

Lab 10. Anna Startseva BS19-04

Ex.1: read committed.

Transaction isolation level is read committed, and it prohibits dirty read (read of uncommitted changes), so, results after 4 steps are different because changes are uncommitted:

```
dvdrental=# begin;
BEGIN
dvdrental=# show transaction_isolation;
transaction_isolation
-----
read committed
(1 строка)

dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      1 | jones   | Alice Jones       |      82 |      1
      2 | bitdiddl | Ben Bitdiddle     |      65 |      1
      3 | mike    | Michael Dole      |      73 |      2
      4 | alyssa  | Alyssa P. Hacker  |      79 |      3
      5 | bbrown  | Bob Brown         |     100 |      3
(5 строк)

dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      1 | jones   | Alice Jones       |      82 |      1
      2 | bitdiddl | Ben Bitdiddle     |      65 |      1
      3 | mike    | Michael Dole      |      73 |      2
      4 | alyssa  | Alyssa P. Hacker  |      79 |      3
      5 | bbrown  | Bob Brown         |     100 |      3
(5 строк)
```

Terminal 1

```

dvdrental=# begin;
BEGIN
dvdrental=# show transaction_isolation;
transaction_isolation
-----
read committed
(1 строка)

dvdrental=# update lab10 set username = 'ajones' where fullname = 'Alice Jones';
UPDATE 1
dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      2 | bitdiddl | Ben Bitdiddle      |      65 |        1
      3 | mike     | Michael Dole       |      73 |        2
      4 | alyssa   | Alyssa P. Hacker   |      79 |        3
      5 | bbrown   | Bob Brown          |     100 |        3
      1 | ajones   | Alice Jones        |      82 |        1
(5 строк)

```

Terminal 2

After committing the changes results became the same because transaction isolation level was "read committed", and non-repeatable read appeared. Transaction from the first terminal read row that was modified by transaction from the second terminal:

```

dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      2 | bitdiddl | Ben Bitdiddle      |      65 |        1
      3 | mike     | Michael Dole       |      73 |        2
      4 | alyssa   | Alyssa P. Hacker   |      79 |        3
      5 | bbrown   | Bob Brown          |     100 |        3
      1 | ajones   | Alice Jones        |      82 |        1
(5 строк)

```

Terminal 1

```

dvdrental=# commit;
COMMIT
dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      2 | bitdiddl | Ben Bitdiddle      |      65 |        1
      3 | mike     | Michael Dole       |      73 |        2
      4 | alyssa   | Alyssa P. Hacker   |      79 |        3
      5 | bbrown   | Bob Brown          |     100 |        3
      1 | ajones   | Alice Jones        |      82 |        1
(5 строк)

```

Terminal 2

After step 8 second terminal was stuck - it was waiting for other changes to be committed. So, after committing changes from the first terminal "update" query from the second terminal was done. Since step 10 is rollback in the second terminal, only changes from the first terminal were saved, and final results after all steps are the same:

```
dvdrental=# begin;
BEGIN
dvdrental=# update lab10 set balance = balance + 10 where fullname = 'Alice Jones';
UPDATE 1
dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      2 | bitdiddl | Ben Bitdiddle      |      65 |      1
      3 | mike     | Michael Dole       |      73 |      2
      4 | alyssa   | Alyssa P. Hacker   |      79 |      3
      5 | bbrown   | Bob Brown          |     100 |      3
      1 | ajones   | Alice Jones        |      92 |      1
(5 строк)

dvdrental=# commit;
COMMIT
dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      2 | bitdiddl | Ben Bitdiddle      |      65 |      1
      3 | mike     | Michael Dole       |      73 |      2
      4 | alyssa   | Alyssa P. Hacker   |      79 |      3
      5 | bbrown   | Bob Brown          |     100 |      3
      1 | ajones   | Alice Jones        |      92 |      1
(5 строк)
```

Terminal 1

```

dvdrental=# begin;
BEGIN
dvdrental=# update lab10 set balance = balance + 20 where fullname = 'Alice Jones';
UPDATE 1
dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      2 | bitdiddl | Ben Bitdiddle      |      65 |        1
      3 | mike     | Michael Dole       |      73 |        2
      4 | alyssa   | Alyssa P. Hacker   |      79 |        3
      5 | bbrown   | Bob Brown          |     100 |        3
      1 | ajones   | Alice Jones        |     112 |        1
(5 строк)

dvdrental=# rollback;
ROLLBACK
dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      2 | bitdiddl | Ben Bitdiddle      |      65 |        1
      3 | mike     | Michael Dole       |      73 |        2
      4 | alyssa   | Alyssa P. Hacker   |      79 |        3
      5 | bbrown   | Bob Brown          |     100 |        3
      1 | ajones   | Alice Jones        |      92 |        1
(5 строк)

```

Terminal 2

Ex.1: repeatable read.

Results after 4 steps are the same as in "ex.1: read committed":

```

dvdrental=# begin;
BEGIN
dvdrental=# set transaction isolation level repeatable read;
SET
dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      2 | bitdiddl | Ben Bitdiddle      |      65 |        1
      3 | mike     | Michael Dole       |      73 |        2
      4 | alyssa   | Alyssa P. Hacker   |      79 |        3
      5 | bbrown   | Bob Brown          |     100 |        3
      1 | jones    | Alice Jones        |      82 |        1
(5 строк)

dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      2 | bitdiddl | Ben Bitdiddle      |      65 |        1
      3 | mike     | Michael Dole       |      73 |        2
      4 | alyssa   | Alyssa P. Hacker   |      79 |        3
      5 | bbrown   | Bob Brown          |     100 |        3
      1 | jones    | Alice Jones        |      82 |        1
(5 строк)

```

Terminal 1

```

dvdrental=# begin;
BEGIN
dvdrental=# set transaction isolation level repeatable read;
SET
dvdrental=# update lab10 set username = 'ajones' where fullname = 'Alice Jones';
UPDATE 1
dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      2 | bitdiddl | Ben Bitdiddle      |      65 |       1
      3 | mike     | Michael Dole       |      73 |       2
      4 | alyssa   | Alyssa P. Hacker   |      79 |       3
      5 | bbrown   | Bob Brown          |     100 |       3
      1 | ajones   | Alice Jones        |      82 |       1
(5 строк)

```

Terminal 2

Results after step 5 are also different because transaction isolation level is repeatable read, and non-repeatable read is prohibited, so, changes from the second terminal does not affect on the output of the first terminal:

```

dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      2 | bitdiddl | Ben Bitdiddle      |      65 |       1
      3 | mike     | Michael Dole       |      73 |       2
      4 | alyssa   | Alyssa P. Hacker   |      79 |       3
      5 | bbrown   | Bob Brown          |     100 |       3
      1 | jones    | Alice Jones        |      82 |       1
(5 строк)

```

Terminal 1

```

dvdrental=# commit;
COMMIT
dvdrental=# select * from lab10;
 user_id | username |      fullname      | balance | group_id
-----+-----+-----+-----+-----
      2 | bitdiddl | Ben Bitdiddle      |      65 |       1
      3 | mike     | Michael Dole       |      73 |       2
      4 | alyssa   | Alyssa P. Hacker   |      79 |       3
      5 | bbrown   | Bob Brown          |     100 |       3
      1 | ajones   | Alice Jones        |      82 |       1
(5 строк)

```

Terminal 2

After step 8 the second terminal was stuck, and after committing changes from the first terminal in terminal 2 error was displayed: "ERROR: failed to serialize access due to concurrent change". It occurs because of repeatable read level of transaction isolation which does not allow non-repeatable read:

```
dvdrental=# begin;  
BEGIN  
dvdrental=# set transaction isolation level repeatable read;  
SET  
dvdrental=# update lab10 set balance = balance + 10 where fullname = 'Alice Jones';  
UPDATE 1  
dvdrental=# commit;  
COMMIT
```

Terminal 1

```
dvdrental=# begin;  
BEGIN  
dvdrental=# set transaction isolation level repeatable read;  
SET  
dvdrental=# update lab10 set balance = balance + 20 where fullname = 'Alice Jones';  
ОШИБКА: не удалось сериализовать доступ из-за параллельного изменения  
dvdrental=# rollback;  
ROLLBACK
```

Terminal 2

Ex.2: read committed.

Moving Bob to group 2 in the second terminal does not affect transaction in the first terminal because it was uncommitted change, so, Bob's balance was not increased. But after committing this update, Bob is in the group 2 with the same balance. Lost update event occurred in these transactions, and it was not prevented by read committed level.

```

dvdrental=# begin;
BEGIN
dvdrental=# select * from lab10 where group_id = 2;
 user_id | username |   fullname   | balance | group_id
-----+-----+-----+-----+-----
      3 | mike    | Michael Dole |      73 |      2
(1 строка)

dvdrental=# select * from lab10 where group_id = 2;
 user_id | username |   fullname   | balance | group_id
-----+-----+-----+-----+-----
      3 | mike    | Michael Dole |      73 |      2
(1 строка)

dvdrental=# update lab10 set balance = balance + 15 where group_id = 2;
UPDATE 1
dvdrental=# commit;
COMMIT
dvdrental=# select * from lab10;
 user_id | username |   fullname   | balance | group_id
-----+-----+-----+-----+-----
      1 | jones    | Alice Jones   |      82 |      1
      2 | bitdiddl | Ben Bitdiddle |      65 |      1
      4 | alyssa   | Alyssa P. Hacker |      79 |      3
      5 | bbrown   | Bob Brown     |     100 |      2
      3 | mike     | Michael Dole  |      88 |      2
(5 строк)

```

Terminal 1

```

dvdrental=# begin;
BEGIN
dvdrental=# update lab10 set group_id = 2 where fullname = 'Bob Brown';
UPDATE 1
dvdrental=# commit;
COMMIT
dvdrental=# select * from lab10;
 user_id | username |   fullname   | balance | group_id
-----+-----+-----+-----+-----
      1 | jones    | Alice Jones   |      82 |      1
      2 | bitdiddl | Ben Bitdiddle |      65 |      1
      4 | alyssa   | Alyssa P. Hacker |      79 |      3
      5 | bbrown   | Bob Brown     |     100 |      2
      3 | mike     | Michael Dole  |      88 |      2
(5 строк)

```

Terminal 2

Ex.2: repeatable read.

Results are the same as in the previous case, although the transaction isolation level was changed. Repeatable read also prohibits reading uncommitted data:

```

dvdrental=# begin;
BEGIN
dvdrental=# set transaction isolation level repeatable read;
SET
dvdrental=# select * from lab10 where group_id = 2;
 user_id | username |   fullname   | balance | group_id
-----+-----+-----+-----+-----
      3 | mike    | Michael Dole |      73 |        2
(1 строка)

dvdrental=# select * from lab10 where group_id = 2;
 user_id | username |   fullname   | balance | group_id
-----+-----+-----+-----+-----
      3 | mike    | Michael Dole |      73 |        2
(1 строка)

dvdrental=# update lab10 set balance = balance + 15 where group_id = 2;
UPDATE 1
dvdrental=# commit;
COMMIT
dvdrental=# select * from lab10;
 user_id | username |   fullname   | balance | group_id
-----+-----+-----+-----+-----
      1 | jones    | Alice Jones   |      82 |        1
      2 | bitdiddl | Ben Bitdiddle |      65 |        1
      4 | alyssa   | Alyssa P. Hacker |      79 |        3
      5 | bbrown   | Bob Brown     |     100 |        2
      3 | mike     | Michael Dole  |      88 |        2
(5 строк)

```

Terminal 1

```

dvdrental=# begin;
BEGIN
dvdrental=# set transaction isolation level repeatable read;
SET
dvdrental=# update lab10 set group_id = 2 where fullname = 'Bob Brown';
UPDATE 1
dvdrental=# commit;
COMMIT
dvdrental=# select * from lab10;
 user_id | username |   fullname   | balance | group_id
-----+-----+-----+-----+-----
      1 | jones    | Alice Jones   |      82 |        1
      2 | bitdiddl | Ben Bitdiddle |      65 |        1
      4 | alyssa   | Alyssa P. Hacker |      79 |        3
      5 | bbrown   | Bob Brown     |     100 |        2
      3 | mike     | Michael Dole  |      88 |        2
(5 строк)

```

Terminal 2

Ex.3: repeatable read.

Error occurred: "ERROR: failed to serialize access due to concurrent change". It occurred because of querying uncommitted data from database while doing update in the first terminal:

```
dvdrental=# begin;
BEGIN
dvdrental=# set transaction isolation level repeatable read;
SET
dvdrental=# select sum(balance) from lab10 where group_id = 2;
sum
-----
 73
(1 строка)

dvdrental=# select * from lab10 where group_id = 2;
 user_id | username |  fullname  | balance | group_id
-----+-----+-----+-----+-----
      3 | mike    | Michael Dole |      73 |      2
(1 строка)

dvdrental=# update lab10 set balance = balance + a.b from (select sum(balance) as b from lab10) as a;
ОШИБКА: не удалось сериализовать доступ из-за параллельного изменения
dvdrental=# commit;
```

Terminal 1

```
dvdrental=# begin;
BEGIN
dvdrental=# set transaction isolation level repeatable read;
SET
dvdrental=# update lab10 set group_id = 2 where fullname = 'Bob Brown';
UPDATE 1
dvdrental=# commit;
COMMIT
dvdrental=# select * from lab10;
 user_id | username |  fullname  | balance | group_id
-----+-----+-----+-----+-----
      1 | jones    | Alice Jones |      82 |      1
      2 | bitdiddl | Ben Bitdiddle |      65 |      1
      3 | mike    | Michael Dole |      73 |      2
      4 | alyssa   | Alyssa P. Hacker |      79 |      3
      5 | bbrown   | Bob Brown   |     100 |      2
(5 строк)
```

Terminal 2

Ex.3: serializable.

Results are the same as in the previous case:

```

dvdrental=# begin;
BEGIN
dvdrental=# set transaction isolation level serializable;
SET
dvdrental=# select sum(balance) from lab10 where group_id = 2;
sum
-----
73
(1 строка)

dvdrental=# select * from lab10 where group_id = 2;
 user_id | username |  fullname  | balance | group_id 
-----+-----+-----+-----+-----
      3 | mike    | Michael Dole |      73 |        2 
(1 строка)

dvdrental=# update lab10 set balance = balance + a.b from (select sum(balance) as b from lab10) as a;
ОШИБКА: не удалось сериализовать доступ из-за параллельного изменения
dvdrental=# commit;

```

Terminal 1

```

dvdrental=# begin;
BEGIN
dvdrental=# set transaction isolation level serializable;
SET
dvdrental=# update lab10 set group_id = 2 where fullname = 'Bob Brown';
UPDATE 1
dvdrental=# commit;
COMMIT

```

Terminal 2