Term Project

Final Submission

|  |
| --- |
| Students Names |
| Mutungi, Vincent M. |
| Rubby, Awuonda D. |
| Wainaina, Teresia W. |
| Wawira, Claris C. |

Submitted on:

04/25/2025

# Abstract

Canyon Ranch, a premier luxury health resort and spa, faces challenges in managing guest data due to a fragmented IT infrastructure centered around the legacy Computerized Lodging Systems (CLS) and contrasting tools like Guestware. This leads to data redundancy, compromised integrity, and inefficiencies in delivering personalized services, ultimately impacting guest experience. To address these issues, this project proposes a centralized database system for Canyon Ranch, designed to consolidate guest information, including demographics, health records, preferences, appointments, reservations, and payments, as detailed in the transactional database schema (milestone schema). A complementary data warehouse (CanyonRanchDW) employing a star schema with fact tables (Fact.Appointments, Fact.Payments) and dimension tables (Dim.Guest, Dim.Service, Dim.Date, Dim.Location, Dim.RoomType, Dim.Feedback) enables advanced analytics for service optimization, guest satisfaction, and revenue trends. The solution enhances personalization by integrating guest preferences and health data, improves operational efficiency by streamlining cross-departmental processes, and supports data-driven decision-making through actionable insights. Expected benefits include a 20-30% increase in guest retention, reduced manual data handling by approximately 10 hours weekly, and scalable infrastructure for future enhancements like AI-driven recommendations. However, risks such as data security breaches, high development costs, and employee adoption resistance are acknowledged, with mitigations including encryption, phased implementation, and comprehensive training. Ethical and privacy concerns, particularly regarding sensitive health data (HealthInfo, MedicalHistory) and analytics in CanyonRanchDW, are addressed through robust consent processes, data minimization, and compliance with regulations. The Entity-Relationship Diagram (ERD) and database diagrams ensure data integrity via primary and foreign key constraints, while functional dependencies and backup strategies (CanyonRanch.bak, CanyonRanchDW.bak) support reliability. This centralized database and data warehouse solution positions Canyon Ranch to deliver superior guest experiences, optimize operations, and achieve sustainable growth while prioritizing data security and ethical considerations.

Table of Contents

[Abstract i](#_Toc196481202)

[List of Figures iv](#_Toc196481203)

[List of Tables iv](#_Toc196481204)

[Description of the Business Problem, Expected Benefits, and Projected Risks 1](#_Toc196481205)

[1. Business Problem 1](#_Toc196481206)

[2. Expected Benefits 1](#_Toc196481207)

[3. Projected Risks 2](#_Toc196481208)

[Entity-Relationship (ER) Diagram with Written Requirements 4](#_Toc196481209)

[1. Entities and Attributes 5](#_Toc196481210)

[2. Relationships 5](#_Toc196481211)

[3. Data Integrity Constraints 6](#_Toc196481212)

[4. Assumptions 6](#_Toc196481213)

[Conclusion 6](#_Toc196481214)

[Entity Relationship Diagram 7](#_Toc196481215)

[Database Diagram 8](#_Toc196481216)

[Entities, Attributes and Keys 9](#_Toc196481217)

[Functional Dependencies 10](#_Toc196481218)

[Database Backup 11](#_Toc196481219)

[Data Dictionary 12](#_Toc196481220)

[1. CanyonRanch Transactional Databases 12](#_Toc196481221)

[Data Warehouse Diagram 19](#_Toc196481222)

[Data Warehouse Data Dictionary 20](#_Toc196481223)

[Data Warehouse Backup File 24](#_Toc196481224)

[Ethical and Privacy Issues 24](#_Toc196481225)

[1. Privacy Risks from Sensitive Health Data 25](#_Toc196481226)

[2. Consent and Data Usage Transparency 25](#_Toc196481227)

[3. Data Sharing and Third-Party Risks 26](#_Toc196481228)

[4. Data Retention and Security in the Data Warehouse 27](#_Toc196481229)

[5. Bias and Fairness in Analytics 27](#_Toc196481230)

# List of Figures

[Figure 1: Entity Relationship Diagram 4](#_Toc196481233)

[Figure 2: Entity Relationship Schema 7](#_Toc196481234)

[Figure 3: Database Diagram MS-SQL 8](#_Toc196481235)

[Figure 4: Data Warehouse Diagram 19](#_Toc196481236)

# List of Tables

[Table 1: milestone.Appointment Data Dictionary 12](#_Toc196481239)

[Table 2: milestone.Feedback Data Dictionary 12](#_Toc196481240)

[Table 3: milestone.Guest Data Dictionary 13](#_Toc196481241)

[Table 4: milestone.Guest\_PhoneNo Data Dictionary 13](#_Toc196481242)

[Table 5: milestone.HealthInfo Data Dictionary 13](#_Toc196481243)

[Table 6: milestone.HotelPackage Data Dictionary 14](#_Toc196481244)

[Table 7: milestone.Location Data Dictionary 14](#_Toc196481245)

[Table 8: milestone.MedicalHistory Data Dictionary 15](#_Toc196481246)

[Table 9: milestone.Payments Data Dictionary 15](#_Toc196481247)

[Table 10: milestone.Preference Data Dictionary 16](#_Toc196481248)

[Table 11: milestone.Reservation Data Dictionary 16](#_Toc196481249)

[Table 12: milestone.Room Data Dictionary 16](#_Toc196481250)

[Table 13: milestone.RoomAssignment Data Dictionary 17](#_Toc196481251)

[Table 14: milestone.RoomType Data Dictionary 17](#_Toc196481252)

[Table 15: milestone.Service Data Dictionary 18](#_Toc196481253)

[Table 16: milestone.Staff Data Dictionary 18](#_Toc196481254)

[Table 17: Fact.Appointments Data Dictionary 20](#_Toc196481255)

[Table 18: Fact.Payments Data Dictionary 21](#_Toc196481256)

[Table 19: Dim.Guest Data Dictionary 21](#_Toc196481257)

[Table 20: Dim.Service Data Dictionary 22](#_Toc196481258)

[Table 21: Dim.Date Data Dictionary 22](#_Toc196481259)

[Table 22: Dim.Location Data Dictionary 23](#_Toc196481260)

[Table 23: Dim.RoomType Data Dictionary 23](#_Toc196481261)

[Table 24: Dim.Feedback Data Dictionary 24](#_Toc196481262)

# Description of the Business Problem, Expected Benefits, and Projected Risks

## Business Problem

Canyon Ranch, a luxury health resort and spa, faces significant challenges in managing customer data and delivering personalized services due to its fragmented IT infrastructure. The current system, centered around the legacy Computerized Lodging Systems (CLS), lacks integration with other tools like Guestware, making it difficult to centralize and analyze critical guest information such as preferences, service history, and health data. This leads to inefficiencies, including data redundancy, lack of data integrity, and difficulty retrieving essential business insights, ultimately resulting in a suboptimal guest experience.

## Expected Benefits

Implementing a centralized database will address these inefficiencies and provide the following advantages:

* **Enhanced Personalization**: By consolidating guest preferences, medical history, and service usage, Canyon Ranch can deliver tailored wellness programs and personalized service recommendations, improving the guest experience.
* **Operational Efficiency**: Streamlining data across departments (e.g., spa, nutrition, fitness, hospitality) will reduce redundant processes, improve coordination, and save staff time (e.g., reducing manual data lookups by approximately 10 hours per week).
* **Better Decision-Making**: The database will provide actionable insights into guest behavior, enabling management to refine service offerings, optimize marketing strategies, and anticipate customer needs.
* **Increased Customer Loyalty**: A high-touch, customized experience will strengthen guest retention, potentially increasing return visits by 20-30%.
* **Revenue Growth**: Targeted promotions and personalized service recommendations (e.g., suggesting complementary services like a nutrition consultation after a spa treatment) will encourage repeat visits and boost ancillary revenue.
* **Scalability**: The database can be expanded to accommodate future growth, such as integrating a loyalty program or AI-driven recommendations.
* **Security**: Controlled access and permissions will protect sensitive guest data, ensuring compliance with regulation.

## Projected Risks

While the database offers significant benefits, several risks must be managed:

* **Data Security Concerns**: Storing sensitive guest data (e.g., health questionnaires, medical alerts) poses risks of breaches and non-compliance with data protection regulations.

**Mitigation**: Implement role-based access control, data encryption, and regular security audits.

* **Implementation Challenges**: Integrating the new database with legacy systems like CLS and Guestware may cause downtime and transitional difficulties.

**Mitigation**: Use a phased implementation approach, starting with a Minimum Viable Product (MVP), and conduct thorough testing before full deployment.

* **Employee Adoption Resistance**: Staff accustomed to existing processes may resist the new system, requiring extensive training.

**Mitigation**: Provide user-friendly training materials, workshops, and ongoing support to ease the transition.

* **High Development Costs**: Building and maintaining a sophisticated database involves substantial financial investment.

**Mitigation**: Prioritize essential features and plan for scalable development to manage costs.

* **Data Overload**: Without proper organization, excessive data collection can overwhelm the system and reduce efficiency.

**Mitigation**: Implement data governance policies and optimize database indexing and partitioning.

* **Data Migration Challenges**: Transitioning from legacy systems may lead to inconsistencies, duplicate records, or data loss.

**Mitigation**: Conduct a comprehensive data audit, implement validation rules, and run test migrations before final transfer.

* **System Downtime**: Initial deployment or updates may disrupt operations.

**Mitigation**: Schedule deployments during non-peak hours and maintain backup systems.

# Entity-Relationship (ER) Diagram with Written Requirements

A diagram of a company

AI-generated content may be incorrect.

Figure 1: Entity Relationship Diagram

## Entities, Attributes and Keys

The ERD includes the following entities with their attributes (as shown in the diagrams):

**Guest:**

GuestID (Primary Key), LastName, FirstName, Email, StreetAddress, City, State, DoB,

**HealthInfo:**

HealthInfoID (Primary Key), QuestionnaireResponses, MedicalAlerts, GuestID (Foreign Key)

**Medical\_History:**

HistoryID (Primary Key), RecordDate, HealthStatus, HealthDetails, GuestID` (Foreign Key)

**Preference:**

PreferenceID (Primary Key), PreferenceType, PDescription, GuestID (Foreign Key)

**Feedback:**

FeedbackID (Primary Key), Rating, Comments, SubmissionDate, GuestID (Foreign Key), AppointmentID (Foreign Key, Optional)

**Reservation:**

ReservationID (Primary Key), CheckInDate, CheckOutDate, Status, RoomTypeID (Foreign Key), GuestID (Foreign Key), LocationID (Foreign Key)

**RoomAssignment:**

AssignmentID (Primary Key), ReservationID (Foreign Key), RoomID (Foreign Key), StartDate, EndDate

**Room:**

RoomID (Primary Key), RoomNumber, AvailabilityStatus`, RoomTypeID (Foreign Key)

**RoomType:**

RoomTypeID (Primary Key), RoomType, Description, PricePerNight

**HotelPackage:**

PackageID (Primary Key), PackageName, Description, Price, DurationDays, AllowanceHealthHealing, AllowanceSpaSports

**Service:**

ServiceID (Primary Key), ServiceName, Category, Price, Department, IsComplimentary, PackageID (Foreign Key, Optional)

**Appointment:**

AppointmentID (Primary Key), DateTime, Status, GuestID (Foreign Key), ServiceID (Foreign Key), StaffID (Foreign Key)

**Staff:**

StaffID (Primary Key), FirstName, LastName, Role, Designation, Qualifications

**Payment:**

PaymentID (Primary Key), ReservationID (Foreign Key, Optional), AppointmentID (Foreign Key, Optional), Amount, PaymentDate, PaymentMethod, PaymentFor

**Location:**

LocationID (Primary Key), StreetAddress, City, State

## Relationships

1. Guest – Reservation; A Guest reserves Reservations (1:M)
2. Guest - Past\_Visits; A Guest has Past\_Visits (1:M)
3. Guest – Preference; A Guest prefers Preferences (1:M)
4. Guest – HealthInfo; A Guest has HealthInfo (1:1)
5. Guest – MedicalHistory; A Guest has several MedicalHistorys (1:M)
6. Guest – Feedback; A Guest provides Feedback (1:M)
7. Guest – Appointment; A Guest books Appointments (1:M)
8. Reservation – Location; A Reservation takes place at a Location (M:1)
9. Reservation – Payment; A Reservation is associated with Payments (1:1)
10. RoomType – HotelPackage; A RoomType accommodates HotelPackages (1:M)
11. HotelPackage – Service; A HotelPackage contains Services (1:M)
12. Service – Appointment; A Service is included in an Appointment (1:M)
13. Service – Payment; A Guest makes Payments (1:1)
14. Appointment – Staff; An Appointment is facilitated by Staff (M:1)
15. Reservation – RoomAssignment; Reservation assigned to a RoomAssignment (1:M)
16. RoomAssignment - RoomType; RoomReservation assigned to a RoomType (1:M)
17. Feedback – Service; A Feedback is associated with a Service (M:N)

## Data Integrity Constraints

* **Primary Keys (PK):** Ensure each entity's primary key is unique.
* **Foreign Keys (FK):** Maintain referential integrity between related entities.
* **Data Types:** Ensure appropriate data types are defined for attributes (e.g., dates for reservation dates, numeric types for prices and quantities).
* **Null Constraints:** Enforce non-null constraints for mandatory fields such as GuestName, ReservationDate, and PaymentAmount.

## Assumptions

* The ERD assumes that packages can contain multiple services.
* Medical data is kept confidential within the HealthInfo entity.
* Room details are separated to ensure flexibility in future updates.

# Conclusion

This report outlines a robust database design to address Canyon Ranch’s data management challenges. By implementing a centralized system, the organization will improve efficiency, enhance guest experience, and support future growth. The proposed structure ensures data consistency, integrity, and scalability while mitigating security and implementation risks.

# Entity Relationship Diagram

A diagram of a company

AI-generated content may be incorrect.

Figure 2: Entity Relationship Schema

* Added the Room table.

# Database Diagram

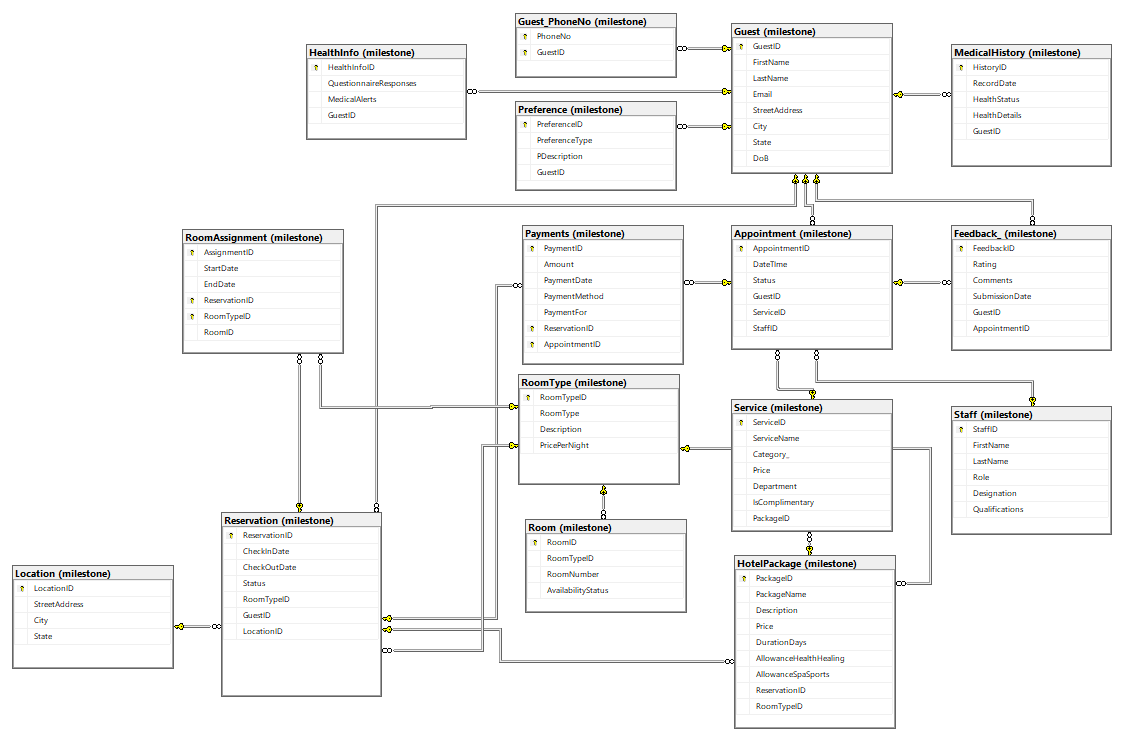


Figure 3: Database Diagram MS-SQL

# Functional Dependencies

**Guest**

* GuestID → LastName, FirstName, Email, StreetAddress, City, State, DoB

**HealthInfo**

* HealthInfoID → QuestionnaireResponses, MedicalAlerts, GuestI

**Medical\_History**

* HistoryID → RecordDate, HealthStatus, HealthDetails, GuestID

**Preference**

* PreferenceID → PreferenceType, PDescription, GuestID

**Feedback**

* FeedbackID → Rating, Comments, SubmissionDate, GuestID, AppointmentID

**Reservation**

* ReservationID → CheckInDate, CheckOutDate, Status, RoomTypeID, GuestID, LocationID

**RoomAssignment**

* AssignmentID → ReservationID, RoomID, StartDate, EndDate

**Room**

* RoomID → RoomNumber, AvailabilityStatus, RoomTypeID

**RoomType**

* RoomTypeID → RoomType, Description, PricePerNight

**HotelPackage**

* PackageID → PackageName, Description, Price, DurationDays, AllowanceHealthHealing, AllowanceSpaSports

**Service**

* ServiceID → ServiceName, Category, Price, Department, IsComplimentary, PackageID

**Appointment**

* AppointmentID → DateTime, Status, GuestID, ServiceID, StaffID

**Staff**

* StaffID → FirstName, LastName, Role, Designation, Qualifications

**Payment**

* PaymentID → Amount, PaymentDate, PaymentMethod, PaymentFor, ReservationID, AppointmentID

**Location**

* LocationID → StreetAddress, City, State

# Database Backup

See the attached OneDrive link: [CanyonRanch.bak](https://coyotesusd-my.sharepoint.com/:u:/g/personal/vincent_mutungi_coyotes_usd_edu/EY-oG-rdneBIkkwXgqrtZ7wBMzEVN5r7ObeODItl95T7hg?e=0nBvEL)

# Data Dictionary

## CanyonRanch Transactional Databases

Table 1: milestone.Appointment Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.Appointment | AppointmentID | INT | Primary key; unique identifier for each appointment. |
|  | DateTIme | DATE | Date of the appointment. |
|  | Status | VARCHAR(100) | Status of the appointment (e.g., Scheduled, Completed, Canceled). |
|  | GuestID | INT | Foreign key; references milestone.Guest.GuestID; identifies the guest. |
|  | ServiceID | INT | Foreign key; references milestone.Service.ServiceID; identifies the service. |
|  | StaffID | INT | Foreign key; references milestone.Staff.StaffID; identifies the staff member performing the service. |

**Description**: Records details of guest appointments for wellness or spa services, including scheduling and staff assignments.

Table 2: milestone.Feedback Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.Feedback\_ | FeedbackID | INT | Primary key; unique identifier for each feedback record. |
|  | Rating | INT | Guest’s rating of the service (e.g., 1–5). |
|  | Comments | VARCHAR(100) | Guest’s comments or feedback details. |
|  | SubmissionDate | DATE | Date the feedback was submitted. |
|  | GuestID | INT | Foreign key; references milestone.Guest.GuestID; identifies the guest providing feedback. |
|  | AppointmentID | INT | Foreign key; references milestone.Appointment.AppointmentID; links to the appointment (nullable). |

**Description**: Captures guest feedback on services or appointments, including ratings and comments.

Table 3: milestone.Guest Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.Guest | GuestID | INT | Primary key; unique identifier for each guest. |
|  | FirstName | VARCHAR(100) | Guest’s first name. |
|  | LastName | VARCHAR(100) | Guest’s last name. |
|  | Email | VARCHAR(150) | Guest’s email address for communication and confirmations. |
|  | StreetAddress | VARCHAR(100) | Guest’s street address. |
|  | City | VARCHAR(100) | Guest’s city of residence. |
|  | State | VARCHAR(100) | Guest’s state of residence. |
|  | DoB | DATE | Guest’s date of birth for age verification or personalization. |

**Description**: Stores personal and contact information for guests booking services or accommodations.

Table 4: milestone.Guest\_PhoneNo Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.Guest\_PhoneNo | PhoneNo | INT | Primary key; guest’s phone number. |
|  | GuestID | INT | Foreign key; references milestone.Guest.GuestID; identifies the guest. |

**Description**: Maintains phone numbers for guests, supporting multiple contact numbers per guest.

Table 5: milestone.HealthInfo Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.HealthInfo | HealthInfoID | INT | Primary key; unique identifier for each health info record. |
|  | QuestionnaireResponses | VARCHAR(150) | Guest’s responses to health-related questions. |
|  | MedicalAlerts | VARCHAR(100) | Medical conditions or alerts (e.g., allergies). |
|  | GuestID | INT | Foreign key; references milestone.Guest.GuestID; identifies the guest. |

**Description**: Records guest health information, including questionnaire responses and medical alerts, for personalized services.

Table 6: milestone.HotelPackage Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.HotelPackage | PackageID | INT | Primary key; unique identifier for each package. |
|  | PackageName | VARCHAR(100) | Name of the package (e.g., Wellness Retreat). |
|  | Description | VARCHAR(100) | Description of package inclusions. |
|  | Price | INT | Total price of the package. |
|  | DurationDays | INT | Duration of the package in days. |
|  | AllowanceHealthHealing | VARCHAR(100) | Allowances for health or healing services (e.g., therapy credits). |
|  | AllowanceSpaSports | VARCHAR(100) | Allowances for spa or sports activities (e.g., massage credits). |
|  | ReservationID | INT | Foreign key; references milestone.Reservation.ReservationID; links to a reservation (nullable). |
|  | RoomTypeID | INT | Foreign key; references milestone.RoomType.RoomTypeID; specifies room type (nullable). |

**Description**: Details wellness or accommodation packages offered, including pricing and inclusions.

Table 7: milestone.Location Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.Location | LocationID | INT | Primary key; unique identifier for each resort location. |
|  | StreetAddress | VARCHAR(150) | Street address of the location. |
|  | City | VARCHAR(100) | City of the location. |
|  | State | VARCHAR(100) | State of the location. |

**Description**: Stores details of resort locations or properties for reservations and services.

Table 8: milestone.MedicalHistory Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.MedicalHistory | HistoryID | INT | Primary key; unique identifier for each medical history record. |
|  | RecordDate | DATE | Date the medical history was recorded. |
|  | HealthStatus | VARCHAR(100) | Guest’s health status (e.g., Stable, Chronic). |
|  | HealthDetails | VARCHAR(150) | Detailed medical history information. |
|  | GuestID | INT | Foreign key; references milestone.Guest.GuestID; identifies the guest. |

**Description**: Tracks guests’ medical history to support health and wellness services.

Table 9: milestone.Payments Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.Payments | PaymentID | INT | Primary key; unique identifier for each payment. |
|  | Amount | INT | Payment amount in currency units. |
|  | PaymentDate | DATE | Date the payment was made. |
|  | PaymentMethod | VARCHAR(100) | Payment method (e.g., Credit Card, Cash). |
|  | PaymentFor | VARCHAR(100) | Purpose of the payment (e.g., Reservation, Service). |
|  | ReservationID | INT | Foreign key; references milestone.Reservation.ReservationID; links to a reservation. |
|  | AppointmentID | INT | Foreign key; references milestone.Appointment.AppointmentID; links to an appointment. |

**Description**: Records financial transactions for reservations, appointments, or packages.

Table 10: milestone.Preference Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.Preference | PreferenceID | INT | Primary key; unique identifier for each preference. |
|  | PreferenceType | VARCHAR(100) | Type of preference (e.g., Dietary, Room). |
|  | PDescription | VARCHAR(150) | Detailed description of the preference. |
|  | GuestID | INT | Foreign key; references milestone.Guest.GuestID; identifies the guest. |

**Description**: Captures guest preferences for personalized services or accommodations.

Table 11: milestone.Reservation Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.Reservation | ReservationID | INT | Primary key; unique identifier for each reservation. |
|  | CheckInDate | DATE | Date the guest checks in. |
|  | CheckOutDate | DATE | Date the guest checks out. |
|  | Status | VARCHAR(100) | Status of the reservation (e.g., Confirmed, Canceled). |
|  | RoomTypeID | INT | Foreign key; references milestone.RoomType.RoomTypeID; specifies room type. |
|  | GuestID | INT | Foreign key; references milestone.Guest.GuestID; identifies the guest. |
|  | LocationID | INT | Foreign key; references milestone.Location.LocationID; specifies resort location. |

**Description**: Manages guest reservations for accommodations, including check-in/out dates and room types.

Table 12: milestone.Room Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.Room | RoomID | INT | Primary key; unique identifier for each room. |
|  | RoomTypeID | INT | Foreign key; references milestone.RoomType.RoomTypeID; specifies room type. |
|  | RoomNumber | VARCHAR(10) | Room number or identifier. |
|  | AvailabilityStatus | VARCHAR(50) | Availability status (e.g., Available, Occupied). |

**Description**: Stores details of individual rooms, including type and availability.

Table 13: milestone.RoomAssignment Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.RoomAssignment | AssignmentID | INT | Primary key; unique identifier for each room assignment. |
|  | StartDate | DATE | Start date of the room assignment. |
|  | EndDate | DATE | End date of the room assignment. |
|  | ReservationID | INT | Foreign key; references milestone.Reservation.ReservationID; links to a reservation. |
|  | RoomTypeID | INT | Foreign key; references milestone.RoomType.RoomTypeID; specifies room type. |
|  | RoomID | INT | Foreign key; references milestone.Room.RoomID; specifies the assigned room (nullable). |

**Description**: Tracks assignments of rooms to reservations for specific dates.

Table 14: milestone.RoomType Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.RoomType | RoomTypeID | INT | Primary key; unique identifier for each room type. |
|  | RoomType | VARCHAR(100) | Name of the room type (e.g., Deluxe Suite). |
|  | Description | VARCHAR(100) | Description of the room type. |
|  | PricePerNight | INT | Price per night for the room type. |

**Description**: Defines room types available for reservations, including pricing and descriptions.

Table 15: milestone.Service Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.Service | ServiceID | INT | Primary key; unique identifier for each service. |
|  | ServiceName | VARCHAR(100) | Name of the service (e.g., Massage, Yoga). |
|  | Category\_ | VARCHAR(100) | Category of the service (e.g., Spa, Fitness). |
|  | Price | INT | Cost of the service. |
|  | Department | VARCHAR(100) | Department offering the service (e.g., Spa, Health). |
|  | IsComplimentary | VARCHAR(100) | Indicates if the service is complimentary (e.g., Yes, No). |
|  | PackageID | INT | Foreign key; references milestone.HotelPackage.PackageID; links to a package (nullable). |

**Description**: Maintains details of wellness or spa services, including pricing and categories.

Table 16: milestone.Staff Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| milestone.Staff | StaffID | INT | Primary key; unique identifier for each staff member. |
|  | FirstName | VARCHAR(100) | Staff member’s first name. |
|  | LastName | VARCHAR(100) | Staff member’s last name. |
|  | Role | VARCHAR(100) | Staff member’s role (e.g., Therapist, Receptionist). |
|  | Designation | VARCHAR(100) | Staff member’s designation (e.g., Senior Therapist). |
|  | Qualifications | VARCHAR(150) | Staff member’s qualifications (e.g., Licensed Massage Therapist). |

**Description**: Tracks staff members, their roles, and qualifications for service delivery.

# Data Warehouse Diagram

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4: Data Warehouse Diagram

# Data Warehouse Data Dictionary

The data warehouse uses a star schema with fact tables for measurable metrics and dimension tables for descriptive context, enabling analytics on appointments, payments, guest satisfaction, and operational trends.

**Fact.Appointments**

**Description**: Central fact table storing metrics for guest appointments, such as service count and revenue, linked to dimensions for analyzing service usage, guest behavior, and regional performance.

Table 17: Fact.Appointments Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| Fact.Appointments | AppointmentFactID | BIGINT | Primary key; unique identifier for each appointment fact record (auto-incremented). |
|  | GuestKey | INT | Foreign key; references Dim.Guest.GuestKey; identifies the guest. |
|  | ServiceKey | INT | Foreign key; references Dim.Service.ServiceKey; identifies the service. |
|  | DateKey | INT | Foreign key; references Dim.Date.DateKey; identifies the appointment date. |
|  | LocationKey | INT | Foreign key; references Dim.Location.LocationKey; identifies the resort location. |
|  | RoomTypeKey | INT | Foreign key; references Dim.RoomType.RoomTypeKey; specifies the room type linked to the guest’s reservation (nullable). |
|  | FeedbackKey | INT | Foreign key; references Dim.Feedback.FeedbackKey; links to feedback for the appointment (nullable). |
|  | AppointmentCount | INT | Number of appointments (default 1 per record). |
|  | ServiceRevenue | INT | Revenue generated from the service, sourced from milestone.Service.Price. |

**Fact.Payments**

**Description**: Fact table tracking payment metrics, such as amount and count, linked to dimensions for analyzing revenue trends by guest, date, and location.

Table 18: Fact.Payments Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| Fact.Payments | PaymentFactID | BIGINT | Primary key; unique identifier for each payment fact record (auto-incremented). |
|  | GuestKey | INT | Foreign key; references Dim.Guest.GuestKey; identifies the guest. |
|  | DateKey | INT | Foreign key; references Dim.Date.DateKey; identifies the payment date. |
|  | LocationKey | INT | Foreign key; references Dim.Location.LocationKey; identifies the resort location. |
|  | RoomTypeKey | INT | Foreign key; references Dim.RoomType.RoomTypeKey; specifies the room type linked to the reservation (nullable for appointment-based payments). |
|  | PaymentAmount | INT | Total payment amount, sourced from milestone.Payments.Amount. |
|  | PaymentCount | INT | Number of payments (default 1 per record). |

**Dim.Guest**

**Description**: Dimension table providing guest context for analytics, enabling segmentation by name, email, or loyalty status.

Table 19: Dim.Guest Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| Dim.Guest | GuestKey | INT | Primary key; surrogate key for the guest in the data warehouse (auto-incremented). |
|  | GuestID | INT | Original GuestID from milestone.Guest; links to transactional data. |
|  | GuestName | VARCHAR(200) | Concatenated FirstName and LastName from milestone.Guest. |
|  | Email | VARCHAR(150) | Guest’s email address, sourced from milestone.Guest.Email. |
|  | DoB | DATE | Guest’s date of birth, sourced from milestone.Guest.DoB. |
|  | LoyaltyStatus | VARCHAR(50) | Guest’s loyalty tier (e.g., Gold, Silver, Regular), derived during ETL (default ‘Regular’). |

**Dim.Service**

**Description**: Dimension table supporting service-based analytics, such as by category, department, or complimentary status.

Table 20: Dim.Service Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| Dim.Service | ServiceKey | INT | Primary key; surrogate key for the service in the data warehouse (auto-incremented). |
|  | ServiceID | INT | Original ServiceID from milestone.Service; links to transactional data. |
|  | ServiceName | VARCHAR(100) | Name of the service (e.g., Massage, Yoga), sourced from milestone.Service.ServiceName. |
|  | Category | VARCHAR(100) | Service category (e.g., Spa, Fitness), sourced from milestone.Service.Category\_. |
|  | Price | INT | Cost of the service, sourced from milestone.Service.Price. |
|  | Department | VARCHAR(100) | Department offering the service (e.g., Spa, Health), sourced from milestone.Service.Department. |
|  | IsComplimentary | VARCHAR(100) | Indicates if the service is complimentary (e.g., Yes, No), sourced from milestone.Service.IsComplimentary. |

**Dim.Date**

**Description**: Dimension table providing time-based context for analytics, enabling breakdowns by date, month, quarter, or year.

Table 21: Dim.Date Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| Dim.Date | DateKey | INT | Primary key; unique identifier for each date (e.g., 20230101 for Jan 1, 2023). |
|  | FullDate | DATE | The actual date, used for joins with transactional data. |
|  | DayOfMonth | INT | Day of the month (1–31). |
|  | Month | INT | Month number (1–12). |
|  | MonthName | VARCHAR(10) | Name of the month (e.g., January). |
|  | Quarter | INT | Quarter of the year (1–4). |
|  | Year | INT | Year of the date. |
|  | DayOfWeek | VARCHAR(10) | Day of the week (e.g., Monday). |

**Dim.Location**

**Description**: Dimension table facilitating location-based analytics, such as by resort property or region.

Table 22: Dim.Location Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| Dim.Location | LocationKey | INT | Primary key; surrogate key for the location in the data warehouse (auto-incremented). |
|  | LocationID | INT | Original LocationID from milestone.Location; links to transactional data. |
|  | StreetAddress | VARCHAR(150) | Street address of the location, sourced from milestone.Location.StreetAddress. |
|  | City | VARCHAR(100) | City of the location, sourced from milestone.Location.City. |
|  | State | VARCHAR(100) | State of the location, sourced from milestone.Location.State. |

**Dim.RoomType**

**Description**: Dimension table supporting reservation-based analytics, such as by room type or price range.

Table 23: Dim.RoomType Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| Dim.RoomType | RoomTypeKey | INT | Primary key; surrogate key for the room type in the data warehouse (auto-incremented). |
|  | RoomTypeID | INT | Original RoomTypeID from milestone.RoomType; links to transactional data. |
|  | RoomType | VARCHAR(100) | Name of the room type (e.g., Deluxe Suite), sourced from milestone.RoomType.RoomType. |
|  | PricePerNight | INT | Price per night, sourced from milestone.RoomType.PricePerNight. |

**Dim.Feedback**

**Description**: Dimension table enabling quality analytics based on guest feedback ratings and submission dates.

Table 24: Dim.Feedback Data Dictionary

| **Table Name** | **Column Name** | **Data Type** | **Description** |
| --- | --- | --- | --- |
| Dim.Feedback | FeedbackKey | INT | Primary key; surrogate key for the feedback in the data warehouse (auto-incremented). |
|  | FeedbackID | INT | Original FeedbackID from milestone.Feedback\_; links to transactional data. |
|  | Rating | INT | Guest’s rating (e.g., 1–5), sourced from milestone.Feedback\_.Rating. |
|  | SubmissionDate | DATE | Date the feedback was submitted, sourced from milestone.Feedback\_.SubmissionDate. |

# Data Warehouse Backup File

[CanyonRanchDW.bak](https://coyotesusd-my.sharepoint.com/:u:/g/personal/vincent_mutungi_coyotes_usd_edu/ESKI_LfNfKVOmRho_g3F3icB_PoZIAYu8TMDNow2KLFs0A?e=z3Lvot) -> Onedrive Link

# Ethical and Privacy Issues

The CanyonRanch database, designed for a wellness resort, collects sensitive personal and health related data across its transactional and data warehouse. This data includes guest demographics (Guest), health information (HealthInfo, MedicalHistory), preferences (Preference), feedback (Feedback\_), appointments (Appointment), reservations (Reservation), and payments (Payments). While this data enables personalized services and business analytics, its collection and use raise significant ethical and privacy concerns. Below, we discuss key issues, drawing on the database structures and the resort’s operational context.

## Privacy Risks from Sensitive Health Data

The HealthInfo and MedicalHistory tables store highly sensitive information, such as QuestionnaireResponses, MedicalAlerts, HealthStatus, and HealthDetails. Similarly, the Guest table includes personally identifiable information (PII) like FirstName, LastName, Email, DoB, and StreetAddress. Collecting and storing such data poses risks of:

* **Data Breaches**: Unauthorized access could expose medical conditions or PII, leading to identity theft or reputational harm. For example, a breach of MedicalHistory.HealthDetails could reveal a guest’s chronic illness, violating their privacy.
* **Ethical Concerns**: Guests may not fully understand how their health data is used or shared. Without transparent consent processes, CanyonRanch risks exploiting trust, especially if data is used for analytics in CanyonRanchDW (e.g., Dim.Guest for loyalty segmentation).

**Mitigation**: Implement encryption for sensitive columns (e.g., HealthDetails), enforce strict access controls (e.g., limiting HealthInfo access to medical staff), and ensure compliance with regulations like HIPAA or GDPR, which mandate explicit consent and data minimization.

## Consent and Data Usage Transparency

The Preference table (e.g., PreferenceType, PDescription) and Feedback\_ table (e.g., Rating, Comments) collect guest preferences and opinions to personalize services. However, ethical issues arise if guests are unaware of how this data is processed, especially in CanyonRanchDW for analytics (e.g., Fact.Appointments linking to Dim.Feedback for service quality analysis).

* **Lack of Informed Consent**: Guests may not know their feedback or preferences are stored indefinitely or used for aggregated analytics, potentially leading to profiling (e.g., targeting high-spending guests based on Fact.Payments.PaymentAmount).
* **Privacy Invasion**: Using Guest.DoB or Preference.PDescription to infer behaviors (e.g., dietary needs) without explicit permission could feel intrusive.

**Mitigation**: Provide clear privacy policies at booking, detailing data usage (e.g., analytics in Dim.Service for service trends). Offer opt-in/opt-out mechanisms for data collection beyond essential operations, and anonymize data in CanyonRanchDW (e.g., excluding GuestName from Dim.Guest).

## Data Sharing and Third-Party Risks

CanyonRanch may share data with third parties (e.g., spa service providers, marketing firms) or use it in CanyonRanchDW for external analytics. Tables like Guest.Email and Payments.PaymentMethod could be shared for targeted marketing or payment processing, raising concerns:

* **Unauthorized Sharing**: Sharing Guest.Email or HealthInfo.QuestionnaireResponses without consent violates privacy and could lead to spam or misuse.
* **Ethical Exploitation**: Using Fact.Appointments.ServiceRevenue or Dim.Feedback.Rating to prioritize profitable services over guest well-being (e.g., pushing high-cost spa treatments) prioritizes profit over ethics.

**Mitigation**: Establish data-sharing agreements with explicit guest consent, limit shared fields (e.g., exclude MedicalHistory), and use aggregated data in CanyonRanchDW (e.g., Fact.Payments without PII) for external analytics.

## Data Retention and Security in the Data Warehouse

The CanyonRanchDW data warehouse retains data indefinitely for historical analytics (e.g., Dim.Date spanning years, Fact.Appointments for trends). Long-term storage of PII (Dim.Guest.GuestName, Email) and sensitive metrics (Dim.Feedback.Rating) increases risks:

* **Retention Risks**: Storing Guest.DoB or Feedback.Comments (via FeedbackKey) beyond necessary periods could expose outdated data in a breach.
* **Security Gaps**: Weak access controls on CanyonRanchDW (e.g., analysts accessing Dim.Guest) could lead to internal misuse.

**Mitigation**: Implement data retention policies (e.g., delete Dim.Guest records after 5 years unless required), pseudonymize PII in CanyonRanchDW, and enforce role-based access (e.g., only managers view Fact.Payments).

## Bias and Fairness in Analytics

The CanyonRanchDW uses LoyaltyStatus in Dim.Guest (derived during ETL) and ServiceRevenue in Fact.Appointments for guest segmentation and service optimization. This raises ethical concerns:

* **Bias in Profiling**: Deriving LoyaltyStatus based on age or spending (e.g., PaymentAmount) could unfairly prioritize older or wealthier guests, marginalizing others.
* **Service Disparities**: Over-emphasizing high revenue services (e.g., via Dim.Service.Price) may reduce access to affordable wellness options, undermining inclusivity.

**Mitigation**: Ensure transparent LoyaltyStatus criteria, audit analytics for bias (e.g., equal service access across Dim.Guest segments), and balance revenue driven decisions with guest well-being.

**Conclusion**

The CanyonRanch databases enable valuable analytics but pose significant ethical and privacy risks due to sensitive health data, potential consent gaps, data sharing, retention practices, and analytical biases. By implementing robust encryption, clear consent processes, data minimization, and bias audits, CanyonRanch can mitigate these risks, ensuring ethical data use while maintaining guest trust and regulatory compliance.