PostgreSQL Constraints Review

Great job, you’ve made it to the end of your lesson in PostgreSQL constraints and data integrity. Hopefully with this information you’re feeling more prepared to design data models that are robust and can handle all of the irregularities of data in the real world. Let’s quickly go over what we’ve learned.

*Constraints* are rules a DB engineer defines as part of the data model to gain more control over what values are allowed in specific columns and tables.

Specifically, Constraints:

* Reject rows containing values that shouldn’t be inserted into a database table, which can help with preserving data integrity and quality.
* Raise an error when they’re violated, which can also help with debugging applications that write to the database.

There are quite a few types of constraints:

* Data types — Are your first line of defense, these rules aren’t constraints but can help reject incorrect data from your database.
* NOT NULL constraints — Reject incoming rows from your table when critical information is missing from a row.
* CHECK constraints — Give you more control over what rules you’d like to apply to your tables. These constraints will allow you to reject a row if it fails the criteria you’ve defined.
* UNIQUE constraints — Help with defining unique values in a table, they also create an index which can improve query and join performance.
* PRIMARY KEY constraints — A column or combination of columns that uniquely identify a row and are both NOT NULL and UNIQUE. PRIMARY KEYs are unique to a table, and will often be used in joins between tables.
* FOREIGN KEY constraints — Allow you to maintain *referential integrity* between two tables by validating the entry in one also appears in the other. Referential integrity depends on FOREIGN KEY constraints.

In case you’d like more practice, here are some resources that may be helpful:

* [PostgreSQL Constraints Docs](https://www.postgresql.org/docs/10/ddl-constraints.html)
* [Codecademy Forums](https://discuss.codecademy.com/)

**Instructions**

As one last exercise, we’ve filled the schema you’ve been working on with some sample data for the following tables.

We’ve included no constraints on these tables, feel free to play around with the data, add constraints, additional sample data, and think how you might model the relationships between these tables.

* attendees
* registrations
* speakers
* talks