Learn SQL : Multiple Tables

codecademy

Inner Join

The `JOIN` clause allows for the return of results from more than one table by joining them together with other results based on common column values specified using an `ON` clause. `INNER JOIN` is the default `JOIN` and it will only return results matching the condition specified by `ON`.

In this example, results from the `books` table are joined with results from the `authors` table in order to display an author for each book through the shared value of `author_id` in both tables.

```
SELECT *
FROM books
JOIN authors
  ON books.author_id = authors.id;
```

Outer Join

The `LEFT JOIN` clause is used to combine data from tables on a common property that the user specifies using an `ON` clause. `LEFT JOIN` takes all the records (rows) from the left table and combines them with only the matching records from the right table based on the column specified in the `ON` clause. If there is no match, the corresponding right table value(s) will be set to `NULL` in the result.

In the given query, all records from `table1` are included in the result set and are combined with the records from `table2` on a record by record basis. If a record from `table1` has the same value in column `c2` that a record from `table2` has in column `c3`, the records are combined.

```
SELECT *
FROM table1
LEFT JOIN table2
  ON table1.c2 = table2.c3;
```

`WITH` Clause

The `WITH` clause stores the result of a query in a temporary table (`temporary_movies`) using an alias.

Multiple temporary tables can be defined with one instance of the `WITH` keyword.

```
WITH temporary_movies AS (
   SELECT *
   FROM movies
)
SELECT *
FROM temporary_movies
WHERE year BETWEEN 2000 AND 2020;
```

`UNION` Clause

The `UNION` clause is used to combine results that appear from multiple `SELECT` statements and filter duplicates.

For example, given a `first_names` table with a column `name` containing rows of data “James” and “Hermione”, and a `last_names` table with a column `name` containing rows of data “James”, “Hermione” and “Cassidy”, the result of this query would contain three `name`s: “Cassidy”, “James”, and “Hermione”.

```
SELECT name
FROM first_names
UNION
SELECT name
FROM last_names
```

`CROSS JOIN` Clause

The `CROSS JOIN` clause is used to combine each row from one table with each row from another in the result set. This `JOIN` is helpful for creating all possible combinations for the records (rows) in two tables.

The given query will select the `shirt_color` and `pants_color` columns from the result set, which will contain all combinations of combining the rows in the `shirts` and `pants` tables. If there are 3 different shirt colors in the `shirts` table and 5 different pants colors in the `pants` table then the result set will contain 3 x 5 = 15 rows.

```
SELECT shirts.shirt_color,
   pants.pants_color
FROM shirts
CROSS JOIN pants;
```