Школа программирования × ВЫСШАЯ ШКОЛА ЭКОНОМИКИ

Системы управления базами данных (СУБД)

Системы управления базами данных (СУБД)

Системы управления базами данных (СУБД) представляют собой специализированные программные продукты, созданные для эффективного управления и обработки данных в базах данных. СУБД предоставляют удобный и структурированный способ хранения, организации, обновления и извлечения информации, обеспечивая при этом целостность данных и безопасность их использования.

Одной из ключевых особенностей СУБД является использование языков запросов, таких как SQL (Structured Query Language), который обеспечивает стандартизированный и удобный интерфейс для взаимодействия с базой данных.

СУБД могут быть различными по своей структуре и функциональности, включая реляционные, иерархические, объектно-ориентированные и другие типы систем.

Разработчики и администраторы баз данных используют СУБД для создания, изменения и управления структурой данных, а также для выполнения запросов, необходимых для получения нужной информации. СУБД играют ключевую роль в обеспечении эффективности и надежности работы с данными, что делает их неотъемлемой частью современных информационных технологий и бизнес-процессов.

Определения

База данных (БД) — это организованная и структурированная коллекция данных, предназначенная для эффективного хранения, управления и обеспечения доступа к информации. База данных характеризуется определенной моделью данных, которая определяет формат, типы данных и отношения между ними. Ключевые характеристики базы данных включают целостность данных, консистентность, доступность и безопасность.

Система управления базами данных (СУБД) обеспечивает программное обеспечение для создания, обновления и запросов данных в базе данных в соответствии с аданными правилами и структурами.

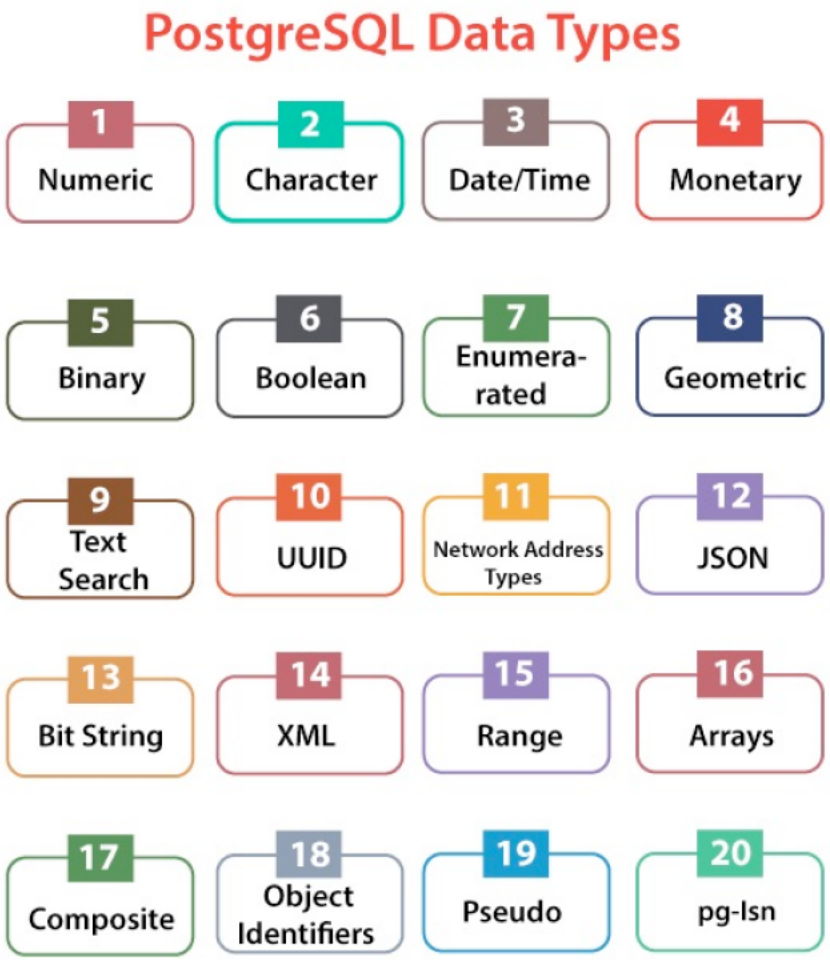
Определения

SQL (Structured Query Language) представляет собой специализированный язык программирования, который используется для управления и взаимодействия с реляционными базами данных. SQL состоит из различных команд и операторов, которые позволяют программистам и администраторам баз данных выполнять разнообразные задачи, такие как создание и изменение структуры таблиц, вставка, обновление и удаление данных, а также выполнение запросов для извлечения информации.

Основные категории команд SQL включают в себя DDL (Data Definition Language) для определения структуры данных, DML (Data Manipulation Language) для управления данными, и DQL (Data Query Language) для извлечения информации из базы данных.

Реляционные базы данных

Типы данных в SQL



Запросы

DDL (Data Definition Language)

Язык определения данных. DDL включает команды, оторые используются для определения или изменения структуры базы данных и её бъектов, таких как таблицы, индексы, схемы, и т.д. Ключевые команды DDL включают:

- CREATE : используется для создания новых объектов в базе данных, например, таблиц или индексов.
- ALTER : позволяет изменять структуру существующих объектов.
- DROP : удаляет объекты из базы данных.
- TRUNCATE : быстро удаляет все строки из таблицы, освобождая место, занимаемое этими строками.

Запросы

DML (Data Manipulation Language) и DQL (Data Query Language)

Язык манипуляции данными. DML включает команды, которые используются для вставки, обновления, удаления и выборки данных из баз данных. Эти команды взаимодействуют с данными в таблицах и включают:

- SELECT : извлекает данные из базы данных.
- INSERT : вставляет новые данные в таблицу.
- UPDATE : изменяет существующие данные в таблице.
- DELETE : удаляет данные из таблицы.

ПО

PgAdmin

https://www.pgadmin.org/download/

Установка postgres на Линукс:

https://www.digitalocean.com/community/tutorials/how-toinstall-postgresql-on-ubuntu-20-04-quickstart

DQL

Оператор SELECT

Это основной оператор для получения данных из базы. Вводить данные в запрос можно как строчными, так и заглавными буквами. Пробелы и табуляция не повлиляют на запрос. Язык SQL - функциональный, поэтому там нельзя просто так взять и создать объект. Зато можно «заселектить» его.

```
SELECT 1
```

В результате выполнения этого запроса мы получили таблицу-выборку с единственным столбцом с автоматически сгенерированным названием и единственной строкой, одержащей в этом столбце 1. Давайте назовем столбец как-нибудь понятно (зададим alias для столбца):

```
SELECT 25 AS number
```

Чтобы получить данные из некоторой таблицы, нужно ее указать в операторе FROM . Таблицы сгруппированы по схемам, по умолчанию у нас уже активирована дефолтная схема.

Про схему

Схема — это логическая структура, которая определяет организацию объектов базы данных. Схема включает в себя описание и отношения между различными объектами базы данных, такими как таблицы, представления, индексы, функции и другие.

Схема включает в себя описание и отношения между различными объектами базы данных, такими как таблицы, представления, индексы, функции и другие.

Каждая схема имеет уникальное имя в пределах базы данных и может содержать набор объектов, связанных с определенной предметной областью или функциональным блоком.

Про схему

Компоненты схемы:

1. **Таблицы:** определяют основную структуру данных, которая хранится в базе данных.
2. **Индексы:** используются для ускорения поиска данных в таблицах.
3. **Представления:** виртуальные таблицы, созданные на основе запросов к другим таблицам или представлениям.
4. **Процедуры и функции:** хранимые программы, которые могут выполнять операции с данными в базе данных.
5. **Права доступа и роли:** определяют уровни доступа к объектам схемы для пользователей и ролей.

Про схему

Основные аспекты схемы в SQL:

1. Структурная организация. Схема определяет структуру базы данных в терминах её объектов и их отношений друг к другу. Это включает в себя определения таблиц, столбцов и типов данных, а также ограничений (constraints) и отношений между таблицами (например, внешние ключи).
2. Пространство имен. Схема действует как пространство имен, которое позволяет однозначно идентифицировать каждый объект в базе данных. В больших системах, где может быть множество таблиц, представлений и других объектов, схемы помогают организовать их в логические группы, упрощая управление и доступ.
3. Безопасность. Схемы могут использоваться для управления доступом к данным. Права доступа могут быть назначены на уровне схемы, позволяя администраторам контролировать, кто может видеть или изменять данные в определенных объектах базы данных.
4. Логическая изоляция. В многопользовательской среде схемы позволяют разделять данные и объекты базы данных между различными пользователями или группами пользователей, обеспечивая логическую изоляцию и организацию данных.

Про схему

В разных системах управления базами данных схемы могут реализовываться по-разному. Например, в некоторых СУБД, таких как PostgreSQL, схема является ключевым элементом архитектуры и активно используется для организации данных.

В других, например в MySQL, аналогичная функциональность может достигаться через использование баз данных в качестве верхнего уровня организации, а схемы и базы данных могут рассматриваться как взаимозаменяемые понятия в контексте SQL команд.

Описание структуры данных

База данных отеля состоит из пяти таблиц:

Таблица rooms

Хранит информацию о номерах отеля.

Поле	Тип данных	Описание
room_number	SERIAL PRIMARY KEY	Уникальный номер комнаты (Primary Key)
type_name	TEXT	Название типа номера
price_per_night	INTEGER	Цена за ночь в рублях
room_size_sqm	INTEGER	Площадь номера в квадратных метрах
max_occupancy	INTEGER	Максимальное количество гостей
has_minibar	BOOLEAN	Наличие минибара (True/False)
payment_option	TEXT	Вариант оплаты

Описание структуры данных

Таблица clients

Содержит информацию о клиентах отеля.

Поле	Тип данных	Описание
id	SERIAL PRIMARY KEY	Уникальный идентификатор клиента (Primary Key)
first_name	TEXT	Имя клиента
last_name	TEXT	Фамилия клиента
address	TEXT	Адрес клиента
email	TEXT	Адрес электронной почты клиента
phone_number	VARCHAR	Номер телефона клиента

Описание структуры данных

Таблица bookings

Содержит информацию о бронированиях номеров отеля.

Поле	Тип данных	Описание
booking_id	SERIAL PRIMARY KEY	Уникальный идентификатор бронирования (Primary Key)
renter_id	INTEGER REFERENCES clients(id)	Идентификатор клиента (внешний ключ)
room_number	INTEGER REFERENCES rooms(room_number)	Номер комнаты (внешний ключ)
check_in_date	TIMESTAMP	Дата заезда
check_out_date	TIMESTAMP	Дата заезда

Описание структуры данных

Таблица payments

Хранит информацию об оплатах за бронирования.

Поле	Тип данных	Описание
payment_id	SERIAL PRIMARY KEY	Уникальный идентификатор платежа (Primary Key)
booking_id	INTEGER REFERENCES bookings(booking_id)	Идентификатор бронирования (внешний ключ)
amount_paid	DECIMAL	Сумма оплаты
payment_date	DATE	Дата платежа
payment_method	VARCHAR	Метод оплаты

Описание структуры данных

Таблица ratings

Хранит информацию об оценках, данной клиентами за проживание в номерах отеля.

Поле	Тип данных	Описание
rating_id	SERIAL PRIMARY KEY	Уникальный идентификатор оценки (Primary Key)
booking_id	INTEGER REFERENCES bookings(booking_id)	Идентификатор бронирования (внешний ключ)
rating_value	INTEGER	Значение оценки (от 1 до 5 звёзд)