

Names: Michelle Nguyen & Preston Sellers
Project Title: Cloud Books

Feedback by the peer reviewer

Review 1

Does the overview describe what problem is to be solved by a website with DB back end?

Yes, needing a DB to manage users, books, orders and user preferences...

One thing that I'd suggest is... often times on many websites... they save searches for an user, usually name them those "previous searches"... preference for me sounds like a search filter that a user can set when they are looking for something to start out with. Also if you are generating user's preference based on what that person buys instead of letting the user to dictate what he/she wants to see, I feel like it could get very complex.

Does the overview list specific facts?

Yes, 3 million revenue and 300k books sales, tracking four things.

Are at least four entities described and does each one represent a single idea to be stored as a list?

Yes, books, orders, users and preferences.

Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

Yes they do, each entity covers those 4 things.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

Yes, I think ERD looks good.

Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

Not perfect yet... for example...

Books_bookID vs Orders_Users_UserID

there's inconsistency there, with bookID with lower case... also the Orders_Users_UserID might

be too long and confusing...

bookID >>> BookID???

Review 2

Hello Michelle and Preston,

Great start to your project. I can see that your overview describes the issue of tracking and managing the company online book-selling operations. I found your overview clear, but was wondering if this company sells e-books, hard copies, or both. I think it would be nice to see that designated in the overview.

I saw that you listed four entities and it makes sense for the company model you have set up. Regarding the outline of entity details, I was wondering if users would have a security credential accompanying their userID. Another idea I have ties a bit into my question about whether your company sells e-books, hard copies, or both. If a hard copy is being sold and it's shipped to the user, it would be good to have their first and last name (or other form of identification outside of their user ID and email).

I think that your relationship diagram was clear and logical to the reader, and the 1:M relationships were correctly formulated. I saw there was 1 M:M relationship. My last point is just a suggestion for consistent formatting. Maybe consider snake casing User Preferences in your relationship diagram to match the other entities. Great work!

Review 3

Hello,

Nice inclusion of quantitative data, however I would also include the number of users, as some of your project focuses on data involving users. Other than that, the overview clearly describes what problem is trying to be solved.

There are four entities present which is good. I like how you clearly illustrate how and why each relationship is the way that it's set, as it makes it very clear to understand your thinking. I can clearly understand the 1:M relationships, and there is clearly a M:M relationship present.

For the ERD diagram, some of the naming, in particular Users_User seemed a little unclear, although those names are auto populated from what I remember. Either way if your team can make the naming a little more clear that would be good. Other than though the naming and format of the ERD diagram look good.

Other than that, you have the required amount of entities and relationships, as far as I can tell, and are well on your way to a great project. Nice job!

Review 4

- **Does the overview describe what problem is to be solved by a website with DB back end?**

The overview does a good job of summarizing the business model and scale of the company, as well as its operational aspects like bookkeeping of books, orders, users, and user preferences. However, it doesn't explicitly mention the problem to be solved by the website with a DB back end. A statement addressing the need for this database would be beneficial.

- **Does the overview list specific facts?**

Yes, the overview lists several specific facts, including the company's annual earnings, the volume of books sold, and the purpose of the database-driven website.

- **Are at least four entities described and does each one represent a single idea to be stored as a list?**

Yes, there are four entities mentioned: Books, Orders, Users, and User Preferences. Each represents a unique idea and would correspond to its separate database table.

- **Does the outline of entity details describe the purpose of each, list attribute data types and constraints and describe relationships between entities?**

Yes, each entity is described with its attributes, data types, constraints, and relationships to other entities. The use of the primary key and foreign key are appropriate, as are the defined relationships among the entities.

- **Are 1:M relationships correctly formulated? Is there at least one M:M relationship?**
Does the ERD present a logical view of the database?

The 1:M relationships between Users and Orders and between Users and User Preferences seem to be correct. The M:M relationship between Books and Orders is correctly identified.

- **Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**

The naming is mostly consistent, but there is room for some change/improvement. For example, in your database outline, all the entities are singular, but the conventional practice is to use plural names for database tables (like 'Books' instead of 'Book'). Also, namings such as Orders_orderID and Users_userID can seem a bit confusing.

Overall, great job!

Actions based on the feedback

- We made the naming in our diagram match with the naming in our outline to create consistency.
- We also decided to make it more clear that we are only selling e-books.
- We added the number of customers to our overview.

Upgrades to the Draft version

- We decided to change our user preferences entity to a wishlist entity. Then that wasn't working so we changed it to a reviews entity. The review entity will store users reviews of specific books.
- We realized that two of our relationships were really two sides of one relationship, so we added a new 1:M relationship between users and reviews so we would have a total of 4 relationships.

a) Overview

Cloud Books is a startup company that wants to sell ebooks. They need a database backend to keep track of their *books*, *orders*, *users*, and *user book reviews*. Without the database, they will not be able to function properly or make informed business

intelligence decisions. Cloud Books hopes to make approximately \$3 million dollars per year from selling 300,000 e-books online to 60,000 customers.

b) Database Outline

- Books: records information about an individual book's availability
 - bookID: int, auto_increment, unique, not NULL, PK
 - title: varchar, not NULL
 - author: text, not NULL
 - genre: text, not NULL
 - price: double, not NULL
 - status: varchar, not NULL
 - Relationship: Many-to-Many relationship between Orders and Books. An Order can have one or many Books and a Book can be a part of one or many Orders. Both FK of each entity will be in the intersection table.
- Orders: records order details.
 - orderID: int, auto_increment, unique, not NULL, PK
 - fkUserID: int, not NULL, FK
 - bookID: int, not NULL, FK
 - paymentMethod: varchar, not NULL
 - address: varchar, not NULL
 - quantity: int, not NULL
 - orderDate: datetime, no NULL
 - orderStatus: varchar, not NULL
 - totalDue: double, not NULL
 - Relationship: Zero-to-Many relationship between Users and Orders. A User can have zero or many Orders. Users FK will be in the Orders entity.
- Users: records user information
 - userID: int, auto_increment, unique, not NULL, PK
 - email: varchar, not NULL
 - paymentMethod: varchar, not NULL
 - address: varchar, not NULL
 - Relationship: One-to-Many relationship between Users and Reviews. Users can have many Reviews. User FK in the Reviews entity.
- Reviews: records information about a books rating by users
 - reviewID: int, auto_increment, unique, not NULL, PK
 - userID: int, not NULL, FK
 - fkBookID: int, not NULL, FK
 - bookID: int, not NULL
 - rating: int, not NULL
 - description: text, not NULL

- Relationship: One-to-Many relationship between Books and Reviews. A Book can have many Reviews. Book FK in the Review entity.

c) Entity - Relationship Diagram

