SQL

DSC-301-A

Brandon Leslie

```
library(DBI)
library(RSQLite)

con <- dbConnect(SQLite(), "employees.db")

connection <- con
```

What is SQl?

SQL (also known as Structured Query Language) is a standard Language used to interact with relational databases.SQL uses various functions to Store, Delete, Move, Alter data from data tables. These functions are put into two categories, *Aggregate* & *Scalar*.

Aggregate

Aggregate functions operate in groups fo rows, but return a single summary value. They are often used in conjunction with the "GROUP BY" argument to calculate statistics across different categories or data segments.

Function	Description
AVG()	Finds the average value of the given variable (if numeric or float)
COUNT()	Gives the count of a specified variable
FIRST()	Gives the first iteration of a row
LAST()	Gives the last iteration of a row
MAX()	Gives the max value
MIN()	Gives the minimum value
SUM()	The sumation of every specified value

Scalar

Scalar functions operate on individual rows and return a single value for each row. These are often used in the SELECT clause to modify or transform column values without grouping.

Function	Description
UNCASE()	Converts string to uppercase.
LCASE()	Converts string to lowercase.
MID()	Extracts sub-string from a string.
LEN()	Returns length of a string.
ROUND()	Rounds a numeric/float value to a specific number of places.
NOW()	Returns current date & time.
FORMAT()	Formats a number or date to a specific format.

Common SQL Engines

MySQL

MySQL is an open-sourced, widely used **RDMBS (Relational Databse Manegement System)**. It is great for web apps, and supports high scalability & reproduction. MySQL is owned by Oracle Corporation.

SQLite

SQLite is a lightweight, file-based database system. There is zero set-up, and it's ideal for mobile and embedded apps. No server is required, and you can read from a single file. SQLIte is used in browsers, android apps, and small projects.

PostgreSQL

PostgreSQL is an Advanced, open-source object-relational database. It supports complex queries, indexing, and custom data types. Postgre is great for large-scale, data-heavy applications. It is also highly extensible, and standards-compliant.

OracleSQL

OracleSQL is an enterprise grade, commercial RDMBS. Is is strong in security, performance, and scalability. Oracle is ideal for mission critical and high-volume workloads. Oracle also offers extensive cloud & analytics support.

Example of SQL in R

Below is an example of me implementing SQL in RStudio.

I'll be using the code from our assignment 14, which includes various string manipulation/condition functions.

Table Creation

```
-- Create Employees Table
CREATE TABLE IF NOT EXISTS Employees (
    emp_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    department VARCHAR(50),
    email VARCHAR(100),
    phone_number VARCHAR(20),
    salary DECIMAL(10,2)
);
```

Table Insertion

```
INSERT OR IGNORE INTO Employees (emp_id, first_name, last_name, department, email, phone_numb
VALUES
(1, 'Alice', 'Johnson', 'HR', 'alice.j@company.com', '123-456-7890', 60000),
(2, 'Bob', 'Smith', 'IT', 'bob.s@company.com', '987-654-3210', 70000),
(3, 'Charlie', 'Brown', 'Finance', 'charlie.b@company.com', '456-789-1234', 80000),
(4, 'David', 'Wilson', 'IT', 'david.w@company.com', '654-321-9876', 65000),
(5, 'Emma', 'Davis', 'Marketing', 'emma.d@company.com', '321-654-7890', 75000),
(6, 'Frank', 'Miller', 'HR', 'frank.m@company.com', '789-123-4567', 72000);
```

Assigment 14

```
-- 1. Retrieve full name
SELECT first_name || ' ' || last_name AS full_name FROM Employees;
```

Table 3: 6 records

full_name
Alice Johnson
Bob Smith
Charlie Brown
David Wilson
Emma Davis
Frank Miller

```sql
-- 2. Uppercase names
SELECT UPPER(first_name), UPPER(last_name) FROM Employees;
```

Table 4: 6 records

UPPER(first_name)	UPPER(last_name)
ALICE	JOHNSON
BOB	SMITH
CHARLIE	BROWN
DAVID	WILSON
EMMA	DAVIS
FRANK	MILLER

```sql
-- 3. Extract domain from email
SELECT email, SUBSTR(email, INSTR(email, '@') + 1) AS domain FROM Employees;
```

Table 5: 6 records

email	domain
alice.j@company.com	company.com
bob.s@company.com	company.com
charlie.b@company.com	company.com
david.w@company.com	company.com
emma.d@company.com	company.com
frank.m@company.com	company.com

```sql
-- 4. First names starting with 'D'
SELECT * FROM Employees WHERE first_name LIKE 'D%';
```

Table 6: 1 records

emp_id	first_name	last_name	department	email	phone_number	salary
4	David	Wilson	IT	david.w@company.com	654-321-9876	65000

```sql
-- 5. Last names containing 'o'
SELECT * FROM Employees WHERE last_name LIKE '%o%';
```

Table 7: 3 records

emp_id	first_name	last_name	department	email	phone_number	salary
1	Alice	Johnson	HR	alice.j@company.com	123-456-7890	60000
3	Charlie	Brown	Finance	charlie.b@company.com	456-789-1234	80000
4	David	Wilson	IT	david.w@company.com	654-321-9876	65000

```sql
-- 6. Length of emails
SELECT email, LENGTH(email) AS email_length FROM Employees;
```

Table 8: 6 records

email	email_length
alice.j@company.com	19
bob.s@company.com	17
charlie.b@company.com	21
david.w@company.com	19
emma.d@company.com	18
frank.m@company.com	19

```sql
-- 7. Replace "." with "_"
SELECT email, REPLACE(email, '.', '_') AS updated_email FROM Employees;
```

Table 9: 6 records

email	updated_email
alice.j@company.com	alice_j@company_com
bob.s@company.com	bob_s@company_com
charlie.b@company.com	charlie_b@company_com
david.w@company.com	david_w@company_com
emma.d@company.com	emma_d@company_com
frank.m@company.com	frank_m@company_com

```sql
-- 8. Last 4 digits of phone number
SELECT phone_number, SUBSTR(phone_number, -4) AS last_four_digits FROM Employees;
```

Table 10: 6 records

phone_number	last_four_digits
123-456-7890	7890
987-654-3210	3210
456-789-1234	1234
654-321-9876	9876
321-654-7890	7890
789-123-4567	4567

```sql
-- 9. Phone numbers containing '123'
SELECT * FROM Employees WHERE phone_number LIKE '%123%';
```

Table 11: 3 records

emp_id	first_name	last_name	department	email	phone_number	salary
1	Alice	Johnson	HR	alice.j@company.com	123-456-7890	60000
3	Charlie	Brown	Finance	charlie.b@company.com	456-789-1234	80000
6	Frank	Miller	HR	frank.m@company.com	789-123-4567	72000

```sql
-- 10. Trim whitespace
SELECT first_name, TRIM(first_name) AS trimmed_name FROM Employees;
```

Table 12: 6 records

first_name	trimmed_name
Alice	Alice
Bob	Bob
Charlie	Charlie
David	David
Emma	Emma
Frank	Frank

You can use SQLite in R to run queries, however it is not recommended for large-scale projects, as most of this storage is local. Also, when rendering a document, it is important to use "OR IGNORE" & "IF NOT EXISTS" statements for database/table creation in order to avoid having to re-delete the databse everytime you wish to render the document.

How can we apply SQL?

We can apply SQl in various industries (if not all of them), here are a few examples!

Casheiring

Many fast-food chains, and some wholesale or department stores incorporate "Point of Sale (POS)" button interfaces. In simple terms, this is a pre-programmed query set to run when triggered by a GUI (Graphic User Interface) event. In many cases, a button is mapped to a product **(i.e. burger_id = 304)**.

Service Now

Service Now is a central system that helps companies automate and track tasks. Many companies across various industries use Service Now. Some low-end companies have just a basic IT email, which is isn't efficient. When too many emails come through, IT employees and others can get overwhelmed. Service Now uses information like email, type-of-problem, priory, and other categorical-based-conditions to sort these tickets. This helps IT teams stay orgnaized, optimizing their workflow.

Internet Traffic Metrics

Internet Traffic refers to the flow of data accross the internet. Every time you watch a youtube video, send a message, or open a website, you're creating internet traffic. SQL interacts with this by accessing how it is stored, queried and analyzed once it reaches the server (Local or Not). An example is that when you visit a website, your IP address, time visited, user ID, and other information can be stored. However this information is private, and when given access to the wrong type of people, you can lose your identity, money, and privacy. One of the ways people can protect against this with SQL, is the creation of **error catching statements**. An error catching statement returns an error message, (Error 101, Please Try Again Later, Wrong Password, etc.) are returned when access fails. Hackers & Pen testers will often attempt brute-force attacks, ere they repeatedly try different inputs to gain unauthorized access (process of elimination). Proper error handling and input validation can reduce the risk of such attacks by limiting feedback and preventing exploitation of vulnerabilities like SQL injection.

Car Tolls/Ticketing Systems

Modern toll & ticketing systems use sensors and cameras to log vehicle data like license plate numbers, time stamps, toll booth locations, and payment status. This data is stored in SQL, allowing for quick look-up for billing, violations or traffic analysis. SQL enables efficient querying (i.e. unpaid tolls), and error catching ensures the system doesn't crash when data is missing or corrupted. When a plate is unreadable or duplicated, an error catching statement might appear, and flag that particular row for manual review. This helps prevent data loss, system erros, and helps to catch fraudulent tags.

Grocery Department/Invetory Recall

Retail systems track inventory in real time using SQL databases that store item names, quantities, expiration dates, suppliers, and batch IDs. This data supports restocking alerts, sales trends, and product recalls.

Library Systems / Large File Storage

Libraries and digital archives rely on SQL to manage vast collections of books, PDFs, videos, and user metadata. SQL allows users to search by title, author, or file type, while also tracking checkouts, returns, and storage usage.

How can I apply SQL to my passions?

Some of my passions include football, and treating the body as a temple. With football, practice, consistency, and effort provide great opportunities. The same can be said with SQL. And with treating my body as a temple, I can improve my SQL so that my ability to understand, apply, and critique various different SQL properties can be well rounded.

Conclusion

SQL is a fundamental aspect of data science, and has been around for decades. SQL is the pillar of data storage, manipulation, and smooth workflows. While there re many different SQL engines & applications, they all draw from the same pool of Aggregate & Scalar functions. SQL needs other languages (like power shell, PHP, JSON, Ruby, and python), in order to be utilized properly. Furthermore, SQL is a fundamental aspect of data science, and should not be overlooked.