Introduction to SQL

Przemysław Rola

March 30, 2025

What is SQL?

- Structured Query Language (SQL) is the standard language for managing relational databases.
- It allows:
  - Accessing and retrieving data.
  - Creating, modifying, and deleting databases and tables.
  - Adding, updating, and deleting data.
  - Processing data using built-in functions.
  - Creating reports (views).
  - Setting access permissions.

SQL Timeline

- 1973 - SEQUEL (Structured English Query Language) by IBM.
- 1979 - First commercial SQL-based system (Oracle).
- 1986 - First SQL standard (SQL:86).
- 1992 - SQL:92 (basis for modern SQL versions, extended to SQL:2003).

SQL Query Types

- **DQL** (Data Query Language) - SELECT
- **DML** (Data Manipulation Language) - INSERT, UPDATE, DELETE
- **DDL** (Data Definition Language) - CREATE, ALTER, DROP
- **DCL** (Data Control Language) - GRANT, REVOKE, DENY
- **TCL** (Transaction Control Language) - COMMIT, ROLLBACK

Query Processing Steps

1. Parsing and optimization by the SQL processor.
2. Execution by the database engine.
3. Returning results.

SQL Syntax: SELECT

Basic structure:

```
SELECT [DISTINCT|ALL] column(s)
FROM table
WHERE condition
GROUP BY column(s)
HAVING condition
ORDER BY column(s) [ASC|DESC]
LIMIT n;
```

SELECT Statement

SELECT - retrieves data from a database

- **SELECT** - keyword for data retrieval
- **DISTINCT** | **ALL** - return unique or all values
- ***** - select all columns
- **Column Name** - specify which columns to retrieve
- **FROM** `table_name` - specify table source
- **WHERE** - filter data
- **GROUP BY ... [HAVING ...]** - group results
- **ORDER BY** - sort results
- **LIMIT** - restrict result count

Example Queries

Select all records:

```
SELECT * FROM film;
```

Select specific columns:

```
SELECT title, release_year, length FROM film;
```

Use CONCAT function:

```
SELECT CONCAT(title, ’ (’, release_year, ’)’) AS Film
FROM film;
```

Filtering with WHERE

```
SELECT * FROM film WHERE length >= 120;
SELECT * FROM film WHERE release_year <> 2006;
SELECT * FROM rental WHERE return_date IS NULL;
```

Logical Operators: AND, OR

```
SELECT * FROM film WHERE rating = ’G’ AND rental_rate
<= 2.99;

SELECT * FROM actor WHERE last_name = ’Harris’ OR
first_name = ’Julia’;
```

Using IN and BETWEEN

```
SELECT * FROM film WHERE rating IN (’G’, ’PG’);
SELECT * FROM film WHERE rental_rate BETWEEN 2.99 AND
4.99;
```

Elements of SQL Language

- **Keywords** - e.g., SELECT, INSERT, DROP
- **Identifiers** - names of databases, tables, columns
- **String literals** - stored in single quotes ('example')
- **Expressions** - consist of keywords, identifiers, literals
- **Variables** - start with @ (e.g., @result)
- **Comments** - Single-line (--), Multi-line (/* ... */)

LIKE Operator

LIKE - used for pattern matching in SQL queries.

- **%** - represents a string of any length (including an empty string).
- _ - represents exactly one occurrence of any character.

Examples:

- '%y%’ - matches any string containing the letter ’y’ (case-insensitive).
- '_d_’ - matches any three-character string where the second character is ’d’.
- 'S%’ - matches any string of at least two characters where the first character is ’S’ (case-insensitive).

Pattern Matching with LIKE

```
SELECT * FROM actor WHERE last_name LIKE ’J%’;
SELECT * FROM film WHERE description LIKE
’%student%’;
```

Sorting with ORDER BY

```
SELECT * FROM film ORDER BY length DESC, title;
SELECT CONCAT(first_name, ’ ’, last_name) FROM actor
ORDER BY last_name;
```

Limiting Results

```
SELECT * FROM payment ORDER BY amount LIMIT 10;
SELECT length FROM film WHERE length > 150 ORDER BY
length DESC LIMIT 10;
```

SQL Exercises - Part 1

1. List information about actors.
2. List information about languages in which movies could be filmed.
3. List information about countries from which rental store customers may come.
4. List different last names of actors.
5. Display the first name, last name, and email of all customers, merging the first and last name into a single column labeled "person".

SQL Exercises - Part 2

1. List titles and descriptions of all movies that cost no more than $1 to rent and are rated ’G’.
2. Display actor names (first and last) in a single column labeled ”actor” where either the first or last name starts with ’J’.
3. List rentals recorded in June for customers with IDs 111, 222, 333, 444, and 555.
4. List distinct actor names that contain the letter ’A’ but do not start or end with ’A’.
5. List movie titles that contain the word ’love’ and either have a replacement cost between 20 and 30 dollars or are at least 90 minutes long.

SQL Exercises - Part 3

1. List all unique payment amounts processed by employee with ID 1.
2. List all actor first names in alphabetical order.
3. List all movie titles in the PG category sorted from shortest to longest, with a secondary alphabetical sorting by title.
4. List the first 100 payment records for customers with IDs between 100 and 199 who rented movies in August, sorted by payment amount and then by payment date.
5. List all active customers of store 1 along with their email addresses in a single column labeled "customer" with the format "First Last (email)", sorted alphabetically by last name and first name.