

# Analytics Jumpstart

## Working with SQLite

---

Nashville Software School



# What is a Relational Database?

- Relational databases store related data in **tabular form**
- Relationships between tables are defined with **keys** (primary and foreign)
- Most relational databases are created for use by applications and optimized for **CRUD *transactions***
  - Create
  - Read
  - Update
  - Delete
- Data Analysts/Data Scientists mostly focused on **Read** - querying the data. They want to look at the data and perform calculations without making any permanent changes.



# What is SQL

SQL stands **S**tructured **Q**uery **L**anguage. It is the language you use to interact with a database. It allows you to write out what you want to search for, goes to a database that you specify, then returns those results to you.

The Structured part of SQL means queries are written in a specific format using specific keywords.

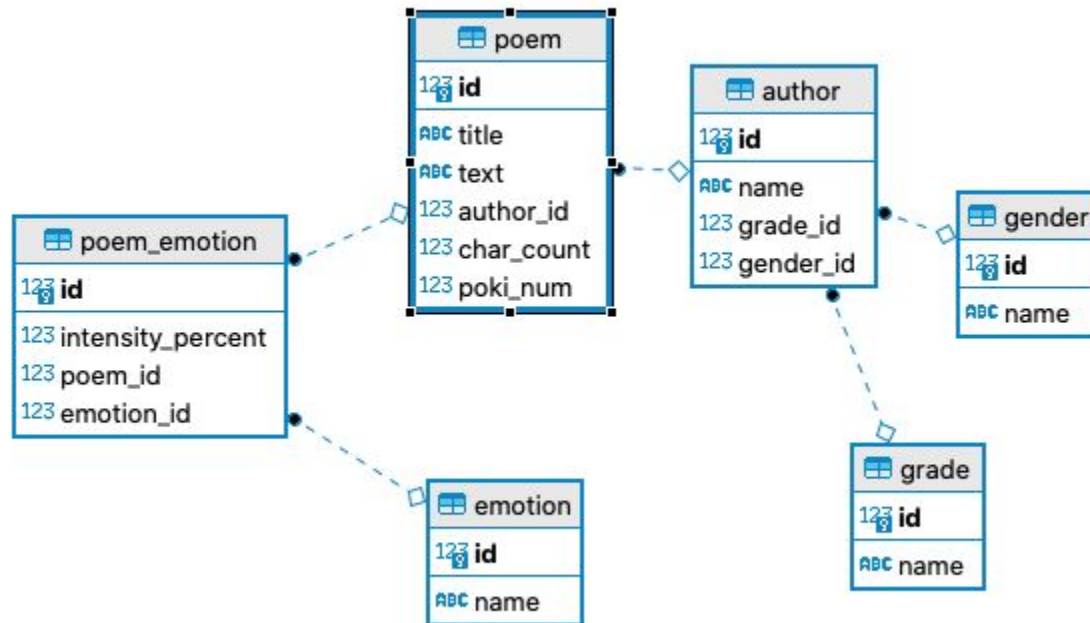


# Keywords

- **SELECT**
- **FROM**
- **AS**
- **LIMIT**
- **DISTINCT**
- **COUNT**
- **WHERE**
- **AND**
- **OR**
- **BETWEEN**
- **IN**
- **(NOT) NULL**
- **LIKE**
- **AVG(), SUM(), MAX(), MIN(), etc.**
- **ORDER BY**
- **GROUP BY**
- **HAVING**



For our learning examples, we will be using the PoetryKids database. Below is an Entity Relationship Diagram (ERD) that shows the tables in the database and the relationships between tables.

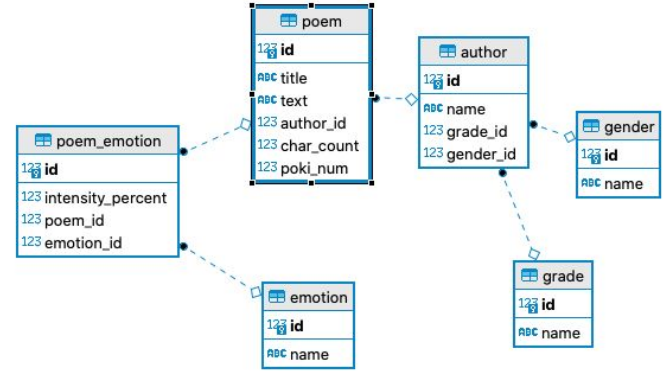


# Format of a Query

Keywords all caps

```
SELECT text  
FROM poem;
```

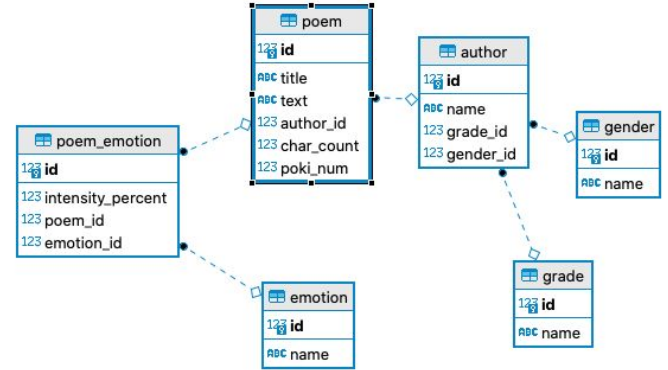
End with semicolon



# Format of a Query

the column(s)  
to retrieve

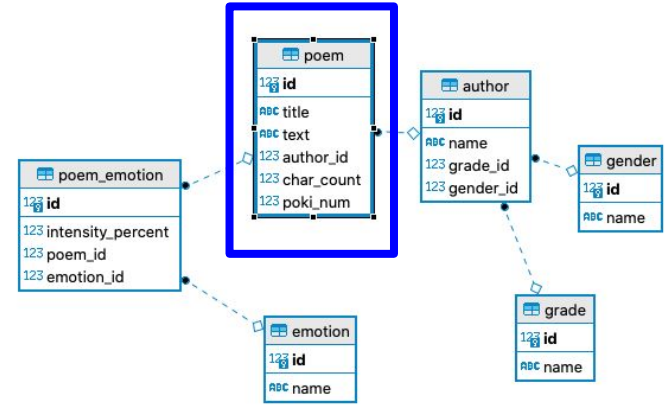
**SELECT** text  
**FROM** poem;



# Format of a Query

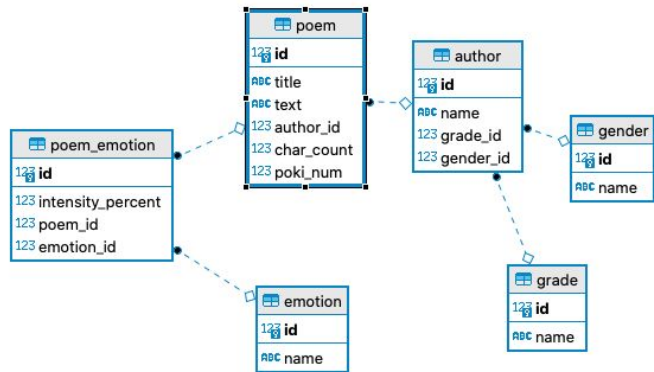
```
SELECT text  
FROM poem;
```

the table to  
retrieve from





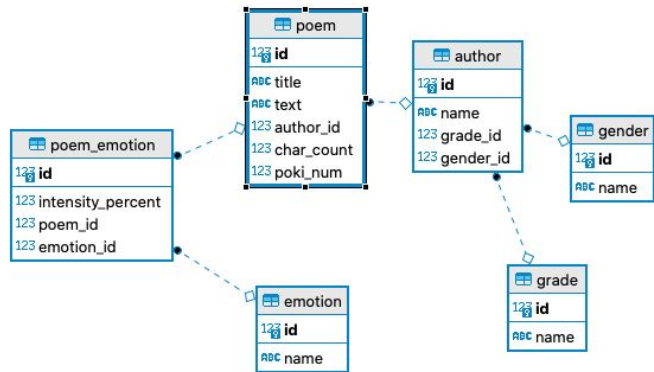
# Format of a Query



**SELECT** text  
**FROM** poem  
**WHERE** char\_count < 20;

Only return rows  
satisfying this  
condition.

# Format of a Query



```
SELECT text
FROM poem
WHERE char_count < 20
LIMIT 5;
```

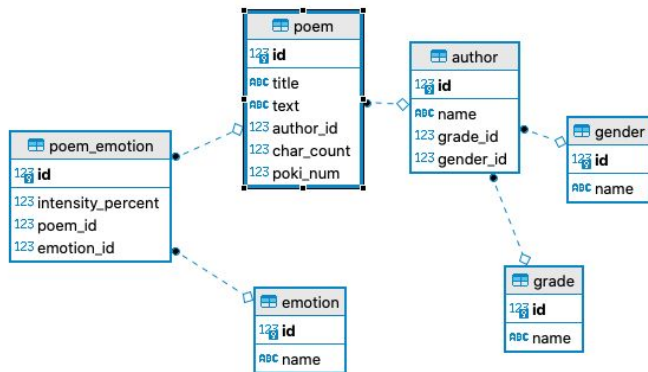
Just return the first  
5 results.

# Selecting single columns from a table

You can also specify individual columns to return, each separated by a ',':

```
SELECT name, grade_id  
FROM author  
LIMIT 5;
```

	<b>name</b> character varying	<b>grade_id</b> integer
1	a	1
2	aab	1
3	aadhya	1
4	aaliyah	1
5	aanna	1

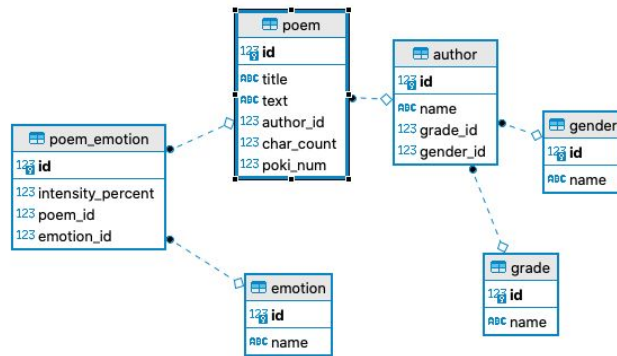


# Selecting all columns from a table

A shorthand to **SELECT ALL** is to use a **\***:

```
SELECT *  
FROM emotion;
```

	id [PK] integer	name character varying
1	1	Anger
2	2	Fear
3	3	Sadness
4	4	Joy



# Joining Data

Often data will be spread across **multiple tables**. In order to perform an analysis you may need to **combine the data from two or more tables**.

Table A		
id	col_1	col_2
1	23-B	12
2	435	45
3	AB145	23
4	BB	56
5	435	123

Table B		
id	lookup_id	col_a
a1	1	a
a2	2	a
b1	3	b
b2	4	b
a3	5	a

# Joining Data

Often data will be spread across **multiple tables**. In order to perform an analysis you may need to **combine the data from two or more tables**.

Table A		
id	col_1	col_2
1	23-B	12
2	435	45
3	AB145	23
4	BB	56
5	435	123

Table B		
id	lookup_id	col_a
a1	1	a
a2	2	a
b1	3	b
b2	4	b
a3	5	a

# Joining Data

Often data will be spread across **multiple tables**. In order to perform an analysis you may need to **combine the data from two or more tables**.

Table A		
id	col_1	col_2
1	23-B	12
2	435	45
3	AB145	23
4	BB	56
5	435	123



Table B		
id	lookup_id	col_a
a1	1	a
a2	2	a
b1	3	b
b2	4	b
a3	5	a



# Joining Data

Often data will be spread across **multiple tables**. In order to perform an analysis you may need to **combine the data from two or more tables**.

Table A		
id	col_1	col_2
1	23-B	12
2	435	45
3	AB145	23
4	BB	56
5	435	123



merge_table			
id	col_1	col_2	col_a
1	23-B	12	a
2	435	45	a
3	AB145	23	b
4	BB	56	b
5	435	123	a

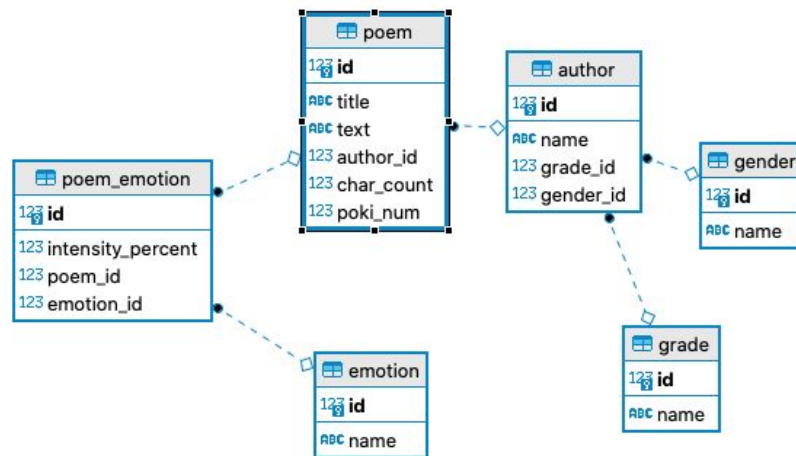
Table B		
id	lookup_id	col_a
a1	1	a
a2	2	a
b1	3	b
b2	4	b
a3	5	a





# Joining Data - example

**Goal:** Retrieve the title of each poem along with the author's name.

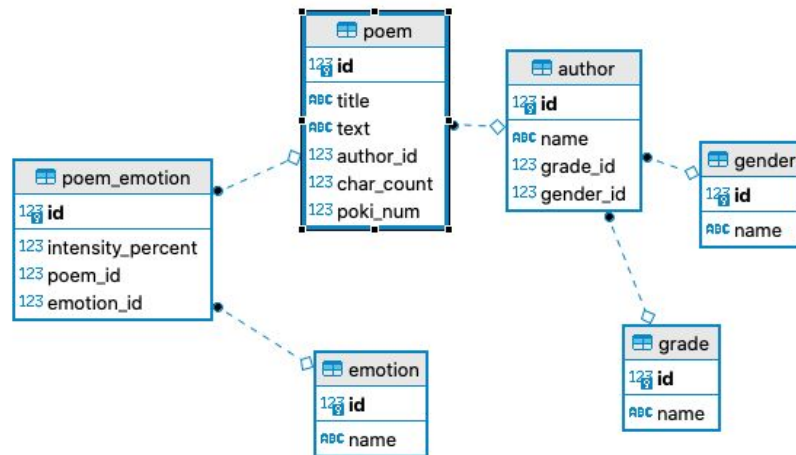


# Joining Data - example

**Goal:** Retrieve the title of each poem along with the author's name.

## Process:

- decide what tables you need
- decide how to connect your tables
- decide which columns to select



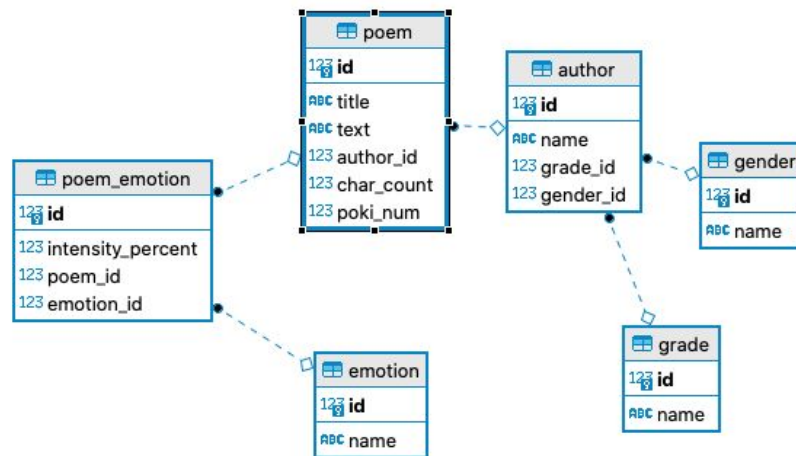
# Joining Data - example

**Goal:** Retrieve the title of each poem along with the author's name.

## Process:

- decide what tables you need
- decide how to connect your tables
- decide which columns to select

**SELECT**  
**FROM**  
**JOIN**  
**ON**



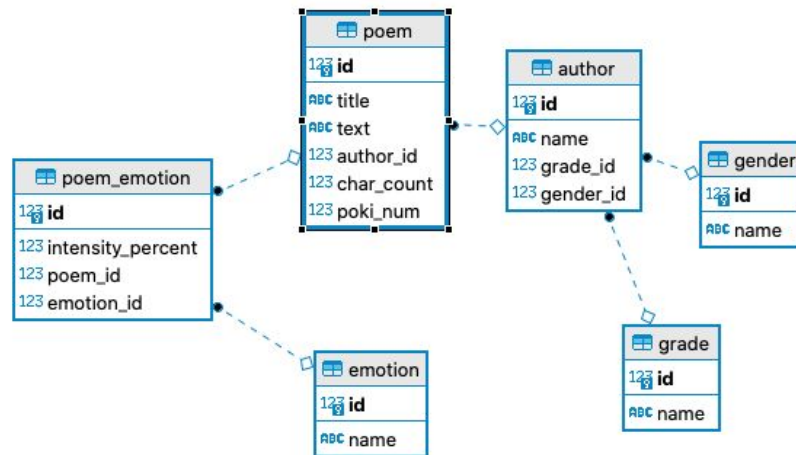
# Joining Data - example

**Goal:** Retrieve the title of each poem along with the author's name.

## Process:

- decide what tables you need
- decide how to connect your tables
- decide which columns to select

**SELECT**  
**FROM**  
**JOIN**  
**ON**



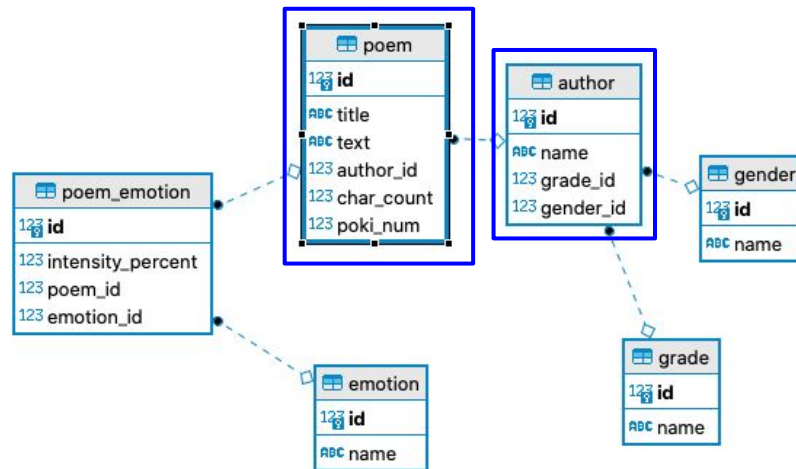
# Joining Data - example

**Goal:** Retrieve the title of each poem along with the author's name.

## Process:

- decide what tables you need
- decide how to connect your tables
- decide which columns to select

**SELECT**  
**FROM**  
**JOIN**  
**ON**



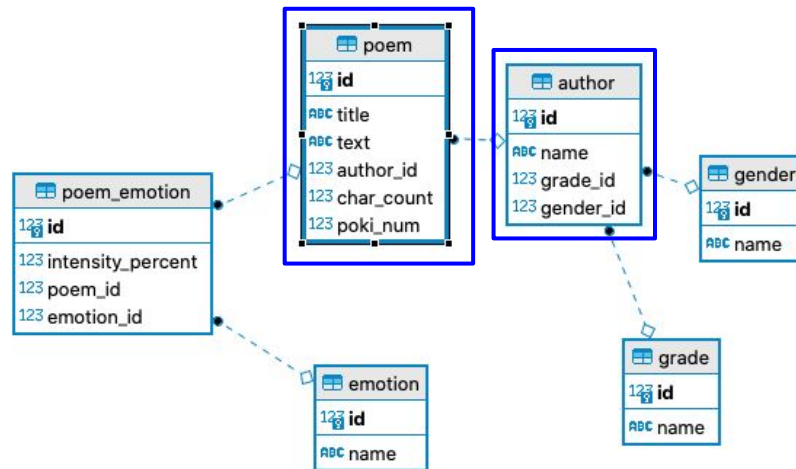
# Joining Data - example

**Goal:** Retrieve the title of each poem along with the author's name.

## Process:

- decide what tables you need
- decide how to connect your tables
- decide which columns to select

```
SELECT  
FROM poem  
JOIN author  
ON
```



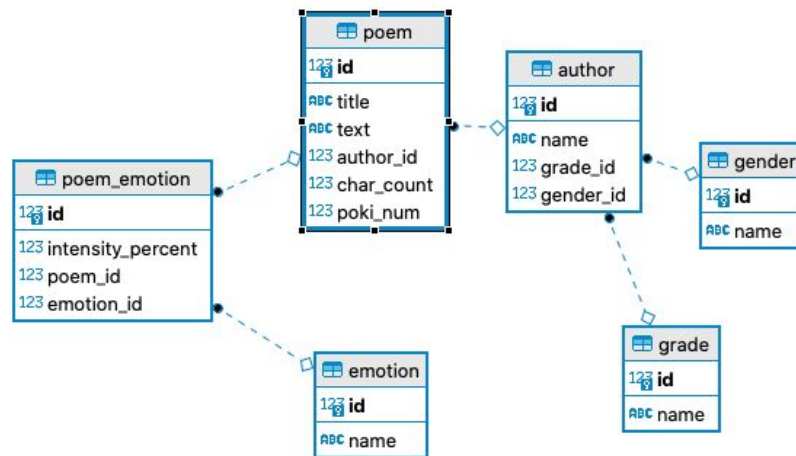
# Joining Data - example

**Goal:** Retrieve the title of each poem along with the author's name.

## Process:

- decide what tables you need
- decide how to connect your tables**
- decide which columns to select

**SELECT**  
**FROM** poem  
**JOIN** author  
**ON**



# Joining Data - example

**Goal:** Retrieve the title of each poem along with the author's name.

## Process:

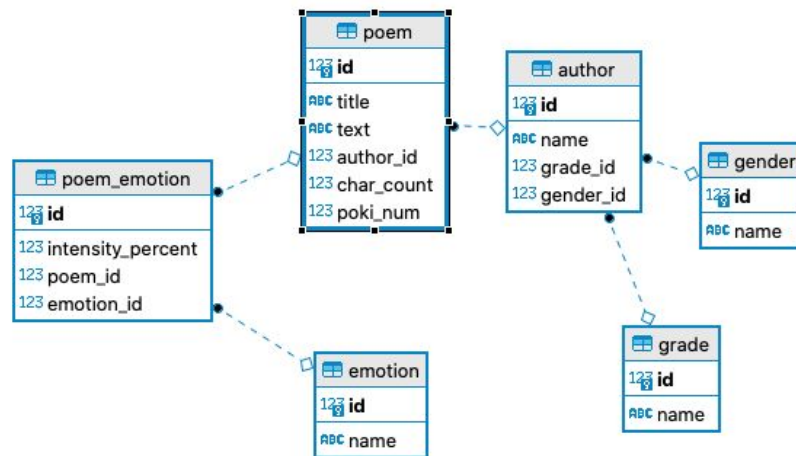
- decide what tables you need
- decide how to connect your tables**
- decide which columns to select

**SELECT**

**FROM** poem

**JOIN** author

**ON** poem.author\_id = author.id;



Specify columns using  
*table.column*



# Joining Data - example

**Goal:** Retrieve the title of each poem along with the author's name.

## Process:

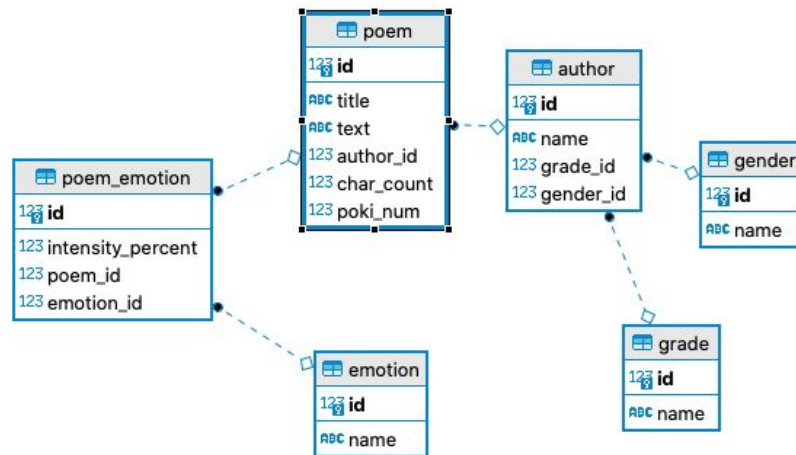
- decide what tables you need
- decide how to connect your tables
- decide which columns to select**

**SELECT**

**FROM** poem

**JOIN** author

**ON** poem.author\_id = author.id;



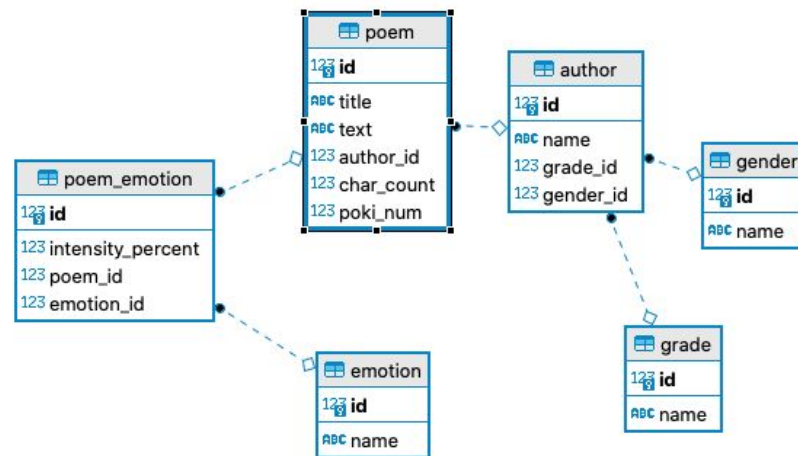
# Joining Data - example

**Goal:** Retrieve the title of each poem along with the author's name.

## Process:

- decide what tables you need
- decide how to connect your tables
- decide which columns to select**

```
SELECT poem.title, author.name  
FROM poem  
JOIN author  
ON poem.author_id = author.id;
```



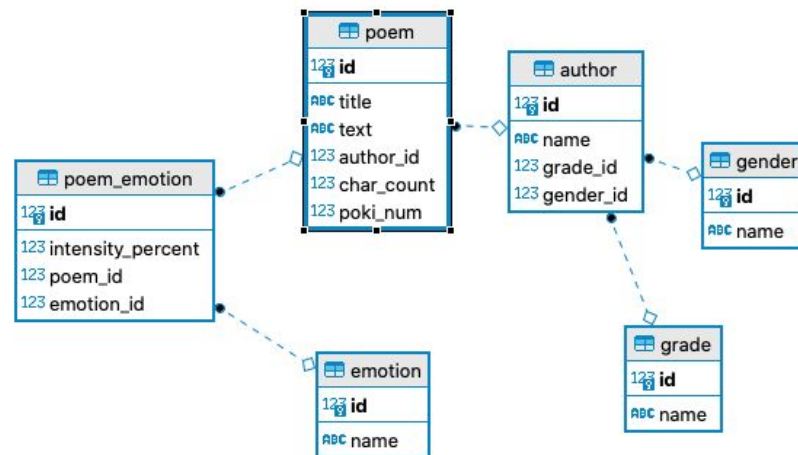
Specify columns  
using *table.column*

# Joining Data - example

**Goal:** Retrieve the title of each poem along with the author's name.

## Process:

- decide what tables you need
- decide how to connect your tables
- decide which columns to select



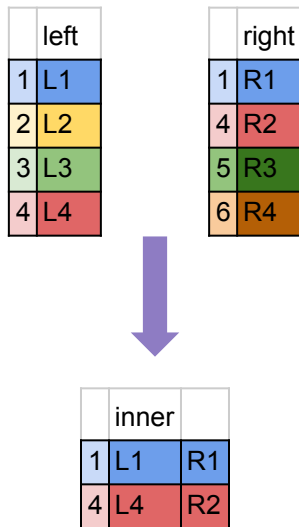
```
SELECT poem.title, author.name  
FROM poem  
JOIN author  
ON poem.author_id = author.id;
```

	title character varying	name character varying
1	Computer	a
2	Angel	aab
3	Nature Nature and Nature	aadhya
4	Jack	aaliyah
5	When I awoke one morning	aanna
6	Mv Blue Berries and Mv Cherries	aarathi

# INNER JOIN

Combining data from a **left** and **right** table is called **JOIN**ing. There are multiple kinds of joins and each one combines the data in a different way.

An **INNER JOIN** keeps **only the rows that have matching values in both tables**. This is the default type of join if you just use the **JOIN** keyword in your query.



# LEFT JOIN and RIGHT JOIN

A **LEFT JOIN** keeps **all** rows from the left table and all **matching** rows from the right table. A **RIGHT JOIN** works similarly, except all rows from the right table are kept.

left		right	
1	L1	1	R1
2	L2	4	R2
3	L3	5	R3
4	L4	6	R4



left		
1	L1	R1
2	L2	
3	L3	
4	L4	R2

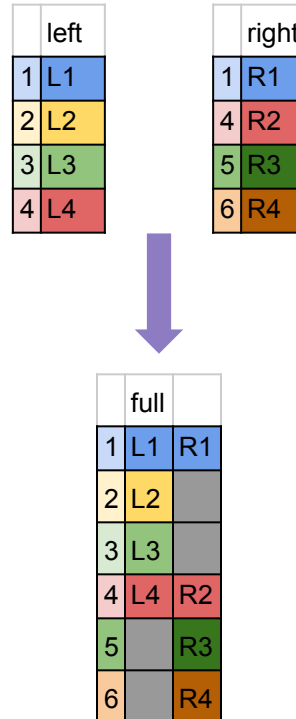
left		right	
1	L1	1	R1
2	L2	4	R2
3	L3	5	R3
4	L4	6	R4



right		
1	L1	R1
4	L4	R2
5		R3
6		R4

# FULL JOIN (Sometimes called An OUTER JOIN depending on the SQL dialect)

A **FULL JOIN** keeps **all rows from both tables**:



## Additional Practice

W3Schools SQL Tutorial - <https://www.w3schools.com/sql/default.asp>

Khan Academy - <https://www.khanacademy.org/computing/computer-programming/sql>

DataCamp - <https://learn.datacamp.com/skill-tracks/sql-fundamentals>

