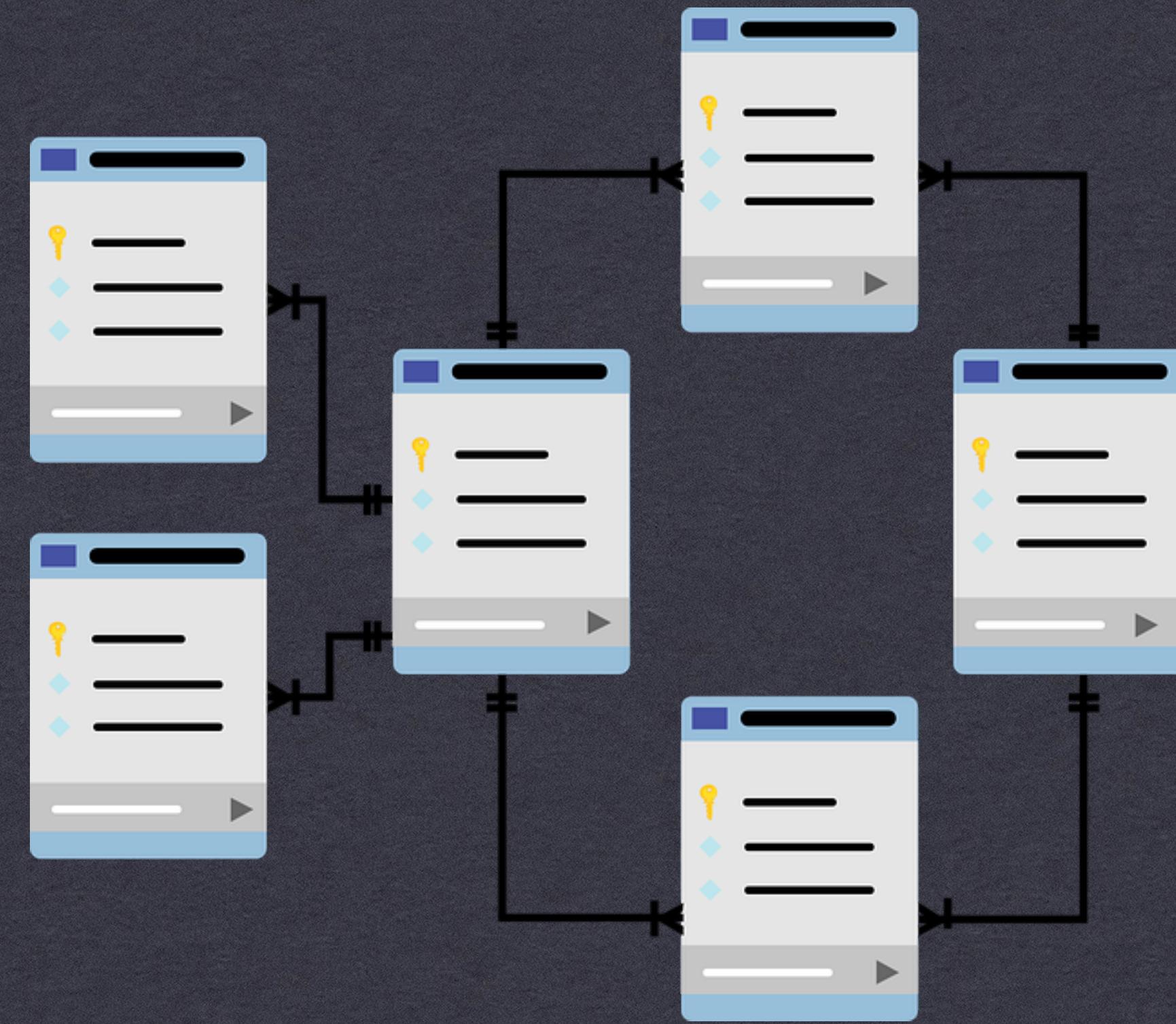


# SUMMARIZER #38

## RELATIONAL DATABASES

LEONARDO HUGENS



**Product table**

Product ID	Product name	Price per kg
389	Banana	4
789	Apple	5
879	Watermelon	5
975	Mango	7

**Transaction table**

Transaction ID	Customer ID	Product ID	Purche date
53666	24221	389	06-02-2023
50333	24222	789	06-02-2023
54673	24223	879	06-02-2023
58930	24224	975	06-02-2023

**Customer table**

Customer ID	Last name	First name
24221	Smith	James
24222	Jones	Sam
24223	Taylor	Ann
24224	Burton	Sue

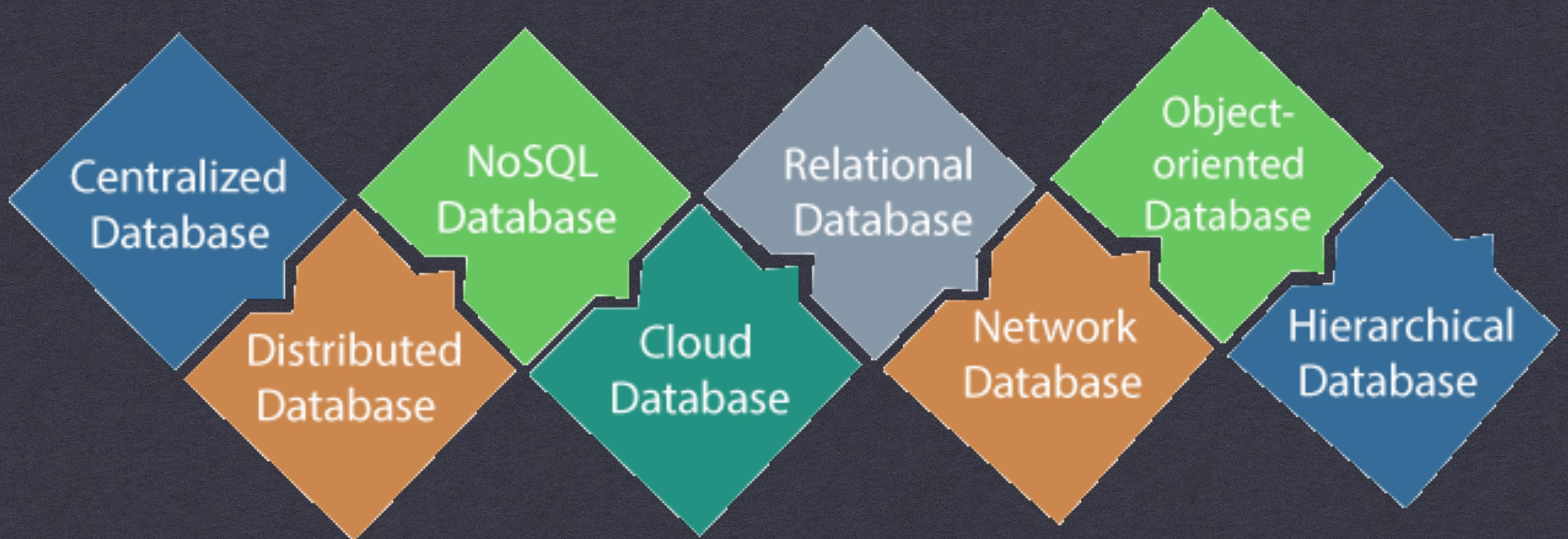
# A RELATIONAL DATABASE

1 DATABASE

3 TABLES

4 TRANSACTIONS IN EACH TABLE

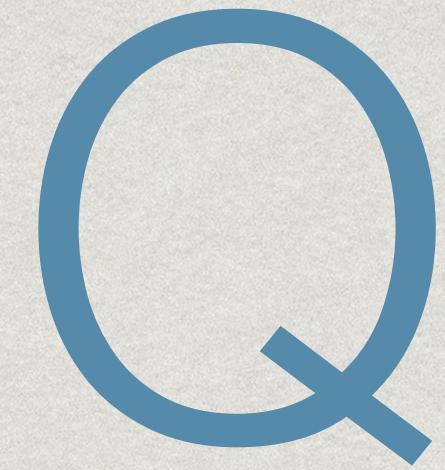
# Types of Database



**NOW, FOLLOW ALONG.....**

# S

To make /s/, place the tip of your tongue lightly against the ridge behind your upper teeth (but do not touch the teeth). As you push air out of your mouth, squeeze the air between the tip of your tongue and the top of your mouth. You should feel some friction (resistance).



The letter "Q" in English is pronounced as /kju:/, which sounds like "kyoo." It has a "k" sound at the start, followed by the long "u" sound. It's similar to the word "cue."

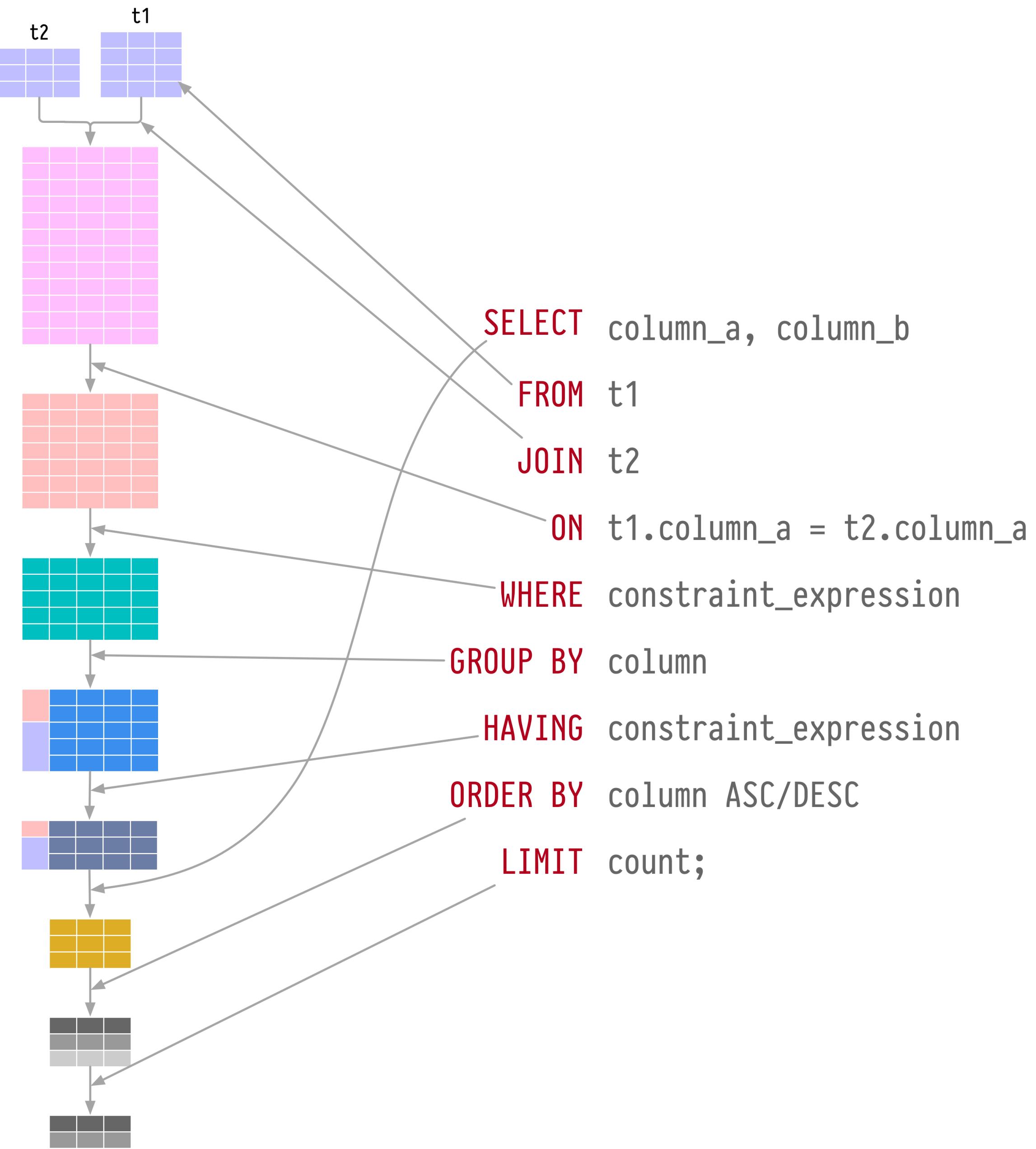
# L

The letter "L" in English is pronounced as /ɛl/, which sounds like "el." It has a short "e" sound followed by an "l" sound.



**SQL - STRUCTURED QUERY LANGUAGE**  
**QUERY - USED TO SELECT, INSERT, UPDATE OR DELETE DATA**

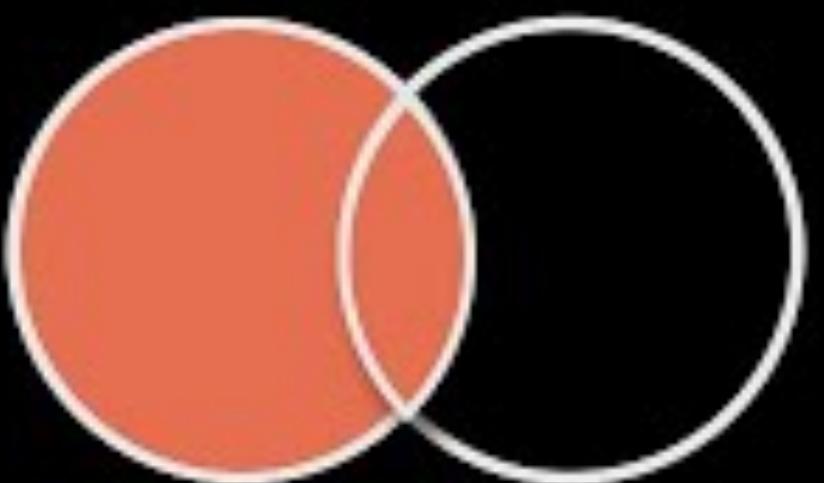
## SQL Query Execution Order



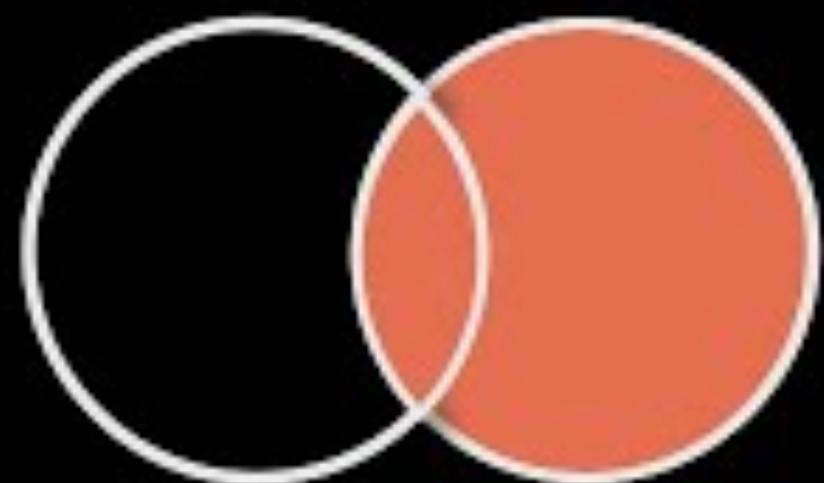
# SQL Joins



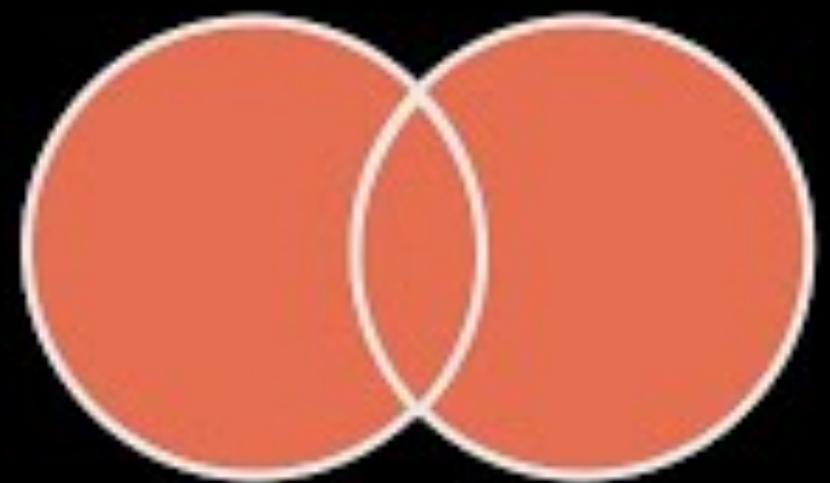
Inner Join



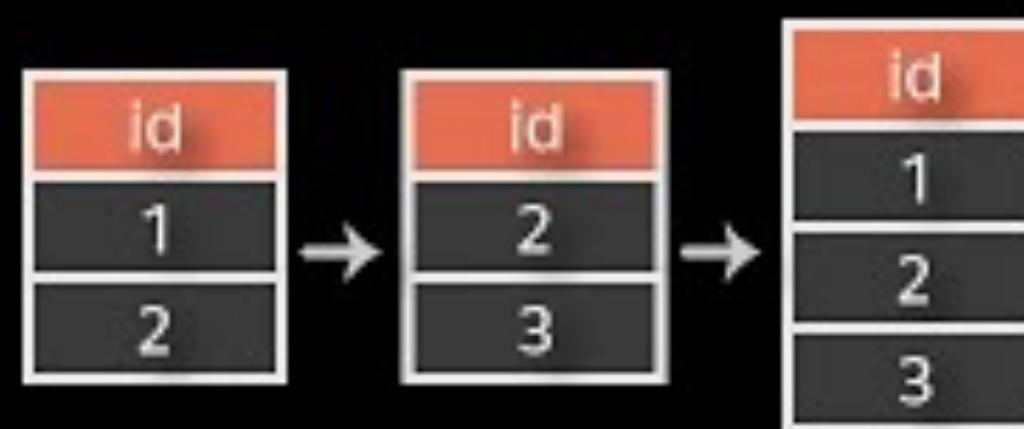
Left Join



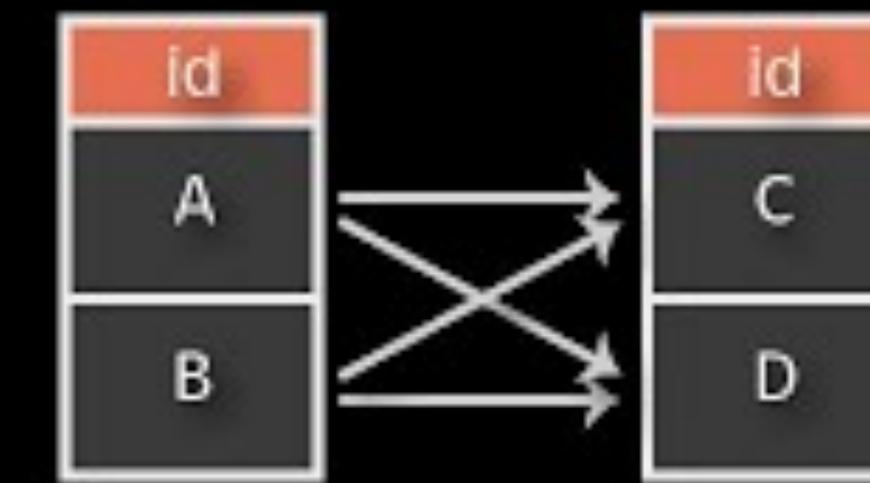
Right Join



Full Join



Union



Cross

# Aggregate Functions



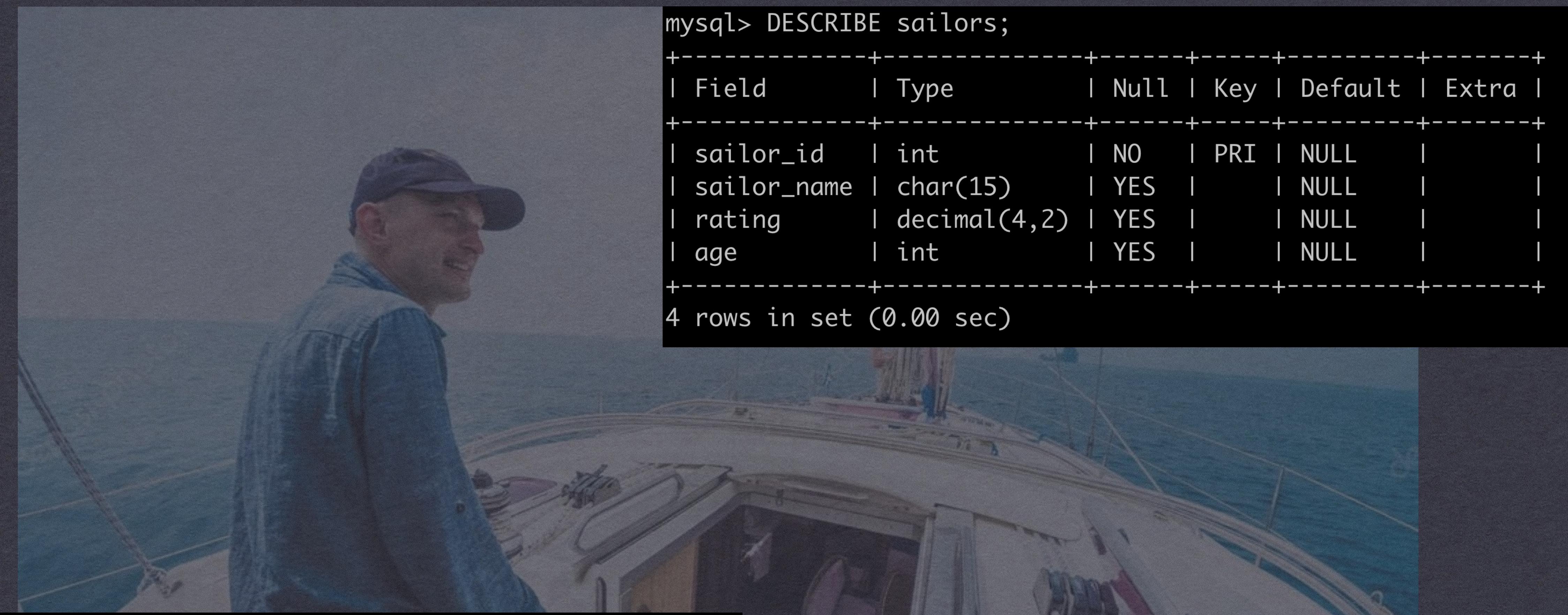
count()

sum()

avg()

max()

min()



```
mysql> DESCRIBE sailors;
```

Field	Type	Null	Key	Default	Extra
sailor_id	int	NO	PRI	NULL	
sailor_name	char(15)	YES		NULL	
rating	decimal(4,2)	YES		NULL	
age	int	YES		NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> DESCRIBE boats;
```

Field	Type	Null	Key	Default	Extra
boat_id	int	NO	PRI	NULL	
boat_name	char(15)	YES		NULL	
color	char(15)	YES		NULL	

```
3 rows in set (0.00 sec)
```

```
mysql> DESCRIBE reservations;
```

Field	Type	Null	Key	Default	Extra
sailor_id	int	YES	MUL	NULL	
boat_id	int	NO	PRI	NULL	
day	date	NO	PRI	NULL	

```
3 rows in set (0.00 sec)
```



```
// what are the names of all boats?
```

```
SELECT boat_name FROM boats;
```

```
// list all red boats
```

```
SELECT * FROM boats WHERE color = "red";
```

```
// list all sailors with age between 20 and 30
```

```
SELECT * FROM sailors WHERE age BETWEEN 20 AND 30;
```

```
// list the id and name of sailors with less
```

```
// than 50 years and id higher than 5
```

```
SELECT sailor_id, sailor_name  
FROM sailors
```

```
WHERE age < 50 AND sailor_id > 5;
```

"What's the fraction of students in the DB's discipline that had a grade superior to 16?"

-> Let's create fake data so we can see how to query the database to obtain this information.

```
DROP DATABASE IF EXISTS school;  
CREATE DATABASE school;  
USE school;
```

```
CREATE TABLE students  
(  
    id INTEGER PRIMARY KEY,  
    name VARCHAR(15) NOT NULL,  
    grade INTEGER NOT NULL  
) ;
```

# STUDENTS TABLE



```
INSERT INTO students(id,name,grade) VALUES (1,'Miguel',18);
INSERT INTO students(id,name,grade) VALUES (2,'Carlos',15);
INSERT INTO students(id,name,grade) VALUES (3,'Carolina',14);
INSERT INTO students(id,name,grade) VALUES (4,'Henrique',15);
INSERT INTO students(id,name,grade) VALUES (5,'Jorge',18);
INSERT INTO students(id,name,grade) VALUES (6,'Diogo',16);
INSERT INTO students(id,name,grade) VALUES (7,'Gustavo',19);
INSERT INTO students(id,name,grade) VALUES (8,'Mafalda',17);
```

# DATA INSERTION QUERIES



```
SELECT  
  (SELECT COUNT(*)  
   FROM students  
   WHERE grade > 16) * 1.0 /  
  (SELECT COUNT(*)  
   FROM students) AS fraction_above_16  
FROM  
  students;
```



```
mysql -u root < students.sql
```

```
+-----+  
| fraction_above_16 |  
+-----+  
| 0.50000 |  
+-----+  
1 row in set (0.00 sec)
```

50% of the students had a grade higher than 16!



WELL DONE