1. What technologies would you use?
2. Database Replication – have multiple live instances of the database with real-time replication based on customer location. Provide multiple instances in locations where there is heavy traffic.
3. Database Mirroring – at any single point, have a byte to byte copy of the database in production as a failsafe measure
4. Database Multithreading – Have a thread pool that manages a growing user and database read/write commands
5. Optimize Queries – ensure that the queries running on the database are optimized for performance
6. Optimize Storage – ensure to employ fast storage for faster read/write to the database
7. Database Monitoring – This allows for the proactive response to issues as it allows for real-time visibility into availability.
8. Why would you choose these technologies?
9. Database Replication – have multiple live instances of the database with real-time replication based on customer location. Provide multiple instances in locations where there is heavy traffic.
10. Database Mirroring – at any single point, have a byte to byte copy of the database in production as a failsafe measure
11. Database Multithreading – Have a thread pool that manages a growing user and database read/write commands
12. Optimize Queries – ensure that the queries running on the database are optimized for performance
13. Optimize Storage – ensure to employ fast storage for faster read/write to the database
14. Database Monitoring – This allows for the proactive response to issues as it allows for real-time visibility into availability.
15. What patterns would you use?
16. Identity Map – The identity map pattern is a database access design pattern used to improve performance by providing a context-specific, in-memory cache to prevent duplicate retrieval of the same object data from the database
17. Unit of Work – Automate the process by which objects are saved to the database, ensuring that only objects that have been changed are updated, and only those that have been newly created are inserted.
18. Domain Object Factory – Populate domain objects based on query results. The advantages include:-
    1. Domain object mapping encapsulation

A domain object factory encapsulates the translation of query result data to domain objects. This decouples application code from the data model and data access components.

* 1. Isolation of database entity and domain object details

Translation details vary for different database entities and domain objects. You can define a different domain object factory implementation for each translation, keeping them isolated from each other.

* 1. Consistent Client Interface

Domain object factory (and indeed any other factory) define a consistent interface that applications and other middleware code can use without requiring references to specific implementations.

1. Lazy Loading - Defer object creation, and even database queries, until they are actually needed