

Database design project

Author: Filip Texl

Associate: Lukáš Král

Brno 2021

Content

1.	Introduction.....	3
2.	Use case for database.....	3
3.	Database design- EER schema.....	4
4.	Tables description and 3NF	4
5.	DDL scripts for PosgreSQL	5
	Scripts for creating tables	5
	Scripts for values	9
6.	Screenshots from PostgreSQL	15
7.	Scripts for MySQL	20
8.	Attachments	25

1. Introduction

The task for this project was to design and then to create the database in PostgreSQL with pgAdmin and in other optional relation database. To achieve that, Data Definition Language was used.

The task for this documentation is to introduce this database and also its use case. Whole project and documentation is on GitHub.

2. Use case for database

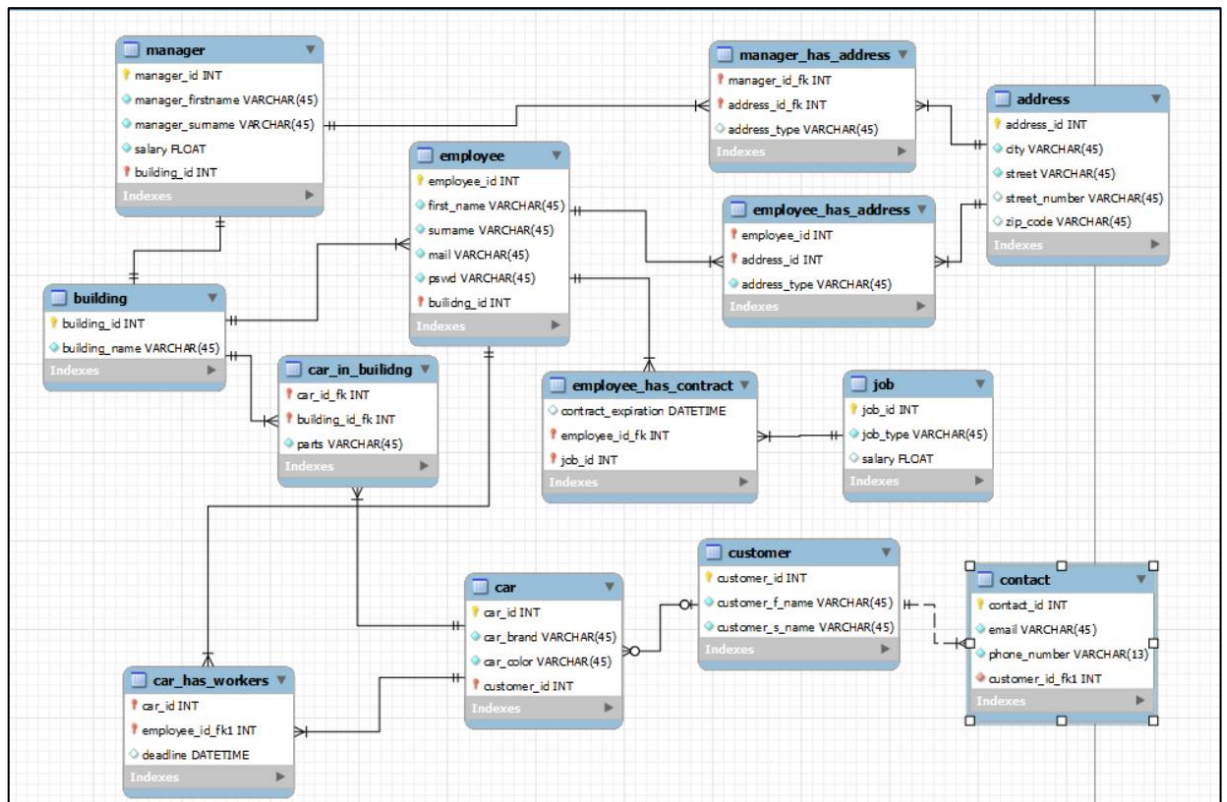
The use case for this database is car factory in automotive. The idea behind our design is to catch basic concepts and needs in car factory where the data are needed to be stored. There are the most common object that you can find in a car factory- employees, buildings, managers and of course cars.

We also need to store some information about employees and managers, so another data, like addresses and what are they doing – jobs, are of course also included.

It is also good to mention that our database is not meant to be general pattern for every car factory. When we thought about relations and concrete concepts in tables, we adapted these things to our fictive car factory. So of course, what is applied here does not have to be well applicable to other factories. The best example is table manager. His relationship with building is 1:1 because for every building we have one manager. He is also separate entity from other employers because of this.

More information is provided in chapter 4 – description of tables.

3. Database design- EER schema



Picture 1 – EER schema in MySQL Workbench

The EER schema and design was created in MySQL Workbench.

4. Tables description and 3NF

Third normal form (3NF) is a database schema design approach for relational databases which uses normalizing principles to reduce the duplication of data, avoid data anomalies, ensure referential integrity, and simplify data management.

The rules for 3NF:

- A relational schema R is in first normal form if the domains of all attributes of R are atomic
- Every non-key attribute of relation R is fully dependent on every candidate key of relation R
- All transitive dependencies must be removed; a non-key attribute may not be functionally dependent on another non-key attribute

We tried to fulfill these requests and discussed them together, so the database is maximally effective and the tables contain only necessary columns. However, we are only beginners in database design so some mistakes in 3NF are possible.

There are 13 tables in our database. The main table is employee- person that is the most important one in factory and without employees no cars are made. We can see 4 relationships with another tables. **Employee** has unique id and also foreign key – building_id, so we know where employee works. Next table is **building**. Buildings are places where the work is done. So, there is relationship between **employee** and **building**. These two are in M: N relationship, because employee can work in multiple buildings and multiple employees can work in one building. The next tables that are in relationship with employee are **address**, **job** and **car**. The relationships are also M: N for similar reasons like in **employee_in_building** case. The next table is manager. **Manager** has M: N relationship with address and 1:1 relationship with **building**, because each building has one and only manager.

Employee_has_contract is table where we can see which employee has certain jobs and when his contract expires. **Car_in_building** table shows us what parts of car are in process of making in buildings.

The table **car** describes car that is made in our factory and then sold to **customers**. Car has M: N relationship with employee (again, multiple employees can work on multiple cars) and 1:N relationship with customer, because customer can buy multiple cars, but one car can be sold only to one customer. Also each customer has one **contact**, so the relationship with this table is 1:1.

5. DDL scripts for PostgreSQL

For DDL scripts, we used forward engineering in MySQL Workbench, which created scripts for MySQL database, so we had to change a little bit those scripts to be suitable for PostgreSQL.

Scripts for creating tables

```
-- -----
-- Table `mydb`.`building`
-- -----
CREATE TABLE IF NOT EXISTS "building" (
    "building_id" BIGSERIAL NOT NULL,
    "building_name" VARCHAR(45) NOT NULL,
    PRIMARY KEY ("building_id"))
;

-- -----
-- Table "mydb"."employee"
-- -----
CREATE TABLE IF NOT EXISTS "employee" (
    "employee_id" BIGSERIAL NOT NULL,
    "first_name" VARCHAR(45) NOT NULL,
```

```

"surname" VARCHAR(45) NOT NULL,
"mail" VARCHAR(45) NOT NULL,
"pswd" VARCHAR(45) NOT NULL,
"building_id" BIGSERIAL NOT NULL,
PRIMARY KEY ("employee_id", "building_id"),
    UNIQUE ("employee_id"),
CONSTRAINT "building_id"
    FOREIGN KEY ("building_id")
    REFERENCES "building" ("building_id")
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

```

```

-- -----
-- Table "mydb"."address"
-- -----
CREATE TABLE IF NOT EXISTS "address" (
    "address_id" BIGSERIAL NOT NULL,
    PRIMARY KEY ("address_id"),
    "city" VARCHAR(45) NOT NULL,
    "street" VARCHAR(45) NOT NULL,
    "street_number" VARCHAR(45),
    "zip_code" VARCHAR(45)
);

```

```

-- -----
-- Table "mydb"."job"
-- -----
CREATE TABLE IF NOT EXISTS "job" (
    "job_id" BIGSERIAL NOT NULL,
    "job_type" VARCHAR(45) NOT NULL,
    "salary" FLOAT NULL,
    PRIMARY KEY ("job_id"));

```

```

-- -----
-- Table "mydb"."employee_has_address"
-- -----
CREATE TABLE IF NOT EXISTS "employee_has_address" (
    "employee_id" BIGSERIAL NOT NULL,
    "address_id" INT NOT NULL,
    "address_type" VARCHAR(45) NOT NULL,
    PRIMARY KEY ("employee_id", "address_id"),
    CONSTRAINT "employee_id"
        FOREIGN KEY ("employee_id")
        REFERENCES "employee" ("employee_id")
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT "address_id"
        FOREIGN KEY ("address_id")
        REFERENCES "address" ("address_id")
        ON DELETE NO ACTION
        ON UPDATE NO ACTION);

```

```

-- -----
-- Table "mydb"."employee_has_contract"
-- -----
CREATE TABLE IF NOT EXISTS "employee_has_contract" (
    "contract_expiration" DATE,
    "employee_id" BIGSERIAL NOT NULL,
    "job_id" BIGSERIAL NOT NULL,
    PRIMARY KEY ("employee_id", "job_id"),
    CONSTRAINT "employee_id"
        FOREIGN KEY ("employee_id")
        REFERENCES "employee" ("employee_id")
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT "job_id"
        FOREIGN KEY ("job_id")
        REFERENCES "job" ("job_id")
        ON DELETE NO ACTION
        ON UPDATE NO ACTION);

```

```

-- -----
-- Table "mydb"."customer"
-- -----
CREATE TABLE IF NOT EXISTS "customer" (
    "customer_id" BIGSERIAL NOT NULL,
    "customer_f_name" VARCHAR(45) NOT NULL,
    "customer_s_name" VARCHAR(45) NOT NULL,
    PRIMARY KEY ("customer_id"),
    UNIQUE("customer_id"));

```

```

-- -----
-- Table "mydb"."car"
-- -----
CREATE TABLE IF NOT EXISTS "car" (
    "car_id" BIGSERIAL NOT NULL,
    "car_brand" VARCHAR(45) NOT NULL,
    "car_color" VARCHAR(45) NOT NULL,
    "customer_id" INT,
    PRIMARY KEY ("car_id"),
    UNIQUE("car_id"),
    CONSTRAINT "customer_id"
        FOREIGN KEY ("customer_id")
        REFERENCES "customer" ("customer_id")
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
;

```

```

-- -----
-- Table "mydb"."contact"
-- -----
CREATE TABLE IF NOT EXISTS "contact" (

```

```

"contact_id" BIGSERIAL NOT NULL,
"email" VARCHAR(45) NOT NULL,
"phone_number" VARCHAR(13) NOT NULL,
"customer_id" BIGSERIAL NOT NULL,
PRIMARY KEY ("contact_id"),
    UNIQUE("contact_id"),
CONSTRAINT "customer_id"
    FOREIGN KEY ("customer_id")
    REFERENCES "customer" ("customer_id")
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
;

```

```

-- -----
-- Table "mydb"."manager"
-- -----

```

```

CREATE TABLE IF NOT EXISTS "manager" (
    "manager_id" BIGSERIAL NOT NULL,
    "manager_firstname" VARCHAR(45) NOT NULL,
    "manager_surname" VARCHAR(45) NOT NULL,
    "salary" FLOAT NOT NULL,
    "building_id" BIGSERIAL NOT NULL,
    PRIMARY KEY ("manager_id", "building_id"),
        UNIQUE("manager_id"),
CONSTRAINT "building_id"
    FOREIGN KEY ("building_id")
    REFERENCES "building" ("building_id")
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

```

```

-- -----
-- Table "mydb"."car_has_workers"
-- -----

```

```

CREATE TABLE IF NOT EXISTS "car_has_workers" (
    "car_id" BIGSERIAL NOT NULL,
    "employee_id" BIGSERIAL NOT NULL,
    "deadline" DATE,
    PRIMARY KEY ("car_id", "employee_id"),
    CONSTRAINT "employee_id"
        FOREIGN KEY ("employee_id")
        REFERENCES "employee" ("employee_id")
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
CONSTRAINT "car_id"
    FOREIGN KEY ("car_id")
    REFERENCES "car" ("car_id")
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

```

```

-- -----
-- Table "mydb"."car_in_buillidng"
-- -----

```

```

CREATE TABLE IF NOT EXISTS "car_in_building" (
    "car_id" BIGSERIAL NOT NULL,

```



```

"building_id" BIGSERIAL NOT NULL,
"parts" VARCHAR(45) NOT NULL,
PRIMARY KEY ("car_id", "building_id"),
CONSTRAINT "car_id"
    FOREIGN KEY ("car_id")
    REFERENCES "car" ("car_id")
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT "building_id"
    FOREIGN KEY ("building_id")
    REFERENCES "building" ("building_id")
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);

-- -----
-- Table "mydb"."manager_has_address"
-- -----

CREATE TABLE IF NOT EXISTS "manager_has_address" (
    "manager_id" BIGSERIAL NOT NULL,
    "address_id" INT NOT NULL,
    "address_type" VARCHAR(45) NOT NULL,
    PRIMARY KEY ("manager_id", "address_id"),
    CONSTRAINT "manager_id"
        FOREIGN KEY ("manager_id")
        REFERENCES "manager" ("manager_id")
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT "address_id"
        FOREIGN KEY ("address_id")
        REFERENCES "address" ("address_id")
        ON DELETE NO ACTION
        ON UPDATE NO ACTION);

```

Scripts for values

```

INSERT INTO building (building_name) values ('výroba');
INSERT INTO building (building_name) values ('účetnictví');
INSERT INTO building (building_name) values ('headquarters');
INSERT INTO building (building_name) values ('závodní jídelna');
INSERT INTO building (building_name) values ('odpadová hala');
INSERT INTO building (building_name) values ('strojírenství');

INSERT INTO manager
(manager_firstname,manager_surname,salary,building_id) values
('Miroslav','Hladký',100000.00,1);
INSERT INTO manager
(manager_firstname,manager_surname,salary,building_id) values
('Natálie','Syrová',82000,2);
INSERT INTO manager
(manager_firstname,manager_surname,salary,building_id) values
('Rudolf','Bednařík',135000,3);
INSERT INTO manager
(manager_firstname,manager_surname,salary,building_id) values
('Alžběta','Stupková',35000,4);

```

```

INSERT INTO manager
(manager_firstname,manager_surname,salary,building_id) values
('Marcel','Šimčík',43200,5);

INSERT INTO job (job_type,salary) values ('dělník',22000.00);
INSERT INTO job (job_type,salary) values ('uklízečka',18000.00);
INSERT INTO job (job_type,salary) values ('vedoucí výroby',33200.00);
INSERT INTO job (job_type,salary) values ('IT technik',35000.00);
INSERT INTO job (job_type,salary) values ('účetní',29500.00);
INSERT INTO job (job_type,salary) values ('kuchař',22000.00);
INSERT INTO job (job_type,salary) values ('správce odpadů',27500.00);
INSERT INTO job (job_type,salary) values ('pomocný kuchař',20000.00);
INSERT INTO job (job_type,salary) values ('strojař',51000.00);

INSERT INTO customer(customer_f_name,customer_s_name) values
('Dominik','Strouhal');
INSERT INTO customer(customer_f_name,customer_s_name) values
('Kateřina','Karelová');
INSERT INTO customer(customer_f_name,customer_s_name) values
('Martin','Kořínek');
INSERT INTO customer(customer_f_name,customer_s_name) values
('Natálie','Pavelková');
INSERT INTO customer(customer_f_name,customer_s_name) values
('Michaela','Jahodová');

INSERT INTO car(car_brand,car_color,customer_id) values
('Volkswagen','červená',3);
INSERT INTO car(car_brand,car_color,customer_id) values
('Volkswagen','černá',2);
INSERT INTO car(car_brand,car_color,customer_id) values
('BMW','černá',4);
INSERT INTO car(car_brand,car_color,customer_id) values
('BMW','bílá',1);
INSERT INTO car(car_brand,car_color,customer_id) values
('Škoda','stříbrná',5);

INSERT INTO contact(email,phone_number,customer_id) values
('domca.str@seznam.cz','+420621345876',1);
INSERT INTO contact(email,phone_number,customer_id) values
('katka.karelova@gmail,pswd.com','+420720753682',2);
INSERT INTO contact(email,phone_number,customer_id) values
('koren@centrum.cz','+420608879998',3);
INSERT INTO contact(email,phone_number,customer_id) values
('natalie.pavelkova@gmail,pswd.com','+420555666777',4);
INSERT INTO contact(email,phone_number,customer_id) values
('miluju,auta@seznam.cz','+420725438876',5);

INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Alena','Samková','alena.samkova@automotive.cz','BgZnZ@Vn7D',2);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Petr','Kalivoda','petr.kal@automotive.cz','P@npUd809L',3);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Hana','Jandová','alena.jand@automotive.cz','tqA*1LIZfj',4);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Alois','Smola','alois.smola@automotive.cz','Xy$y^mWXVn',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Luboš','Veverka','lubos.vever@automotive.cz','^#BMoZKb5F',1);

```

```

INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Markéta','Janková','mj.1@automotive.cz','G@VjSFS&P7',5);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Jakub','Ferenc','jakub.ferenc@automotive.cz','53dI$eNipl',5);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Zbyněk','Hrnčíř','zbynek.hrncir@automotive.cz','3IdrL9aJ@x',2);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Vojtěch','Sluka','vojtech.sluk@automotive.cz','=A3ResP4qa',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Štěpán','Bečvár','stepan.becvar@automotive.cz','8*Lxothlz8',4);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Šimon','Motl','simon.motl@automotive.cz','SP$t?ln2T2',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Rostislav','Kropáček','rost.krop@automotive.cz','7rETH*9o#u',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Richard','Maršík','richard.mars@automotive.cz','fuR049o#AH',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Adam','Vybíral','nevybral@automotive.cz','3Ebr$ciM1b',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Bohuslav','Berky','bohus.berky@automotive.cz','tHe$6Wr&va',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Jan','Hromádka','jan.hromadka@automotive.cz','2l*u?OMAdI',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Martin','Provazník','martin.provaz@automotive.cz','dR4swetr1*',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Vítězslav','Černý','vita.cerny@automotive.cz','*r6kOdrLyE',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Miloš','Vrzal','vrzvrz@automotive.cz','nU*plviPra',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Petr','Trčka','petr.trcka@automotive.cz','7iD=woChav',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Dalibor','Válek','dalibor.valek@automotive.cz','$1kIWro8He',5);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Barbora','Ptáčková','bara.ptack@automotive.cz','xl-HuPhA#2',2);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Jiří','Vašák','jiri.vasak@automotive.cz','m2Th&n8XER',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Jitka','Peterková','jitka.peter@automotive.cz','1It=#PrAji',4);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Luděk','Dědek','ludek.dedek@automotive.cz','s#3ke@ogEk',3);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Natálie','Danielová','natalie.dan@automotive.cz','pUD!R77L?o',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Božena','Fuksová','bozena.fuks@automotive.cz','s4A&=tIFId',5);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Michal','Starý','michal.stary@automotive.cz','&a4OcH_tad',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Rostislav','Šebek','rosta.sebek@automotive.cz','gIy0TucoK$',5);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Tomáš','Adam','tomas.adam@automotive.cz','pasT8?lzu+',3);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Jana','Čížková','jana.cizk@automotive.cz','qaHlq$Mi5P',5);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Soňa','Bauerová','sona.bauer@automotive.cz','3@MupR&w3C',3);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Šimon','Sokol','simon.sokol@automotive.cz','sI*1+hiD5o',3);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Helena','Slezáková','helena.slezak@automotive.cz','wlmA9ra3@m',4);

```

```

INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Tomáš','Bartošek','tomas.bartos@automotive.cz','s&aP6!hEml',3);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('František','Benda','fanos.benda@automotive.cz','?osT7trEpa',3);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Emilie','Kubátová','emilie.kubat@automotive.cz','Pro8-TaZ8s',2);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Markéta','Dvorská','maka.dvorska@automotive.cz','7h3JurOf-U',5);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Radim','Doležal','radim.dolez@automotive.cz','w*prE_O0RU',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Vendula','Pechová','vendy.pech@automotive.cz','w?uqadlr4S',3);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Lubomír','Vaníček','lubomir.van@automotive.cz','ph1+R&drOX',6);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Filip','Myška','filip.myska@automotive.cz','flIdRA+ine',6);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Lubomír','Beneš','lubomir.benes@automotive.cz','Y_hu&iR5Me',6);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Ludvík','Daniš','ludva.danis@automotive.cz','xa*I=8stlh',6);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Ján','Kudrna','jan.kudrna@automotive.cz','b+u3elT*jA',6);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Matěj','Dvořáček','matej.dvor@automotive.cz','9$zEYUBOxe',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Tomáš','Hlaváč','tom.hlavac@automotive.cz','3h&viCho@o',1);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Ludvík','Gajdoš','ludva.gajdos@automotive.cz','C@U2ug9pUt',5);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Zbyněk','Trnak','zbynek.trnka@tomotive.cz','4*CoNlSwiT',5);
INSERT INTO employee (first_name,surname,mail,pswd,building_id) values
('Kamil','Koutný','kamil.koutny@automotive.cz','4*CoNlSwiT',5);

```

```

INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2023-01-01',1,5);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2023-01-01',2,5);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2023-01-01',3,4);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2023-01-01',4,1);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2023-01-01',5,1);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2023-01-01',6,7);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2023-01-01',7,1);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2023-01-01',8,5);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2023-01-01',9,1);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2023-01-01',10,6);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values (NULL,11,1);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2025-01-01',12,3);

```

[illegible]

```

INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values (NULL,42,9);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2023-01-01',43,9);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values (NULL,44,9);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values ('2023-01-01',45,9);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values (NULL,46,1);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values (NULL,47,1);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values (NULL,48,1);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values (NULL,49,1);
INSERT INTO employee_has_contract
(contract_expiration,employee_id,job_id) values (NULL,50,7);

```

```

INSERT INTO address (city,street,street_number,zip_code) values
('Nechanice','Na Kopečku', 1544, 50315);
INSERT INTO address (city,street,street_number,zip_code) values
('Louny','Louny',17123,44001);
INSERT INTO address (city,street,street_number,zip_code) values
('Úhretice','Jiráskova',1045,53832);
INSERT INTO address (city,street,street_number,zip_code) values
('Zajecov','K Lukárně',5188,26736);
INSERT INTO address (city,street,street_number,zip_code) values
('Rožmitál pod Tremšínem', 'U medvídků',2155,26242);

```

```

INSERT INTO car_has_workers (car_id,employee_id,deadline)
values(1,12,'2022-10-05');
INSERT INTO car_has_workers (car_id,employee_id,deadline)
values(1,14,'2022-10-7');
INSERT INTO car_has_workers (car_id,employee_id,deadline)
values(1,44,'2022-10-12');
INSERT INTO car_has_workers (car_id,employee_id,deadline)
values(3,1,'2022-4-22');
INSERT INTO car_has_workers (car_id,employee_id,deadline)
values(3,15,'2022-4-27');

```

```

INSERT INTO car_in_building (car_id,building_id,parts)
values(1,1,'Volant');
INSERT INTO car_in_building (car_id,building_id,parts)
values(1,6,'Motor');
INSERT INTO car_in_building (car_id,building_id,parts)
values(2,1,'Zrcátka');
INSERT INTO car_in_building (car_id,building_id,parts)
values(2,6,'Motor');
INSERT INTO car_in_building (car_id,building_id,parts)
values(4,1,'Sedadla');
INSERT INTO car_in_building (car_id,building_id,parts)
values(4,6,'Motor');

```

```

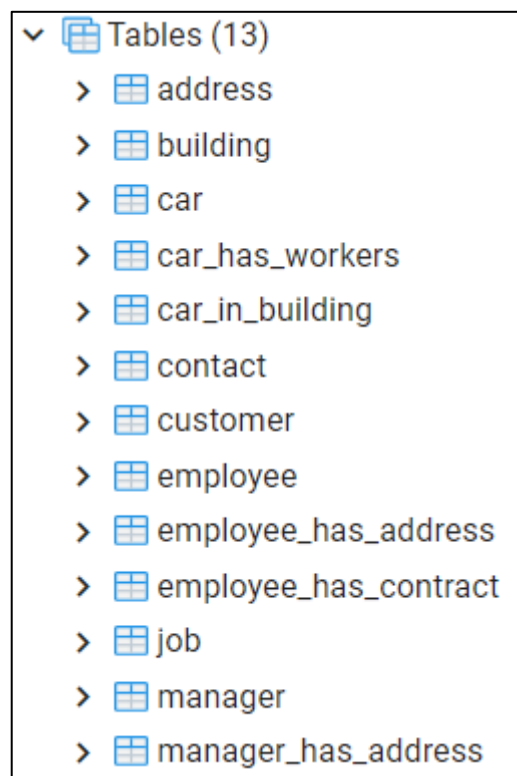
INSERT INTO employee_has_address (address_id, employee_id,
address_type) values (1,1,'Korespondenční adresa');

```

```
INSERT INTO employee_has_address (address_id, employee_id,
address_type) values (3,1,'Adresa trvalého bydliště');
INSERT INTO employee_has_address (address_id, employee_id,
address_type) values (4,2,'Korespondenční adresa');
INSERT INTO employee_has_address (address_id, employee_id,
address_type) values (2,3,'Korespondenční adresa');
INSERT INTO employee_has_address (address_id, employee_id,
address_type) values (5,3,'Adresa trvalého bydliště');
```

```
INSERT INTO manager_has_address (manager_id, address_id, address_type)
values (1,1,'Adresa trvalého bydliště');
INSERT INTO manager_has_address (manager_id, address_id, address_type)
values (5,1,'Adresa trvalého bydliště');
INSERT INTO manager_has_address (manager_id, address_id, address_type)
values (4,4,'Korespondenční adresa');
INSERT INTO manager_has_address (manager_id, address_id, address_type)
values (2,3,'Korespondenční adresa');
INSERT INTO manager_has_address (manager_id, address_id, address_type)
values (3,5,'Adresa trvalého bydliště');
```

6. Screenshots from PostgreSQL



car_factory_5/postgres@bds-db						
Query Editor Query History						
<pre> 1 SELECT * FROM public.employee 2 ORDER BY employee_id ASC, building_id ASC </pre>						
Data Output Explain Messages Notifications						
	employee_id [PK] bigint	first_name character varying (45)	surname character varying (45)	mail character varying (45)	pswd character varying (45)	building_id [PK] bigint
1	1	Alena	Samková	alena.samkova@automotive.cz	BgZNZ@Vn7D	2
2	2	Petr	Kalivoda	petr.kal@automotive.cz	P@npUd809L	3
3	3	Hana	Jandová	alena.jand@automotive.cz	tqA*1LIZfj	4
4	4	Alois	Smola	alois.smola@automotive.cz	Xy\$y*mWXVn	1
5	5	Luboš	Veverka	lubos.vever@automotive.cz	^#BMoZKb5F	1
6	6	Markéta	Janková	mj.1@automotive.cz	G@VjSFS&P7	5
7	7	Jakub	Ferenc	jakub.ferenc@automotive.cz	53dl\$eNipl	5
8	8	Zbyněk	Hrnčář	zbynek.hrncir@automotive.cz	3ldrL9aJ@x	2
9	9	Vojtěch	Sluka	vojtech.sluk@automotive.cz	=A3ResP4qa	1
10	10	Štěpán	Bečvár	stepan.becvar@automotive.cz	8*Lxothlz8	4
11	11	Šimon	Motl	simon.motl@automotive.cz	SP\$t?ln2T2	1
12	12	Rostislav	Kropáček	rost.krop@automotive.cz	7rETh*9o#u	1
13	13	Richard	Maršík	richard.mars@automotive.cz	fuRO49o#AH	1
14	14	Adam	Vybíral	nevybral@automotive.cz	3Ebr\$clM1b	1
15	15	Bohuslav	Berky	bohus.berky@automotive.cz	tHe\$6Wr&va	1
16	16	Jan	Hromádka	jan.hromadka@automotive.cz	2l*u?OMAdI	1
17	17	Martin	Provozník	martin.provaz@automotive.cz	dR4swetrI*	1
18	18	Vítězslav	Černý	vita.cerny@automotive.cz	*r6kOdrLyE	1



car_factory_5/postgres@bds-db

Query Editor

Query History

```
1 SELECT * FROM public.job
2 ORDER BY job_id ASC
```

Data Output

Explain

Messages

Notifications

	job_id [PK] bigint	job_type character varying (45)	salary double precision
1	1	dělník	22000
2	2	uklízečka	18000
3	3	vedoucí výroby	33200
4	4	IT technik	35000
5	5	účetní	29500
6	6	kuchař	22000
7	7	správce odpadů	27500
8	8	pomocný kuchař	20000
9	9	strojař	51000



car_factory_5/postgres@bds-db

Query Editor

Query History

```
1 SELECT * FROM public.employee_has_contract
2 ORDER BY employee_id ASC, job_id ASC
```

Data Output

Explain

Messages

Notifications

	contract_expiration date	employee_id [PK] bigint	job_id [PK] bigint
1	2023-01-01	1	5
2	2023-01-01	2	5
3	2023-01-01	3	4
4	2023-01-01	4	1
5	2023-01-01	5	1
6	2023-01-01	6	7
7	2023-01-01	7	1
8	2023-01-01	8	5
9	2023-01-01	9	1
10	2023-01-01	10	6
11	[null]	11	1
12	2025-01-01	12	3
13	[null]	13	1
14	[null]	14	1
15	[null]	15	1
16	[null]	16	1
17	[null]	17	1
18	[null]	18	1



automotive/postgres@bds-db

Query Editor

Query History

```
1 SELECT * FROM public.employee_has_address
2 ORDER BY employee_id ASC, address_id ASC
```

Data Output

Explain

Messages

Notifications

	employee_id [PK] bigint	address_id [PK] integer	address_type character varying (45)	
1	1	1	Korespondenční adresa	
2	1	3	Adresa trvalého bydliště	
3	2	4	Korespondenční adresa	
4	3	2	Korespondenční adresa	
5	3	5	Adresa trvalého bydliště	

Date	Rows Affected	Duration
24. 10. 2021 15:44:18	1	102 msec

Copy

```

INSERT INTO building (building_name) values ('výroba');
INSERT INTO building (building_name) values ('účetnictví');
INSERT INTO building (building_name) values ('headquarte');
INSERT INTO building (building_name) values ('závodní jí');
INSERT INTO building (building_name) values ('odpadová h');
INSERT INTO building (building_name) values ('strojírens');

INSERT INTO manager (manager_firstname,manager_surname,s);
INSERT INTO manager (manager_firstname,manager_surname,s);
INSERT INTO manager (manager_firstname,manager_surname,s);
INSERT INTO manager (manager_firstname,manager_surname,s);
INSERT INTO manager (manager_firstname,manager_surname,s);

INSERT INTO job (job_type,salary) values ('dělník',220);

```

7. Scripts for MySQL

For MySQL scripts, MySQL Workbench was very helpful. With the use of forward engineering, DDL scripts were created. However, Workbench doesn't like duplicate names even in foreign keys so we had to rename some foreign keys.

```

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----
-- Schema mydb
-- -----

-- -----
-- Schema mydb
-- -----

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;

-- -----

```

```
-- Table `mydb`.`building`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`building` (
  `building_id` INT NOT NULL,
  `building_name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`building_id`),
  UNIQUE INDEX `building_id_UNIQUE` (`building_id` ASC) VISIBLE)
ENGINE = InnoDB;
```

```
-- -----
-- Table `mydb`.`employee`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`employee` (
  `employee_id` INT NOT NULL,
  `first_name` VARCHAR(45) NOT NULL,
  `surname` VARCHAR(45) NOT NULL,
  `mail` VARCHAR(45) NOT NULL,
  `pswd` VARCHAR(45) NOT NULL,
  `builidng_id` INT NOT NULL,
  PRIMARY KEY (`employee_id`, `builidng_id`),
  UNIQUE INDEX `employee_id_UNIQUE` (`employee_id` ASC) VISIBLE,
  INDEX `building_id_idx` (`builidng_id` ASC) VISIBLE,
  CONSTRAINT `building_id`
    FOREIGN KEY (`builidng_id`)
    REFERENCES `mydb`.`building` (`building_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Table `mydb`.`address`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`address` (
  `address_id` INT NOT NULL,
  `city` VARCHAR(45) NOT NULL,
  `street` VARCHAR(45) NOT NULL,
  `street_number` VARCHAR(45) NULL,
  `zip_code` VARCHAR(45) NULL,
  UNIQUE INDEX `address_id_UNIQUE` (`address_id` ASC) VISIBLE,
  PRIMARY KEY (`address_id`))
ENGINE = InnoDB;
```

```
-- -----
-- Table `mydb`.`job`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`job` (
  `job_id` INT NOT NULL,
  `job_type` VARCHAR(45) NOT NULL,
  `salary` FLOAT NULL,
  PRIMARY KEY (`job_id`),
  UNIQUE INDEX `job_id_UNIQUE` (`job_id` ASC) VISIBLE)
ENGINE = InnoDB;
```

```
-- -----
```

```

-- Table `mydb`.`employee_has_address`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`employee_has_address` (
  `employee_id` INT NOT NULL,
  `address_id` INT NOT NULL,
  `address_type` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`employee_id`, `address_id`),
  INDEX `address_id_idx` (`address_id` ASC) VISIBLE,
  CONSTRAINT `employee_id`
    FOREIGN KEY (`employee_id`)
      REFERENCES `mydb`.`employee` (`employee_id`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `address_id`
    FOREIGN KEY (`address_id`)
      REFERENCES `mydb`.`address` (`address_id`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`employee_has_contract`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`employee_has_contract` (
  `contract_expiration` DATETIME NULL,
  `employee_id_fk` INT NOT NULL,
  `job_id` INT NOT NULL,
  PRIMARY KEY (`employee_id_fk`, `job_id`),
  INDEX `job_id_idx` (`job_id` ASC) VISIBLE,
  CONSTRAINT `employee_id_fk`
    FOREIGN KEY (`employee_id_fk`)
      REFERENCES `mydb`.`employee` (`employee_id`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `job_id`
    FOREIGN KEY (`job_id`)
      REFERENCES `mydb`.`job` (`job_id`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`customer`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`customer` (
  `customer_id` INT NOT NULL,
  `customer_f_name` VARCHAR(45) NOT NULL,
  `customer_s_name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`customer_id`),
  UNIQUE INDEX `customer_id_UNIQUE` (`customer_id` ASC) VISIBLE)
ENGINE = InnoDB;

-- -----
-- Table `mydb`.`car`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`car` (
  `car_id` INT NOT NULL,
  `car_brand` VARCHAR(45) NOT NULL,
  `car_color` VARCHAR(45) NOT NULL,
  `customer_id` INT NULL,
  UNIQUE INDEX `car_id_UNIQUE` (`car_id` ASC) VISIBLE,
  PRIMARY KEY (`car_id`, `customer_id`),
  INDEX `customer_id_idx` (`customer_id` ASC) VISIBLE,
  UNIQUE INDEX `customer_id_UNIQUE` (`customer_id` ASC) VISIBLE,
  CONSTRAINT `customer_id`
    FOREIGN KEY (`customer_id`)
    REFERENCES `mydb`.`customer` (`customer_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`car_has_workers`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`car_has_workers` (
  `car_id` INT NOT NULL,
  `employee_id_fk1` INT NOT NULL,
  `deadline` DATETIME NULL,
  PRIMARY KEY (`car_id`, `employee_id_fk1`),
  INDEX `employee_id_idx` (`employee_id_fk1` ASC) VISIBLE,
  CONSTRAINT `employee_id_fk1`
    FOREIGN KEY (`employee_id_fk1`)
    REFERENCES `mydb`.`employee` (`employee_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `car_id`
    FOREIGN KEY (`car_id`)
    REFERENCES `mydb`.`car` (`car_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`manager`
-----
CREATE TABLE IF NOT EXISTS `mydb`.`manager` (
  `manager_id` INT NOT NULL,
  `manager_firstname` VARCHAR(45) NOT NULL,
  `manager_surname` VARCHAR(45) NOT NULL,
  `salary` FLOAT NOT NULL,
  `building_id` INT NOT NULL,
  PRIMARY KEY (`manager_id`, `building_id`),
  UNIQUE INDEX `manager_id_UNIQUE` (`manager_id` ASC) VISIBLE,
  INDEX `buliding_id_idx` (`building_id` ASC) VISIBLE,
  CONSTRAINT `buliding_id`
    FOREIGN KEY (`building_id`)
    REFERENCES `mydb`.`building` (`building_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- Table `mydb`.`car_in_buildng`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`car_in_buildng` (
  `car_id_fk` INT NOT NULL,
  `building_id_fk` INT NOT NULL,
  `parts` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`car_id_fk`, `building_id_fk`),
  INDEX `building_id_idx` (`building_id_fk` ASC) VISIBLE,
  CONSTRAINT `car_id_fk`
    FOREIGN KEY (`car_id_fk`)
      REFERENCES `mydb`.`car` (`car_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `building_id_fk`
    FOREIGN KEY (`building_id_fk`)
      REFERENCES `mydb`.`building` (`building_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- Table `mydb`.`manager_has_address`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`manager_has_address` (
  `manager_id_fk` INT NOT NULL,
  `address_id_fk` INT NOT NULL,
  `address_type` VARCHAR(45) NULL,
  PRIMARY KEY (`manager_id_fk`, `address_id_fk`),
  INDEX `address_id_idx` (`address_id_fk` ASC) VISIBLE,
  CONSTRAINT `manager_id_fk`
    FOREIGN KEY (`manager_id_fk`)
      REFERENCES `mydb`.`manager` (`manager_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `address_id_fk`
    FOREIGN KEY (`address_id_fk`)
      REFERENCES `mydb`.`address` (`address_id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- -----
-- Table `mydb`.`contact`
-- -----
CREATE TABLE IF NOT EXISTS `mydb`.`contact` (
  `contact_id` INT NOT NULL,
  `email` VARCHAR(45) NOT NULL,
  `phone_number` VARCHAR(13) NOT NULL,
  `customer_id_fk1` INT NOT NULL,
  PRIMARY KEY (`contact_id`),
  INDEX `customer_id_idx` (`customer_id_fk1` ASC) VISIBLE,
  CONSTRAINT `customer_id_fk1`
    FOREIGN KEY (`customer_id_fk1`)
      REFERENCES `mydb`.`customer` (`customer_id`)

```



```
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

8. Attachments

- DDL scripts in .sql files
- Documentation on GitHub, link: <https://github.com/Philippeer/bds-db-design>
- .zip file