Intermediate Joins in SQL: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2020

Syntax

- Joining data from more than two tables:

```
SELECT [column_names] FROM
[table_name_one]
    [join_type] JOIN [table_name_two] ON
[join_constraint]
    [join_type] JOIN [table_name_three] ON
[join_constraint]
    ...
    ...
    [join_type] JOIN [table_name_three] ON [join_constraint]
```

- Combining columns into a single column:

```
SELECT
album_id,
artist_id,
    "album id is " || album_id
col_1,
    "artist id is " || artist_id
col2,
    album_id || artist_id
col3
FROM album LIMIT 3;
```

- Matching a part of a string:

```
SELECT
first_name,
last_name,
phone
FROM
customer
WHERE first_name LIKE "%Jen%";
```

- Using if/then logic in SQL:

```
CASEWHEN [comparison_1] THEN

[valWHEN][comparison_2] THEN

[valELSE]

[value_3]

END[new_column_name]
```

Concepts

- A schema diagram helps us understand the available columns and the structure of the data.
- In a schema diagram, relationships are shown using lines between tables.
- Each row's primary key must be unique.
- A recursive join is joining a table to itself.
- The SQL engine will concatenate multiple columns and columns with a string. Also, the SQL engine also handles converting different types where needed.
- We can use the pipe operator (`||`) to concatenate columns.
- You can use the `LIKE` statement for partial matches:
    - `%Jen` : will match Jen at the end of a string, e.g., Sarah-Jen.
    - `Jen%` : will match Jen at the start of a string, e.g., Jenny.
    - `%Jen%` : will match Jen anywhere within the string, e.g., Kris Jenner.
- `LIKE` in SQLite is case insensitive but it may be case sensitive for other flavors of SQL.
    - You might need to use the `LOWER()` function in other flavors of SQL if is case sensitive.

Resources

- LOWER function
- Database Schema