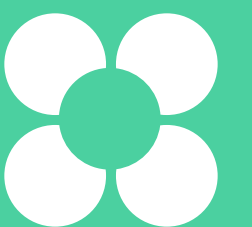


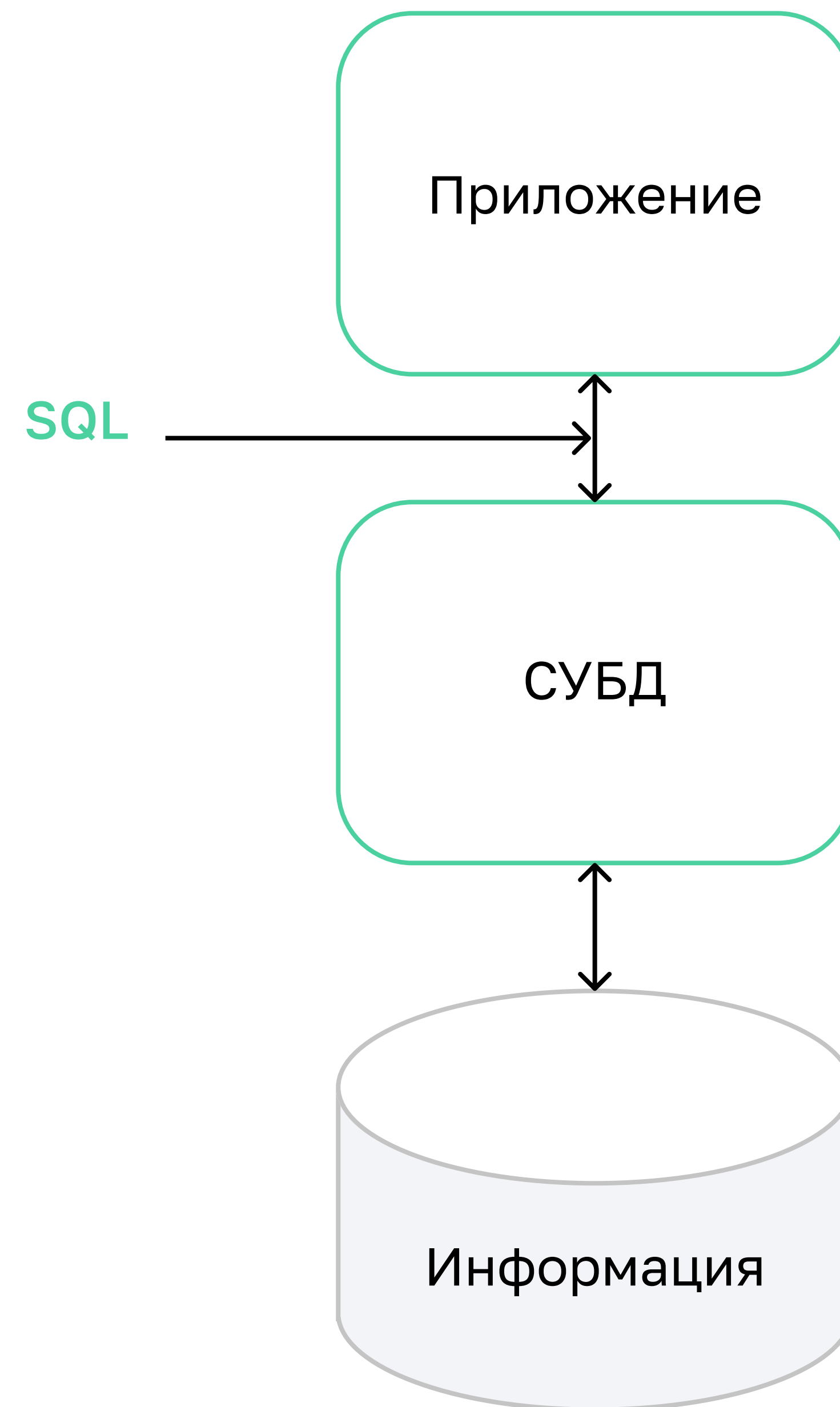
Работа с PostgreSQL, создание БД

Александр Иванов
Senior backend engineer в OneSoil



Structured Query Language

SQL — это язык структурированных запросов. Данный язык понимает СУБД, которая и производит операции с данными



Типы запросов в SQL

DDL (Data Definition Language)

→ CREATE, ALTER, DROP

DML (Data Manipulation Language)

→ SELECT, INSERT, UPDATE, DELETE

TCL (Transaction Control Language)

→ COMMIT, ROLLBACK, SAVEPOINT

DCL (Data Control Language)

→ GRANT, REVOKE, DENY

TokenAuthentication

Синтаксис:

```
CREATE TABLE [IF NOT EXISTS] <name> (  
    <col_name_1> <col_type_1> [CONSTRAINTS],  
    ...,  
    <col_name_N> <col_type_N> [CONSTRAINTS],  
    [CONSTRAINTS]  
);
```

Пример:

```
CREATE TABLE IF NOT EXISTS Student (  
    Id serial PRIMARY KEY,  
    Name varchar(40) NOT NULL,  
    GPA real CHECK(GPA > 0)  
);
```

DDL (Data Definition Language). Типы полей

Основные типы:

| Тип | Пояснение | Пример |
|-------------------|-------------------------------|-----------------------------|
| integer | Целые числа | id integer |
| serial | Целые числа с автоинкрементом | id serial |
| numeric | Десятичные числа | gpa numeric (3, 2) |
| character varying | Строки ограниченной длины | name varchar (40) |
| text | Строки произвольной длины | message text |
| date | Дата (без времени) | birthday date |
| timestamp | Дата + время | created_at timestamp |
| boolean | Булевы значений | active boolean |
| jsonb | JSON-поля | data jsonb |

DDL (Data Definition Language). CONSTRAINTS

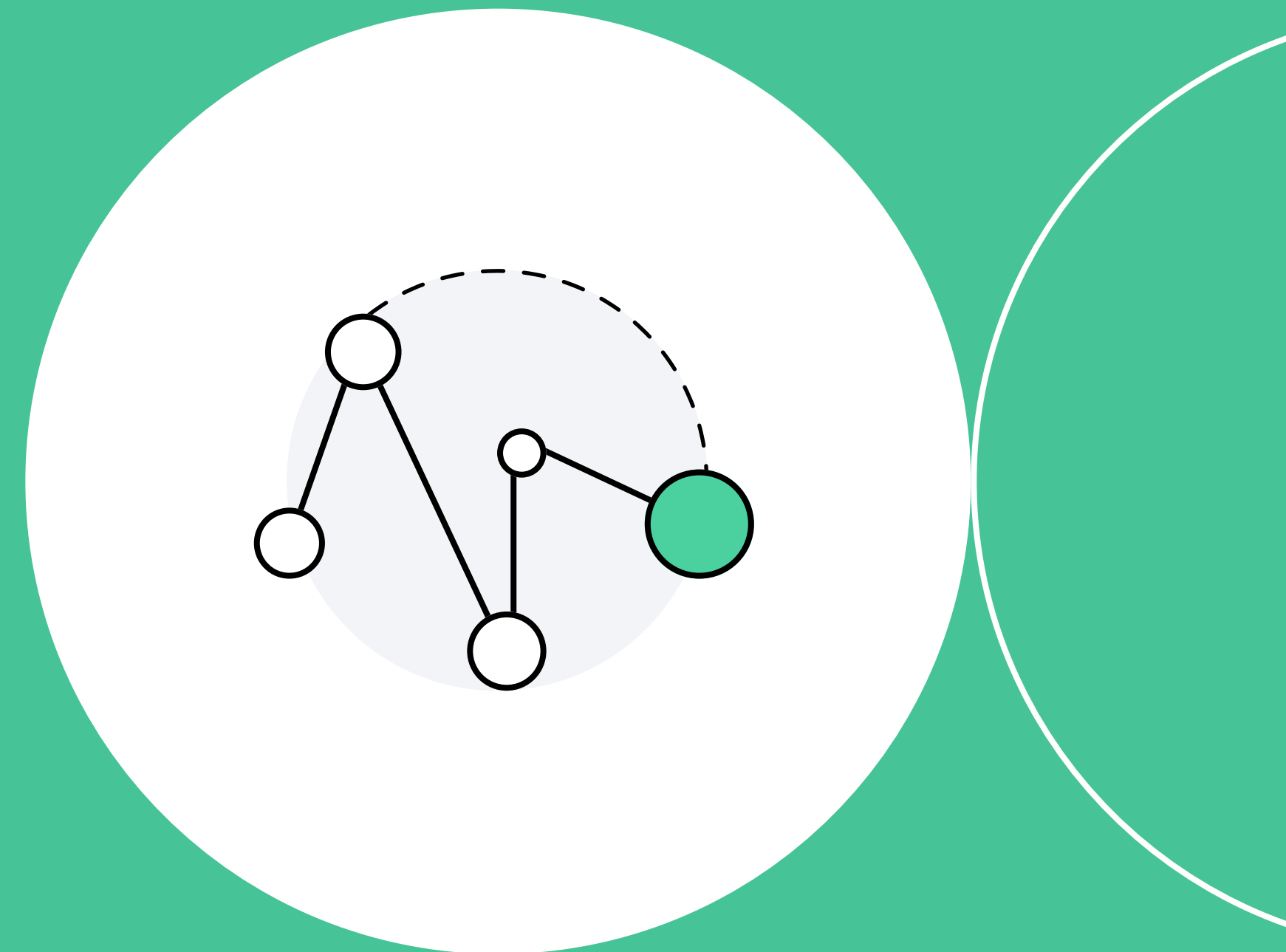
| Ограничение | Пояснение | Пример |
|-------------|---|--|
| primary key | Первичный ключ, обязывает поле быть уникальным и не пустым | id serial primary key |
| not null | Значение не может быть пустым (не может отсутствовать) | name varchar(40) not null |
| unique | Все значения в этом поле должны быть уникальным | tag varchar(80) unique |
| check | Добавить проверку значения на описанное условие | price numeric check(price > 0) |
| foreign key | Внешний ключ, обязывает значение соответствовать значению из другой таблицы | product_id integer references products(id) |

DDL (Data Definition Language). ALTER и DROP

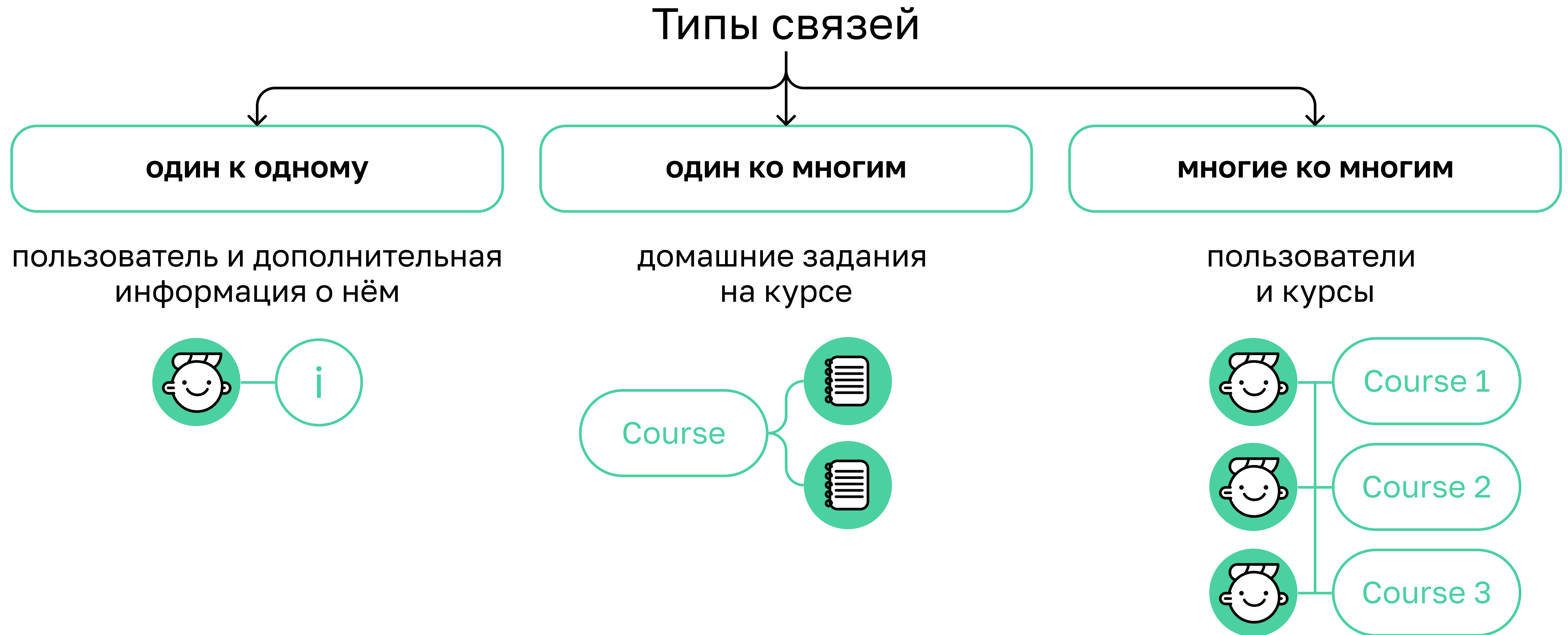
```
alter table <name> ...  
    # добавить атрибут  
    add column <col_name> <col_type> [constraints];  
  
    # переименовать таблицу  
    rename to <new_table_name>;  
  
    # переименовать атрибут  
    rename <col_name> to <new_col_name>;  
  
    # изменить тип атрибута  
    alter column <col_name> set data type <col_type>;  
  
    # добавить ограничение  
    add constraint <constraint_name> <constraint>;  
  
    # удалить ограничение  
    drop constraint <constraint_name>;  
  
    # удалить атрибут  
    drop column <col_name>;  
  
# удалить таблицу  
drop table <name>;
```

Связи между отношениями

Александр Иванов
Senior backend engineer в OneSoil

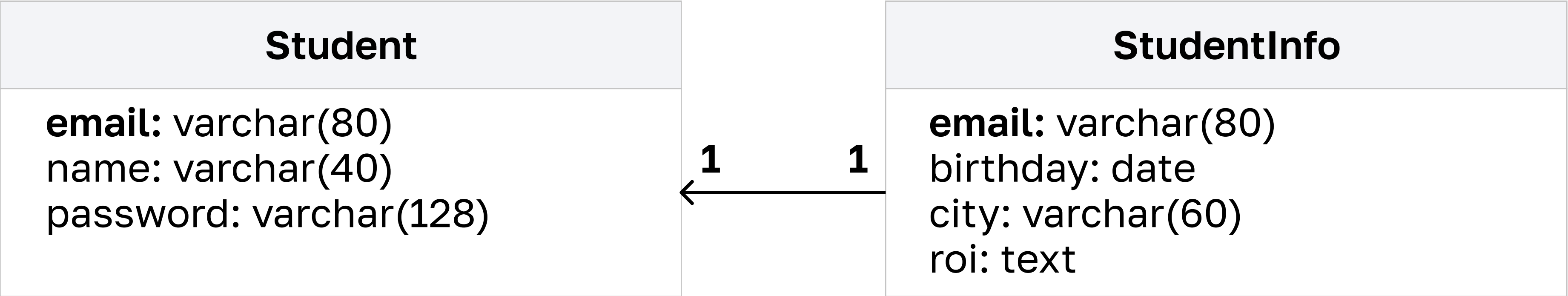


Связи между отношениями



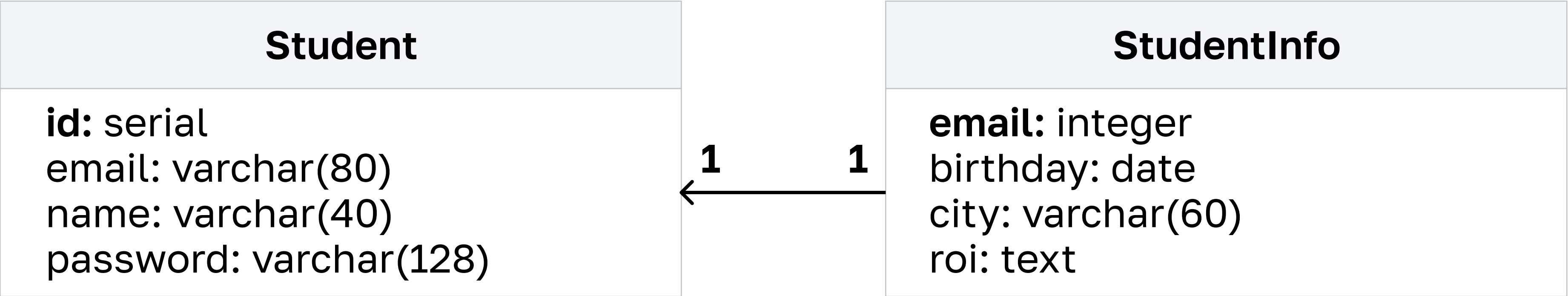
! Связи, как и первичный ключ, можно попросить контролировать СУБД. Для этого используется ограничение foreign key

Один к одному. Вариант 1



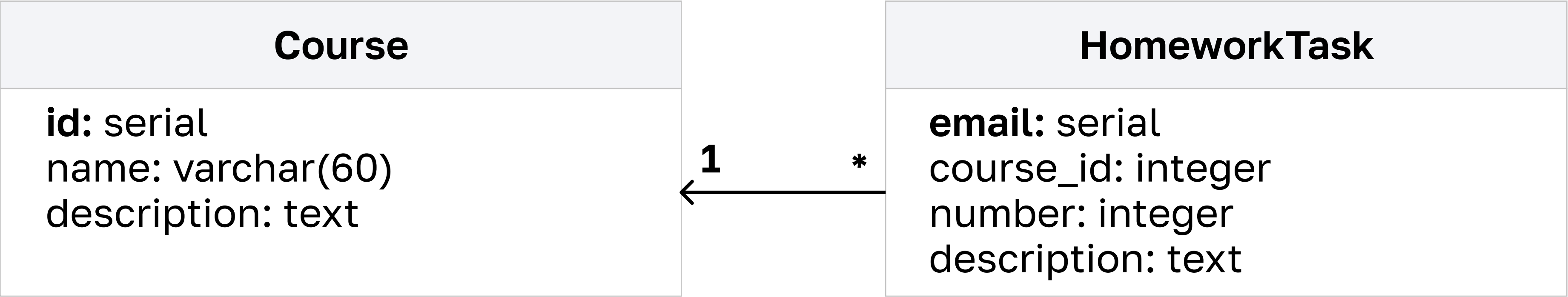
```
create table if not exists Student (  
    email varchar(80) primary key,  
    name varchar(40) not null,  
    password varchar(128) not null  
);  
  
create table if not exists StudentInfo (  
    email varchar(80) primary key references Student(email),  
    birthday date,  
    city varchar(60),  
    roi text  
);
```

Один к одному. Вариант 2



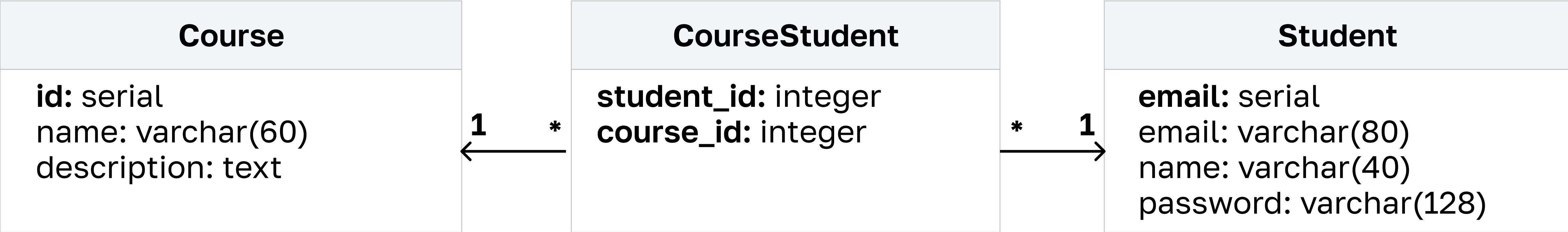
```
create table if not exists Student (  
    id serial primary key,  
    email varchar(80) unique not null,  
    name varchar(40) not null,  
    password varchar(128) not null  
);  
  
create table if not exists StudentInfo (  
    id integer primary key references Student(id),  
    birthday date,  
    city varchar(60),  
    roi text  
);
```

ОДИН КО МНОГИМ



```
create table if not exists Course (  
  id serial primary key,  
  name varchar(60) not null,  
  description text  
);  
  
create table if not exists HomeworkTask (  
  id serial primary key,  
  course_id integer not null references Course(id),  
  number integer not null,  
  description text not null  
);
```

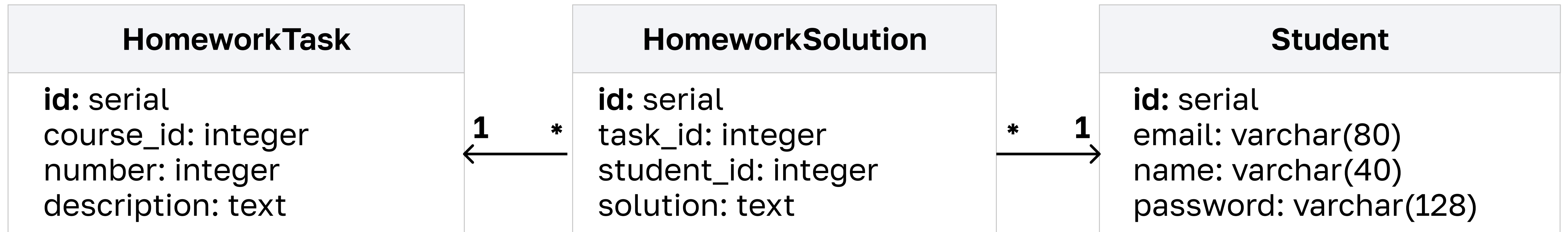
Многие ко многим. Вариант 1



```
create table if not exists CourseStudent (  
    course_id integer references Course(id),  
    student_id integer references Student(id),  
    constraint pk primary key (course_id, student_id)  
);
```

| student_id | course_id |
|------------|------------|
| 1 (Вова) | 1 (Python) |
| 1 (Вова) | 2 (Java) |
| 2 (Дима) | 1 (Python) |

Многие ко многим. Вариант 2



```
create table if not exists HomeworkSolution (  
    id serial primary key,  
    task_id integer not null references HomeworkTask(id),  
    student_id integer not null references Student(id),  
    solution text not null  
);
```

Итоговая схема

