Literature Review

SQLite and SQL Server

The Basic Fundamentals

When it comes to choosing a database system, it really depends on the needs of the individual and organization. From building a native software application to web and mobile development, whether the needs are for daily users or to a niche market in medical facilities, database is an important part of software development process. The class CSC 210 Database Fundamentals, started by focusing on SQLite and SQL query, where this process allows learners to understand the basic fundamental of query in the most basic form without getting into setting up servers, SQLite has been proven a good starting point.

However, to have a better and bigger picture of database management, SQL Server is fairly important in today's organization needs for their business. Therefore, this short literature review is to focus on the capability of each database and their downside as a database management.

Servers Operating Systems

When it comes to database, being able to store your information is the one most crucial function. SQL Server can run on both Linux and Windows. Whereas SQLite is server-free and because of that, it is a portable database resource. Almost all programming languages are compatible to have access to the database. According to SQLite database website, it mentioned that most client/server SQL would implement shared repository that focuses on "scalability, concurrency, centralization and control" whereas SQLite focus on "economy, efficiency, reliability, independence, and simplicity." In which, from the start, SQL Server and SQLite have different usage intention for different users.

Same Difference

Description	Microsoft SQL Server	SQLite
Server Operating System	Linux | Windows	Server-less
XML	Yes	No
Server-side scripts/ Stored Procedures	T-SQL, .NET, R, Python, Java	No
Partitioning Methods	Tables can be distributed across several files	No
Replication Methods	Yes	No

It may seem that SQLite is on its disadvantage on not being able to do all the above compared to SQL Server. However, SQLite because of its server-less, it has its own advantages in portability where SQLite can be implemented in mobile application without much redundancy and features that might affect the processing speed. The advantages of SQLite will include: self-contained, high-reliability, embedded and full-featured in SQL.

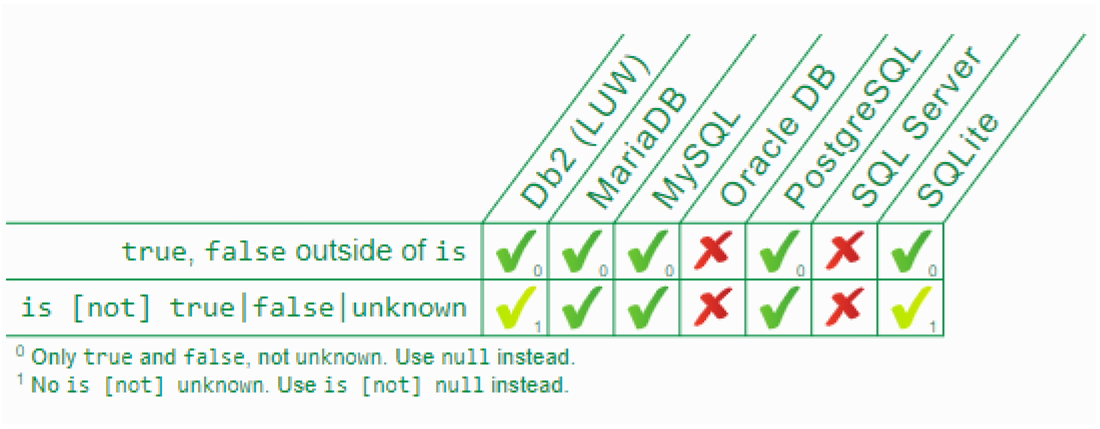
Like we have mentioned earlier, in a perspective, they are both are different product because of its capability but the common thing is that they are both using SQL. The question now would be that, if, the capability is different, would it change the implementation of SQL within the software?

<u>SQL</u>

In this section, I will be highlighting some of the features of within the SQL of different dialect or flavors between the two platform, SQL Server and SQLite. Mainly focusing on SQLite advantages due to the nature of SQL Server being able to function in a much larger scale and is a paid service for the enterprise customers, the expectation of SQL Server, in my opinion should be able to cover everything that SQLite can do.

1st Comparison (BOOLEAN)

It is intuitive that to create a Boolean test, it will be true or false. In SQLite, true and false are both treated or represented by numerical values of 0s and 1s where it accepts Boolean as type name. The interesting part would be that both paid services like Oracle DB and SQL Server does not have this feature in their SQL flavor. However, SQL Server does have a replacement to the Boolean, which is called the 'BIT' where it works the same which it is to have 0 and 1.



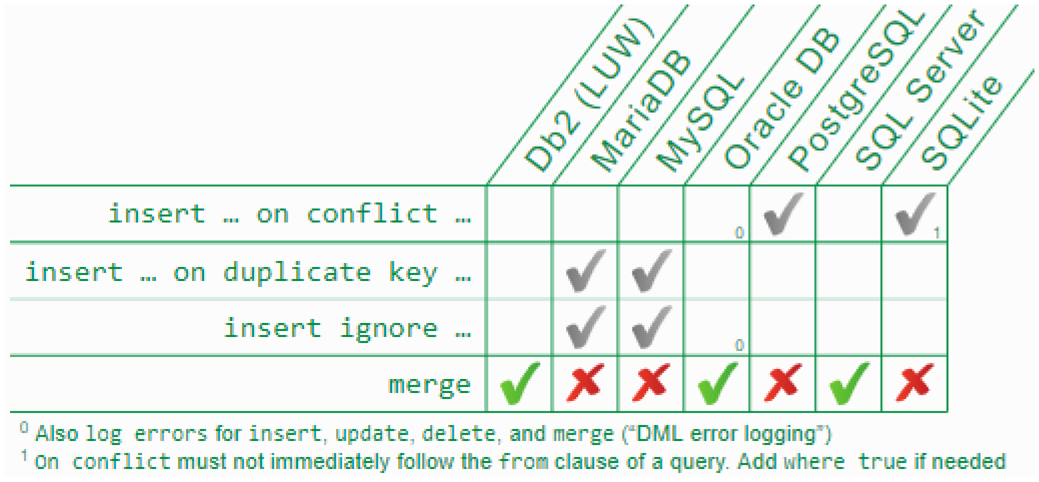
	Db2 (LUW)	MariaDB	MySQL	Oracle DB	PostgreSQL	SQL Server	SQLite
true, false outside of is	✓[0]	✓[0]	✓[0]	✗	✓[0]	✗	✓[0]
is [not] true|false|unknown	✓[1]	✓	✓	✗	✓	✗	✓[1]

[0] Only true and false, not unknown. Use null instead.
[1] No is [not] unknown. Use is [not] null instead.

Source: https://modern-sql.com/blog/2019-01/sqlite-in-2018

2nd Comparison (MERGE)

In comparison to both, SQLite is not utilizing MERGE, however when I did some research online, there are ways where you maybe able merge tables but may take longer steps and can be troublesome assuming that tables needed to be updated manually.



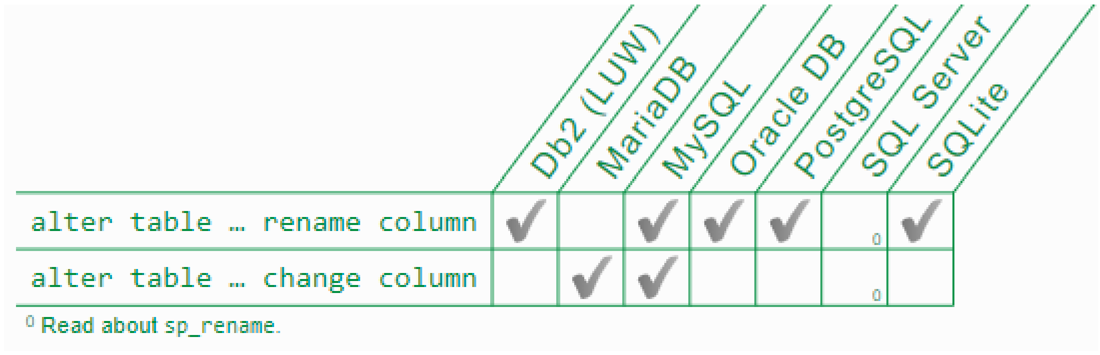
	Db2 (LUW)	MariaDB	MySQL	Oracle DB	PostgreSQL	SQL Server	SQLite
insert … on conflict …				[0]	✓		✓[1]
insert … on duplicate key …		✓	✓				
insert ignore …		✓	✓	[0]			
merge	✓	✗	✗	✓	✗	✓	✗

[0] Also log errors for insert, update, delete, and merge ("DML error logging")
[1] on conflict must not immediately follow the from clause of a query. Add where true if needed

Source: https://modern-sql.com/blog/2019-01/sqlite-in-2018

3rd Comparison (Rename Column)

On different flavor of SQL, renaming a column are both different with SQL Server and SQLite. While SQLite can use "ALTER TABLE … RENAME COLUMN … TO … " SQL Server uses sp_rename.



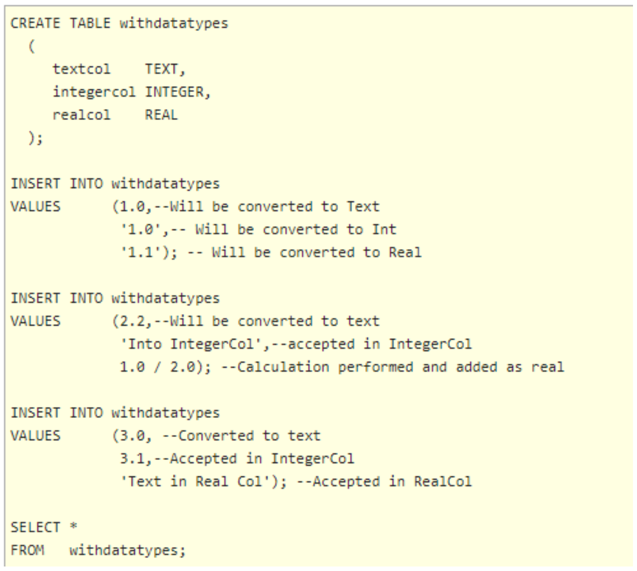
	Db2 (LUW)	MariaDB	MySQL	Oracle DB	PostgreSQL	SQL Server	SQLite
alter table … rename column	✓		✓	✓	✓	[0]	✓
alter table … change column		✓	✓			[0]	

[0] Read about sp_rename.

Source: https://modern-sql.com/blog/2019-01/sqlite-in-2018

4th Comparison (Typing)

SQLite uses dynamic typing and SQL Server uses static typing. When creating a table using SQLite, it is not necessary to assign datatype where it will track the datatype with the value while not with column.

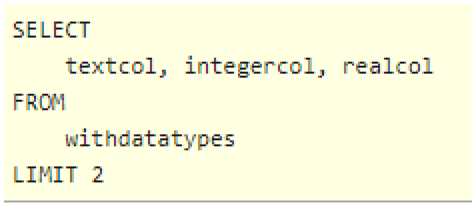


```
CREATE TABLE withdatatypes
  (
     textcol    TEXT,
     integercol INTEGER,
     realcol    REAL
  );

INSERT INTO withdatatypes
VALUES      (1.0,--Will be converted to Text
             '1.0',-- Will be converted to Int
             '1.1'); -- Will be converted to Real

INSERT INTO withdatatypes
VALUES      (2.2,--Will be converted to text
             'Into IntegerCol',--accepted in IntegerCol
             1.0 / 2.0); --Calculation performed and added as real

INSERT INTO withdatatypes
VALUES      (3.0, --Converted to text
             3.1,--Accepted in IntegerCol
             'Text in Real Col'); --Accepted in RealCol

SELECT *
FROM   withdatatypes;
```

Source: https://www.mssqltips.com/sqlservertip/4777/comparing-some-differences-of-sql-server-to-sqlite/

5th Comparison (TOP | Limit)

SQL Server utilizes TOP to limit the rows to be returned in a search. Usually, TOP is place at the very beginning of the syntax, however 'limit' which is used in SQLite, is placed at the bottom of its syntax.

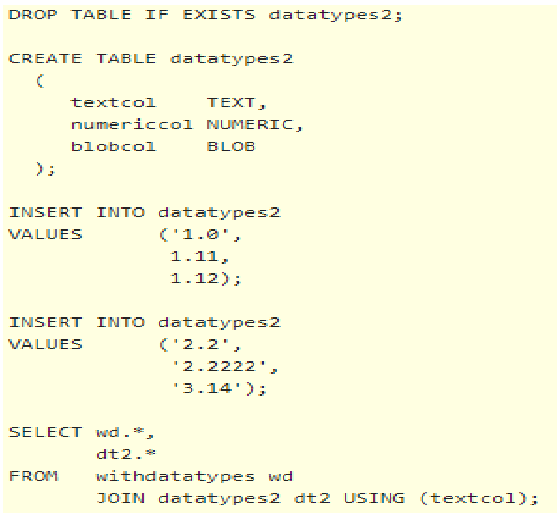


```
SELECT
    textcol, integercol, realcol
FROM
    withdatatypes
LIMIT 2
```

Source: https://www.mssqltips.com/sqlservertip/4777/comparing-some-differences-of-sql-server-to-sqlite/

6th Comparison (Joins | Using)

SQLite would support cross join, inner join, and left out joins but not the right or the full outer joins, just like SQL Server will. The inner joins in SQLite can be expressed using 'USING' and 'NATURAL JOIN.'



```
DROP TABLE IF EXISTS datatypes2;

CREATE TABLE datatypes2
  (
     textcol    TEXT,
     numericcol NUMERIC,
     blobcol    BLOB
  );

INSERT INTO datatypes2
VALUES      ('1.0',
             1.11,
             1.12);

INSERT INTO datatypes2
VALUES      ('2.2',
             '2.2222',
             '3.14');

SELECT wd.*,
       dt2.*
FROM   withdatatypes wd
       JOIN datatypes2 dt2 USING (textcol);
```

Source: https://www.mssqltips.com/sqlservertip/4777/comparing-some-differences-of-sql-server-to-sqlite/

Verdict

With some limitation in both SQLite and SQL Server, both platforms are obviously not lacking in features when it is time to perform. However, both have their own ways of writing SQL, in which I believe, it has to do with the capability of its technology. It will interesting to research further as to a deeper understanding on why both SQL flavors are having different SQL dialect for the reason of possibility of user friendliness or because of its capability or features.

Overall, along with my 50 SQL queries and understanding of SQL structure and usage, I believe that SQLite provides a great option as a database platform for many reasons. I do have to admit that SQL Server, is a power tool when it comes to enterprise use where paid services would be customer support is present to be able to help with issues. This translate to reliability, where all companies in the corporate world would value most.

Therefore, I am convinced that it is both different products targeted to different user groups or customers but have the same kind of capability to run queries and manage databases.