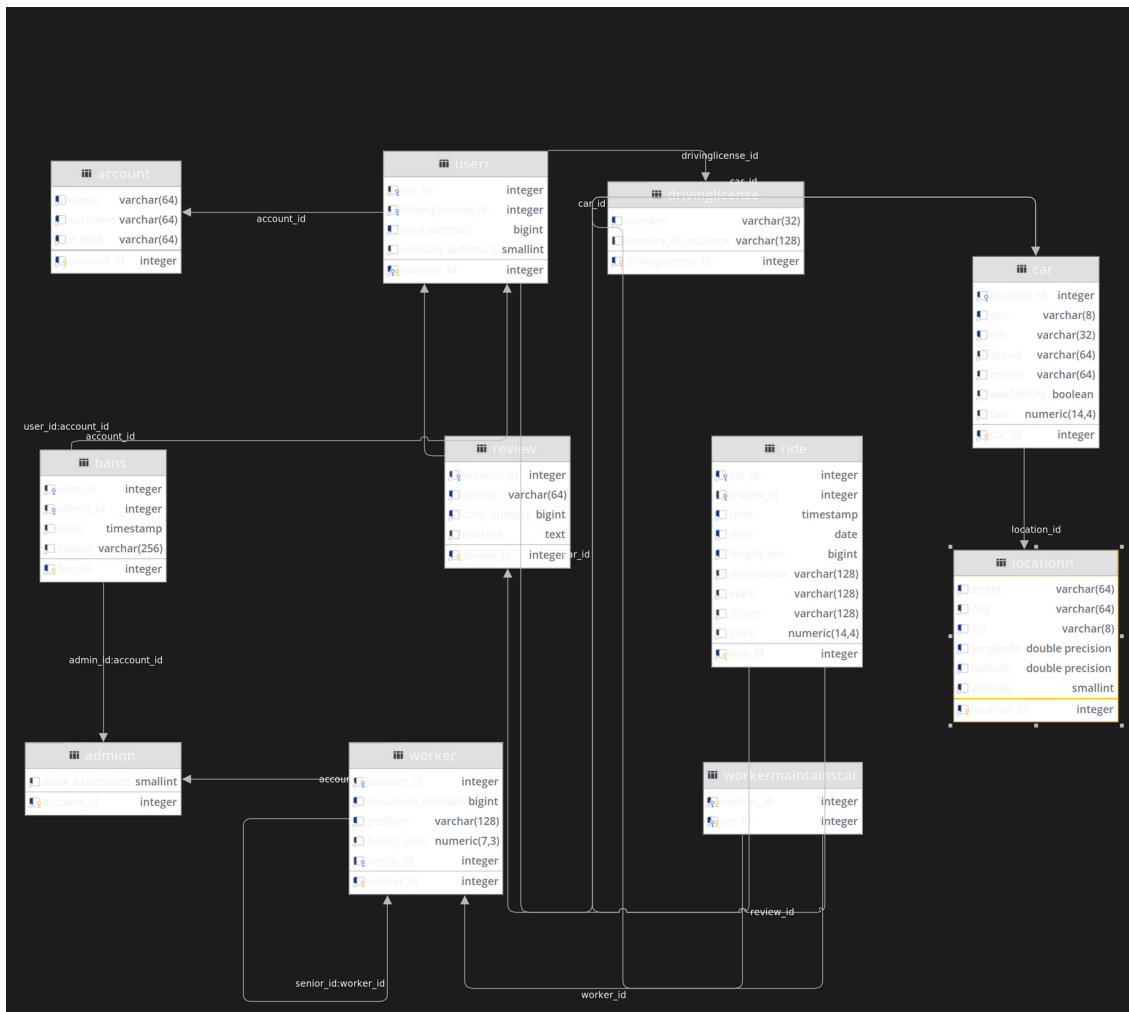


ER:



RM:

Location(street, city, ZIP)

GPS_coords(longitude, latitude, altitude)

LocationHasGps_coords(street, city, ZIP, longitude, latitude, altitude)

FK: (street, city, ZIP) ⊆ Location(street, city, ZIP)

FK: (longitude, latitude, altitude) ⊆ GPS_coords(longitude, latitude, altitude)

Car(SPZ, VIN, make, model, availability, street, city, ZIP)

FK: (street, city, ZIP) ⊆ Location(street, city, ZIP)

Ride(time, date, length, SPZ, price, driver, start, destination)

FK: (SPZ) ⊆ Car(SPZ)

Worker(insurance_number, position, hourly rate, name, surname, e-mail)

FK: (name, surname, e-mail) ⊆ Admin(name, surname, e-mail)

WorkerMaintainsCar(insurance_number, position, SPZ)

FK: (insurance_number, position) ⊆ Worker(insurance_number, position)

FK: (SPZ) ⊆ Car(SPZ)

Senior_to(insurance_number, position, senior)

FK: (insurance_number, position) ⊆ Worker(insurance_number, position)

FK: (senior) ⊆ Worker(insurance_number, position)

Account(name, surname, e-mail, birth date)

User(name, surname, e-mail, card_number, driving_license, conduct_summary, admin, time, reason, SPZ)

FK: (name, surname, e-mail) ⊆ Account(name, surname, e-mail)

FK: (admin) ⊆ Admin(name, surname, e-mail)

FK: (SPZ) ⊆ Car(SPZ)

Admin(name, surname, e-mail, work_experience)

FK: (name, surname, e-mail) ⊆ Account(name, surname, e-mail)

Review(author, card_number, content, time, date, length)

FK: (card_number) ⊆ User(card_number)

FK: (time, date, length) ⊆ Ride(time, date, length)

SQL dotazy pro vytvoření databáze včetně adekvátních integritních omezení:

```
CREATE TABLE Locationn
(
    location_id serial PRIMARY KEY,
    street      varchar(64)    NOT NULL,
    city        varchar(64)    NOT NULL,
    ZIP         varchar(8)     NOT NULL,
    longitude   double precision NOT NULL,
    latitude    double precision NOT NULL,
    altitude    smallint       NOT NULL,
    UNIQUE (street, city, ZIP),
    UNIQUE (longitude, latitude, altitude)
);
```

```
CREATE TABLE Car
(
    car_id      serial PRIMARY KEY,
    location_id integer      NOT NULL
                DEFAULT 1
                REFERENCES Locationn
                ON UPDATE CASCADE ON DELETE SET DEFAULT,
    SPZ         varchar(8)   NOT NULL,
    VIN         varchar(32)  NOT NULL,
    brand       varchar(64)  NOT NULL,
    model       varchar(64)  NOT NULL,
    availability boolean     NOT NULL,
    fare        numeric(14, 4) NOT NULL,
    UNIQUE (SPZ, VIN)
);
```

ON UPDATE CASCADE - protože chceme zajistit integritu dat při updatování.

ON DELETE SET DEFAULT – defaultní hodnota je location_id = 1, což je základní lokalita, která je vozovým parkem pro auta, protože v případě odstranění z uvedené polohy bude vozidlo přesměrováno do vozového parku.

```
CREATE TABLE Account
(
    account_id serial PRIMARY KEY,
    name        varchar(64) NOT NULL CHECK ( name <> '' ),
    surname     varchar(64) NOT NULL CHECK ( surname <> '' ),
    e_mail      varchar(64) NOT NULL CHECK ( e_mail <> '' ),
    UNIQUE (name, surname, e_mail)
);
```

```
CREATE TABLE Userr
(
    account_id    int PRIMARY KEY REFERENCES Account,
    car_id        integer NOT NULL
                DEFAULT 1
                REFERENCES Car
                ON UPDATE CASCADE ON DELETE SET DEFAULT,
    card_number   bigint NOT NULL UNIQUE,
    conduct_summary smallint NOT NULL
                CHECK ( conduct_summary BETWEEN 0 AND 10000 )
);
```

car_id:

- ON UPDATE CASCADE - protože chceme zajistit integritu dat při updatování.
- ON DELETE SET DEFAULT – default hodnota je car_id = 1, aby při odebrání vozidla uživatel měl automaticky nainstalované auto s číslem 1, což je Toyota Corolla.

CHECK:

- souhrnné chování uživatele, které by mělo být mezi 0 body, což je nejhorší chování uživatele, a 10 000 – nejlepší.

```
CREATE TABLE DrivingLicense
```

```
(
    drivingLicense_id serial PRIMARY KEY,
    user_id           integer NOT NULL REFERENCES Userr ON UPDATE CASCADE,
    number            varchar(32) NOT NULL UNIQUE,
    country_of_issuance varchar(128)
);
ON UPDATE CASCADE - protože chceme zajistit integritu dat při updatování.
```

```
CREATE TABLE Ride
```

```
(
    ride_id    serial PRIMARY KEY,
    car_id     integer NOT NULL REFERENCES Car
                ON UPDATE CASCADE ON DELETE CASCADE,
    time       timestamp NOT NULL,
    date       date      NOT NULL,
    length_sec bigint     NOT NULL,
    destination varchar(128) NOT NULL,
    start      varchar(128) NOT NULL,
    driver     varchar(128) NOT NULL,
    price      numeric(14, 4) NOT NULL CHECK ( price >= 0 ),
    UNIQUE (time, date, length_sec, car_id)
);
```

car_id:

- ON UPDATE CASCADE - protože chceme zajistit integritu dat při updatování.
- ON DELETE CASCADE - aby se při odstraňování auta odstranily i související jízdy.

CREATE TABLE Review

```
(
    review_id serial PRIMARY KEY,
    account_id integer NOT NULL REFERENCES User
        ON UPDATE CASCADE,
    ride_id integer NOT NULL REFERENCES Ride
        ON UPDATE CASCADE ON DELETE CASCADE,
    author varchar(64) NOT NULL,
    card_number bigint NOT NULL,
    content text NOT NULL CHECK ( content <> '' ),
    UNIQUE (author, card_number)
);
```

account_id:

- ON UPDATE CASCADE - protože chceme zajistit integritu dat při updatování.

ride_id:

- ON UPDATE CASCADE - protože chceme zajistit integritu dat při updatování.
- ON DELETE CASCADE - protože recenze je napsána na jednu cestu, takže při odstranění cesty nemá smysl o ní nechávat recenzi, protože jízda již neexistuje.

CREATE TABLE Adminn

```
(
    account_id int PRIMARY KEY,
    work_experience smallint NOT NULL
);
```

CREATE TABLE Worker

```
(
    worker_id serial PRIMARY KEY,
    account_id integer NOT NULL
        DEFAULT 1
        REFERENCES Adminn
        ON UPDATE CASCADE ON DELETE SET DEFAULT,
    insurance_number bigint NOT NULL,
    position varchar(128) NOT NULL,
    hourly_rate numeric(7, 3) NOT NULL CHECK ( hourly_rate > 0 ),
    senior_id integer REFERENCES Worker
        ON UPDATE CASCADE ON DELETE SET NULL,
    UNIQUE (insurance_number, position)
);
```

account_id:

- ON UPDATE CASCADE - protože chceme zajistit integritu dat při updatování.
- ON DELETE SET DEFAULT - každý zaměstnanec musí mít správce, takže pokud bude nějaký Správce odstraněn, zaměstnanci, kteří byli pod jeho vedením, budou přiděleni jinému administrátorovi.

senior_id:

- ON UPDATE CASCADE - protože chceme zajistit integritu dat při updatování.
- ON DELETE SET NULL - zaměstnanec může mít nadřízeného (max. 1), pokud bude nadřízený odstraněn, pak zaměstnanec nebude mít nadřazenou osobu (NULL).

```
CREATE TABLE WorkerMaintainsCar
```

```
(
    worker_id integer NOT NULL REFERENCES Worker
        ON UPDATE CASCADE ON DELETE CASCADE,
    car_id integer NOT NULL REFERENCES Car
        ON UPDATE CASCADE ON DELETE CASCADE,
    PRIMARY KEY (worker_id, car_id)
```

```
);
```

```
worker_id && car_id:
```

- ON UPDATE CASCADE - protože chceme zajistit integritu dat při updatování.
- ON DELETE CASCADE - pokud je záznam v tabulce Worker nebo Car odstraněn, všechny přidružené záznamy v tabulce WorkerMaintainsCar jsou automaticky odstraněny, aby byla zachována integrita databáze.

```
CREATE TABLE Bans
```

```
(
    ban_id serial PRIMARY KEY,
    user_id integer NOT NULL REFERENCES User
        ON UPDATE CASCADE,
    admin_id integer NOT NULL REFERENCES Admin
        ON UPDATE CASCADE,
    time timestamp NOT NULL,
    reason varchar(256) NOT NULL CHECK ( reason <> '' )
```

```
);
```

```
user_id && admin_id:
```

- ON UPDATE CASCADE - protože chceme zajistit integritu dat při updatování.

SQL dotazy pro získání údajů z databáze:

vnější spojení tabulek:

```
SELECT *
```

```
FROM account
```

```
LEFT OUTER JOIN adminn ON account.account_id = adminn.account_id;
```

	account.account_id	name	surname	e_mail	adminn.account_id	work_experience
1	4	Rory	Russel	rorruss@gmail.com	4	5
2	5	Luc	White	lucwhite@gmail.com	5	1
3	6	Greg	Waller	rorruss@gmail.com	6	10
4	2	Oleg	Smirnov	olsmirn@gmail.com	<null>	<null>
5	1	Sergey	Kozlov	serkoz@gmail.com	<null>	<null>
6	3	Andrew	Smith	andsmith@gmail.com	<null>	<null>

vybere všechny záznamy z tabulky “account” a pokusí se spojit každý záznam s odpovídajícími záznamy z tabulky “admin” na základě account_id v “account” a account_id v “adminn”. Pokud v tabulce “admin” není shoda, zobrazí se hodnoty NULL pro sloupce z tabulky zákazů.

```
SELECT *
FROM account
RIGHT OUTER JOIN adminn ON account.account_id = adminn.account_id;
```

	account.account_id	name	surname	e_mail	adminn.account_id	work_experience
1	4	Rory	Russel	rorruss@gmail.com	4	5
2	5	Luc	White	lucwhite@gmail.com	5	1
3	6	Greg	Waller	rorruss@gmail.com	6	10

vybere všechny záznamy z tabulky “adminn” a pokusí se spojit každý záznam s odpovídajícími záznamy z tabulky “account” na základě account_id v “account” a account_id v “adminn”. Pokud v tabulce “account” není shoda, zobrazí se hodnoty NULL pro sloupce z tabulky zákazů.

FULL OUTER JOIN bude to samé jako LEFT OUTER JOIN (u mě).

vnitřní spojení tabulek:

```
SELECT *
FROM car
INNER JOIN locationnn ON car.location_id = locationnn.location_id
LIMIT 10;
```

	car_id	car.location_id	spz	vin	brand	model	availability	fare	locationnn.location_id	street	city	zip	longitude	latitude	altitude
1	1	1	A001234	W016T8E8601091157	Toyota	Corolla	• true	0.2000	1	Olympijska 1902	Prague	16017	50.0816	14.39198	320
2	1392	601	miqdsyb	ln8eu0a8dgoyqivql	A8	2	• true	4.1700	601	78171 Magnus Ports	Johns Creek	38752	-71.6214	-64.2246	11
3	1853	821	yfj7cz5	2oufh09stwxkzae8o	Model 3	Volt	false	7.9100	821	211 Roberts Union	Ontario	69668	5.5677	31.3494	345
4	2037	68	a0voht8	pxc2xulg550ii6f26	Model Y	XC90	• true	9.6300	68	23101 Leland Tunnel	Westfield	82336	-34.9443	-30.7538	163
5	4055	943	jv3pjo1	ghwy24tk9sm7mb5sq	V90	2	• true	0.1000	943	5634 Waters Path	El Dorado Hills	16776	-85.3691	-55.1375	114
6	4667	771	qsdhw06	y2r0sx8y3tmsqofnf	XTS	1	false	0.2200	771	807 Crooks Glen	Findlay	46460	62.2121	86.629	395
7	6376	718	6tcvui5	0y8apd7us4na7c0s6	Durango	Cruze	false	2.2300	718	441 Jenkins Falls	Wichita Falls	46842	13.9931	43.3459	312
8	6574	758	qz4l34i	iq1k4i1ezk4rsifjx	A4	Volt	false	0.8400	758	3840 White Manor	Draper	38651	-96.7106	66.4625	110
9	8221	62	zndx40x	rpsmzjby8j7wq4u7u	Volt	Colorado	false	0.8300	62	76400 Renee Forge	Cerritos	29430	-150.9537	-70.8034	-124
10	9368	10	w8znml	w6nj4m0mgaw0pzu42	1	F-150	• true	0.4700	10	0674 Veum Rue	Everett	79720	44.2062	32.7517	191

vybere všechny záznamy z tabulky “car” a porovnává každý záznam s odpovídajícími záznamy z tabulky “location” na základě sloupce location_id. Do sady výsledků budou zahrnuty pouze záznamy s odpovídajícími hodnotami location_id v obou tabulkách.

podmínku na data:

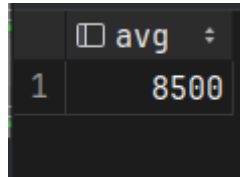
```
SELECT *
FROM adminn WHERE work_experience > 3;
```

	account_id	work_experience
1	4	5
2	6	10

vybere všechny záznamy z tabulky “adminn”, kde hodnota ve sloupci work_experience je větší než 3.

agregaci a podmínku na hodnotu agregační funkce:

```
SELECT AVG(conduct_summary)
FROM Userr
WHERE Userr.account_id <> (SELECT account_id FROM account WHERE name =
'Sergey');
```



	avg
1	8500

vypíše průměrnou hodnotu chování všech uživatelů, kteří nejsou identifikováni jako 'Sergey'.

řazení a stránkování:

```
SELECT *
FROM Ride
ORDER BY time DESC
LIMIT 2 OFFSET 2;
```

	ride_id	car_id	time	date	length_sec	destination	start	driver	price
1		4	5 2024-04-21 18:11:30.000000	2024-04-21		1800 Na Lysine 774	Olympijska 1902	Andrew Smith	360.0000
2		2	6 2024-03-21 18:11:30.000000	2024-03-21		5580 Na Lysine 774	Olympijska 1902	Andrew Smith	20757.6000

vybere všechny záznamy z tabulky “ride”, seřadí je podle sloupce time v sestupném pořadí a načte pouze 2 záznamů počínaje 3. záznamem (offset 2).

množinové operace:

```
SELECT *
FROM drivinglicense
WHERE country_of_issuance = 'Rusko'
UNION
SELECT *
FROM drivinglicense
WHERE country_of_issuance = 'USA';
```

	drivinglicense_id	user_id	number	country_of_issuance
1	1	1	RL128756	Rusko
2	2	3	DL123456	USA
3	3	2	DL111456	Rusko

vybere všechny záznamy z tabulky “drivinglicense”, kde je country_of_issuance "Rusko" nebo "USA", pomocí operátora UNION, který kombinuje výsledky dvou příkazů SELECT do jedné sady výsledků a odstraní všechny duplicitní řádky.

vnořený SELECT:

```
SELECT *
```

```
FROM userr
```

```
WHERE userr.account_id IN (SELECT user_id FROM bans)
```

	account_id	car_id	card_number	conduct_summary
1	2	2	2211222233334444	7000
2	3	1	2211222233330009	10000

vybere všechny záznamy z tabulky “userr”, kde account_id odpovídá libovolným hodnotám account_id vráceným vnořeným příkazem SELECT, který načte všechny hodnoty user_id z tabulky “bans”.