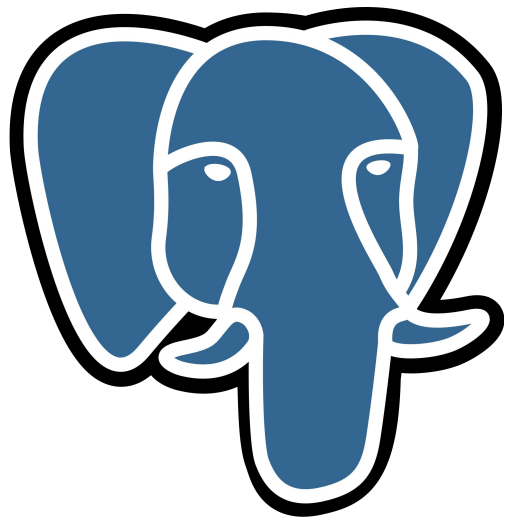


SQL

AI Education

Язык запросов SQL

- SQL — декларативный язык
- И у него есть диалекты!
- Работаем с PostgreSQL



PostgreSQL в контейнере

- PostgreSQL — это свободная объектно-реляционная система управления базами данных, основанная на языке SQL
- Подробнее про преимущества PostgreSQL написано [здесь](#)
- ```
$ docker run --name club -p 5432:5432 -e POSTGRES_USER=student -e POSTGRES_PASSWORD=password -e POSTGRES_DB=club -d aiedu/club:latest
```

# SQL

- SQL (Structured Query Language — Структурированный Язык Запросов) — декларативный язык программирования, который используется в качестве эффективного способа управления данными (их извлечения, поиска, модификации и удаления из хранилища)
- SQL содержит 4 основные группы операторов (DDL, DML, DCL, TCL)

# Операторы в SQL

- **DDL** (Data Definition Language ): CREATE, DROP, ALTER...
- **DML** (Data Manipulation Language): SELECT, INSERT, DELETE, UPDATE...
- **DCL** (Data Control Language): GRANT / REVOKE, LOCK / UNLOCK , SET LOCK MODE)
- **TCL** (Transaction Control Language): BEGIN TRANSACTION, COMMIT TRANSACTION...

# ACID

- Atomicity — Атомарность
- Consistency — Согласованность
- Isolation — Изолированность
- Durability — Стойкость

Подробнее про ACID можно прочитать [здесь](#).

# Основные понятия

- база данных
- схема
- сущность
- атрибут
- первичный ключ
- внешний ключ
- транзакция

# Типы данных

- числовые типы
- символьные типы
- типы для работы с датой и временем
- логические типы
- ...



# Структура БД

Будем работать с данными загородного клуба (члены клуба, бронирования, услуги).

**cd.members**

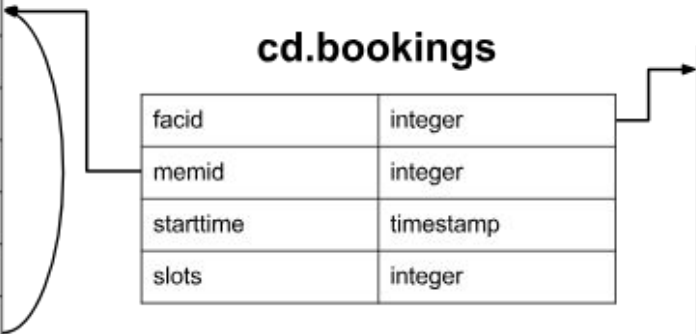
|               |                        |
|---------------|------------------------|
| memid         | integer                |
| surname       | character varying(200) |
| firstname     | character varying(200) |
| address       | character varying(300) |
| zipcode       | integer                |
| telephone     | character varying(20)  |
| recommendedby | integer                |
| joindate      | timestamp              |

**cd.bookings**

|           |           |
|-----------|-----------|
| facid     | integer   |
| memid     | integer   |
| starttime | timestamp |
| slots     | integer   |

**cd.facilities**

|                    |                        |
|--------------------|------------------------|
| facid              | integer                |
| name               | character varying(100) |
| membercost         | numeric                |
| guestcost          | numeric                |
| initialoutlay      | numeric                |
| monthlymaintenance | numeric                |



# Создание таблицы members

```
CREATE TABLE cd.members
(
 memid integer NOT NULL,
 surname character varying(200) NOT NULL,
 firstname character varying(200) NOT NULL,
 address character varying(300) NOT NULL,
 zipcode integer NOT NULL,
 telephone character varying(20) NOT NULL,
 recommendedby integer,
 joindate timestamp NOT NULL,
 CONSTRAINT members_pk PRIMARY KEY (memid),
 CONSTRAINT fk_members_recommendedby FOREIGN KEY (recommendedby)
 REFERENCES cd.members(memid) ON DELETE SET NULL
);
```

# Создание таблицы facilities

```
CREATE TABLE cd.facilities
(
 facid integer NOT NULL,
 name character varying(100) NOT NULL,
 membercost numeric NOT NULL,
 guestcost numeric NOT NULL,
 initialoutlay numeric NOT NULL,
 monthlymaintenance numeric NOT NULL,
 CONSTRAINT facilities_pk PRIMARY KEY (facid)
);
```

# Создание таблицы bookings

```
CREATE TABLE cd.members
(
 memid integer NOT NULL,
 surname character varying(200) NOT NULL,
 firstname character varying(200) NOT NULL,
 address character varying(300) NOT NULL,
 zipcode integer NOT NULL,
 telephone character varying(20) NOT NULL,
 recommendedby integer,
 joindate timestamp NOT NULL,
 CONSTRAINT members_pk PRIMARY KEY (memid),
 CONSTRAINT fk_members_recommendedby FOREIGN KEY (recommendedby)
 REFERENCES cd.members(memid) ON DELETE SET NULL
);
```

# Первый запрос

- Получить всю информацию из таблицы cd.facilities  
`SELECT *`  
`FROM cd.facilities`
- Вывести список всех услуг и их стоимость для участников клуба  
`SELECT facilities.name, facilities.membercost`  
`FROM cd.facilities`

## Фильтрация (добавляем условие в запрос)

- Вывести список всех платных услуг

```
SELECT *
```

```
FROM cd.facilities
```

```
WHERE membercost > 0
```

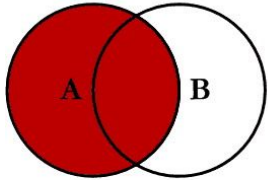
- Упражнение!** Вывести список платных услуг, где стоимость составляет менее 1/50 от ежемесячной стоимости членства в клубе  
(в ответе укажите следующие поля: `facid`, `name`, `membercost`, `monthlymaintenance`)

# Логические связи

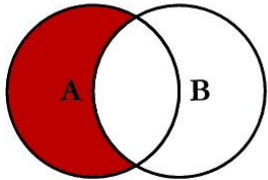
- AND – логическое И
- OR – логическое ИЛИ
- NOT – логическое НЕ

# JOIN

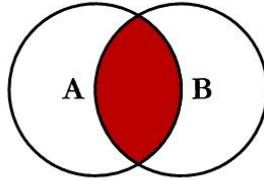
## SQL JOINS



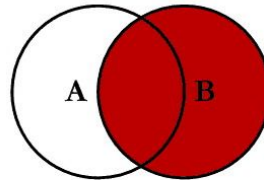
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



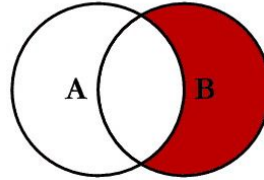
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



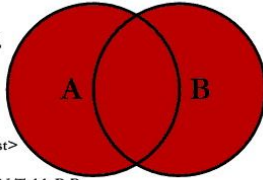
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



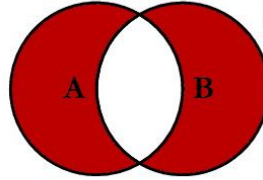
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```