Writing Comparison

Friday, 3rd June 2022

By

Codie Shannon

30013375

Submitted to

Andrew Chapman

Assignment Three

In COMP.6212

Data Management

Table of Contents

Writing Comparison

In this comparison, I will compare SQL databases (SQL Server 2019 databases) to NOSQL databases (MongoDB databases) to conclude about which provides the best efficiency, functionality, and ease of use under the circumstance of creating a PMP distributions database for a supervisor.

SQL databases are comprised of tables which can store a variety of fixed data sets within rows (records) and columns (fields). SQL databases are relational in that they link multiple tables together to create a connected network of tables.

NOSQL databases on the other hand, are essentially the opposite of SQL databases. Instead of being relational, NOSQL databases utilize collections to store data in documents, which contain fields in a vertical fashion.

SQL Table Layout

	ContactsID	FirstName	LastName	Phone	Email	Street	City	isAvailable	DateAvail
1	1	Tony	Martin	021941580	tony.martin@ou…	Douglas St	Whakatane	1	*NULL*
2	2	Sarah	Clark	021255635	sarah.clark@ou…	Nelson St	Whakatane	1	*NULL*
3	3	Henry	King	022556536	henry.king@gma…	Bracket St	Whakatane	1	*NULL*
4	4	Jennifer	Jones	021660334	jennifer.jones…	McAlister St	Whakatane	1	*NULL*
5	5	Jeremy	Stewart	022561528	jeremy.stewart…	Simpkins St	Whakatane	1	*NULL*
6	6	Jennifer	Smith	021799279	jennifer.smith…	The Fairway	Whakatane	1	*NULL*
7	7	Edward	Wilson	021429775	edward.wilson@…	Awatapu Dr	Whakatane	1	*NULL*
8	8	Henry	Martin	021192129	henry.martin@y…	Henderson St	Whakatane	1	*NULL*
9	9	Ashley	Thompson	021865080	ashley.thompso…	Hawera St	Whakatane	1	*NULL*
1…	10	Kurt	Anderson	021223188	kurt.anderson@…	Barry Ave	Whakatane	1	*NULL*
1…	11	Edward	Jones	021859738	edward.jones@g…	Landscape Rd	Whakatane	1	*NULL*
1…	12	Cory	King	021499021	cory.king@yaho…	Omega PI	Whakatane	1	*NULL*
1…	13	Jack	King	021486311	jack.king@outl…	Harbour Rd	Whakatane	1	*NULL*
1…	14	Kurt	Kumar	022161524	kurt.kumar@yah…	Ocean Rd	Whakatane	1	*NULL*
1…	15	Ashley	Campbell	021821902	ashley.campbel…	James St	Whakatane	1	*NULL*
1…	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

NOSQL Document Structure


```
_id: ObjectId("62861bef459c250f314ab798")
title: "Run 1"
contact: Object
  delivererName: Object
      first: "Sarah"
      last: "Stewart"
    phone: "021609930"
    email: "sarah.stewart@gmail.com"
  address: Object
      street: "King St"
      city: "Whakatane"
  availability: Object
      isavailable: false
      dateavail: 2022-06-05T12:00:00.000+00:00
delivery: Object
  pamphlets: Array
    0: Object
        pamtype: "Farmers"
        bundlequantity: 100
    1: Object
        pamtype: "Pizza Hut"
        bundlequantity: 1550
    isdelivered: true
    deldate: 2022-06-05T12:00:00.000+00:00
map: Object
    mapimg: "https://github.com/CodieShannon/OvatoDistributions/blob/main/route_tem..."
    deliverables: 143
```

In my opinion SQL databases and NOSQL databases both have valid advantages and disadvantages.

A few key advantages of SQL databases are a standardized schema, flexible queries, and ease of use once grouped with stored procedures (Edwards, 2021; Srivastava, 2021).

SQL databases are constructed utilizing a default schema known as DBO. SQL schemas are often thought of as rigid and difficult to modify. However, the default schema does provide a few key benefits by way of limiting database tables to a set data model set forth by developers. Now some may find SQL schemas limiting, however, I see it as a method to ensure data consistency, integrity, security, and compliance without expending time (Edwards, 2021; Wolter et al., 2021).

```
-- SET DATA MODEL --
CREATE TABLE Contacts (
    ContactsID int IDENTITY(1, 1) PRIMARY KEY NOT NULL,
    FirstName NVARCHAR(100) NOT NULL,
    LastName NVARCHAR(100) NOT NULL,
    Phone VARCHAR(50) NOT NULL,
    Email NVARCHAR(255),
    Street VARCHAR(100) NOT NULL,
    City VARCHAR(100) NOT NULL,
    isAvailable BIT NOT NULL DEFAULT(1),
    DateAvail DATETIME
);
```

SQL databases place a strong emphasis on the flexibility of queries, alongside the ease of use once grouped with stored procedures. Queries are the method of performing CRUD (create, read, update, and delete) operations within databases. A neat query of SQL databases is a query by the name of JOIN. This query has the capacity to read data from multiple tables at once utilizing primary and foreign keys. Alongside the aforementioned queries, stored procedures come into play, stored procedures improve the efficiency of queries by giving developers the ability to create and store queries for later use (Bijeesh, 2008).

```
-- STORED UPDATE PROCEDURE --
-- ========================================================================== --
CREATE PROCEDURE update_deliverydate (@DeliveriesID int, @isDelivered BIT, @DelDate DATETIME)
AS
UPDATE Deliveries
SET isDelivered = (@isDelivered), DelDate = (@DelDate)
WHERE DeliveriesID = (@DeliveriesID);
GO
```

On the other hand, a key disadvantage of SQL databases is data normalization. SQL databases attempt to eliminate data duplication by separating different data into different tables, there by slimming down on data duplication. However, this has a downside as when SQL databases get larger, the join queries required between numerous tables can grow to untameable growth making it difficult to read the query, alongside slowing down the database. This can even be seen in some smaller databases such as one created for a PMP distributions supervisor (Edwards, 2021).

```
-- UNTAMEABLE JOIN QUERY --
SELECT (C.FirstName + ' ' + C.LastName) AS Full_Name, D.isDelivered, D.DelDate AS Date_Delivered,
    D.PamType1 AS Pamphlet_1, PT1.BundleQuantity AS Quantity_1, D.PamType2 AS Pamphlet_2, PT2.BundleQuantity AS Quantity_2,
    D.PamType3 AS Pamphlet_3, PT3.BundleQuantity AS Quantity_3, M.Deliverables AS Quantity
FROM Deliveries AS D

INNER JOIN Deliverers AS Del
ON D.DeliverersID = Del.DeliverersID

INNER JOIN Contacts AS C
ON Del.ContactsID = C.ContactsID

LEFT JOIN PamphletTypes AS PT1
ON D.PamType1 = PT1.PamType

LEFT JOIN PamphletTypes AS PT2
ON D.PamType2 = PT2.PamType

LEFT JOIN PamphletTypes AS PT3
ON D.PamType3 = PT3.PamType

INNER JOIN Maps AS M
ON Del.MapsID = M.MapsID;
GO
```

A couple of key advantages of NOSQL databases are dynamic schemas, and query efficiency by way of denormalization.

NOSQL databases are great for modern agile development teams, however, for the purposes of the creation of the PMP distributions database, are a bit overkill.

NOSQL database schemas perform efficiently in that they do not need to be predefined, instead the schemas are dynamic with the potential to store all kinds of data sets, including unstructured, semi-structured, and structured data sets. This also feeds into the ease of use, as you can launch NOSQL databases containing fields and values without expending time on creating a schema (Edwards, 2021).

NOSQL Semi-Structured Data Set

```
_id: ObjectId("62861bef459c250f314ab79b")
title: "Run 4"
contact: Object
  delivererName: Object
      first: "Jeremy"
      last: "Smith"
    phone: "021898120"
    email: "jeremy.smith@gmail.co.nz"
  address: Object
      street: "Bracket St"
      city: "Whakatane"
  availability: Object
      isavailable: true
delivery: Object
  pamphlets: Array
    0: Object
        pamtype: "Repco"
        bundlequantity: 200
    1: Object
    2: Object
    3: Object
  isdelivered: false
map: Object
    mapimg: "https://github.com/CodieShannon/OvatoDistributions/blob/main/route_tem..."
    deliverables: 249
```

NOSQL Structured Data Set

NOSQL recommends embedded documents over multiple collections linked together to create a connected network. This increases query efficiency as untameable join queries are no longer necessary (Pedamkar, 2022).

NOSQL Embedded Contact Document

A key disadvantage of NOSQL databases is denormalization. Denormalization can be helpful in some areas of database utilization such as query efficiency. However, denormalization can also be troublesome as it can lead to duplicates of data that are difficult to manage. Alongside this, NOSQL databases do not enforce data consistency, integrity, security, and compliance, which could lead to corrupted data (XTIVIA, 2019).

In conclusion, although NOSQL provides a range of benefits, I preferred utilizing SQL to design and develop the PMP Distributions supervisor's database. I found SQL visually easier to understand, alongside providing the necessary components I required to create the database with full efficiency, ease of use, and functionality.

References

Edwards, R. (2021, September 29). SQL versus NoSQL: Pros and Cons. *DataStax.* https://www.datastax.com/blog/sql-vs-nosql-pros-cons

Srivastava, S. (2021, July 7). *Advantages and Disadvantages of SQL.* GeeksforGeeks. https://www.geeksforgeeks.org/advantages-and-disadvantages-of-sql/

Bijeesh, P. (2008, April 23). What is the difference between a Query and Stored Procedure? *C-Sharp Corner*. https://www.c-sharpcorner.com/interview-question/what-is-the-difference-between-a-query-and-stored-procedure

Andreas Wolter et al. (2021, August 10). *Ownership and User-Schema separation in SQL Server*. Microsoft. https://docs.microsoft.com/en-us/sql/relational-databases/security/authentication-access/ownership-and-user-schema-separation?view=sql-server-ver16

Pedamkar, P. (2022). *Advantages of NOSQL*. EDUCBA. https://www.educba.com/advantages-of-nosql/

XTIVIA. (2019, August 8). The Pros and Cons of MongoDB. *Virtual DBA.* https://virtual-dba.com/blog/pros-and-cons-of-mongodb/