Database System Overview

Aidan Polivka

CS660 – Database Systems

Colorado Technical University

Dr. John Conklin

April 21, 2024

Table of Contents

Database System Overview (Week 1)

The purpose of this assignment is to create a course of action to develop and implement a purposeful database management system for a fictional company of our choice. Throughout my time at Colorado Technical University, I've developed a fictional bookstore called Heartland Escapes. Previous projects include a migration plan from an on-premises system to an Azure Cloud environment, a system networking plan for cloud computing, and a system security management and maintenance plan. I think it's only natural to continue working with this fictional store here and develop a purposeful data environment for this business as well.

As stated previously, Heartland Escapes is a bookstore in Lincoln Nebraska with two locations. They have seen recent growth and consumer interest due to their growing presence in social media. Their business model is heavily centered around hosting events at their stores, including author meet and greets, summer reading programs, holiday-oriented events (like scary story readings on Halloween, Santa reads Christmas stories), and many others. Because of their marketing of these events, they've received a lot of local publicity and foot traffic in store. With the growth they've seen over the past year, Heartland Escapes would like to expand to two neighboring cities in Nebraska with a new location in Omaha Nebraska, and a new location in Beatrice Nebraska. To perform this feat, they have the desire to upgrade their existing technology to better support their employees and customers, which includes and improvement to their existing database management system and the development of an e-commerce marketplace.

Business Environment

Heartland Escapes started out as a form of community necessity. The public libraries were well stocked with books but lacked in providing young people with the spark to enjoy reading. The owners of the company are avid book readers themselves and have always enjoyed sharing their literary journeys with other like-minded individuals. Heartland Escapes mission is to be a shared “Escape” for others to enjoy captivating stories and their love for books with each other. The first store opened in 2010, and it was a slow start to get people in the door with competing companies like Barnes and Noble around. As people began seeing the benefits of community-oriented storytelling, traction accrued to the point that Heartland Escapes had the financial stability to open a second store in 2017. The first year of the pandemic was difficult for the company, but after the guidelines loosened and the public started feeling more comfortable leaving their homes, business caught its stride once again. Since then, Heartland Escapes has seen nothing but growth and enthusiastic customers.

Both stores are equipped with 12 staff, 2 assistant managers, one manager, and one of the owners acts as a general manager of both locations. The other original owner is responsible for human resources and supports the GM with day-to-day operations. There are two additional stakeholders that fit the roles of chief financial officer and chief information officer.

When Heartland Escapes started their journey, they developed their own point-of-sale system and inventory management API. Married to these systems are the Accounting Database and the Inventory Database. Additionally, they have a public facing website that users can search store inventory with and see store hours and event schedules. These home-grown systems are all still being used to this day. As their store has risen in popularity, Heartland Escapes junior and senior staff have all began to feel growing pains. These pains include:

1. Customer Management
2. Inventory Control
3. System Documentation
4. Lack of Online Capabilities
5. Lack of Events Management

As stated previously, Heartland Escapes has the desire to offer e-commerce services to their customers, which will require customer data storage and management along with more robust inventory controls. In previous meetings with Heartland Escapes senior management, they've expressed the desire to offer subscription boxes to interested customers that contain Heartland Escapes books and merchandise. Additionally, the pandemic made stakeholders incredibly nervous. Foot traffic through the store is the main avenue through which Heartland Escapes makes its revenue. If customers had the ability to order books online, that may have lessened the financial impact of the pandemic on their stores.

Advancement in inventory controls would allow for optimal stock of products and automated re-stocking of books. With the impending expansion of their business, these functions will be crucial to maintain between all four stores. The original in-house software and database systems are not well documented, which makes maintenance and upgrades difficult to perform.

Because their business model is very event based, Heartland Escapes desires the ability to post events more easily on their website. As it is now, customers need to subscribe to Heartland Escapes social media channels to stay up to date with up-coming events. Although posts to their social media profiles will continue, having a central place to see all event schedules for each store would be incredibly helpful to their customers when Heartland Escapes opens their two new locations.

Database Goals and Objectives

A Database Management System's purpose is to support and serve the applications through which Heartland Escapes users interact. This is inclusive between all applications and individuals, administrators, customers, and staff. Our database goals for Heartland Escapes are to support their expansion interests in the best way we can. With this in mind, here is our list of objectives as database administrators:

1. Support an enhanced customer management system.
2. Support for subscription-based product management.
3. Support a flexible event management system.
4. Improve and automate replenishment and ordering processes.
5. Protect customer data using regulations provided by the Payment Card Industry Data Security Standard.
6. Architect our databases in such a way that they support scaling and growth.
7. Utilize industry data security standards using role-based access controls.
8. Facilitate data loss prevention through use of regular backup schedules and data replication processes.

How the proposed database system addresses business problem(s)

Heartland Escapes is seeing unprecedented growth in their business and feels that the best way to capitalize on this growth is to expand both physically, and technologically. With this expansion effort, they've verbalized their desire to support customer management, customer online sales, subscription-based products, smart and automated inventory management, and store-to-store event management. They've also expressed that their current inventory management system is lack-luster and fails to accommodate any of these features. Our database goals explicitly state the desire to support these efforts requested by Heartland Escapes stakeholders.

Mission and Goals Alignment

Heartland Escapes' primary mission is to provide their communities with a space for individuals to fall in love with reading. Within this larger mission statement lies many objectives and opportunities to better achieve this goal. In the eyes of Heartland Escapes stakeholders, the current optimal course of action is to provide a central area that customers can receive information about upcoming events, offer consistent availability of books to customers, allow for customers to order books from the comfort of their own homes, and build customer engagement by offering subscription-based products. Our goals as database administrators are derived from the goals of Heartland Escapes, with our primary goal being to support Heartland Escapes in their current endeavor.

Analysis of how the project fulfills the mission/goals of the target organization.

In the Problem Based Learning scenario, the example company is looking to:

1. improve customer satisfaction by allowing them to order business products online.
2. Reduce employee turnover rate by improving system efficiency and useability (therefore also reducing customer complaints).
3. Take advantage of back-office system operation automation.
4. Improve back-office systems across the board.

I believe that the narrative I've written for the fictional Heartland Escapes business fully emulates the same (if not similar) concerns as the Problem Based Learning scenario. I also believe that the goals I've outlined for Heartland Escapes, and the goals I've expressed for the database administration team are in sync with one another. Later sections of this assessment will further explain how these goals will be achieved for Heartland Escapes.

Entity Relationship Model (Week 2)

Subjects of Interests

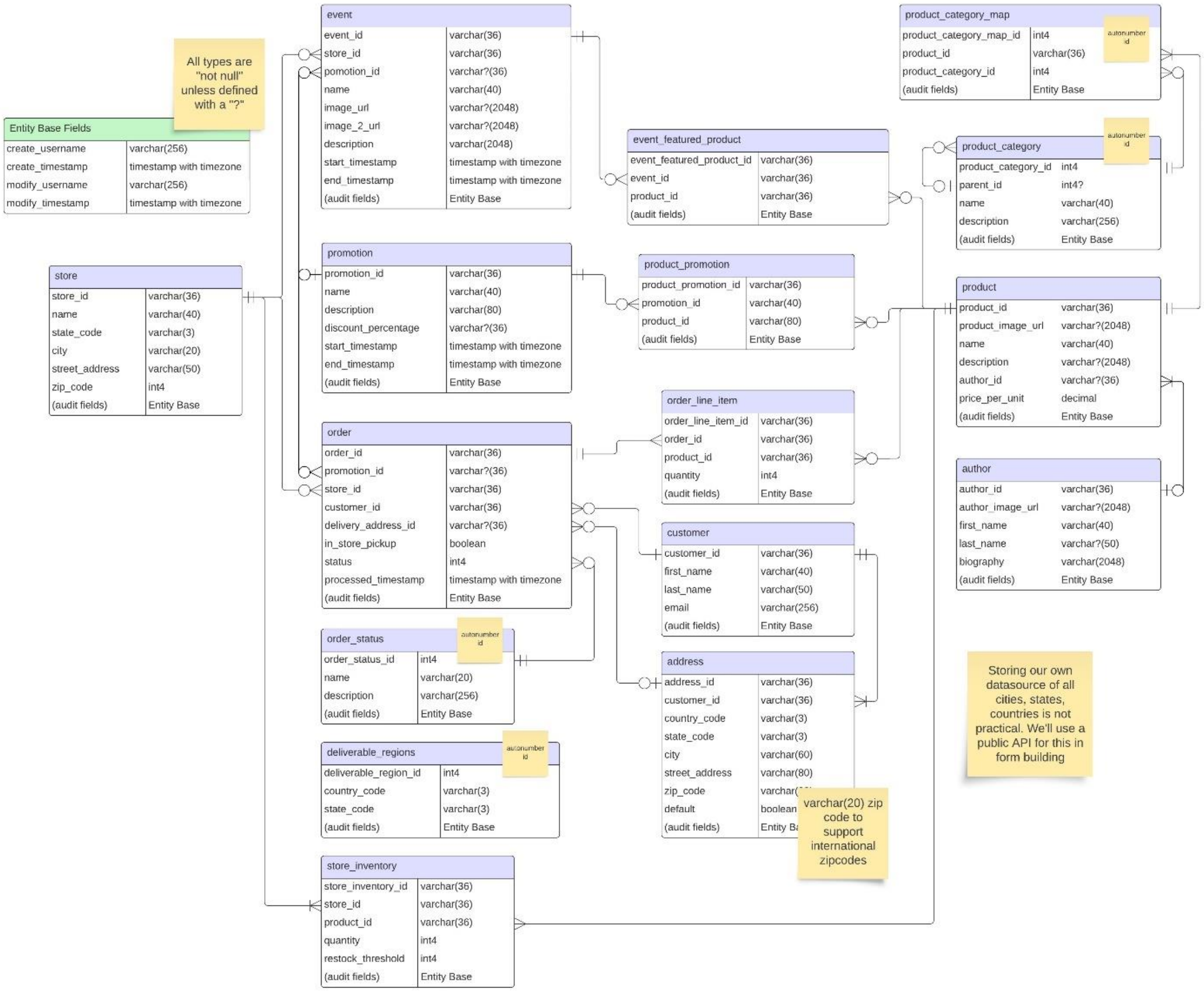
There are several entities in this preliminary entity relationship model. The primary subjects of interest are Stores, Customers, Orders, Products, Events, Promotions and Store Inventory. Because our client wants to begin the process of supporting an e-commerce site, we need to build a robust and flexible entity relationship model that supports all these primary subjects of interest. It also needs to be open to expansion. This includes a lot of supporting entity tables, data link tables, and tables explicitly used to enforce business rules.

Business Rules

- A store must have its own inventory of products.
- Each product must have at least one store inventory
- Each store inventory product must have a minimum restock threshold
- Each store must have a unique name.
- Each Customer must have at least one address.
- Each Product has a minimum of one category.
- Each Order has a minimum of one line-item.
- Each Order has one and only one status.
- Status must exist within the order_status table.
- Each Order must have an originating store.
- Additional business rules are defined below by entity relationships

Entity Relationship (ER) Diagram

This is the entity relationship diagram using crows foot notation. Each column in each table is by default not nullable, unless defined otherwise with a "?" character after the type definition. Each table contains audit fields defined by the structure titled "Entity Base Fields".



All types are "not null" unless defined with a "?"

Entity Base Fields	
create_username	varchar(256)
create_timestamp	timestamp with timezone
modify_username	varchar(256)
modify_timestamp	timestamp with timezone

store	
store_id	varchar(36)
name	varchar(40)
state_code	varchar(3)
city	varchar(20)
street_address	varchar(50)
zip_code	int4
(audit fields)	Entity Base

event	
event_id	varchar(36)
store_id	varchar(36)
pomotion_id	varchar?(36)
name	varchar(40)
image_url	varchar?(2048)
image_2_url	varchar?(2048)
description	varchar(2048)
start_timestamp	timestamp with timezone
end_timestamp	timestamp with timezone
(audit fields)	Entity Base

promotion	
promotion_id	varchar(36)
name	varchar(40)
description	varchar(80)
discount_percentage	varchar?(36)
start_timestamp	timestamp with timezone
end_timestamp	timestamp with timezone
(audit fields)	Entity Base

order	
order_id	varchar(36)
promotion_id	varchar?(36)
store_id	varchar(36)
customer_id	varchar(36)
delivery_address_id	varchar?(36)
in_store_pickup	boolean
status	int4
processed_timestamp	timestamp with timezone
(audit fields)	Entity Base

order_status	
order_status_id	int4
name	varchar(20)
description	varchar(256)
(audit fields)	Entity Base

autonumber id

deliverable_regions	
deliverable_region_id	int4
country_code	varchar(3)
state_code	varchar(3)
(audit fields)	Entity Base

autonumber id

store_inventory	
store_inventory_id	varchar(36)
store_id	varchar(36)
product_id	varchar(36)
quantity	int4
restock_threshold	int4
(audit fields)	Entity Base

event_featured_product	
event_featured_product_id	varchar(36)
event_id	varchar(36)
product_id	varchar(36)
(audit fields)	Entity Base

product_promotion	
product_promotion_id	varchar(36)
promotion_id	varchar(40)
product_id	varchar(80)
(audit fields)	Entity Base

order_line_item	
order_line_item_id	varchar(36)
order_id	varchar(36)
product_id	varchar(36)
quantity	int4
(audit fields)	Entity Base

customer	
customer_id	varchar(36)
first_name	varchar(40)
last_name	varchar(50)
email	varchar(256)
(audit fields)	Entity Base

address	
address_id	varchar(36)
customer_id	varchar(36)
country_code	varchar(3)
state_code	varchar(3)
city	varchar(60)
street_address	varchar(80)
zip_code	varchar(
default	boolean
(audit fields)	Entity Ba

varchar(20) zip code to support international zipcodes

product_category_map	
product_category_map_id	int4
product_id	varchar(36)
product_category_id	int4
(audit fields)	Entity Base

autonumber id

product_category	
product_category_id	int4
parent_id	int4?
name	varchar(40)
description	varchar(256)
(audit fields)	Entity Base

autonumber id

product	
product_id	varchar(36)
product_image_url	varchar?(2048)
name	varchar(40)
description	varchar?(2048)
author_id	varchar?(36)
price_per_unit	decimal
(audit fields)	Entity Base

author	
author_id	varchar(36)
author_image_url	varchar?(2048)
first_name	varchar(40)
last_name	varchar?(50)
biography	varchar(2048)
(audit fields)	Entity Base

Storing our own datasource of all cities, states, countries is not practical. We'll use a public API for this in form building

Table of Entities, Attributes (with data types), Relationships, and Cardinality Constraint

Entity	**Attributes**	**Relationships**	**Cardinality**
customer	customer_id (PK) (varchar(36))	Has Addresses Has Orders	1:Many 1:Many
customer	first_name (varchar(40))		
customer	last_name (varchar(50))		
customer	email (varchar(256))		
product	product_id (PK) (varchar(36))	Can have product_promotions Has at least one product_category_map Can have order_line_items Has at least one store_inventory Can be associated with an Author	1:Zero or More 1:Many 1:Zero or More 1:Many 1:Zero or One
product	product_image_url (varchar?(2048))		
product	name (varchar(40))		
product	description (varchar?(2048))		
product	author_id (varchar?(36))	Can have author	1:Zero or One
product	price_per_unit (decimal)		
store	store_id (PK) (varchar(36))	Can have events Can have orders Has store_inventories	1:Zero or More 1:Zero or More 1:Many
store	name (varchar(40))		
store	state_code (varchar(3))		

store	city (varchar(20))		
store	street_address (varchar(50))		
store	zip_code (int4)		
store_inventory	store_inventory_id (PK) (varchar(36))	Is associated with a product Is associated with a store	1:1 1:1
store_inventory	store_id (FK) (varchar(36))	Is associated with a store	1:1
store_inventory	product_id (FK) (varchar(36))	Is associated with a product	1:1
store_inventory	quantity (int4)		
store_inventory	restock_threshold (int4)		
address	address_id (PK) (varchar(36))	Has a customer Can be associated with an order	1:1 1:Zero or More
address	customer_id (FK) (varchar(36))	Has a customer	1:1
address	country_code (varchar(3))		
address	state_code (varchar(3))		
address	city (varchar(60))		
address	street_address (varchar(80))		
address	zip_code (varchar(20))		
address	default (boolean)		
order	order_id (PK) (varchar(36))	Has at least one order_line_item	1:1 or more

		Has a customer May have a delivery address Has a store Has an order_status May have a promotion	1:1 1:Zero or 1 1:1 1:1 1:Zero or 1
order	promotion_id (FK) (varchar?(36))	May have a promotion	1:Zero or 1
order	store_id (FK) (varchar(36))	Has a store	1:1
order	customer_id (FK) (varchar(36))	Has a customer	1:1
order	delivery_address (varchar?(36))	May have a delivery address	1:Zero or 1
order	in_store_pickup (boolean)		
order	status (int4)	Has an order_status	1:1
order	processed_timestamp (timestamp with timezone)		
order_status	order_status_id (PK) (int4)	May have multiple orders	1:Zero or More
order_status	name (varchar(20))		
order_status	description (varchar(256))		
order_line_item	order_line_item_id (PK) (varchar(36))	Is associated with an order Is associated with a product	1:1 1:1
order_line_item	order_id (FK) (varchar(36))	Is associated with an order	1:1
order_line_item	product_id (FK) (varchar(36))	Is associated with a product	1:1
order_line_item	quantity (int4)		

event	event_id (PK) (varchar(36))	Is associated with a store May have associated event_featured_products May be associated with a promotion	1:1 1:Zero or more 1:Zero or 1
event	store_id (FK) (varchar(36))	Is associated with a store	1:1
event	pomotion_id (varchar?(36))	May be associated with a promotion	1:Zero or 1
event	name (varchar(40))		
event	image_url (varchar?(2048))		
event	image_2_url (varchar?(2048))		
event	description (varchar(2048))		
event	start_timestamp (timestamp with timezone)		
event	end_timestamp (timestamp with timezone)		
event_featured_product	event_featured_product_id (PK) (varchar(36))	Is associated to an event Is associated to a product	1:1 1:1
event_featured_product	event_id (FK) (varchar(36))	Is associated with an event	1:1
event_featured_product	product_id (FK) (varchar(36))	Is associated with a product	1:1
product_category	product_category_id (PK) (int4)	Can be associated with a product_category Can have many product_categories Can have many product_category_maps	1:Zero or 1 1:Zero or Many 1:Zero or Many
product_category	parent_id (FK) (int4?)	Can be associated with a product_category	1:Zero or 1
product_category	name (varchar(40))		

product_category	description (varchar(256))		
author	author_id (PK) (varchar(36))	Is associated with at least one product	1:1 or More
author	author_image_url (varchar?(2048))		
author	first_name (varchar(40))		
author	last_name (varchar?(50))		
author	biography (varchar(2048))		
deliverable_regions	deliverable_region_id (PK) (int4)		
deliverable_regions	country_code (varchar(3))		
deliverable_regions	state_code (varchar(3))		
product_category_map	product_category_map_id (PK) (int4)	Is associated with a product_category Is associated with a product	1:1 1:1
product_category_map	product_id (FK) (varchar(36))	Is associated with a product	1:1
product_category_map	product_category_id (FK) (int4)	Is associated with a product_category	1:1
promotion	promotion_id (PK) (varchar(36))	Can be associated with multiple events Can be associated with multiple orders Can be associated with multiple product_promotions	1:Zero or More 1:Zero or More 1:Zero or More
promotion	name (varchar(40))		
promotion	description (varchar(80))		

promotion	discount_percentage (varchar?(36))		
promotion	start_timestamp (timestamp with timezone)		
promotion	end_timestamp (timestamp with timezone)		
product_promotion	product_promotion_id (PK) (varchar(36))	Is associated with a single promotion Is associated with a single product	1:1 1:1
product_promotion	promotion_id (FK) (varchar(40))	Is associated with a single promotion	1:1
product_promotion	product_id (FK) (varchar(80))	Is associated with a single product	1:1

Analysis of how the project fulfills the mission/goals of the target organization.

This project fulfills the missions and goals of Heartland Escapes in multiple ways. We now have a way to track inventory individually for each store. The inventory table is also built in a way that will support automated restocking of products for each store. This system is also more flexible and useable in that it will open the door to supporting an e-commerce website. Some additional tables will need to be added in the future to store payment information, or a third party service like google pay will support that behavior depending business decisions. Also, the flexible product category system can be extended in the future to support subscriptions when that functionality is desired.

This is just a preliminary entity relationship model, to support all the features desired by Heartland Escapes we will need to perform much more careful planning and consultation with the client. This, however, will give us the bedrock to further expand the data model as correspondence with Heartland Escapes continues.

Structured Query Language Scripts (Week 3)

DDL – Database Create, Table Create, and Table Alter FK Constraint Scripts

DDL.sql

DML – Data Insert, Update, and Delete Scripts

DML.sql

Reporting – CrossTab, Aggregation, and Querying scripts

Report SQL.sql

Analysis of how the project fulfills the mission/goals of the target organization.

This week was centered around creating a database around the entity relationship diagram and data dictionary provided in the previous section. This was a massive amount of work to complete, but it was a great exercise in finalizing some things about the data dictionary. For example, I found that the seed data I created for the names in the product table were much longer than the max length, so I was able to find out early that that length needed to be more than 40 characters. It also gave me a better idea about how the table relationships would work, and the level of complexity within the product categories table.

I found that it was incredibly difficult to aggregate the total number of products for parent columns in the product category table. I gave up on it for now, but that may require a change in direction for how this part of the system was implemented, which is great to know early on.

These scripts are the first steps in fulfilling the missions and goals of Heartland Escapes. The table creation scripts with constraints gives us a tangible platform to begin visualizing and modelling how the new Heartland Escapes system might work. These constraints enforce the business rules discussed in the previous section, and the tables fulfil our first section of business logic and functionality. Additionally, the seed data inserted allows us to begin working with how to process, manipulate, and query the data in the new Heartland Escapes system. It also allows us to test parts of the request functionality early, and determine what pain points exist in this plan and what refactoring may need to be conducted.

Database Administration Plan (Week 4)

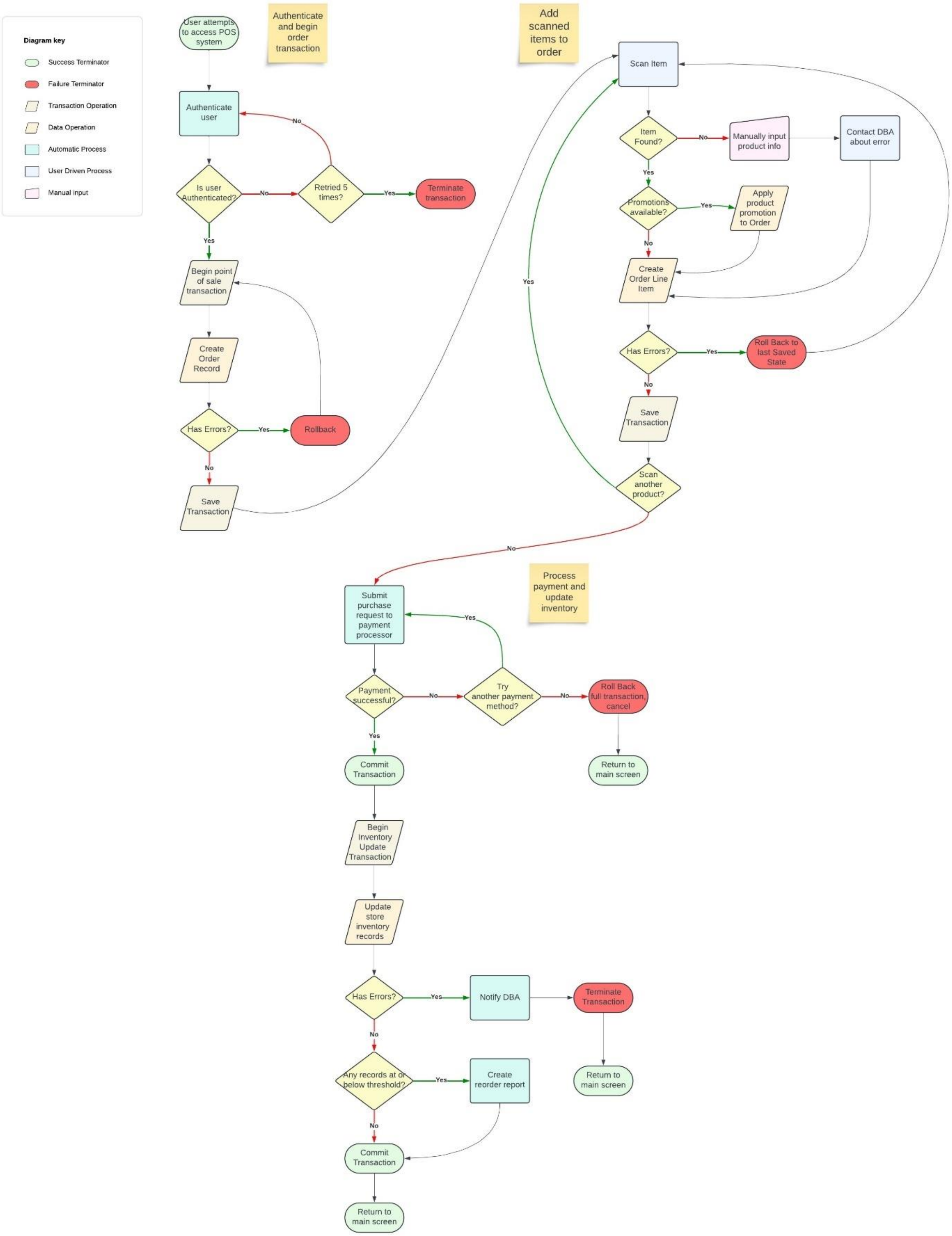
Data security, transaction management, backup plans and recovery procedures are all integral parts of a database administration plan. In order to ensure that our database system is resilient to failure and customer data is only accessible via authenticated users, we'll need to define procedures around all of these concepts for the Heartland Escapes database. Our first step in this process is to outline assumptions about our system.

1. A high volume of the orders often occurs during the daytime.
2. One person will serve the role of database administrator.
3. The database administrator account will serve as database owner.
4. The transaction log must be backed up.
5. Point-in-time recovery is required.
6. There is an always-on availability group.
7. The ability to purchase products online will be addressed in a future database project.

Over the next few sections, we'll outline a transaction management plan, database security procedures, and a backup plan and recovery procedure. These procedures will support Heartland Escapes in the case of failure or emergency. For example, if Heartland Escapes were to receive a ransom-ware attack, a system backup would be incredibly helpful to get systems back online. Also, sufficient security guidelines and procedures would help protect Heartland Escapes from being a victim of such an attack. A transaction management plan would also ensure that operations within the Heartland Escapes system are atomic, guaranteeing that if an error occurred within a transaction that the full transaction would be rolled back.

Transaction Management Plan

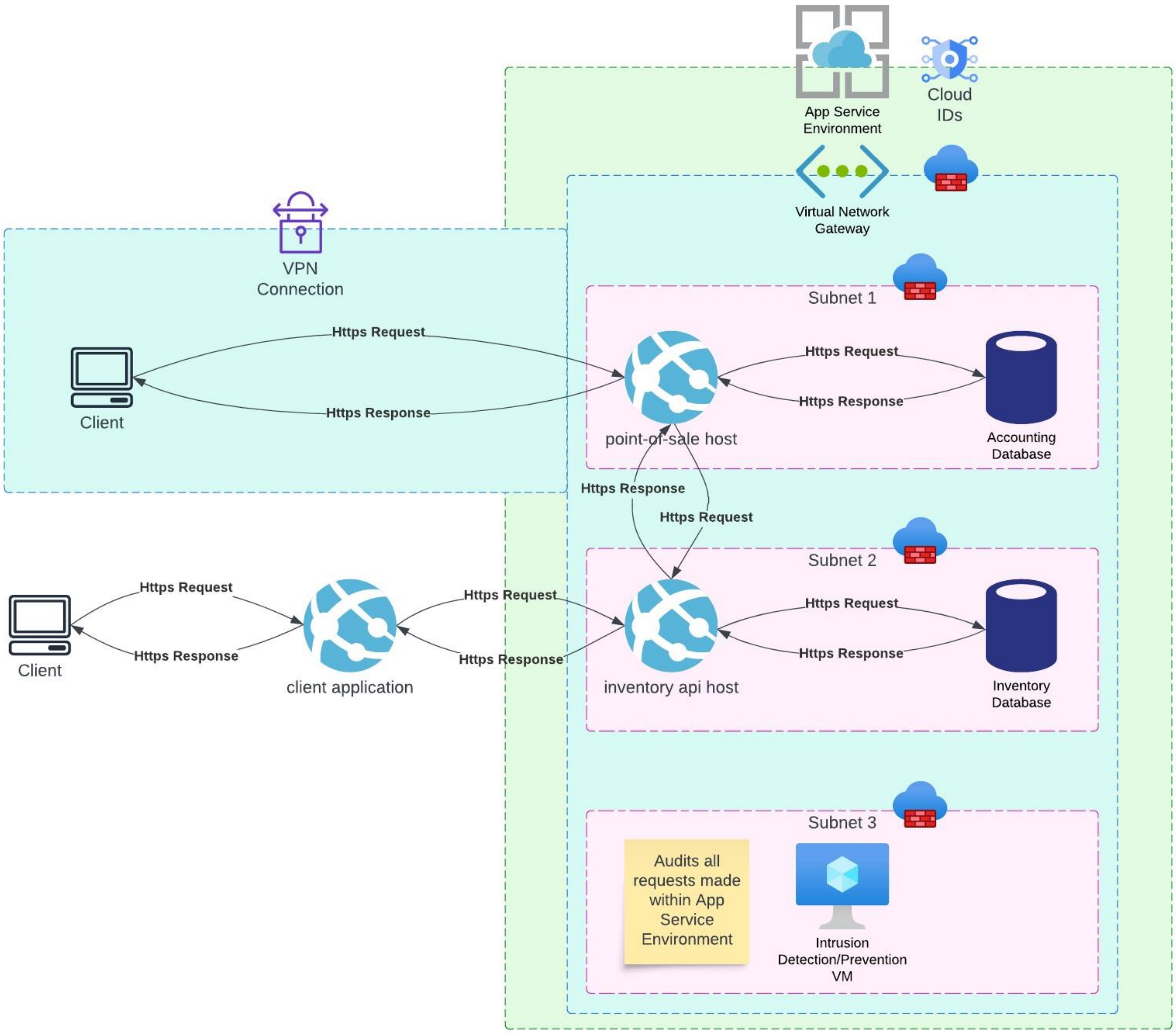
As stated previously, in any of our database transactions we'll need to make sure that if an error occurs in a process, the transaction is rolled back, and an error is sent to the requesting service. This is important because we don't want the state of any of our tables to be incomplete, and the user needs to know if the transaction failed. Below is an example flow chart of an order being placed in the Heartland Escapes database system. This flow chart captures how transactions interact with different necessary processes in order to fulfill an order placement. It also captures a secondary transaction to update the store inventory records applicable to the order.



Here is a link to the image for readability: Order_Transaction_Diagram.png

Database Security procedure

Most of the security concerns around the database have been handled in another document which will be linked with this assignment, but we will reiterate some of that information here. Heartland Escapes is going through a migration process to be hosted in Azure Cloud. As a result, much of our data concerns will be handled within the Azure Cloud Portal. Here is our network security diagram, which shows the accessibility of the Heartland Escapes inventory database:



As you can see, users will not be directly interacting with this database, but rather the API stood up between the client applications and the database. Also, the database is only accessible from within the virtual network, and only accepts secure http protocol requests. In the production environment, the only individual with direct read access to the database will be the database administrator. In development and staging environments, the DBA and development teams will all have read/write access. The Inventory API will have access to the heartland escapes inventory database via Azure Managed Identities. This access will be granted using Azure Active Directory, which utilizes role-based access controls. We'll create roles and access permissions for those roles to ensure that only users that reside within those roles have access to the database in each environment.

The DBA will be responsible for configuring the Heartland Escapes inventory database within Azure Cloud Portal and will also be responsible for configuring the Azure Managed Identities of the App Services. Password guidelines and rotation schedules will be the responsibility of the Heartland Escapes information security team, not the responsibility of the DBA. However, assigning individuals database access roles will be up to the discretion of the DBA.

Backup Plan and Recovery Procedure

The database administrator is responsible for developing a backup plan and recovery procedure. Instead of removing the original on premises Heartland Escapes server, we'll utilize it to receive and store backups of the Heartland Escapes system. The DBA will determine the frequency in which we perform backups and will create a plan to automate this process. This will give us physical redundancy of our data so that we aren't relying completely on the Azure server farm. This will also reduce our risk in the event of a natural or technological disaster.

The database administrator will be responsible for making decisions on what data is required to be backed up and will need to coordinate with the Heartland Escapes legal department for data retention policies. The DBA will also need to draft a plan on data redundancy. Heartland Escapes may benefit from having a data warehouse which could serve a couple of purposes. We could maintain data redundancy between the active database and the long-term data storing warehouse, and we could use the data warehouse as our data source for reporting. This would involve concocting a plan for data replication from Microsoft SQL Server, and a replication schedule for table records (batch or individual).

Our initial data recovery plan would be to restore data from backups if data recovery is necessary. After the data warehouse is built, this plan may be more nuanced. Additionally, to address the need of point-in-time

Analysis of how the project fulfills the mission/goals of the target organization

The responsibilities of the database administrator are many, and the importance of these responsibilities is significant. With this role defined and filled by a capable individual, Heartland Escapes will be able to put together a resilient, secure, and effective database that could one day be the backbone of an e-commerce environment. With proper data security, we can support our customers effectively and sleep well at night knowing that we've done our due diligence to ensure that their data is safe from bad actors. Also, with a data backup, redundancy, and recovery plan we'll be able to take comfort in the fact that if something does happen to the Heartland Escapes system, we'll be able to act swiftly and accordingly. Finally, with well thought out transaction control we will be able to better guarantee our data integrity. All of these outcomes are direct asks from Heartland Escapes, therefore I believe that the information outlined in this section directly fulfils the goals of our target organization.

Future Database System Implementation Plan (Week 5)

Database Differences

BASIS	RDBMS	OODBMS
Long Form	Stands for Relational Database Management System.	Stands for Object Oriented Database Management System.
Way of storing data	Stores data in Entities, defined as tables hold specific information.	Stores data as Objects.
Data Complexity	Handles comparatively simpler data.	Handles larger and complex data than RDBMS.
Grouping	Entity type refers to the collection of entity that share a common definition.	Class describes a group of objects that have common relationships, behaviors, and also have similar properties.
Data Handling	RDBMS stores only data.	Stores data as well as methods to use it.
Main Objective	Data Independence from application program.	Data Encapsulation.
Key	A Primary key distinctively identifies an object in a table.	An object identifier (OID) is an unambiguous, long-term name for any type of object or entity.
Data Retrieval	SQL (Structured Query Language)	Object Query Language (OQL)
Scalability	RDBMS has Limited scalability due to rigid schema	OODBMS has Highly scalable due to flexible schema
Concurrency Control	RDBMS has Fine-grained locking	OODBMS has Optimistic concurrency control
Data Relationships	In RDBMS Relational data is stored in tables and linked via foreign keys	In OODBMS faster for complex object-oriented queries
Performance	RDBMS is Efficient for complex queries involving multiple tables	OODBMS is Faster for complex object-oriented queries
Flexibility	RDBMS has Limited flexibility due to fixed schema	OODBMS has highly flexible due to object-oriented nature
Data Persistence	In RDBMS Data is stored in tables on disk	In OODBMS Data is stored in objects in memory or on disk
Examples	MySQL, Oracle, SQL Server	db4o, Versant, Objectivity/DB

(GeeksForGeeks, 2023)

GeeksForGeeks has done a great job at comparing and contrasting Relational Database Management Systems (RDBMS) and Object-Oriented Database Management Systems (OODBMS), which provided the data in the table above. OODBMS have a much more flexible nature, and offer functional concepts like inheritance, polymorphism, and encapsulation. These concepts should be familiar to those who work with object-oriented software development practices. This functionality allows for more flexibility in the development of systems.

Inheritance allows for multiple object properties to be defined by a single object. A good example of this would be the audit fields in the new Heartland Escapes database. Each table has the same auditing fields, which becomes a pain when creating these tables and inserting records. With an object-oriented database management system, we could define these fields on each entity by implementing an inheriting “auditable entity” base object for each table. This would remove the need to define those fields for every entity and allow for the simple addition of new audit fields across the system.

Polymorphism would allow for additional flexibility when implementing the e-commerce functionality requested by Heartland Escapes. As an example, we could have multiple types of orders that share the same field types (order ID, customer ID, store ID, promotional code, etc.) and attach a “GetAddress” function. This definition of the type would be called an “OrderInterface”. Then we could have two different types of order objects, “InStoreOrder” and “OnlineOrder” that would utilize their own implementation of the “GetAddress” function by inheriting the “OrderInterface”. This is a good example of how the Object-Oriented concepts of polymorphism and encapsulation could support the Heartland Escapes system. It may be a worthwhile venture to explore the idea of using OODBMS as we work to upgrade their data architecture.

A Web Database System is a catch-all term for web applications and services that store and retrieve data from a database. From my understanding, both a relational database and an object-oriented database can be used within a web database system. The current architecture defined above is a web database system that utilizes a relational database. (“Web-Based Database Management System”)

Changes needed to build a data warehouse/data mart.

As stated in the week 4 section, Heartland Escapes may benefit from utilizing a data warehouse as a data source for reporting, long term data storage, and as an additional data backup. This would not only be a consideration for the inventory database that we’re focusing on in this project, but also the accounting database that the point-of-sale system utilizes. As data from the accounting database would be stored in the warehouse, we’ll need to make considerations for PII and ensure that data is stored in a safe manner. Additionally, we would need to decide for what data is useful to keep in the warehouse, and any transformations that need to take place between our operating data system and the storage system.

Another consideration that would be required is how the data will be replicated to the warehouse. Would this be a nightly batch process, or would we rely on a data replication tool to send data to the warehouse as it’s written in the SQL Server DB using triggers? There are a lot of considerations to be made for this endeavor that are outside of the scope of this project. It’s important to begin thinking about data warehousing now so we can build flexibly to make the warehousing endeavor as painless as possible.

Distributed Database Considerations

As it stands, the Heartland Escapes database is not distributed. The current architecture of the database is built to support multiple stores within a single data environment. This has pros and cons:

Pros:

1. Inventory is known across stores, so if a customer is looking for an item that is not in stock in one store, they can easily retrieve inventory availability status of other stores.
2. Product pricing and descriptions are consistent across stores.
3. Events are easily tracked across stores.
4. Concurrency issues are easier to manage.
5. Security is easier to manage.

Cons:

1. Technologically, there is a single point of failure. If the database goes down, all stores are offline.
2. Network load on the database is not distributed, which means that a powerful database is needed to support all network transactions between the four stores.
3. Another point on the concept of a “single point of failure”, if the security is compromised for the database, this is true for all stores.

As a proposal to implement a distributed database system, I’d tentatively recommend data replication between two databases with geographic redundancy. We could load-balance requests to the database between stores and rely on data replication to maintain the state between the two database systems. This would allow for many of the benefits of having a central database and mitigate some of the pitfalls. With this method comes some considerations as well:

1. Concurrency becomes more complex with data replication.
2. Increased overhead to replicate data between the two systems.
3. Has network security implications.

Business Intelligence

Data is the primary driver of creating business intelligence. Through the data collected over time, companies take advantage of this information to create analytics that will support their business in marketing, product placement, and promotion timing. This is one of the most important reasons to develop and long-term data storage plan like a data warehouse. It's also a reason why companies pay Meta and other social media platforms for customer demographic information.

Putting data to the right use with demographic and geographic data can be a pivotal factor in increasing revenue for a store. With the accumulated data, companies can create reports on product relationships to demographics, time of year, and geography. These reports fuel future promotions and targeted advertising. Additionally, companies with the proper data analysis infrastructure can determine which products are performing well, and which products would be better left off the re-order sheet.

Data Warehouse Benefits

Sprinkled throughout this project has been hints toward the benefits of data warehousing. Business intelligence is a primary reason to have a data warehouse, as well as data redundancy, separation of operation data vs analytic data, and point-in-time data recovery.

Return on Investment

Using the data warehouse, we can create reports that will support year-over-year increases in the return on investments. By retrieving geographical and demographic data on product purchases, we can better target advertisements to individuals who are likely to be interested in said product. This is just one way we can use data warehousing to increase ROI. It's difficult to predict what the return on investment will be for Heartland Escapes, due to the addition of two stores. Keeping only the Lincoln stores in mind (since they are well established) we'll hope to be able to utilize the demographic data collected from attendees of events and the featured products to better target advertising for future events. Within the first three years, we'd like to see average event attendance double.

Competitive Advantage

By maintaining analytical data and reporting on Heartland Escapes revenue, and using informed advertising campaigns, we can maintain competitiveness with other bookstores. Depending on how robust and detailed our data collection is, we can pursue making Heartland Escapes the go-to books store for the Lincoln, Omaha, and Beatrice areas of Nebraska. Meticulous maintenance and data capturing is needed to compete with larger retail stores like Barnes and Noble.

Increased Productivity

Having this reporting capability can keep stake holders and other decision-makers informed on what direction to take the company in advertising campaigns, event planning, and store specific featured products that will bring foot traffic to the store. Keeping these individuals informed is crucial to progress the Heartland Escapes business at a fast pace.

Data Warehouse Problems

As with any technology, there needs to be a maintenance and expansion plan for on-going development. Even after Heartland Escapes stakeholders are "satisfied" with their system, they may find a need to again refactor their data system or add new features. The data warehousing implication handling for any new features added to the Heartland Escapes system will primarily be the responsibility of the database administrator, but it's always a good idea to cover processes for common scenarios.

New Features and System Maintenance

A plan needs to be in place to expand the captured fields in the data warehouse. If a new feature appears that has analytic implications, the database administrator needs to be able to a implement the ability to report on this field in the data warehouse. An example of this would be if the Heartland Escapes system wanted to begin capturing the marital status of customers for additional analytics. This would allow them to target books on marriage to married customers. A process needs to be in place for the DBA to implement this change to the data warehouse after implementation of the feature on the application and operating data system is completed.

It's also expected that Heartland Escapes has a mechanism for prioritizing and testing this work. If it is decided that our contracting company is no longer needed to support the new system, Heartland Escapes would do well to utilize the Agile development method for creating, organizing, and prioritizing work and creating acceptance criteria. They would also benefit from maintaining development environments, CI/CD pipelines, and potentially Liquibase for the DBA and potential future contract arrangements.

Increased Resource Demand

It is important to consistently monitor the Azure environment to make sure that there aren’t bottlenecks in processing, and that disk space isn’t approaching maximum capacity. The process to add additional disk space through Azure is simple, but there are cost considerations to be made. It is our recommendation that Heartland Escapes creates a process around adding additional hardware resources so that when the time comes to do so, the database administrator can easily make the necessary changes. It would make sense to have the DBA request the changes from the CFO and CTO and report the increase in monthly cost once the change is implemented. Azure also allows for advanced Azure component cost analysis.

Bug Fixes

Following a similar pattern to creating new features, bug remediation needs to have a defined process as well. A process needs to be defined for communication between the DBA and stakeholders for prioritization of bugs, and risk assessment. This again requires a development environment that is like the production environment, so that the fix can be well tested before it rolls out to the production environment.

Analysis of how the project fulfills the mission/goals of the target organization

This section I believe fulfils Heartland Escapes goals by giving them the outline of what will be needed of ongoing maintenance of their system post implementation. It also defines the opportunities that come with having an operational data warehouse. Additionally, hints toward required processes and documentation have been identified to give them a start toward regular maintenance and agile development. It’s the expectation that if we win this contract with Heartland Escapes, we will be able to provide more details for creating and documenting these procedures. With solid documentation, they can rest easy knowing that they have a plan if something goes awry or if they need to expand the features of their current system.

References

"Web-Based Database Management System." Www.w3schools.in, www.w3schools.in/dbms/web-based-database-management-system/. Accessed 21 Apr. 2024.

"Difference between RDBMS and OODBMS." GeeksforGeeks, 11 Mar. 2019, www.geeksforgeeks.org/difference-between-rdbms-and-oodbms/.