1. Created table

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 60 msec.

2. Executing transaction 1:

```
TITUE CLANDACCION
9
     BEGIN;
     SELECT txid_current();
10
11
12 ▼ INSERT INTO public.employee (name, s
     VALUES ('Alice', 'Not fired');
13
14
15 ➤ SELECT *, xmin, xmax
     FROM public.employee e;
16
17
18
     COMMIT;
19
```

Data Output Messages Notifications

BEGIN

Query returned successfully in 60 msec.

```
SELECT txid_current();

10
11
12 \ INSERT INTO public.employee (name, status)
13 VALUES ('Alice', 'Not fired');

Data Output Messages Notifications

The status of the sta
```

This is current transaction_id

```
INSERT INTO public.employee (name, status)
VALUES ('Alice', 'Not fired');

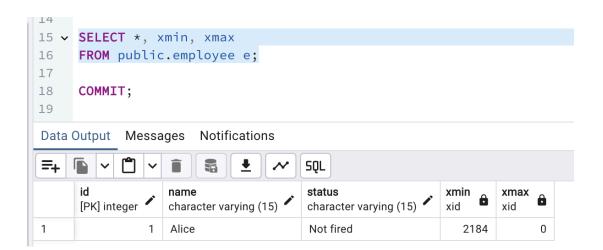
SELECT *, xmin, xmax
FROM public.employee e;

COMMIT;

Data Output Messages Notifications

INSERT 0 1

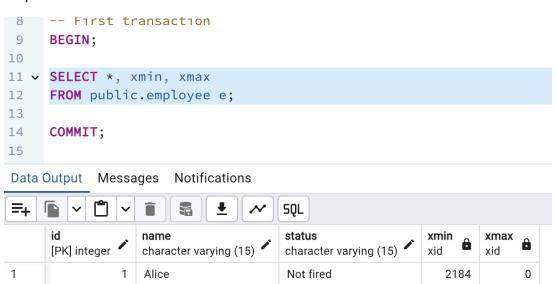
Query returned successfully in 57 msec.
```



After executing insert and select statements on the 1 option, doing select on the 2 option does not show anything, transaction is not committed.



All queries executed as in lecture

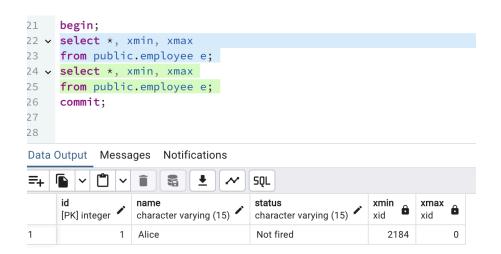


Now it is visible as well because we commit our transaction

Query returned successfully in 63 msec.

Transaction 2.

Beginning each transaction, and seeing transaction ids on both transactions



```
29
     begin;
30
      select txid_current();
31
32 v delete from public.employee
33 where id = 1;
34 v select *, xmin, xmax
35 from public.employee e;
Data Output Messages
                           Notifications
=+
      txid_current
       bigint
1
               2187
 20
 21
      begin;
 22 v select *, xmin, xmax
      from public.employee e;
 24 v select *, xmin, xmax
      from public.employee e;
 25
      commit.
 Data Output Messages Notifications
                                  SQL
                 name
                                  status
                                                   xmin
                                                          xmax
      [PK] integer
                 character varying (15)
                                  character varying (15)
                                                   xid
                                                          xid
                                  Not fired
                 Alice
                                                      2190
                                                               0
```

```
delete from public.employee
where id = 1;
select *, xmin, xmax
from public.employee e;
commit;
```

Data Output Messages Notifications

DELETE 1

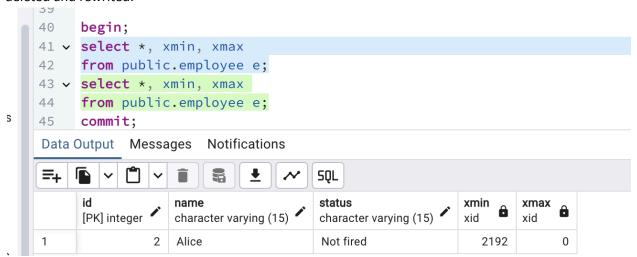
Query returned successfully in 58 msec.



Deleting row from the 2 option and seeing on the 1 option, that xmax(deleting id) is updated to the transaction id from the right side. So in the 2 option, it is still able to see the row, even though it is deleted. After committing 2 side, the row is deleted

Transaction 3.

In the third transaction, when on the 2 option I updated the row and it showed the updated version, Alice-Fired, but on the 1 option, since it is not committed transaction, i can see that xmax is updated to transaction id from the right side, means that this column is deleted, even its updated, it is deleted and rewrited.



```
begin;
select txid_current();

update public.employee
set status = 'Fired'

where id = 2;
select *, xmin, xmax
from public.employee e;

Data Output Messages Notifications

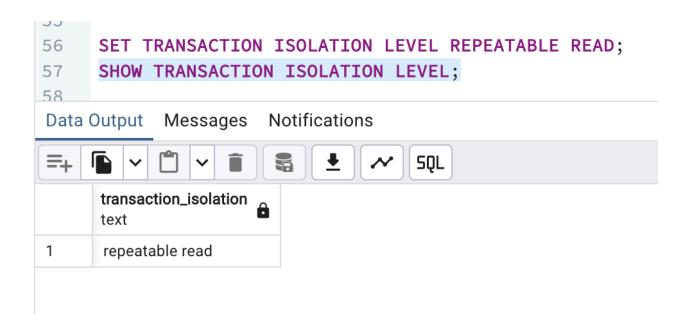
UPDATE 1

Query returned successfully in 48 msec.
```

After committing, it shows that alice fired updated on the 1 option as well.

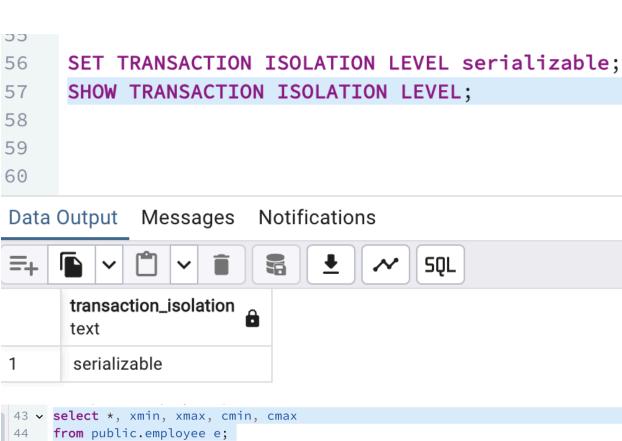


3.Set isolation level to repeatable read



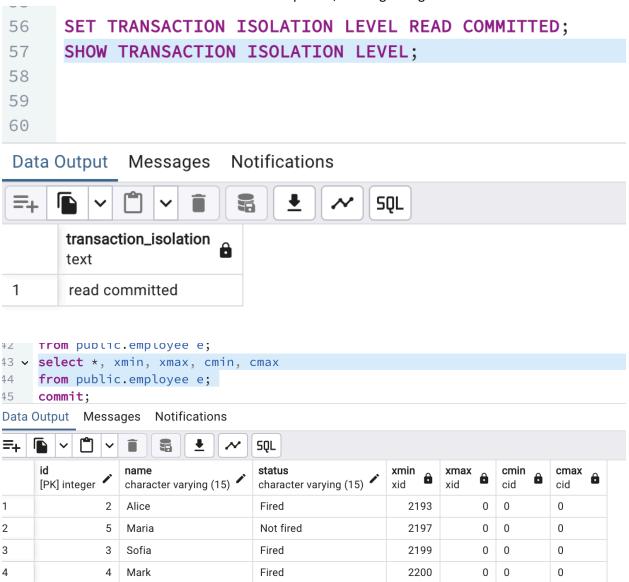
While inserting with Repeatable Read, I can see what's happening with cmin and cmax; they are not changing







Here I tried read committed level and run some updates, nothing change in xmax



With read committed, it is not prevented to lost update anomalies because it only ensures that each statement sees a consistent snapshot of the database.

concurrent transactions can overwrite each other's changes without awareness. In this case, last one that updated was fired and Mark status was set to Fired.