Introduction to SQL

- Introduction to SQL
- Databases and data management
- Writing basic queries

Why Do We Need Databases?

- Large amounts of data
- Persistent storage
- Efficient querying
- Structured access and consistency

What Is a Database?

- System for storing data
- Supports querying and updates
- Manages consistency and reliability

Types of Databases

- Relational databases
- NoSQL databases
- Specialized databases

Relational Databases

- Data stored in tables
- Fixed schema
- Relationships via keys
- Uses SQL

NoSQL Databases

- Flexible or schema-less
- Optimized for scalability
- Different data models

Common NoSQL Models

- Key–value stores
- Document databases

What Is SQL?

- Structured Query Language
- Used for relational databases
- Declarative language

SQL Is a Standard

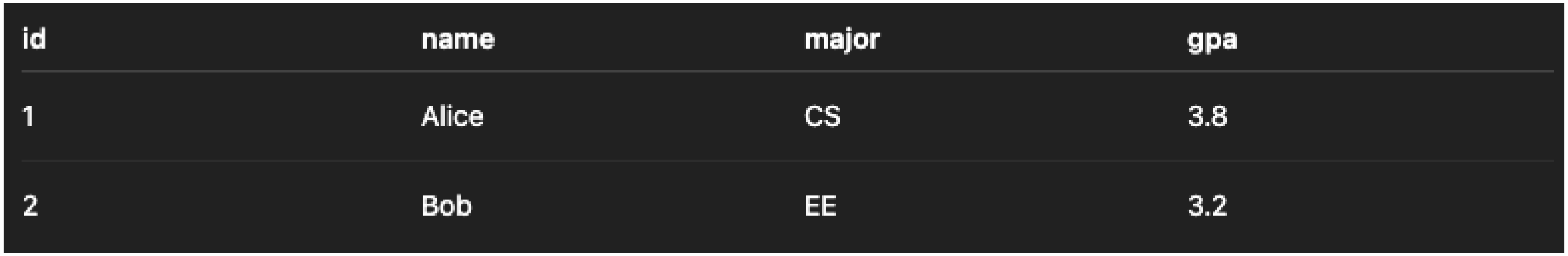
- Standardized language
- Minor syntax differences
- Same core concepts

What Is a Table?

- Rows = records
- Columns = attributes
- Defined schema

Example Table

- Students(id, name, major, gpa)



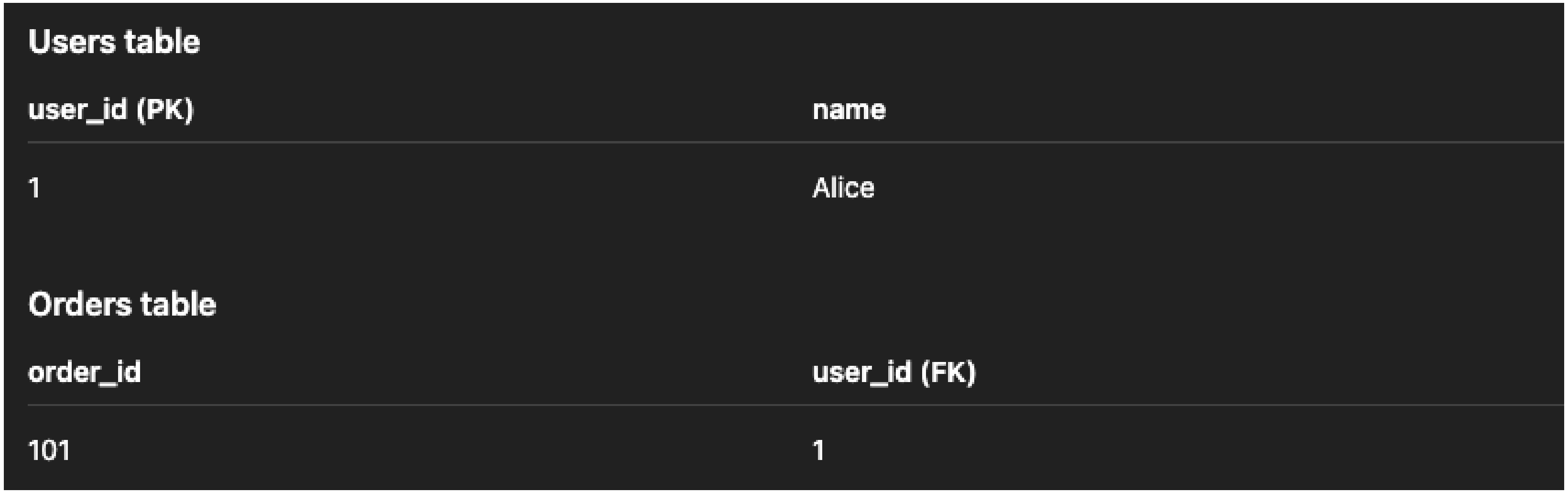
id	name	major	gpa
1	Alice	CS	3.8
2	Bob	EE	3.2

Primary Keys

- Uniquely identify rows
- Cannot be NULL
- One per table

Foreign Keys

- Reference another table
- Define relationships
- Enforce consistency



Users table

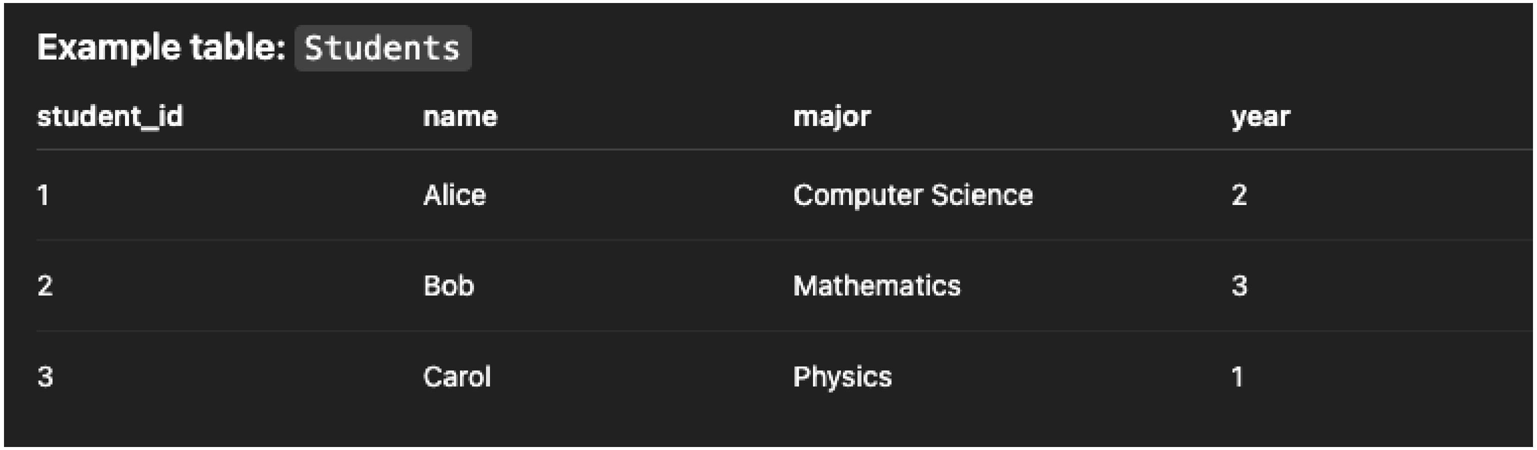
user_id (PK)	name
1	Alice

Orders table

order_id	user_id (FK)
101	1

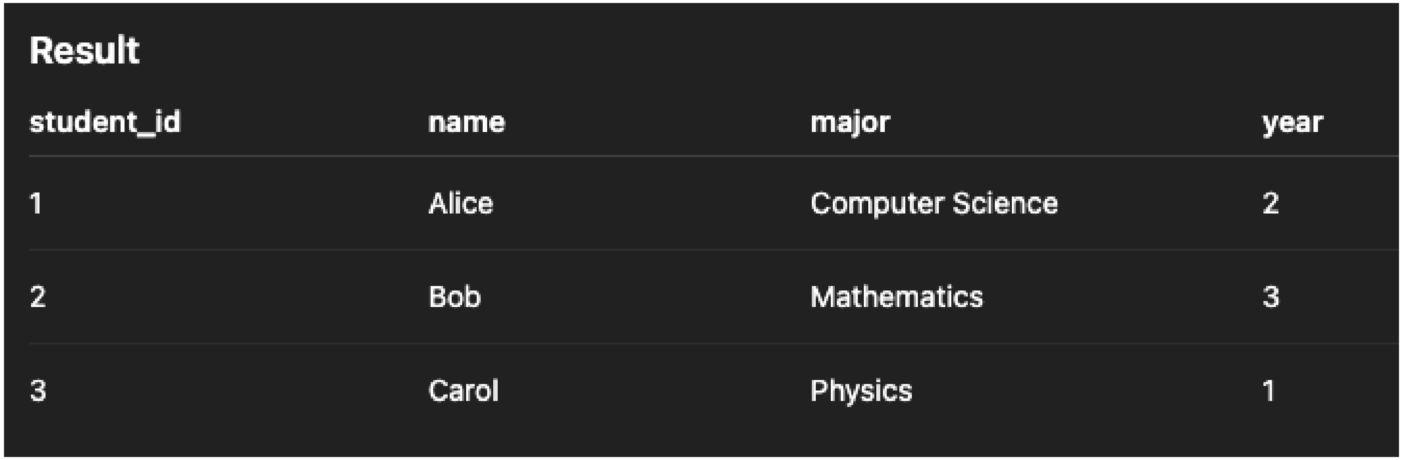
Basic SQL Query

- SELECT * FROM Students;



Example table: Students

student_id	name	major	year
1	Alice	Computer Science	2
2	Bob	Mathematics	3
3	Carol	Physics	1

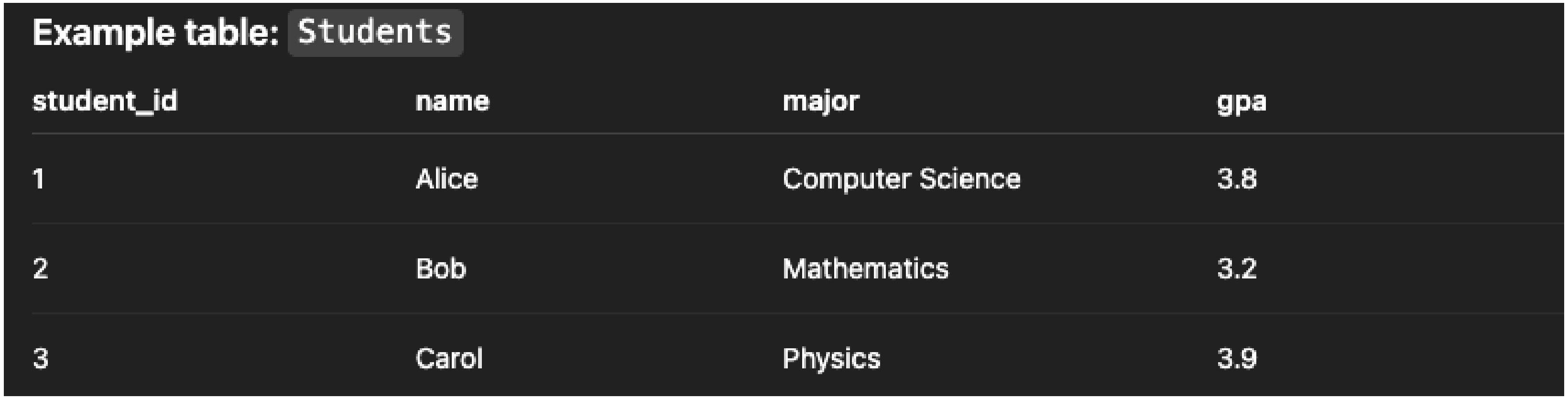


Result

student_id	name	major	year
1	Alice	Computer Science	2
2	Bob	Mathematics	3
3	Carol	Physics	1

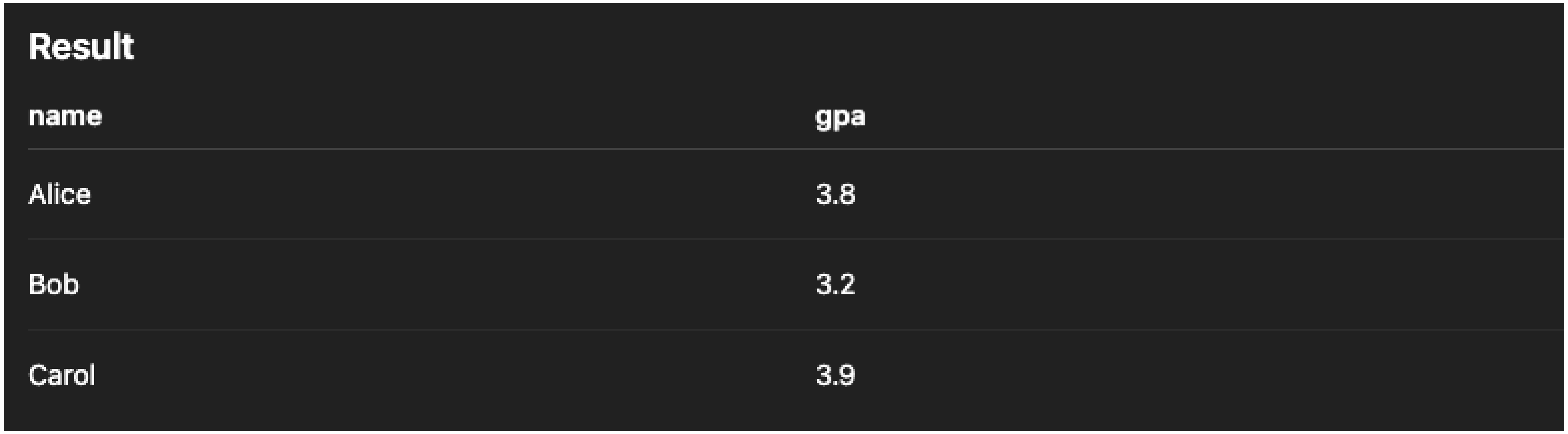
Selecting Columns

- SELECT name, gpa FROM Students;



Example table: `Students`

student_id	name	major	gpa
1	Alice	Computer Science	3.8
2	Bob	Mathematics	3.2
3	Carol	Physics	3.9

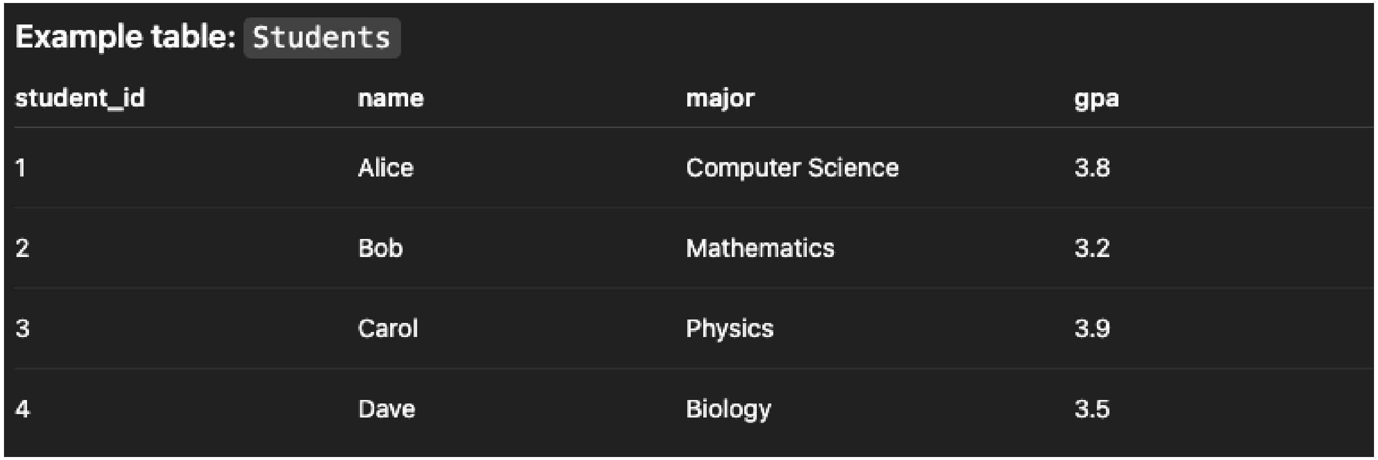


Result

name	gpa
Alice	3.8
Bob	3.2
Carol	3.9

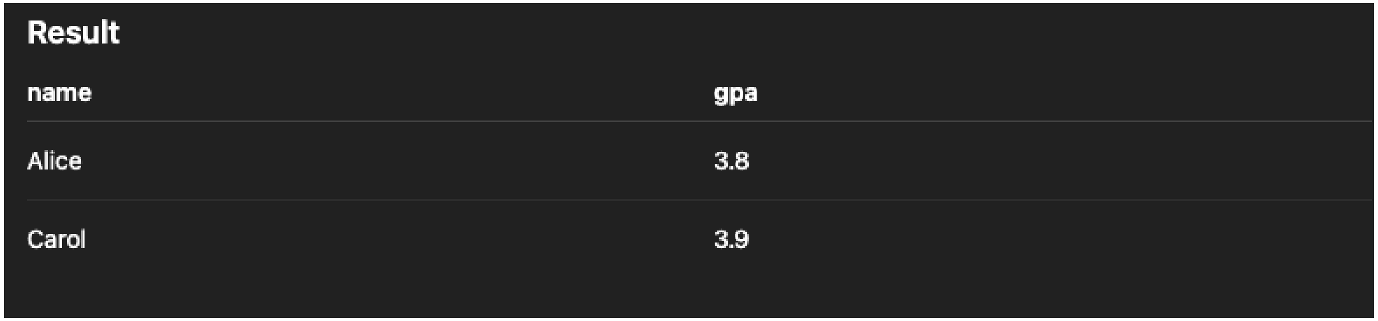
Filtering Rows

- SELECT name, gpa FROM Students WHERE gpa > 3.5;



Example table: `Students`

student_id	name	major	gpa
1	Alice	Computer Science	3.8
2	Bob	Mathematics	3.2
3	Carol	Physics	3.9
4	Dave	Biology	3.5

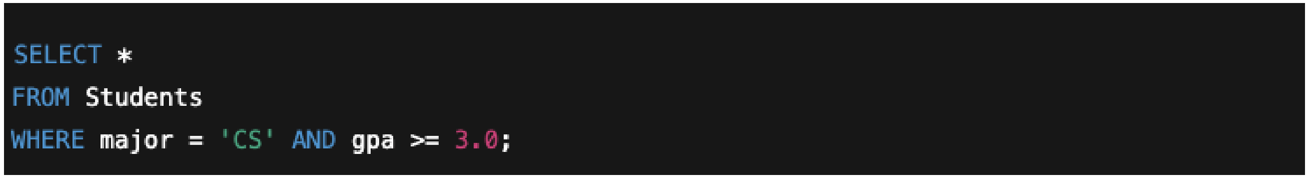


Result

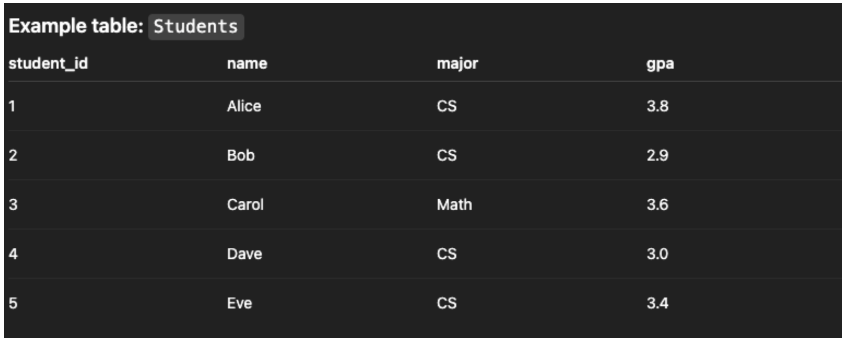
name	gpa
Alice	3.8
Carol	3.9

Multiple Conditions

- WHERE major = 'CS' AND gpa >= 3.0;

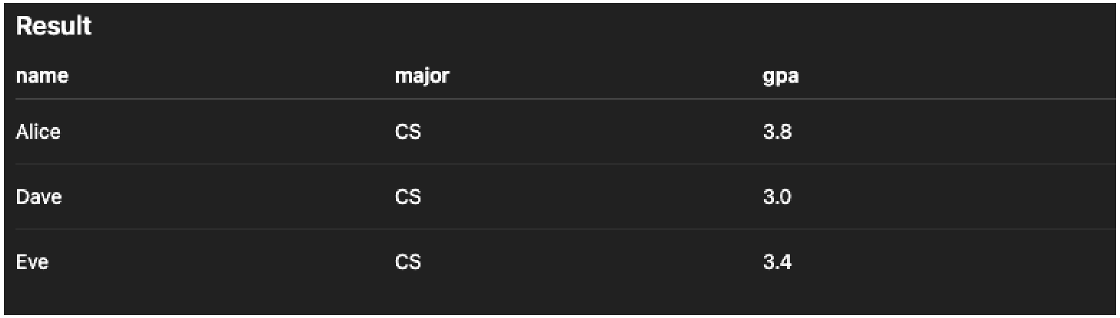


```
SELECT *
FROM Students
WHERE major = 'CS' AND gpa >= 3.0;
```



Example table: `Students`

student_id	name	major	gpa
1	Alice	CS	3.8
2	Bob	CS	2.9
3	Carol	Math	3.6
4	Dave	CS	3.0
5	Eve	CS	3.4

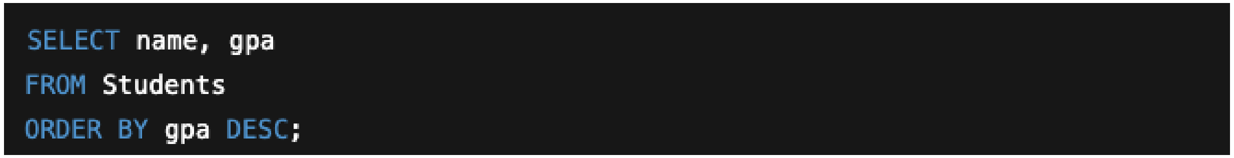


Result

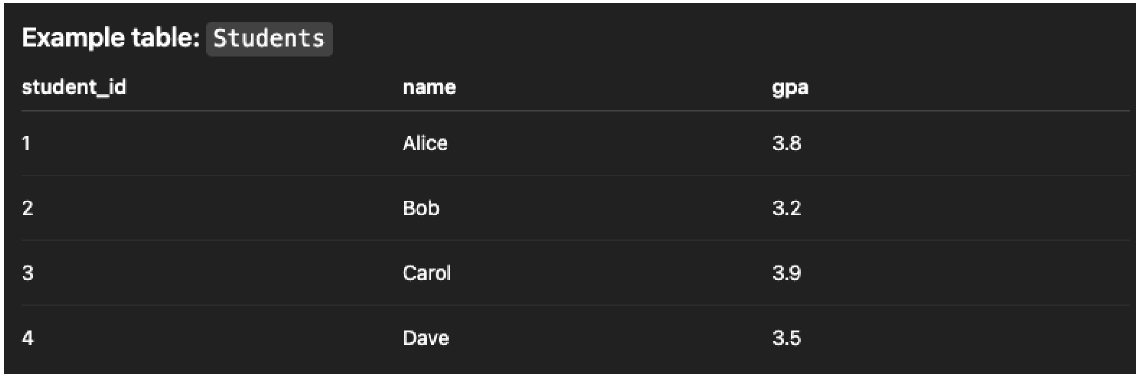
name	major	gpa
Alice	CS	3.8
Dave	CS	3.0
Eve	CS	3.4

Sorting Results

- ORDER BY gpa DESC;

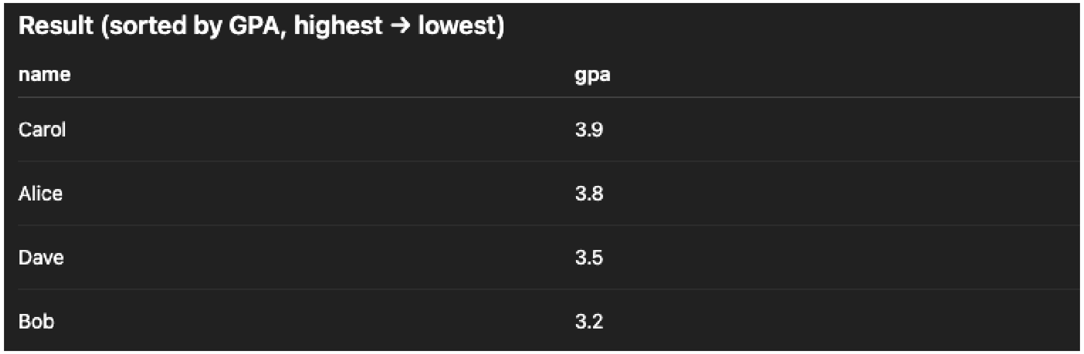


```
SELECT name, gpa
FROM Students
ORDER BY gpa DESC;
```



Example table: `Students`

student_id	name	gpa
1	Alice	3.8
2	Bob	3.2
3	Carol	3.9
4	Dave	3.5

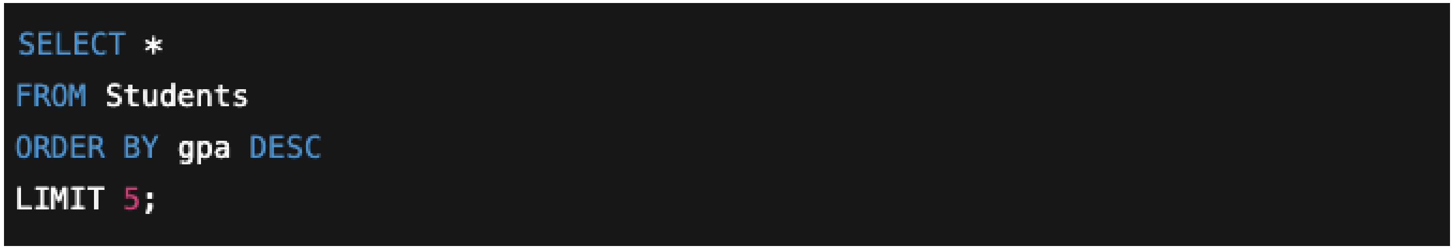


Result (sorted by GPA, highest → lowest)

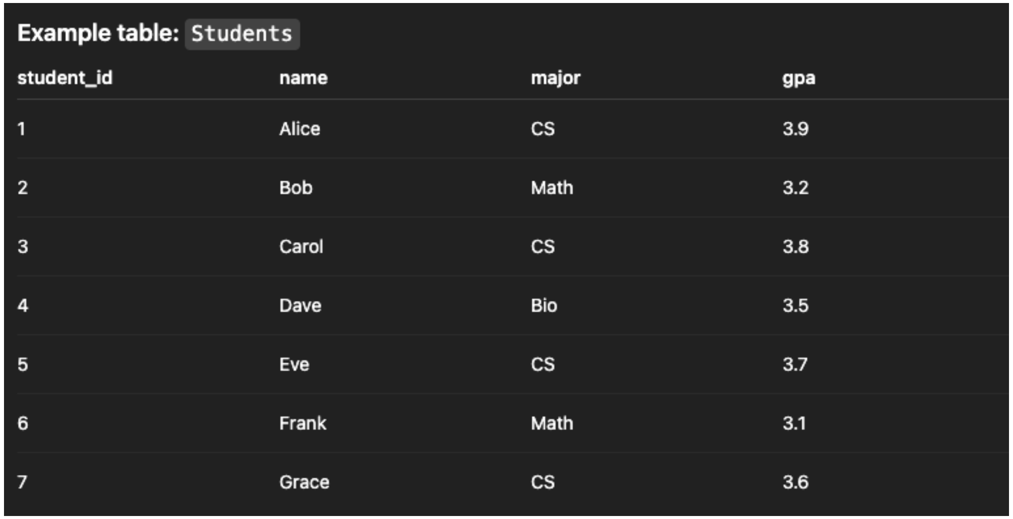
name	gpa
Carol	3.9
Alice	3.8
Dave	3.5
Bob	3.2

Limiting Results

- LIMIT 5;



```
SELECT *
FROM Students
ORDER BY gpa DESC
LIMIT 5;
```

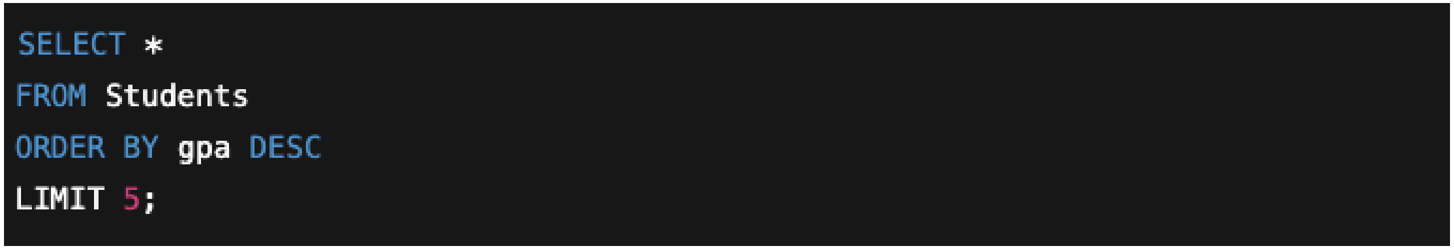


Example table: `Students`

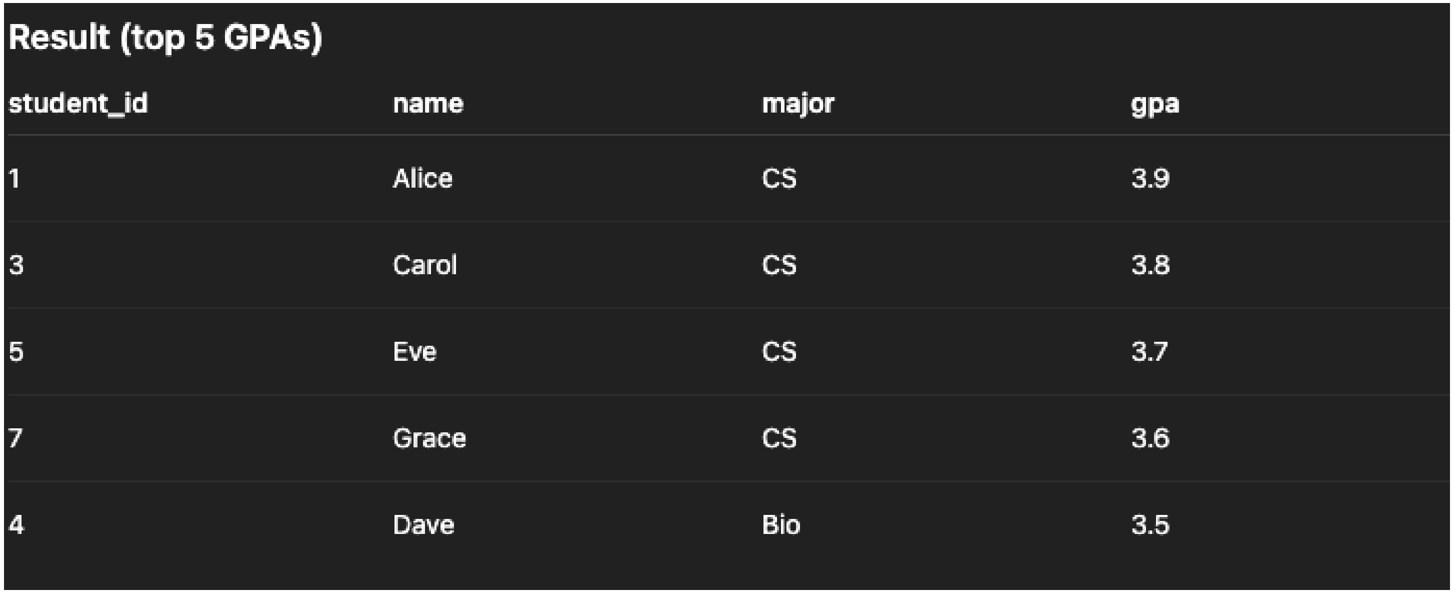
student_id	name	major	gpa
1	Alice	CS	3.9
2	Bob	Math	3.2
3	Carol	CS	3.8
4	Dave	Bio	3.5
5	Eve	CS	3.7
6	Frank	Math	3.1
7	Grace	CS	3.6

Limiting Results

- LIMIT 5;



```
SELECT *
FROM Students
ORDER BY gpa DESC
LIMIT 5;
```

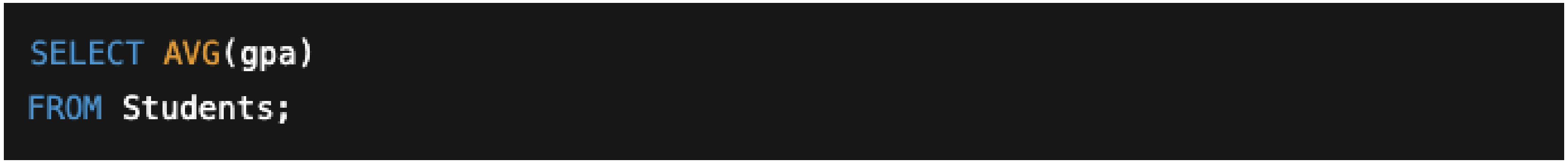


Result (top 5 GPAs)

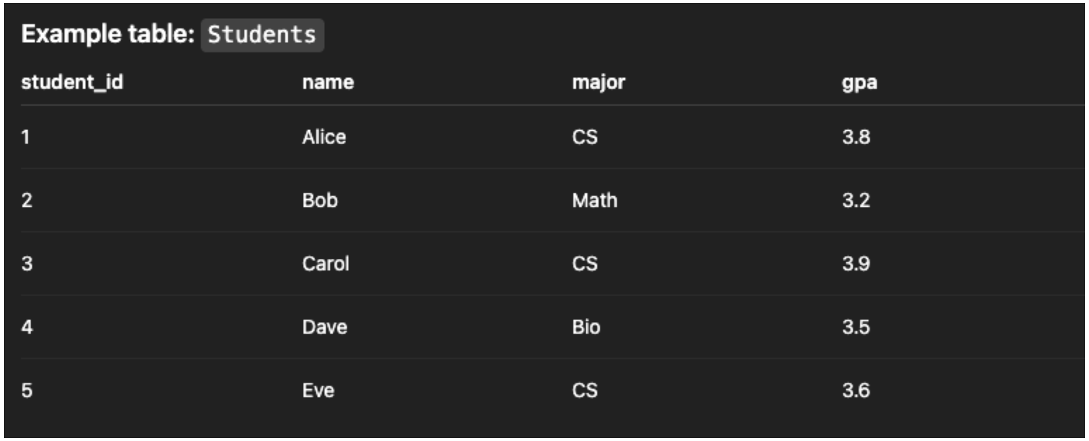
student_id	name	major	gpa
1	Alice	CS	3.9
3	Carol	CS	3.8
5	Eve	CS	3.7
7	Grace	CS	3.6
4	Dave	Bio	3.5

Aggregate Functions

- COUNT, AVG, SUM, MIN, MAX

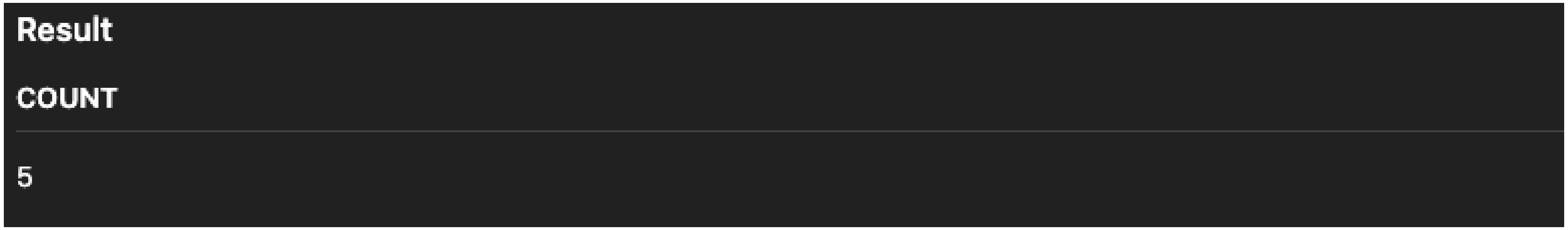


```
SELECT AVG(gpa)
FROM Students;
```



Example table: `Students`

student_id	name	major	gpa
1	Alice	CS	3.8
2	Bob	Math	3.2
3	Carol	CS	3.9
4	Dave	Bio	3.5
5	Eve	CS	3.6

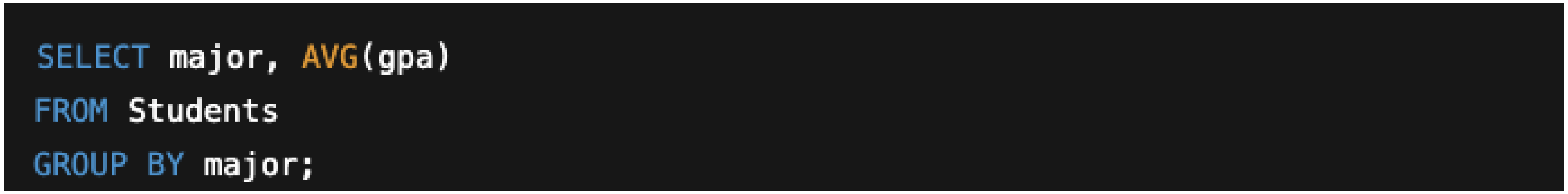


Result

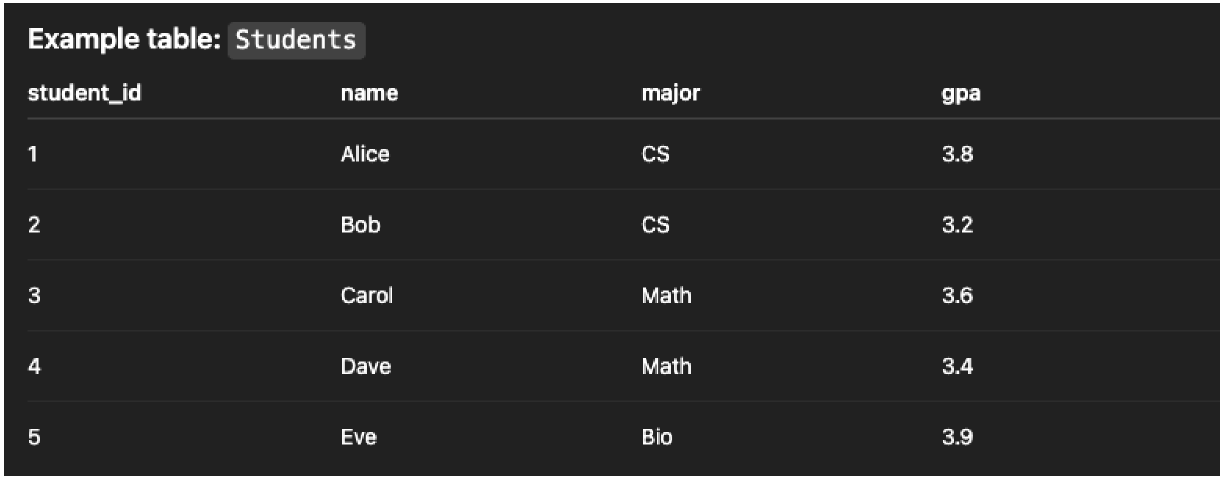
COUNT
5

GROUP BY

- SELECT major, AVG(gpa) FROM Students GROUP BY major;



```sql
SELECT major, AVG(gpa)
FROM Students
GROUP BY major;
```

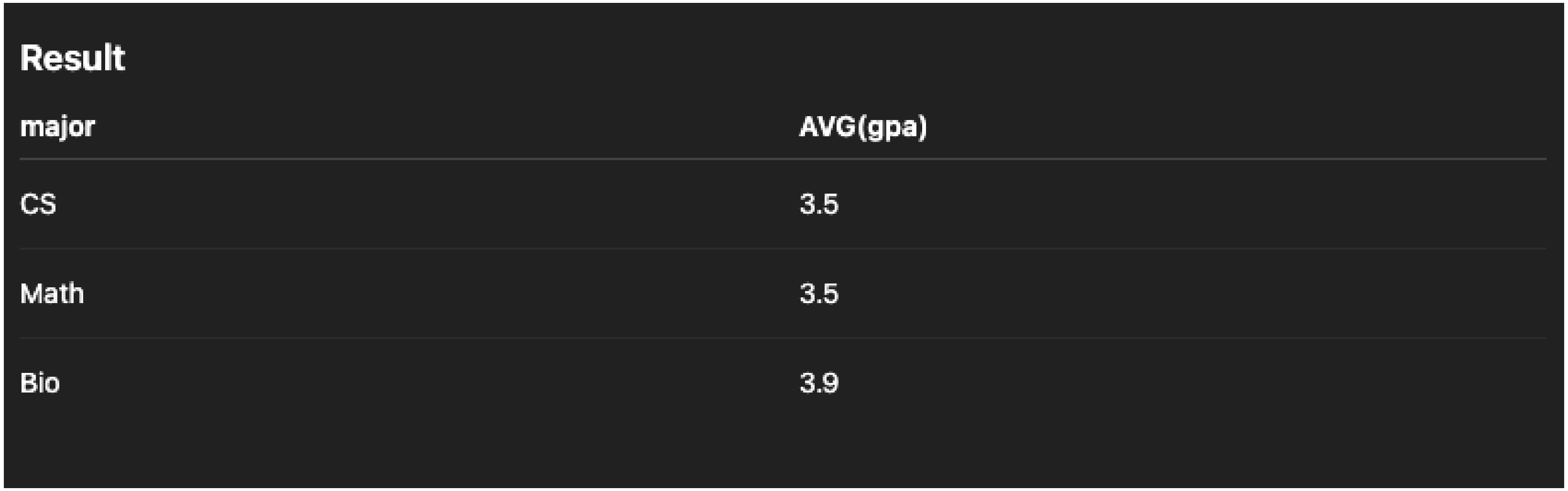


Example table: `Students`

student_id	name	major	gpa
1	Alice	CS	3.8
2	Bob	CS	3.2
3	Carol	Math	3.6
4	Dave	Math	3.4
5	Eve	Bio	3.9

GROUP BY

- SELECT major, AVG(gpa) FROM Students GROUP BY major;

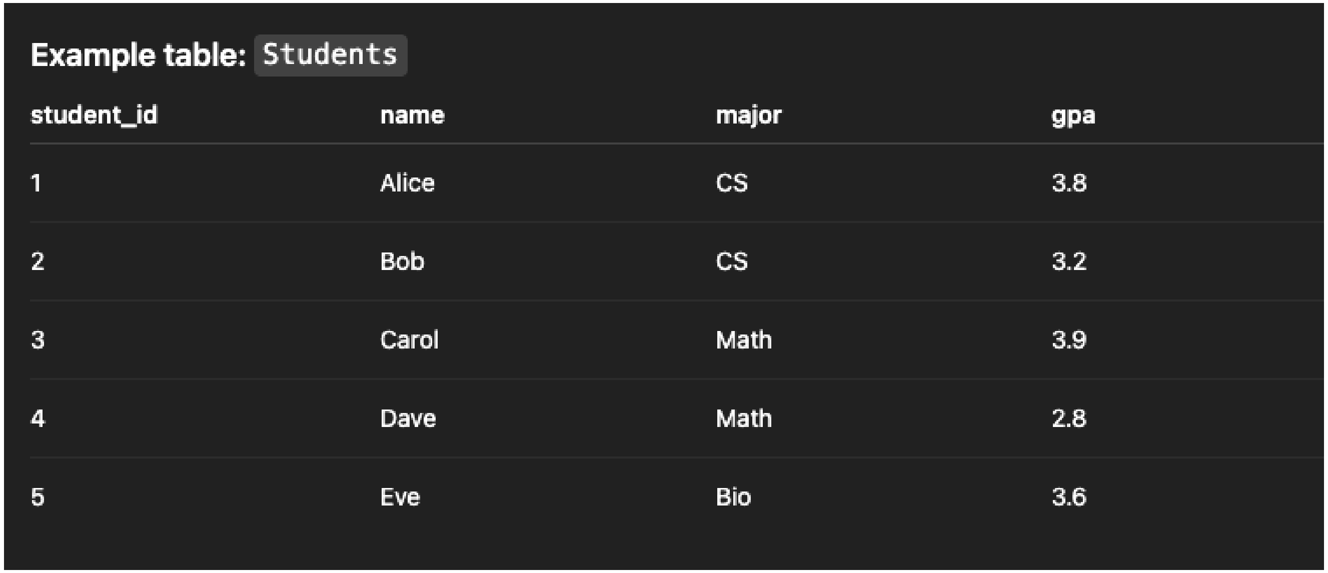


Result

major	AVG(gpa)
CS	3.5
Math	3.5
Bio	3.9

HAVING vs WHERE

- WHERE filters rows
- HAVING filters groups

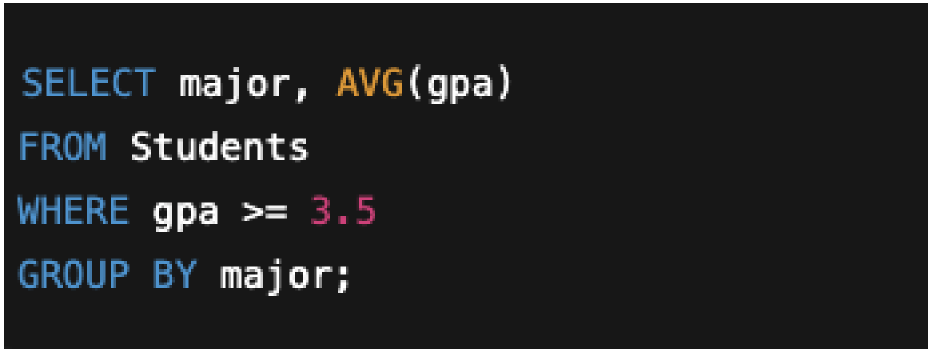


Example table: `Students`

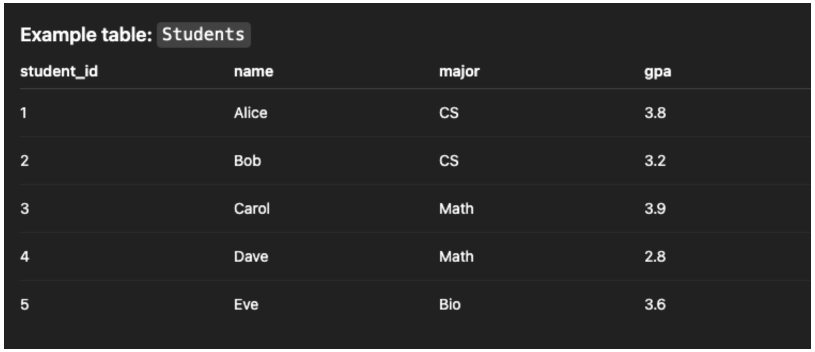
student_id	name	major	gpa
1	Alice	CS	3.8
2	Bob	CS	3.2
3	Carol	Math	3.9
4	Dave	Math	2.8
5	Eve	Bio	3.6

HAVING vs WHERE

- WHERE filters rows

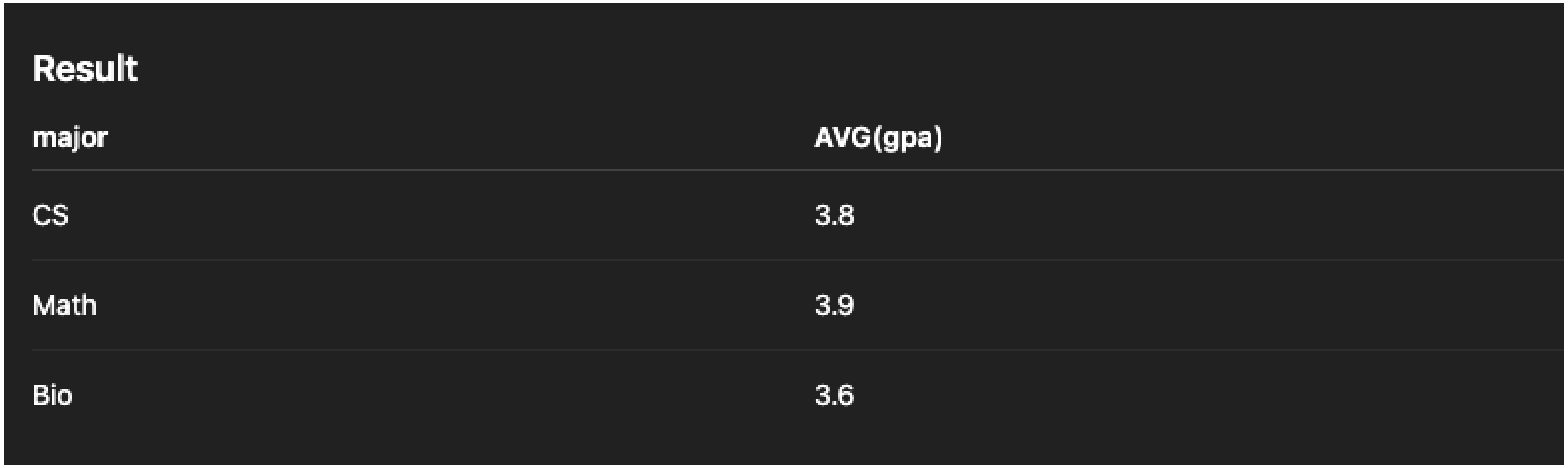


```
SELECT major, AVG(gpa)
FROM Students
WHERE gpa >= 3.5
GROUP BY major;
```



Example table: `Students`

student_id	name	major	gpa
1	Alice	CS	3.8
2	Bob	CS	3.2
3	Carol	Math	3.9
4	Dave	Math	2.8
5	Eve	Bio	3.6

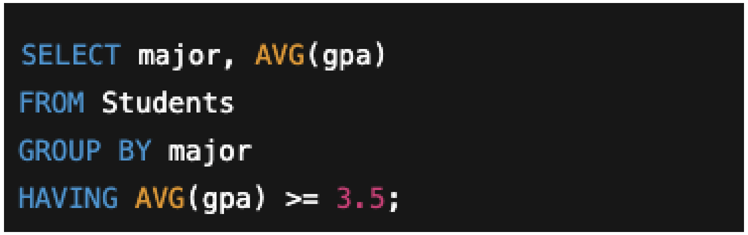


Result

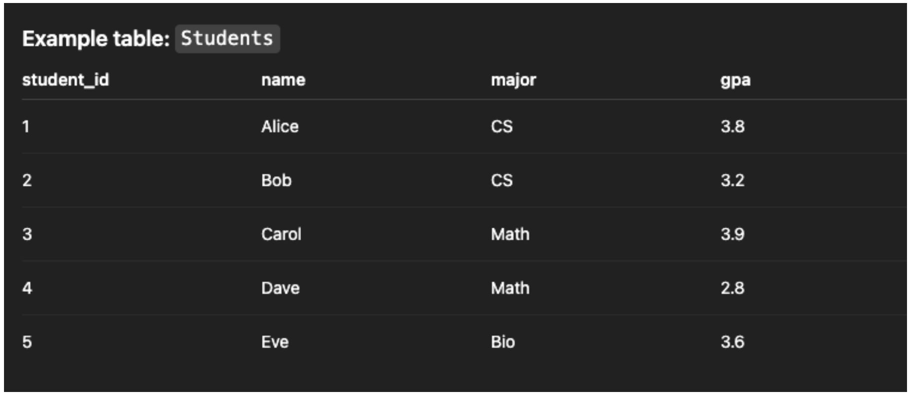
major	AVG(gpa)
CS	3.8
Math	3.9
Bio	3.6

HAVING vs WHERE

- HAVING filters groups

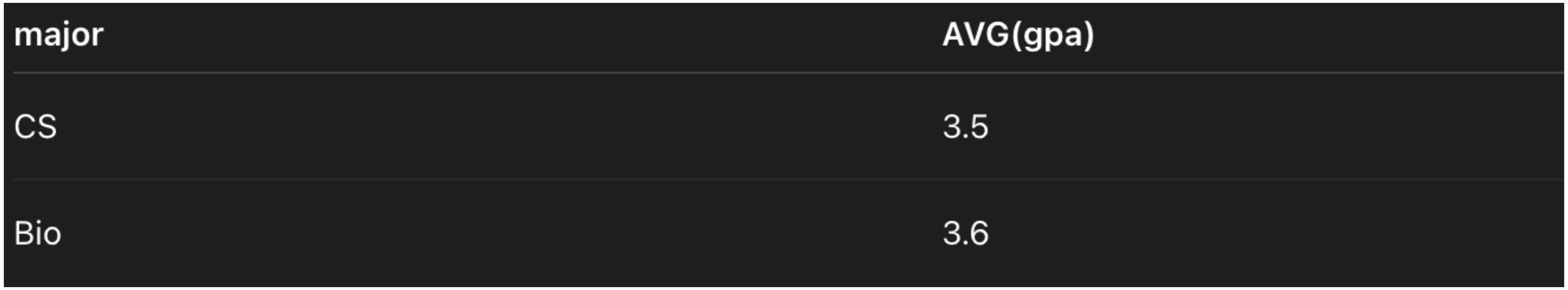


```sql
SELECT major, AVG(gpa)
FROM Students
GROUP BY major
HAVING AVG(gpa) >= 3.5;
```



Example table: `Students`

student_id	name	major	gpa
1	Alice	CS	3.8
2	Bob	CS	3.2
3	Carol	Math	3.9
4	Dave	Math	2.8
5	Eve	Bio	3.6



major	AVG(gpa)
CS	3.5
Bio	3.6

Summary

- Databases overview
- SQL basics
- Core queries