Last Resort Hotel Database Report

Group A: William Camacho, Cindy Chen, Max Yan, Franklin Dong

Business Overview:

The Last Resort Hotel operates as a full-service business, offering not only guest accommodations and dining, but also several event services, allowing the appointment of an outside local host. Its primary business goal is to maximize the room occupation rate and total revenue while maintaining security and customer satisfaction. In recent years, it has grown rapidly with the acquisition of the Club Med Chain and several very high-end luxury hotels worldwide. From a management perspective, the database should support hotel operations and decision-making by providing insights into occupancy trends, revenue performance across different services, and the profitability of each customer.

As the first step in creating the ERD model, the team aims to include as many entities as mentioned in the project description, while maintaining normalization to avoid data redundancy as the primary objective.

To briefly introduce the main business operations, the hotel's operations involve several core processes, including room reservations, event organization, billing, and access tracking.

For room assignments, guests or hosts make reservations for rooms or events. The ERD model includes tables such as reservation, room, guest, and host to ensure room availability and status. Reservations can extend beyond rooms; they can also include specific facilities. Thus, an expected facility usage table is created to keep track of this.

For event organization, each event's organizer is either a host or an organization. An organization will have a representative, so, depending on the organizer, either the host or the representative will be billed for the event. This person is listed in "billedParty", and their name appears on the "bill." A bill can include multiple payments and charges, with charges sourced from "services" and "facilities."

Logically, "billed_party" has a many-to-one relationship with "bill" since a party can have many different bills. "Payment" has a many-to-one relationship with "bill," since multiple payments can contribute to a single "bill." "Bill" has a one-to-many relationship with "charge," since many charges can appear on one bill. "Charge" has a one-to-one relationship with "facility" and with "service_type" because each charge covers only one thing.

Problem Statement:

Due to the company's expansion, more data is being generated, but there is no efficient way to manage or use it. This causes the company to fail to track its rooms, which affects the customer experience. As a result, a proper database implementation is needed to ensure the company has quick and accurate access to data related to its operations.

Objectives:

By creating the database, the company will be able to track its customers (of all types, including guests and hosts), rooms, reservation processes, facility usage, transactions, and any other information associated with the rooms, such as events.

Entity and Relationship Summary:

Our ERD model includes several essential entities, including room, event, guest, host, bill, and reservation. Each of these plays a crucial role in representing different aspects of the business operations. In total, the database consists of 30 tables, including several bridge tables to support many-to-many relationships. The report will focus on explaining the key entities and their significant relationships that represent the core business.

Who the Users are:

In terms of the users who make up the company's operations, "guest," "host," "organization," "representative," and "billed_party" are all entities that must be included to ensure the company has a precise understanding of key information. For example, "guest" is required to keep track of who is allocated to which room(s), and "host" is necessary to keep track of who reserved a meeting room. Since organizations can host meetings, having an "organization" entity will be critical to accommodate this, and since organizations need a representative, an entity for one has been created as well. Finally, since the billing party can be either of the entities mentioned earlier or someone who does not fit into any of those groups, we created a separate entity to store information about that specific person.

What Data Must be Tracked:

As for the types of data to be tracked, "guest_history," "event," "room," "reservation," "charge," and "extension_request" are all pieces of information that will be tracked, each providing vital information. For instance, since the hotel needs to keep track of whether a room is under maintenance, an entity has been created to store this information, along with the assigned person. Additionally, since events can be hosted within the hotel, a separate entity has been designed to track them. "Reservation" and "extension_request" were also made to track what customers have already booked and any special requests that may arise. "Guest_history" was created to document a guest's past stays and relationship with the chain, as described. Ultimately, the hotel having detailed records of charges and finances will be vital so that the company can properly manage its relationship with customers (in case they have any questions about their charges) and with its stockholders.

Assumptions/Notes:

“Wing” table notes/assumptions:

- A composite key is required since a wing with the same name can exist in two different buildings. Adding the buildingID makes each combination unique
    - Wing 1A in Building A vs Wing 1A in Building B

“Room” table notes/assumptions:

- A composite key is required since a room with the same name can exist in two different wings
    - Room 1A in Wing A vs Room 1A in Wing B
- Assuming that a suite is treated like any other room (it's given a roomID), it falls under this table

“Room_bed_bridge_table” notes/assumptions:

- A bridging table is needed for the bed and room, since a room can have many types of beds, and a type of bed can appear in multiple rooms

“Function” table notes/assumptions:

- Assuming that we know the other uses and are inputting them, we can get rid of info that dictates the uses, such as whether or not a meeting room has toilet facilities and rollaway beds
    - If a meeting room can serve as a sleeping room, then in the “function” table, it would appear twice: one row denotes it as a “meeting” room, and the other as a “sleeping” room.

“Guest_history” table notes/assumptions:

- Since a paragraph in the business description was dedicated to the customer’s past history with the hotel chain, a table was warranted to capture the customer’s relationship with the hotel. The customer can have many visits to the hotel and produce many bills as a result, with each bill generating a guest_history entity that captures the promptness of payment, a numeric value calculated as the distance between the bill's dueDate and the customer’s paymentTime.

“Expected_usage” tables:

- The business description stated that “the hotel also records expected or authorized charges…” so I created a table of expected facility usage. This entity is created when a reservation is made.

“Event” table notes/assumptions:

- Each event must have exactly one host, but one host may organize multiple events.
- An event is recorded in the system only when both estimatedGuest and roomAssignment are confirmed, so these attributes cannot be NULL.
- Each event must have a start time, and an end time is optional and can be updated later.
- The hostID is a NOT NULL foreign key referencing the "host" table.

"Host" table notes/assumptions:

- A host can be a guest currently staying at the hotel, or an external individual not staying at the hotel.
- The guestID can be a NULL foreign key referencing the "guest" table if the host is not a hotel guest.
- A host is recorded in the system only if they have organized at least one event.

"Guest" table notes/assumptions:

- A guest can attend multiple events, and each event can have multiple guests. This many-to-many relationship is illustrated by a bridge table.
- A guest is recorded in the system only if they have made a reservation at the hotel.
- Each guest can participate in various events, and each event can include various guests. Thus, there is a many-to-many relationship illustrated by the bridge table.
- currentRoomID tracks which room the guest currently occupies, which can be NULL when the guest has checked out or has not yet been assigned a room.
- confidentialFlag is a boolean indicating whether the guest chooses to keep their location private.
- A guest may have a historical record of past stays, which can be NULL if it is their first stay.

"Organization" table notes/assumptions:

- An organization is recorded in the system only if it has at least one business relationship with the hotel, such as hosting an event or being billed for a service.
- Each organization must have an associated billing part, which cannot be NULL.
- An organization has one representative registered in the system. A representative cannot exist without being linked to an organization.
- The type of organization cannot be NULL when the record is created, which specifies the nature of the organization.
- Each organization must have a valid contactNumber for communication, which cannot be NULL.

"Billed_party" table notes/assumptions:

- guestID, hostID, and organizationID can be null. However, they can not all be null (at most two of the three can be null).

- The representative is the one on the bill, not the
- Organizations will be (TBD, don't know if need org entity)

“Charge” table notes/assumptions:

- Charges will all have a bill that it goes towards
- Charges may be assigned to a reservation or may be assigned to a service/facility usage.
- Charges will contain detailed information on the amount and status.

“Reservation” table notes/assumptions:

- Services can not be reserved. Only facilities can.

“Bill” table notes/assumptions:

- Covers all charges for a particular party.

Eating Usage assumptions:

- Regarding the paragraph about different extension times and eating/non-eating usages of meeting times, etc. We decided to omit this information for now and could incorporate it later.

Team Assignment:

We had an initial meeting in which we divided the content into four parts to complete the first round of table-making. We then had three more online team meetings, during which the four of us discussed points of confusion and worked together to fill in the gaps we identified through revision. We asked questions in our team group chat, where we helped each other complete the diagram asynchronously.

ERD Part

Cindy was responsible for creating the following tables: event, host, guest, and guest-event and event-room bridge tables. After our group discussions, she also made the organization table and defined the cardinalities and assumptions among these entities. Through further discussions, she assisted teammates in revising and refining their table designs to ensure overall consistency across the model. Franklin created these tables: room_bed_bridge_table, room_assignment_bridge_table, reservation (later modified by other team members), guest_history, and expected_facility_usage. He also reviewed other members' work and offered suggestions after the initial round of table-making, and eventually standardized the notations and structured the diagram's layout. Max created the tables: billed_party, bill, payment, charge, service_type, facility, reservation (modifier based on Franklin's baseline), and reservation_facility_bridge_table. He also double-checked the relationships in the table and ensured consistency between foreign keys and relationships. William was responsible for creating the tables: building, wing, room, room_function_bridge_table, type_of_bed, function, extension_customer_bridge, and extension_request. Additionally, he helped review all other tables/relationships and provided feedback when needed.

Report Part

Before the team began filling in the report content, Cindy drafted the initial framework and structure, providing a clear outline for the team to build upon, and wrote the section summarizing the hotel's business scope, management objectives, and the goals of ERD model design. Franklin added the relevant assumptions to his report tables. Max worked on the main business operations section. William helped finalize the format for the report and also included any additional information.

Next Steps:

Our next step will be to normalize the database design based on the ERD model we created before implementing our idea in the SQL Workbench. The goal is to reduce dependencies and eliminate redundancy.