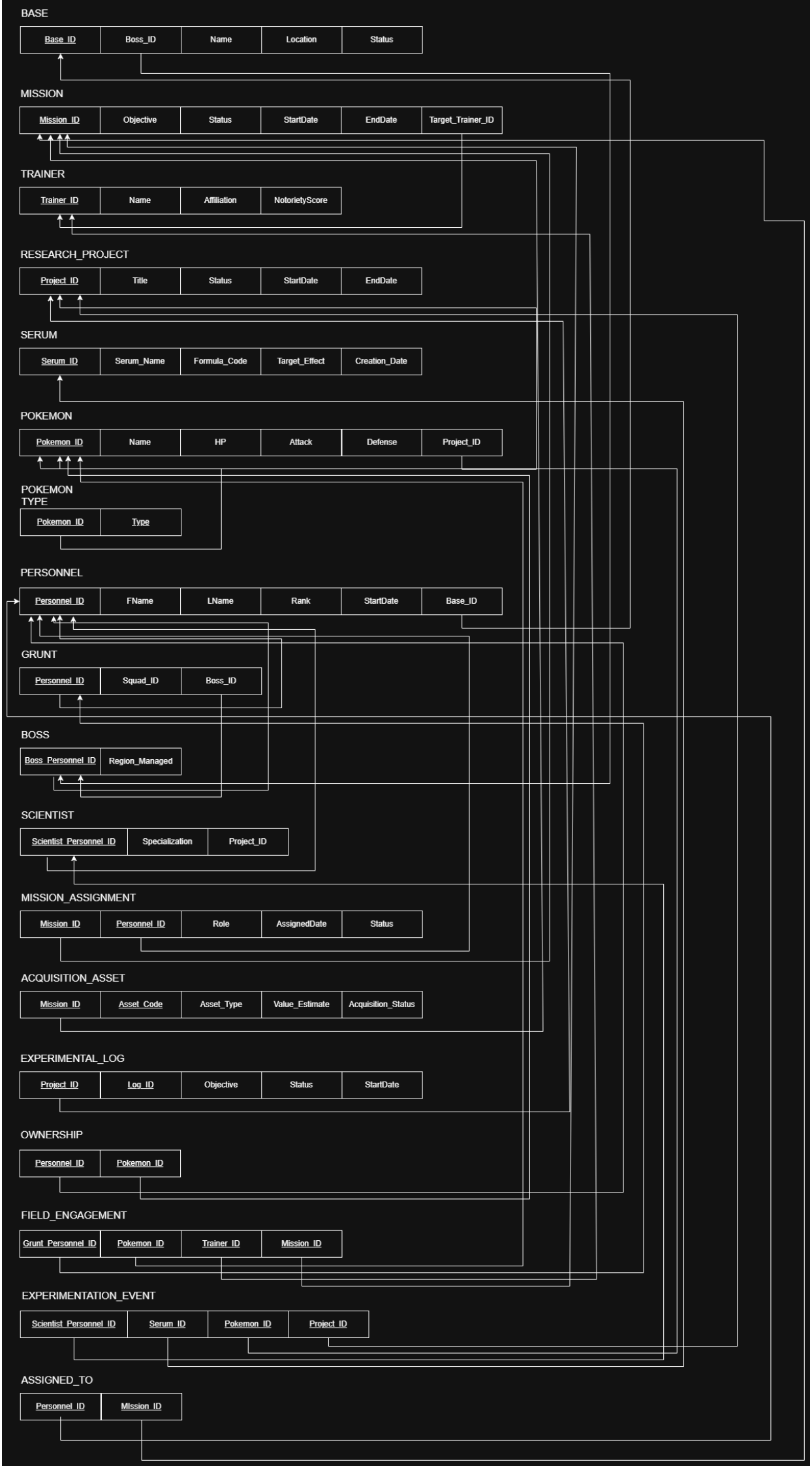
Team number - 88 (Data Dynasty)

1. Relational Model Conversion

This section summarizes the conversion of the conceptual Entity-Relationship (ER) model for **Project Overlord** into a set of normalized relations (tables), detailing the changes made at each stage (1NF, 2NF, 3NF).

1.1. Relational Model After Mapping ER to Relational Model


BASE
Base_ID
Boss_ID
Name
Location
Status
MISSION
Mission_ID
Objective
Status
StartDate
EndDate
Target_Trainer_ID
TRAINER
Trainer_ID
Name
Affiliation
NotorietyScore
RESEARCH_PROJECT
Project_ID
Title
Status
StartDate
EndDate
SERUM
Serum_ID
Serum_Name
Formula_Code
Target_Effect
Creation_Date
POKEMON
Pokemon_ID
Name
HP
Attack
Defense
Project_ID
POKEMON TYPE
Pokemon_ID
Type
PERSONNEL
Personnel_ID
FName
LName
Rank
StartDate
Base_ID
GRUNT
Personnel_ID
Squad_ID
Boss_ID
BOSS
Boss_Personnel_ID
Region_Managed
SCIENTIST
Scientist_Personnel_ID
Specialization
Project_ID
MISSION_ASSIGNMENT
Mission_ID
Personnel_ID
Role
AssignedDate
Status
ACQUISITION_ASSET
Mission_ID
Asset_Code
Asset_Type
Value_Estimate
Acquisition_Status
EXPERIMENTAL_LOG
Project_ID
Log_ID
Objective
Status
StartDate
OWNERSHIP
Personnel_ID
Pokemon_ID
FIELD_ENGAGEMENT
Grunt_Personnel_ID
Pokemon_ID
Trainer_ID
Mission_ID
EXPERIMENTATION_EVENT
Scientist_Personnel_ID
Serum_ID
Pokemon_ID
Project_ID
ASSIGNED_TO
Personnel_ID
MIssion_ID

This stage involves translating all ER components—entities, subclasses, multi-valued attributes, and relationships—into initial tables.

Changes to Entity Types

- **Subclass Hierarchy (Personnel)**: A relation was created for the superclass **PERSONNEL** with attributes $Attrs(\textbf{PERSONNEL}) = \{Personnel_ID, ...\}$ and $PK = Personnel_ID$. A relation was created for each subclass, **GRUNT**, **BOSS**, **SCIENTIST**, with attributes {Personnel_ID} $\cup \{attributesofrespectivesubclass\}$ and $PK = Personnel_ID$.

- **Weak Entity Types**: Each weak entity was converted into its own relation (table), which included all of its own attributes. This new table also inherited the primary key from its "owner" entity, which serves as a foreign key. The primary key of the new weak entity table is a composite key, formed by combining the owner's primary key and the weak entity's own partial key. This process was applied to our three weak entities: **MISSION_ASSIGNMENT**, **ACQUISITION_ASSET**, and **EXPERIMENT_LOG**.

- **Multi-Valued Attributes**: Attributes like *Type* (**POKEMON**) were resolved by placing them in separate tables (**POKEMON_TYPE**), using the original entity's PK as a component of the new Composite PK.

- **Relationships**: 1:M relationships (e.g., *Base houses Personnel*) were resolved by placing the Foreign Key ($Base_ID$) on the 'Many' side (**PERSONNEL**). M:N, and N-ary relationships became dedicated tables (e.g., **OWNERSHIP**, **MISSION_ASSIGNMENT**).

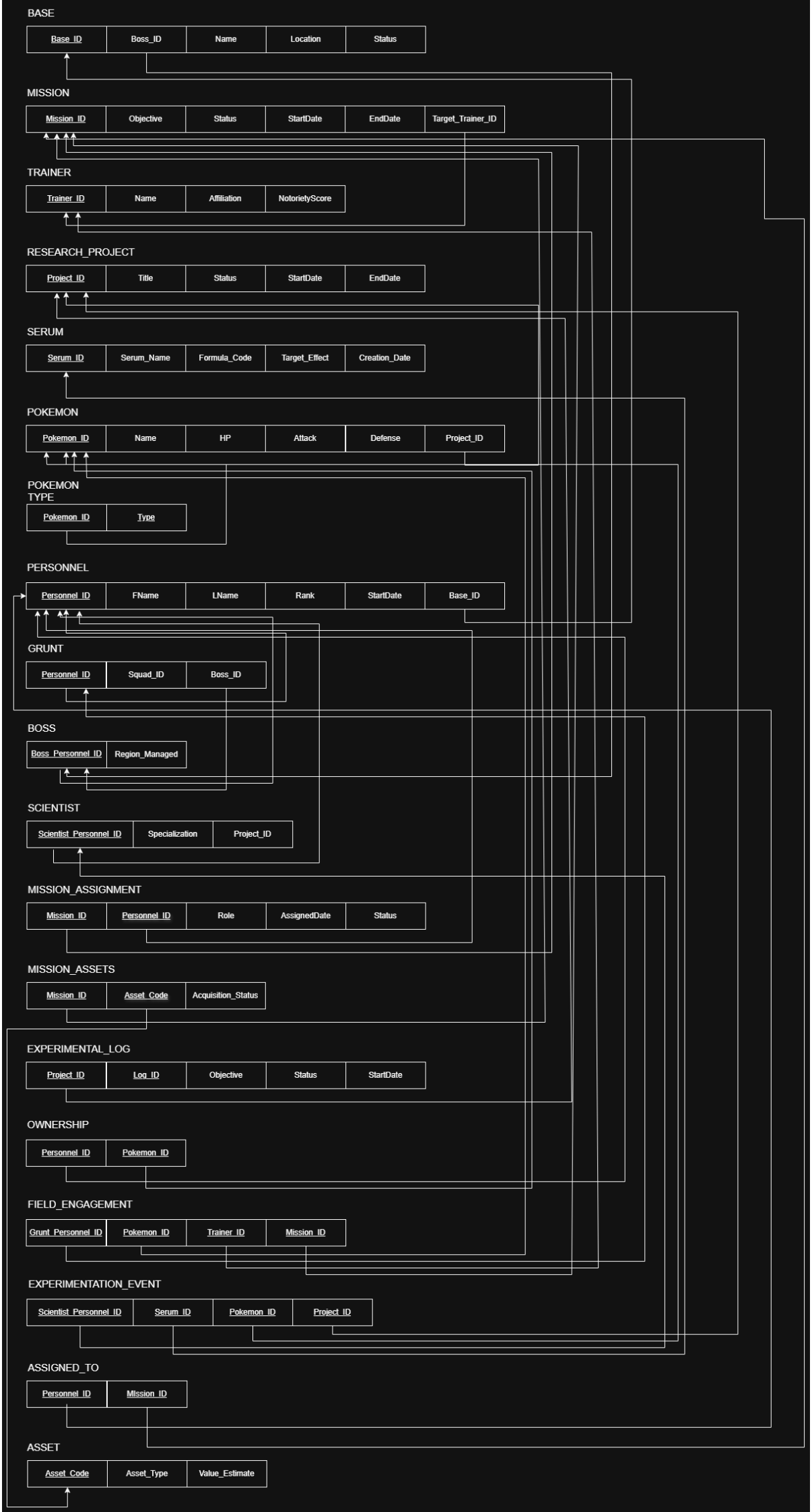
1.2. Relational Model After Conversion to 1NF

Explanation No further conversion was strictly necessary. The initial ER-to-Relational mapping process inherently satisfied First Normal Form (1NF) by ensuring all attributes were atomic. This was achieved by:

- Decomposing composite attributes (e.g., *Name* into $FName$, $LName$).

- Isolating multi-valued attributes into separate relations (e.g., **POKEMON_TYPE**).

1.3. Relational Model After Conversion to 2NF

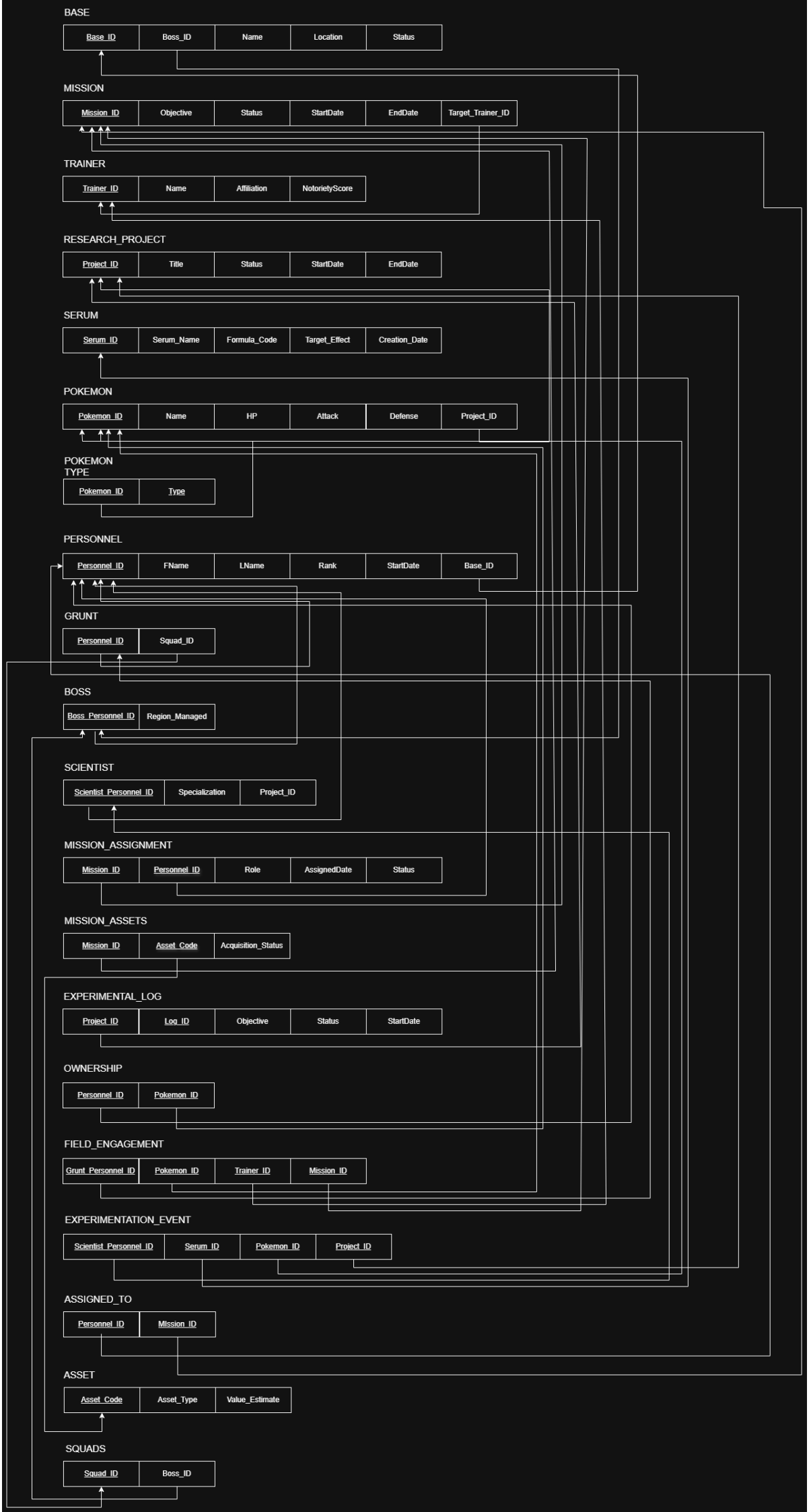

BASE
Base_ID
Boss_ID
Name
Location
Status
MISSION
Mission_ID
Objective
Status
StartDate
EndDate
Target_Trainer_ID
TRAINER
Trainer_ID
Name
Affiliation
NotorietyScore
RESEARCH_PROJECT
Project_ID
Title
Status
StartDate
EndDate
SERUM
Serum_ID
Serum_Name
Formula_Code
Target_Effect
Creation_Date
POKEMON
Pokemon_ID
Name
HP
Attack
Defense
Project_ID
POKEMON TYPE
Pokemon_ID
Type
PERSONNEL
Personnel_ID
FName
LName
Rank
StartDate
Base_ID
GRUNT
Personnel_ID
Squad_ID
Boss_ID
BOSS
Boss_Personnel_ID
Region_Managed
SCIENTIST
Scientist_Personnel_ID
Specialization
Project_ID
MISSION_ASSIGNMENT
Mission_ID
Personnel_ID
Role
AssignedDate
Status
MISSION_ASSETS
Mission_ID
Asset_Code
Acquisition_Status
EXPERIMENTAL_LOG
Project_ID
Log_ID
Objective
Status
StartDate
OWNERSHIP
Personnel_ID
Pokemon_ID
FIELD_ENGAGEMENT
Grunt_Personnel_ID
Pokemon_ID
Trainer_ID
Mission_ID
EXPERIMENTATION_EVENT
Scientist_Personnel_ID
Serum_ID
Pokemon_ID
Project_ID
ASSIGNED_TO
Personnel_ID
Mission_ID
ASSET
Asset_Code
Asset_Type
Value_Estimate

The conversion to 2NF addressed partial dependencies in tables that possessed a Composite Primary Key.

Changes to Entity Types

- **Acquisition Asset Decomposition**: The **ACQUISITION_ASSET** table contained attributes ($Asset_Type$, $Value_Estimate$) dependent only on the partial key ($Asset_Code$), violating 2NF.
- **Resolution**: The non-key attributes that exhibited partial dependency were moved to a new table, **ASSET** ($\underline{Asset_Code}$ PK). The original table was revised to **MISSION_ASSETS** ($\underline{Mission_ID}, \underline{Asset_Code}$ PK), serving purely as a link.

1.4. Relational Model After Conversion to 3NF


BASE
Base_ID
Boss_ID
Name
Location
Status
MISSION
Mission_ID
Objective
Status
StartDate
EndDate
Target_Trainer_ID
TRAINER
Trainer_ID
Name
Affiliation
NotorietyScore
RESEARCH_PROJECT
Project_ID
Title
Status
StartDate
EndDate
SERUM
Serum_ID
Serum_Name
Formula_Code
Target_Effect
Creation_Date
POKEMON
Pokemon_ID
Name
HP
Attack
Defense
Project_ID
POKEMON TYPE
Pokemon_ID
Type
PERSONNEL
Personnel_ID
FName
LName
Rank
StartDate
Base_ID
GRUNT
Personnel_ID
Squad_ID
BOSS
Boss_Personnel_ID
Region_Managed
SCIENTIST
Scientist_Personnel_ID
Specialization
Project_ID
MISSION_ASSIGNMENT
Mission_ID
Personnel_ID
Role
AssignedDate
Status
MISSION_ASSETS
Mission_ID
Asset_Code
Acquisition_Status
EXPERIMENTAL_LOG
Project_ID
Log_ID
Objective
Status
StartDate
OWNERSHIP
Personnel_ID
Pokemon_ID
FIELD_ENGAGEMENT
Grunt_Personnel_ID
Pokemon_ID
Trainer_ID
Mission_ID
EXPERIMENTATION_EVENT
Scientist_Personnel_ID
Serum_ID
Pokemon_ID
Project_ID
ASSIGNED_TO
Personnel_ID
Mission_ID
ASSET
Asset_Code
Asset_Type
Value_Estimate
SQUADS
Squad_ID
Boss_ID

The conversion to 3NF addressed transitive dependencies (where a non-key attribute determines another non-key attribute).

Changes to Entity Types

- **Grunt/Boss Transitivity**: The **GRUNT** table, structured as $\underline{Personnel_ID} \rightarrow Squad_ID \rightarrow Boss_ID$, exhibited a transitive dependency, as $Squad_ID$ (non-key) determined $Boss_ID$ (non-key). This violated 3NF.
- **Resolution**: This transitive dependency was isolated into a new table, **SQUAD** ($\underline{Squad_ID}$ PK, $Boss_ID$ FK). The **GRUNT** table was revised to only retain the necessary Foreign Key, $Squad_ID$, eliminating the redundancy of storing $Boss_ID$ for every Grunt.

Final State: All resulting relations are now verified to be in 3NF, minimizing data redundancy and ensuring high data integrity.

2. draw.io Links

The following links provide access to the visual representations of the diagram:

- **Relational Schema/1NF Diagram Link:** `https://drive.google.com/file/d/1iMK4yhB2IiWyKLTZDwbzgTupDcQ4aCoh/view?usp=sharing`
- **2NF Diagram Link:** `https://drive.google.com/file/d/1fi5Wk4r8GPbyvvXE7BwNJdDatb2Q`
  `view?usp=sharing`
- **3NF Diagram Link:** `https://drive.google.com/file/d/1paHy-TWDIZUatwGJFIZCahaT9O7G`
  `view?usp=sharing`