* ﻿﻿﻿﻿**Команды языка SQL для работы с запросами на выборку данных**

**SELECT с условиями (WHERE)**

Для фильтрации записей используется WHERE

String query = "SELECT id, name, age FROM employees WHERE department\_id = 2";

try (Connection conn = DriverManager.getConnection(url, user, password);

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(query)) {

while (rs.next()) {

// Обработка каждой строки результата

}

} catch (SQLException e) {

e.printStackTrace();

}

**Агрегация данных (GROUP BY, COUNT, MAX, MIN, AVG)**

SQL предоставляет мощные инструменты для агрегации данных, такие как COUNT, MAX, MIN, AVG:

String query = "SELECT department\_id, COUNT(\*) AS num\_employees FROM employees GROUP BY department\_id";

try (Connection conn = DriverManager.getConnection(url, user, password);

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(query)) {

while (rs.next()) {

int departmentId = rs.getInt("department\_id");

int count = rs.getInt("num\_employees");

System.out.println("Department ID: " + departmentId + " has " + count + " employees.");

}

} catch (SQLException e) {

e.printStackTrace();

}

**Соединения таблиц (JOIN)**

Для объединения данных из нескольких таблиц можно использовать различные типы JOIN, например, INNER JOIN, LEFT JOIN:

String query = "SELECT e.name, d.name AS department\_name FROM employees e INNER JOIN departments d ON e.department\_id = d.id";

try (Connection conn = DriverManager.getConnection(url, user, password);

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(query)) {

while (rs.next()) {

String employeeName = rs.getString("name");

String departmentName = rs.getString("department\_name");

System.out.println("Employee: " + employeeName + ", Department: " + departmentName);

}

} catch (SQLException e) {

e.printStackTrace();

}