



Read Paper review (The Design of the Force.com Multitenant Internet Application Development Platform).

Answer the following questions:

1. What is metadata driven architecture?

Answer: Metadata-driven architecture is a design strategy that uses metadata, or data that describes the application, to dynamically generate application components during runtime. This approach is particularly useful in multitenant applications, where different tenants have varying expectations and requirements. The metadata-driven architecture involves separating the compiled runtime engine, application data, and metadata describing the application's basic functions from the metadata for each tenant's data and customizations. This separation makes it possible to update the system kernel, modify the core application, or customize tenant-specific components independently without affecting the others. In summary, metadata-driven architecture allows for dynamic application generation and independent updates or customization of different components in a multitenant environment.

2. What is data definition and storage model?

Answer: The data definition and storage model used by Force.com manages virtual database structures using a combination of metadata, data, and pivot tables. Instead of managing actual database structures for each application and tenant, metadata is used to keep track of the custom application objects, their fields, relationships, and other object definition characteristics. Structured and unstructured data for all virtual tables are stored in a few large database tables, while specialized pivot tables maintain denormalized data to make the combined data set highly functional.

3. Explain the Data Table and its role.

Answer: The Data Table in Force.com's storage model stores application-accessible data for all custom objects and their fields using metadata defined in the Objects and Fields tables. Each row in the Data table includes a global unique identifier (GUID), an organization identifier (OrgID), and an object identifier (ObjID), along with a "natural name" field for object instances. Custom fields can use various structured datatypes, and the platform enforces required fields and custom validation rules. Force.com manages object and field definitions as metadata rather than actual database structures, allowing it to tolerate multitenant application schema maintenance activities without blocking other users. Flex columns in the Data table use a universal datatype, allowing them to manage the information of multiple fields that use different structured datatypes. The platform stores all flex column data in a canonical format and uses datatype-conversion functions as necessary.

4. Explain the concept of the Pivot Table, its role and its structure.

Answer: Pivot tables are used in Force.com to store specific types of data that cannot be efficiently stored in the primary table due to their size or complexity.

Role: Pivot tables store data related to specific features or functionalities of the Force.com platform, such as large text fields, indexes for efficient data retrieval, enforcing uniqueness of field values, and managing relationships between objects.

Structure: Pivot tables consist of a set of columns that correspond to specific data types or fields, along with one or more unique indexes that allow for efficient data retrieval or enforcement of data constraints. The specific structure of each pivot table is designed to support the specific functionality it is responsible for.

5. What is Apex and what it is used for?

Answer: Apex is a programming language used by developers to add custom business logic to Force.com applications. It is strongly typed, object-oriented, and similar to Java. Apex enforces governors and resource limits to ensure scalability and performance of the platform. Before deploying new applications, strict management and unit tests are required to validate functionality. After deployment, the built-in performance profiler provides feedback for administrators to analyze application functionality.

6. How is internal query optimization performed?

Answer: internal query optimization in Force.com:

1. When a request for data is received, Force.com intercepts the request and considers multitenant-aware statistics to determine optimal query execution plans.
 2. The platform executes pre-queries based on tenant-, group-, and user-level statistics, as well as other types of statistics such as custom index and picklist field histograms.
 3. Based on the results of the pre-queries, the platform builds an optimal database query for execution in the specific setting.
 4. Force.com's optimizer relies on internal security-related tables such as Groups, Members, GroupBlowout, and Customs hare to maintain information about the security domains of platform users.
 5. The optimizer employs various strategies to help build reasonably optimal queries, such as using the FallbackIndex pivot table to efficiently find requested object instances or dynamically generating missing statistics at runtime.
 6. Force.com can execute the same query in different ways depending on who submits the query and the selectivity of the query's filter conditions, such as using nested loops join, index-related filters, or ordered hash join.
7. What is full text search engine?

Answer: tool that allows users to search for words or phrases within large amounts of text data, such as documents or records. It works by creating indexes of the text data and using algorithms to rank the relevance of search results. Force.com uses an external search engine to provide this functionality for its web-based applications, optimizing the indexing process and ranking of search results based on user access and update frequency

ON THE OTHER HAND, Tool used to provide an interactive search capability for web-based applications. It allows users to search an application's data and return up-to-date ranked results in sub-second response times