3. List out the entities and identify the relationship between them. Also, identify related attributes supposed to be recorded while considering the normalization rule.

In a Face Recognition System (FRS), several entities can be identified along with their relationships and related attributes. The design and organization of these entities should consider database normalization rules for efficient data management and accuracy. Below, I'll list some key entities, their relationships, and related attributes in the context of an FRS:

Entities:

1 **User**:

* Attributes: UserID, Username, Password, Email, Full Name, User Type (e.g., regular user, administrator), Registration Date, Status (active or inactive).
* Relationships: Users can have multiple face images registered.

**2.Face Image**:

* Attributes: FaceImageID, UserID (foreign key), Facial Features (e.g., distances between facial landmarks), Image Timestamp, Image Quality, Image Location (URL or file path).
* Relationships: Belongs to a User. Used for face recognition.

**Access Log**:

* Attributes: LogID, UserID (foreign key), Access Timestamp, Action (e.g., login, logout, failed authentication), Device/Location (where access was attempted), Result (success or failure).
* Relationships: Linked to the User entity for user-specific access logs.

4. **System Administrator**:

* Attributes: AdminID, AdminName, AdminEmail, Password.
* Relationships: None or a separate "Admin Log" entity to track administrative actions.

5. **Face Recognition Data**:

* Attributes: FaceDataID, FaceImageID (foreign key), Data Type (e.g., feature vectors, Eigenfaces, eigenvalues), Data Timestamp.
* Relationships: Associated with a Face Image for data storage and comparison.

6. **Access Control System**:

* Attributes: DeviceID, DeviceName, Location, Access Control Policies.
* Relationships: May relate to the Access Log for tracking access control device activities.

Relationships:

**User** has a **one-to-many** relationship with **Face Image**, as a user can have multiple registered face images.

**Face Image** has a **one-to-one** relationship with **Face Recognition Data**, as each face image is associated with specific recognition data.

**User** has a **one-to-many** relationship with **Access Log**, as a user can have multiple access log entries.

**ystem Administrator** may have a **one-to-many** relationship with an **Admin Log** entity to record administrative activities.

**ccess Log** may have a **many-to-one** relationship with **Access Control System**, representing which device/location an access attempt was made.

Normalization:

**First Normal Form (1NF)**:

* Each attribute in an entity should contain atomic values (no multi-valued attributes).
* Make sure that attributes like Facial Features are properly structured and not containing multiple values.
* **Second Normal Form (2NF)**:
  + The database should be in 1NF, and all non-key attributes should be functionally dependent on the entire primary key.
  + Splitting the data into multiple tables can help achieve 2NF.
* **Third Normal Form (3NF)**:
  + The database should be in 2NF, and there should be no transitive dependencies between non-key attributes.
  + For example, if there are derived attributes, consider moving them to separate tables.
* **Eliminate Partial Dependencies**:
  + Ensure that attributes depend on the entire primary key and not just a part of it.
* **Eliminate Redundancy**:
  + Avoid storing the same data in multiple places. For example, a user's name should only be recorded once, not for every access log entry.

Normalization helps minimize data redundancy, improve data consistency, and enhance the efficiency of queries and updates. The specific organization and normalization of the database will depend on the system's requirements and the actual data model used (e.g., relational database, NoSQL, etc.).

Top of Form