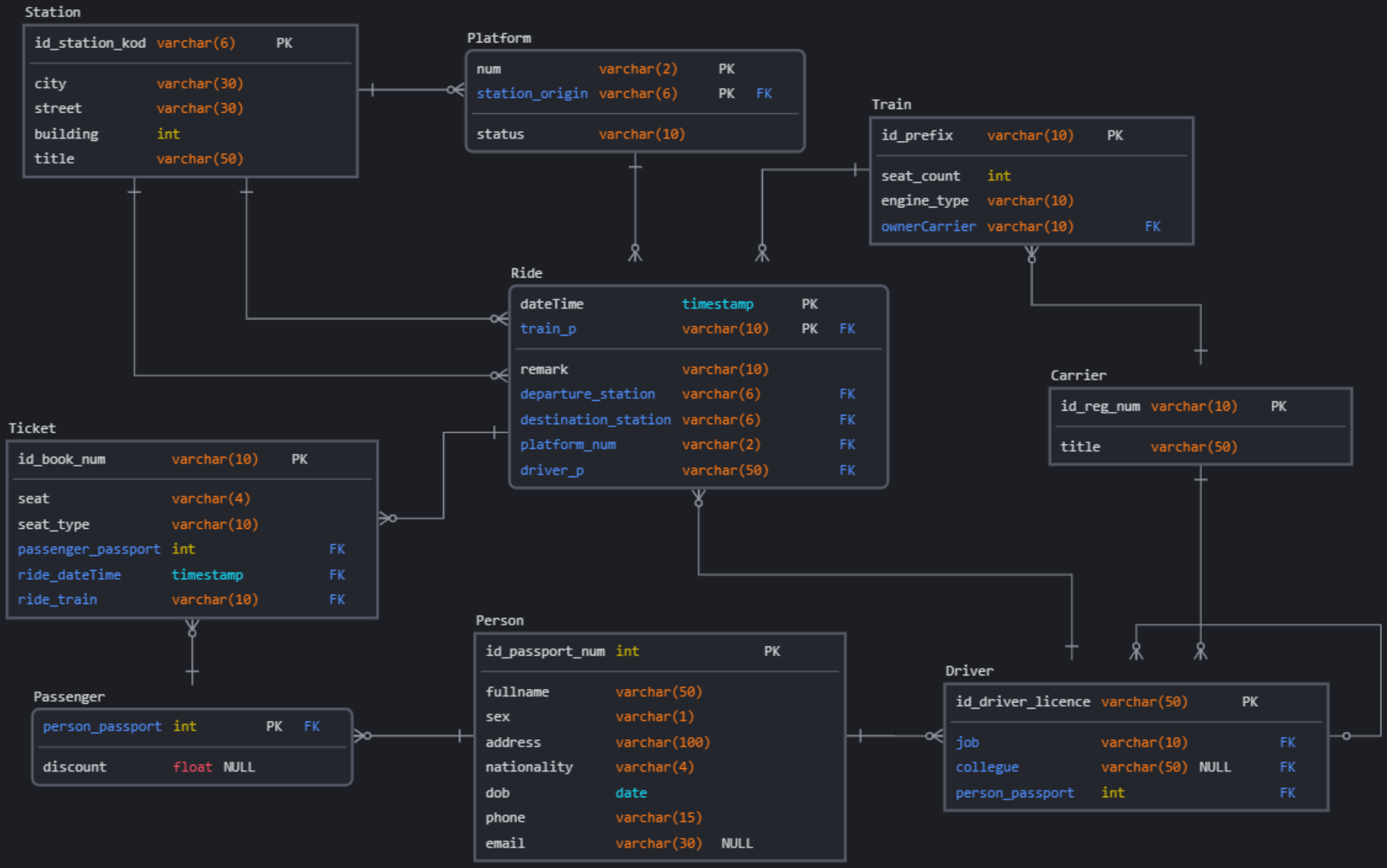
author: @timusfed



SQL queries [tables creation]:

1. Person table

```
create table if not exists Person
(
     id_passport_num INT not null primary key,
     fullname VARCHAR(50) not null,
     sex VARCHAR(1) not null,
     address VARCHAR(100) not null,
     nationality VARCHAR(4) not null,
     dob DATE not null,
     phone VARCHAR(15) not null,
     email VARCHAR(30)
);
```

2. Carrier table

```
create table if not exists Carrier
(
     id_reg_num VARCHAR(10) not null primary key,
     title VARCHAR(50) not null
);
```

3. Driver table

```
create table if not exists Driver
(
     id_driver_licence VARCHAR(50) not null primary key,
     job VARCHAR(10) not null,
     colleague VARCHAR(50),
     person_passport INT not null,
     constraint fk_passport foreign key (person_passport)
          references Person(id_passport_num) on update cascade on delete cascade,
     constraint fk_job foreign key (job)
          references Carrier(id_reg_num) on update cascade on delete set null,
     constraint fk_collegue foreign key (colleague)
          references Driver(id_driver_licence) on update cascade on delete set null
);
```

4. Passenger table

```
create table if not exists Passenger
(
     person_passport INT not null primary key,
     discount FLOAT,
     constraint fk_passport foreign key (person_passport)
          references Person(id_passport_num) on update cascade on delete cascade
);
```

5. Train table

```
create table if not exists Train
(
     id_prefix VARCHAR(10) not null primary key,
     seat_count INT not null,
     engine_type VARCHAR(10) not null,
     ownerCarrier VARCHAR(10) not null,
     constraint fk_owner foreign key (ownerCarrier)
          references Carrier(id_reg_num) on update cascade on delete set null
);
```

6. Station table

```
create table if not exists Station
(
     id_station_kod VARCHAR(6) not null primary key,
     city VARCHAR(30) not null,
     street VARCHAR(30) not null,
     building INT not null,
     title VARCHAR(50) not null
);
```

7. Platform table

```
create table if not exists Platform
(
     num VARCHAR(2) not null,
     station_origin VARCHAR(6) not null,
     status VARCHAR(10) not null,
     primary key (num, station_origin),
     constraint fk_origin foreign key (station_origin)
          references Station(id_station_kod) on update cascade on delete cascade
);
```

8. Ride table

```
create table if not exists Ride
(
     dateTime TIMESTAMP not null,
     remark VARCHAR(10) not null,
     train_p VARCHAR(10) not null,
     departure_station VARCHAR(6) not null,
     destination_station VARCHAR(6) not null,
     platform_num VARCHAR(2) not null,
     driver_p VARCHAR(50) not null,
     primary key (dateTime, train_p),
     constraint fk_departure foreign key (departure_station)
          references Station(id_station_kod) on update cascade on delete cascade,
     constraint fk_destination foreign key (destination_station)
          references Station(id_station_kod) on update cascade on delete set null,
     constraint fk_train foreign key (train_p)
          references Train(id_prefix) on update cascade on delete set null,
     constraint fk_platform foreign key (platform_num, departure_station)
          references Platform(num, station_origin) on update cascade on delete set null,
     constraint fk_driver foreign key (driver_p)
          references Driver(id_driver_licence) on update cascade on delete set null
);
```

9. Ticket table

```
create table if not exists Ticket
(
     id_book_num VARCHAR(10) not null primary key,
     seat VARCHAR(4) not null,
     seat_type VARCHAR(10) not null,
     passenger_passport INT not null,
     ride_dateTime TIMESTAMP not null,
     ride_train VARCHAR(10) not null,
     constraint fk_owner foreign key (passenger_passport)
          references Passenger(person_passport) on update cascade on delete cascade,
     constraint fk_ride foreign key (ride_dateTime, ride_train)
          references Ride(dateTime, train_p) on update cascade on delete cascade
);
```

SQL queries [queries examples]:

1. left join → 'display all people, if the person is a driver - indicate the license number'

```
select person.fullname, driver.id_driver_licence from person
left join driver on person.id_passport_num = driver.person_passport
```

fullname	id_driver_licence
Timushev Fedor	DL12345678
Random Matej	

2. right join → 'display all stations, if it has free platform - indicate it'

```
select station.title, platform.num from platform
right join station on platform.station_origin = station.id_station_kod and platform.status = 'free'
```

title	num
Nádraží Holešovice	1A
Hlavní Nádraží	

3. condition selection → 'display all platfroms from Nadrazi Holesovice, which are currently occupied'

```
select platform.num from platform
where platform.station_origin in
     (
          select station.id_station_kod from station
          where station.title = 'Nádraží Holešovice'
     ) and platform.status = 'occupied'
```

num
2A

4. aggregate function (let it be COUNT) → 'show count of trains owned by carrier with DA7564HF registration number'

```
select count(train.ownercarrier) from train
where train.ownercarrier = 'DA7564HF'
```

```
count|
-----+
    1|
```

5. aggregate function (let it be AVG) → 'show average discount of all passengers'

```
select avg(passenger.discount) from passenger
```

```
avg|
---+
8.5|
```

6. sorting → 'show all persons in reverse alphabetical order'

```
select * from person
order by person.fullname desc
```

```
id_passport_num|fullname      |sex|address|nationality|dob       |phone      |email               |
---------------+--------------+---+-------+-----------+----------+-----------+--------------------+
     1423506978|Timushev Fedor|M  |Prague |Rus        |1999-06-17|778 563 867|timusfed@fel.cvut.cz|
     1522546989|Random Matej  |M  |Brno   |Cz         |1986-04-13|778 765 534|random@fel.cvut.cz  |
```

7. pagination → 'i need to control carriers, ordered by their registration number. yesterday i checked only 2 so i can skip them'

```
select * from carrier order by carrier.id_reg_num offset 2
```

```
id_reg_num|title|
----------+-----+
CRN57365  |ALG  |
CRN71843  |ASR  |
CRN96023  |DBS  |
DA7564HF  |MHD  |
```

8. union → 'show all passengers names and drivers names at the same time'

```
select person.fullname from person, passenger
     where person.id_passport_num = passenger.person_passport
union
select person.fullname from person, driver
     where person.id_passport_num = driver.person_passport
```

```
fullname      |
--------------+
Timushev Fedor|
Random Matej  |
```

9. intersect → 'show all train codes, which are assigned to any ride'

```
select train.id_prefix from train
intersect
select ride.train_p from ride
```

```
id_prefix |
----------+
TP87654321|
```

10. nested select → 'try to find the carrier number, which ends with 5 from the list of carriers, which title starts with "A"'

```
select * from carrier
where carrier.title in (
     select carrier.title from carrier
     where carrier.title like 'A%'
) and carrier.id_reg_num like '%5'
```

```
id_reg_num|title|
----------+-----+
CRN57365  |ALG  |
```