Ski Resort Database Design

Group 14: Connor, Luis, Mohammad, Nathan

Spring 2025

Contents

§1 Conceptual database design

Figure 1 shows the E–R diagram of the ski resort database. To design the conceptual schema of the database we first did a detailed walk through fot the program specifications to record what tables are going to be needed to store data. The ER diagram we made captures the ski resorts day to day operations tracking members and their activites as well as the resorts staff and income via its services. Attributes are constrained via sql datatypes.

§2 Logical database design

Below is the final relational schema derived from the E–R diagram. Each table lists its attributes, primary keys (**PK**), and foreign keys (**FK**) explicitly. This design reflects a direct and normalized translation of the conceptual model into the relational model used in Oracle SQL.

- **Property**(<u>propertyID</u>, name, address, propertyType)
- **Shop**(<u>shopID</u>, name, shopType, buildingID$^{\mathrm{FK}}$, income)
- **Equipment**(<u>equipmentID</u>, equipmentType, equipmentSize, archived)
- **Rental**(<u>rentalID</u>, equipmentID$^{\mathrm{FK}}$, passID$^{\mathrm{FK}}$, rentalTime, returnStatus)
- **Pass**(<u>passID</u>, memberID$^{\mathrm{FK}}$, numUses, passType, price, exprDATE)
- **LiftLog**(passID$^{\mathrm{FK}}$, liftID$^{\mathrm{FK}}$, dateTime)
- **Lift**(<u>liftID</u>, liftName, openTime, closeTime, abilityLevel, status)

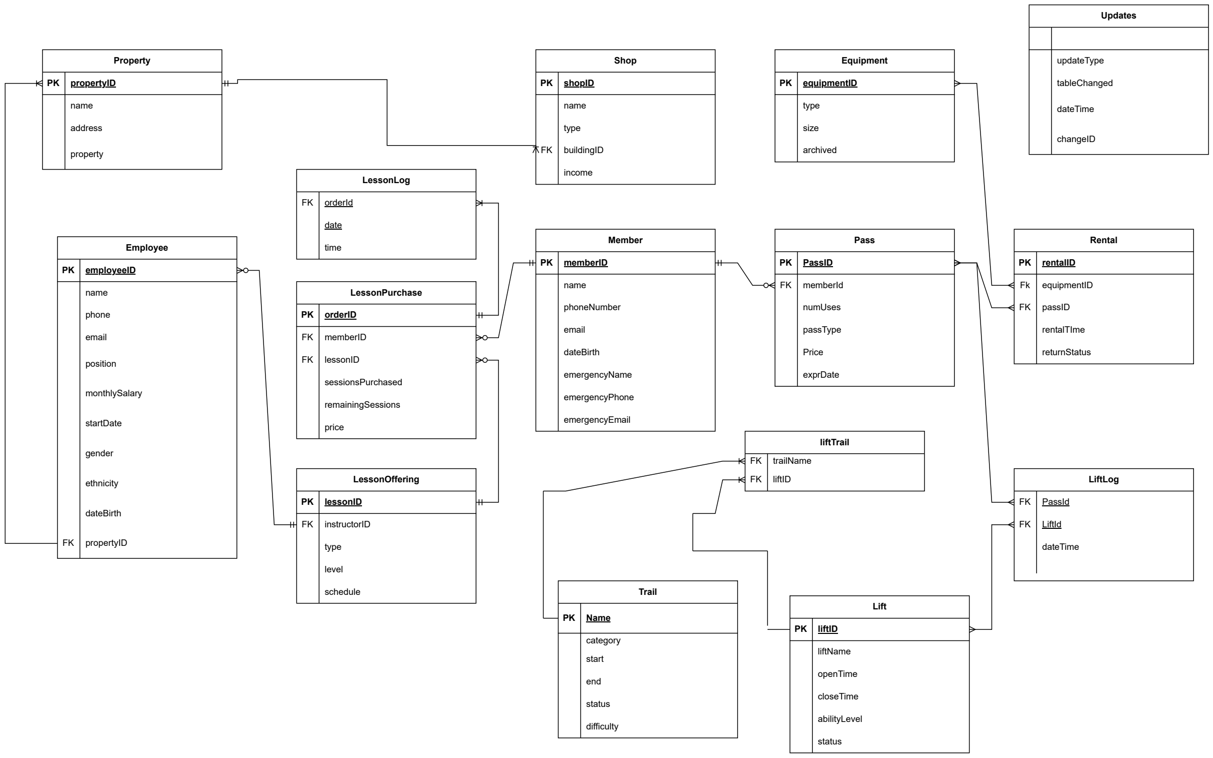

Figure 1: E–R Diagram of the Ski Resort Database

- **TrailLift**(liftID$^{\text{FK}}$, trailName$^{\text{FK}}$)
- **Trail**(<u>name</u>, category, startPos, endPos, status, difficulty)
- **Member**(<u>memberID</u>, name, phoneNumber, email, dateBirth, emergencyName, emergencyPhone, emergencyEmail)
- **LessonLog**(orderID$^{\text{FK}}$, dateTime)
- **LessonPurchase**(<u>orderID</u>, memberID$^{\text{FK}}$, lessonID$^{\text{FK}}$, sessionsPurchased, remainingSessions, price)
- **LessonOffering**(<u>lessonID</u>, instructorID$^{\text{FK}}$, lessonType, skillLevel, schedule)
- **Employee**(<u>employeeID</u>, name, phone, email, position, monthlySalary, startDate, gender, ethnicity, dateBirth, propertyID$^{\text{FK}}$)
- **Updates**(updateType, tableChanged, changeID, dateTime)

§3 Normalization analysis

Normalization Analysis

All relations in the schema adhere to the First Normal Form (1NF) as all attributes are not set values. They also conform to the Second Normal Form (2NF) since all non-prime attributes are fully functionally dependent on the candidate key(s). Below, we analyze each relation for Boyce-Codd Normal Form (BCNF) or the Third Normal Form (3NF).

1. `Property(propertyID, name, address, propertyType)`
   - **Functional Dependencies:**

     propertyID → name, address, propertyType

   - **Candidate Key:** propertyID
   - **BCNF Status:** *Yes*, all functional dependencies have a superkey on the left-hand side.

2. `Shop(shopID, name, type, buildingID, income)`
   - **FDs:** shopID → name, type, buildingID, income
   - **PK:** shopID
   - **BCNF:** Yes because all attributes are functionally dependent on the primary key.

3. `Equipment(equipmentID, type, size, archived)`
   - **FDs:** equipmentID → type, size, archived
   - **PK:** equipmentID
   - **BCNF:** Yes, all attributes are functionally dependent on the primary key.

4. `Rental(rentalID, equipmentID, passID, rentalTime, returnStatus)`
   - **FDs:** rentalID → equipmentID, passID, rentalTime, returnStatus
   - **PK:** rentalID
   - **BCNF:** Yes

5. `Pass(passID, memberID, numUses, passType, price, exprDate)`
   - **FDs:** passID → memberID, numUses, passType, price, exprDate
   - **PK:** passID
   - **BCNF:** Yes

6. `LiftLog(passID, liftID, dateTime)`
   - **FDs:** (passID, liftID, dateTime) is the composite key; no partial dependencies.
   - **PK:** Composite (passID, liftID, dateTime)
   - **BCNF:** Yes

7. `Lift(liftID, liftName, openTime, closeTime, abilityLevel, status)`
   - **FDs:** liftID → liftName, openTime, closeTime, abilityLevel, status
   - **PK:** liftID
   - **BCNF:** Yes

8. `TrailLift(liftID, trailName)`

- **FDs:** Composite PK (liftID, trailName); no non-prime attributes
- **BCNF:** Yes

9. `Trail(name, category, start, end, status, difficulty)`

- **FDs:** name → category, start, end, status, difficulty
- **PK:** name
- **BCNF:** Yes

10. `Member(memberID, name, phoneNumber, email, dateBirth, emergencyName, emergencyPhone, emergencyEmail)`

- **FDs:** memberID → all other attributes
- **PK:** memberID
- **BCNF:** Yes

11. `LessonLog(orderID, dateTime)`

- **FDs:** orderID → dateTime
- **PK:** orderID
- **BCNF:** Yes

12. `LessonPurchase(orderID, memberID, lessonID, sessionsPurchased, remainingSessions, price)`

- **FDs:** orderID → memberID, lessonID, sessionsPurchased, remainingSessions, price
- **PK:** orderID
- **BCNF:** Yes

13. `LessonOffering(lessonID, instructorID, lessonType, skillLevel, schedule)`

- **FDs:** lessonID → instructorID, lessonType, skillLevel, schedule
- **PK:** lessonID
- **BCNF:** Yes

14. `Employee(employeeID, name, phone, email, position, monthlySalary, startDate, gender, ethnicity, dateBirth, propertyID)`

- **FDs:** employeeID → all other attributes
- **PK:** employeeID
- **BCNF:** Yes

15. `Updates(updateType, tableChanged, changeID, dateTime)`

- **FDs:** dateTime → updateType, tableChanged, changeID
  Note: This assumes that 'dateTime' is unique (no two updates occur at the same time).
- **PK:** A composite key of (updateType, tableChanged, changeID, dateTime) is possible, but not necessary.
- **BCNF:** Yes.

§4 Query description

§4.1 Custom Query: Monthly Income Summary

Query Goal: Calculate the gross monthly income of the resort by subtracting total employee salaries from the sum of all incomes recorded across the properties or at one specific property.

Motivation: This query helps stakeholders monitor the profitability of the resort's operations, combining staff payroll and property performance in a single monthly snapshot.

Relations Involved:

- `Property`
- `Shop`
- `Employee`

Query Details: For each month, aggregate total income from properties (e.g., gift shops, rental centers), subtract the sum of salaries of all employees, and report the net income. This could be extended to include breakdowns by property type or department.