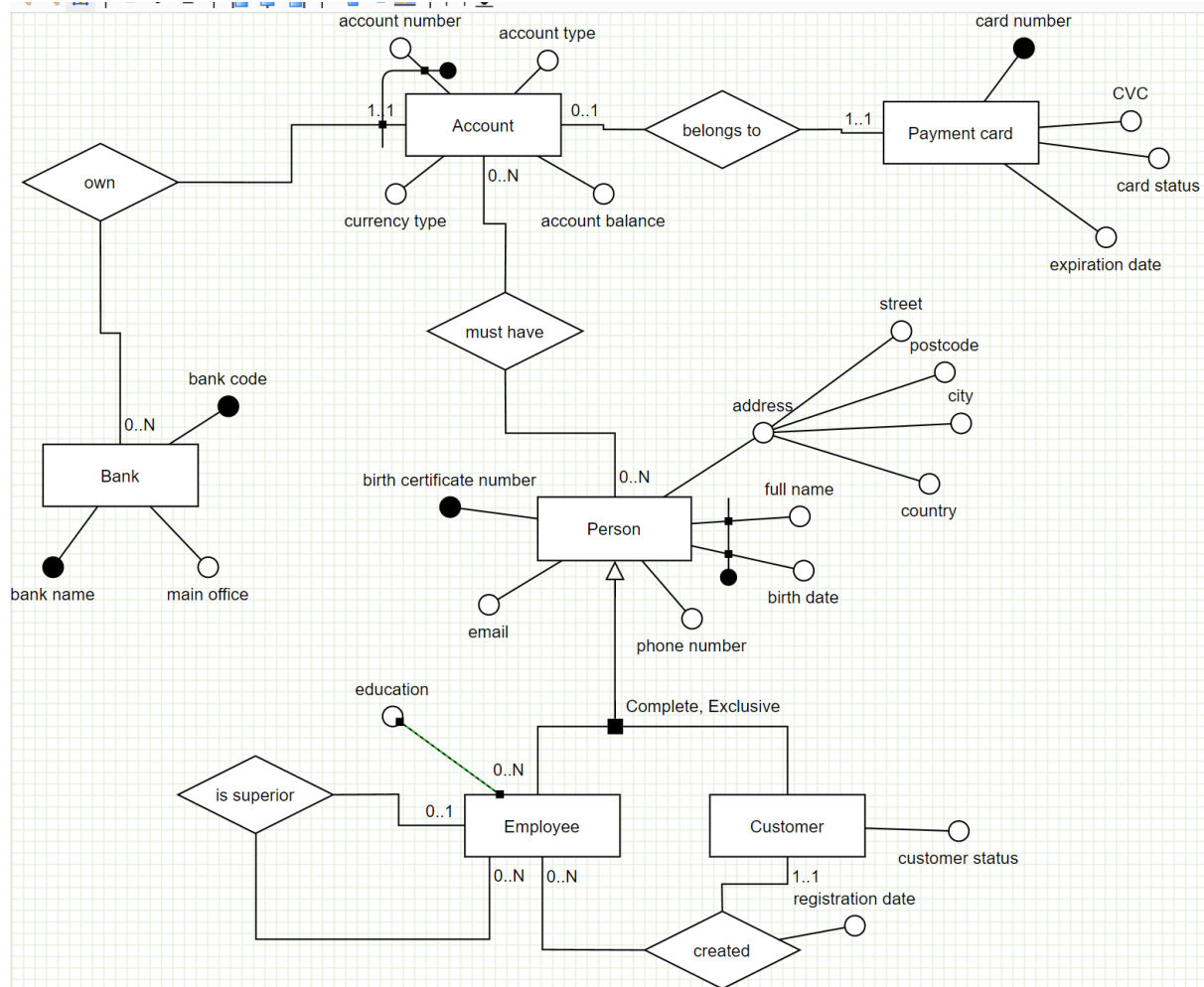


Conceptual model



Relational Model

- **Bank** (bank name, bank code, main office)
- **Account** (bank code, account number, account balance, account type, currency type)
 - FK: (bank code) \subseteq Bank (bank code)
- **Payment_Card** (card number, bank code, account number, CVC, status, expiration date)
 - FK: (bank code, account number) \subseteq Bank (bank code, account number)
- **Person** (birth certificate number, birth date, full name, email, phone number, country, city, postcode, street)
- **Person_Account** (birth certificate number, bank code, account number)
 - FK: (birth certificate number) \subseteq Person (birth certificate number)
 - FK: (bank code, account number) \subseteq Account (bank code, account number)
- **Customer** (birth certificate number, customer status, employee, registration date)
 - FK: (birth certificate number) \subseteq Person (birth certificate number)
 - FK: (employee) \subseteq Employee (birth certificate number)
- **Employee** (birth certificate number)
 - FK: (birth certificate number) \subseteq Person (birth certificate number)
- **Education** (birth certificate number, name of institution)
 - FK: (birth certificate number) \subseteq Employee (birth certificate number)
- **Is_superior**(employee, superior)
 - FK: (employee) \subseteq Employee (birth certificate number)
 - FK: (superior) \subseteq Employee (birth certificate number)

SQL queries for creating database:

CREATE TABLE Bank (

```
id_bank SERIAL PRIMARY KEY,  
bank_name VARCHAR(30) NOT NULL UNIQUE,  
main_office VARCHAR(30) NOT NULL  
iban VARCHAR(34) NOT NULL UNIQUE  
);
```

```
CREATE TABLE Account (  
id_account INTEGER PRIMARY KEY,  
iban VARCHAR(34) NOT NULL,  
account_number VARCHAR(10) UNIQUE NOT NULL,  
account_balance DECIMAL(15, 2) NOT NULL DEFAULT 0.00,  
account_type VARCHAR(50) NOT NULL CHECK ( account_type IN ('checking', 'savings', 'student',  
'VIP')),  
currency_type VARCHAR(3) NOT NULL CHECK (currency_type IN ('CZK', 'EUR', 'USD')),  
FOREIGN KEY (iban) REFERENCES Bank(iban) ON UPDATE CASCADE ON DELETE CASCADE  
);
```

```
CREATE TABLE Payment_Card (  
id_card SERIAL PRIMARY KEY,  
card_number VARCHAR(20) NOT NULL UNIQUE,  
card_status VARCHAR(7) NOT NULL CHECK (card_status IN ('blocked', 'active', 'expired')),  
expiration_date DATE NOT NULL,  
CVC SMALLINT NOT NULL CHECK (CVC BETWEEN 100 AND 999),  
account_id INTEGER UNIQUE NOT NULL,  
FOREIGN KEY (account_id) REFERENCES Account(id_account) ON UPDATE CASCADE ON DELETE  
CASCADE  
);
```

```
CREATE TABLE Person (  
id_person SERIAL PRIMARY KEY,  
birth_certificate_number VARCHAR(10) UNIQUE NOT NULL CHECK (birth_certificate_number ~  
'^[0-9]+$'),
```

```

    birth_date DATE NOT NULL CHECK (birth_date <= CURRENT_DATE AND birth_date >= '1900-01-01'),
    full_name VARCHAR(50) NOT NULL,
    email VARCHAR(50) NOT NULL CHECK (email ~ '^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'),
    phone_number VARCHAR(13) NOT NULL CHECK (phone_number ~ '^\+[0-9]+$'),
    country VARCHAR(50) NOT NULL,
    city VARCHAR(50) NOT NULL,
    postcode VARCHAR(7) NOT NULL,
    street VARCHAR(50) NOT NULL,
    CONSTRAINT unique_fullname_birthdate UNIQUE (full_name, birth_date)
);

```

```

CREATE TABLE Person_Account (
    id_person_account SERIAL PRIMARY KEY,
    account_id INTEGER NOT NULL,
    person_id INTEGER NOT NULL,
    FOREIGN KEY (account_id) REFERENCES Account(id_account) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (person_id) REFERENCES Person(id_person) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT unique_account_person UNIQUE (account_id, person_id)
);

```

```

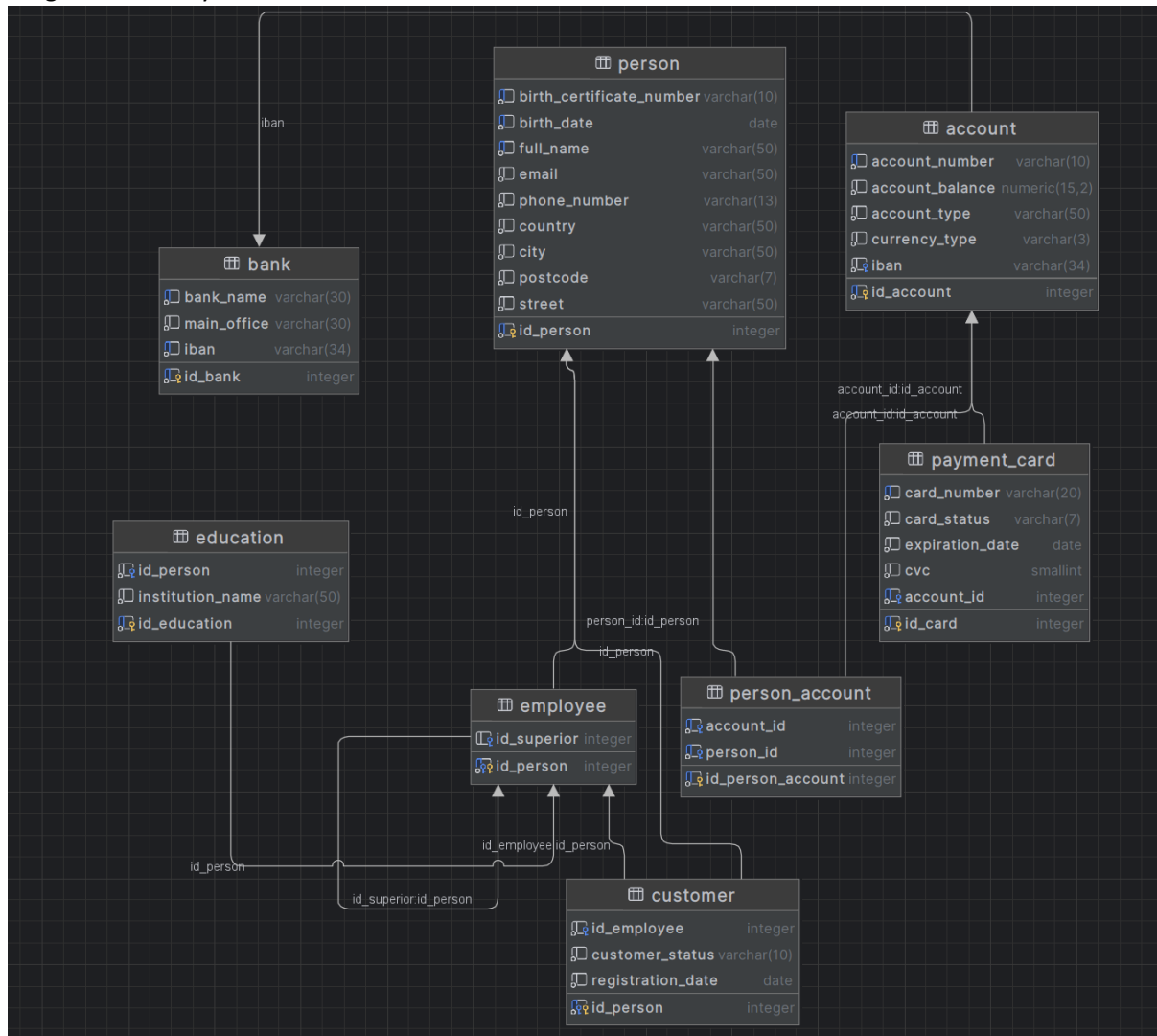
CREATE TABLE Employee (
    id_person INTEGER PRIMARY KEY,
    id_superior INTEGER,
    FOREIGN KEY (id_person) REFERENCES Person(id_person) ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (id_superior) REFERENCES Employee(id_person) ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT different_superior CHECK (id_superior <> id_person)
);

```

```
CREATE TABLE Education (  
    id_education SERIAL PRIMARY KEY,  
    id_person INTEGER NOT NULL,  
    institution_name VARCHAR(50) NOT NULL,  
    FOREIGN KEY (id_person) REFERENCES Employee(id_person) ON UPDATE CASCADE ON DELETE  
    CASCADE  
);
```

```
CREATE TABLE Customer (  
    id_person INTEGER PRIMARY KEY,  
    id_employee INTEGER NOT NULL,  
    customer_status VARCHAR(10) NOT NULL,  
    registration_date DATE NOT NULL CHECK (registration_date <= CURRENT_DATE),  
    FOREIGN KEY (id_person) REFERENCES Person(id_person) ON UPDATE CASCADE ON DELETE  
    CASCADE,  
    FOREIGN KEY (id_employee) REFERENCES Employee(id_person) ON UPDATE CASCADE ON DELETE  
    CASCADE,  
    CONSTRAINT check_employee_not_customer CHECK (id_employee <> id_person)  
);
```

Diagram databazy:



Inserty:

Bank:

INSERT INTO Bank (bank_name, main_office, iban)

VALUES

('Czech Savings Bank', 'Prague', 'CZ0800'),

('Commercial Bank', 'Prague', 'CZ0100'),

('CSOB', 'Prague', 'CZ0300'),

('UniCredit Bank', 'Prague', 'CZ2700'),

('Raiffeisenbank', 'Prague', 'CZ5500'),

('Air Bank', 'Prague', 'CZ1000'),

('Fio banka', 'Brno', 'CZ2100'),

```
('mBank', 'Warsaw', 'PL2200'),  
( 'Expobank', 'Prague', 'CZ2800'),  
( 'Českomoravská hypoteční banka', 'Prague', 'CZ8100');
```

Account:

```
INSERT INTO Account (id_account, iban, account_number, account_balance,  
account_type, currency_type)
```

VALUES

```
(1, '0800', '123456', 1000.00, 'checking', 'CZK'),  
(2, '0800', '987654', 5000.00, 'savings', 'EUR'),  
(3, '0100', '123456', 2000.00, 'checking', 'USD'),  
(4, '0100', '987654', 3000.00, 'savings', 'CZK'),  
(5, '0300', '123456', 1500.00, 'checking', 'EUR'),  
(6, '0300', '987654', 2500.00, 'savings', 'USD'),  
(7, '2700', '123456', 4000.00, 'checking', 'CZK'),  
(8, '2700', '987654', 6000.00, 'savings', 'EUR'),  
(9, '5500', '123456', 7000.00, 'checking', 'USD'),  
(10, '5500', '987654', 8000.00, 'savings', 'CZK'),  
(11, '1000', '765432', 1000.00, 'checking', 'CZK'),  
(12, '1000', '987654', 5000.00, 'savings', 'EUR'),  
(13, '2100', '123456', 2000.00, 'checking', 'USD'),  
(14, '2100', '987654', 3000.00, 'savings', 'CZK'),  
(15, '2200', '123456', 1500.00, 'checking', 'EUR'),  
(16, '2200', '987654', 2500.00, 'savings', 'USD'),  
(17, '2800', '123456', 4000.00, 'checking', 'CZK'),  
(18, '2800', '987654', 6000.00, 'savings', 'EUR'),  
(19, '8100', '123456', 7000.00, 'checking', 'USD'),  
(20, '8100', '987654', 8000.00, 'savings', 'CZK');
```

Payment_card:

```
INSERT INTO Payment_Card (card_number, card_status, expiration_date, CVC,  
account_id)
```

```
VALUES
```

```
('1111222233334444', 'active', '2025-06-30', 111, 1),  
( '2222333344445555', 'blocked', '2025-07-31', 222, 2),  
( '4444555566667777', 'active', '2025-09-30', 444, 4),  
( '5555666677778888', 'blocked', '2025-10-31', 555, 5),  
( '6666777788889999', 'active', '2025-11-30', 666, 6),  
( '7777888899990000', 'active', '2025-12-31', 777, 7),  
( '9999000011112222', 'active', '2026-02-28', 999, 9),  
( '0000111122223333', 'blocked', '2026-03-31', 100, 10),  
( '3333444455556666', 'expired', '2024-01-02', 333, 3),  
( '8888999900001111', 'expired', '2024-01-01', 888, 8),  
( '1234567890123456', 'active', '2026-06-30', 111, 11),  
( '2345678901234567', 'active', '2026-07-31', 222, 12),  
( '3456789012345678', 'active', '2026-08-30', 333, 13),  
( '4567890123456789', 'active', '2026-09-30', 444, 14),  
( '5678901234567890', 'active', '2026-10-31', 555, 15),  
( '6789012345678901', 'active', '2026-11-30', 666, 16),  
( '7890123456789012', 'active', '2026-12-31', 777, 17),  
( '8901234567890123', 'active', '2027-02-28', 888, 18),  
( '9012345678901234', 'active', '2027-03-31', 999, 19),  
( '0123456789012345', 'active', '2027-04-30', 100, 20);
```

Person:

```
INSERT INTO Person (birth_certificate_number, birth_date, full_name, email,  
phone_number, country, city, postcode, street)
```

VALUES

('1234567890', '1990-01-01', 'John Doe', 'john.doe@example.com', '+1234567890', 'USA', 'New York', '10001', 'Main Street'),

('2345678901', '1991-02-02', 'Alice Smith', 'alice.smith@example.com', '+2345678901', 'UK', 'London', 'SW1A', 'King Street'),

('3456789012', '1992-03-03', 'Michael Johnson', 'michael.johnson@example.com', '+3456789012', 'Canada', 'Toronto', 'M5H', 'Queen Street'),

('4567890123', '1993-04-04', 'Emily Brown', 'emily.brown@example.com', '+4567890123', 'Australia', 'Sydney', '2000', 'George Street'),

('5678901234', '1994-05-05', 'Daniel Wilson', 'daniel.wilson@example.com', '+5678901234', 'Germany', 'Berlin', '10117', 'Friedrichstraße'),

('6789012345', '1995-06-06', 'Sophia Martinez', 'sophia.martinez@example.com', '+6789012345', 'France', 'Paris', '75001', 'Rue de Rivoli'),

('7890123456', '1996-07-07', 'Matthew Taylor', 'matthew.taylor@example.com', '+7890123456', 'Italy', 'Rome', '00184', 'Via del Corso'),

('8901234567', '1997-08-08', 'Olivia Garcia', 'olivia.garcia@example.com', '+8901234567', 'Spain', 'Madrid', '28001', 'Calle de Alcalá'),

('9012345678', '1998-09-09', 'Liam Rodriguez', 'liam.rodriguez@example.com', '+9012345678', 'Japan', 'Tokyo', '1000001', 'Chiyoda'),

('0123456789', '1999-10-10', 'Emma Lopez', 'emma.lopez@example.com', '+0123456789', 'Brazil', 'São Paulo', '01000', 'Avenida Paulista'),

('9876543210', '2000-11-11', 'Noah Perez', 'noah.perez@example.com', '+9876543210', 'India', 'Mumbai', '400001', 'Nariman Point'),

('1111111111', '1990-01-01', 'Emma Watson', 'emma.watson@example.com', '+1111111111', 'USA', 'Los Angeles', '90001', 'Hollywood Blvd'),

('2222222222', '1991-02-02', 'Jack Johnson', 'jack.johnson@example.com', '+2222222222', 'Canada', 'Vancouver', 'V5K', 'Granville Street'),

('3333333333', '1992-03-03', 'Sophie Turner', 'sophie.turner@example.com', '+3333333333', 'UK', 'Manchester', 'M1', 'Deansgate'),

('4444444444', '1993-04-04', 'Ryan Reynolds', 'ryan.reynolds@example.com', '+4444444444', 'Australia', 'Melbourne', '3000', 'Collins Street'),

('5555555555', '1994-05-05', 'Natalie Portman', 'natalie.portman@example.com', '+5555555555', 'France', 'Nice', '06000', 'Promenade des Anglais'),

('6666666666', '1995-06-06', 'Leonardo DiCaprio',
'leonardo.dicaprio@example.com', '+6666666666', 'Italy', 'Florence', '50123', 'Ponte
Vecchio'),

('7777777777', '1996-07-07', 'Jennifer Lawrence', 'jennifer.lawrence@example.com',
'+7777777777', 'Spain', 'Barcelona', '08001', 'Las Ramblas'),

('8888888888', '1997-08-08', 'Chris Hemsworth', 'chris.hemsworth@example.com',
'+8888888888', 'Japan', 'Osaka', '5500002', 'Umeda'),

('9999999999', '1998-09-09', 'Scarlett Johansson',
'scarlett.johansson@example.com', '+9999999999', 'Brazil', 'Rio de Janeiro', '20040',
'Copacabana Beach'),

('0000000000', '1999-10-10', 'Brad Pitt', 'brad.pitt@example.com', '+0000000000',
'India', 'New Delhi', '110001', 'Connaught Place');

Person_account: INSERT INTO Person_Account (account_id, person_id)

VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(10, 10),

(10, 11),

(11, 15),

(12, 16),

(13, 17),

(14, 18),

(15, 19),

(16, 20),

(17, 21),

(18, 22),

(19, 23),

(20, 24);

Employee:

INSERT INTO Employee (id_person, id_superior)

VALUES

(1, NULL),

(2, 1),

(3, 2),

(4, 3),

(5, 4);

Customer:

INSERT INTO Customer (id_person, id_employee, customer_status, registration_date)

VALUES

(6, 1, 'active', '2024-04-18'),

(7, 2, 'active', '2024-04-18'),

(8, 3, 'active', '2024-04-18'),

(9, 4, 'active', '2024-04-18'),

(10, 5, 'active', '2024-04-18'),

(11, 1, 'active', '2024-04-18'),

(15, 2, 'active', '2024-04-18'),

(16, 3, 'active', '2024-04-18'),

(17, 4, 'active', '2024-04-18'),

(18, 5, 'active', '2024-04-18'),

(19, 1, 'active', '2024-04-18'),

(20, 2, 'active', '2024-04-18'),

(21, 3, 'active', '2024-04-18'),
(22, 4, 'active', '2024-04-18'),
(23, 5, 'active', '2024-04-18'),
(24, 1, 'active', '2024-04-18');

Education:

INSERT INTO Education (id_person, institution_name)
VALUES

(1, 'Institution 1'),
(2, 'Institution 2'),
(3, 'Institution 3'),
(4, 'Institution 4'),
(5, 'Institution 5');

SQL dotazy pro získání údajů z databáze :

1(vnější spojení tabulek + podmínku na data) :

SELECT *
FROM Person
LEFT OUTER JOIN Person_Account ON Person.id_person = Person_Account.person_id
LEFT OUTER JOIN Account ON Person_Account.account_id = Account.id_account
WHERE Account.currency_type = 'CZK';

Cinnost: Vypiše ke každému uctu , cloveka , kteremu ten ucet patri pres vnejsi spojeni tabulek a jeste ma podminky kvuli ktere vypise jen ty ucty ktere maji typ meny czk.

Vysledek:

id_person	birth_certificate_number	birth_date	full_name	email	phone_number	country	city	postcode	street	id_person_account	account_id	person_id	id_account	account_number	account_balance	account_type
1	1234567890	1990-01-01	John Doe	john.doe@example.com	+1234567890	USA	New York	10001	Main Street	1	1	1	1	1000123456	1000.00	checking
4	4567890123	1993-04-04	Emily Brown	emily.brown@example.com	+4567890123	Australia	Sydney	2000	George Street	4	4	4	4	40100987654	3000.00	savings
7	7890123456	1996-07-07	Matthae Taylor	matthae.taylor@example.com	+7890123456	Italy	Rome	00184	Via del Corso	7	7	7	7	72780123456	4000.00	checking
10	0123456789	1999-10-10	Emma Lopez	emma.lopez@example.com	+0123456789	Brazil	São Paulo	01000	Avenida Paulista	10	10	10	10	105500987654	8000.00	savings
11	9876543210	2000-11-11	Noah Perez	noah.perez@example.com	+9876543210	India	Mumbai	400001	Nariman Point	11	10	11	10	105500987654	8000.00	savings
15	1111111111	1998-01-01	Emma Watson	emma.watson@example.com	+1111111111	USA	Los Angeles	90001	Hollywood Blvd	12	11	15	11	1000765432	1000.00	checking
18	4444444444	1993-04-04	Ryan Reynolds	ryan.reynolds@example.com	+4444444444	Australia	Melbourne	3000	Collins Street	15	14	18	14	2100987654	3000.00	savings
21	7777777777	1996-07-07	Jennifer Lawrence	jennifer.lawrence@example.com	+7777777777	Spain	Barcelona	08001	Las Ramblas	18	17	21	17	2800123456	4000.00	checking
24	0000000000	1999-10-10	Brad Pitt	brad.pitt@example.com	+0000000000	India	New Delhi	110001	Connaught Place	21	20	24	20	8100987654	8000.00	savings

account_type ▾	currency_type ▾	iban ▾
checking	CZK	CZ0800
savings	CZK	CZ0100
checking	CZK	CZ2700
savings	CZK	CZ5500
savings	CZK	CZ5500
checking	CZK	CZ1000
savings	CZK	CZ2100
checking	CZK	CZ2800
savings	CZK	CZ8100

2 (vnitřní spojení tabulek):

SELECT *

FROM Account

INNER JOIN Person_Account ON Account.id_account = Person_Account.account_id

INNER JOIN Person ON Person_Account.person_id = Person.id_person;

Cinnost: Vypiše ke každému uctu , cloveka , kteremu ten ucet patri pres vnitri spojeni tabulek.

Vysledek:

id_account ▾	account_number ▾	account_balance ▾	account_type ▾	currency_type ▾	iban ▾	id_person_account ▾	account_id ▾	person_id ▾
1	0800123456	1000.00	checking	CZK	CZ0800	1	1	1
2	0800987654	5000.00	savings	EUR	CZ0800	2	2	2
3	0100123456	2000.00	checking	USD	CZ0100	3	3	3
4	0100987654	3000.00	savings	CZK	CZ0100	4	4	4
5	0300123456	1500.00	checking	EUR	CZ0300	5	5	5
6	0300987654	2500.00	savings	USD	CZ0300	6	6	6
7	2700123456	4000.00	checking	CZK	CZ2700	7	7	7
8	2700987654	6000.00	savings	EUR	CZ2700	8	8	8
9	5500123456	7000.00	checking	USD	CZ5500	9	9	9
10	5500987654	8000.00	savings	CZK	CZ5500	10	10	10
11	5500987654	8000.00	savings	CZK	CZ5500	11	10	11
12	1000765432	1000.00	checking	CZK	CZ1000	12	11	15
13	1000987654	5000.00	savings	EUR	CZ1000	13	12	16
14	2100123456	2000.00	checking	USD	CZ2100	14	13	17
15	2100987654	3000.00	savings	CZK	CZ2100	15	14	18
16	2200123456	1500.00	checking	EUR	PL2200	16	15	19
17	2200987654	2500.00	savings	USD	PL2200	17	16	20
18	2800123456	4000.00	checking	CZK	CZ2800	18	17	21
19	2800987654	6000.00	savings	EUR	CZ2800	19	18	22
20	8100123456	7000.00	checking	USD	CZ8100	20	19	23
21	8100987654	8000.00	savings	CZK	CZ8100	21	20	24

id	person_id	id_person	birth_certificate_number	birth_date	full_name	email	phone_number	country	city
1	1	1	1234567890	1990-01-01	John Doe	john.doe@example.com	+1234567890	USA	New York
2	2	2	2345678901	1991-02-02	Alice Smith	alice.smith@example.com	+2345678901	UK	London
3	3	3	3456789012	1992-03-03	Michael Johnson	michael.johnson@example.com	+3456789012	Canada	Toronto
4	4	4	4567890123	1993-04-04	Emily Brown	emily.brown@example.com	+4567890123	Australia	Sydney
5	5	5	5678901234	1994-05-05	Daniel Wilson	daniel.wilson@example.com	+5678901234	Germany	Berlin
6	6	6	6789012345	1995-06-06	Sophia Martinez	sophia.martinez@example.com	+6789012345	France	Paris
7	7	7	7890123456	1996-07-07	Matthew Taylor	matthew.taylor@example.com	+7890123456	Italy	Rome
8	8	8	8901234567	1997-08-08	Olivia Garcia	olivia.garcia@example.com	+8901234567	Spain	Madrid
9	9	9	9012345678	1998-09-09	Liam Rodriguez	liam.rodriguez@example.com	+9012345678	Japan	Tokyo
10	10	10	0123456789	1999-10-10	Emma Lopez	emma.lopez@example.com	+0123456789	Brazil	São Paulo
10	11	11	19876543210	2000-11-11	Noah Perez	noah.perez@example.com	+9876543210	India	Mumbai
11	15	15	11111111111	1990-01-01	Emma Watson	emma.watson@example.com	+1111111111	USA	Los Angeles
12	16	16	22222222222	1991-02-02	Jack Johnson	jack.johnson@example.com	+2222222222	Canada	Vancouver
13	17	17	33333333333	1992-03-03	Sophie Turner	sophie.turner@example.com	+3333333333	UK	Manchester
14	18	18	44444444444	1993-04-04	Ryan Reynolds	ryan.reynolds@example.com	+4444444444	Australia	Melbourne
15	19	19	55555555555	1994-05-05	Natalie Portman	natalie.portman@example.com	+5555555555	France	Nice
16	20	20	66666666666	1995-06-06	Leonardo DiCaprio	leonardo.dicaprio@example.com	+6666666666	Italy	Florence
17	21	21	77777777777	1996-07-07	Jennifer Lawrence	jennifer.lawrence@example.com	+7777777777	Spain	Barcelona
18	22	22	88888888888	1997-08-08	Chris Hemsworth	chris.hemsworth@example.com	+8888888888	Japan	Osaka
19	23	23	99999999999	1998-09-09	Scarlett Johansson	scarlett.johansson@example.com	+9999999999	Brazil	Rio de Janeiro
20	24	24	00000000000	1999-10-10	Brad Pitt	brad.pitt@example.com	+0000000000	India	New Delhi

city	postcode	street
New York	10001	Main Street
London	SW1A	King Street
Toronto	M5H	Queen Street
Sydney	2000	George Street
Berlin	10117	Friedrichstraße
Paris	75001	Rue de Rivoli
Rome	00184	Via del Corso
Madrid	28001	Calle de Alcalá
Tokyo	1000001	Chiyoda
São Paulo	01000	Avenida Paulista
Mumbai	400001	Nariman Point
Los Angeles	90001	Hollywood Blvd
Vancouver	V5K	Granville Street
Manchester	M1	Deansgate
Melbourne	3000	Collins Street
Nice	06000	Promenade des Anglais
Florence	50123	Ponte Vecchio
Barcelona	08001	Las Ramblas
Osaka	5500002	Umeda
Rio de Janeiro	20040	Copacabana Beach
New Delhi	110001	Connaught Place

3 (agregaci a podmínku na hodnotu agregační funkce):

SELECT b.bank_name, COUNT(a.id_account) AS num_accounts

FROM Bank b

LEFT JOIN Account a ON b.iban = a.iban

GROUP BY b.id_bank

HAVING COUNT(a.id_account) > 0;

Cinnost : spočítává počet účtů spojených s každou bankou a zobrazuje pouze banky, které mají více než jeden účet.

Vysledek:

	bank_name	count_accounts
1	Air Bank	2
2	Commercial Bank	2
3	Czech Savings Bank	2
4	Českomoravská hypoteční banka	2
5	Fio banka	2
6	Raiffeisenbank	2
7	mBank	2
8	Expobank	2
9	UniCredit Bank	2
10	CSOB	2

4 (množinové operace+vnořený SELECT):

SELECT *

FROM Account

WHERE iban IN (

SELECT iban

FROM Bank

WHERE bank_name = 'Air Bank'

UNION

SELECT iban

FROM Bank

WHERE bank_name = 'UniCredit Bank'

) AND currency_type = 'CZK';

Cinnost: vybere účty, které mají IBAN buď u banky "Air Bank" nebo "UniCredit Bank" a měnu typu "CZK".

Vysledek:

id_account	account_number	account_balance	account_type	currency_type	iban
7	2700123456	4000.00	checking	CZK	CZ2700
11	1000765432	1000.00	checking	CZK	CZ1000

5 (řazení a stránkování):

```
SELECT birth_certificate_number, full_name, email
```

```
FROM Person
```

```
ORDER BY full_name
```

```
OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;
```

Cinnost: Vypiše jen 10 prvních lidí. Sloupce omezi na birth_certificate_number,full_name,email. Záznamy seřadí podle jmen Osob.

Vysledek:

	birth_certificate_number	full_name	email
1	2345678901	Alice Smith	alice.smith@example.com
2	0000000000	Brad Pitt	brad.pitt@example.com
3	8888888888	Chris Hemsworth	chris.hemsworth@example.com
4	5678901234	Daniel Wilson	daniel.wilson@example.com
5	4567890123	Emily Brown	emily.brown@example.com
6	0123456789	Emma Lopez	emma.lopez@example.com
7	1111111111	Emma Watson	emma.watson@example.com
8	2222222222	Jack Johnson	jack.johnson@example.com
9	7777777777	Jennifer Lawrence	jennifer.lawrence@example.com
10	1234567890	John Doe	john.doe@example.com

Code for inseting 32k fields:

```
DO $$
```

```
DECLARE
```

```
    i INTEGER := 21; -- Starting value for user ID
```

```
    my_string VARCHAR;
```

```
BEGIN
```

```
    WHILE i <= 32000 LOOP
```

```
        my_string := CAST(i AS VARCHAR);
```

```
    INSERT INTO "bank" (
```

```
        bank_name,
```

```
        main_office,
```

```
        iban
```

```
    )
```

```
    VALUES (
```

```
        'bank_name' my_string,
```

```
        'main_office' my_string,
```

```
'iban' my_string,  
);  
  
i := i + 1;  
END LOOP;  
END;  
$$;
```