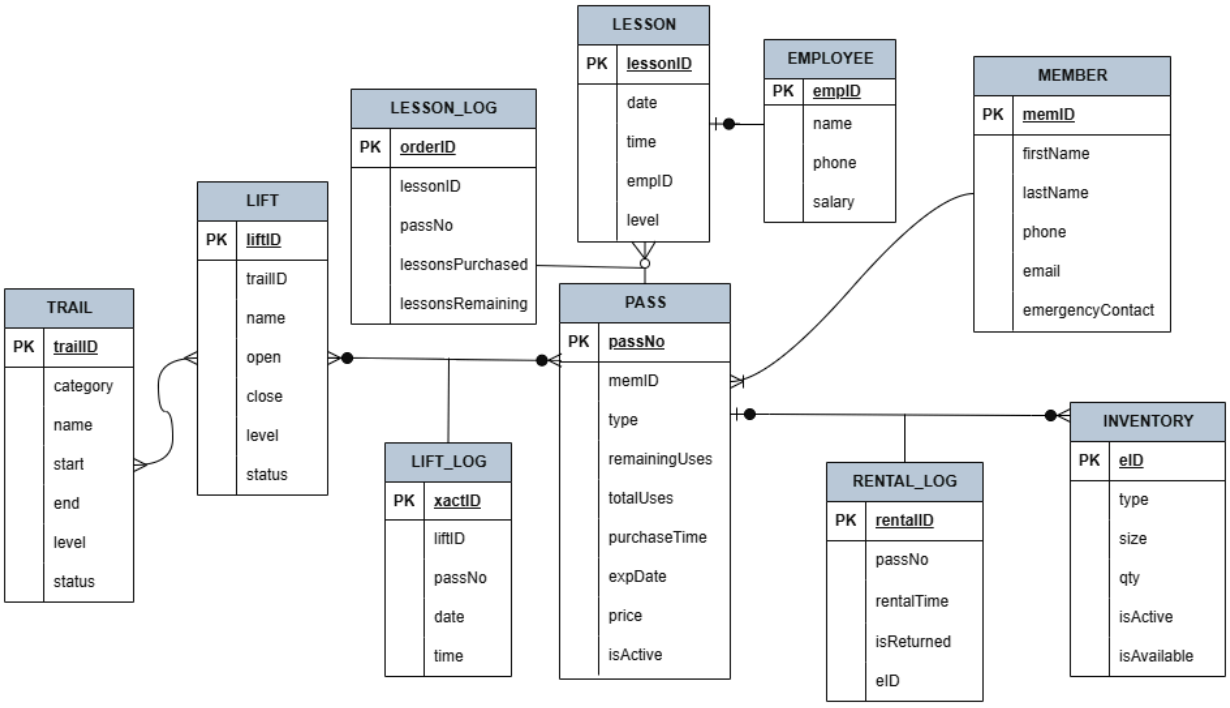
ENTITY RELATIONSHIP DIAGRAM



Entities and Attributes:

1. MEMBER
    a. Represents individuals registered at the resort.
    b. Attributes: memID (PK), firstName, lastName, phone, email, emergencyContact.
    c. Each member is uniquely identified by memID and may purchase multiple passes, rent equipment, and book lessons.
2. PASS
    a. Represents a ski pass associated with a member.
    b. Attributes: passNo (PK), memID (FK), type, remainingUses, totalUses, purchaseTime, expDate, price, isActive
    c. The isActive flag allows for archival without deletion. A pass is valid until uses are exhausted or the expiration date is reached.
3. EMPLOYEE
    a. Represents instructors and staff working at the resort
    b. Attributes: empID (PK), name, phone, salary
    c. Employees are linked to lessons they teach, if applicable.
4. LESSON
    a. Represents a scheduled lesson session.
    b. Attributes: lessonID (PK), date, time, empID (FK), level.

    c. Lessons are assigned an instructor and associated with a skill level (e.g., beginner, intermediate).
5. LESSON_LOG
    a. Tracks lesson purchases and uses by members.
    b. Attributes: orderID (PK), lessonID (FK), passNo (FK), lessonsPurchased, lessonsRemaining.
    c. Reflects a M:N relationship between PASS (via passNo) and LESSON.
6. TRAIL
    a. Represents a ski trail at the resort
    b. Attributes: trailID (PK), category, name, start, end, level, status
    c. Trails are categorized by difficulty level (beginner, intermediate, expert), and category (groomed, park, moguls, and glade skiing).
    d. The status flag indicates whether a trail is currently open or not.
7. LIFT
    a. Represents ski lifts connected to trails.
    b. Attributes: liftID (PK), trailID (FK), name, open, close, level, status.
    c. Lifts operate on specific trails and are categorized by difficulty level.
    d. The status flag again indicates whether a lift is currently operational.
8. LIFT_LOG
    a. Tracks lift usage per ski pass.
    b. Attributes: xactID (PK), liftID (FK), passNo (FK), date, time
    c. Records when and where a pass was used.
    d. Reflects a M:N relationship between PASS and LIFT.
9. INVENTORY
    a. Represents available equipment for rent
    b. Attributes: eID (PK), type, size, qty, isActive, isAvailable.
    c. The isActive flag indicates whether equipment is archived. The isAvailable flag is used to check whether a piece of equipment is currently rented out or reserved.
10. RENTAL_LOG
    a. Tracks rental activity.
    b. Attributes: rentalID (PK), passNo (FK), rentalTime, isReturned, eID (FK)
    c. Records when equipment is rented and whether it has been returned.
    d. Implements a 1:M relationship between PASS and INVENTORY

Relationships:

1. MEMBER–PASS
    a. Cardinality: 1:M. A member may own multiple passes.
2. PASS–LIFT_LOG
    a. Cardinality: 1:M. A pass may be used on multiple lifts.

3. PASS–RENTAL_LOG
    a. Cardinality: 1:M. Rentals are tied to specific passes.
4. PASS–LESSON_LOG
    a. Cardinality: 1: M. Lesson purchases are tracked per pass.
5. EMPLOYEE–LESSON
    a. Cardinality: 1:M. An instructor may teach many lessons.
6. LIFT–LIFT_LOG
    a. Cardinality: 1:M. Logs reflect each instance a lift is used.
7. TRAIL–LIFT
    a. Cardinality: 1:M. A trail can have multiple lifts.
8. INVENTORY–RENTAL_LOG
    a. Cardinality: 1:M. Each rental log references a specific item.

Additional Information/Constraints:

1. Deletion:
    a. Members cannot be deleted if they have active passes, rentals, or lessons remaining.
    b. Passes are only deletable (archived) if expired and fully used.
    c. Equipment must not be deleted unless it’s not rented or reserved.
2. Additional attribute information:
    a. isActive flags in PASS and INVENTORY are used to archive rather than delete passes and equipment.
    b. isAvailable and isReturned track status of equipment.

Normalization Analysis

All of the relations along with their functional dependencies are listed below.
1NF: A relation is in 1NF if it does not contain any set-valued attributes.
2NF: A relation is in 2NF if each non-prime attribute (each attribute that is not part of a candidate key) is fully functionally dependent on every candidate key.
3NF: A relation is in 3NF if, for every non-trivial FD X-> A that holds, either X is a superkey of the relation, or A is a prime attribute.
BCNF: A relation is in BCNF if for each functional dependency X->A, X is a superkey of the underlying relation.

1. MEMBER
   Functional Dependencies:
   memID -> firstName, lastName, phone, email, emergencyContact

   Analysis:
   1NF: no set valued attributes exist.
   2NF: The only prime attribute and candidate key of this table is its primary key, memID, and every other non-prime attribute depends on it. Furthermore, the FD above only contains that key on its LHS, so removing it would necessarily destroy the FD. This means that the FD is a full functional dependency, and that the table is in 2NF.
   3NF: There is only one non-trivial FD, and in it, X= memID, the primary key of the relation, which is also a superkey. Therefore, MEMBER is in 3NF.
   BCNF: Because the above FD meets only the first requirement of 3NF implies that MEMBER is also in BCNF.

2. PASS
   Functional Dependencies:
   passNo -> memID, type, remainingUses, totalUses, purchaseTime, expDate, price, isActive

   Analysis:
   1NF: no set valued attributes exist.
   2NF: The only prime attribute and candidate key of PASS is its primary key, passNo, and every other non-prime attribute depends on it. Furthermore, the FD above only contains that key on its LHS, so removing it would necessarily destroy the FD. This means that the FD is a full functional dependency, and that the table is in 2NF.

3NF: There is only one non-trivial FD, and in it, X= passNo, the primary key of PASS, which is also a superkey. Therefore, PASS is in 3NF.
BCNF: Because the above FD meets only the first requirement of 3NF implies that PASS is also in BCNF.

3. EMPLOYEE
   Functional Dependencies:
   empID -> name, phone, salary

   Analysis:
   1NF: no set valued attributes exist.
   2NF: The only prime attribute and candidate key of EMPLOYEE is its primary key, empID, and every other non-prime attribute depends on it. Furthermore, the FD above only contains that key on its LHS, so removing it would necessarily destroy the FD. This means that the FD is a full functional dependency, and that the table is in 2NF.
   3NF: There is only one non-trivial FD, and in it, X= empID, the primary key of EMPLOYEE, which is also a superkey. Therefore, EMPLOYEE is in 3NF.
   BCNF: Because the above FD meets only the first requirement of 3NF implies that EMPLOYEE is also in BCNF.

4. LESSON
   Functional Dependencies:
   lessonID -> date, time, empID, level

   Analysis:
   1NF: no set valued attributes exist.
   2NF: The only prime attribute and candidate key of LESSON is its primary key, lessonID, and every other non-prime attribute depends on it. Furthermore, the FD above only contains that key on its LHS, so removing it would necessarily destroy the FD. This means that the FD is a full functional dependency, and that the table is in 2NF.
   3NF: There is only one non-trivial FD, and in it, X= lessonID, the primary key of LESSON, which is also a superkey. Therefore, LESSON is in 3NF.
   BCNF: Because the above FD meets only the first requirement of 3NF implies that LESSON is also in BCNF.

5. LESSON_LOG
   Functional Dependencies:

orderID -> memID, lessonID, lessonsPurchased, lessonsRemaining

Analysis:
1NF: no set valued attributes exist.
2NF: The only prime attribute and candidate key of LESSON_LOG is its primary key, orderID, and every other non-prime attribute depends on it. Furthermore, the FD above only contains that key on its LHS, so removing it would necessarily destroy the FD. This means that the FD is a full functional dependency, and that the table is in 2NF.
3NF: There is only one non-trivial FD, and in it, X= orderID, the primary key of LESSON_LOG, which is also a superkey. Therefore, LESSON_LOG is in 3NF.
BCNF: Because the above FD meets only the first requirement of 3NF implies that LESSON_LOG is also in BCNF.

6. TRAIL
   Functional Dependencies:
   trailID -> name, level, start, end, status, category

   Analysis:
   1NF: no set valued attributes exist.
   2NF: The only prime attribute and candidate key of TRAIL is its primary key, trailID, and every other non-prime attribute depends on it. Furthermore, the FD above only contains that key on its LHS, so removing it would necessarily destroy the FD. This means that the FD is a full functional dependency, and that the table is in 2NF.
   3NF: There is only one non-trivial FD, and in it, X= trailID, the primary key of TRAIL, which is also a superkey. Therefore, TRAIL is in 3NF.
   BCNF: Because the above FD meets only the first requirement of 3NF implies that TRAIL is also in BCNF.

7. LIFT
   Functional Dependencies:
   liftID -> name, open, close, trailID, level, status

   Analysis:
   1NF: no set valued attributes exist.
   2NF: The only prime attribute and candidate key of LIFT is its primary key, liftID, and every other non-prime attribute depends on it. Furthermore, the FD above only contains that key on its LHS, so removing it would necessarily destroy the

FD. This means that the FD is a full functional dependency, and that the table is in 2NF.
3NF: There is only one non-trivial FD, and in it, X= liftID, the primary key of LIFT, which is also a superkey. Therefore, LIFT is in 3NF.
BCNF: Because the above FD meets only the first requirement of 3NF implies that LIFT is also in BCNF.

8. LIFT_LOG
   Functional Dependencies:
   xactID -> liftID, passNo, date, time

   Analysis:
   1NF: no set valued attributes exist.
   2NF: The only prime attribute and candidate key of LIFT_LOG is its primary key, xactID, and every other non-prime attribute depends on it. Furthermore, the FD above only contains that key on its LHS, so removing it would necessarily destroy the FD. This means that the FD is a full functional dependency, and that the table is in 2NF.
   3NF: There is only one non-trivial FD, and in it, X= xactID, the primary key of LIFT_LOG, which is also a superkey. Therefore, LIFT_LOG is in 3NF.
   BCNF: Because the above FD meets only the first requirement of 3NF implies that LIFT_LOG is also in BCNF.

9. INVENTORY
   Functional Dependencies:
   eID -> type, size, qty, isActive, isAvailable

   Analysis:
   1NF: no set valued attributes exist.
   2NF: The only prime attribute and candidate key of INVENTORY is its primary key, eID, and every other non-prime attribute depends on it. Furthermore, the FD above only contains that key on its LHS, so removing it would necessarily destroy the FD. This means that the FD is a full functional dependency, and that the table is in 2NF.
   3NF: There is only one non-trivial FD, and in it, X= eID, the primary key of INVENTORY, which is also a superkey. Therefore, INVENTORY is in 3NF.
   BCNF: Because the above FD meets only the first requirement of 3NF implies that INVENTORY is also in BCNF.

10. RENTAL_LOG

Functional Dependencies:

rentalID -> passNo, rentalTime, isReturned

Analysis:

1NF: no set valued attributes exist.

2NF: The only prime attribute and candidate key of RENTAL_LOG is its primary key, rentalID, and every other non-prime attribute depends on it. Furthermore, the FD above only contains that key on its LHS, so removing it would necessarily destroy the FD. This means that the FD is a full functional dependency, and that the table is in 2NF.

3NF: There is only one non-trivial FD, and in it, X= rentalID, the primary key of RENTAL_LOG, which is also a superkey. Therefore, RENTAL_LOG is in 3NF.

BCNF: Because the above FD meets only the first requirement of 3NF implies that RENTAL_LOG is also in BCNF.

Self-designed Query

For a date entered by the user, show all instructors who had scheduled lessons and how many students each taught.

This query can be used to see demand trends for lessons, as well as to assess the distribution and scheduling of instructors. It can help management better understand which instructors are popular and which days are most popular for lessons.This can affect scheduling, pricing, and revenue.