Databases and SQL for Data Science with Python Notes

Chuks Okoli

13 December, 2023

Week 1

Relational Database Concepts

Create Schema

A SQL schema is identified by a schema name, and includes a authorization identifier to indicate the user or account who owns the schema. Schema elements include tables, constraints, views, domains and other constructs that describe the schema. A schema is created using the CREATE SCHEMA statement. For example, we can create a schema called LIBRARY for this course:

CREATE SCHEMA LIBRARY AUTHORIZATION 'Robert'

The data types used can be: numeric, character-string, bit-string, Boolean, DATE, timestamp, etc.

CREATE TABLE Statement

The CREATE TABLE statement includes these clauses:

· DEFAULT

· CHECK

Use the DEFAULT clause in the CREATE TABLE statement to specify the default value for the database server to insert into a column when no explicit value for the column is specified.

Use the CHECK clause to designate conditions that must be met before data can be assigned to a column during an INSERT or UPDATE statement.

During an insert or update, if the check constraint of a row evaluates to false, the database server returns an error. The database server does not return an error if a row evaluates to NULL for a check constraint. In some cases, you might want to use both a check constraint and a NOT NULL constraint.

SELECT Statement

The basic structure of the SELECT statement is formed from three clauses: SELECT, FROM and WHERE.

<attribute list> is a list of attribute names whose values are to be retrieved by the query

<table list> is a list of the relation names required to process the query

<condition> is a conditional(Boolean) expression that identifies the tuples to be retrieved by the query

In situations where you might want to use multiple IF-THEN-ELSE statements, you can often use a single SELECT statement instead. The SELECT statement allows a CLIST to select actions from a list of possible actions. An action consists of one or more statements or commands. The SELECT statement has the following syntax, ending with the END statement. You can use the SELECT statement with or without the initial test expression.

SELECT [test expression]

```
    WHEN [expression1]
  ...
      (action)
  ...
    WHEN [expression2]
  WHEN [expression3]
  ...
  [OTHERWISE]
  ...
    (action)
  ...
END
```

Data Types

A database table represents a single entity and the columns in the table represent attributes of that entity. The commonly used data types in an RDBMS are:

- Character string data types include fixed length data types and variable length. The length of a fixed length character string is often denoted in brackets after the type name, such as CHAR(10). This type uses the same amount of space in the database irrespective of the length of the actual data stored in it.
- Variable length character strings, often named VARCHAR, can specify a maximum length for the string.
- Numeric data types include integer types and decimal types.
  - Integer data types only hold whole numbers, with no decimal parts. Integer data types typically use 2 or 4 bytes of storage to hold at numbers from negative 2 million or 32 thousand to positive 2 million or 32 thousand.
  - Smallints enable you to use less space for smaller numbers and bigints increase the size of the number that the data type can hold.
  - Decimal data types can store whole numbers and decimal numbers. The sizes and precision of these data types vary between RDBMSs and have names such as numeric, decimal, dec, real, double, float, decfloat, etc.
  - Date/time data can be categorised into dates, times, and timestamps. Dates consist of three-part values for the year, month, and day. And times also generally consist of a three part value for the hours, minutes, and seconds. A timestamp column is a combination of both and consists of seven parts: year, month, day, hour, minute, second, and microsecond.
  - Other commonly used data types include:
    - A Boolean which only holds 1 bit of information: a 0 or a 1. You can use these for true/false or yes/no type data.
    - A binary string which holds a sequence of bytes that represent image, voice, or other media data.
    - The XML data type can store platform agnostic unstructured data in a hierarchical form. In addition to the various built-in data types, many relational databases also allow you to create your own custom or "user defined" data types (UDTs) that are derived or extended from the built in types.

Summary

- The relational model is the most used data model for databases because this model allows for logical data independence, physical data independence, and physical storage independence.
- Entities are objects that exist independently of any other entities in the database, while attributes are the data elements that characterize the entity.
- The building blocks of a relationship are entities, relationship sets, and crows foot notations.
- Relationships can be one-to-one, one-to-many, or many-to-many.
- When translating an Entity-Relationship Diagram to a relational database table, the entity becomes the table and the attributes become columns in the table.
- Data types define the type of data that can be stored in a column and can include character strings, numeric values, dates/times, Boolean values and more.
- The advantages of using the correct data type for a column are data integrity, data sorting, range selection, data calculations, and the of standard functions.
- In a relational model, a relation is made up of two parts: A relation schema specifying the name of a relation and the attributes and a relation instance, which is a table made up of the attributes, or columns, and the tuples, or rows.
- Degree refers to the number of attributes, or columns, in a relation.
- Cardinality refers to the number of tuples, or rows in a relation.

Relational Database Products

There are four types of database topology:

- **Single tier.** The database is installed on a user's local desktop.
- **2-tier.** The database resides on a remote server and users access it from client systems.
- **3-tier.** The database resides on a remote server and users access it through an application server or a middle tier.
- **Cloud deployments.** The database resides in the cloud, and users access it through an application server layer or another interface that also resides in the cloud.

In shared disk distributed database architectures, multiple database servers process the workload in parallel, allowing the workload to be processed faster. There are three shared nothing distributed database architectures:

- **Replication.** Changes taking place on a database server are replicated to one or more database replicas. In a single location, database replication provides high availability. When database replica is stored in a separate location, it provides a copy of the data for disaster recovery.
- **Partitioning.** Very large tables are split across multiple logical partitions.
- **Sharding.** Each partition has its own compute resources.

There are different classes of database users, who use databases in different ways:

- Three main classes of users are Data Engineers, Data Scientists and Business Analysts, and Application Developers.
- Database users can access databases through Graphical and Web interfaces, command line tools and scripts, and APIs and ORMs.
- Major categories of database applications include Database Management tools, Data Science and BI tools, and purpose built or off the shelf business applications.
- Relational databases are available with commercial licenses or open source.
- MySQL is an object-relational database that supports many operating systems, a range of languages for client application development, relational and JSON data, multiple storage engines, and high availability and scalability options.
- PostgreSQL is an open source, object-relational database that supports a range of languages for client application development, relational, structured, and non-structured data, and replication and partitioning for high availability and scalability

Week 2

Creating Tables and Loading Data

Types of SQL Statements

- SQL Statement types: DDL and DML
- Data Definition Language (DDL) statments are used to define, change, or drop database objects such as tables:
  - CREATE - for creating tables and defining its columns
  - ALTER - for altering tables including adding, dropping columns and modifying datatypes
  - TRUNCATE - for deleting data in a table but not the table itself
  - DROP - for deleting tables
- Data Manipulation Language (DML) statements are used to read and modify data in tables.
  - Refered to as CRUD operations, that is, Create, Read, Update and Delete rows in a table. Common DML statements include:
  - INSERT - inserting row or several rows of data into a table
  - SELECT - reads or select row or rows from a table
  - UPDATE - edits row or rows in a table
  - DELETE - removes a row or rows of data from a table

Designing Keys, Indexes, and Constraints

The objects in a Relational Database Management System (RDBMS) object hierarchy include:

- **Instances:** This is a logical boundary for a database or set of databases where you organize and isolate database objects and set configuration parameters.
- **Relational databases:** This is a set of objects used to store, manage, and access data.
- **Schemas:** A user or system schema is a logical grouping of tables, views, nicknames, triggers, functions, packages, and other database objects. Schemas provide naming contexts so that you can distinguish between objects with the same name.
- **Database partitions:** You can split very large tables across multiple partitions to improve performance.
- **Database objects:** Database objects are the items that exist within the database, such as tables, constraints, indexes, views, and aliases.

Primary key and Foreign Keys have several uses:

- Primary keys enforce uniqueness of rows in a table, whereas Foreign keys are columns in a table that contain the same information as the primary key in another table.
- You can use primary and foreign keys to create relationships between tables. Relationships between tables reduce redundant data and improve data integrity.
- Indexes provide ordered pointers to rows in tables and can improve the performance of SELECT queries, but can decrease the performance of INSERT, UPDATE, and DELETE queries.

Normalization reduces redundancy and increases consistency of data. There are two forms of normalization:

- **First normal form (1NF)** - In this form, the table contains only single values and has no repeating groups.
- **Second normal form (2NF)** - This form splits data into multiple tables to reduce redundancy.

You can define six relational model constraints:

- **Entity integrity constraint** - Ensures that the primary key is a unique value that identifies each tuple (or row.)
- **Referential integrity constraint** - Defines relationships between tables.
- **Semantic integrity constraint** - Refers to the correctness of the meaning of the data.
- **Domain constraint** - Specifies the permissible values for a given attribute.

- **Null constraint -** Specifies that attribute values cannot be null.
- **Check constraint -** Limits the values that are accepted by an attribute.