

**ДОКЛАД**

по дисциплине “Введение в специальность”  
на тему “Реляционные базы данных”

Выполнил:  
Студент группы  
К 0709 23/3  
Скляр Илья Андреевич

Принял:  
Старший преподаватель  
Тенигин Альберт Андреевич

---

## Оглавление

Введение .....	3
Что такое SQL .....	4
История создания SQL .....	6
Диалекты SQL .....	7
Возможности SQL .....	9
Область применения .....	13
Вывод .....	14
Список литературы .....	15

## **Введение**

В современном мире, где интернет имеет важную роль в жизни человека, не обойтись без баз данных. Зачем нужны базы данных в интернете? Можно сказать, что база данных, это такой виртуальный ящик в котором находится информация в том или ином виде. Они нужны для хранения данных, на что указывает их название, в них может храниться любая информация в объеме который определяется характеристиками вашего компьютера/сервера на котором находится база данных, если это ваша база данных только для вас, то нет необходимости размещать её на сервер и можно оставить локально, но если к базе данных постоянно идут запросы, то необходимо разметить ее на сервер, чтобы она могла принимать запросы и отвечать на них в любое время. Но как взаимодействовать с базой данных? Для этого был создан язык запросов к базе данных SQL. SQL создан для работы с реляционными базами данных, если база данных реляционная, это означает что в ней строгая структура хранения данных, самый простой пример реляционного хранения данных, это таблица в которой есть название у каждой колонки и они не повторяются а так же в колонке может храниться только тот тип данных, который был указан при её создании. Для работы с запросами к базам данных sql было создано несколько систем управления базами данных, одни из самых популярных это: PostgreSQL, MySQL, Microsoft SQL Server, SQLite, Oracle Database.

## Что такое SQL

SQL интуитивно понятный язык с низким порогом входа, своим синтаксисом он очень похож на английский язык и за счет этого человек который даже не знает sql, но знает английский может понять, что происходит в коде. SQL используется везде, где необходимо хранить данные, например на сайтах которые используются только для чтения информации, где нет необходимости использования базы данных.

## Плюсы SQL

- Быстрая и эффективная обработка запросов, что делает такие команды как извлечь почти моментальными
- Низкий порог входа, за счет того, что он очень сильно похож на английский а так же интуитивно понятен
- SQL может использоваться как на ПК так и серверах, так же он может использоваться на разных ОС
- Базы данных sql имеют встроенные функции безопасности
- Создание резервной копии, базы данных sql имеют встроенные средства копирования для того, чтоб в случае сбоя была возможность восстановить данные.
- Таблицы sql могут быть связаны и sql гарантирует, что если изменения произошли в одной таблице то все связанные с ней таблицы получат изменения.

## Недостатки SQL

- SQL недостаточно гибкая по сравнению с NOSQL.
- Нет возможности следить за базой данных в реальном времени из-за того, что sql предназначен для пакетной обработки.

## **История создания SQL**

SQL создали Рэймондом Бойсом и Дональдам Чемберлин в 1974 и изначально назывался SEQUEL, что расшифровывалось как «Structured English Query Language», но потом было убрано слово «English», после чего аббревиатура приняли привычный вид «SQL». SQL создавался в рамках проекта экспериментальной реляционной СУБД System R. Занималась этим проектом компания IBM. SQL была не единственным языком управления реляционными базами данных, в Калифорнийском Университете Беркли была разработана СУБД Ingres, но она не выдержала конкуренции по количеству поддерживающих пользователей.

SQL была разработана:

Рэймондом Бойсом(1947-1974, 27 лет, умер от аневризмы) - американский информатик, известен исследованиями реляционных баз данных. Работал в IBM и будучи главой группы разработки баз данных вместе с Дональдам Чемберлином создал язык SQL.

Дональдам Чемберлин(1944, 79 лет) - американский ученый-компьютерщик, является одним из главных разработчиков языка SQL, в 1997 был избран членом национальной инженерной академии за вклад в язык запросов к базам данных SQL

## Диалекты SQL

Что такое диалекты SQL и как они появляются? Диалекты это результат появления различных систем управления базами данных, Диалекты это разновидности sql которые за счёт дополнительных функций отошли от промышленного стандарта ANSI SQL.

Вот несколько примеров диалектов, чем они похожи, то что в них от ANSI SQL:

PostgreSQL:

code:

```
select * from users where id > 3
```

# из таблицы «users» вывести все данные где «id» больше 3

MySQL:

code:

```
select * from users where id > 3
```

# из таблицы «users» вывести все данные где «id» больше 3

Microsoft SQL Server:

code:

```
select * from users where id > 3
```

# из таблицы «users» вывести все данные где «id» больше 3

SQLite:

code:

```
select * from users where id > 3
```

# из таблицы «users» вывести все данные где «id» больше 3

Диалекты нужны для создания более сложных и продвинутых баз данных, ориентированных на конкретную деятельность и оптимизированные под конкретную задачу, так же благодаря диалектам разработчики могут использовать различные функции которые недоступны в стандарте ANSI SQL. К примеру некоторые диалекты дают разработчикам работать с географическими данными.

Как можно использовать и что нужно учитывать? Разработчикам при создании приложения очень важно учитывать используемый sql диалект к примеру при работе с PostgreSQL, стоит учитывать и использовать только его синтаксис и его функции.

Так же есть один метод обработки диалектов sql, это использование ORM(объектно-реляционное отображение). Благодаря ORM разработчики могут писать код скрывая особенности диалекта а ORM сам генерирует необходимые SQL запросы.

Диалекты крайне необходимая вещь для разработчиков, а ORM делает диалекты универсальными и это дает очень большие возможности в создании баз данных.



## Возможности SQL

У SQL обладает большим количеством возможностей, которые необходимы при работе с реляционной базой данных, такие как типы данных и методы работы с ними.

### Типы данных

Основные типы данных SQL:

- Строковые(string): строковый тип данных самый распространенный, в них хранятся такие данные как: адреса, имена, название и т.д. Все строковые данные записываются в одинарных кавычках. Так же строки могут быть фиксированной длины и переменной.

Основные типы данных:

CHAR - фиксированная длина строки. В процессе создания таблицы определяется точное значение - от 1 до 225 символов;

NCHAR - одна из разновидностей CHAR, которая поддерживает Unicode или многобайтовые символы

TEXT - строки переменной длины

NVARCHAR - подвид TEXT, которые поддерживает Unicode или многобайтовые символы

- Числовые данные: этот тип данных при записи не требует кавычек, над ним можно проводить логические операции, а типы данных отличаются диапазоном для хранимого числа, чем меньше диапазон тем меньше места он занимает, а так же они отличаются возможностью хранить дробные числа.

Основные типы данных(в зависимости от СУБД некоторые названия могут меняться):

NUMBER или FLOAT - числа с плавающими точками

NUMERIC или DECIMAL - числа с фиксированными или плавающими точками;

BIT - одноразрядное значение, которое используют для битовых флагов: 0 или 1;

REAL - 4-байтовые числа с плавающими точками;

INTEGER или INT - целые 4-байтовые числа, у которых диапазон значений варьируется от -2147483648 до 2147483647;

TINYINT - целые 1-байтовые числа в диапазоне от 0 до 255;

SMALLINT - целые 2-байтовые числа в диапазоне от -32768 до 32767.

- Дата и время: типы данных отличаются точностью и допустимым диапазоном.

Основные типы данных(есть и более точные, но эти основные):

DATE -дата;

TIME - время;

TIMESTAMP или DATETIME - дата и время;

SMALLDATETIME - дата и время с точностью до минуты.

- Бинарные типы данных: эти типы данных обеспечивают хранение графики, текста, аудио в бинарном коде, но используют их редко, отличаются типы данных хранимым диапазоном и могут быть фиксированной и не фиксированной длины.

Основные типы данных:

BINARY – данные в двоичном виде в диапазоне от 255 до 8000 байт;

RAW – данные фиксированной длины в двоичном виде в диапазоне до 255 байт;

LONG RAW – данные переменной длины в двоичном виде в диапазоне до 2Гбайт;

VARBINARY – данные переменной длины в двоичном виде в диапазоне до 8000 байт или до 255 байт в зависимости от реализации.

- Булевыe и логические типы данных: этот тип данных позволяет хранить значения True или False

Типы даннх:

Bool -позволяет хранить значения True или False, что означает «да» или «нет»

### Методы работы с базой данных

Задачи	Метод	Работа метода
Создание таблиц	CREATE TABLE	Позволяет создавать новые таблицы в базе данных, определяя структуру данных.
Вставка данных	INSERT INTO	Добавляет новые строки данных в таблицу.
Выбор данных	SELECT	Извлекает данные из одной или нескольких таблиц.
Обновление данных	UPDATE	Меняет значения в существующих строках таблицы.
Удаление данных	DELETE	Удаляет строки данных из таблицы.

Управление таблицами	ALTER TABLE	Позволяет изменять структуру существующей таблицы, добавлять, изменять или удалять столбцы.
Создание базы данных	CREATE DATABASE	Создает новую базу данных.
Удаление базы данных	DROP DATABASE	Удаляет базу данных.
Управление пользователями и привилегиями	GRANT, REVOKE	Предоставляет привилегии пользователям, отменяет привилегии.
Фильтрация данных	WHERE	Позволяет задавать условия для выбора определенных строк данных.
Сортировка данных	ORDER BY	Сортирует результаты запроса по указанным столбцам.
Группировка данных	GROUP BY	Группирует строки на основе значений в одном или нескольких столбцах.
Агрегационные функции	SUM, AVG, MIN, MAX, COUNT	Выполняют агрегационные вычисления над группами данных.
Соединение таблиц	JOIN	Объединяет строки из двух или более таблиц на основе условий соответствия.
Транзакции	BEGIN TRANSACTION, COMMIT, ROLLBACK	Обеспечивают целостность данных в базе данных, позволяя группировать операции в транзакции.
Подзапросы	SELECT внутри SELECT	Позволяет выполнять запросы внутри других запросов.

Эти возможности делают SQL мощным инструментом для управления данными в реляционных базах данных.

## **Область применения**

У SQL очень большая область применения, он используется банками веб-сайтами, компьютерными играми, мед учреждениями, различными приложениями, в научных исследованиях, в государственных системах, облачных хранилищах и т.д. SQL применим во многих видах программного обеспечения и приложений. Это мощный язык программирования при управлении информацией в реляционных базах данных. Широко распространен в промышленности и программных системах.

## **Вывод**

SQL это крайне необходимый инструмент в постиндустриальном обществе где ключевую роль играет информация, SQL позволяет хранить информацию и обрабатывать её, он крайне необходим при разработке программ, сайтов, сервисов и т.д. Он открывает новые возможности в хранении и управлении базами данных.

### Список литературы

1. Алан Болье «Изучаем SQL». - М.: Компьютерное издательство "Диалектика", 2021. - 403с.
2. Аллен Тейлор «SQL для чайников». - М.: Компьютерное издательство "Диалектика". 2020. - 546с.
3. Уолтер Шилдс «SQL: «быстрое погружение»». - М.: Издательский дом «Питер». 2022. - 223с.
4. Молинаро Энтони, де Грааф Роберт «SQL. Сборник рецептов». - М.: «БХВ-Петербург», 2022. - 594с.
5. Фиайли Крис «SQL» : Пер. с англ. – «ДМК Пресс». 2013. - 456с.
6. Фуфаев Э.В. «Базы данных». - М.: «Академия», 2005. - 320с.
7. Кузнецов С.Д. «SQL: Язык реляционных баз данных», - М.: «Майор», 2001. - 192с.
8. Савицкий Н.И. «Технологии организации, хранения и обработки данных», - М.: «ИНФРА-М», 2021. - 232с.