

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

**ФГБОУ ВО «ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

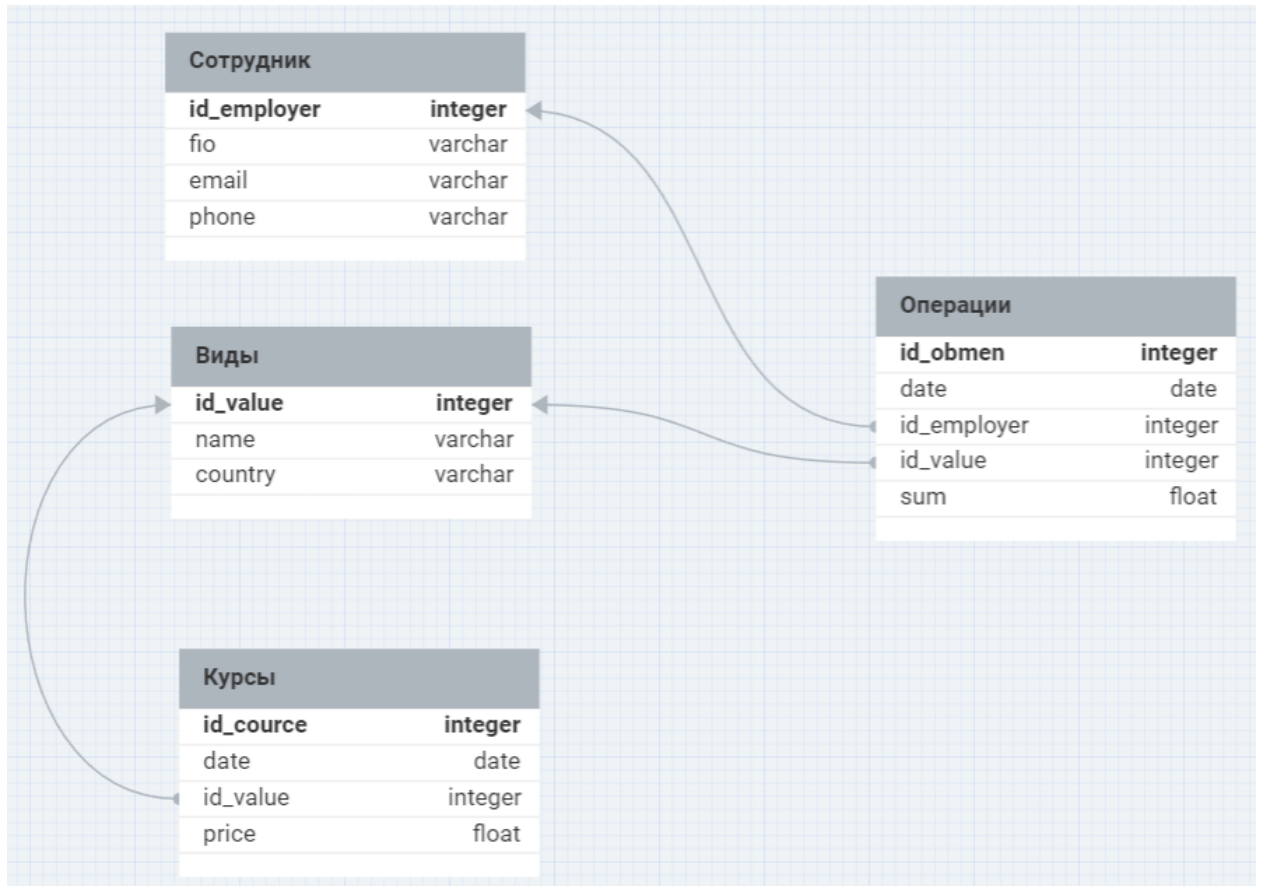
БЕЗОПАСНОСТЬ СИСТЕМ БАЗ ДАННЫХ

ПРАКТИЧЕСКАЯ РАБОТА №2

**ВЫПОЛНИЛ: АИСТОВА ЕЛИЗАВЕТА
ГРУППА: УБ-02**

**ВОРОНЕЖ
2023**

Логическая схема базы пункта обмена.



Таблицы Виды и Курсы связаны связью один ко многим по полю **id_value**:

- Каждая Курс может быть связана с несколькими валютами.
- Каждый Валюта может быть связан только с одним Курсом.

Таблицы Сотрудник и Операции связаны связью один ко многим по полю **id_employer**.

- Каждый Сотрудник может быть связан с несколькими Операциями.
- Каждая Операция связана только с одним Сотрудником.

Таблицы Виды и Операции связаны связью один ко многим по полю **id_value**.

- Каждый Вид может быть связана с несколькими Операциями.
- Каждая Операция связана только с одним Видом.

Для создания новой базы данных используется оператор CREATE DATABASE. Для того, чтобы подключиться к ней, используем команду \connect.

Для создания таблиц базы данных используется оператор CREATE TABLE. Создадим таблицы: Employer, Vid, Source и Operation.

```
postgres=# CREATE DATABASE obmen;
CREATE DATABASE
postgres=# \c обмен;
Вы подключены к базе данных "обмен" как пользователь "postgres".
```

```
обмен=# CREATE TABLE employer(
обмен(# id_employer SERIAL PRIMARY KEY,
обмен(# fio varchar(60) NULL,
обмен(# email varchar(15) NULL,
обмен(# phone varchar(15) NULL
обмен(# );
CREATE TABLE
обмен=# CREATE TABLE vid(
обмен(# id_value SERIAL PRIMARY KEY,
обмен(# name varchar(20) NULL,
обмен(# country varchar(20) NULL
обмен(# );
CREATE TABLE
обмен=# CREATE TABLE source(
обмен(# id_source serial PRIMARY KEY,
обмен(# date DATE NOT NULL,
обмен(# id_value int REFERENCES vid(id_value),
обмен(# price float NULL
обмен(# );
CREATE TABLE
```

```
обмен=# CREATE TABLE operation(
обмен(# id_operation SERIAL PRIMARY KEY,
обмен(# date DATE NOT NULL,
обмен(# id_employer int REFERENCES employer(id_employer),
обмен(# id_value int REFERENCES vid(id_value),
обмен(# sum float NULL
обмен(# );
CREATE TABLE
```

После выполнения данного запроса можно получить список созданных таблиц и связей с помощью команды \d:

```
obmen=# \d
```

Список отношений			
Схема	Имя	Тип	Владелец
public	course	таблица	postgres
public	course_id_course_seq	последовательность	postgres
public	employer	таблица	postgres
public	employer_id_employer_seq	последовательность	postgres
public	operation	таблица	postgres
public	operation_id_operation_seq	последовательность	postgres
public	vid	таблица	postgres
public	vid_id_value_seq	последовательность	postgres

(8 строк)

Список созданных индексов можно получить с помощью команды \di:

```
obmen=# \di
```

Список отношений				
Схема	Имя	Тип	Владелец	Таблица
public	course_pkey	индекс	postgres	course
public	employer_pkey	индекс	postgres	employer
public	operation_pkey	индекс	postgres	operation
public	vid_pkey	индекс	postgres	vid

(4 строки)

Для вставки данных в таблицы служит оператор INSERT INTO. Для того, чтобы вывести заполненную таблицу используем оператор SELECT * FROM название_таблицы.

```
obmen=# INSERT INTO employer (  
obmen(# fio, email, phone)  
obmen=# VALUES ('Аистова Елизавета', 'aisotva@mail.ru', '+79124780524');  
INSERT 0 1  
obmen=# INSERT INTO employer (  
obmen(# fio, email, phone)  
obmen=# VALUES ('Иванов Иван', 'ivanov@mail.ru', '+79217408542');  
INSERT 0 1  
obmen=# INSERT INTO employer (  
obmen(# fio, email, phone)  
obmen=# VALUES ('Петров Пётр', 'petrov@mail.ru', '+79564567890');  
INSERT 0 1  
obmen=# INSERT INTO employer (  
obmen(# fio, email, phone)  
obmen=# VALUES ('Романов Роман', 'roma@mail.ru', '+79159876543');  
INSERT 0 1  
obmen=# SELECT * FROM employer;  
 id_employer |      fio      |      email      |      phone  
-----+-----+-----+-----  
          1 | Аистова Елизавета | aisotva@mail.ru | +79124780524  
          2 | Иванов Иван      | ivanov@mail.ru  | +79217408542  
          3 | Петров Пётр     | petrov@mail.ru  | +79564567890  
          4 | Романов Роман    | roma@mail.ru    | +79159876543  
(4 строки)
```

```
obmen=# INSERT INTO vid (  
obmen(# name, country)  
obmen=# VALUES ('Доллар', 'США');  
INSERT 0 1  
obmen=# INSERT INTO vid (  
obmen(# name, country)  
obmen=# VALUES ('Рубль', 'Россия');  
INSERT 0 1  
obmen=# INSERT INTO vid (  
obmen(# name, country)  
obmen=# VALUES ('Фунт', 'Англия');  
INSERT 0 1  
obmen=# INSERT INTO vid (  
obmen(# name, country)  
obmen=# VALUES ('Йен', 'Япония');  
INSERT 0 1  
obmen=# SELECT * FROM vid;  
 id_value | name | country  
-----+-----+-----  
          1 | Доллар | США  
          2 | Рубль  | Россия  
          3 | Фунт   | Англия  
          4 | Йен    | Япония  
(4 строки)
```

```

obmen=# INSERT INTO course (
obmen(# date, id_value, price)
obmen-# VALUES ('2021-05-24', '2', '60.5');
INSERT 0 1
obmen=# INSERT INTO course (
obmen(# date, id_value, price)
obmen-# VALUES ('2020-06-15', '1', '5.75');
INSERT 0 1
obmen=# INSERT INTO course (
obmen(# date, id_value, price)
obmen-# VALUES ('2022-07-30', '3', '45.3');
INSERT 0 1
obmen=# INSERT INTO course (
obmen(# date, id_value, price)
obmen-# VALUES ('2019-08-5', '4', '28.9');
INSERT 0 1
obmen=# SELECT * FROM course;
 id_course |      date      | id_value | price
-----+-----+-----+-----
          1 | 2021-05-24 |         2 |  60.5
          2 | 2020-06-15 |         1 |   5.75
          3 | 2022-07-30 |         3 |  45.3
          4 | 2019-08-05 |         4 |  28.9
(4 строки)

```

```

obmen=# INSERT INTO operation (
obmen(# date, id_employer, id_value, sum)
obmen-# VALUES ('2021-06-20', '1', '2', '11456');
INSERT 0 1
obmen=# INSERT INTO operation (
obmen(# date, id_employer, id_value, sum)
obmen-# VALUES ('2022-08-1', '3', '3', '5555');
INSERT 0 1
obmen=# INSERT INTO operation (
obmen(# date, id_employer, id_value, sum)
obmen-# VALUES ('2020-06-17', '2', '1', '10000');
INSERT 0 1
obmen=# INSERT INTO operation (
obmen(# date, id_employer, id_value, sum)
obmen-# VALUES ('2019-08-07', '4', '4', '100');
INSERT 0 1

```

```

obmen=# SELECT * FROM operation;
 id_operation |      date      | id_employer | id_value | sum
-----+-----+-----+-----+-----
          2 | 2021-06-20 |         1 |         2 | 11456
          3 | 2022-08-01 |         3 |         3 |  5555
          4 | 2020-06-17 |         2 |         1 | 10000
          5 | 2019-08-07 |         4 |         4 |   100
(4 строки)

```