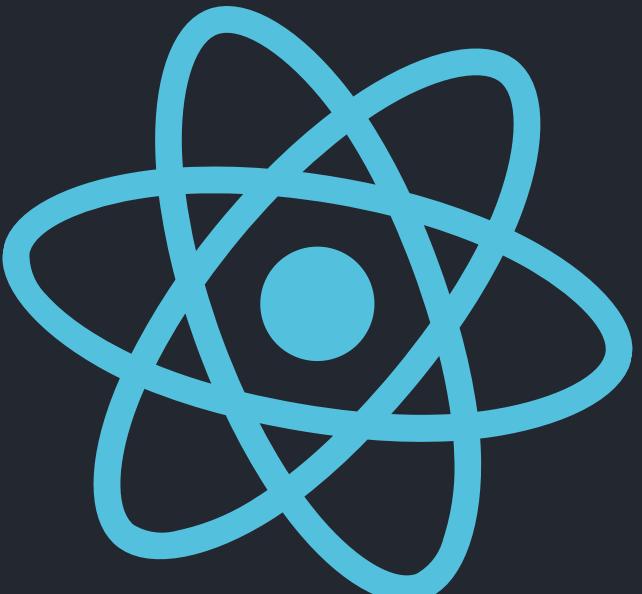


Testing React App With Vitest and RTL



Postgres

PostgreSQL is a powerful, open-source relational database management system.

It is known for its reliability, performance, and advanced features.

PostgreSQL supports a wide range of data types, including JSON and XML.

It also includes support for complex queries and functions.

PostgreSQL is a popular choice for web applications and enterprise systems.

It is available under the PostgreSQL License, which is a BSD-style license.

PostgreSQL is developed by the PostgreSQL Global Development Group.

It is currently in version 12, with many more versions planned for the future.

PostgreSQL is a great choice for any database needs you may have.

It is reliable, performant, and feature-rich.

Give PostgreSQL a try today and see what it can do for your application!

Postgres

**The World's Most Advanced Open Source Relational Database
Management System**

Postgres

1

Open Source

Postgres

1

Open Source

2

RDBMS

Postgres

1

Open Source

2

RDBMS

3

Modern

Postgres

1

Open Source

2

RDBMS

3

Modern

4

ACID Compliance

Postgres

1

Open Source

2

RDBMS

3

Modern

4

ACID Compliance

5

Advanced Data Types

Postgres

1

Open Source

2

RDBMS

3

Modern

4

ACID Compliance

5

Advanced Data Types

6

Scalability

Postgres

1

Open Source

2

RDBMS

3

Modern

4

ACID Compliance

5

Advanced Data Types

6

Scalability

7

Indexing

Postgres

1

Open Source

2

RDBMS

3

Modern

4

ACID Compliance

5

Advanced Data Types

6

Scalability

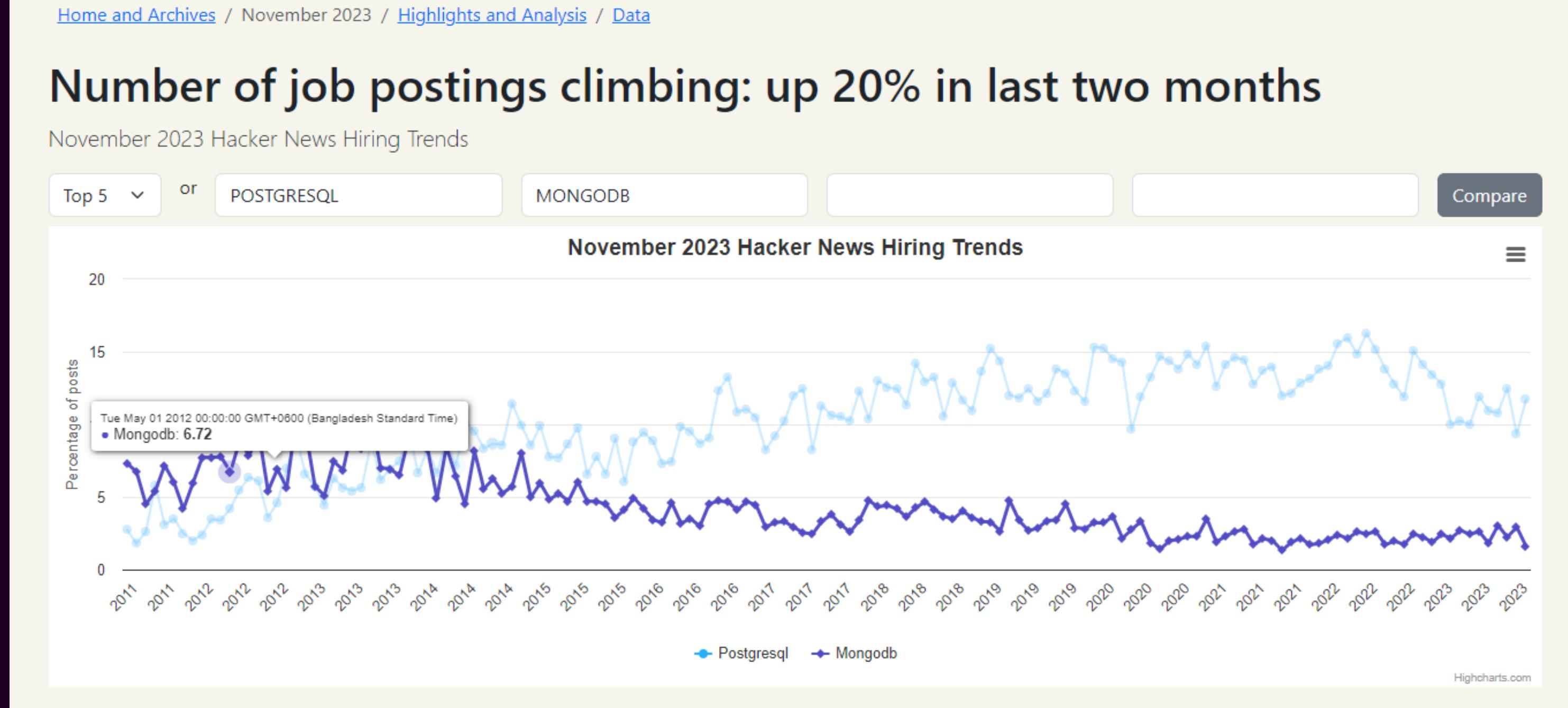
7

Indexing

8

Community Support

Postgres

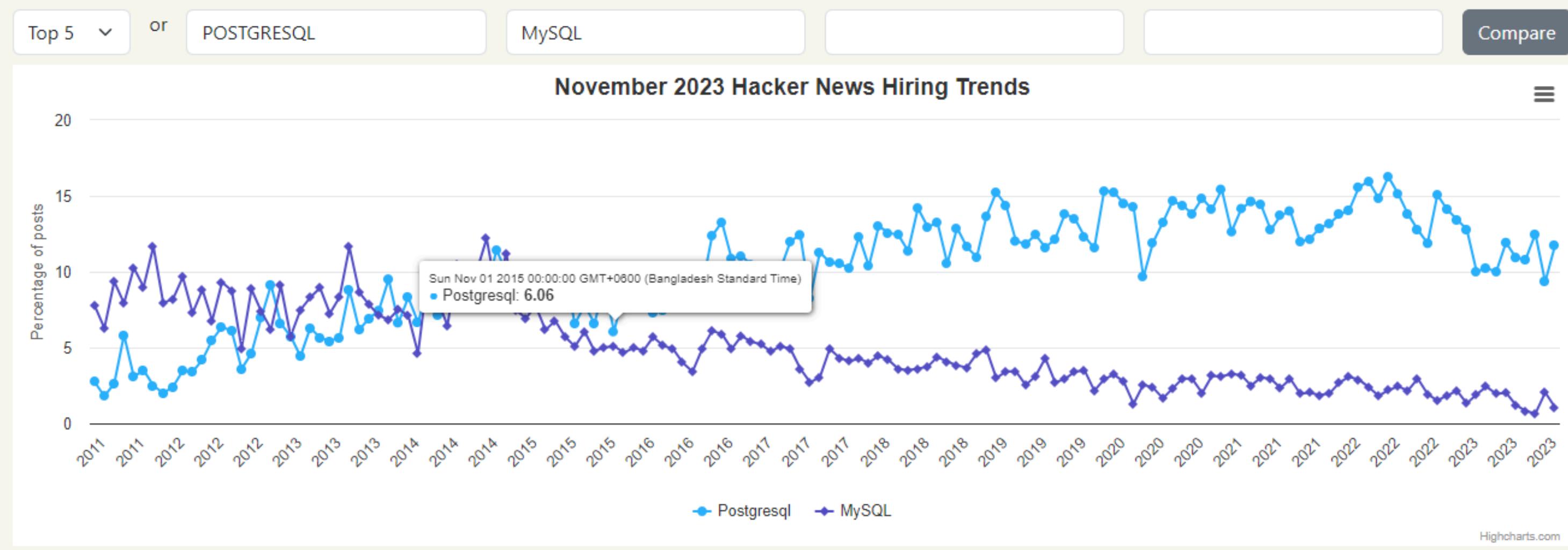


Postgres

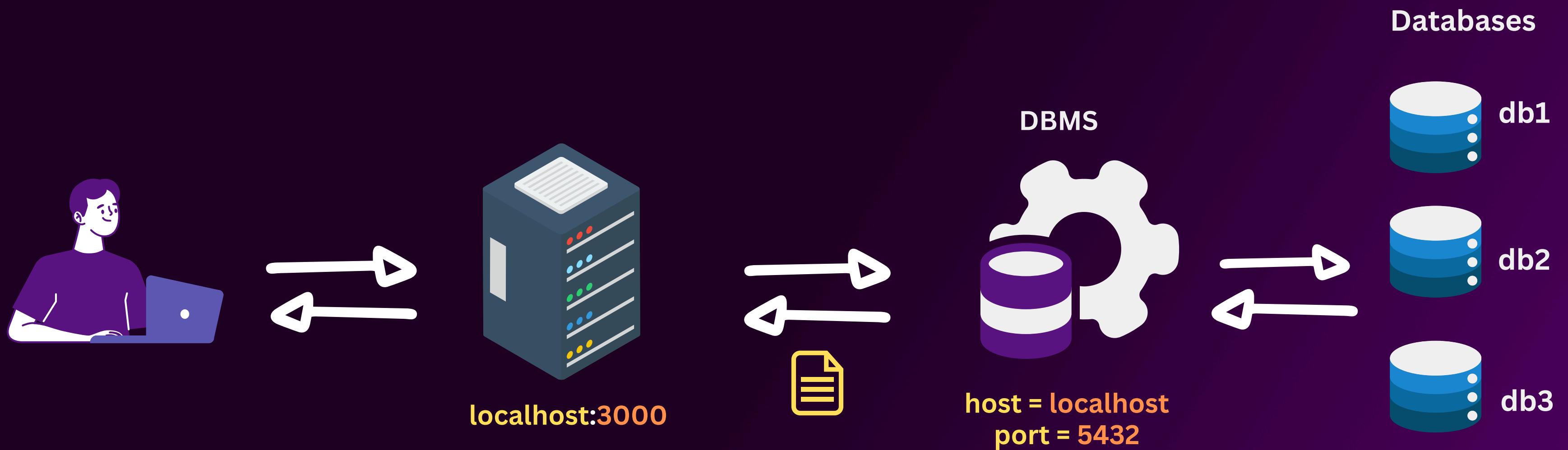
[Home and Archives](#) / November 2023 / [Highlights and Analysis](#) / [Data](#)

Number of job postings climbing: up 20% in last two months

November 2023 Hacker News Hiring Trends



Postgres



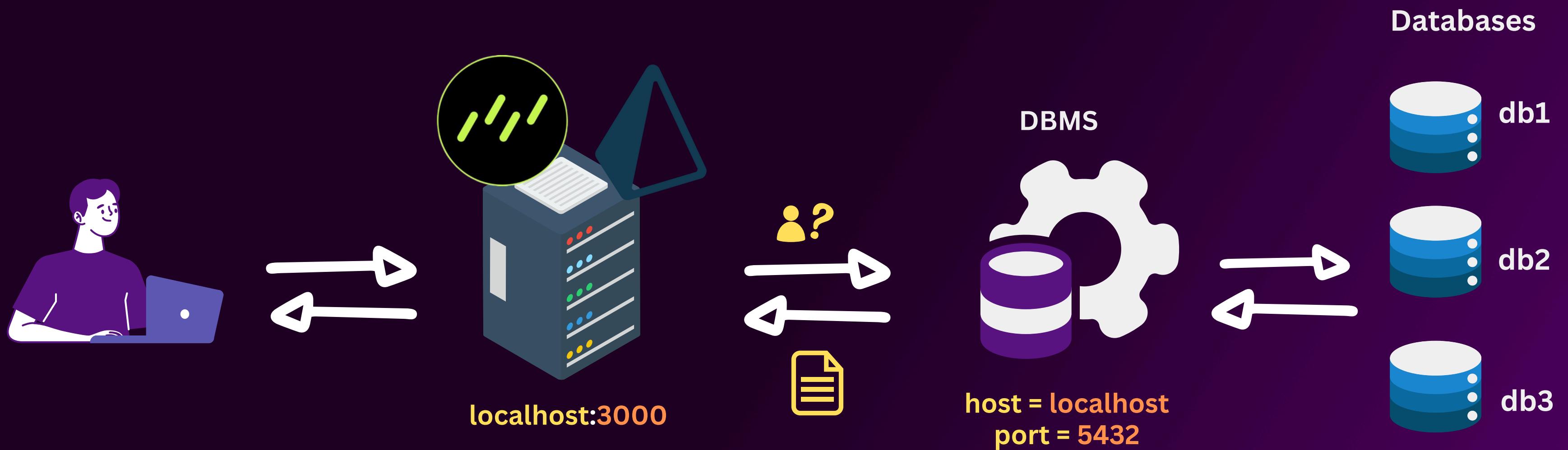
Postgres



Postgres



Postgres



SQL



SQL

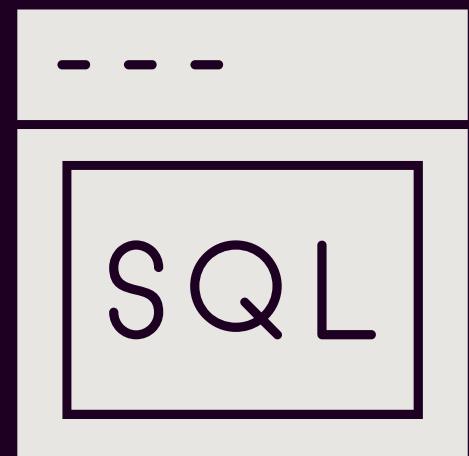
Structured Query Language

SQL

Structured Query Language Sequel

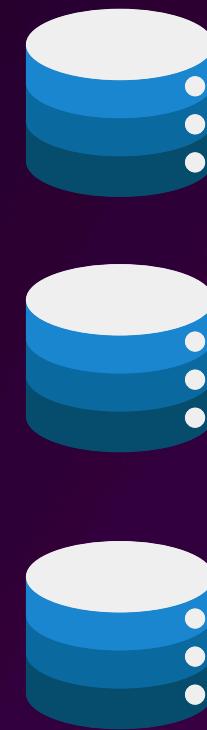
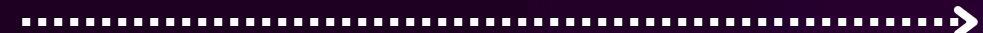
SQL

Structured Query Language Sequel



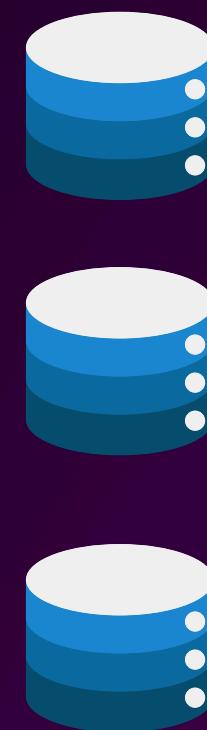
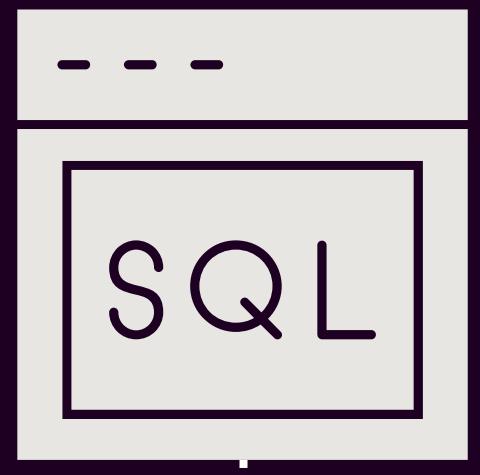
SQL

Structured Query Language Sequel



SQL

Structured Query Language Sequel

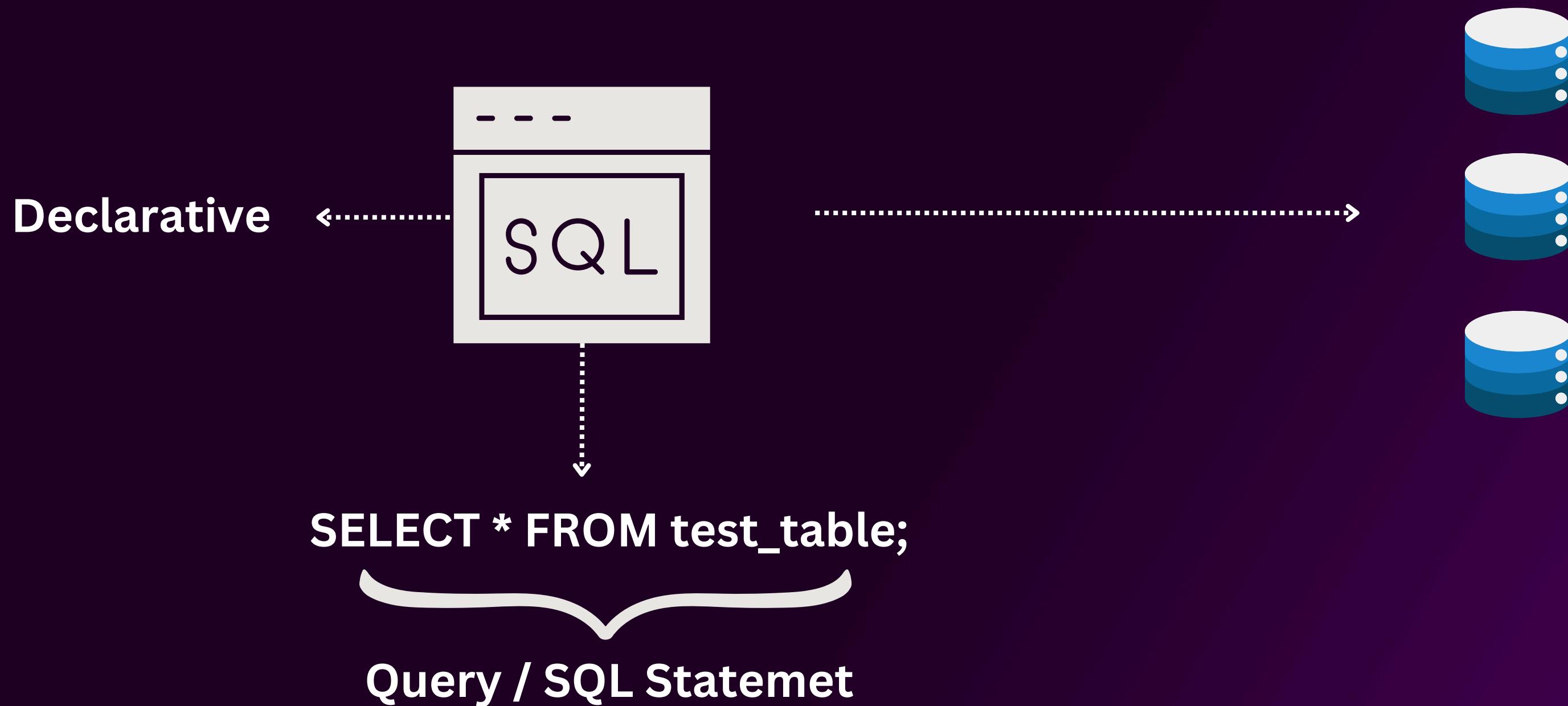


SELECT * FROM test_table;

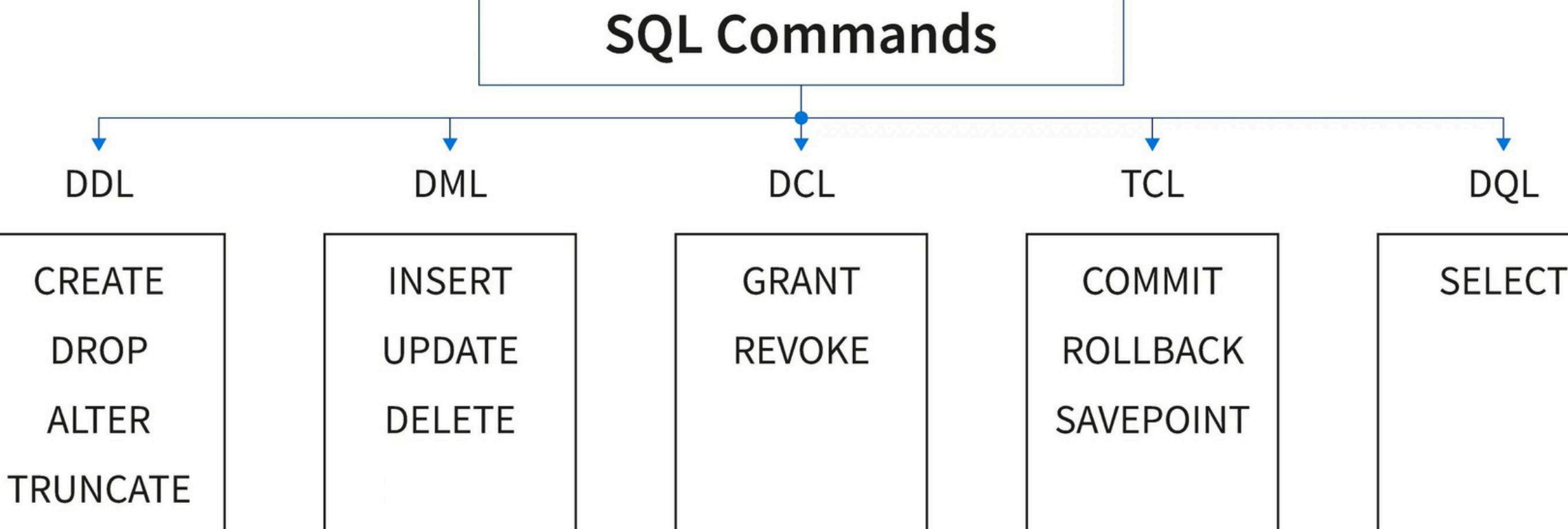
Query / SQL Statement

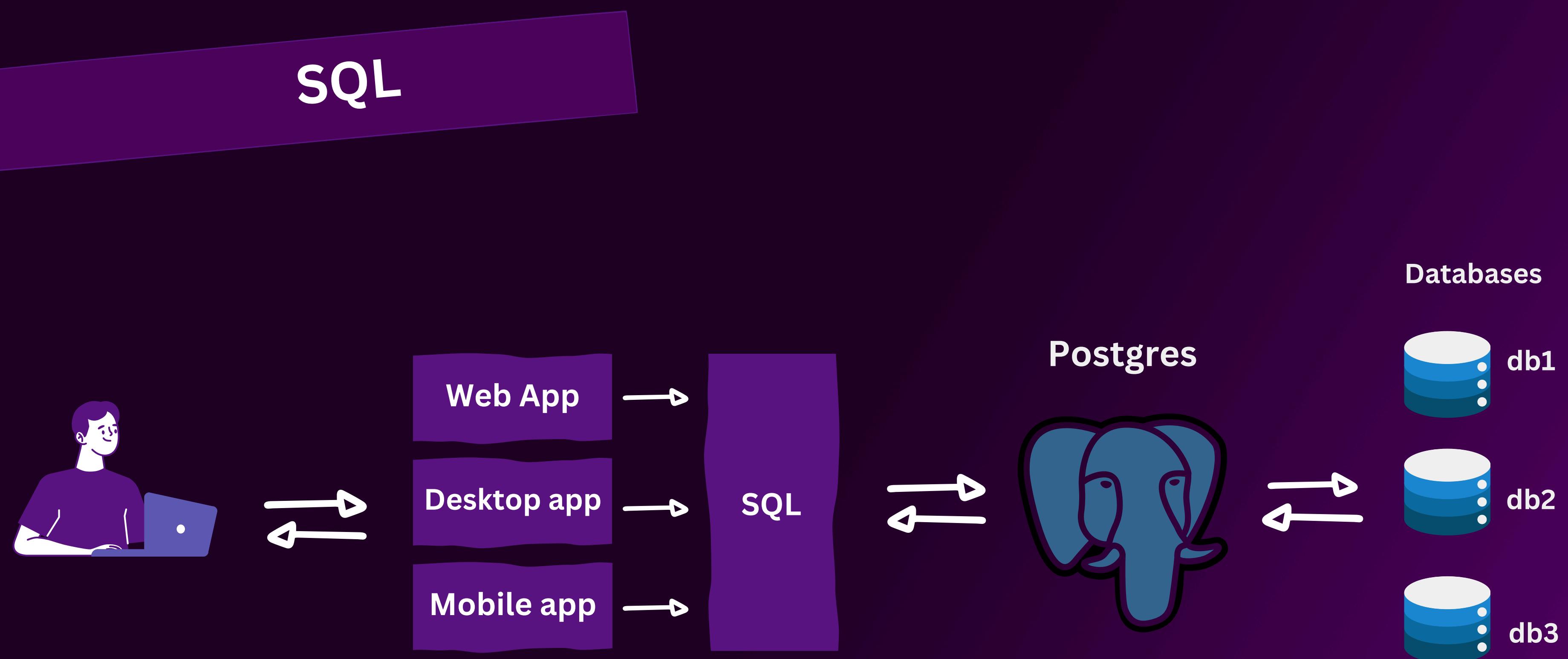
SQL

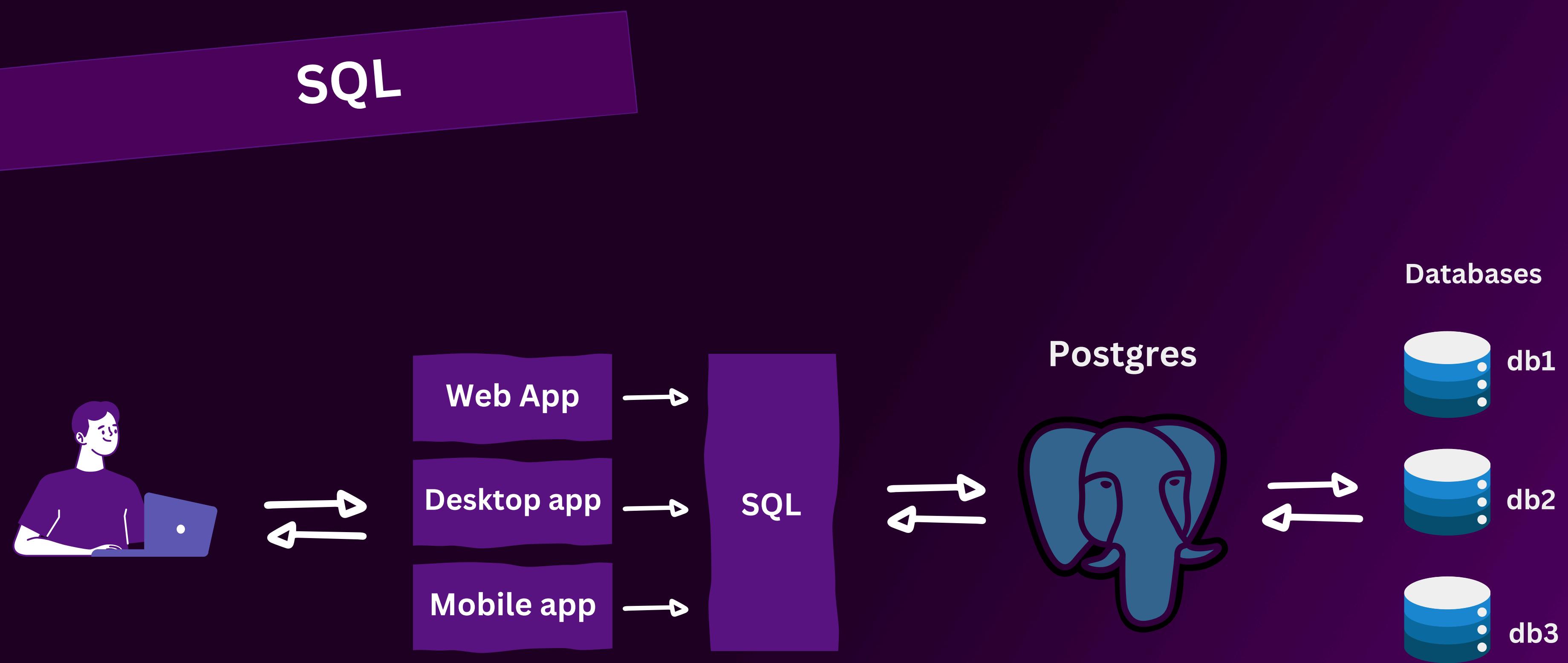
Structured Query Language Sequel



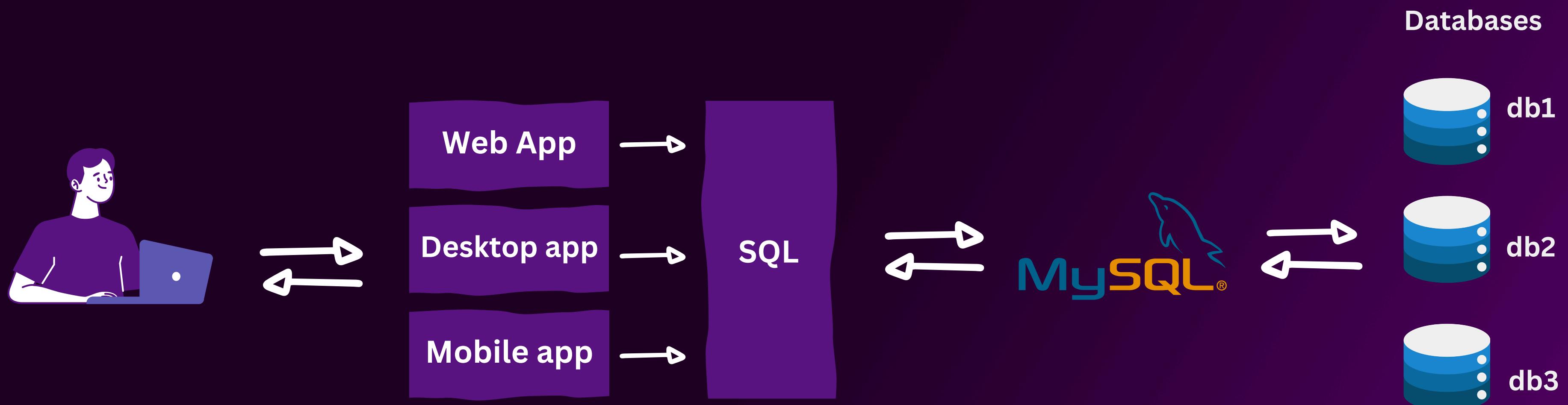
SQL



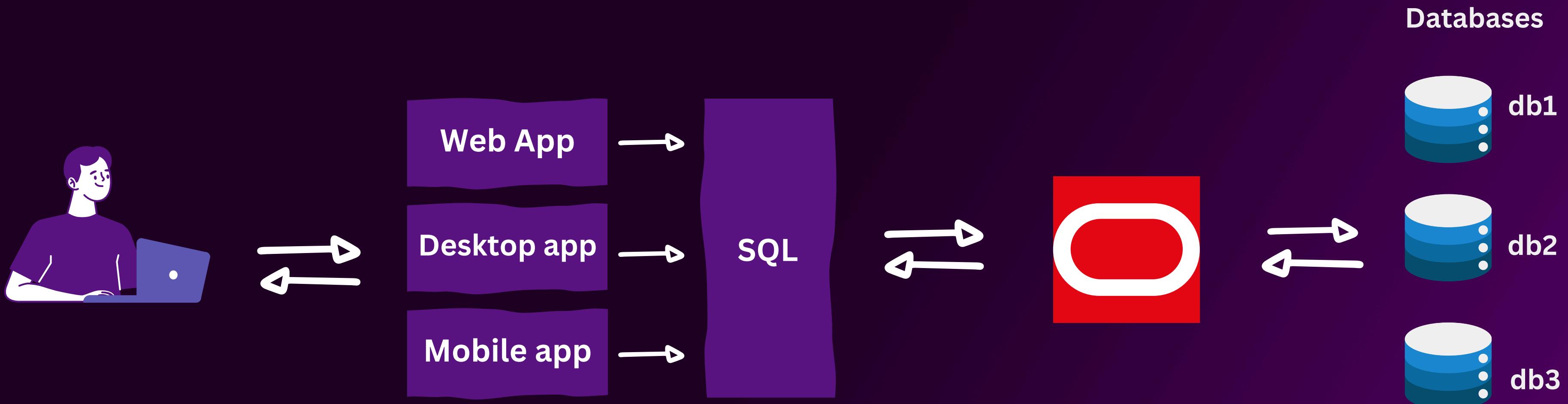




SQL



SQL



Data Types

id (SERIAL)	employee_id (INTEGER)	name (VARCHAR(50))
1	4560	John
2	8962	Doe

Data Types

id (SERIAL)	employee_id (INTEGER)	name (VARCHAR(50))	dob (DATE)	is_active (BOOLEAN)	salary (NUMERIC(10, 2))
1	4560	John	1990-05-15	true	65000.00
2	8962	Doe	1985-08-22	false	45000.00

Data Consistency

Performance

Constraint Enforcement

Convenience and Functionality

Storage Efficiency

Data Types

Boolean	Numbers	Binary
Date/Time	Json	Character
UUID	Array	XML

Data Types

Boolean	Numbers	Binary
Date/Time	Json	Character
UUID	Array	XML

Data Types

Boolean

true false null

Data Types

Integers

- INT:
 - Range: -2,147,483,648 to 2,147,483,647
 - Storage: 4 bytes
 - The INT data type is commonly used for integer values within the standard range.
- BIGINT:
 - Range: -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
 - Storage: 8 bytes
 - BIGINT provides a larger range than INT and is suitable for storing very large integer values.
- SMALLINT:
 - Range: -32,768 to 32,767
 - Storage: 2 bytes
 - SMALLINT is suitable for storing small integer values within a more limited range.
- FLOAT4 (Single Precision):
 - Storage: 4 bytes
 - 6 digit precision
- FLOAT8 (Double Precision):
 - 15 digit precision
 - Storage: 8 bytes
- NUMERIC(precision, scale):
 - Range: Unlimited precision
 - Storage: Variable (depends on the input precision and scale)
 - ex: NUMERIC(4,3)
- Serial:
 - Range: Like integers
 - Auto-incrementing integer

Data Types

Characters

- **CHAR:**
 - **CHAR** is a **fixed-length character type**.
 - **Storage:** The length is fixed and specified when defining the column.
 - If the actual string is shorter than the specified length, it is padded with spaces.
 - **Example:** CHAR(10) means a fixed-length string of 10 characters.
- **VARCHAR:**
 - **VARCHAR** is a **variable-length character type**.
 - **Storage:** The length is not fixed and can vary up to the specified maximum length.
 - No padding with spaces is done, so it is more storage-efficient when dealing with variable-length strings.
 - **Example:** VARCHAR(255) means a variable-length string with a maximum length of 255 characters.
- **TEXT:**
 - **TEXT** is also a **variable-length character type**.
 - **Storage:** Similar to VARCHAR, it is not fixed in length.
 - Typically used for longer text strings where the exact length is not known or can vary widely.
 - **Example:** TEXT is often used for columns containing large amounts of text.

Data Types

Date

20-Dec-1980

1980-12-20

Dec-20-1980

TIME

HH:MM:SS

TIME HH:MM:SS

YYYY-MM-DD HH:MI:SS

TIMESTAMPTZ (TIMESTAMP WITH TIME ZONE)

YYYY-MM-DD HH:MI:SS:00-08

Data Types

UUID

- **UUID (Universally Unique Identifier):**
 - **Definition:** UUID is a data type in PostgreSQL used to store universally unique identifiers.
 - **Storage:** It is a 128-bit value, typically displayed as 32 hexadecimal characters separated by hyphens.
 - **ex:** 3c0ab34f-51f4-4d7b-84ee-b197af61dcb3

Create table

```
CREATE TABLE table_name
(
    column1 datatype constraint,
    column2 datatype constraint,
    column3 datatype constraint,
    ....
);
```

Column Constraints

Column Constraints

NOT NULL

```
CREATE TABLE example (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50) NOT NULL
);
```

Column Constraints

UNIQUE

```
CREATE TABLE example (
    id SERIAL PRIMARY KEY,
    email VARCHAR(255) UNIQUE
);
```

Column Constraints

PRIMARY KEY

```
CREATE TABLE example (
    id SERIAL PRIMARY KEY,
    username VARCHAR(50) UNIQUE
);
```

Column Constraints

Foreign Key

```
CREATE TABLE orders (
    order_id SERIAL PRIMARY KEY,
    customer_id INTEGER REFERENCES customers(customer_id)
);
```

Column Constraints

Foreign Key

```
CREATE TABLE orders (
    order_id SERIAL PRIMARY KEY,
    customer_id INTEGER REFERENCES customers(customer_id)
);
```

Product

prod_id	prod_title
1	shoe
2	t-shirt

Order

order_id	prod_id
1	1
2	2

Column Constraints

DEFAULT

```
CREATE TABLE example (
    id SERIAL PRIMARY KEY,
    status BOOLEAN DEFAULT true
);
```

Column Constraints

CHECK

```
CREATE TABLE example (
    id SERIAL PRIMARY KEY,
    age INTEGER CHECK (age >= 18)
);
```

Column Constraints

```
CREATE TABLE person (
    person_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    is_active BOOLEAN DEFAULT true,
    age INTEGER CHECK (age >= 0),
    email VARCHAR(255) UNIQUE
);
```

Insert

Single-Row Insert:

Insert

Single-Row Insert:

```
INSERT INTO your_table (column1, column2, column3)  
VALUES (value1, value2, value3);
```

Insert

Single-Row Insert:

```
INSERT INTO your_table (column1, column2, column3)
VALUES (value1, value2, value3);
```

```
INSERT INTO employees (first_name, last_name, hire_date)
VALUES ('John', 'Doe', '2022-01-15');
```

Insert

Multi-Row Insert:

```
INSERT INTO your_table (column1, column2, column3)
VALUES
    (value1_1, value2_1, value3_1),
    (value1_2, value2_2, value3_2),
    ...;
```

Insert

Multi-Row Insert:

```
INSERT INTO your_table (column1, column2, column3)
VALUES
    (value1_1, value2_1, value3_1),
    (value1_2, value2_2, value3_2),
    ...;
```

```
INSERT INTO employees (first_name, last_name, hire_date)
VALUES
    ('Jane', 'Smith', '2022-02-20'),
    ('Bob', 'Johnson', '2022-03-25');
```

Insert

```
INSERT INTO your_table  
VALUES (value1, value2, value3, ...);
```

```
-- Assuming the table has columns (id, name, age)  
INSERT INTO persons  
VALUES (1, 'John Doe', 25);
```

Alter

100

Alter

id (SERIAL)	employee_id (INTEGER)	name (VARCHAR(50))
1	4560	John
2	8962	Doe

Alter

id (SERIAL)	employee_id (INTEGER)	name (VARCHAR(50))
1	4560	John
2	8962	Doe

id (SERIAL)	employee_id (INTEGER)	name (VARCHAR(50))	dob (DATE)
1	4560	John	1990-05-15
2	8962	Doe	1985-08-22

Alter

id (SERIAL)	employee_id (INTEGER)	name (VARCHAR(50))
1	4560	John
2	8962	Doe

id (SERIAL)	employee_id (INTEGER)	name (VARCHAR(50))	dob (DATE)
1	4560	John	1990-05-15
2	8962	Doe	1985-08-22

id (SERIAL)	employee_id (INTEGER)	name (VARCHAR(25))	dob (DATE)
1	4560	John	1990-05-15
2	8962	Doe	1985-08-22

Alter

```
ALTER TABLE table_name  
action;
```

Alter

```
ALTER TABLE table_name  
action;
```

1

Rename a table

Alter

```
ALTER TABLE table_name  
action;
```

1

Rename a table

2

Modify data type of a column

Alter

```
ALTER TABLE table_name  
action;
```

1

Rename a table

2

Modify data type of a column

3

Add/Drop column

Alter

```
ALTER TABLE table_name  
action;
```

1

Rename a table

2

Modify data type of a column

3

Add/Drop column

4

Setting default value for a column

Alter

**ALTER TABLE table_name
action;**

1

Rename a table

2

Modify data type of a column

3

Add/Drop column

4

Setting default value for a column

5

Rename a column

Alter

**ALTER TABLE table_name
action;**

1

Rename a table

2

Modify data type of a column

3

Add/Drop column

4

Setting default value for a column

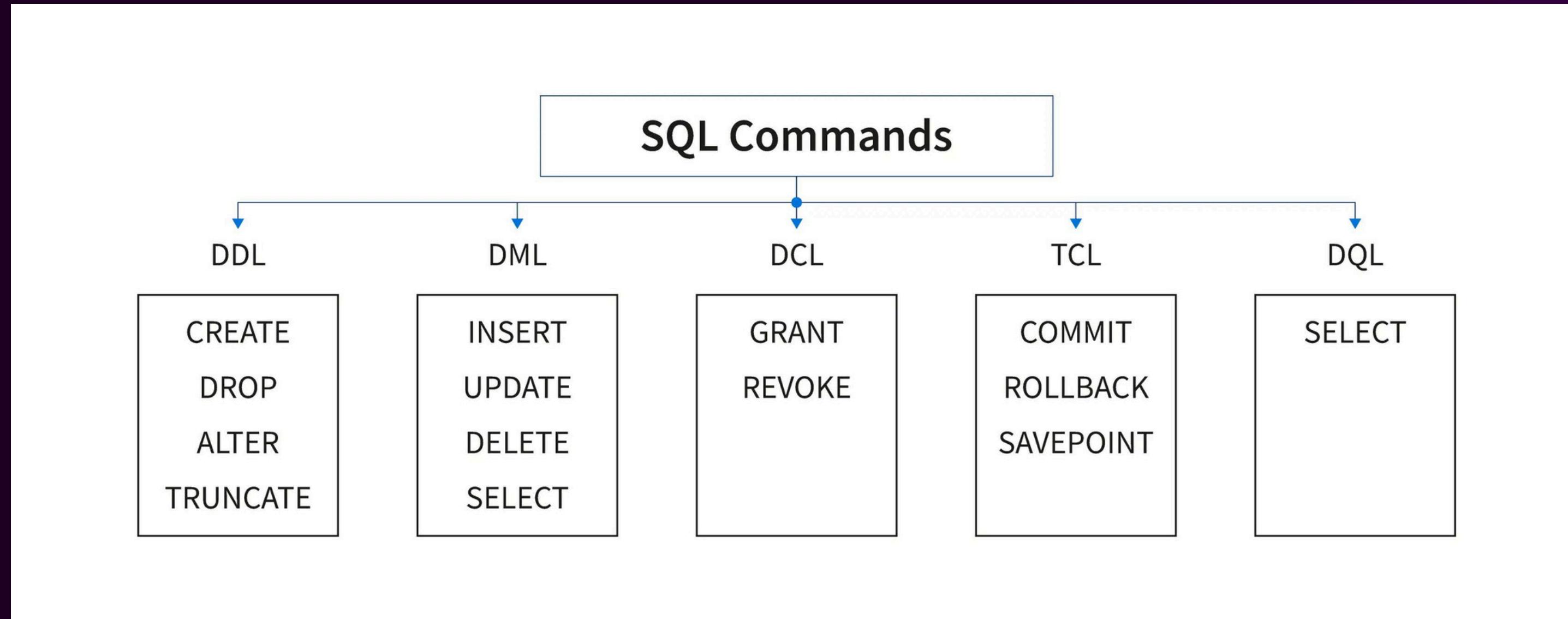
5

Rename a column

5

Add/Drop constraint for a column

SQL



Select

The **SELECT** statement is used to retrieve data from one or more tables and can be customized with conditions, sorting, and other clauses.

Select

The **SELECT** statement is used to retrieve data from one or more tables and can be customized with **conditions, sorting, and other clauses.**

Select

The **SELECT** statement is used to retrieve data from one or more tables and can be customized with **conditions, sorting, and other clauses**.

- **SELECT**: Retrieves data from one or more tables.
- **FROM**: Specifies the table from which to retrieve data.
- **WHERE**: Filters data based on specified conditions.
- **ORDER BY**: Sorts the result set based on specified columns.
- **GROUP BY**: Groups rows that have the same values in specified columns.
- **HAVING**: Filters the results of a GROUP BY clause based on specified conditions.
- **JOIN**: Combines rows from two or more tables based on a related column.
- **DISTINCT**: Returns unique values in the result set.
- **LIMIT**: Specifies the maximum number of rows to return.
- **OFFSET**: Specifies the number of rows to skip before starting to return rows.

Select

```
SELECT old_column_name AS 'new_column_name  
FROM your_table;
```

Functions

1. Scalar
2. Aggregate

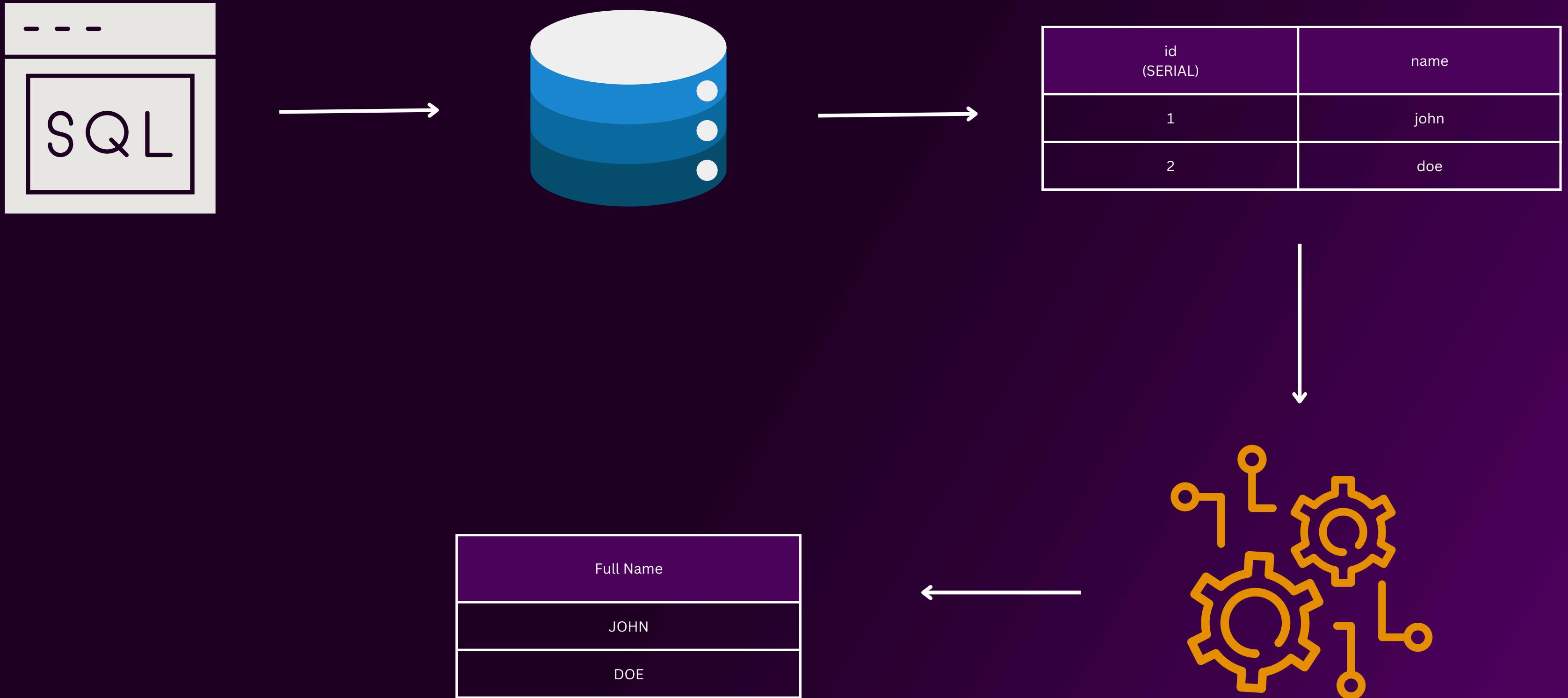
Scalar

Scalar functions operate on a single value and return a single value. They perform an operation on each row's data independently

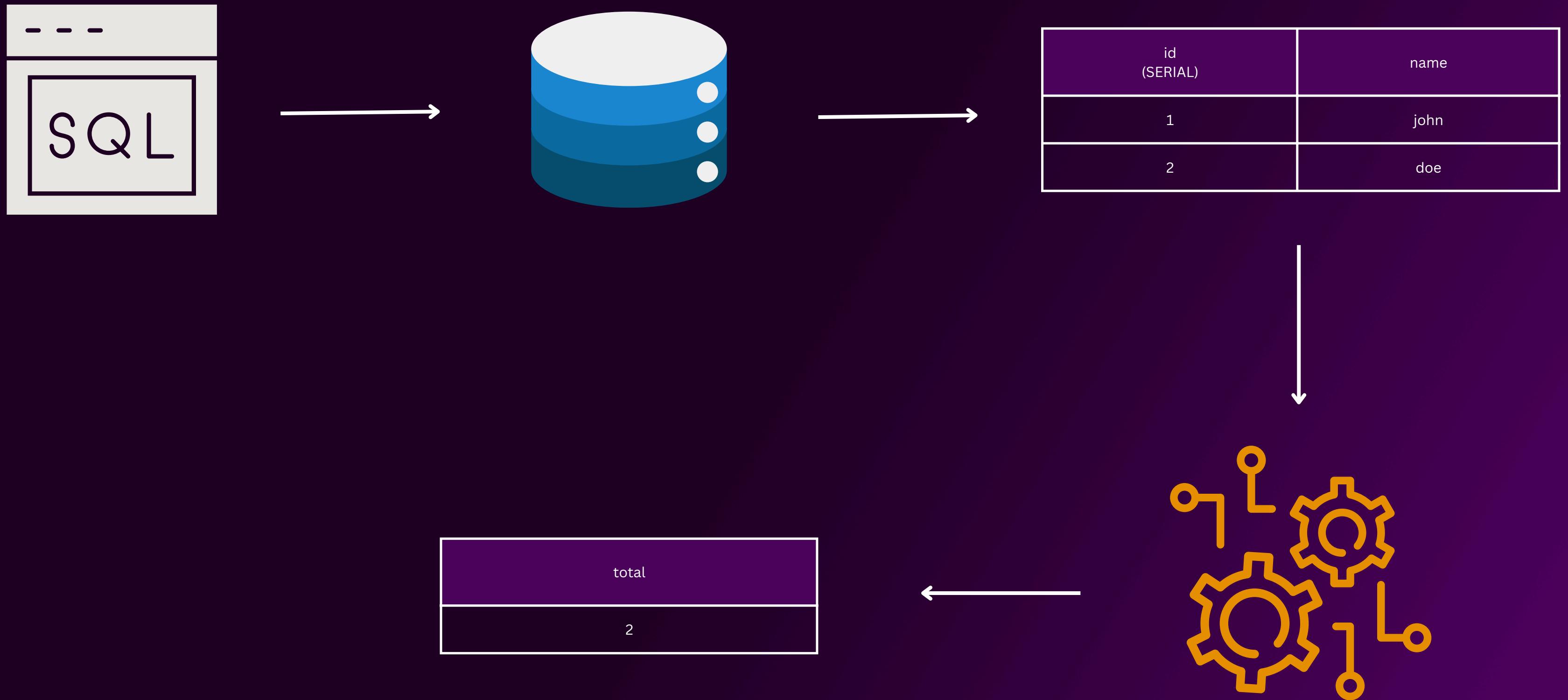
Aggregate

Aggregate functions operate on a set of values and return a single value summarizing that set. They perform an operation across multiple rows, often used with the GROUP BY clause

Scalar

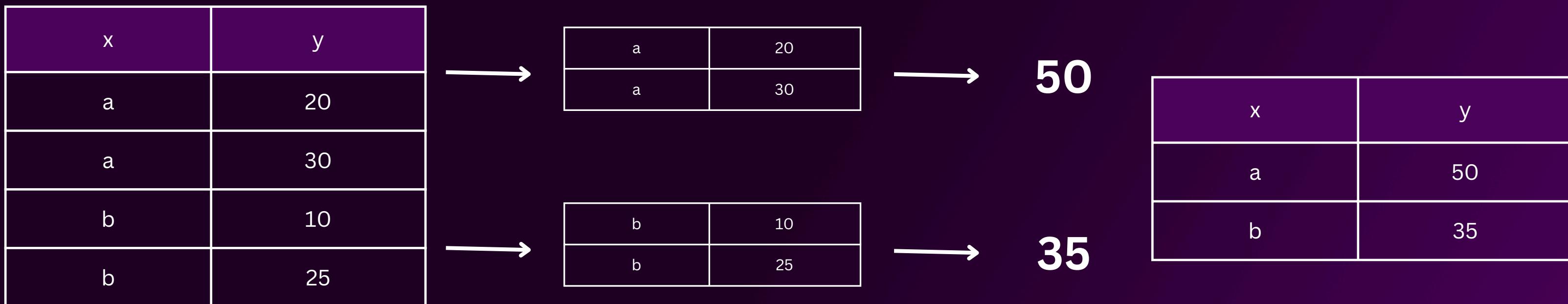


Aggregate



Group By

Split, Apply, Combine



Foreign Key

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥦	1
3	Exploring adventures with Sagor.⭐️	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Primary Key

Foreign Key

Uniquely identifies row to another table which is connected to the row of this table

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Primary Key

Foreign Key

```
CREATE TABLE post(  
    id SERIAL PRIMARY KEY,  
    title TEXT,  
    user_id INTEGER REFERENCES user(id)  
);
```

Foreign Key

```
CREATE TABLE post(  
    id SERIAL PRIMARY KEY,  
    title TEXT,  
    user_id INTEGER REFERENCES user(id)  
);
```

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥦	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Primary Key

Foreign Key

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Primary Key

Foreign Key

```
CREATE TABLE post(  
    id SERIAL PRIMARY KEY,  
    title TEXT,  
    user_id INTEGER REFERENCES user(id)  
);
```

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥗	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Primary Key

Foreign Key

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Primary Key

Foreign Key

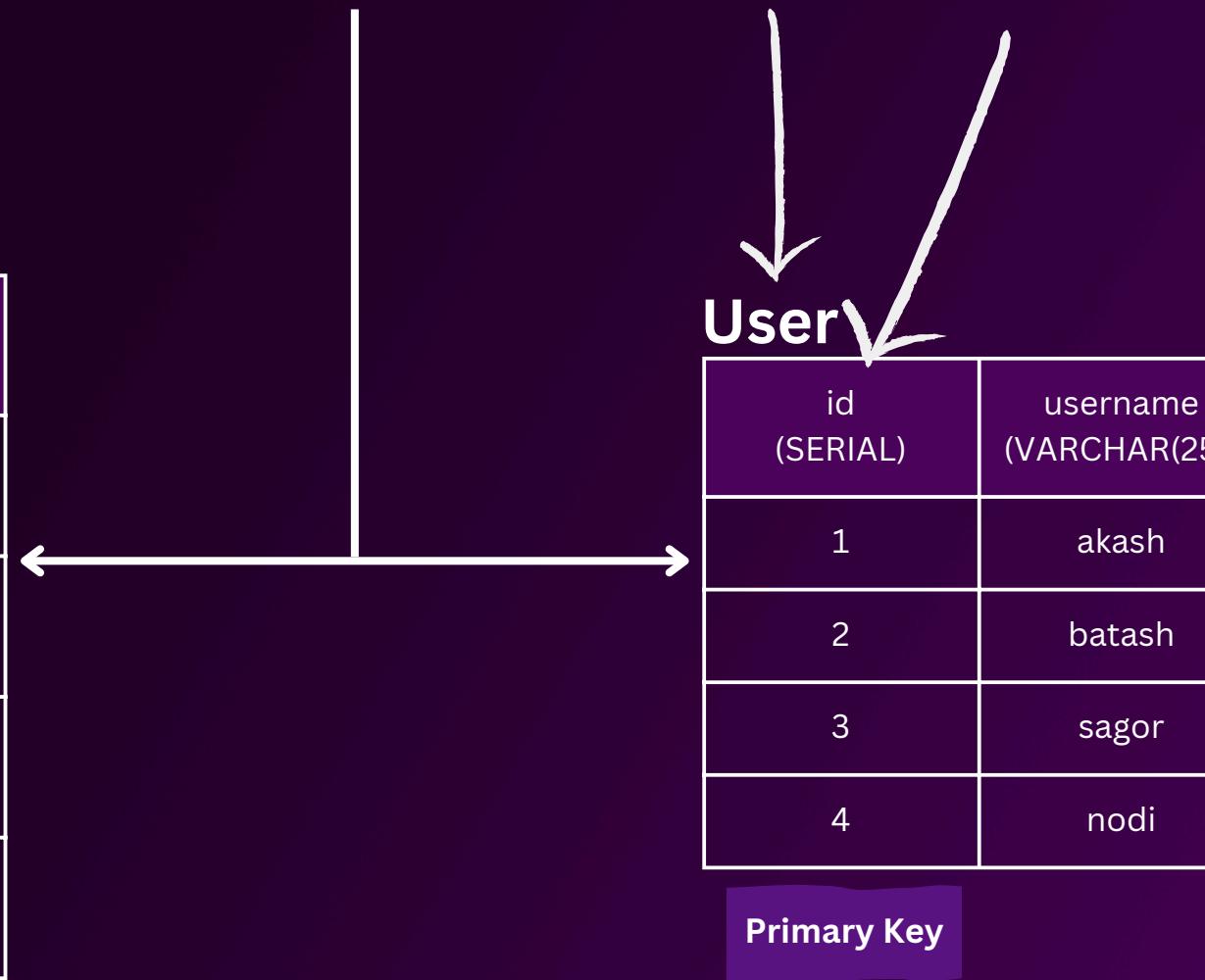
```
CREATE TABLE post(  
    id SERIAL PRIMARY KEY,  
    title TEXT,  
    user_id INTEGER REFERENCES user(id)  
);
```

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥗	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Primary Key

Foreign Key



Foreign Key

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥪	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Primary Key

Foreign Key

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Primary Key

-- Retrieve all posts for user with ID 4

Foreign Key

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥤	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Primary Key

Foreign Key

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Primary Key

-- Retrieve all posts for user with ID 4

```
SELECT * FROM post  
WHERE user_id = 4;
```

3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Foreign Key

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥪	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Primary Key

Foreign Key

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Primary Key

-- Retrieve all posts for user with ID 4

```
SELECT * FROM post  
WHERE user_id = 4;
```

3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

-- Retrieve all posts title with their username

Foreign Key

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🍲	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Primary Key

Foreign Key

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Primary Key

-- Retrieve all posts for user with ID 4

```
SELECT * FROM post  
WHERE user_id = 4;
```

3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

-- Retrieve all posts title with their username

title (TEXT)	username
Enjoying a sunny day with Akash! ☀️	batash
Batash just shared an amazing recipe! 🍲	akash
Exploring adventures with Sagor.🌟	nodi
Nodi's wisdom always leaves me inspired. 📚	nodi

Foreign Key

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥪	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Primary Key

Foreign Key

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Primary Key

-- Retrieve all posts for user with ID 4

```
SELECT * FROM post  
WHERE user_id = 4;
```

3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

-- Retrieve all posts title with their username



title (TEXT)	username
Enjoying a sunny day with Akash! ☀️	batash
Batash just shared an amazing recipe! 🥪	akash
Exploring adventures with Sagor.🌟	nodi
Nodi's wisdom always leaves me inspired. 📚	nodi

Foreign Key

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Primary Key

Foreign Key

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Primary Key

Deletion

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥟	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Primary Key

Foreign Key

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Primary Key

Deletion

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥟	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

Primary Key

Foreign Key

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

-- Retrieve all posts title with their username



title (TEXT)	username
Enjoying a sunny day with Akash! ☀️	batash
Batash just shared an amazing recipe! 🥑	akash
Exploring adventures with Sagor.🌟	nodi
Nodi's wisdom always leaves me inspired. 📚	nodi

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Post & User

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	1	akash

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥪	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2		
2	Batash just shared an amazing recipe! 🥪	1		
3	Exploring adventures with Sagor.🌟	4		
4	Nodi's wisdom always leaves me inspired. 📚	4		

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥟	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥟	1		
3	Exploring adventures with Sagor.🌟	4		
4	Nodi's wisdom always leaves me inspired. 📚	4		

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥟	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥟	1	1	akash
3	Exploring adventures with Sagor.🌟	4		
4	Nodi's wisdom always leaves me inspired. 📚	4		

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥑	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

Post & User

	title (TEXT)			username (VARCHAR(25))
	Enjoying a sunny day with Akash! ☀️			batash
	Batash just shared an amazing recipe! 🥑			akash
	Exploring adventures with Sagor.🌟			nodi
	Nodi's wisdom always leaves me inspired. 📚			nodi

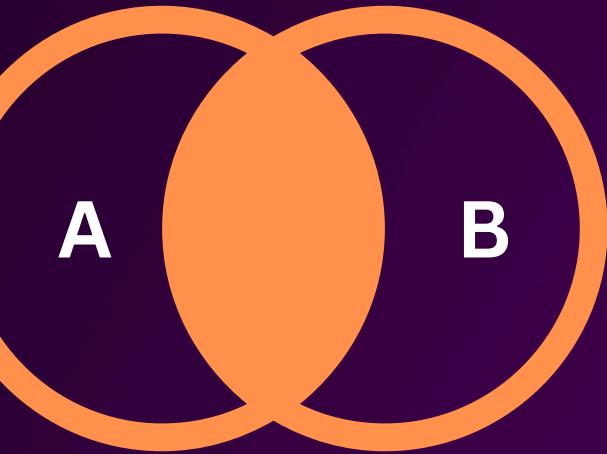
JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi



Inner Join

Post & User

	title (TEXT)			username (VARCHAR(25))
	Enjoying a sunny day with Akash! ☀️			batash
	Batash just shared an amazing recipe! 🥑			akash
	Exploring adventures with Sagor.🌟			nodi
	Nodi's wisdom always leaves me inspired. 📚			nodi

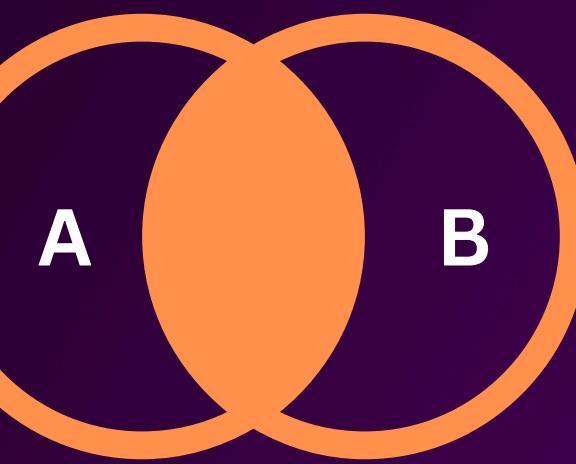
JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥪	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi



Inner Join

Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥪	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥪	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

A

B

Inner Join

Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥪	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

A

B

Inner Join

Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥑	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi
5	this is a test post title	NULL		

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

A

B

Inner Join

Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥑	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi
5	this is a test post title	NULL		

JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥪	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

A

B

Inner Join

Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥪	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi

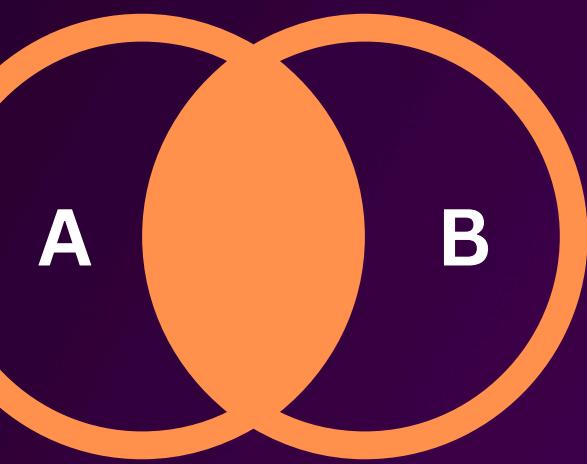
JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥪	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi



Inner Join

Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥪	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi

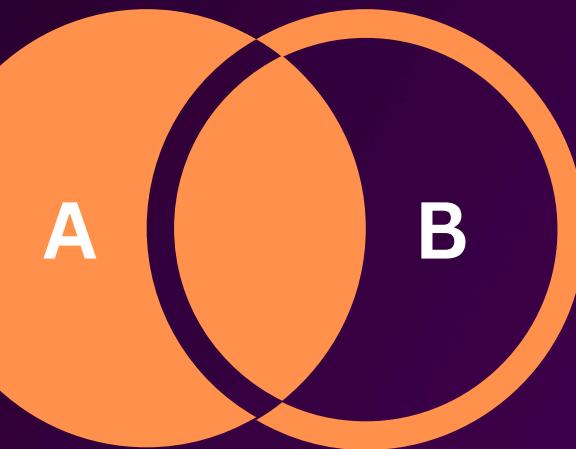
Left JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi



Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥑	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi
5	this is a test post title	NULL		

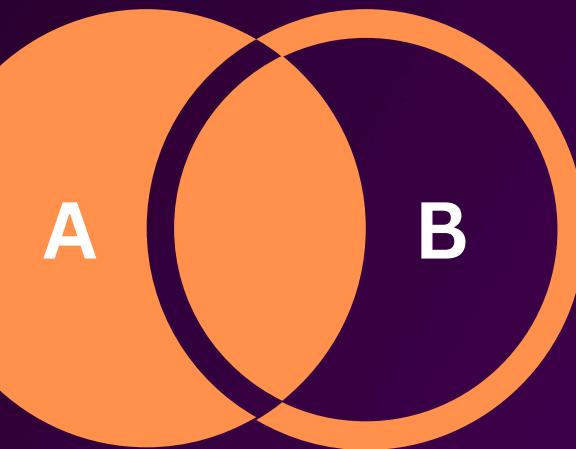
Left JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi



Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥑	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi
5	this is a test post title	NULL	?	?

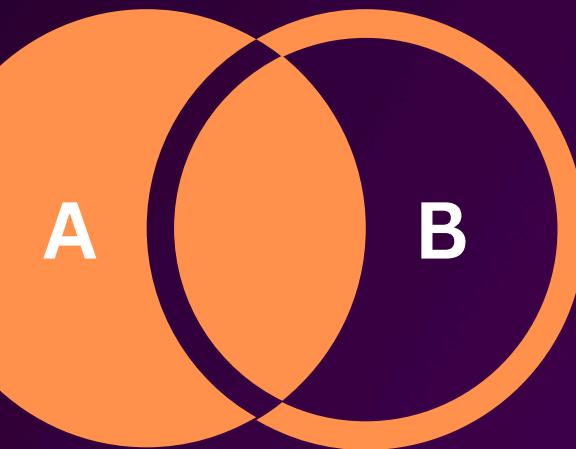
Left JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi



Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥑	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi
5	this is a test post title	NULL	NULL	NULL

Right JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi



Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥑	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi

Right JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥪	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi



Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥪	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi
			3	sagor

Right JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥪	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi



Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥪	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi
NULL	NULL	NULL	3	sagor

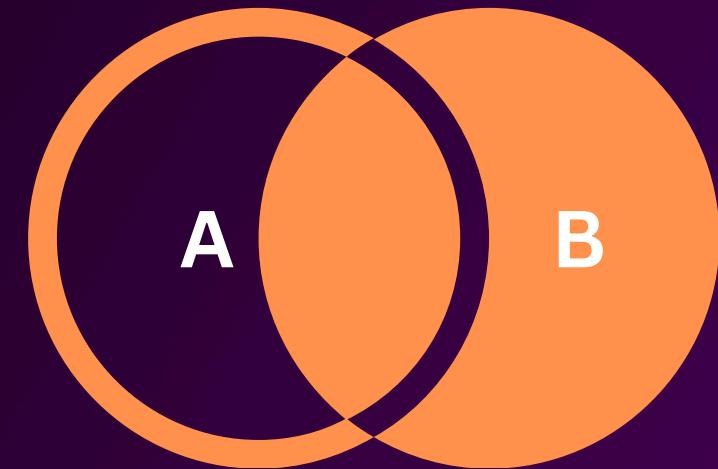
Right JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi



A

B

Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥑	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi
NULL	NULL	NULL	3	sagor

Right JOIN

Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥑	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

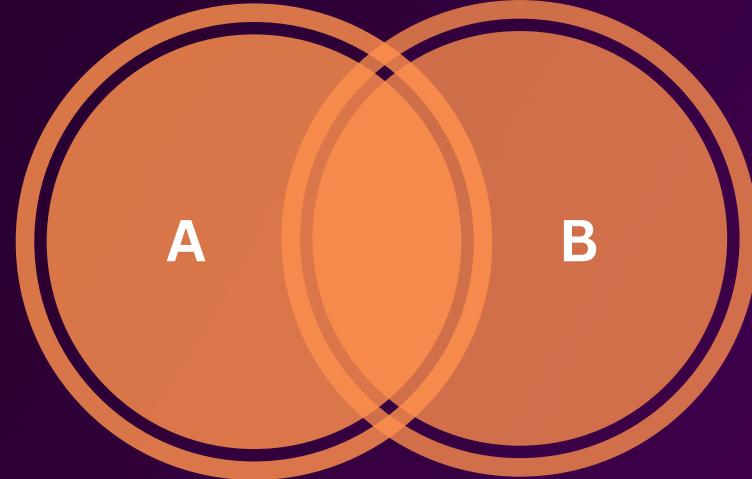


Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥑	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi
NULL	NULL	NULL	3	sagor

Full JOIN

Full Join



Post

id (SERIAL)	title (TEXT)	user_id
1	Enjoying a sunny day with Akash! ☀️	2
2	Batash just shared an amazing recipe! 🥪	1
3	Exploring adventures with Sagor.🌟	4
4	Nodi's wisdom always leaves me inspired. 📚	4
5	this is a test post title	NULL

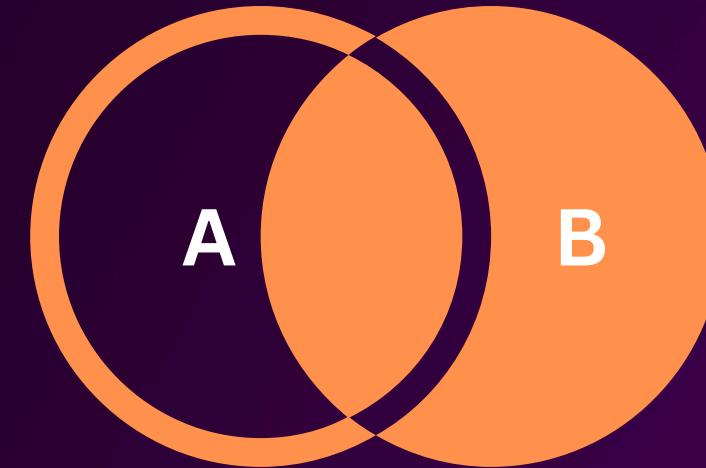
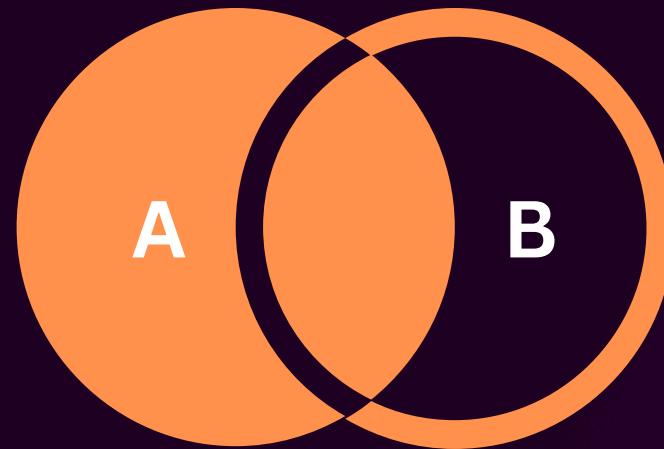
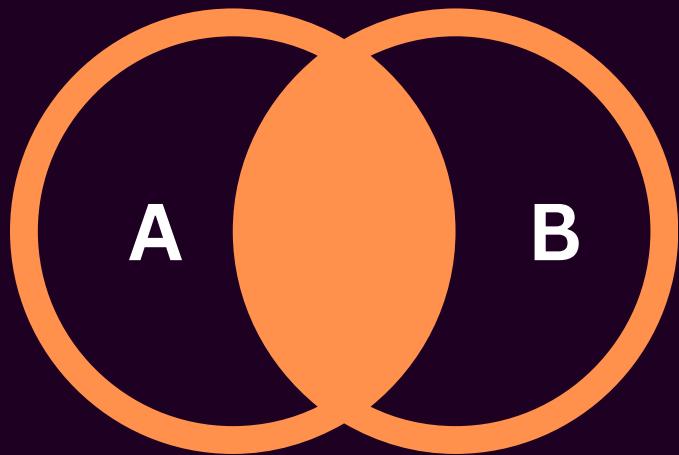
User

id (SERIAL)	username (VARCHAR(25))
1	akash
2	batash
3	sagor
4	nodi

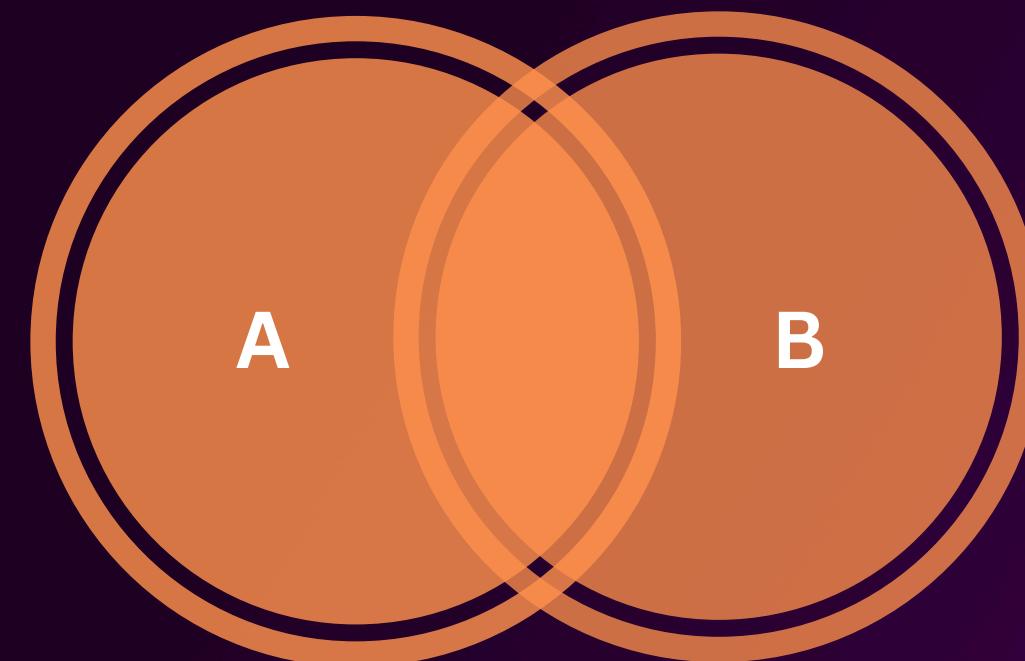
Post & User

id (SERIAL)	title (TEXT)	user_id	id (SERIAL)	username (VARCHAR(25))
1	Enjoying a sunny day with Akash! ☀️	2	2	batash
2	Batash just shared an amazing recipe! 🥪	1	1	akash
3	Exploring adventures with Sagor.🌟	4	4	nodi
4	Nodi's wisdom always leaves me inspired. 📚	4	4	nodi
5	this is a test post title	NULL	NULL	NULL
NULL	NULL	NULL	3	sagor

JOIN



Full Join



JOIN

emp_id	emp_name	dept_id
1	John Doe	101
2	Jane Smith	102

dept_id	dept_name
101	Human Resources
102	Marketing



emp_id	emp_name	dept_id	dept_id	dept_name
1	John Doe	101	101	Human Resources
2	Jane Smith	102	101	Human Resources
1	John Doe	101	102	Marketing
2	Jane Smith	102	102	Marketing

JOIN

emp_id	emp_name	dept_id
1	John Doe	101
2	Jane Smith	102

dept_id	dept_name
101	Human Resources
102	Marketing



emp_id	emp_name	dept_id	dept_id	dept_name
1	John Doe	101	101	Human Resources
2	Jane Smith	102	101	Human Resources
1	John Doe	101	102	Marketing
2	Jane Smith	102	102	Marketing

JOIN

emp_id	emp_name	dept_id
1	John Doe	101
2	Jane Smith	102

dept_id	dept_name
101	Human Resources
102	Marketing



emp_id	emp_name	dept_id	dept_id	dept_name
1	John Doe	101	101	Human Resources
2	Jane Smith	102	101	Human Resources
1	John Doe	101	102	Marketing
2	Jane Smith	102	102	Marketing

JOIN

emp_id	emp_name	dept_id
1	John Doe	101
2	Jane Smith	102

dept_id	dept_name
101	Human Resources
102	Marketing



emp_id	emp_name	dept_id	dept_id	dept_name
1	John Doe	101	101	Human Resources
2	Jane Smith	102	102	Marketing

JOIN

emp_id	emp_name	dept_id
1	John Doe	101
2	Jane Smith	102

dept_id	dept_name
101	Human Resources
102	Marketing



emp_id	emp_name	dept_id	dept_id	dept_name
1	John Doe	101	101	Human Resources
2	Jane Smith	102	102	Marketing

JOIN

emp_id	emp_name	dept_id
1	John Doe	101
2	Jane Smith	102

dept_id	dept_name
101	Human Resources
102	Marketing



emp_id	emp_name	dept_id	dept_name
1	John Doe	101	Human Resources
2	Jane Smith	102	Marketing

JOIN

emp_id	emp_name	dept_id
1	John Doe	101
2	Jane Smith	102

dept_id	dept_name
101	Human Resources
102	Marketing



dept_id	emp_id	emp_name	dept_name
101	1	John Doe	Human Resources
102	2	Jane Smith	Marketing

