УНИВЕРСИТЕТ СИНЕРГИЯ

Основы работы с базами данных

Преподаватель – Борисова Татьяна Андреевна

Сегодня на уроке:

- Обсудим, что такое базы данных, зачем они нужны;
- SQL и PostgreSQL;
- Особенности написания;
- Ответим на тестовые вопросы;
- Обговорим домашнее задание;

Введение

Хранить всю информацию в одной таблице неудобно: в таких данных непросто разобраться, к тому же они занимают больше места и долго обрабатываются.

Представьте, как выглядит документация компании, которая продает разные товары. Компания хранит данные о продажах и самих товарах, например артикул товара и его характеристики. Товары кто-то покупает, значит, понадобятся и данные о клиентах. А ещё информация о сотрудниках, которые эти товары продают.

Введение

Собирать все данные в одну таблицу — не лучшее решение. Намного проще и быстрее работать с данными, которые поделены на небольшие сегменты или таблицы.

Каждая таблица хранит определённую информацию, например данные о сотрудниках. Связать сотрудника с продажей можно по номеру — например, по идентификатору сотрудника. Теперь в таблице с заказами не нужно хранить имя и почту продавца — достаточно указать его или её идентификатор. А если какие-то данные о сотруднике изменились, их легко обновить: нужно поменять только одну строку в таблице.

Что такое БД?

УНИВЕРСИТЕТ СИНЕРГИЯ

Для упрощения работы с данными используют специальную структуру — базу данных. База данных похожа на большую библиотеку, в которой данные разделены на справочники. Данные в такой библиотеке не хранятся произвольно — они упорядочены и связаны между собой.

Наглядный пример

База данных - ?

УНИВЕРСИТЕТ СИНЕРГИЯ

Базы данных бывают разных видов в зависимости от способа хранения данных. Если данные в базе представлены в виде связанных таблиц, такую базу данных называют *реляционной* (англ. relation, «связь»). Реляционные базы данных особенно популярны благодаря своей простоте и надёжности. Вы будете работать именно с ними.

У таблиц реляционных баз есть несколько особенностей, которые отличают их от обычных таблиц. Читая документацию или общаясь с другими специалистами, вы обнаружите, что элементы таблицы реляционной базы данных носят особые названия. Столбцы называют *полями*, строки — *записями*, объекты на пересечении полей и записей — *ячейками*. Слова «столбец» и «поле» могут быть взаимозаменяемыми, но не забудьте об этой терминологии, когда встретите непривычное слово.

База данных - ?

Другая особенность — в таблице реляционной базы данных не должно быть дубликатов. Если все записи в таблице уникальны, вы будете уверены, что работаете с нужной строкой. Для этого к каждой записи добавляется уникальный признак — *первичный ключ*. Такой ключ не позволит создать абсолютно одинаковые записи.

Пример

№	Фамилия	Имя	Премия
1	Иванов	Виктор	1000
2	Петров	Илья	1200
3	Сидоров	Пётр	1100
4	Смирнов	Евгений	1400
5	Колчин	Игорь	1600
6	Митяев	Алексей	1100
7	Смирнов	Евгений	1400

Разъяснение примера

Первичный ключ может быть не только номером, но и набором любых символов. Главное, чтобы ключ не повторялся.

Обратите внимание на записи 4 и 7. В них одинаковые данные, но первичный ключ у них разный. Уникальный ключ сделает уникальной всю запись, а значит, записи не потеряются и путаницы не возникнет. Оба сотрудника-тёзки получат премию.

Таблица содержит в себе данные о пациентах в частном медицинском центре. Какие поля могут быть первичным ключом?

Номер медицинской карты	Дата первого обращения	ФИО пациента	Серия и номер паспорта	Пол	Код заболевания по МКБ	Номер удостоверения врача
28645	28.02.2019	Кривенков Алексей Николаевич	АВ 2073374	М	146	523785
32156	02.02.2018	Иванов Пётр Александрович	АВ 6084632	М	235	246372
25371	16.07.2017	Шумик Лариса Григорьевна	АВ 4756320	Ж	345	935476
89053	04.11.2017	Кузнецов Спартак Юрьевич	АВ 1064980	М	835	307737
65381	28.02.2019	Кирьянова Маргарита Сергеевна	АВ 8432156	Ж	146	284537
22235	30.01.2017	Соболь Вячеслав Викторович	АВ 4853460	М	635	495345

Для чего нужен первичный ключ?

УНИВЕРСИТЕТ СИНЕРГИЯ

Первичный ключ предоставляет доступ пользователя к базе данных.

Первичный ключ — это просто номер записи.

Первичный ключ — это система, которая позволяет управлять базой данных.

Первичный ключ сделает запись уникальной и позволит избежать полных дубликатов в таблице. Если структура в базе сложная, одного первичного ключа может не хватить. В таком случае сразу несколько полей будут первичными ключами в таблице. Такие поля не уникальны по отдельности. Уникальными будут сочетания значений, и благодаря этому в таблице не появятся дубликаты.

Пример


В каждом из полей есть повторы
Идентификатор пользователя
Идентификатор магазина
Фамилия пользователя
1 2 Васичев
1 3 Васичев
2 3 Климов
2 4 Климов
2 5 Климов
3 6 Крылова
3 2 Крылова
3 1 Крылова
4 5 Логачёва
4 3 Логачёва
Сочетание значений в полях уникально

Как вы думаете, отличается ли база данных от файловой системы?

1. Они ничем не отличаются. И в базе данных, и в файловой системе просто хранится информация.
2. В файловой системе информация в файлах не связана между собой. В базе данных, наоборот, информация упорядочена и связана.

Как вы думаете, можно ли писать запросы к базе данных на любом языке программирования?

1. Да, почему нет. Главное подключиться к базе, а искать информацию можно с помощью любых инструментов.
2. Нет, понадобится специальный язык для запросов к базе.

СУБД

База данных чем-то похожа на библиотеку: в базе много документов, которые упорядочены и структурированы. Однако таким объёмом информации нужно как-то управлять. В библиотеке за порядком следят библиотекари: они хорошо ориентируются в книгах и быстро найдут среди них нужную.

Базе данных тоже нужен хороший поиск: нет смысла собирать вместе много данных, если ими сложно управлять. База данных без системы управления будет простым набором структурированных документов.

Как вы думаете, отличается ли база данных от файловой системы?

1. Они ничем не отличаются. И в базе данных, и в файловой системе просто хранится информация.
2. В файловой системе информация в файлах не связана между собой. В базе данных, наоборот, информация упорядочена и связана.В базах данных роль библиотекаря выполняет специальная система управления базами данных, или СУБД.

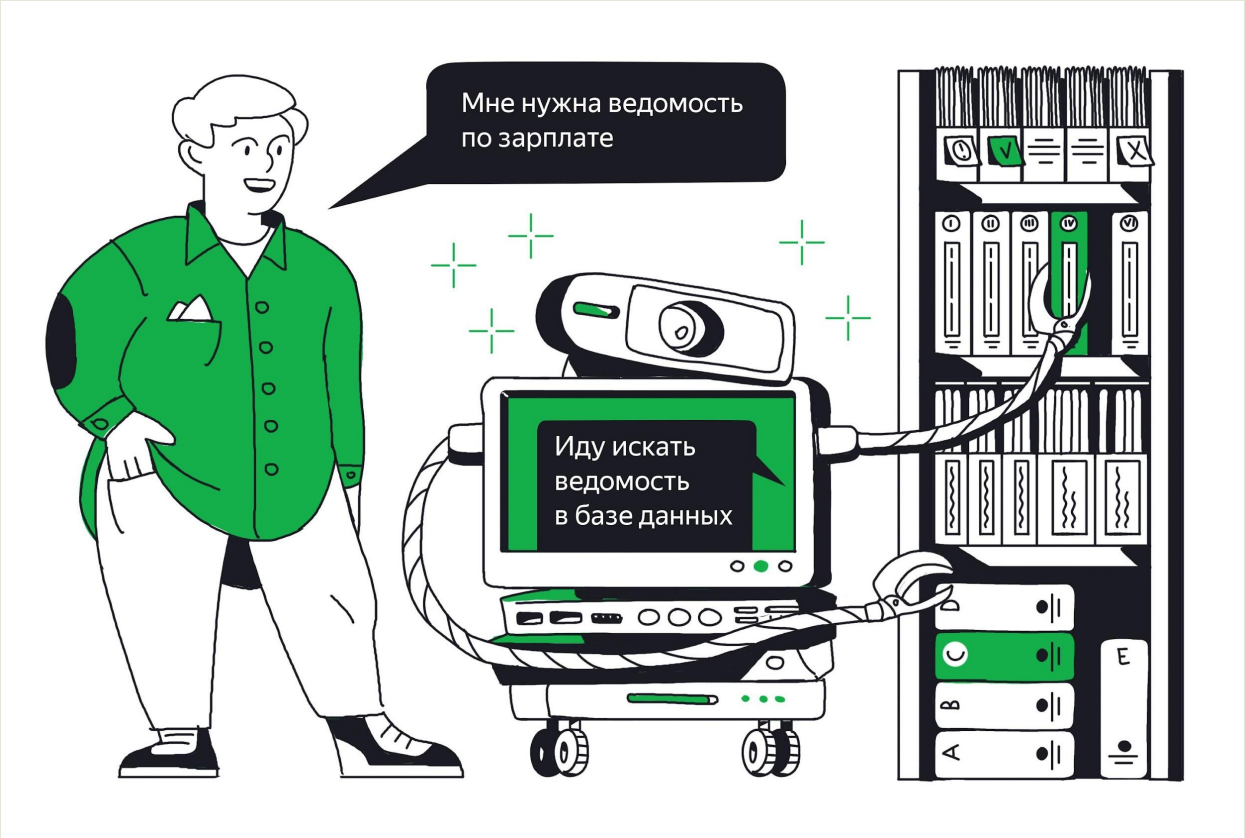
СУБД способна на многое, например, с её помощью можно:

создать базу или таблицу в базе;

внести новые данные или удалить устаревшие;

выгрузить нужную информацию, задав условие;

обеспечить безопасный доступ к данным.

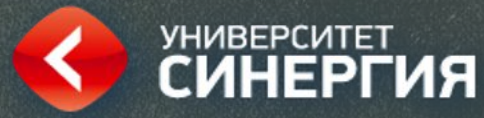

Мне нужна ведомость по зарплате
Иду искать ведомость в базе данных
A
B
C
D
E

Виды СУБД

УНИВЕРСИТЕТ СИНЕРГИЯ

- СУБД существует очень много. В этом курсе вы будете пользоваться PostgreSQL — это мощная СУБД с открытым исходным кодом, у которой почти нет ограничений по объёмам. С PostgreSQL можно работать на Windows, Linux и Mac.

Преимущества PostgreSQL



Перечислим несколько преимуществ:

- нет ограничений по размеру базы данных;
- большой выбор встроенных языков программирования;
- возможность создавать собственные типы данных.

Помимо всего, PostgreSQL — популярная СУБД. В компаниях, которым требуются специалисты по SQL, часто используют PostgreSQL в качестве основного инструмента.

Как вы думаете, можно ли писать запросы к базе данных на любом языке программирования?

- Да, почему нет. Главное подключиться к базе, а искать информацию можно с помощью любых инструментов.
- Нет, понадобится специальный язык для запросов к базе.

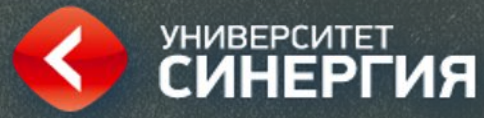
Язык запросов SQL

УНИВЕРСИТЕТ СИНЕРГИЯ

Для любого языка программирования важен правильный синтаксис. Одна опечатка и всё — код перестал работать. В жизни, конечно, не так. Вернёмся к аналогии с библиотекой. Если попросить библиотекаря принести книгу «Три Д’Артаньяна», тот смутится, но нужную книгу найдёт, если знает её правильное название. С базой данных такое не пройдёт: понадобятся точные команды.

Для работы с базой данных существует свой язык программирования. Реляционными базами данных управляют с помощью языка SQL, или Structured Query Language. По-английски это означает «язык структурированных запросов».

Синтаксис SQL



Синтаксис языка SQL отличается от синтаксиса многих других языков программирования. Важная особенность: SQL — декларативный язык. В SQL описывают не алгоритм действий, а данные, которые хотят получить. Как выполнить такую команду, за вас решит СУБД.

Язык SQL — это стандарт, который используют почти во всех реляционных базах данных. Но у этого языка, как у русского или английского, много диалектов. Диалект — это набор дополнительных команд, расширяющих стандартные возможности языка SQL.

У СУБД PostgreSQL тоже есть свои диалекты. Они немного отличаются от стандарта. Однако, разобравшись с PostgreSQL, вы сможете легче освоить любую другую реляционную СУБД — знаний синтаксиса вам для этого хватит.

Первый запрос

УНИВЕРСИТЕТ СИНЕРГИЯ

- Запросы в языке SQL пишут с помощью операторов. Операторы описывают действия над данными: «выбрать», «удалить» или «добавить». Чаще всего вы будете выгружать данные из базы, а для этого понадобятся два оператора: SELECT и FROM. Названия этих операторов по-английски означают «выбрать из».

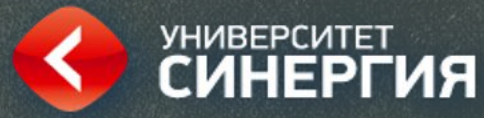
Повторение

Повторим:

- SELECT — оператор, с помощью которого описывают, что выгружать из базы.
- FROM — оператор, с помощью которого описывают, откуда выгружать данные.

Для других запросов существуют другие операторы,
но SELECT и FROM используют чаще всего, ведь без них выгрузить данные не получится.

Практика



Операторы SELECT и FROM — неразлучная пара, но запрос с одним оператором SELECT тоже сработает, но по-другому. Если указать в SELECT строку или число, оператор выведет их на экран.

Практика

УНИВЕРСИТЕТ СИНЕРГИЯ

- Уже можно писать первый запрос, но сначала нужно познакомиться с данными. Вы будете работать с базой данных онлайн-магазина, который специализируется на фильмах и музыке. В базе несколько таблиц с информацией об исполнителях, треках, фильмах, счетах и клиентах, но пока вам понадобятся три из них:

- genre — таблица с жанрами кино и музыки;
- media_type — таблица с используемыми форматами, например MPEG для видео и AAC для аудио;
- playlist — таблица с плейлистами.
- invoice — таблица с данными о заказах и выставленных счетах.

Практика

УНИВЕРСИТЕТ СИНЕРГИЯ

- Так выглядит запрос к базе данных:

```
SELECT *

FROM genre;
```

Часть таблицы

УНИВЕРСИТЕТ СИНЕРГИЯ

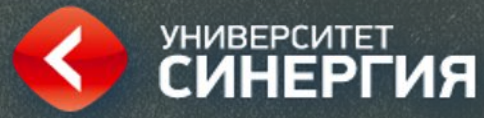
Genre_ID	Name
1	Rock
2	Jazz
3	Metal
4	Alternative & Punk
5	Rock And Roll
6	Blues
7	Latin
8	Reggae
9	Pop
10	Soundtrack
25	Opera

Разберем таблицу

УНИВЕРСИТЕТ СИНЕРГИЯ

- Теперь разберём синтаксис. С операторами вы уже знакомы — SELECT и FROM. Все, что указывают после оператора — *ключевые слова*. Ключевым словом может быть название поля, если нужно выгрузить конкретные значения. Даже знак * — ключевое слово. Его используют, чтобы выгрузить все данные из таблицы. Знак ; означает конец запроса.

Комментирование



Чтобы оставлять пометки в коде, в SQL можно использовать комментарии. Добавить в комментарий можно что угодно — они не влияют на выдачу. Однострочный комментарий начинают с двух дефисов:

-- Так

Закомментировать несколько строк можно так: поставить символы /* в начале комментария, а символы */ — в конце. В такой комментарий поместится целая частушка.

/*В век высоких технологий Без БД не обойтись, SQL - язык запросов Легче делает нам жизнь!*/ выглядит однострочный комментарий

Стиль запросов

УНИВЕРСИТЕТ СИНЕРГИЯ

- Во многих языках программирования есть правила хорошего стиля: использовать нижний регистр, разделять слова нижним подчёркиванием. Эти правила упрощают работу с чужим кодом и не только: чтобы разобраться в собственном коде через пару недель, вам тоже понадобятся комментарии.
- В SQL тоже есть правила стиля. Но SQL — язык, не чувствительный к регистру, значит, между операторами SELECT и select нет разницы. Казалось бы, зачем тогда соблюдать правила стиля. Но постепенно вы начнёте писать более сложные запросы, и в них не так легко будет разобраться.

Практика

УНИВЕРСИТЕТ СИНЕРГИЯ

- Запрос к базе данных может выглядеть так:

```
select sp.advertising_id,sp.install_date,sp.session_num,sp.payer,sp.last_active as last_date, DATE_DIFF(sp.last_active, install_date, day) as max_play from players as sp where sp.date = '2021-02-28' and sp.install_date between '2021-02-01' and '2021-02-26';
```

- Этот запрос выдаст то что нужно, но даже квалифицированный специалист с трудом прочитает такой код.

Практика

Так будет выглядеть этот запрос, написанный по правилам:

```
SELECT sp.advertising_id, sp.install_date, sp.session_num, sp.payer, sp.last_active AS
last_date,

DATE_DIFF(sp.last_active, install_date, DAY)

AS max_play

FROM players

AS sp

WHERE sp.date = '2021-02-28’

AND sp.install_date BETWEEN '2021-02-01' AND '2021-02-26';
```

Практика

Действительно, стало лучше. Разберём, что же изменилось.

Первое правило — писать операторы в верхнем регистре, чтобы визуально отделить их от ключевых слов. Поэтому в прошлом уроке данные из таблицы genre выгружали так:

```
SELECT *

FROM genre;
```

Вот так писать запросы не стоит: операторы недостаточно отделены от ключевых слов.

```
Select *

From genre;
```

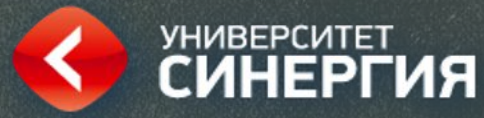
Практика

Когда столбцов и таблиц будет много, написать правильный запрос будет сложнее.

Второе правило — переносить новое ключевое слово на другую строку. Так легче понять, какие столбцы будут выгружены.

```
SELECT last_name,
        first_name,
        title
FROM staff;
```

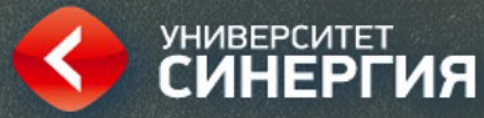
Практика



Так выглядят основные правила для запросов. Но в разных компаниях и организациях могут быть свои рекомендации. Их лучше соблюдать, чтобы не вносить хаос в общую работу.

Оформить запрос по всем правилам можно вручную или с помощью SQL-форматера, например SQLFormat. SQL-запрос вводят в специальное окно и на выходе получают его оформленную версию.

Подводим итоги



- какие особенности есть у языка SQL и СУБД PostgreSQL,
- из каких элементов состоят SQL-запросы,
- как выгружать все данные из таблиц,
- какие запросы соответствуют хорошему стилю.

Домашнее задание

УНИВЕРСИТЕТ СИНЕРГИЯ

Проработать тест по пройденной теме и прочитать дополнительный материал

СПАСИБО ЗА ВНИМАНИЕ!