Contents

[Table of Figures 2](#_Toc142142400)

[Introduction 2](#_Toc142142401)

[Database Technologies 2](#_Toc142142402)

[Relational Database Management Systems (RDBMS) 2](#_Toc142142403)

[NoSQL – Not Only SQL 4](#_Toc142142404)

[Hierarchical Databases: 4](#_Toc142142405)

[Document Database 4](#_Toc142142406)

[Key-Value Database 5](#_Toc142142407)

[Graph Database 6](#_Toc142142408)

[Wide-Column Database 7](#_Toc142142409)

[Data Lakes 7](#_Toc142142410)

[Database Modelling 8](#_Toc142142411)

[Scenario Overview 8](#_Toc142142412)

[Gemini Jeweler’s (GJ), a producer and wholesaler of batches of Jewelry, implements a pre-approval process for individual item orders by company. Items, which come in various color finishes and material selections, are currently housed in two depots. However, GJ envisions future expansion and desires to add more locations. 8](#_Toc142142413)

[Appendices 8](#_Toc142142414)

[Appendix A: 8](#_Toc142142415)

[Figure 1 – RDBMS Table 8](#_Toc142142416)

[Figure 2 – Stack Overflow Database Environments Survey: 8](#_Toc142142417)

[Figure 3 – ACID Principles: 8](#_Toc142142418)

[Figure 4 – Hierarchical Data Tree 9](#_Toc142142419)

[Figure 5 – Document Model Example 9](#_Toc142142420)

[Figure 6 – Key-Value Database Example 9](#_Toc142142421)

[Figure 7 – Shortest Path Key-Value Data Access 9](#_Toc142142422)

[Figure 8 – Graph Database example Schema 9](#_Toc142142423)

[Figure 9 – Wide Column Database Example 9](#_Toc142142424)

[Figure 10 – The Data Lake Design Pattern 10](#_Toc142142425)

[Appendix B: 10](#_Toc142142426)

[Figure 1 – RDBMS Table: 10](#_Toc142142427)

[Figure 2 – Hierarchical Data Example: 10](#_Toc142142428)

[References 11](#_Toc142142429)

[Glossary 12](#_Toc142142430)

# Table of Figures

[Figure 1 Example RDBMS SQL Table Relations (W3Resource 2022) 2](#_Toc141821737)

[Figure 2 - Database Environments used by Professional Developers - Stack Overflow Survey 2023 3](#_Toc141821738)

[Figure 3 - ACID Principles 3](#_Toc141821739)

[Figure 4 - Hierarchical Data Structure - MariaDB 2015 4](#_Toc141821740)

# Introduction

This report uses a variety of specialized terminology and abbreviations. Explanations are defined in the Glossary.

# Task 1 - Database Technologies

## Relational Database Management Systems (RDBMS)

The RDBMS uses a structure based on the relational model proposed by E.F. Codd in 1970 (Connolly 2015 Part 2 – Chapter 4), to allow us to identify and access data attributes via table-based (Appendix B - Figure 1) relations using unique keys.

A screenshot of a computer

Description automatically generated

Figure 1 Example RDBMS SQL Table Relations (W3Resource 2022)

They are designed to represent complex data schemas and minimize data redundancy through normalization whilst offering performant transactions using SQL[[1]](#endnote-1).

RDBMS vendors, such as *PostgreSQL, MySQL & SQLite* are amongst the most common databases, evidenced below.

A screenshot of a graph

Description automatically generated

Figure 2 - Database Environments used by Professional Developers - Stack Overflow Survey 2023

An excellent reason to use an RDBMS’ is when your transactions must adhere to the ACID principles[[2]](#endnote-2) which most vendors support implicitly, and if your organisation has complex querying requirements.

A screenshot of a computer

Description automatically generated

Figure 3 - ACID Principles

Finance, education & ecommerce systems’ complex structure prefer RDBMSs for their reliable and consistent data storage.

However,’ RDBMS’ performance is hindered by ACID, making them unsuitable for platforms like YouTube (Shivang, 2019). They’re unsuited for unstructured JSON data, prevalent in the Web and schema changes add significant complexity, also leading to a poor representation of “real-world” entities (Connolly 2015, Part 2 – Chapter 9).

## NoSQL – Not Only SQL

### Hierarchical Databases:

Hierarchical Databases, like JSON, display data in a tree-form[[3]](#endnote-3) with unidirectional parent-child links (Appendix B - Figure 2).

A diagram of a network

Description automatically generated

Figure 4 - Hierarchical Data Structure - MariaDB 2015

Highly efficient for read/write operations due to their structure, they excel in explicit hierarchical relationships like File Systems (MariaDB, 2015). Their limitation lies in an inability to support multi-dimensional or many-to-many relationships.

### Document Database

These don’t store entities like Spreadsheets, Instead, they store structured formats like JSON, BSON[[4]](#endnote-4) or XML[[5]](#endnote-5) (Hoffer, 2019).

Each named, key-addressed record stores its data as a “value”. Unlike hierarchical databases, they facilitate many-to-many relationships by referencing another document’s key.

A close-up of a computer code

Description automatically generated

Figure 5 Document Data Model Representation (Pore, 2018)

They are scalable, can handle complex data and have flexible schemas, making them great for CMS’[[6]](#endnote-6) using *MongoDB* or *Amazon DynamoDB* however they lack proper structure and complex queries can be slow.

### Key-Value Database

KVP [[7]](#endnote-7) databases, fully non-relational, are favored by Twitter (Pandori, 2022) & LinkedIn for rapid lookup in distributed cloud environments.

A screenshot of a computer

Description automatically generated

Figure 6 Example of a Key Value database. (Redis 2023)

Utilizing a hash-table[[8]](#endnote-8) to store data, they offer near instantaneous access but don’t support querying the unstructured values (Hoffer, 2019).

A graph with different colored bars

Description automatically generated

Figure 7 DB Providers & DB Types Read Time for Related Nodes (ArangoDB 2018)

Different from other databases, KVDs excel in speed and scalability, yet lack query versatility.

### Graph Database

Employing graph theory[[9]](#endnote-9) to store, and query relationships, they’re advantageous for interconnected relationships, making them ideal for recommendation engines[[10]](#endnote-10), advanced analytics (Neo4J, n.d.) and fraud detection.

A diagram of a person

Description automatically generated

Figure 8 A simple Graph schema example (Morgante 2020)

They’re unique in NoSQL realm for their relational nature, but struggle with scalability and require complex data modelling.

### Wide-Column Database

Excellent for large datasets with simple querying requirements, favoring tables with numerous, varying column structures (Hoffer, 2019) over RDBMS-like table-joins. However, they have a steep learning curve and struggle with complex querying.

A diagram of a personal information

Description automatically generated

Figure 9 A simple example of a Wide-Column Database

They’re often found in data-warehousing for analytics, aggregation [[11]](#endnote-11)and data mining[[12]](#endnote-12).

### Data Lakes

A data lake is an architectural approach, involving a centralized repository that allows storage of structured and unstructured data at any scale.

A diagram of data processing

Description automatically generated

Figure 10 Data Lake Design Pattern (DataKitchen.io, 2017)

Unlike traditional systems, data lakes store raw data, offering flexibility and the ability to integrate with multiple data sources, but demands robust governance for efficient usage.

They enable diverse analytic methods like machine learning[[13]](#endnote-13).

# Task 2 - Database Modelling

## Scenario Overview

Gemini Jeweler’s (GJ), a producer and wholesaler of Jewelry, implements a pre-approval process for individual item orders by company as well as authorized users for ordering. Items, which come in various color finishes and material selections (presently silver, gold and platinum), are currently housed in two depots. However, GJ envisions future expansion and desires to add more locations.

Currently operating a paper-based system to track stock at each depot, customers approved for ordering by item and material options and prices for each item combination, they desire to move to a web-based CRM [[14]](#endnote-14) & ordering system built on a Relational Database.

## Conceptual Modelling

As the scenario provided contains business requirements with no sample raw data, a ‘top-down’[[15]](#endnote-15) data modelling approach will be implemented.

### Entity Identification

Extracted from the specification, the below figure displays the raw entities, without consideration for relationships or attributes.

A grid of rectangles with text

Description automatically generated

Figure 11 Entity Diagram

### Relation Identification

At a high level, the attribute analysis has revealed a relation of the ‘ItemPrice’ with multiple entities, hence its re-extraction to its own entity.

These relations have started to represent tangible meaning and responsibility of each entity and the control-flow of business processes have emerged.

A diagram of a company

Description automatically generated

Figure 12 Entity Relations Identification

### Conceptual Diagram

Using the relations identified above, annotated in the crows-foot notation, a conceptual diagram has been established.

A diagram of a data flow

Description automatically generated with medium confidence

A list of assumptions has been made, based on the current state of modelling:

1. The Price entity can be subsumed into our Item Entity
2. GJ Employee’s require zero relations to the greater CRM & Order Management system.
3. Companies can exist without Users but not vice versa.
4. Items are dependent on at least a single Material and Colour entity.

## Logical Modelling

### Attribute Identification

Extending the rudimentary Entity Diagram and disregarding potential relationships attributes have been added. At this stage, Stock & Price have been reduced to attributes as they had no defining attributes of their own & we’ve discovered a need for a GJ Employee Entity.

‘CK’ denotes a Candidate Key[[16]](#endnote-16), on all entities they are present on, a fallback ‘ID’ attribute was preferred as all CK attributes were potentially non-uniquely defining or susceptible to change.

A screenshot of a computer screen

Description automatically generated

Figure 14 Entity Diagram with Attributes Identified

### Primitive Logical Diagram

In the diagram below, the basic Relations and Attributes outlined above have been adapted into a .

A diagram of a computer

Description automatically generated

# Appendices

## Appendix A:

### Figure 1 – RDBMS Table

A simple representation of a table in a Relational Database Management System.

Adapted from “RDBMS Table Terminologies”, a blogpost by Wentz Wu. 8/07/2019.

### Figure 2 – Stack Overflow Database Environments Survey:

A chart displaying database environments used by professional developers (60,369 respondents). Answers are multiple choice to encapsulate total usage rather than the greatest usage of a specific vender.

Adapted from the 2023 Developer Survey by Stack Overflow.

### Figure 3 – ACID Principles:

A graphic showing an overview of the ACID Principles and their definitions.

Adapted from an independent blogpost by Dave Pinal. 9/12/2007.

### Figure 4 – Hierarchical Data Tree

A simple graphic showing the tree-like structure of hierarchical databases.

Adapted from an article from MariaDB. 06/06/2015.

### Figure 5 – Document Model Example

An image that represents a collection of documents within a document database. Records can contain different data if the overall structure is adhered to.

Adapted from an article from Akshay Pore, 16/2/2018.

### Figure 6 – Key-Value Database Example

An image that shows a human readable example of a Key-Value data store. Using a Name as a Key & a phone number as a Value

Adapted from an article by Redis DB, 06/07/2023.

### Figure 7 – Shortest Path Key-Value Data Access

A chart that displays the read time of related nodes in an inter-connected Social Media database context. ArangoDB(Rocks) is a Key-Value database.

Adapted from an article of benchmarks by ArangoDB, 14/02/2018.

### Figure 8 – Graph Database example Schema

This chart shows a simple graph database, each line represents a relationship between nodes.

Adapted from an article by Victor Morgante on Towards Data Science, 16/09/2020.

### Figure 9 – Wide Column Database Example

An example of a Wide Column store with an example of a User and the data for said user’s information across “column families”.

Adapted from an article from Database Town 29/1/2023.

### Figure 10 – The Data Lake Design Pattern

A design diagram of an architectural approach to integrating and transforming multiple data through a centralized repository known as a ‘Data Lake’.

Adapted from an article from Data Kitchen 17/01/2017.

### Figure 11 – Entity Diagram

A primitive design diagram built to display the Entities found in the specification overview.

### Figure 12 – Attribute Entity Diagram

This diagram, adapted from Figure 11, has been extended to include obvious attributes of the entities, as well as explicit Primary Keys & attributes chosen as Candidate Keys (CKs).

## Appendix B:

### Figure 1 – RDBMS Table:

A diagram of a table

Description automatically generated

A simple example of an RDBMS table and a visual representation of definitions associated with them.

Adapted from an example from W3Resource’s website. 19/08/2022.

### Figure 2 – Hierarchical Data Example:

A screenshot of a computer screen

Description automatically generated

A visual representation of a Hierarchical data structure.

Organization is the Root of the hierarchy, with Departments & Employees as nested child entities. This demonstrates the limitation of the hierarchical data structure, as children can only have a single parent node.

# References

Akshay Pore, 16/02/2018. NoSQL Data Architecture & Data Governance: Everything You Need To Know [article] Available at: <https://www.dataversity.net/nosql-data-architecture-data-governance-everything-need-know/> [Accessed 01/08/2023]

ArangoDB, 14/02/2018. NoSQL Performance Benchmark 2018 – MongoDB, PostgreSQL, OrientDB, Neo4j and ArangoDB. [article] Available at: <https://www.arangodb.com/2018/02/nosql-performance-benchmark-2018-mongodb-postgresql-orientdb-neo4j-arangodb/> [Accessed 03/08/2023]

Connolly, Thomas, 17/4/2015. Database systems: a practical approach to design. Part 2 Chapter 4 – The Relational Model, 6th Edition. Pearson Education. Rotherham, England. [Accessed 31/07/2023]

Connolly, Thomas, 17/4/2015. Database systems: a practical approach to design. Part 2 Chapter 9 – Object-Relational DBMS, 6th Edition. Pearson Education. Rotherham, England. [Accessed 01/08/2023]

Data Kitchen, 17/01/2017. The Data Lake Is a Design Pattern. [article] Available at: <https://medium.com/data-ops/the-data-lake-is-a-design-pattern-888323323c66> [Accessed 05/08/2023]

Database Town, 29/1/2023. Wide Column Database (Use Cases, Examples, Advantages & Disadvantages) [article] Available at: <https://databasetown.com/wide-column-database-use-cases/#wide-column-database-examples> [Accessed 03/08/2023]

Hoffer, Jeff. Modern database management. Part IV Chapter 10 – Big Data Technologies, page 485. Global Edition. Pearson Education. Rotherham, England. [Accessed 01/08/2023]

MariaDB, 06/06/2015. Understanding the Hierarchical Database Model. [article] Available at: <https://mariadb.com/kb/en/understanding-the-hierarchical-database-model/> [Accessed 01/08/2023]

Neo4J, 04/02/2015. Why Graph Database? [online] Available at: <https://neo4j.com/why-graph-databases/> [Accessed 03/08/2023]

Redis, 6/07/2023. What is a Key-Value Database. [article] Available at: <https://redis.com/nosql/key-value-databases/> [Accessed 02/08/2023]

Stack Overflow, June 2023. 2023 Developer Survey. [survey] Available at: <https://survey.stackoverflow.co/2023/#section-most-popular-technologies-databases> [Accessed 01/08/2023]

Sahib Pandori, 28/02/2022. Data transfer in Manhattan using RocksDB – Twitter Blogs. [blog] Available at: <https://blog.twitter.com/engineering/en_us/topics/infrastructure/2022/data-transfer-in-manhattan-using-rocksdb> [Accessed 02/08/2023]

Shivang, 15/12/2019. YouTube Database and how does it store so many videos without running out of storage space. [article] Available at: <https://scaleyourapp.com/youtube-database-how-does-it-store-so-many-videos-without-running-out-of-storage-space/> [Accessed 01/08/2023]

Pinal, Dave, 9/12/2007. SQL Server – ACID (Atomicity, Consistency, Isolation, Durability) [blog] Available at: <https://blog.sqlauthority.com/2007/12/09/sql-server-acid-atomicity-consistency-isolation-durability/> [Accessed 01/08/2023]

Victor Morgante, 16/09/2020. What is a graph database? Towards Data Science. [article] Available at: <https://towardsdatascience.com/what-is-a-graph-database-249cd7fdf24d> [Accessed 03/08/2023]

Wentz Wu, 8/07/2019, RDBMS Table Terminologies, RDBMS Table. [online] Available at: <https://wentzwu.com/2019/07/08/rdbms-table-terminologies/> [Accessed 31/07/2023]

W3Resources, 19/08/2022, SQL Table. [online] Available at: <https://www.w3resource.com/sql/sql-table.php> [Accessed 01/08/2023]

W3Schools, n.d. Introduction to XML. [online] Available at: <https://www.w3schools.com/xml/xml_whatis.asp> [Accessed 01/08/2023]

# Glossary

1. SQL – Structured Query Language

   The standard language of accessing and manipulating Relational Database Management Systems. [↑](#endnote-ref-1)
2. ACID Principles:

   Atomicity, Consistency, Isolation & Durability. A set of principles applied to database transactions to enhance the reliability of data, avoid stateful concurrency issues in transactions and eliminate data loss through critical outages. [↑](#endnote-ref-2)
3. Tree Structure

   A data structure that contains a single root node that can have a recursive number of child elements with only one parent. Called a “Tree” due to its triangular visual representation and because each child node can be referred to as a branch, where each branch has its own branches. [↑](#endnote-ref-3)
4. BSON – Binary JSON

   A binary representation of JSON, often storing metadata about the type & length of data. [↑](#endnote-ref-4)
5. XML – Extensible Markup Language

   A file format for storing, transmitting, and reconstructing arbitrary data. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. (W3Schools, n.d.) [↑](#endnote-ref-5)
6. CMS – Content Management System

   A software system that creates, manages, and modifies digital content on a website or application.] [↑](#endnote-ref-6)
7. KVP – Key-Value Pair

   A name & value pairing. Sometimes called an attribute-value pair. These are fundamental for data representation in computing systems and applications. [↑](#endnote-ref-7)
8. Hash Table

   In computing, a hash map is an abstract data type that is used to store an unordered collection of key-value pairs. Each key is unique which offers extremely fast data retrieval if you have the address of the key ahead of time. [↑](#endnote-ref-8)
9. Graph Theory

   Graph theory is a mathematical field studying graph structures, modeling pairwise relations between objects, characterized by vertices (nodes) and edges (connections) [↑](#endnote-ref-9)
10. Recommendation Engine

    This is a system that suggest products, services or information to users based on analysis of their behaviour. [↑](#endnote-ref-10)
11. Aggregation

    In data warehousing, aggregation refers to the process of combining or grouping data together to provide summarized, more digestible information. [↑](#endnote-ref-11)
12. Data Mining

    Applying a statistical method to create measurable, analysed data, revealing patterns or trends, contributing to model development and predictive analytics. [↑](#endnote-ref-12)
13. Machine Learning

    A subset of artificial intelligence that enables computers to learn and improve from experience without being explicitly programmed. [↑](#endnote-ref-13)
14. CRM – Customer Relation Management System

    A system to help manage customer data to support sales management. [↑](#endnote-ref-14)
15. Top-Down Approach

    In Data Modelling, a top-down approach means you only have a set of business requirements to begin modelling. The opposite ‘Bottom Up’ approaches are taken when you have access to raw data. [↑](#endnote-ref-15)
16. CK – Candidate Key

    An attribute on an entity that could be a good choice for a Primary Key. [↑](#endnote-ref-16)