



# POSTGRESQL INTRO

CLASS TWO – DATA TYPES AND CONSTRAINTS

# Benefits of Specifying Data Types

- ▶ **Consistency** – Operations against columns of same data type give consistent results and are usually the fastest.
- ▶ **Validation** – Proper use of data types implies format validation of data and rejection of data outside the scope of data type.
- ▶ **Compactness** – As a column can store a single type of value, it is stored in a compact way.
- ▶ **Performance** – Proper use of data types gives the most efficient storage of data. The values stored can be processed quickly, which enhances the performance.

# Categories of Data Types

- ▶ Numeric Types
- ▶ Monetary Types
- ▶ Character Types
- ▶ Binary Data Types
- ▶ Date/Time Types
- ▶ Boolean Types
- ▶ Enumerated Types

# Numeric Data Types

- ▶ These have varying storage sizes and range of values they can hold
- ▶ Examples of numeric data types are:
- ▶ Smallint. This holds 2bytes of data
- ▶ Integer. This holds 4 bytes of data
- ▶ Bigint. This holds 8 bytes of data
- ▶ Decimal and Numeric are essentially the same

# Monetary

- ▶ Money. Storage size is 8 bytes
- ▶ The *money* type stores a currency amount with a fixed fractional precision.

# Character

- ▶ Character varying(n) This stores a general purpose character type with a fixed length of n
- ▶ Text . This has unlimited length

# Binary Data Type

- ▶ Bytea - 1 or 4 bytes plus the actual binary string

# Date/Time Data Types

- ▶ **Date:** This data type only stores date. It will not store any time. The low value of this data type is 4713 BC, and the highest value is 294276 AD.
- ▶ **Time:** This data type only stores time. It will not store any date. .
- ▶ **Timestamp:** This data type stores the date and time. It will not store any timestamp. The low value of this data type is 4713 BC, and the highest value is 294276 AD.
- ▶ **Timestamptz:** This data type stores the date and time with the timestamp. The low value of this data type is 4713 BC, and the highest value is 294276 AD.
- ▶ **Interval:** The storage size of these data types is 12 bytes. It will describe the time interval value



# Boolean Type

- ▶ The Boolean data type can have the states *true*, *false*, and a third state, *unknown*, which is represented by the SQL null value.

# Enumerated Data Type

- ▶ Enumerated (enum) types are data types that comprise a static, ordered set of values. They are equivalent to the enum types supported in a number of programming languages.
- ▶ Unlike other types, Enumerated Types need to be created using CREATE TYPE command. This type is used to store a static, ordered set of values. For example compass directions, i.e., NORTH, SOUTH, EAST, and WEST

# CONSTRAINTS

- ▶ **NOT NULL Constraint** – Ensures that a column cannot have NULL value.
- ▶ **UNIQUE Constraint** – Ensures that all values in a column are different.
- ▶ **PRIMARY Key** – Uniquely identifies each row/record in a database table.
- ▶ **FOREIGN Key** – Constrains data based on columns in other tables.
- ▶ **CHECK Constraint** – The CHECK constraint ensures that all values in a column satisfy certain conditions.
- ▶ **EXCLUSION Constraint** – The EXCLUDE constraint ensures that if any two rows are compared on the specified column(s) or expression(s) using the specified operator(s), not all of these comparisons will return TRUE.

# NOT NULL

- ▶ A NULL is not the same as no data; rather, it represents unknown data.
- ▶ By default, a column can hold NULL values

# UNIQUE

- ▶ The UNIQUE Constraint prevents two records from having identical values in a particular column.

# PRIMARY KEY

- ▶ The PRIMARY KEY constraint uniquely identifies each record in a database table. There can be more UNIQUE columns, but only one primary key in a table. Primary keys are important when designing the database tables. Primary keys are unique ids.
- ▶ We use them to refer to table rows. Primary keys become foreign keys in other tables, when creating relations among tables. Due to a 'longstanding coding oversight', primary keys can be NULL in SQLite. This is not the case with other databases
- ▶ A primary key is a field in a table, which uniquely identifies each row/record in a database table. Primary keys must contain unique values. A primary key column cannot have NULL values.
- ▶ A table can have only one primary key, which may consist of single or multiple fields. When multiple fields are used as a primary key, they are called a **composite key**.
- ▶ If a table has a primary key defined on any field(s), then you cannot have two records having the same value of that field(s).

# FOREIGN KEY

- ▶ A foreign key constraint specifies that the values in a column (or a group of columns) must match the values appearing in some row of another table. We say this maintains the referential integrity between two related tables. They are called foreign keys because the constraints are foreign; that is, outside the table. Foreign keys are sometimes called a referencing key.

# CHECK

- ▶ The CHECK Constraint enables a condition to check the value being entered into a record. If the condition evaluates to false, the record violates the constraint and is not entered into the table.