SQL Basics: Comprehensive Review

DSC 208R: Data Management for Analytics

Introduction to SQL

SQL (Structured Query Language) is a powerful language used for managing and manipulating relational databases.

Components of SQL

SQL is broadly categorized into:

- **Data Definition Language (DDL)**: Used to create, alter, and delete tables and their attributes.
- **Data Manipulation Language (DML)**: Used to query one or more tables, and to insert, delete, or modify tuples (rows) in tables.
- **Triggers and Advanced Constraints**: Actions executed by the DBMS on updates, and used to specify complex integrity constraints.

Basic SQL Query Structure

A basic SQL query follows this structure:

```
SELECT [DISTINCT] <column expression list>
FROM <list of tables>
WHERE <predicate>
```

- `SELECT`: Specifies columns to be retained in the results. The `DISTINCT` keyword (optional) ensures the resulting table does not have duplicate rows.
- `FROM`: Specifies the cross-product of tables from which data is retrieved.
- `WHERE`: Specifies selection conditions applied to the tables mentioned in the `FROM` clause.

SQL Operations Examples

Schema Used in Examples

- `Movie` (name, year, genre)
- `ActedIN` (actorname, moviename)

Projection in SQL

Query:

```
SELECT name, genre
FROM Movies
```

Description: Return movies names and their genres.

Input Table (Movies):

Name	**Year**	**Genre**
Apocalypse Now	1989	War
The God Father	1972	Crime
Planet Earth II	2016	Nature, Documentary

Resulting Table:

Name	**Genre**
Apocalypse Now	War
The God Father	Crime
Planet Earth II	Nature, Documentary

Selection in SQL

Query:

```
SELECT *
FROM Movies
WHERE year > 2000
```

Description: Return movies produced after 2000.

Input Table (Movies):

Name	**Year**	**Genre**
Apocalypse Now	1989	War
The God Father	1972	Crime
Planet Earth II	2016	Nature, Documentary

Resulting Table:

Name	**Year**	**Genre**
Planet Earth II	2016	Nature, Documentary

Selection and Projection in SQL

Query:

```
SELECT name
FROM Movies
WHERE year > 2000
```

Description: Return movie names produced after 2000.

Input Table (Movies):

Name	**Year**	**Genre**
Apocalypse Now	1989	War
The God Father	1972	Crime
Planet Earth II	2016	Nature, Documentary

Resulting Table:

Name
Planet Earth II

Joins in SQL

Query:

```
SELECT DISTINCT genre
FROM Movie, ActedIN
WHERE Movie.name = ActedIN.moviename
```

Input Tables: Movie Table:

Name	**Year**	**Genre**
Apocalypse Now	1979	War
The God Father	1972	Crime
Planet Earth II	2016	Nature Documentary

ActedIN Table:

Actorname	**Moviename**
Marlon Brando	Apocalypse Now
Al Pacino	The God Father
Marlon Brando	The God Father

What does this query return?

This query performs a join operation between the ‘Movie‘ and ‘ActedIN‘ tables based on the condition that ‘Movie.name‘ matches ‘ActedIN.moviename‘. It then selects the distinct genres from the resulting joined table.

Step-by-step breakdown:

1. **Cross-product (FROM Movie, ActedIN):** All combinations of rows from ‘Movie‘ and ‘ActedIN‘ tables are generated.

2. **Selection (WHERE Movie.name = ActedIN.moviename):** Rows from the cross-product are filtered where the ‘name‘ from the ‘Movie‘ table matches the ‘moviename‘ from the ‘ActedIN‘ table.

   - (Apocalypse Now, 1979, War) JOIN (Marlon Brando, Apocalypse Now)
   - (The God Father, 1972, Crime) JOIN (Al Pacino, The God Father)
   - (The God Father, 1972, Crime) JOIN (Marlon Brando, The God Father)

3. **Projection (SELECT DISTINCT genre):** From the filtered rows, only the ‘genre‘ column is selected, and duplicate genre values are removed.

Resulting Table (genres of movies that have actors listed in ActedIN):

Genre
War
Crime