Requirements for Data Model

Your database that you model shall consist of 4 related database tables (as specified below).

**IMPORTANT:** Do not use any SQL KEYWORDS as table names or field names. Google "SQL keywords" to see what to avoid, but you definitely cannot use these: user, role, password, grant, date... You will get lab deductions, but more importantly, you will have problems as you attempt to implement your web application.

1. **A user role table.** Your user role table shall have at least these two fields:
   - role id (PK, not auto-increment, integer or small varChar storing values like "ADMIN", "VIEW", etc.)
   - role name (must be unique - add database constraint).

2. Some kind of **user table** (named appropriately, depending on who your users are, like customer, or traveler). This table shall include these fields:
   - auto-increment primary key (to uniquely identify a particular user record),
   - logon name (typically it is an email address, must be unique – add database constraint),
   - logon password,
   - screen name,
   - foreign key that references the user role table,
   - **At least one more field** (you choose) that is a null-able non-character field (e.g., date, decimal, or integer). Null-able means that it is OK for the user to not put something into that field - it is optional for the user.
     - Decimal is a good choice for money type fields.
     - Pick Date over DateTime or else you will get an unintelligible real number that stores milliseconds form the beginning of time.

3. An **"other" table** named according to what you will store in it (can't be named "other") and including these fields:
   - auto-increment primary key (to uniquely identify a particular "other" record),
   - descriptive character field (must be unique – add database constraint),
   - **At least two more fields** (you choose), one of which must be a null-able non-character field (e.g., date, decimal, or integer). Null-able means that it is OK for the user to not put something into that field (optional for the user).

4. An **associative table** that implements a "many to many" relationship between your user table and your "other" table. This table shall be named according to what you will store in it (can't be named "associative"). If you think of your user table as the subject of a sentence and the "other" table as the object of the sentence, then your associative table describes the verb within the sentence. Attributes might be something like "number of items purchased", "when purchased", "discount amount", etc.
   - auto-increment primary key (to uniquely identify a particular associative record),
   - foreign key that references the user table,
   - foreign key that references your "other" table,
   - **At least two more fields** (you choose), one of which is a null-able non-character field (e.g., date, decimal, or integer). Null-able means that it is OK for the user to not put something into that field (optional for the user).

5. **Naming conventions:**
   - Every table shall have a PK that is named: tableName_id
   - All FKs shall be named tableName_id (referring to the tableName that they are pointing to)

6. **Somewhere in your user table and/or your "other" table** (didn't say associative table) shall be the following:
   - at least one optional Date type field, (e.g., you didn't click “Not Null” in the Table Design Screen).
   - at least one optional Decimal (usually a money amount, don’t click “Not Null”) or an optional integer.
   - at least one LongText.