author: @timusfed

- Complex transaction:

***remark**: the problem can occur when accessing data for different purposes at the same time (for example, deleting and reading), but this can be prevented by creating a savepoint before the commit*

```
begin;
      savepoint sp1;
            delete from carrier where title = 'FIX';
      savepoint sp2;
            update person set fullname = 'Supercool Guy' where fullname
= 'Timushev Fedor';
      savepoint sp3;
            delete from carrier where title = 'MHD';
      rollback to sp3;
      release savepoint sp3;
commit;
```

- View:

***remark:** A view that returns the trains that are currently assigned to travel*

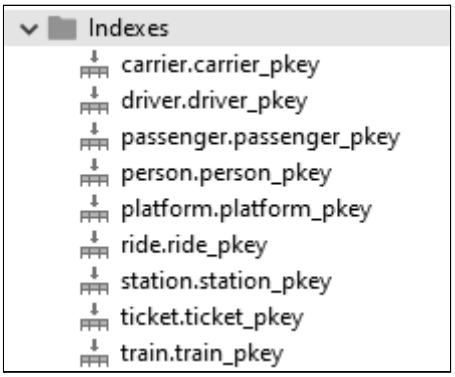
```
create view busy_transport
as select train.id_prefix
      from train
intersect
select ride.train_p
      from ride
```

- Index:

***remark**: indexes are used to quickly retrieve data from a table. They are not visible to users. Next: is an example of creating an index on the "carrier" table:*

```
create index carrier_title on carrier (title);
```

Similarly, indexes were created for each table, by their primary key. Query efficiency has increased: for example, it takes 150-250ms to display all records of the "carrier" table instead of 2-3 seconds (it has 32k of data).



- Trigger:

remark: *creating a trigger in PostgreSQL means creating a custom function that the trigger will use. This trigger creates an empty record of type "passenger" with 0 discount when a new record "person" is added*

```
create function auto_passenger()
returns trigger as
$$
begin
      insert into passenger(person_passport, discount)
      values (new.id_passport_num, 0);
return new;
end;
$$
language plpgsql;
```

```
create trigger autoPassenger
after insert
on person
for each row
execute procedure auto_passenger()
```