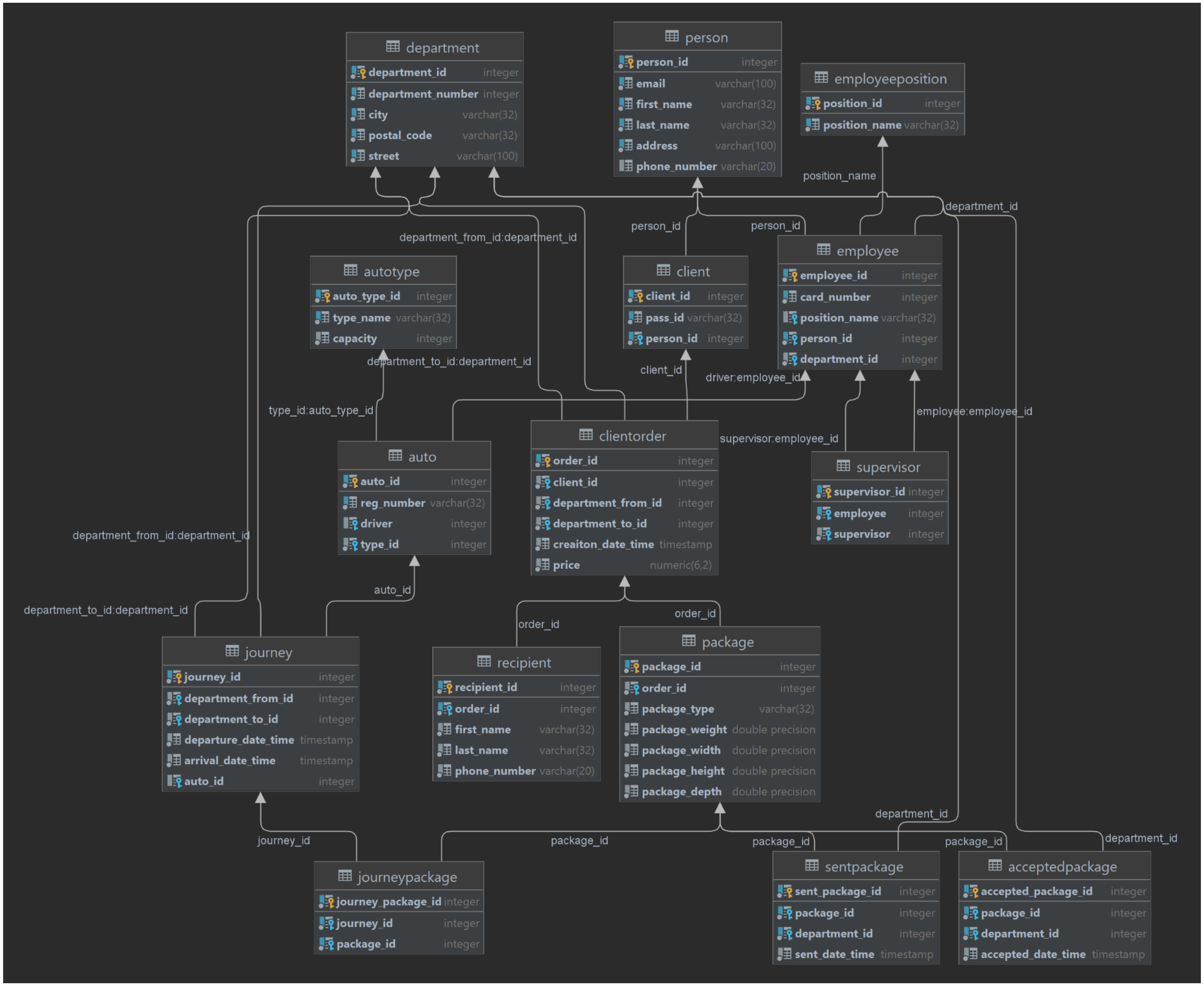
CP3 SQL

ER Model


department
department_id integer
department_number integer
city varchar(32)
postal_code varchar(32)
street varchar(100)
person
person_id integer
email varchar(100)
first_name varchar(32)
last_name varchar(32)
address varchar(100)
phone_number varchar(20)
employeeposition
position_id integer
position_name varchar(32)
position_name
department_id
person_id
person_id
department_from_id:department_id
autotype
auto_type_id integer
type_name varchar(32)
capacity integer
client
client_id integer
pass_id varchar(32)
person_id integer
employee
employee_id integer
card_number integer
position_name varchar(32)
person_id integer
department_id integer
department_to_id:department_id
client_id
driver:employee_id
type_id:auto_type_id
employee:employee_id
supervisor:employee_id
auto
auto_id integer
reg_number varchar(32)
driver integer
type_id integer
clientorder
order_id integer
client_id integer
department_from_id integer
department_to_id integer
creaiton_date_time timestamp
price numeric(6,2)
supervisor
supervisor_id integer
employee integer
supervisor integer
department_from_id:department_id
auto_id
department_to_id:department_id
order_id
order_id
journey
journey_id integer
department_from_id integer
department_to_id integer
departure_date_time timestamp
arrival_date_time timestamp
auto_id integer
recipient
recipient_id integer
order_id integer
first_name varchar(32)
last_name varchar(32)
phone_number varchar(20)
package
package_id integer
order_id integer
package_type varchar(32)
package_weight double precision
package_width double precision
package_height double precision
package_depth double precision
department_id
journey_id
package_id
package_id
package_id
department_id
journeypackage
journey_package_id integer
journey_id integer
package_id integer
sentpackage
sent_package_id integer
package_id integer
department_id integer
sent_date_time timestamp
acceptedpackage
accepted_package_id integer
package_id integer
department_id integer
accepted_date_time timestamp

SQL dotazy pro vytváření tabulek

Psal jsem to ručně)

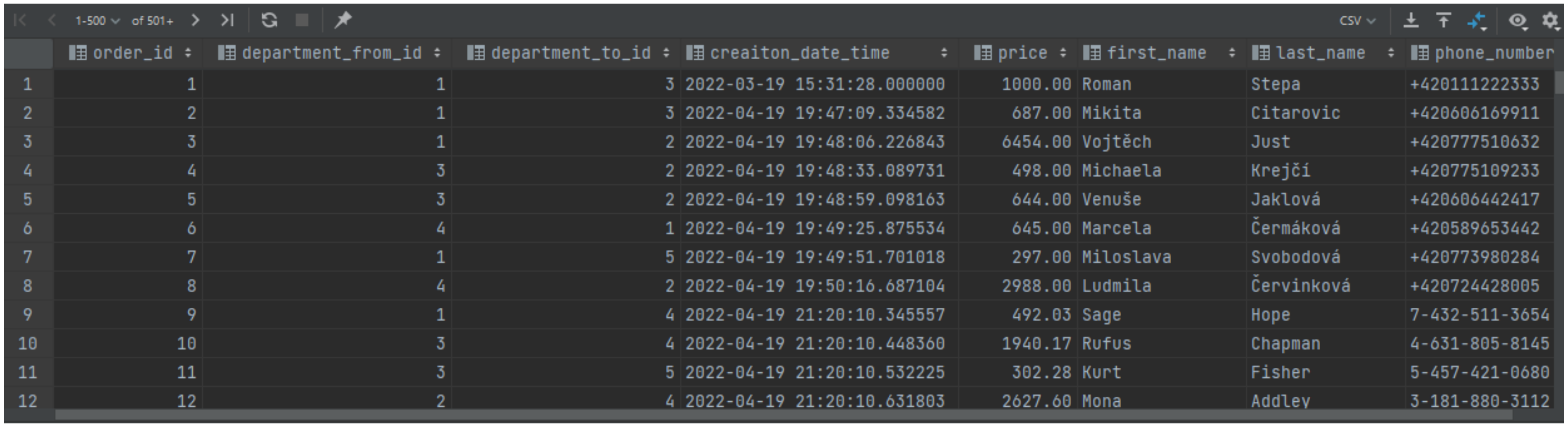
```sql
DROP TABLE IF EXISTS JourneyPackage, Journey;
DROP TABLE IF EXISTS AcceptedPackage, SentPackage, Package, ClientOrder, Recipient;
DROP TABLE IF EXISTS Auto, Autotype;
DROP TABLE IF EXISTS Supervisor, Client, employee, EmployeePosition, Person;
DROP TABLE IF EXISTS Department;

/* Department */

CREATE TABLE IF NOT EXISTS Department (
    department_id SERIAL PRIMARY KEY,
    department_number INT NOT NULL,
    city VARCHAR(32) NOT NULL,
    postal_code VARCHAR(32) NOT NULL,
    street VARCHAR(100) NOT NULL,
    CONSTRAINT department_number_key UNIQUE(department_number),
    CONSTRAINT department_address_key UNIQUE(city, postal_code, street)
);

/* Person */

CREATE TABLE IF NOT EXISTS Person (
    person_id SERIAL PRIMARY KEY,
    email VARCHAR(100) NOT NULL,
    first_name VARCHAR(32) NOT NULL,
    last_name VARCHAR(32) NOT NULL,
    address VARCHAR(100) NOT NULL,
    phone_number VARCHAR(20),
    CONSTRAINT email_key UNIQUE(email),
    CONSTRAINT person_data_key UNIQUE(first_name, last_name, address)
);

CREATE TABLE IF NOT EXISTS Client (
    client_id SERIAL PRIMARY KEY,
    pass_id VARCHAR(32) NOT NULL,
    person_id INT NOT NULL,
    CONSTRAINT client_pass_id_key UNIQUE(pass_id),
    CONSTRAINT client_person_id_key UNIQUE(person_id),
    CONSTRAINT fk_person_id FOREIGN KEY(person_id) REFERENCES Person(person_id) ON DELETE
CASCADE
);

CREATE TABLE IF NOT EXISTS EmployeePosition (
    position_id SERIAL PRIMARY KEY,
    position_name VARCHAR(32) NOT NULL,
    CONSTRAINT employeeposition_name_key UNIQUE(position_name)
);

CREATE TABLE IF NOT EXISTS Employee (
    employee_id SERIAL PRIMARY KEY,
    card_number INT NOT NULL,
    position_name VARCHAR(32) DEFAULT NULL,
    person_id INT NOT NULL,
    department_id INT NOT NULL,
    CONSTRAINT employee_card_number_key UNIQUE(card_number),
    CONSTRAINT employee_person_id_key UNIQUE(person_id),
    CONSTRAINT fk_position_name FOREIGN KEY(position_name) REFERENCES
EmployeePosition(position_name) ON DELETE SET NULL ON UPDATE CASCADE,
    CONSTRAINT fk_person_id FOREIGN KEY(person_id) REFERENCES Person(person_id) ON DELETE
CASCADE,
    CONSTRAINT fk_department_id FOREIGN KEY(department_id) REFERENCES
Department(department_id)
);

CREATE TABLE IF NOT EXISTS Supervisor (
    supervisor_id SERIAL PRIMARY KEY,
    employee INT NOT NULL,
    supervisor INT NOT NULL,
    CONSTRAINT supervisor_employee_key UNIQUE(employee),
    CONSTRAINT fk_employee FOREIGN KEY(employee) REFERENCES Employee(employee_id) ON
DELETE CASCADE,
    CONSTRAINT fk_supervisor FOREIGN KEY(supervisor) REFERENCES Employee(employee_id) ON
DELETE CASCADE
);

/* Auto */

CREATE TABLE IF NOT EXISTS AutoType (
    auto_type_id SERIAL PRIMARY KEY,
    type_name VARCHAR(32) NOT NULL,
    capacity INT NOT NULL,
    CONSTRAINT autotype_type_name_key UNIQUE(type_name)
);

CREATE TABLE IF NOT EXISTS Auto (
    auto_id SERIAL PRIMARY KEY,
    reg_number VARCHAR(32) NOT NULL,
    driver INT DEFAULT NULL,
    type_id INT NOT NULL,
    CONSTRAINT auto_reg_number_key UNIQUE(reg_number),
    CONSTRAINT fk_driver FOREIGN KEY(driver) REFERENCES Employee(employee_id) ON DELETE
SET NULL,
    CONSTRAINT fk_type_id FOREIGN KEY(type_id) REFERENCES AutoType(auto_type_id)
);

/* Order */

CREATE TABLE IF NOT EXISTS ClientOrder (
  order_id SERIAL PRIMARY KEY,
    client_id INT NOT NULL,
    department_from_id INT NOT NULL,
    department_to_id INT NOT NULL CHECK (department_from_id != department_to_id),
    creaiton_date_time timestamp NOT NULL DEFAULT current_timestamp,
    price DECIMAL(6, 2) NOT NULL,
    CONSTRAINT fk_client_id FOREIGN KEY(client_id) REFERENCES Client(client_id) ON DELETE
CASCADE,
    CONSTRAINT fk_department_from_id_key FOREIGN KEY(department_from_id) REFERENCES
Department(department_id),
    CONSTRAINT fk_department_to_id_key FOREIGN KEY(department_to_id) REFERENCES
Department(department_id)
);

CREATE TABLE IF NOT EXISTS Recipient (
    recipient_id SERIAL PRIMARY KEY,
    order_id INT NOT NULL,
    first_name VARCHAR(32) NOT NULL,
    last_name VARCHAR(32) NOT NULL,
    phone_number VARCHAR(20) NOT NULL,
    CONSTRAINT recipient_order_id_ky UNIQUE(order_id),
    CONSTRAINT fk_order_id FOREIGN KEY(order_id) REFERENCES ClientOrder(order_id) ON
DELETE CASCADE
);

/* Package */

CREATE TABLE IF NOT EXISTS Package (
    package_id SERIAL PRIMARY KEY,
    order_id INT NOT NULL,
    package_type VARCHAR(32) NOT NULL,
    package_weight FLOAT NOT NULL,
    package_width FLOAT NOT NULL,
    package_height FLOAT NOT NULL,
    package_depth FLOAT NOT NULL,
    CONSTRAINT fk_order_id FOREIGN KEY(order_id) REFERENCES ClientOrder(order_id) ON
DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS AcceptedPackage (
    accepted_package_id SERIAL PRIMARY KEY,
    package_id INT NOT NULL,
    department_id INT NOT NULL,
    accepted_date_time timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT acceptedpackage_data_key UNIQUE(package_id, department_id),
    CONSTRAINT fk_package_id FOREIGN KEY(package_id) REFERENCES Package(package_id) ON
DELETE CASCADE,
    CONSTRAINT fk_department_id FOREIGN KEY(department_id) REFERENCES
Department(department_id)
);

CREATE TABLE IF NOT EXISTS SentPackage (
    sent_package_id SERIAL PRIMARY KEY,
    package_id INT NOT NULL,
    department_id INT NOT NULL,
    sent_date_time timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT sentpackage_data_key UNIQUE(package_id, department_id),
    CONSTRAINT fk_package_id FOREIGN KEY(package_id) REFERENCES Package(package_id) ON
DELETE CASCADE,
    CONSTRAINT fk_department_id FOREIGN KEY(department_id) REFERENCES
Department(department_id)
);

/* Journey */

CREATE TABLE IF NOT EXISTS Journey (
    journey_id SERIAL PRIMARY KEY,
    department_from_id INT NOT NULL,
    department_to_id INT NOT NULL,
    departure_date_time timestamp NOT NULL,
    arrival_date_time timestamp NOT NULL CHECK (departure_date_time < arrival_date_time),
    auto_id INT DEFAULT NULL,
    CONSTRAINT fk_department_from_id FOREIGN KEY(department_from_id) REFERENCES
Department(department_id),
    CONSTRAINT fk_department_to_id FOREIGN KEY(department_to_id) REFERENCES
Department(department_id),
    CONSTRAINT fk_auto_id FOREIGN KEY(auto_id) REFERENCES Auto(auto_id) ON DELETE SET
NULL
);

CREATE TABLE IF NOT EXISTS JourneyPackage (
    journey_package_id SERIAL PRIMARY KEY,
    journey_id INT NOT NULL,
    package_id INT NOT NULL,
    CONSTRAINT journey_package_data_key UNIQUE(journey_id, package_id),
    CONSTRAINT fk_journey_id FOREIGN KEY(journey_id) REFERENCES Journey(journey_id) ON
DELETE CASCADE,
    CONSTRAINT fk_package_id FOREIGN KEY(package_id) REFERENCES Package(package_id) ON
DELETE CASCADE
);
```

SQL dotazy pro získání údajů z databáze

1. Vnější spojení tabulek

1. Získání seznamu všech objednávek najednou s údaji o příjemci. (**OUTER JOIN**)

```
SELECT clientorder.order_id,
       clientorder.department_from_id,
       clientorder.department_to_id,
       clientorder.creaiton_date_time,
       clientorder.price,
       recipient.first_name,
       recipient.last_name,
       recipient.phone_number
     FROM ClientOrder
     FULL OUTER JOIN Recipient
     ON ClientOrder.order_id = Recipient.order_id;
```

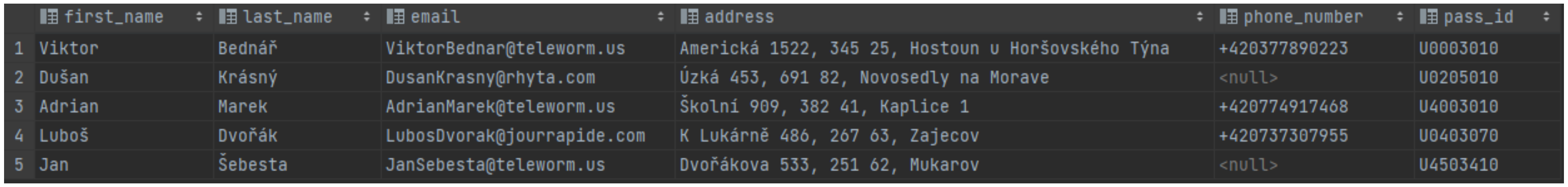


	order_id	department_from_id	department_to_id	creaiton_date_time	price	first_name	last_name	phone_number
1	1	1	3	2022-03-19 15:31:28.000000	1000.00	Roman	Stepa	+420111222333
2	2	1	3	2022-04-19 19:47:09.334582	687.00	Mikita	Citarovic	+420606169911
3	3	1	2	2022-04-19 19:48:06.226843	6454.00	Vojtěch	Just	+420777510632
4	4	3	2	2022-04-19 19:48:33.089731	498.00	Michaela	Krejčí	+420775109233
5	5	3	2	2022-04-19 19:48:59.098163	644.00	Venuše	Jaklová	+420606442417
6	6	4	1	2022-04-19 19:49:25.875534	645.00	Marcela	Čermáková	+420589653442
7	7	1	5	2022-04-19 19:49:51.701018	297.00	Miloslava	Svobodová	+420773980284
8	8	4	2	2022-04-19 19:50:16.687104	2988.00	Ludmila	Červinková	+420724428005
9	9	1	4	2022-04-19 21:20:10.345557	492.03	Sage	Hope	7-432-511-3654
10	10	3	4	2022-04-19 21:20:10.448360	1940.17	Rufus	Chapman	4-631-805-8145
11	11	3	5	2022-04-19 21:20:10.532225	302.28	Kurt	Fisher	5-457-421-0680
12	12	2	4	2022-04-19 21:20:10.631803	2627.60	Mona	Addley	3-181-880-3112

2. Získání seznamu všech klientů najednou se všemi údaji. (**LEFT JOIN**)

```
SELECT person.first_name,
          person.last_name,
          person.email,
          person.address,
          person.phone_number,
          client.pass_id
```

```
    FROM Client
    LEFT JOIN Person
    ON client.person_id = person.person_id;
```

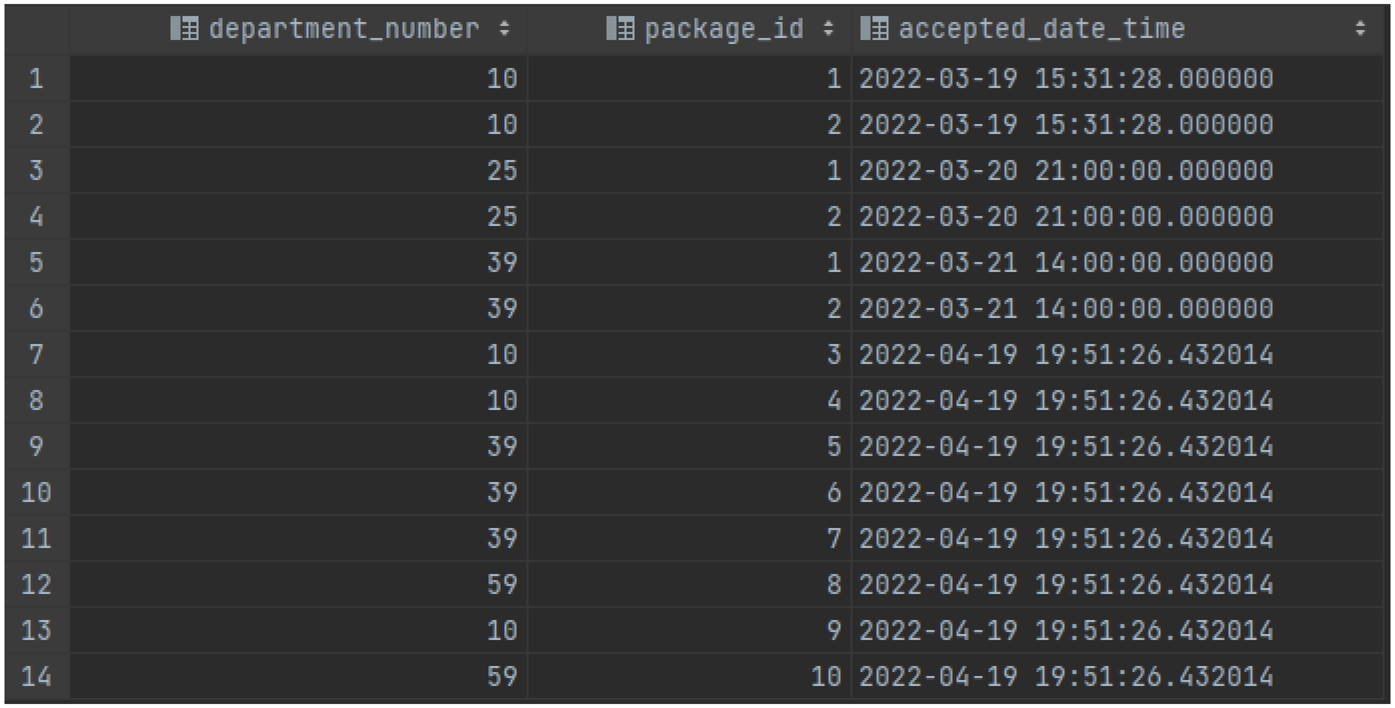


	first_name	last_name	email	address	phone_number	pass_id
1	Viktor	Bednář	ViktorBednar@teleworm.us	Americká 1522, 345 25, Hostoun u Horšovského Týna	+420377890223	U0003010
2	Dušan	Krásný	DusanKrasny@rhyta.com	Úzká 453, 691 82, Novosedly na Morave	<null>	U0205010
3	Adrian	Marek	AdrianMarek@teleworm.us	Školní 909, 382 41, Kaplice 1	+420774917468	U4003010
4	Luboš	Dvořák	LubosDvorak@jourrapide.com	K Lukárně 486, 267 63, Zajecov	+420737307955	U0403070
5	Jan	Šebesta	JanSebesta@teleworm.us	Dvořákova 533, 251 62, Mukarov	<null>	U4503410

2. Vnitřní spojení tabulek

Získání informací o tom, jaké zásilky a v jaké době přijaly pobočky.

```
SELECT department.department_number,
       acceptedpackage.package_id,
       acceptedpackage.accepted_date_time
FROM   AcceptedPackage
       INNER JOIN Department
               ON acceptedpackage.department_id = department.department_id;
```

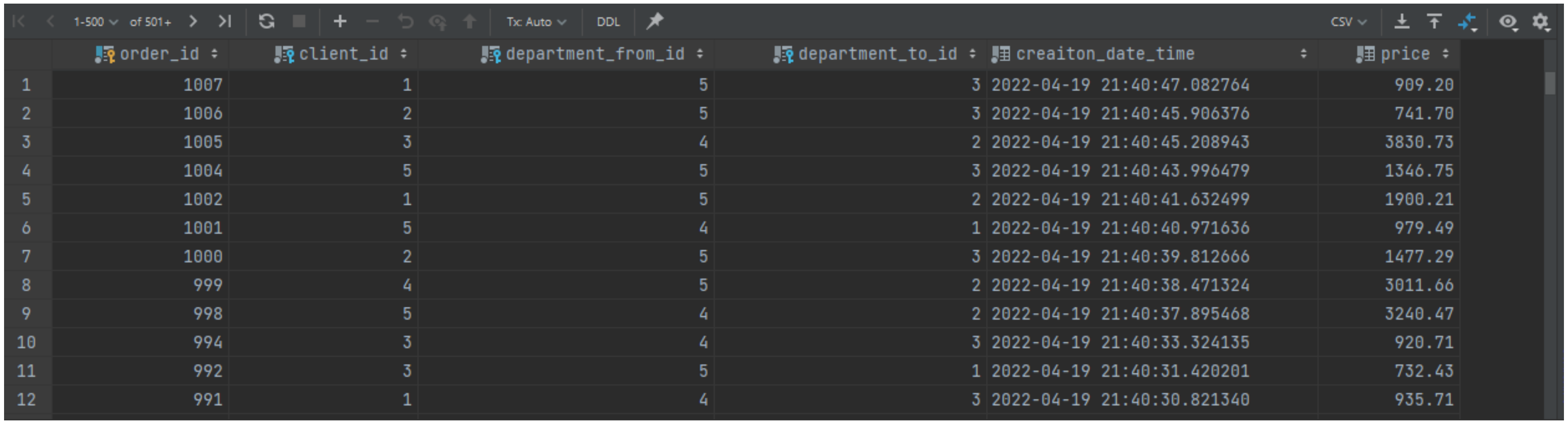


	department_number	package_id	accepted_date_time
1	10	1	2022-03-19 15:31:28.000000
2	10	2	2022-03-19 15:31:28.000000
3	25	1	2022-03-20 21:00:00.000000
4	25	2	2022-03-20 21:00:00.000000
5	39	1	2022-03-21 14:00:00.000000
6	39	2	2022-03-21 14:00:00.000000
7	10	3	2022-04-19 19:51:26.432014
8	10	4	2022-04-19 19:51:26.432014
9	39	5	2022-04-19 19:51:26.432014
10	39	6	2022-04-19 19:51:26.432014
11	39	7	2022-04-19 19:51:26.432014
12	59	8	2022-04-19 19:51:26.432014
13	10	9	2022-04-19 19:51:26.432014
14	59	10	2022-04-19 19:51:26.432014

3. Podmínka na data

Získání objednávek vytvořených později než datum 2022-04-17 a které mají vyšší cenu než 600, také seřazené podle data.

```
SELECT *
    FROM   ClientOrder
    WHERE  creaiton_date_time > '2022-04-17'
           AND price > 600
    ORDER  BY creaiton_date_time DESC;
```

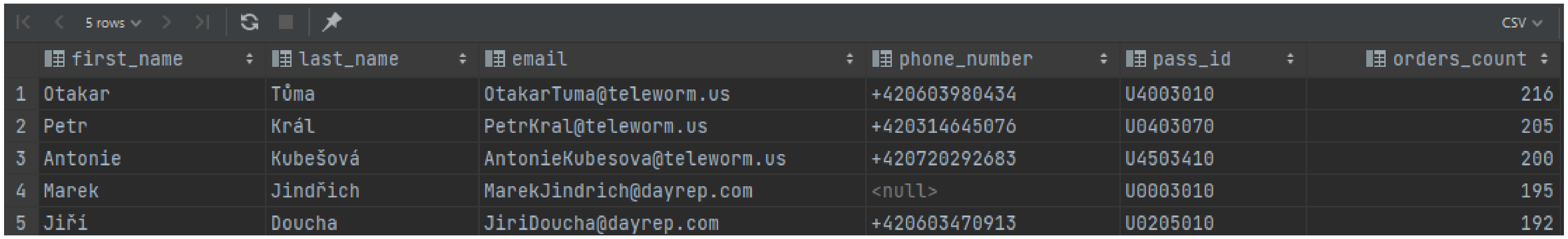


	order_id	client_id	department_from_id	department_to_id	creaiton_date_time	price
1	1007	1	5	3	2022-04-19 21:40:47.082764	909.20
2	1006	2	5	3	2022-04-19 21:40:45.906376	741.70
3	1005	3	4	2	2022-04-19 21:40:45.208943	3830.73
4	1004	5	5	3	2022-04-19 21:40:43.996479	1346.75
5	1002	1	5	2	2022-04-19 21:40:41.632499	1900.21
6	1001	5	4	1	2022-04-19 21:40:40.971636	979.49
7	1000	2	5	3	2022-04-19 21:40:39.812666	1477.29
8	999	4	5	2	2022-04-19 21:40:38.471324	3011.66
9	998	5	4	2	2022-04-19 21:40:37.895468	3240.47
10	994	3	4	3	2022-04-19 21:40:33.324135	920.71
11	992	3	5	1	2022-04-19 21:40:31.420201	732.43
12	991	1	4	3	2022-04-19 21:40:30.821340	935.71

4. Agregaci a podmínku na hodnotu agregační funkce

Získání všech zákazníků, kteří udělali objednávku více než 3 krát s počítáním jejich objednávek a seřazením podle počtu.

```
SELECT person.first_name,
       person.last_name,
       person.email,
       person.phone_number,
       client.pass_id,
       COUNT(clientorder.client_id) AS orders_count
FROM   ClientOrder
       LEFT JOIN Client
              ON clientorder.client_id = client.client_id
       LEFT JOIN Person
              ON client.client_id = person.person_id
GROUP  BY clientorder.client_id,
          person.first_name,
          person.last_name,
          person.email,
          person.phone_number,
          client.pass_id
HAVING COUNT(clientorder.client_id) > 3
ORDER  BY COUNT(clientorder.client_id) DESC
```

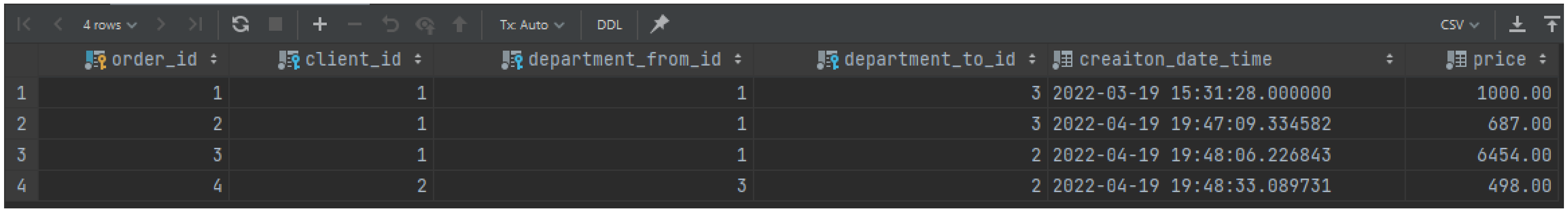


	first_name	last_name	email	phone_number	pass_id	orders_count
1	Otakar	Tůma	OtakarTuma@teleworm.us	+420603980434	U4003010	216
2	Petr	Král	PetrKral@teleworm.us	+420314645076	U0403070	205
3	Antonie	Kubešová	AntonieKubesova@teleworm.us	+420720292683	U4503410	200
4	Marek	Jindřich	MarekJindrich@dayrep.com	<null>	U0003010	195
5	Jiří	Doucha	JiriDoucha@dayrep.com	+420603470913	U0205010	192

5. Řazení a stránkování

Získání čtřech objednavek z první stránky a řazení podle data (od nejsarších).

```
SELECT *
     FROM   ClientOrder
     ORDER  BY creaiton_date_time ASC
     LIMIT  4
     OFFSET 0
```

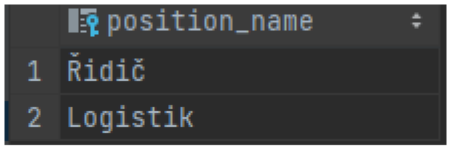


	order_id	client_id	department_from_id	department_to_id	creaiton_date_time	price
1	1	1	1	3	2022-03-19 15:31:28.000000	1000.00
2	2	1	1	3	2022-04-19 19:47:09.334582	687.00
3	3	1	1	2	2022-04-19 19:48:06.226843	6454.00
4	4	2	3	2	2022-04-19 19:48:33.089731	498.00

6. Množinové operace

1. Získání informace, jaké jsou pozice ve dvou různých pobočkách (UNION).

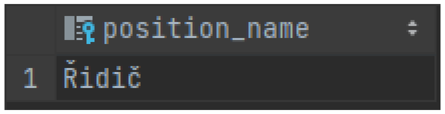
```
SELECT position_name
     FROM   employee
     WHERE  department_id = 4
     UNION
SELECT position_name
     FROM   employee
     WHERE  department_id = 3;
```



```
  position_name
1 Řidič
2 Logistik
```

2. Získání informace, jaké jsou různé pozice ve dvou různých pobočkách (INTERSECT)

```
SELECT position_name
     FROM   employee
     WHERE  department_id = 4
     INTERSECT
SELECT position_name
     FROM   employee
     WHERE  department_id = 3;
```

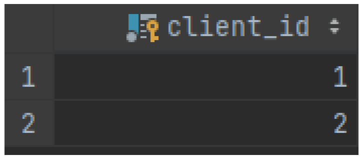


	position_name
1	Řidič

7. Vnořený SELECT

Získání id všech klientů, kteří objednali aspoň jednou.

```
SELECT client.client_id
FROM   client
WHERE  client_id IN (SELECT clientorder.client_id
                     FROM   clientorder);
```



	client_id
1	1
2	2