

Ski Resort Database Design

GROUP 14: CONNOR, LUIS, MOHAMMAD, NATHAN

Spring 2025

Contents

1	Conceptual database design	1
2	Logical database design	1
3	Normalization analysis	2
4	Query description	3
4.1	Custom Query: Monthly Income Summary	3

§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 for 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 activities 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**(propertyID, name, address, propertyType)
- **Shop**(shopID, name, shopType, buildingID^{FK}, income)
- **Equipment**(equipmentID, equipmentType, equipmentSize, archived)
- **Rental**(rentalID, equipmentID^{FK}, passID^{FK}, rentalTime, returnStatus)
- **Pass**(passID, memberID^{FK}, numUses, passType, price, expirDATE)
- **LiftLog**(passID^{FK}, liftID^{FK}, dateTime)
- **Lift**(liftID, liftName, openTime, closeTime, abilityLevel, status)

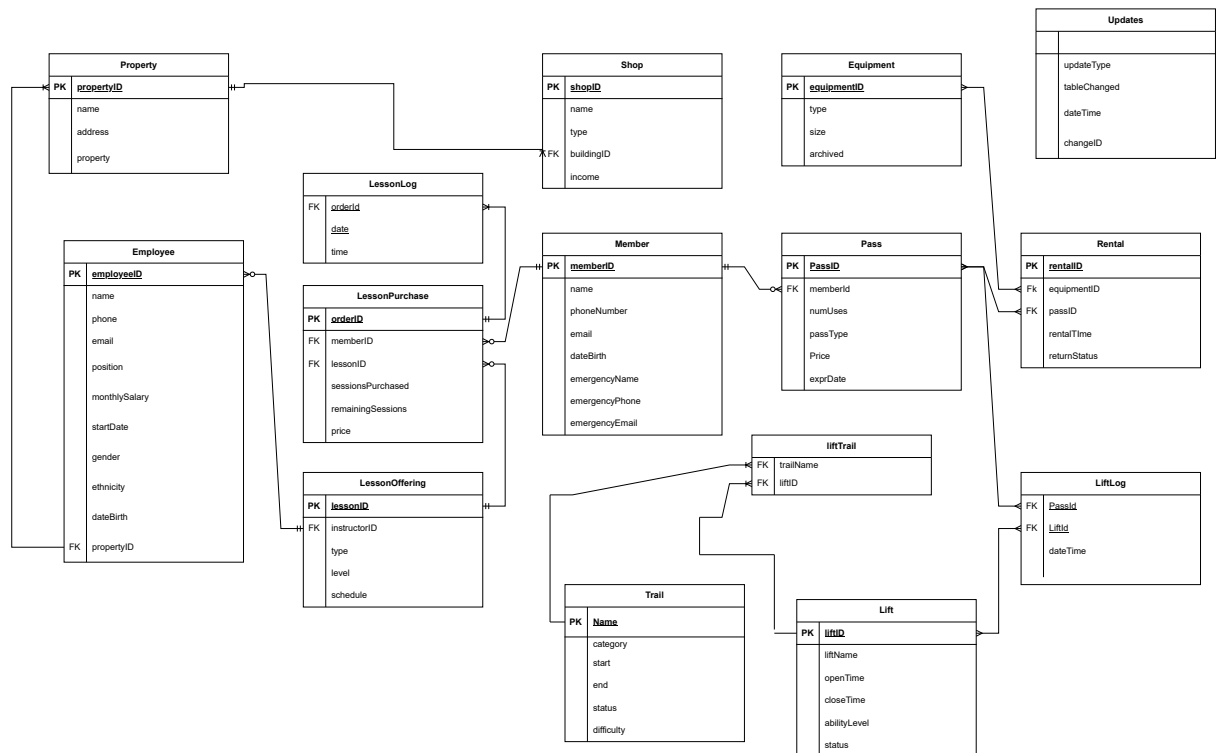


Figure 1: E–R Diagram of the Ski Resort Database

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

§3 Normalization analysis

These all adhere to 1NF as all set attributes are non set values. Also, they all adhere to 2NF because all non-prime attributes are fully functionally dependant on the candidate keys.

Property(propertyID, name, address, propertyType) FDs: $\text{propertyID} \rightarrow \text{name}$, $\text{propertyID} \rightarrow \text{address}$, $\text{propertyID} \rightarrow \text{propertyType}$ This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

Shop(shopID, name, type, buildingID, income) FDs: shopID \rightarrow name, shopID \rightarrow type, shopID \rightarrow buildingID, shopID \rightarrow income This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

Equipment(equipmentID, type, size, archived) FDs: equipmentID \rightarrow type, equipmentID \rightarrow size, equipmentID \rightarrow archived This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

Rental(rentalID, equipmentID, passID, rentalTime, returnStatus) FDs: rentalID \rightarrow equipmentID, rentalID \rightarrow passID, rentalID \rightarrow rentalTime, rentalID \rightarrow returnStatus This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

Pass(passID, memberID, numUses, passType, price, exprDate) FDs: passID \rightarrow memberID, passID \rightarrow numUses, passID \rightarrow passType, passID \rightarrow price, passID \rightarrow exprDate This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

LiftLog(passIDFK, liftID, dateTime) FDs: passID \rightarrow liftID, passID \rightarrow dateTime This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

Lift(liftID, liftName, openTime, closeTime, abilityLevel, status) FDs: liftID \rightarrow liftName, liftID \rightarrow openTime, liftID \rightarrow closeTime, liftID \rightarrow abilityLevel, liftID \rightarrow status This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

TrailLift(liftID, trailName) FDs: liftID \rightarrow trailName This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

Trail(name, category, start, end, status, difficulty) FDs: name \rightarrow category, name \rightarrow start, name \rightarrow end, name \rightarrow status, name \rightarrow difficulty This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

Member(memberID, name, phoneNumber, email, dateBirth, emergencyName, emergencyPhone, emergencyEmail) FDs: memberID \rightarrow name, memberID \rightarrow phoneNumber, memberID \rightarrow email, memberID \rightarrow dateBirth, memberID \rightarrow emergencyName, memberID \rightarrow emergencyPhone, memberID \rightarrow emergencyEmail This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

LessonLog(orderID, dateTime) FDs: orderID \rightarrow dateTime This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

LessonPurchase(orderID, memberID, lessonID, sessionsPurchased, remainingSessions, price) FDs: orderID \rightarrow memberID, orderID \rightarrow lessonID, orderID \rightarrow sessionsPurchased, orderID \rightarrow remainingSessions, orderID \rightarrow price This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

LessonOffering(lessonID, instructorID, type, level, schedule) FDs: lessonID \rightarrow instructorID, lessonID \rightarrow type, lessonID \rightarrow level, lessonID \rightarrow schedule This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

Employee(employeeID, name, phone, email, position, monthlySalary, startDate, gender, ethnicity, dateBirth, propertyIDFK) FDs: employeeID \rightarrow name, employeeID \rightarrow phone, employeeID \rightarrow email, employeeID \rightarrow position, employeeID \rightarrow monthlySalary, employeeID \rightarrow startDate This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

Updates(updateType, tableChanged, changeID, dateTime) FDs: dateTime \rightarrow updateType, dateTime \rightarrow tableChanged, dateTime \rightarrow changeID This is in BCNF as for all of the FDs $X \rightarrow Y$, X is a superkey (it is the PK)

§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.