**Task 1: Relational Database Design and Implementation**

Krescens Kok

Department of Technology, Western Governors University

D597: Data Management

Sean Pettersen

2025 June 12

**Part 1: Design Document**

**A.  Select one of the provided scenarios to complete the following:** Scenario 2

**1.  Describe a business problem that can be solved with a database solution and is in alignment with the chosen scenario.**

EcoMart is an emerging company that is in the marketplace industry. Their goal is to foster sustainability and environmental consciousness. EcoMart has a platform where customers can find ethically sourced, sustainable, and eco-friendly products, such as groceries, apparel, home goods, and personal care items. The business problem that can be solved is to allow flexibility in scaling their platform and also allow for faster lookups of products by the customers, all in an effort to reach their goal of being sustainable. The solution will be to implement a relational database in third normal form. In addition, the database solution will provide robust security measures (encryption, access controls, and audit logging), ensure data consistency, and ensure long-term stability and reliability of the database implementation.

**2.  Propose a data structure to solve the identified business problem.**

A data structure to solve the business problem is to format the data into a relational database. Currently, the data is in a raw csv file, so it will need to be ingested into the database and be normalized into third normal form to eliminate transitive dependencies. Transitive dependencies can cause database issues, data redundancy, anomalies, and inconsistencies (Bhalla, 2023).

In order to solve the business issue with a relational database, the raw data will be ingested, then it will be ensured that all the values are atomic, transitive dependencies will be eliminated, and every non-primary or foreign key column will depend on the primary key of the table by splitting the raw data into multiple tables, such as a table for the region, a table for the country, a table for item types, and a table that holds all the sales record information.

**3.  Justify why a database solution will solve the identified business problem.**

This database solution will solve the business problem because it will allow the database to be scalable and consistent. By normalizing the table, this will make sure data anomalies will occur less frequently as there is only one place to update information instead of multiple different places. It will also ensure data integrity as required fields could be implemented so updates or inserts can’t happen unless those fields are being included. This solution will also help with scalability because new record types or regions can be added without changing the existing table that holds the sales record. Also, by splitting the data into multiple columns, queries can run faster since they will be smaller, which will also allow for faster joins or even lookups. Lastly, with a database implementation, security can be implemented as the server can be protected by a username and password, or even multifactor authentication. Audit logging can also be implemented and it can track when a table was last updated and by which user.

**4.  Explain how the business data will be used within the database solution.**

The business data will be used within the database solution because the tables that are being created are determined by the raw csv file that is provided. Multiple tables will be created to implement this solution, and the data from the csv file will be used to fill those tables. These tables will then be able to be joined with each other so that the business can still understand its sales records. By implementing the database solution, the business can also find all the regions or countries in which they are selling their products by looking at those individual tables. They could understand how many of each type of product they are selling, perform calculations, or aggregate calculations on the data to understand revenue, how many items sold, etc Lastly, multiple different queries can be run to understand more about business insights.

**B.  Create a logical data model for storing data in the database solution.**

**A screenshot of a graph

AI-generated content may be incorrect.**

**C.  Describe the database objects and storage, identifying the file attributes within the database solution.**

There are a total of 4objects/tables: Country, Region, Item\_Type, and Sales\_Records.

Each table has a primary key, for Country, there is a country\_id, for Region there is a region\_id, for item\_type there is a item\_type\_id, and lastly, for sales\_records, the primary key is order\_id. The sales\_records table has foreign keys that join together with the other tables on the ids to get the country name, region name, and item\_type name.

The country table has 2 columns: Country\_ID which is an int and Country, which is a varchar(1000).

The region table has 2 columns: Region\_ID which is an int and Region, which is a varchar(1000)

The item\_type table also has 2 columns: Item\_Type\_ID which is an int and item\_type which is a varchar(1000)

Lastly, the sales\_records table has multiple columns:

Sales\_channel: varchar(500)

Order\_priority: varchar(5)

Order\_date: Date

Order\_ID: Int

Ship\_date: Date

Units\_sold: Int

Unit\_price: Decimal(50,2)

Unit\_cost: Decimal(50,2)

Total\_revenue: Decimal(50,2)

Total\_cost: Decimal(50,2)

Total\_profit: Decimal(50,2)

Region\_ID: Int

Country\_ID: Int

Item\_Type\_ID: Int

**D.  Discuss how the proposed database design addresses scalability concerns, including strategies that align with the chosen scenario.**

The proposed database design addresses scalabilityby making it easier to add more tables and information because the database design is in third normal form. This means that the data are separated into different tables (region, country, item types, and sales orders) and relationships are maintained using primary and foreign key constraints. This reduces data redundancy because certain values are only stored once (ex: regions), which reduces the size of the database as a whole, which increases the query performance even if the database grows in size.

Normalizing the database also allows for easier maintainability because there is less data that needs to be updated when tables are split out. If there are changes that need to be made to the tables that contain additional information, such as country or region, instead of changing all the fields in the singular table with all the information, only the country table or region table would need to be updated. Lastly, if a new region or country is introduced into the data, the country table or region table would have an additional entry, and the main sales order table would use the primary key of the region/country instead of the actual region/country name, allowing the database to scale without complexity.

Lastly, since PostgreSQL automatically indexes primary keys, normalizing the database into multiple related tables ensures that each table has an indexed primary key. This means that the database does a binary search when looking for the keys, instead of scanning each individual row. This helps with the query performance because it’s much faster to find certain keys when the value has been indexed, leading to faster joins and where clauses (Salmany, 2025). For example, when joining the sales\_orders table together with the region table on region\_id, this join performs faster because region\_id on the region table is indexed. Therefore, as the database grows, the solution is still scalable, and queries will still perform well.

**E.  Outline the privacy and security measures that should be implemented in the proposed database design.**

Any data should be protected, especially since it contains company information. So, privacy and security measures that should be implemented could mean that the Database requires a login and 2 factor authentication. That way only verified users can view the data, and in addition, each user could have a different set of permissions. For example, engineers could have write, read, and update permissions, while analysts may only have read permissions. By having a login, this ensures privacy and security as only verified users can view the data and it can’t be accessed by the public. Another measure that could be implemented is audit logging. This will ensure that all past operations are recorded, so if something goes wrong, it can be tracked to when and what caused the issue. Lastly, another measure that could be implemented is stress testing and database security testing (*What Is Database Security Testing - Complete Guide*, n.d.). This helps discover new ways that the database could be hacked and will allow for implementations to be in place before it becomes an issue.

**Part 2: Implementation**

**F.  Implement the proposed database design in the WGU Virtual Lab environment by completing the following:**

**1.  Write script to create a database instance named “D597 Task 1” using the appropriate query language, based on the logical data model in part B. Provide a screenshot showing the script and the database instance in the platform.**

A screenshot of a computer

AI-generated content may be incorrect.

**2.  Write script to import the data records from the chosen scenario CSV files into the database instance. Provide a screenshot showing the script and the data correctly inserted or mapped into the database.**

Creating the initial table:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.Inserting/Importing the data:

**A screenshot of a computer

AI-generated content may be incorrect.** Creating the Region table:

A screenshot of a computer

AI-generated content may be incorrect.Creating the Country table:

Creating the item\_types table:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.The next step is to alter the original table we created (sales\_record) and normalize it by using the additional tables we created. First, we’ll add the ID columns (foreign keys).

A screenshot of a computer

AI-generated content may be incorrect.Now we can populate those fields with the values from the other tables we created, by doing an update query and matching it on the region name, country name, and sales record type

A screenshot of a computer

AI-generated content may be incorrect.Next, we’ll drop the region, country, and item\_type columns from the original table (sales\_records)

Lastly, we’ll need to specify that the region\_id, country\_id, and item\_type\_id are foreign keys since they are used to join with the other tables we created

A screenshot of a computer

AI-generated content may be incorrect.

**3.  Write script for three queries to retrieve specific information from the database that will help to solve the identified business problem. Provide a screenshot showing the script for each query and each query successfully executed.**

A screenshot of a computer

AI-generated content may be incorrect. Find the top 3 item types to help determine if they are sustainable and environmentally friendly. This will also help the business understand what customers are mostly buying and if they are reaching their goal of fostering sustainability and environmental consciousness:

Find the average time between the order date and the ship date. This will help the business focus on what they can do to improve, decrease the time, and increase efficiency. By doing so, this could increase their sales and promote their company to continue to foster and promote sustainable products:

A screenshot of a computer

AI-generated content may be incorrect.

Order the regions by total\_profit ascending to show which regions have the lowest total profit. This will help the business understand which areas they may need to focus marketing in to increase their sales and profit. By finding the regions that aren’t buying many products, the business can also understand whether or not the region needs more knowledge on eco-friendly products:

A screenshot of a computer

AI-generated content may be incorrect.

**4.  Apply optimization techniques to improve the run time of your queries from part F3, providing output results via a screenshot.**

For the first query, indexes were created on item\_type\_id from the sales\_records table and on since the item\_type\_id from the item\_types table is a primary key, it was already indexed. This allowed for a faster lookup time and join time between the sales\_records and item\_types table because the query no longer needed to look through the entire table. Instead, it uses a hash method where it can skip to the rows that match the item\_type\_id. In addition, an inner join was used instead of a left join, which made the query faster because the database didn’t have to keep track of the null records that wouldn’t match in the join clause (GeeksforGeeks, 2025). Lastly, the group by was done on the primary key of the item\_types table, which helps with optimization because the key is an integer type, which means that they are stored as a fixed binary length. Grouping by a string would take longer because each character would need to be checked.

A screenshot of a computer

AI-generated content may be incorrect.

For the second query, the same concept was applied where indexes were applied on the foreign key; Item\_type\_id on the sales\_records table, since the order\_id on the sales\_records table and item\_type\_id on the item\_types table are both primary keys, these columns are already indexed. This allowed for a faster join and lookup between the sales\_records and item\_types table due to hashing and jumping to the item\_type\_ids that match. In addition, an inner join and grouping by the primary key were used again.

**A screenshot of a computer

AI-generated content may be incorrect.**

For the third query, the same idea was applied where an index was created on the region\_id on the sales\_records table and since the region\_id on the region table is a primary key, it is already indexed. The group by was updated to group by the primary key, region\_id, instead of the region name. Since primary keys are indexed and, in this case, the primary key is an integer, it allowed the grouping to perform faster since the look up was faster and the comparison was easier to compute since the query was comparing 2 integers instead of comparing letter by letter. Lastly, an inner join was updated from a left join to increase the query time and also allowing for less storage use for null or unmatched records.

**A screenshot of a computer

AI-generated content may be incorrect.**

First query: .091 vs .063

Second query: .105 vs .078

Third query: .082 vs .057

**References**

Bhalla, H. (2023, March 3). Why every developer should care about Transitive Dependency? *Medium*. <https://medium.com/@himankbh/why-every-developer-should-care-about-transitive-dependency-34f24e9e9c08>

GeeksforGeeks. (2025, June 27). *SQL joins (Inner, left, right and full join)*. GeeksforGeeks. <https://www.geeksforgeeks.org/sql/sql-join-set-1-inner-left-right-and-full-joins/>

Salmany, N. (2025, March 28). How indexing Enhances query Performance - Digma. *Digma*. https://digma.ai/how-indexing-enhances-query-performance/

*What Is Database Security Testing - Complete Guide*. (n.d.).

www.softwaretestinghelp.com. <https://www.softwaretestinghelp.com/database->

security-testing/