**Landscape Analysis 1: NoSQL versus Relational Databases**

For each of the following use cases, make an argument as to the best database choice for the expressed need—relational or NoSQL. If NoSQL, specify a subtype from those we’ve discussed in class: column store, graph database, document store, or key-value store. You will need to do some digging on the web in order to gather supporting information for your recommendations.

There is no “right” answer—rather, the important part is to justify your choice. Imagine you are a vendor or consultant making a recommendation to a client. You may even recommend a combination of different solutions to support different functions of the applications described.

Each use case response should be a maximum of one page. Data architecture diagrams are welcomed but not required. You should use at least three sources of information in making your recommendations. Cite those sources in your write-up.

**Use Case 1: Book recommendation website**

|  |  |
| --- | --- |
| **Description** | Our website will allow users to rate books they’ve read on a 5-star scale, and then recommend other books based on their ratings, genre preferences, and other factors. (Imagine Netflix, but for books.) In particular, we want to be able to base recommendations on the books that similar users liked. |
| **Types of data** | Book metadata (e.g., title, author, publisher, genre), user book ratings, user libraries (i.e., which books a given user has read), user data (e.g., username, demographics, dates and lengths of site visits) |
| **Analytical needs** | We will need to frequently adjust our recommendation algorithm, so we need to be able to query how well our recommendations match up with the star rating that users ultimately give. For example, if we highly recommend a book to a user, but they rate that book only 1 star, we need that information. We also need to be able to determine which users are “similar” based on their book choices and star ratings. We would also like to be able to produce “Top Book Lists,” such as Most Popular Book Today or Highest Rated Books this Year. |

**Use Case 2: Money exchange app (like Venmo or CashApp)**

|  |  |
| --- | --- |
| **Description** | Our mobile app will allow users to pay each other instantaneously. We will link with users’ bank accounts and deduct money as needed. Users can also hold a balance within our app, or can “cash out” their balance by transferring money back to their connected bank account. Money exchanged between users will be instantly debited from the payor’s balance and instantly added to the payee’s balance in app, though it may take a couple of business days to transfer money to and from external bank accounts. |
| **Types of data** | Account balances, records of debit and credit transactions, linked bank account information, user data, user “friend” data (i.e., users who pay each other) |
| **Analytical needs** | Users need to be able to query their balances and transaction histories. Internally, we need to be able to query amounts of money transferred between users over various time periods (days, weeks, months, quarters, etc.), which banks our customers use the most (and how much money we are passing back and forth between those banks). We also need to capture extremely specific and detailed information about each user session for regulatory purposes. |

**Grading Rubric**

|  |  |
| --- | --- |
| **%** | **Item** |
| **40** | Recommendations are well-justified with evidence gathered from outside sources. |
| **20** | Each use case cites at least 3 sources. |
| **30** | All stated requirements are addressed in the response. |
| **10** | Writing is clear and professional. |