EUROPEAN HIGHER EDUCATION INSTITUTE

Data Management - Final Exam Submission

Prepared by: Çiçek Akkaya

Your company is implementing an information system to manage a new on-line food delivery business.

Your boss has purchased a list of potential customers from a lead generation company. The names are:

• Mouse, Mickey living in Mouseton – United States of America

• Mouse, Minnie living in Mouseton - United States of America

• Duck, Donald living in Duckburg - Canada

• Duck, Daisy living in Duckburg - Canada

• McDuck, Scrooge living in Duckburg – Canada

The products to sell are:

• Hamburgers, price 12$ each

• Hot dogs, price 9$ each

• Apple pies, price 30$ each

• Pancakes, price 3$ each

• Crispbreads, price 0,05$ each

Orders soon begin to arrive:

• On June 21, Mouse, Mickey buys 3 hamburgers and 5 hot dogs

• On June 22, Mouse, Minnie buys 50 crispbreads and 1 apple pie

• On June 22, Duck, Donald buys 36 pancakes

• On June 23, Duck, Daisy buys 3 apple pies

• On June 24, Mouse, Mickey buys 5 hamburgers

• On June 24, McDuck, Scrooge buys 10 crispbreads

The company also undertakes a marketing campaign to distribute discount coupons and inform about its commercial initiatives.

EXERCISES:

1 - Analyze the dataset from the Data Manager's point of view

What purposes can you have with this data? What kind of database is best suited for these purposes? Classify the data (by type, data source, etc.).

The data can be leveraged for several key purposes:

- Operational: The primary goal is to manage the day-to-day business. This includes processing orders, tracking which customer bought what products, and calculating order totals for billing.
- Analytical (Business Intelligence): By analyzing trends, you can make smarter business decisions. This involves answering questions like:
    - Who are our most valuable customers? (e.g., Mickey Mouse has placed two orders already).
    - What are our best-selling products? (e.g., Hamburgers vs. Crispbreads).
    - Are there geographic trends? (e.g., Are Canadian customers ordering different items than US customers?).
    - When are our peak order times?
- Marketing & Personalization: The data is crucial for targeted marketing. We can segment customers based on their location or purchase history to send them relevant discount coupons and promotions. For example, we could send a special offer on apple pies to Daisy Duck.

Best-Suited Database

A Relational Database (managed by an RDBMS like PostgreSQL, MySQL, or SQL Server) is the best choice for this scenario.

The reasons are:

- Structured Data: The data has a clear, predictable structure (customers, products, orders).
- Data Integrity: Relational databases enforce rules (like ensuring an order is always linked to a valid customer) which prevents data corruption. This is critical for a transactional system involving money.
- Reduces Redundancy: A well-designed relational database ensures that information is stored only once, saving space and preventing inconsistencies.
- Powerful Queries: Using SQL (Structured Query Language), you can easily join tables to answer the complex analytical questions mentioned above.

Data Classification

The data can be classified as follows:

- By Source:
    - External Data: The initial customer list was purchased from a lead generation company.
    - Internal Data: All product information, orders, and marketing campaign details are generated by the company's own operations.

- By Data Type:
  - For CUSTOMERS table:
    - Full_Name - nvarchar(40) (For example: Mickey Mouse)
    - Address - nvarchar (100) (For example: Mouseton – United States of America)
    - Customer_ID - int (For example: 1,2,3, ...)
  - For PRODUCTS table:
    - Item_ID - int
    - Item – nvarchar (40) (For example: Hamburger)
    - Price - decimal (10, 2) (For example: 100,50$)
  - For ORDERS table:
    - ORDER_ID - int
    - OrderDate - Date (2025-06-23)
    - Customer_ID – int
  - For ORDER_ITEMS table:
    - ORDER_ITEMS_ID - int
    - ORDER_ID - int
    - Item_ID - int
    - Order_Item_Qty – int
    - PriceAtSale - decimal (10, 2)
- By Structure:
    - Structured Data: All the information fits neatly into rows and columns in a table format.
- By Content Type:
    - Personally Identifiable Information (PII): Customer names and addresses (Mickey Mouse, Mouseton - United States of America). This data is sensitive and must be handled securely.
    - Transactional Data: The order history, which captures the events of the business.
    - Product Data: Information about the items for sale.

2 - Design the database

What tables would you create? With what columns? Specify primary keys and decide table relationships. What kind of relationships are you creating?

(The database should be in 3NF).

To create an efficient and scalable database that avoids data redundancy, we will design it in the Third Normal Form (3NF). This involves creating separate tables for each distinct entity: Customers, Products, and Orders. Because an order can contain multiple products and a product can be in multiple orders, we need a linking table (Order_Items) to manage this relationship.

Column names for Customers table which stores information about each unique customer:

- Full_Name - nvarchar(40) (For example: Mickey Mouse)
- Address - nvarchar (100) (For example: Mouseton – United States of America)
- Customer_ID - int (For example: 1,2,3, ...)

Column names for Products table stores information about each unique product available for sale:

- Item_ID - int
- Item – nvarchar (40) (For example: Hamburger)
- Price - decimal (10, 2) (For example: 100,50$)

Column names for Orders table stores high-level information about each order placed:

- ORDER_ID - int
- OrderDate - Date (2025-06-23)
- Customer_ID – int

Column names for Order_Items table which is a linking table that connects the Orders and Products tables:

- ORDER_ITEMS_ID - int
- ORDER_ID - int
- Item_ID - int
- Order_Item_Qty – int
- PriceAtSale - decimal (10, 2)

Primary keys:

- Primary key of Customers table is Customer_ID column.
- Primary key of Products table is Item_ID column.
- Primary key of Orders table is Order_ID column.
- Primary key of Order_Items table is Order_Items_ID column.

Table Relationships:

The design uses two types of relationships to link the tables together logically.

- One-to-Many Relationship:
    - A single customer can place many orders. The link is between Customers.Customer_ID and Orders.Customer_ID.
    - A single order can contain many product items. The link is between Orders.Order_ID and Order_Items.Order_ID.
    - A single product can be included in many order items. The link is between Products.Item_ID and Order_Items.Item_ID.

- Many-to-Many Relationship:
    - The underlying business rule is that an order can contain many products, and a product can appear in many orders. This many-to-many relationship is resolved by using the Order_Items linking table, which breaks it down into two one-to-many relationships.

This design is in 3NF because all columns in each table depend directly on that table's primary key and nothing but the primary key, eliminating data redundancy and improving data integrity.

We can apply these relationships with foreign key constraints.

3 - Implement the database

Following the logical schema designed in task 2, write the SQL code to:

• Create the database (name: TEST_DATAMANAGEMENT)

• Create the tables

• Insert the data

• Count how many orders you have received

• List the orders reporting the date, item, quantity, price and total amount of the sale

You are encouraged to test your SQL Queries on SQL Server Management Studio.

I have implemented the database by following the logical schema designed in task 2 with writing SQL codes in the SQL Server Management Studio.

Here is the drive link of the related database:

- https://drive.google.com/file/d/1pM3sTpe6lSbz9N-DGoc0Kslpx0u1aq9B/view?usp=sharing

4 - Disaster recovery

Design a backup plan for your database. Specify frequency, backup approach, and destination(s) and explain why you would adopt this strategy.

This database, which handles transactional data like CUSTOMERS, PRODUCTS, and ORDERS, requires a disaster recovery plan that prioritizes a low Recovery Point Objective (RPO) which is the maximum tolerable data loss, due to the continuous flow of new orders.

The recommended backup plan employs a tiered approach following the 3-2-1 rule for maximum resilience.

Table 1 – General Design of the Backup Plan

Backup Type	Frequency	Approach	Purpose
Full Backup	Weekly (Sunday night)	Complete copy of the entire database.	Provides a reliable, complete baseline for all recovery operations.
Differential Backup	Daily (Monday to Saturday night)	Captures all changes made since the last successful Full Backup.	Faster than daily full backups and more efficient for daily restorations.
Transaction Log Backup	Every 15 minutes (during business hours)	Captures all transaction log entries since the last log backup.	Enables point-in-time recovery (low RPO), critical for transactional data.

The strategy adheres to the 3-2-1 rule which is 3 copies of data, on 2 different media, with 1 copy stored off-site.

Table 2 - Destinations of the Backup Plan

Destination of Copy	Media/Location	Purpose
Primary (Local)	High-speed Local Storage Area Network (SAN) or Network Attached Storage (NAS)	Used for the fastest operational recovery (e.g., restoring a recently corrupted table or minor data loss).
Secondary (Cloud)	Encrypted, geographically separate cloud object storage (e.g., Google Cloud Storage)	Off-site copy. Used for disaster recovery scenarios (e.g., a total physical server or site failure).
Tertiary (Archive)	Separate, air-gapped storage or an alternate cloud region/provider	Used for long-term retention, regulatory compliance, and protection against ransomware attacks (since it is disconnected/immutable).

The strategy is adopted to achieve a balance between a very low Recovery Point Objective (RPO) and a fast Recovery Time Objective (RTO), while ensuring resilience.

The primary justification for the Low RPO via Log Backup with 15-minute log backups is to minimize data loss. In the event of a failure, the database can be recovered to a point in time no more than 15 minutes before the failure, preventing significant loss of new ORDER data.

One of the justifications of the RTO via Full and Differential Backups are Full Backups provide the baseline. Additionally, Differential Backups save time and storage compared to running a Full Backup every night. To perform a daily restore, you only need two files: the most recent weekly Full Backup and the most recent daily Differential Backup. This makes the daily restore process relatively quick and straightforward (low RTO).

Lastly, the 3-2-1 Rule provides resilience. The multiple destinations ensure that no single failure can lead to catastrophic data loss. Therefore, the local copy facilitates quick, everyday restores and the off-site cloud copy protects against a physical disaster at the primary data center. Lastly, the archival/air-gapped copy acts as a final safeguard against systemic failures or data corruption events (like a virus or ransomware), ensuring a clean rollback point is always available.

5 - GDPR compliance

What actions would you take to ensure GDPR compliance for the database, and in the marketing campaign?

I. Database Compliance Actions (Technical & Security)

These actions focus on ensuring the database itself (where customer, order, etc., data is stored) aligns with core GDPR principles.

1. Data Security and Confidentiality

- Encryption and Pseudonymization: All Personally Identifiable Information (PII) stored in the database (names, email, address, etc.) must be encrypted while at rest. For analytical or testing purposes, implement pseudonymization (masking/hashing PII) to reduce risk.
- Access Control: Enforce the Principle of Least Privilege. Database access must be restricted strictly to personnel who need it to perform their jobs, using role-based access control (RBAC).
- Audit Logs: Maintain comprehensive audit logs tracking who accessed which data, when, and for what purpose, ensuring accountability.

2. Data Minimization and Retention

- Data Minimization: Only collect and store data that is strictly necessary for the defined, explicit, and legitimate purposes (e.g., fulfilling an order or sending a consented marketing email).
- Retention Policy: Establish a clear data retention policy. For instance, implement automated processes to anonymize or securely delete customer data a set period (e.g., 3 years) after the account is closed or after the last meaningful interaction (order/login).

3. Data Subject Rights Preparedness

- Right to Erasure (RTF): Establish reliable procedures and tools to permanently delete all associated personal data (including from backups, within a reasonable timeframe) when a customer exercises their "right to be forgotten."
- Right to Access (DSAR): Create a mechanism to quickly and securely provide the customer's data in a structured, commonly used, and machine-readable format (e.g., JSON or CSV) upon request.

II. Marketing Campaign Compliance Actions (Consent & Transparency)

These steps address the most critical area for marketers under GDPR: the collection and use of personal data for promotional activities.

1. Clear and Documented Consent

- Active Opt-In: Marketing emails must be sent only with affirmative, explicit consent. Use clear, unticked boxes where the user must perform an action (like clicking the box) to subscribe. Pre-checked boxes are illegal.
- Granular Consent: Offer separate, detailed consent options for different marketing types (e.g., email newsletters, SMS offers, third-party sharing).

- Record of Consent: Maintain a clear, auditable log of *when*, *how*, and *what exact text* the consent was given against. This is essential for proving compliance.

2. Transparency and Privacy Notice

- Accessible Privacy Notice: Provide an easily accessible link to the Privacy Policy/Notice at every data collection point (website forms, checkout pages, etc.).
- Clarity on Processing: The notice must clearly state what data is being collected, why (the purpose, e.g., product promotion), and the lawful basis (e.g., explicit consent) being relied upon.

3. Ease of Withdrawal of Consent

- Single-Click Unsubscribe: Every single marketing communication must include a clear and prominent "Unsubscribe" link that allows the recipient to withdraw consent with a single click.
- Timely Execution: The withdrawal of consent must be processed without undue delay (ideally immediately) to ensure the user receives no further communications.

4. Profiling and Automated Decision-Making

- If the campaign uses automated profiling to segment customers or target ads, users must be informed of this activity and be given the right to object to such processing.

6 - Marketing campaign data

Where would you store the data collected with the marketing campaign? (names, email addresses, telephone numbers, consents to the various marketing purposes).

The alternatives are:

1. Relational database

2. NO-SQL database

3. Data Warehouse

4. Data Mart

5. Data Lake

6. Data Lakehouse

Select the most suitable structure and motivate your choice.

- The Most Suitable Structure is the Relational Database. Because the collected marketing data, often referred to as Customer Master Data or PII (Personally Identifiable Information), is highly structured and requires absolute data integrity, making a Relational Database Management System (RDBMS) the ideal choice.
- The choice of an RDBMS is driven by three critical requirements for managing customer PII and consent records: Data Integrity, Compliance, and Query Performance.
- GDPR compliance requires absolute certainty regarding a user's consent status. An RDBMS ensures this through ACID properties (Atomicity, Consistency, Isolation, Durability), which are fundamental to transactional data.

- For example, the consistency guarantees that any change (e.g., a user unsubscribing or updating their email) moves the database from one valid state to another. This is crucial for accurately tracking consent status under GDPR.
- An RDBMS allows you to enforce data types (ensuring an email field actually contains text formatted as an email) and relationships (ensuring a consent record is always linked to an existing customer record).
- PII (Name, Email, Phone) and consent records (Opt-in date, purpose, boolean flag) are inherently structured data. An RDBMS is explicitly designed to handle this rigid, predictable schema efficiently.
- RDBMS allows for robust indexing on key fields (like the customer ID or email address). This is vital for real-time operations, such as checking consent status before sending a marketing email and quickly retrieving all data related to a customer when they exercise their Right to Access (DSAR).
- The system needs to handle frequent, fast, small-scale transactions (inserting a new lead, updating an address, recording an unsubscribe action). An RDBMS excels at these Online Transaction Processing (OLTP) tasks, unlike Data Warehouses or Data Lakes, which are designed for large-scale analytical processing (OLAP).