction isolation level serializable ;
BEGIN
=*> update pizzeria set rating = 3.0 where name = 'Pizza Hut';
UPDATE 1
=> commit;
COMMIT
=> select * from pizzeria where name = 'Pizza Hut';
id | name  | rating
---+-----+-----
1 | Pizza Hut | 3.0
(1 row)

```

No

Yes

## Exercise 05



```
-- Session #1
begin transaction isolation level read committed;
select sum(rating) from pizzeria;
select sum(rating) from pizzeria;
commit;
select sum(rating) from pizzeria;

--Session #2
begin transaction isolation level read committed;
update pizzeria set rating = 1 where name = 'Pizza Hut';
commit;
select sum(rating) from pizzeria;
```

- The result of Session #1

```
=> begin transaction isolation level read committed ;
BEGIN
=*> select sum(rating)from pizzeria;
sum
---
21.9
(1 row)
=*> select sum(rating)from pizzeria;
sum
---
19.9
(1 row)
=> commit;
COMMIT
=> select sum(rating)from pizzeria;
sum
---
19.9
(1 row)
```

- The result of Session #2

```
=> begin transaction isolation level read committed ;
BEGIN
=*> update pizzeria set rating = 1 where name = 'Pizza Hut';
UPDATE 1
=*> commit;
COMMIT
=> select sum(rating) from pizzeria;
sum
---
19.9
(1 row)
```

No

✓ Yes



## Exercise 06

Checks for the file day08\_ex06.sql

- The SQL script looks like below.

```
-- Session #1
begin transaction isolation level repeatable read;
select sum(rating) from pizzeria;
select sum(rating) from pizzeria;
commit;
select sum(rating) from pizzeria;

--Session #2
begin transaction isolation level repeatable read;
update pizzeria set rating = 5 where name = 'Pizza Hut';
commit;
select sum(rating) from pizzeria;
```

- The result of Session #1

```
=> begin transaction isolation level repeatable read ;
BEGIN
=*> select sum(rating)from pizzeria;
sum
---
19.9
(1 row)
=*> select sum(rating)from pizzeria;
sum
---
19.9
(1 row)
=*> commit;
COMMIT
=*> select sum(rating)from pizzeria;
sum
---
23.9
(1 row)
```

- The result of Session #2

```
=> begin transaction isolation level repeatable read ;
BEGIN
=*> update pizzeria set rating = 5 where name = 'Pizza Hut';
UPDATE 1
=*> commit;
COMMIT
=> select sum(rating)from pizzeria;
sum
---
23.9
(1 row)
```

Yes

## Exercise 07

Checks for the file day08\_ex07.sql

- The SQL script looks like below.

```
-- Session #1
begin;
update pizzeria set rating = 4 where name = 'Dominos';
update pizzeria set rating = 4 where name = 'Pizza Hut';
commit;
select sum(rating) from pizzeria;
```

```
--Session #2
begin;
update pizzeria set rating = 3 where name = 'Pizza Hut';
update pizzeria set rating = 3 where name = 'Dominos';
commit;
```

- The result of Session #1

```
=> begin;
BEGIN
=*> update pizzeria set rating = 4 where name = 'Dominos';
UPDATE 1
=*> update pizzeria set rating = 4 where name = 'Pizza Hut';
UPDATE 1
=*> commit;
COMMIT
```

- The result of Session #2 (should be error is deadlock detected)

```
=> begin;
BEGIN
=*> update pizzeria set rating = 4 where name = 'Pizza Hut';
UPDATE 1
=*> update pizzeria set rating = 4 where name = 'Dominos';
ERROR: deadlock detected
DETAIL: ...
=*> commit;
ROLLBACK
```

Yes

Fails 

Cheat

Forbidden functions

Empty work

### Record of the online-review

[Select a file](#) or drag it here

Allowed file types are: .avi, .mov, .mp4, .webm and file size: max 300 MiB

### Comment

Leave a comment...

 Review