E-R DIAGRAM LINK

https://drive.google.com/file/d/1_N6RJYV2p0MKGQhJVjrNOlc_sfBRA51C/view?usp=drive_link

DB RELATION LINK

https://drive.google.com/file/d/1EfY1VN4uQRM_ctE7oEycRPTGIqo8_8Xs/view?usp=drive_link

https://github.com/LuckieMusngi/Ski-Resort-Database

NORMALIZATION ANALYSIS:

Member:

{memberID} → {name, phone, email, dob, emergencyContact}

SkiPass:

{skiPassID} → {price, timeOfPurchase, expDate, totalUses, remainingUses, passType, status, memberID, rentalID}

GearRental:

{rentalID} → {startDate, returnStatus, status, skiPassID}

Equipment:

{equipmentID} → {type, eSize, status}

EquipmentUpdate:

{equipmentUpdateID} → {equipmentID, type, notes, updateDate}

GearRentalUpdate:

{rentalUpdateID} → {rentalID, type, notes, updateDate}

Trail:

{trailName} → {location, difficulty, category, status}

Lift:

{liftName} → {abilityLevel, openTime, closeTime, status}

LessonOrder:

{lessonOrderID} → {memberID, lessonsPurchased, remainingSessions}

Lesson:

{lessonID} → {lessonName, employeeID}

LessonSession:

{sessionID} → {startTime, endTime, lessonID}

Instructor:

{employeeID} → {certificationLevel}

Employee:

{employeeID} → {name, age, sex, race, monthlySalary, jobTitle}

Lodge:

{lodgeID} → {location}

IncomeSource:

{sourceID} → {day, lodgeID, sourceName, dailyIncome}

Shuttle:

{shuttleID} → {shuttleName, status}

Our tables adhere to 3NF because we only have PKs as determinants. This makes our determinants Superkeys as well as being prime attributes. All other attributes are then also dependent on the PK making our tables. To be in 3NF the tables must be in 1NF and 2NF. All tables are in 1NF because every attribute is an atomic value. All tables are in 2NF because removing any determinant breaks the dependency table since the determinants are PKs we have no partial dependency. Also every non-prime attribute is dependent on the PK determinant.

Custom Query description:

The custom query asks for a memberID and prints out information, including their personal information, ski passes, and all of the gear they have rented across those ski passes and gear rentals.

The query prints "Member does not have a ski pass." or "Member does not have any rentals." if the Member has not registered anything yet. And if the member has all of the information the query prints the SkiPassID, rentalID, and the type/size of the equipment they have rented

This is relevant to the database because we can use the information to quickly find rental equipment if a member is speaking to customer service. This can also be used for data analysis to find which types of members rent each type of gear.

This is the link to our E-R Diagram: It is pretty big so it is difficult to maintain readability on PDF
https://lucid.app/lucidchart/98d27958-8089-4674-a97c-65550f0245b4/edit?viewport_loc=-6828%2C1890%2C5860%2C2969%2C0_0&invitationId=inv_5effd6f2-6f2b-43b2-a48d-216e9107f9f7

Entity table

Entities:

Member: memberID, name, phone#, email, dob, emergency contact
(1:M ski pass, 1:M lessonOrder)

Ski pass: skiPassID, price, timeOfPurchase, expDate, totalUses, remainingUses. passType, status, **memberID, rentalID**
(M:1 member, 1:1 gear rental, M:N lift)

Gear Rental: **rentalID**, startDate, expDate, return status, status, **skiPassID**
(1:1 ski pass, 1:M gear rental update, M:N Equipment)

Equipment: **EquipmentID**, type, eSize, status
(M:N gear rental, 1:M Equipment Update)

Equipment Update: equipmentUpdateID, **equipmentID**, type, notes
(M:1 Equipment)

Gear Rental Update: **rentalUpdateID**, **rentalID**, type, notes
(M:1 gear rental)

Trail: **trailName**, location, difficulty, category, status
(M:N lift)

Lift: **liftName**, ability level, openTime, closeTime, status
(M:N trail, M:N ski pass)

LessonOrder: **lessonOrderID**, **memberID**, lessonsPurchased, remainingSessions
(1:M member, M:N Lesson, LessonSession M:N)

Lesson: **LessonID**, lessonName, **EmployeeID**
(1:M LessonSession, M:1 Instructor) // we have each instructor do all of the lesson's sessions

LessonSession: **SessionID**, startTime, endTime**, lessonID** // date is included in startTime and endTime
(M:1 lesson)

Instructor: **EmployeeID**, Certification level // this entity is only necessary if certification matters
(1:1 Employee [subEnt], 1:M Lesson) // an instructor can take multiple lessons

Employee: **EmployeeID**, Name, age, sex, race, monthly salary, job title // ! not sure what other demographic info to put
(M:1 lessonSession, M:N IncomeSource, 1:1 Instructor)

Lodge: **lodgeID**, Location
(M:1 IncomeSource, M:N Shuttle)

IncomeSource: **sourceID,** day**, lodgeID**, sourceName, dailyIncome // these are per day, which is weird but works
(1:M lodge)

Shuttle: **ShuttleID**, shuttleName, status // status is manually set, active/inactive
(M:N lodge)

Relation entities:

LessonToOrder: **SessionID, LessonOrderID //** connects the lessonSessions to the orders

TrailLift: **trailName**, **liftName** // shows which lifts lead to what

LiftPassUsage: **skiPassID**, **liftName**, dateUsed // logs ski pass uses on lifts

ShuttleLodge: **shuttleID, lodgeID** // logs which shuttles go to which lodges

RentalEquipment: **rentalID, EquipmentID** // logs rental's equipments, and equipment rentals

EmployeeIncomeSource: **EmployeeID, sourceID** // logs which employees worked at different income sources (shows shifts)

Prompt Breakdown:

Customers register as members:
- Name, number, email, dob, emergency contact
- Buy ski passes, rent equipment (these 2 are tied together)
  - Ski pass module: # uses, remaining uses, time of purchase, exp date, and member account
  - Equipment rental: Ski boots, ski poles, skis (snowboard, alpine), protective gear
    - Ski boots have sizes, ski poles have lengths, skis have lengths, snowboard has lengths
    - (no notes on protective gear)

Ski resort:
- Contains multiple lodges, each maybe containing: (following are potentially all part of a single lodge)
  - Dining options, gift shops, rental center, visitor center, ski school, free parking lots, paid parking lots
- Free shuttle service
- Each one must keep track of daily income

Skiing area
- Trails:
  - Start, end, status, diff-level (3), category (4), name
- Lifts:
  - Ability level (3), name, openTime, closeTime, trails <- separate entity for MvN, status
  - Uses ski pass

Ski lessons:
- Instructors (M:N or M:1 w/ customers)
- Lesson schedules, purchases, and usage records
- Employee:
  - Job position (an entity), start date, demographic info, monthly salary paid (monthly comes from lodges)

As a ski resort striving to stand out in the local tourism market, we have recently expanded our facilities to include a comprehensive entertainment and leisure area. This aims to enhance the overall customer experience and attract more families and young visitors. While ski passes and skiing activities remain our primary sources of revenue, we also wish to develop this integrated area into a highlight of its own. To achieve this, we hope to build a system that enables close integration between the skiing area and entertainment facilities, thereby improving both customer experience and operational efficiency for staff.

- System, close integration between skiing area and entertainment facilities

The ski resort owns different types of properties, including multiple lodges that offer dining options, several gift shops, a rental center, a visitor center where visitors can buy passes, a ski school, free parking lots, and paid parking lots. The ski resort offers free shuttle service between properties for visitors. The stakeholder may want to keep track of the daily income from different sources.

- Therefore, ski resort vs. the skiing area

Ski resort:
- Contains multiple lodges, each maybe containing: (following are potentially all part of a single lodge)
  - Dining options, gift shops, rental center, visitor center, ski school, free parking lots, paid parking lots
- Free shuttle service
- Each one must keep track of daily income

Before using any skiing facilities, customers must first register as members. For each member, we collect a basic yet essential set of information, e.g., name, telephone number, email address, date of birth, and emergency contact. Once registration is complete, members

may purchase ski passes, rent equipment, and begin their skiing journey either through self-service kiosks or at the front desk. To manage access and skiing rights within the resort, we have designed a "Ski Pass" module. The system should comprehensively track information such as the total number of uses, remaining uses, time of purchase, expiration date, and associated member account. To note, we offer different types of passes, e.g., 1-day, 2-day, 4-day, and season passes.

The ski resort provides equipment rental services to visitors who need them. Equipment rental is tied to the ski pass: one use of the ski pass allows for one equipment rental. Equipment includes ski boots, ski poles, skis (both snowboard and alpine), and protective gear. Ski boots range from size 4 to 14 with half sizes, the length of ski poles ranges from 100cm to 140cm (integer), the length of ski ranges from 115cm to 200cm (integer), and the length of snowboard ranges from 90cm to 178cm (integer). The system must manage inventory and rental records.

Customers register as members:

Name, number, email, dob, emergency contact

Buy ski passes, rent equipment (these 2 are tied together)

Ski pass module: # uses, remaining uses, time of purchase, exp date, and member account

Equipment rental: Ski boots, ski poles, skis (snowboard, alpine), protective gear

Ski boots have sizes, ski poles have lengths, skis have lengths, snowboard has lengths

(no notes on protective gear)

The ski resort has a lot of trails for visitors to have fun. Each trail contains the start and end location, status for open, the difficulty level (beginner, intermediate, expert, for simplicity), and category (groomed, park, moguls, and glade skiing). We also want to offer different names for each trail so that people can easily remember them. Feel free to come up with creative names.

Trails:

Start, end, status, diff-level (3), category (4), name

To reach the trails on the mountain, visitors can take the lifts. Each lift has its unique name for people to remember. To prevent possible accidents, the lift should indicate its ability level (beginner, intermediate, expert, for simplicity) to better accommodate the visitors. Each lift has a regular open and close time within a day for visitor to plan their runs in the resort. Besides, visitors also want to know which trails each lift takes them to. Visitors also want to check their status before they visit, since some of the lifts may open later in the season for a deeper snow cover. Members scan their ski pass at the entrance of the lift. A valid entry counts as one ski pass use. The system records all entry logs for people who want to check their statistics.

Lifts:

Ability level (3), name, openTime, closeTime, trails <- separate entity for MvN, status

Uses ski pass

The ski school offers ski lessons for kids and adults. The resort employs PSIA-AASI certified instructors (Level I, II, III). Instructors can manage private and group lessons. The system must store lesson schedules, purchases, and usage records.

The ski resort employs a lot of people to maintain the service. Their job positions should cover all the services provided above. The ski resort also wants to keep track of employees' demographic information and employment start dates. Besides, the stakeholder wants to know the gross income per month by calculating the difference between the monthly income from all the properties and the monthly salary paid to every employee.

Ski lessons:

Instructors (MVN or MV1 w/ customers)

Lesson schedules, purchases, and usage records

Employee:

Job position (an entity), start date, demographic info, monthly salary paid (monthly comes from lodges)

LOGS:
// trail (..., start,end, ..) changed to location

// member to ski pass changed to 1:M (and vice versa) instead of 1:1
// removed memberID from gear rental (also changing entity relation 1:1 thing)
// added status attribute to ski pass for archival
// added status attribute to gear rental bc I think its included in ski pass so it also gets archived
// changed equipment length to size
// added equipment update (same as gear rental but for equipment)

// removed employeeID from lesson session (therefore, M:1 instructor is gone from lesson session)
// added employeeID to lesson (therefore, lesson has M:1 instructor)

// added status to lift
// added status to trail

// ! this is some new stuff
// ! removed expDate from gear rental
// ! changed size to eSize in equipment
// ! removed date from lessionSession
// ! removed date from liftpassusage

// ! removed day from employeeincomesource
// ! added dates to gear rental update and equipment update
// ! **lessonToOrder connects to lessonSession**

Status = active, inactive // this is used on deletion