**1. Введение в SQL**

**SQL (Structured Query Language)** – стандартизированный язык для работы с реляционными базами данных. Позволяет:

* Создавать и изменять структуру БД (DDL).
* Вставлять, обновлять, удалять данные (DML).
* Запрашивать данные (SELECT).
* Управлять правами доступа (DCL).

**2. Стандарты SQL**

* **SQL-86 / SQL-87** – первый стандарт.
* **SQL-92 (SQL2)** – базовый стандарт, поддерживается всеми СУБД.
* **SQL:1999 (SQL3)** – добавлены рекурсивные запросы, триггеры, ООП-возможности.
* **SQL:2003, SQL:2008, SQL:2011, SQL:2016** – расширения (JSON, временные данные, оконные функции).

🔹 *Большинство СУБД (PostgreSQL, MySQL, Oracle) поддерживают SQL-92 + свои расширения.*

**3. Виды SQL**

1. **Интерактивный SQL** – выполнение запросов напрямую (через консоль, GUI).
2. **Встроенный SQL** – интеграция в языки программирования (Python, Java).
3. **Динамический SQL** – генерация запросов на лету (например, в веб-приложениях).

**4. Категории SQL-команд**

| **Категория** | **Описание** | **Примеры команд** |
| --- | --- | --- |
| **DDL (Data Definition Language)** | Создание и изменение структуры БД | CREATE, ALTER, DROP, TRUNCATE, RENAME |
| **DML (Data Manipulation Language)** | Работа с данными (вставка, обновление, удаление) | SELECT, INSERT, UPDATE, DELETE, MERGE |
| **DCL (Data Control Language)** | Управление доступом | GRANT, REVOKE, DENY |
| **TCL (Transaction Control Language)** | Управление транзакциями | COMMIT, ROLLBACK, SAVEPOINT |

**Примеры команд:**

sql

Copy

-- DDL: Создание таблицы

CREATE TABLE Students (

id INT PRIMARY KEY,

name VARCHAR(50),

age INT

);

-- DML: Вставка данных

INSERT INTO Students (id, name, age) VALUES (1, 'Alice', 20);

-- DML: Запрос данных

SELECT \* FROM Students WHERE age > 18;

-- DCL: Дать права пользователю

GRANT SELECT ON Students TO user1;

**5. Пример запроса (SELECT)**

sql

Copy

SELECT

s.name AS student\_name,

c.name AS course\_name,

g.grade

FROM

Students s

JOIN

Grades g ON s.id = g.student\_id

JOIN

Courses c ON g.course\_id = c.id

WHERE

c.department = 'Computer Science'

ORDER BY

s.name;

**6. Объединения (JOINs) в SQL**

Объединения связывают данные из разных таблиц по ключам.

**Типы JOIN (круги Эйлера)**

| **Тип JOIN** | **Описание** | **Пример (SQL)** | **Визуализация** |
| --- | --- | --- | --- |
| **INNER JOIN** | Только совпадающие строки | SELECT \* FROM A INNER JOIN B ON A.id = B.id | Пересечение A и B |
| **LEFT JOIN** | Все строки из A + совпадения из B (NULL, если нет) | SELECT \* FROM A LEFT JOIN B ON A.id = B.id | Вся A + пересечение с B |
| **RIGHT JOIN** | Все строки из B + совпадения из A (NULL, если нет) | SELECT \* FROM A RIGHT JOIN B ON A.id = B.id | Вся B + пересечение с A |
| **FULL JOIN** | Все строки из A и B (NULL, если нет совпадений) | SELECT \* FROM A FULL JOIN B ON A.id = B.id | Объединение A и B |
| **CROSS JOIN** | Декартово произведение (все комбинации) | SELECT \* FROM A CROSS JOIN B | A × B |

**Примеры JOIN (на таблицах**Students**и**Grades**)**

**Таблица Students**

| **id** | **name** |
| --- | --- |
| 1 | Alice |
| 2 | Bob |

**Таблица Grades**

| **student\_id** | **course** | **grade** |
| --- | --- | --- |
| 1 | Math | A |
| 3 | Physics | B |

**1. INNER JOIN**

sql

Copy

SELECT s.name, g.course, g.grade

FROM Students s

INNER JOIN Grades g ON s.id = g.student\_id;

**Результат:**

| **name** | **course** | **grade** |
| --- | --- | --- |
| Alice | Math | A |

**2. LEFT JOIN**

sql

Copy

SELECT s.name, g.course, g.grade

FROM Students s

LEFT JOIN Grades g ON s.id = g.student\_id;

**Результат:**

| **name** | **course** | **grade** |
| --- | --- | --- |
| Alice | Math | A |
| Bob | NULL | NULL |

**3. RIGHT JOIN**

sql

Copy

SELECT s.name, g.course, g.grade

FROM Students s

RIGHT JOIN Grades g ON s.id = g.student\_id;

**Результат:**

| **name** | **course** | **grade** |
| --- | --- | --- |
| Alice | Math | A |
| NULL | Physics | B |

**4. FULL JOIN**

sql

Copy

SELECT s.name, g.course, g.grade

FROM Students s

FULL JOIN Grades g ON s.id = g.student\_id;

**Результат:**

| **name** | **course** | **grade** |
| --- | --- | --- |
| Alice | Math | A |
| Bob | NULL | NULL |
| NULL | Physics | B |

**7. Визуализация JOIN (круги Эйлера)**

Copy

INNER JOIN: A ∩ B

LEFT JOIN: A ∪ (A ∩ B)

RIGHT JOIN: B ∪ (A ∩ B)

FULL JOIN: A ∪ B

CROSS JOIN: A × B (все возможные пары)

**Итог**

* SQL – мощный язык для работы с реляционными БД.
* Основные категории команд: **DDL, DML, DCL, TCL**.
* **JOIN** позволяет связывать данные из разных таблиц.
* Выбор типа JOIN зависит от задачи (INNER, LEFT, RIGHT, FULL, CROSS).