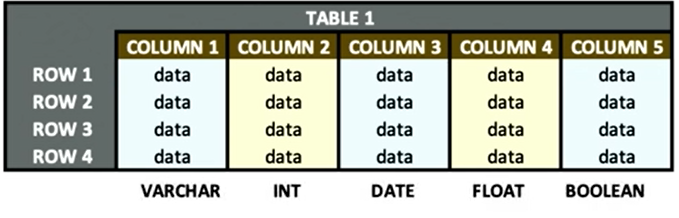
**DataTypes**

-- Tables in RDBMS consists of rows and columns.

-- Each column has a datatype associated to it.

-- The data type is like a data rule applicable to that column. Meaning that only the data which is satisfying this data rule can be inserted into that specific column.



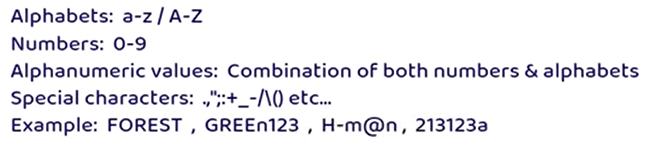
-- There are several types of data types.

-- For our learning purpose, we are going to look at 5 most used datatypes across the RDBMS.

VARCHAR

-- It stands for variable character.

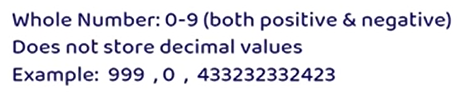
-- If a column is associated to varchar datatype, then this column can store alphabets, numbers, alphanumeric values, special characters etc.



Int

-- It stands for integer.

-- Only integer values and whole numbers are allowed in this column.



Date

-- It is used to store date values.

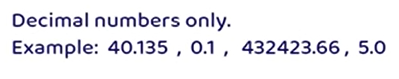
-- These values can be in any date format.



Float

-- It stands for floating point number.

-- It holds only decimal numbers.

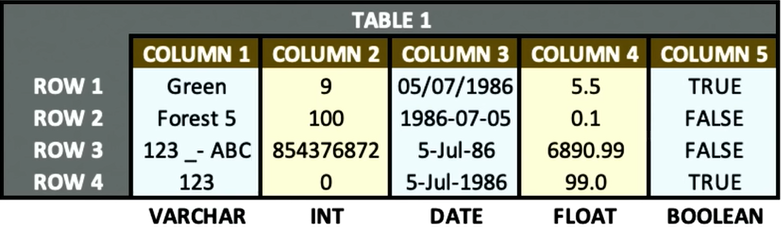


Boolean

-- It stores only two values: True or False.

-- It is kind of a binary representation where 1 stands for True whereas 0 stands for False.

-- Following table shows sample data for each datatype.



Identity Column (Auto Increment)

-- For a relational database like PostgreSQL, it could widely be considered a sin among developers not to include a primary key in every table.

-- PostgreSQL provides two ways to add primary key.

1. Using SERIAL or BIGSERIAL

2. Using Custom SEQUENCE

1. Using SERIAL or BIGSERIAL

-- Simplest and most used technique.

-- SERIAL keyword tells PostgreSQL to create an auto-incremented, unique identifier for the specified column.

> *CREATE TABLE test\_schema.identity\_test\_column*

*(*

*i\_id SERIAL primary key,*

*name varchar(100) NOT NULL*

*);*

*INSERT INTO test\_schema.identity\_test\_column(name) VALUES ('testing 1');*

*INSERT INTO test\_schema.identity\_test\_column(name) VALUES ('testing 2');*

-- By simply setting our id column as SERIAL with PRIMARY KEY attached, Postgres will handle all the complicated behind-the-scenes work and automatically increment our id column with a unique, primary key value for every INSERT.

Using Custom SEQUENCE

-- In some rare cases, the standard incremental nature built into the SERIAL and BIGSERIAL data types may not suit your needs.

-- In these cases, you can perform the same auto incremented primary key functionality for your column by creating a custom SEQUENCE.

> *CREATE SEQUENCE auto\_i\_id*

*start 100*

*increment 1*

*NO MAXVALUE;*

*INSERT INTO test\_schema.identity\_test\_column(i\_id, name) VALUES (nextval('auto\_i\_id'), 'testing 3');*