SER 322 Project – Group 10

Jacob Gillow – Christopher Paquin – Brynn Ayers – Matthew Palmer

Star Wars Bounty Board Database

Application Domain

Star Wars–themed contract management and tracking for bounties placed on targets by criminal syndicates, for bounty hunters to accept.

Database Application Description

This database supports a bounty board service that is accessible from multiple planets. Bounty hunters can access the database to identify available contracts that syndicates have placed on targets.

Entities:

Character: This is a superclass and every character must be 1 or both of the available sub-classes which are targets and Bounty Hunters. Characters have IDs and names. Targets have values and a “last seen on” planet; Bounty Hunters have credits which increase when completing bounties.

Planet: Planets are locations for targets. They have IDs and names.

Contract: Contracts are created by syndicates to indicate they want a bounty hunter to get the target. Their attributes are ContractID, syndicateID, targetID, Bounty Hunter (can be null before one is assigned), and CompletionStatus (Boolean).

Syndicate: A Syndicate is a criminal organization that puts hits out on targets. It has an ID and name.

Relationship Requirements

1. Syndicate–Contract
    - Each contract must refer to exactly one Syndicate (Contract.SyndicateID is not null).
    - A Syndicate can have many contracts.
    - A Syndicate can insert, update, or delete bounties at any time.
2. Target–Contract
    - Each contract must reference exactly one target (Contract.TargetID is not null).
    - A target can be in multiple contracts.
    - If a contract completes then all active contracts with that targetID are deleted.
3. BountyHunter–Contract
    - Each contract has at most one assigned bounty hunter at any time:
    - Contract.BountyHunterID is either null (unassigned) or references exactly one BountyHunter.
    - A Bounty Hunter may be assigned to many contracts simultaneously.
4. Planet usage
    - Target.LastLocationPlanetID represents the target's current last known location and may change over time.

Referential Integrity Constraints

5. BountyHunter.ID must exist in Character.ID.
6. Target.ID must exist in Character.ID.
7. Target.LastLocationPlanetID must exist in Planet.ID.
8. Contract.SyndicateID must exist in Syndicate.ID.
9. Contract.TargetID must exist in Target.ID.
10. Contract.BountyHunterID, if not null, must exist in BountyHunter.ID

Additional Rules

1. Planet, Syndicate, Character, BountyHunter, and Target rows must not be deleted through the application.
2. If a Character is both a BountyHunter and a Target, and a contract completes where that Character is the Target, then that Character must be removed as the assigned hunter from all currently active contracts.

3. Every Character must appear in at least one of the subtype tables (Target and/or BountyHunter).
4. On contract completion the target’s value gets added to the bounty hunter’s credits.
5. A character can’t be both the hunter and target of the same contract.