During the requirements gathering meeting, Mark Dogfurry, the lead game architect for the *Loot!* pilot, explains what they want. *Loot!* is a *quest*-based game. But it will be so much better than previous quest games. It will be VR, for one thing!

People who sign up for *Loot!* — call them players — will have *accounts*. For each player's account, we need to keep track of the player's *name*, *address*, the date that they *joined*, a *cc#* (a valid credit card number we keep on file), and a *balance* (how much money in Canadian dollars they presently have in their game account). Each player will also have a unique *login*, which is the account's "name". It is assumed that a given player will have just one account.

A player may create over time any number of *avatars*. An avatar is an in-game persona. Thus, an avatar *belongs to* a player (an account). An avatar has a *name*, *gender*, *look*, and skill *level*. There is a small set of *looks* available, which may be extended over time; e.g., "wizard", "faerie", "footballer", "shark", and "raccoon".

The game's virtual word is divided up into a hundred or so *realms*. On any given *day*, a player may log in (*visit*) as one of his or her avatars into one of the realms. (The player may not then switch avatars or realms for that day.) Of course, some days, a player might not log in at all.

Realms will be intricate, amazing virtual worlds unto themselves! But, for the purposes of this database, we just need to know the realm's unique name (*realm*).

The *Questeme* adventure-scripting staff will be in charge of creating daily *quests*. We will try to ensure that each realm on each day has several *quests* available. Thus, a given quest is associated with a day and a realm. Quests will be intricate, amazing challenges! But, for the purposes of this database, we just need to know the quest's name (*quest*). The quest's name is guaranteed to be unique on its day in its realm. (That is, there will not be two quests named the same thing on a given day within a given realm). And, for a quest, we need to keep the time it was completed, if it was completed. (This is explained more later.)

more later.)
A player when he or she logs into a realm — as one of his or her avatars — on a given day may just hang out. (Believe me, our realms will be amazing! The most amazing realms you've ever hung out in!) Or their avatar can join one of realm's daily quests. We call this *acting* in a quest. When an avatar (the player as the avatar) joins a quest, they choose a *role* to play.

Associated with a quest is *loot*. Each piece of loot is a type of *treasure*. Each treasure type has a unique *name*, and a description (*desc*). A quest can have any number of pieces of loot. And, of course, each treasure type can be loot for any number of quests.

If the quest is successfully completed by its "team" — that is, by the avatars signed up for (*acting in*) the quest — then the quest's loot is given to those players. (The most famous treasure in the game is a golden lute. Lute. Loot. Get it? Snickers.)

Each piece of loot from the quest will be *given* to just one of the players who participated in the quest (as decided by the game engine). We need to record who has been given what. If a quest was successfully completed, we should record the *finish* time. (A quest is just for a given day, so we already know the day.)

And that is pretty much it for now! We would like the schema design to capture tightly the game logic as described above. We want the database to be able to protect against incorrect data updates to protect the integrity of the game.