

Project Phase 3 Report

Team 66: pyramid

1. ER Diagram to Relational Model Mapping

Explanation

The relational model was derived from the Phase 1 ER diagram by applying the standard mapping procedures.

- **Strong Entities:** Each strong entity type (`Participant`, `Universe`, `InvestmentTier`, `MarketingCampaign`, `Portals`) was mapped to its own table. The attributes of the entity became the columns of the table, and the entity's primary key became the table's primary key.
- **Specialization (Subclasses):** The `Member` and `Employee` subclasses of `Participant` were mapped using a 1:1 relationship, where each subclass table's primary key (`participant_id`) is also a foreign key referencing the `Participant` superclass table.
- **Weak Entities:** The weak entity types (`RecruitmentEvent` and `PortalCalibration`) were mapped to tables. The primary key of each table is a composite key, formed by combining the primary key of the owning entity (`Member` and `Portals`, respectively) and the weak entity's partial key (`recruit_id` and `calibration_code`, respectively).
- **1:N Relationships:** Relationships like `BELONGS_TO` (Member-Universe), `MANAGES` (Employee-Member), and `PROCESSES` (Employee-Transaction) were mapped by including the primary key of the '1' side entity as a foreign key in the 'N' side table.
- **Multi-valued & Composite Attributes (1NF Handling):**
 - The composite `address` attribute in `Participant` was flattened into separate atomic columns: `city_id`, `country`, `planet`, and `universe_id`.
 - The multi-valued `members_involved` attribute in `MarketingCampaign` was resolved by creating a new linking table, `MarketingCampaign_members_involved`, with a composite primary key (`program_id`, `members_involved`).

Relational Model Snapshot

[Linked here](#)

2. Relational Model after Conversion to 1NF

Explanation

The mapping process from the ER diagram to the relational model was designed to directly produce a model in First Normal Form (1NF).

1. **Atomicity:** All attributes in the relational model are atomic and hold only single values.
2. **Multi-valued Attributes:** The primary 1NF violation from the ER diagram, the `members_involved` multi-valued attribute in the `MarketingCampaign` entity, was resolved during the mapping. We created a separate `MarketingCampaign_members_involved` table to represent this many-to-many relationship, with each row holding a single `program_id` and a single `members_involved` ID.
3. **Composite Attributes:** The `address` composite attribute from the `Participant` entity was flattened into its atomic components (`city_id`, `country`, `planet`, `universe_id`).

Since all attributes are atomic and there are no repeating groups, the model as mapped in the previous step is already in 1NF. No further changes are necessary.

1NF Snapshot

[As no changes were required, this snapshot is identical to [the one above](#). As per the assignment instructions, we can state that no conversion was necessary for this step.]

3. Relational Model after Conversion to 2NF

Explanation

To ensure Second Normal Form (2NF), the model was analyzed for any partial dependencies in tables with composite primary keys.

1. **PortalCalibration (PK: portal_id, calibration_code)**: The non-key attributes `engineer_id` and `calibration_timestamp` describe a specific calibration event. The `calibration_code` is a *partial key*, meaning it is only unique for a given `portal_id`. Therefore, both non-key attributes are fully functionally dependent on the *entire* composite key. This table is already in 2NF.
2. **MarketingCampaign_members_involved (PK: program_id, members_involved)**: This table has no non-key attributes, so it trivially satisfies 2NF.

These are the only two composite primary keys in our tables and none of them have any partial dependencies. Therefore, our 2NF is the same as our 1NF.

2NF Snapshot

[Linked here](#)

4. Relational Model after Conversion to 3NF

Explanation

The 2NF model is now analyzed for transitive dependencies, where a non-key attribute is dependent on another non-key attribute.

1. **Violation:** The `MarketingCampaign` table has transitive dependencies. `program_id` is the primary key and `program_id -> program_code` is a dependency. `program_code` to all other attributes (`universe_id, name, budget, start_date, end_date, status`) in the table are also dependencies and these form a transitive dependency. Therefore, we create two separate tables.
 - a. `MarketingCampaign` containing two attributes `program_id`, `program_code` with `program_id` as the primary key.
 - b. `MarketingCampaignAdditional` table containing `program_code`, `universe_id, name, budget, start_date, end_date, status` as the attributes with `program_code` as the primary key which is also a foreign key to `program_code` in `MarketingCampaign`.

All other tables remain the same as in 2NF.

3NF Snapshot

[Linked here](#)