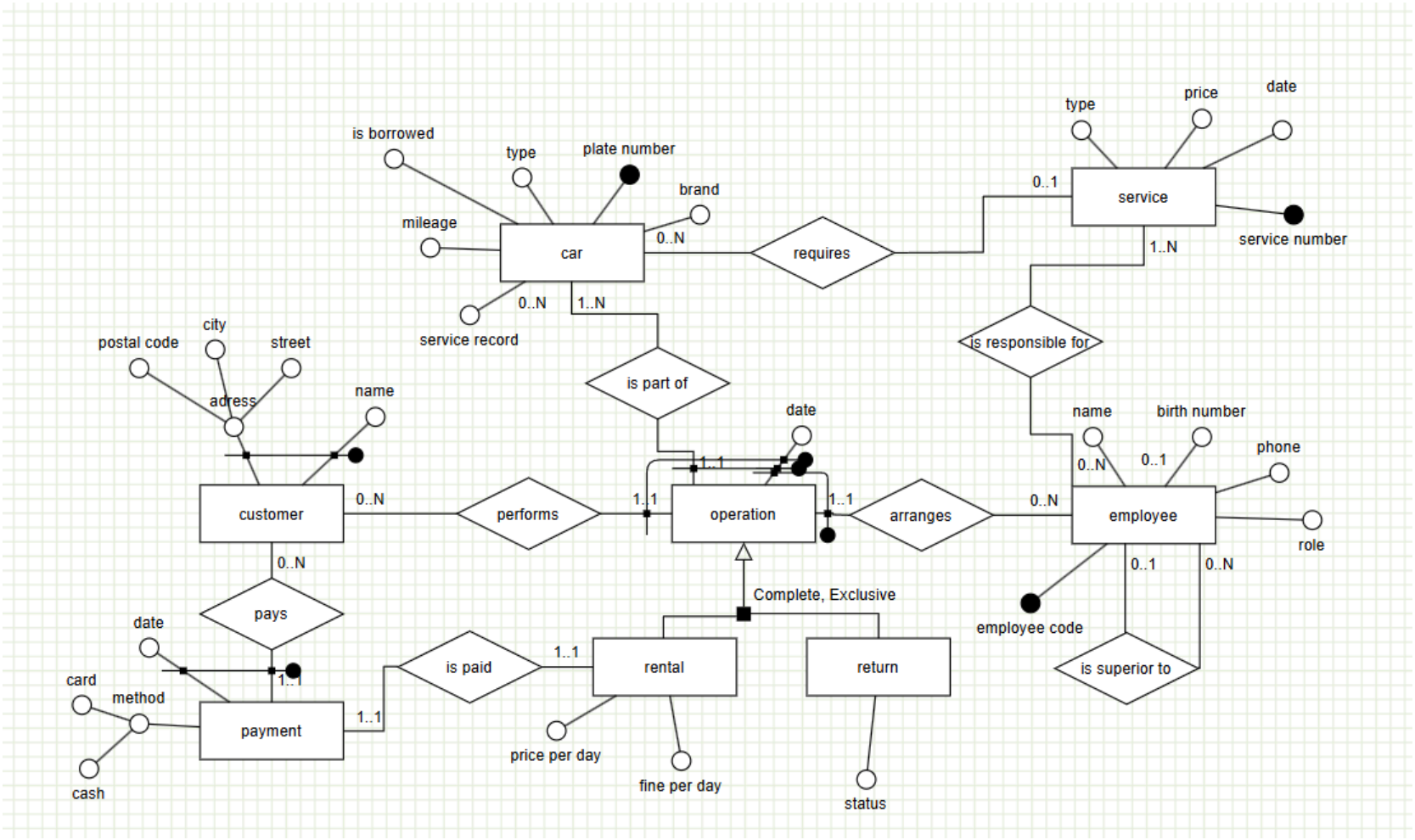
DBS_CP3

Last edited by **Filip Kopecký** just now

ER:



Relační schema:

- Car(plate_number, type, mileage,is_borrowed,brand)
- Service_record(service_record, plate_number)
  - FK: plate_number ⊆ Car(plate_number)
- Requires(service_number, plate_number)
- Service(service_number, price, date, type)
  - FK: plate_number ⊆ Car(plate_number)
  - FK: service_number ⊆ Service(service_number)
- Employee(employee_code,name, phone, role)
- Birth_number(employee, birth_number)
- Is_responsible_for(employee, service_number)
  - FK: employee ⊆ Employee(employee_code)
  - FK: service_number ⊆ Service(service_number)
- Is_superior_to(employee, supervisor)
  - FK: employee ⊆ Employee(employee_code)
  - FK: supervisor ⊆ Employee(employee_code)
- Operation(date, employee, date, customer,date,car)
  - FK: employee ⊆ Employee(employee_code)
  - FK: customer ⊆ Customer(name, street, city, postal_code)
  - FK: car ⊆ Car(plate_number)
- Rental(date, customer, fine_per_day, price_per_day)
  - FK: date, customer ⊆ Operation(date, customer)
- Return(date, customer, status)
  - FK: date, customer ⊆ Operation(date, customer)
- Customer(name, street, city, postal_code)
- Payment(customer,date, card, cash, rental)
  - FK: rental ⊆ Rental(date, customer)
  - FK: customer ⊆ Customer(street, city, postal_code,name)

```
-- UNION This query retrieves the license plates of all cars that have either
-- extremely low mileage (less than 1) or high mileage (greater than 20,000)
-- without duplicates - množinové operace
```

```
SELECT plate_number
FROM car
WHERE mileage < 1
UNION
SELECT plate_number
FROM car
WHERE mileage > 20000;
```

	plate_number
1	PQR2345
2	JKL3456

```
-- Find customers who rented a car more than zero times, sorted by the number of rentals desc - řazení

SELECT customer.name, COUNT(*) AS rentals FROM customer
JOIN operation ON (customer.id = operation.customer_id)
JOIN rental ON (operation.id = rental.operation_id)
GROUP BY customer.name
HAVING COUNT(*) > 0
ORDER BY COUNT(*) DESC;
```

	name	rentals
1	Petr Svoboda	2
2	Tomáš Černý	1
3	Filip Kopecký	1
4	Anna Srdíčková	1
5	Matěj Nový	1
6	Lucie Krásná	1

```
-- Find the customer with the most rentals - vnořený SELECT, podmínka na data, vnitřní spojení
SELECT name,rentals FROM (
SELECT name, COUNT(*) as rentals FROM customer JOIN operation ON (customer.id = operation.customer_id)
JOIN rental ON (operation.id = rental.operation_id)
GROUP BY customer.name
 ) AS customer_stats
WHERE rentals = (
SELECT MAX(rentals)
FROM (
SELECT COUNT(*) as rentals
FROM customer
JOIN operation ON customer.id = operation.customer_id
 JOIN rental ON operation.id = rental.operation_id
 GROUP BY customer.name
 ) AS subquery
);
```

	name	rentals
1	Petr Svoboda	2

```
-- most expensive car to rent - podmínku na hodnotu agregační funkce
SELECT plate_number, brand,price_per_day AS price_per_day
FROM (SELECT car.plate_number, car.brand, rental.price_per_day
 FROM car
JOIN operation o ON car.id = o.car_id
 JOIN rental ON o.id = rental.operation_id
 GROUP BY plate_number, price_per_day,car.brand
HAVING price_per_day = (SELECT MAX(price_per_day)
FROM (
car JOIN operation o ON car.id = o.car_id
 JOIN rental ON o.id = rental.operation_id))) as pnbppd;
```

	plate_number	brand	price_per_day
1	PQR2345	Škoda	2200

```
-- list the brands of cars that individual customers have rented - stránkování
SELECT brand,name FROM
car JOIN operation o on car.id = o.car_id
JOIN customer c on o.customer_id = c.id
JOIN rental r on o.id = r.operation_id
GROUP BY brand,name
ORDER BY brand ASC
LIMIT 4 OFFSET 2;
```

	brand	name
1	Škoda	Anna Srdíčková
2	Škoda	Petr Svoboda
3	Tesla	Lucie Krásná
4	Toyota	Tomáš Černý

```
-- selects the names of employees who are not assigned to any service,limited by 15 - vnější spojení tabulek
SELECT employee.name
FROM employee
LEFT JOIN is_responsible_for ON employee.id = is_responsible_for.employee_id
WHERE is_responsible_for.service_id IS NULL
LIMIT 15 OFFSET 0;
```

	name
1	Jáchym Hoďhodostroje
2	Employee 32
3	Employee 33
4	Employee 34
5	Employee 35
6	Employee 36
7	Employee 37
8	Employee 38
9	Employee 39
10	Employee 40
11	Employee 41
12	Employee 42
13	Employee 43
14	Employee 44
15	Employee 45

code:

```
-- ON UPDATE/DELETE oduvodneni --Zachování datové integrity --Neumožňuje smazat záznam v rodičovské tabulce, pokud na něj odkazují záznamy v jiných tabulkách. Tím se zabrání vzniku "sirotků" (např. objednávka bez zákazníka).

--Ochrana důležitých vazeb --aby data zůstala konzistentní a kompletní

-- CAR CREATE TABLE car ( id SERIAL PRIMARY KEY, plate_number CHAR(7) UNIQUE NOT NULL, type TEXT NOT NULL, mileage INT CHECK (mileage >= 0), is_borrowed BOOLEAN DEFAULT FALSE, brand TEXT NOT NULL );

-- SERVICE

CREATE TABLE service ( id SERIAL PRIMARY KEY, price INT CHECK (price > 0) NOT NULL, service_date DATE CHECK (service_date > CURRENT_DATE) NOT NULL, type TEXT NOT NULL );

-- REQUIRES

CREATE TABLE requires ( id SERIAL PRIMARY KEY, service_id INT REFERENCES service(id) ON UPDATE CASCADE ON DELETE RESTRICT, car_id INT REFERENCES car(id) ON UPDATE CASCADE ON DELETE RESTRICT );

-- SERVICE RECORD

CREATE TABLE service_record ( id SERIAL PRIMARY KEY, description TEXT, car_id INT REFERENCES car(id) ON UPDATE CASCADE ON DELETE RESTRICT );
```

```
-- EMPLOYEE
```

```
CREATE TABLE employee ( id SERIAL PRIMARY KEY, name TEXT NOT NULL, phone VARCHAR(13) NOT NULL CHECK (phone ~ '^+420[0-9]{9}$'), role TEXT NOT NULL );
```

```
-- BIRTH NUMBER
```

```
CREATE TABLE birth_number ( id SERIAL PRIMARY KEY, employee_id INT REFERENCES employee(id) ON UPDATE CASCADE ON DELETE RESTRICT, birth_number VARCHAR(11) NOT NULL CHECK (birth_number ~ '^[0-9]{6}/[0-9]{4}$') );
```

```
-- IS RESPONSIBLE FOR
```

```
CREATE TABLE is_responsible_for ( id SERIAL PRIMARY KEY, employee_id INT REFERENCES employee(id) ON UPDATE CASCADE ON DELETE RESTRICT, service_id INT REFERENCES service(id) ON UPDATE CASCADE ON DELETE RESTRICT );
```

```
-- IS SUPERIOR TO
```

```
CREATE TABLE is_superior_to ( id SERIAL PRIMARY KEY, employee_id INT REFERENCES employee(id) ON UPDATE CASCADE ON DELETE RESTRICT, supervisor_id INT REFERENCES employee(id) ON UPDATE CASCADE ON DELETE RESTRICT );
```

```
-- CUSTOMER
```

```
CREATE TABLE customer ( id SERIAL PRIMARY KEY, name TEXT NOT NULL, street TEXT NOT NULL, city TEXT NOT NULL, postal_code VARCHAR(5) CHECK (postal_code ~ '^[0-9]{5}$') );
```

```
-- OPERATION
```

```
CREATE TABLE operation ( id SERIAL PRIMARY KEY, operation_date DATE NOT NULL, customer_id INT REFERENCES customer(id) ON UPDATE CASCADE ON DELETE RESTRICT, car_id INT REFERENCES car(id) ON UPDATE CASCADE ON DELETE RESTRICT, employee_id INT REFERENCES employee(id) ON UPDATE CASCADE ON DELETE RESTRICT );
```

```
-- RENTAL
```

```
CREATE TABLE rental ( operation_id INT PRIMARY KEY REFERENCES operation(id) ON UPDATE CASCADE ON DELETE RESTRICT, fine_per_day INT NOT NULL, price_per_day INT NOT NULL );
```

```
-- RETURN
```

```
CREATE TABLE return ( operation_id INT PRIMARY KEY REFERENCES operation(id) ON UPDATE CASCADE ON DELETE RESTRICT, status TEXT NOT NULL );
```

```
-- PAYMENT
```

```
CREATE TABLE payment ( id SERIAL PRIMARY KEY, payment_date DATE NOT NULL, card DECIMAL(10, 2) NOT NULL, cash DECIMAL(10, 2) NOT NULL, customer_id INT REFERENCES customer(id) ON UPDATE CASCADE ON DELETE NO ACTION, rental_id INT REFERENCES rental(operation_id) ON UPDATE CASCADE ON DELETE NO ACTION );
```

```
INSERT INTO car (plate_number, type, mileage, is_borrowed, brand) VALUES ('ABC1234', 'diesel', 12000, FALSE, 'Volkswagen'), ('DEF5678', 'electric', 3000, TRUE, 'Tesla'), ('GHI9012', 'hybrid', 20000, FALSE, 'Toyota'), ('JKL3456', 'petrol', 50000, FALSE, 'Ford'), ('MNO7890', 'electric', 1000, TRUE, 'BMW'), ('PQR2345', 'diesel', 250000, FALSE, 'Škoda');
```

```
INSERT INTO service (price, service_date, type) VALUES (2000, CURRENT_DATE + INTERVAL '5 days', 'oil change'), (3500, CURRENT_DATE + INTERVAL '10 days', 'tire replacement'), (5000, CURRENT_DATE + INTERVAL '15 days', 'brake check'), (7500, CURRENT_DATE + INTERVAL '20 days', 'engine diagnostics'), (10000, CURRENT_DATE + INTERVAL '25 days', 'clutch replacement'), (3000, CURRENT_DATE + INTERVAL '30 days', 'battery check');
```

```
INSERT INTO requires (service_id, car_id) VALUES (1,1), (2,2), (3,3), (4,4), (5,5), (6,6);
```

```
INSERT INTO service_record (description, car_id) VALUES
```

```
('Regular check-up', 1),
('Tire analysis', 2),
('Brake fluid test', 3),
('Battery replacement', 4),
('Exhaust diagnostics', 5),
('Oil filter replacement', 6);
```

```
INSERT INTO employee (name, phone, role) VALUES ('Jan Novák', '+420777123456', 'mechanic'), ('Eva Malá', '+420777654321', 'technician'), ('Pavel Velký', '+420777999888', 'manager'), ('Petr Malý', '+420333222999', 'mechanic'), ('Lucie Velká', '+420111345765', 'technician'), ('Ondřej Moudrý', '+420098765432', 'inspector'), ('Jáchym Hoďhodostroje', '+420888999000', 'magician');
```

```
INSERT INTO birth_number (employee_id, birth_number) VALUES (1, '850101/1234'), (2, '900202/2345'), (3, '791231/3456'), (4, '850505/4567'), (5, '920313/5678'), (6, '890120/6789');
```

```
INSERT INTO is_responsible_for (employee_id, service_id) VALUES (1,1), (2,2), (3,3), (4,4), (5,5),(6,6);

INSERT INTO is_superior_to (employee_id, supervisor_id) VALUES (1,3), (2,3), (4,3), (5,3), (6,3), (3,3);

INSERT INTO customer (name, street, city, postal_code) VALUES ('Petr Svoboda', 'Hlavní 123', 'Praha', '11000'), ('Lucie Krásná', 'Dlouhá 45', 'Brno', '60200'), ('Tomáš Černý', 'Krátká 7', 'Ostrava', '70030'), ('Filip Kopecký', 'Jedlová 123', 'Plzeň', '30100'), ('Matěj Nový', 'Javorová 12', 'Liberec', '46001'), ('Anna Srdíčková', 'Na Příkopě 10', 'Zlín', '76001'), ('Jáchym Hoďhodostroje', 'U Emeta', 'Legoland', '00000');

INSERT INTO operation (operation_date, customer_id, car_id, employee_id) VALUES (CURRENT_DATE, 1, 1, 1), (CURRENT_DATE, 2, 2, 2), (CURRENT_DATE, 3, 3, 3), (CURRENT_DATE, 4, 4, 4), (CURRENT_DATE, 5, 5, 5), (CURRENT_DATE, 6, 6, 6), (CURRENT_DATE, 1, 6, 6);

INSERT INTO rental (operation_id, fine_per_day, price_per_day) VALUES (1, 500, 1000), (2, 300, 800), (3, 450, 900), (4, 600, 1100), (5, 200, 950), (6, 0, 1200), (7, 0, 2200);

INSERT INTO return (operation_id, status) VALUES (1, 'returned'), (2, 'late'), (3, 'returned'), (4, 'damaged'), (5, 'returned'), (6, 'late'), (7, 'late');

INSERT INTO payment (payment_date, card, cash, customer_id, rental_id) VALUES (CURRENT_DATE, 500.00, 500.00, 1, 1), (CURRENT_DATE, 800.00, 0.00, 2, 2), (CURRENT_DATE, 300.00, 600.00, 3, 3), (CURRENT_DATE, 600.00, 500.00, 4, 4), (CURRENT_DATE, 950.00, 0.00, 5, 5), (CURRENT_DATE, 600.00, 600.00, 6, 6); SELECT * FROM payment; -- DROP TABLE IF EXISTS -- payment, return, rental, operation, -- is_superior_to, is_responsible_for, birth_number, employee, -- service_record, requires, service, -- customer, car -- CASCADE;

-- UNION This query retrieves the license plates of all cars that have either -- extremely low mileage (less than 1) or high mileage (greater than 20,000) -- without duplicates - množinové operace

SELECT plate_number FROM car WHERE mileage < 1 UNION SELECT plate_number FROM car WHERE mileage > 20000;

-- Find customers who rented a car more than zero times, sorted by the number of rentals desc - řazení

SELECT customer.name, COUNT( ) AS rentals FROM customer JOIN operation ON (customer.id = operation.customer_id) JOIN rental ON (operation.id = rental.operation_id) GROUP BY customer.name HAVING COUNT( ) > 0 ORDER BY COUNT(*) DESC;

-- Find the customer with the most rentals - vnořený SELECT, podmínka na data, vnitřní spojení

SELECT name,rentals FROM ( SELECT name,COUNT( ) as rentals FROM customer JOIN operation ON (customer.id = operation.customer_id) JOIN rental ON (operation.id = rental.operation_id) GROUP BY customer.name ) AS customer_stats WHERE rentals = ( SELECT MAX(rentals) FROM ( SELECT COUNT( ) as rentals FROM customer JOIN operation ON customer.id = operation.customer_id JOIN rental ON operation.id = rental.operation_id GROUP BY customer.name ) AS subquery );

-- most expensive car to rent - podmínku na hodnotu agregační funkce

 SELECT plate_number, brand,price_per_day AS price_per_day FROM (SELECT car.plate_number, car.brand, rental.price_per_day FROM car JOIN operation o ON car.id = o.car_id JOIN rental ON o.id = rental.operation_id GROUP BY plate_number, price_per_day,car.brand HAVING price_per_day = (SELECT MAX(price_per_day) FROM ( car JOIN operation o ON car.id = o.car_id JOIN rental ON o.id = rental.operation_id))) as pnbppd;

-- list the brands of cars that individual customers have rented - stránkování

SELECT brand,name FROM car JOIN operation o on car.id = o.car_id JOIN customer c on o.customer_id = c.id JOIN rental r on o.id = r.operation_id GROUP BY brand,name ORDER BY brand ASC LIMIT 4 OFFSET 2;

-- selects the names of employees who are not assigned to any service,limited by 15 - vnější spojení tabulek SELECT employee.name FROM employee LEFT JOIN is_responsible_for ON employee.id = is_responsible_for.employee_id WHERE is_responsible_for.service_id IS NULL LIMIT 15 OFFSET 0;

-- DELETE FROM is_responsible_for WHERE employee_id = 6;

-- fill employee table with 32k generated data

INSERT INTO employee (name, phone, role) SELECT 'Employee ' || gs.series AS name, '+420' || LPAD((100000000 + gs.series) :: TEXT, 9, '0') AS phone, CASE WHEN gs.series % 2 = 0 THEN 'Manager' ELSE 'Worker' END AS role FROM generate_series(32, 32000) AS gs(series);

-- SELECT COUNT( ) from employee; -- SELECT COUNT( ) FROM car;

-- role definiton for another member

GRANT ALL PRIVILEGES ON DATABASE kopecfi3 TO fronelad;

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO fronelad;

GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO fronelad;

GRANT ALL PRIVILEGES ON SCHEMA public TO fronelad;
```