Python

PostgreSQL

Хранение данных

Базы данных:

1. Реляционные - PostgreSQL, MSSQL, MySQL, SQLite…
2. Документоориентированные - MongoDB, CouchDB…
3. Ключ-значение - Redis, Berkeley DB…

….

SQL-базы данных

Подмножество реляционных БД.

Поддерживают стандарт SQL (последняя редакция SQL:2008)
SQL - декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных.

PostgreSQL

Кроссплатформенная объектно-реляционная СУБД с открытым исходным кодом.

Начало разработки - 1986 год, Калифорнийский университет в Беркли, Майкл Стоунбрейкер, **Post** In**gres.**

1995 год - переход на язык SQL и переименование в PostgreSQL.

Преимущества

- Поддержка сложных структур данных и пользовательских типов
  - сетевые адреса
  - многомерные массивы
  - геометрические данные
  - поддержка json
  - создание нового типа
- Размеры данных
  - размер БД - неограничен
  - размер таблицы - 32 ТБ
  - размер строки - 1.6 ТБ
  - размер поля - 1 ГБ
- Целостность данных - стандарт ANSI-SQL:2008.

Преимущества

- Индексы
  - частичные
  - функциональные
  - GIST/GIN (Generalized Search Tree/Generalized Inverted Index)
- CTE (Common Table Expressions) и рекурсия
- Материализованные представления
- Функции на четырех встроенных языках
- Языковые расширения

Основные типы

- Числовые
  - `smallint/integer/bigint` - целое 2/4/8 байт
  - `decimal/numeric` - вещественное число с указанной точностью
  - `real/double precision` - точность в пределах 6/15 десятичных цифр
  - `smallserial/serial/bigserial` - целое 2/4/8 байт с автоувеличением
- Символьные типы
  - `varchar(n)` - строка ограниченной переменной длины
  - `char(n)` - строка фиксированной длины, дополненная пробелами
  - `text` - строка неограниченной переменной длины
- Логический тип

Основные типы

- Типы даты/времени
  - `timestamp` - дата и время без часового пояса
  - `timestamptz` - дата и время с часовым поясом
  - `date` - дата (без времени суток)
  - `time` - время суток (без даты)
- Составные типы

```
CREATE TYPE inventory_item AS (
    name            text,
    supplier_id     integer,
    price           numeric
);
```

Синтаксис SQL

Создание таблиц:

```
create table example (
    id bigserial PRIMARY KEY,
    content text,
    author varchar(200),
    rating smallint,
    published bool NOT NULL DEFAULT false,
    created_at timestamptz NOT NULL DEFAULT now()
);
```

Синтаксис SQL

Вставка строки в таблицу:

```
insert into <название таблицы> (<название столбца>, <название столбца>, ...) values (<значение>, <значение>, ...)
```

```
insert into example (content, author, rating) values ('some text', 'Miles Taylor', 5)
```

Синтаксис SQL (выборка строк)

Выборка строк:

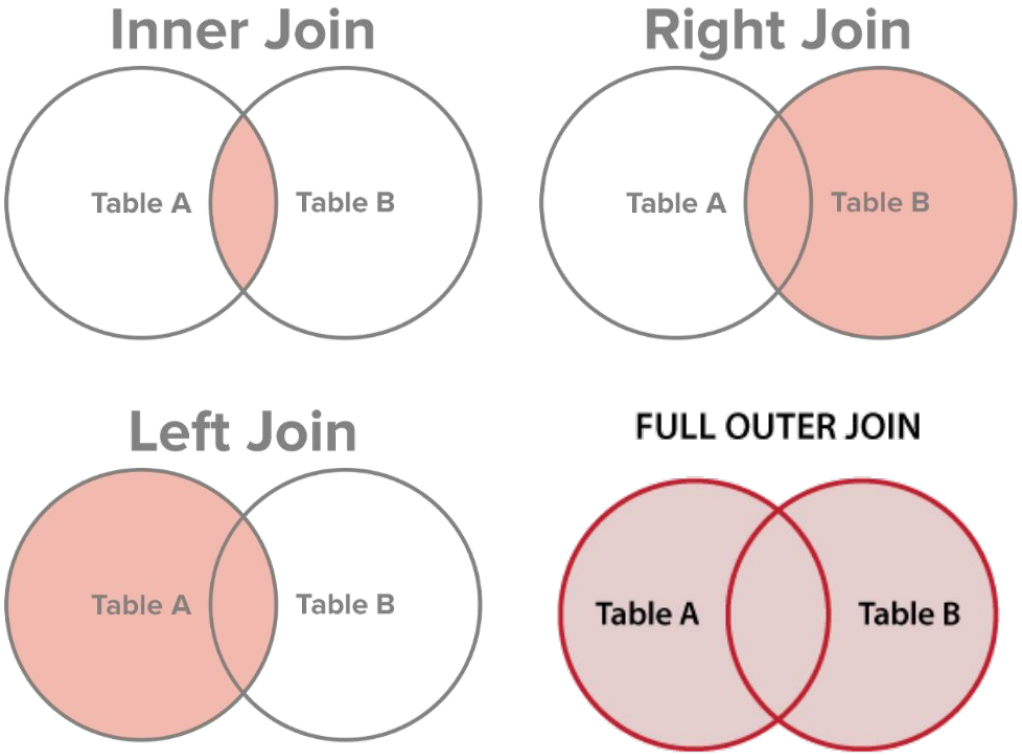
```
select * from <название таблицы>;
select * from <название таблицы> WHERE <условия>;

select * from users;
select username from users;
select * from users where username='foo';
```

Синтаксис SQL (выборка строк)

```
SELECT * FROM <название таблицы>
LEFT/RIGHT/INNER/FULL/CROSS JOIN <название другой таблицы> ON
<условия>
WHERE <условия>
GROUP BY <название столбца/столбцов>
HAVING <условия>
ORDER BY <название столбца/столбцов>
LIMIT count OFFSET offset
```

Понимание JOIN



Синтаксис SQL

Обновление строк:

```
update <название таблицы> set <название столбца>=<значение>,
<название столбца>=<значение>, ... where <условия>;

update users set username='foo_new' where user_id=1;
```

Синтаксис SQL

Удаление строк:

```
delete from <название таблицы> where <условия>;

delete from users where user_id = 1;
```

timestamp с/без таймзоны

- Часовые пояса - это сложно :)
- Время без таймзоны, только для пользовательских событий.
- Время с таймзоной хранится в UTC.
- Выдача результата зависит от таймзоны подключения.

```
select now(); # ???
set timezone = 'America/New_York';
select now(); # ???

set timezone = 'Europe/Moscow';
select now()::timestamp at time zone 'America/New_York'; # ???
```

Сравнение строк и операции

В зависимости от настройки БД, строковые столбцы создаются с определенной кодировкой LC_COLLATE.

Если LC_COLLATE='C', то возможна следующая ситуация:
SELECT lower(t1) FROM t; => ТЕСТ

Решения:

- указать в запросе: `SELECT lower(t1 COLLATE "C.UTF-8") FROM t;`
- при создании таблицы: `CREATE TABLE t (t1 text COLLATE "C.UTF-8")`

Индексы

Специальная структура, позволяющая ускорить поиск и сортировку по определенному полю.

```
create table t (a integer, b text, c boolean);

create [unique] index on t (a);
create index on t (a,b);
create index on t ((lower(b)));
```

Частичные:

```
create index on t (a) where c;
```

Внешние ключи

Внешний ключ (foreign key) - столбец или комбинация столбцов, значения которых соответствуют Первичному ключу в другой таблице.

```
CREATE TABLE cities (
    id serial NOT NULL PRIMARY KEY, name varchar(100)
)

CREATE TABLE streets (
    id serial NOT NULL PRIMARY KEY, name varchar(100),
    city_id integer NOT NULL REFERENCES cities(id)
)
```

Упражнение

Разработать таблицы для хранения информации о книгах в библиотеке (название, автор, год выпуска, количество страниц, количество экземпляров) и данных о выданных книгах (кому, когда выдана, когда возвращена)

Написать SQL запросы, позволяющие:

1. Добавить/удалить/изменить данные в таблице с книгами.
2. Посчитать количество выданных в данный момент книг.
3. Найти топ-3 книг, чаще всего выдаваемых на руки.