







DAY 1

План

-  Термины
-  Ключи и их виды
-  Нормализация данных и нормальные формы
-  Типы данных SQL
-  Управление таблицами с помощью SQL
-  Управление данными с помощью SQL

Термины

Что такое СУБД?

Система управления базами данных, СУБД — специальная программа-сервер позволяющая использовать и управлять базами данных. СУБД позволяет читать и записывать данные, искать по ним и выполнять сложные выборки.

Что такое БД?

База данных, БД — фактически создаваемые на диске файлы в которых хранится информация записанная с помощью СУБД

Важно!

Часто значение БД и СУБД путают, можно услышать фразу “БД PostgreSQL” когда имелось ввиду “СУБД PostgreSQL”

Что такое РСУБД?

Реляционная система управления базами данных, РСУБД —

СУБД которая хранит данные в виде таблиц и строк в этих таблицах.

Любая данные в РСУБД должны быть структурированы в соответствии с реляционной моделью.

Примеры РСУБД:

- PostgreSQL
- MySQL
- MS SQL
- Oracle

Что такое РМД?

Реляционная модель данных, РМД — математическая теория лежащая в основе **РСУБД**, она описывает как мы подходим к структурированию данных, чтобы ими было легко манипулировать.

Именно она требует от нас наличие таблиц, строк и связей между ними.

Таблицы, строки, столбцы

Таблица — совокупность строк и столбцов

Строка — запись в БД о конкретной сущности

Столбец — любой атрибут сущности хранящейся в строке

id	username	password	email	name	is_admin
1	mirgan2020	%2F%sdqqq	vasya12@mail.ru	Vasya	false
2	breisya	%2F%sdxxx	pomidor9@yahoo.com	Alex	true
3	anton	%2F%anton	anton@anton.ru	Anton	false

Ключи

🔑 **Первичный ключ, Primary Key, PK** — набор столбцов, позволяющих гарантированно отличить одну строку таблицы от другой. Если первичный ключ состоит из одного столбца, то он называется простым, если из нескольких столбцов — составным. Бывает естественным или суррогатным.

🔑 **Внешний ключ, Foreign Key, FK** — столбец таблицы, содержащий значение ключевого столбца другой таблицы. Нужен для корректной работы систем, контролирующей согласованность данных и связи данных из разных таблиц.

Нормализация данных

Что такое нормализация данных?

Нормализация данных — последовательный процесс преобразования структуры данных таким образом, чтобы она соответствовала реляционной модели.

РСУБД вполне способна работать и без нормализации, но теряются все преимущества реляционной модели.

Нормализация позволяет РСУБД:

- обнаружить и предотвратить рассогласование данных
- гарантировать отсутствие дублирующихся данных
- обеспечить транзакционность

Что такое НФ?

Нормальная форма, НФ — набор требований к структуре таблиц и связей между ними.

Существует 8 нормальных форм, каждая последующая ужесточает требования по сравнению с предыдущими.

Для работы с РСУБД нужно добиться чтобы структура таблиц соответствовала усиленной третьей нормальной форме, которую ещё называют **нормальной формой Бойса-Кодда, НФБК**.

Какие НФ мы будем использовать?

- Первая нормальная форма, 1НФ
- Вторая нормальная форма, 2НФ
- Третья нормальная форма, 3НФ
- Нормальная форма Бойса-Кодда, НФБК

Остальные четыре формы используются редко, в очень специфичных задачах.

Первая нормальная форма

- Не должно быть дублирующихся строк
- В каждой ячейке хранится атомарное значение (одно, не составное значение)
- В столбце хранятся данные только одного типа
- Отсутствуют массивы и списки в любом виде

Первая нормальная форма

Сотрудник	Контакт
Иванов И.И.	123-456-789, 987-654-321
Сергеев С.С.	Рабочий телефон 555-666-777, Домашний телефон 777-888-999
John Smith	123-456-780
John Smith	123-456-780

Не подходит под критерии первой нормальной формы:

- Есть дублирующиеся строки
- Присутствуют составные значения
- Присутствуют массивы

Первая нормальная форма

Сотрудник	Телефон	Тип телефона
Иванов И.И.	123-456-789	
Иванов И.И.	987-654-321	
Сергеев С.С.	555-666-777	Рабочий телефон
Сергеев С.С.	777-888-999	Домашний телефон
John Smith	123-456-780	

Правильный вариант таблицы в первой нормальной форме

Вторая нормальная форма

- Таблица должна находиться в 1НФ
- Таблица должна иметь первичный ключ, по которому можно гарантированно отличить каждую строку от любой другой
- Все неключевые столбцы должны зависеть от полного ключа

Вторая нормальная форма

ФИО	Должность	Подразделение	Описание подразделения
Иванов И.И.	Программист	Отдел разработки	Разработка и сопровождение приложений и сайтов
Сергеев С.С.	Бухгалтер	Бухгалтерия	Ведение бухгалтерского учета
John Smith	Продавец	Отдел реализации	Организация сбыта продукции

Подходит под критерии 1НФ, но не подходит под критерии второй нормальной формы:

- Нет простого или составного ключа

Вторая нормальная форма

 Таб. номер	ФИО	Должность	Подразделение	Описание подразделения
1	Иванов И.И.	Программист	Отдел разработки	Разработка и сопровождение приложений и сайтов
2	Сергеев С.С.	Бухгалтер	Бухгалтерия	Ведение бухгалтерского учета
3	John Smith	Продавец	Отдел реализации	Организация сбыта продукции

Правильный вариант таблицы во второй нормальной форме


Вторая нормальная форма


 Название проекта	 Участник	Должность	Срок проекта (мес.)
Внедрение приложения	Иванов И.И.	Программист	8
Внедрение приложения	Сергеев С.С.	Бухгалтер	8
Внедрение приложения	John Smith	Менеджер	8
Открытие нового магазина	Сергеев С.С.	Бухгалтер	12
Открытие нового магазина	John Smith	Менеджер	12



Подходит под критерии 1НФ, но не подходит под критерии второй нормальной формы:

- Часть неключевых столбцов зависит от неполного ключа

Вторая нормальная форма

 ID проекта	Название проекта	Срок
1	Внедрение приложения	8
2	Открытие нового магазина	12

 ID участника	ФИО	Должность
1	Иванов И.И.	Программист
2	Сергеев С.С.	Бухгалтер
3	John Smith	Менеджер

 ID проекта	 ID участника
1	1
1	2
1	3
2	2
2	3

Правильный вариант таблиц во второй нормальной форме

Третья нормальная форма

- Таблица должна находиться в 2НФ
- В таблице должны отсутствовать транзитивные зависимости, то есть зависимости одних неключевых столбцов от других


Третья нормальная форма


 Таб. номер	ФИО	Должность	Подразделение	Описание подразделения
1	Иванов И.И.	Программист	Отдел разработки	Разработка сайтов
2	Сергеев С.С.	Бухгалтер	Бухгалтерия	Ведение бухгалтерии
3	John Smith	Продавец	Отдел реализации	Организация сбыта
4	Деев Э.Ю.	Программист	Отдел разработки	Разработка сайтов

Подходит под критерии 2НФ, но не подходит под критерии 3НФ:

- Существуют такие неключевые столбцы, от которых зависят другие неключевые столбцы

Третья нормальная форма

 Таб. номер	ФИО	Должность	Подразделение
1	Иванов И.И.	Программист	1
2	Сергеев С.С.	Бухгалтер	2
3	John Smith	Продавец	3

 ID подразделения	Название	Описание
1	Отдел разработки	Разработка сайтов
2	Бухгалтерия	Ведение бухгалтерии
3	Отдел реализации	Организация сбыта

Правильный вариант таблиц в третьей нормальной форме

Нормальная форма Бойса-Кодда

- Таблица должна находиться в 3НФ
- В таблице должны отсутствовать зависимости ключевых столбцов от неключевых столбцов

Нормальная форма Бойса-Кодда

 ID проекта	 Направление	Куратор
1	Разработка	Иванов И.И.
1	Бухгалтерия	Сергеев С.С.
2	Разработка	Иванов И.И.
2	Бухгалтерия	Петров П.П.
2	Реализация	John Smith
3	Разработка	Андреев А.А.

Подходит под критерии 3НФ, но не подходит под критерии НФБК:

- Существуют такие ключевые столбцы, которые зависят от неключевых столбцов

Нормальная форма Бойса-Кодда

 ID куратора	ФИО	Направление
1	Иванов И.И.	Разработка
2	Сергеев С.С.	Бухгалтерия
3	Петров П.П.	Бухгалтерия
4	John Smith	Реализация
5	Андреев А.А.	Разработка

 ID куратора	 ID проекта
1	1
1	2
2	1
3	2
4	2
5	3

Правильный вариант таблиц в нормальной формой Бойса-Кодда

SQL

Что такое SQL?

Structured Query Language, SQL — декларативный язык программирования, предназначенный для описания, изменения и извлечения данных из СУБД. Чаще всего используется в реляционных СУБД.

Типы данных SQL: числа

- **serial** — целое с автоувеличением, от 1 до 2147483647;
- **smallint, int2** — целое, от -32768 до +32767;
- **integer, int, int4** — целое, от -2147483648 до +2147483647;
- **bigint, int8** — целое, от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807;
- **double precision, float8** — с плавающей точкой, от 1E-307 до 1E+308

Типы данных SQL: строки

- **character(n), char(n)** — представляет строку из фиксированного количества символов. С помощью параметра задается количество символов в строке;
- **character varying(n), varchar(n)** — представляет строку из фиксированного количества символов. С помощью параметра задается количество символов в строке;
- **text** — представляет текст произвольной длины.

Типы данных SQL: время и даты

- **timestamp** — хранит дату и время, с учётом часового пояса или без. Для дат самое нижнее значение 4713 год до н. э., самое верхнее значение 294276 год н. э.;
- **date** — представляет дату от 4713 год до н. э. до 5874897 года н.э;
- **time** — хранит время суток, без даты, с учётом часового пояса или без. Принимает значения от 00:00:00 до 24:00:00.

Типы данных SQL: другое

- **boolean** — булево значение, true или false
- **jsonb** — данные произвольного формата в формате JSON

DDL и DCL

DDL и DCL

Data Definition Language, DDL — подмножество языка SQL, предназначенное для создания, изменения и удаления баз данных и таблиц в них. К нему относятся все команды, начинающиеся с **CREATE**, **ALTER** и **DROP**.

Data Control Language, DCL — подмножество языка SQL, предназначенное для управления правами доступа к базам данным и таблицам в них. С помощью него можно разрешить и запрещать пользователям создавать, изменять и удалять базы данных или таблицы. К нему относятся все команды, начинающиеся на **GRANT** и **REVOKE**.

SQL: создание пользователя СУБД

```
CREATE USER test_user WITH PASSWORD 'qwerty';
```

Пользователь создаётся именно в СУБД!

SQL: создание базы данных

```
CREATE DATABASE test_db OWNER test_user;
```

Любая база должна быть создана, прежде чем к ней можно будет подключиться.

Пользователь, указанный как owner, будет иметь максимальные права доступа к этой базе данных.

SQL: права пользователя в БД

```
GRANT ALL PRIVILEGES ON DATABASE test_db TO test_user;
```

По умолчанию доступ к базе имеет только owner и супер-администратор СУБД. Всем остальным пользователям доступ нужно добавлять явным образом.

SQL: удаление базы данных

```
DROP DATABASE test_db;
```

Удалить базу данных может её владелец, супер-администратор СУБД или пользователь, которому явным образом даны на это права.

SQL: создание таблицы

```
CREATE TABLE groups (  
    id serial PRIMARY KEY,  
    name varchar(80) UNIQUE  
);
```

```
CREATE TABLE students (  
    id serial PRIMARY KEY,  
    group_id integer REFERENCES  
groups ON DELETE CASCADE,  
    name varchar(80),  
    age integer  
);
```

При создании таблицы нужно указать её название и все входящие в неё столбцы. Не забудьте указать первичный ключ!

DML

DML

Data Manipulation Language, DML — подмножество языка SQL, предназначенное для создания, изменения и удаления данных в таблицах внутри базы данных. К нему относятся все команды, начинающиеся с **SELECT**, **INSERT**, **UPDATE** и **DELETE**.

В целом SQL DML очень похожи среди разных баз данных, но иногда бывают серьезные отличия, поэтому всегда полезно сверяться с документацией по конкретной БД в случае проблем.

SQL: добавление строк в таблицу

```
INSERT INTO students (name, age) VALUES  
    ('Anna', 25),  
    ('Maria', 23),  
    ('Roman', 28);
```

Обратите внимание, что поле `id` существует в таблице, но указывать его значение не обязательно. Тип данных **serial** позволяет ему заполняться автоматически.

SQL: извлечение данных

```
SELECT * FROM students;
```

```
SELECT name, age FROM students;
```

```
SELECT * FROM students WHERE age < 18;
```

```
SELECT * FROM students ORDER BY age ASC;
```

```
SELECT AVG(age) AS 'Средний возраст' FROM students;
```

С помощью **SELECT** можно не просто вытаскивать данные, но и проводить фильтрацию, сортировать и проводить несложные агрегации.

SQL: изменение данных

```
UPDATE students SET age = 26 WHERE name = 'Anna' ;
```

С помощью **UPDATE** можно изменять любые данные в таблице.

Использование **WHERE** не обязательно, хотя и крайне рекомендуется. Без **WHERE** будут обновлены все существующие строки в таблице.

SQL: удаление данных

```
DELETE FROM students WHERE id = 1;
```

Чаще всего удаление происходит по id, но можно удалять и по любому другому полю:

```
DELETE FROM students WHERE name = 'Anna';
```

Возможно использование **DELETE** вообще без **WHERE**, в таком случае будут удалены вообще все данные в таблице.