International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Optimizing Salesforce Data Model Design for ISV Partners

Praveen Kotholliparambil Haridasan

Abstract
Salesforce provides a powerful platform to Independent Software Vendors (ISVs) for building and deployment applications, but the design of the data model poses challenges considering the multitenant architecture. A comprehensive review of Salesforce design model for ISV partners is delineated in this article which focuses on key considerations like performance optimization, data normalization, object relationships, and security practices. The article also explores the differences between the usage of standard and custom objects. It also addresses the importance of indexing, and other strategies for maintaining data integrity. Field encryption, record-level security, and package compliance are also addressed in this article. Consideration of these principles by the ISV partners will enable them to build high-performance and scalable applications to meet the diverse needs of the customers while staying within Salesforce's platform limitations.

Keywords: Salesforce, ISV, Data Model, AppExchange, Object Relationships, Data Architecture, Custom Objects, Performance Optimization, Security, Governor Limits

1. Introduction

In today's era of cloud-driven enterprise solutions, Salesforce occupies a prominent position, enabling independent software vendors (ISVs) to create, deliver, and oversee applications for the global market. Whereas the design of an efficient and scalable data model requires careful consideration of the unique multitenant architecture of Salesforce, data security requirements, and the governor limits.

This article also provides a technical overview of the strategies of Salesforce data model design, that emphasizes the ways in which ISV partners can balance performance, security, and scalability for both custom and standard objects. Moreover, the complex object relationships, data indexing, and normalization are also discussed. By considering these practices the ISV partners can build efficient applications to meet the diverse needs of the customers while staying within Salesforce's platform limitations [1].

2. Salesforce Data Architecture Overview

The concepts of entities, domains, and connections form the foundation of Salesforce's database design. Two main object kinds are available in Salesforce:

Standard Objects: Salesforce provides preset along with ready-to-use options. Account, Contact, and Opportunity are included.

Custom Objects: customers and developers, developed it to hold information unique to a given program.

A main decision of ISV partners is to determine the appropriate usage of custom and standard objects [1]. The standard objects enhance the interoperability of the salesforce products with third party

applications, and the custom objects provide the necessary flexibility for storing data [2]. The careful design of custom objects is highly recommended to avoid complexity that may affect the performance.

3. Multi-tenancy Considerations

Salesforce runs on the multitenant model in which, many different customers use the same physical server and hardware infrastructure [6]. This architecture affects data model design because ISV applications must run smoothly in various customers' environments and not stall systems. The challenge then lies in the fact that while ISVs need to ensure that their data models are very efficient, they have to ensure that these meet the specific requirements of its customers.

4. Key Considerations for Data Model Design

4.1. Object Relationships

Salesforce supports three types of relationships between objects:

- **Lookup Relationships**: They bear loose coupling where one object can access another object.
- **Master-Detail Relationships**: These are the examples of tightly coupled relationships where the child object is always dependent on the parent object. This means that if the parent is deleted, so also will be the child record.
- **Hierarchical Relationships**: They are normally employed in an organization as a means of defining the relationship between users.

Often the choice of relationships becomes crucial when designing an ISV data model to identify the way the objects will interconnect [3]. Master detail reports are more rigid and are used when there are strong parent –child relationships as is the case with invoicing or taking of inventories. On the other hand, it is possible to choose lookup relationships which are more flexible and loosely coupled with the data as compared to using cascading deletes or permissions for objects.

For instance, in applying the human resources context, the use of the lookup relationship between Employee and Department objects where an employee is described as being employed by a particular department is much more flexible in case of changes in the employees' departments while a master-detail relationship will be best suited between the Salary Record and Employee objects to ensure that salary transactions are exactly matched to the right employee.

4.2. Data Normalization vs. Denormalization

Normalization involves the structuring of data for reduction of redundancy and the enhancement of data integrity for efficient management of systems. Whereas, in Salesforce environment denormalization is often required to enhance the performance of the system. In denormalization, redundant data is added to reduce the joins in the database tables.

For example, let us assume that an ISV has built a customer relationship management (CRM) application in which users execute reports on various objects, namely, Account, Contact and Opportunity, among others. In this case, denormalization or the direct insertion of some fields, extracted from Contact object, in the Account object may help to reduce the number of joins, which may be extremely helpful while creating reports and dashboards.

The use of denormalization increases performance, but it has disadvantages, for example, to maintain the integrity of each table in the database, you may end up with having to maintain the integrity of each redundant field. The considerations taken by the various ISV partners have to find a balance based on the expected access to the data and the query performance.

4.3. Indexed Fields

The keys like Record ID, foreign keys (relationship fields), audit fields (CreatedDate, LastModifiedDate), and fields marked as External ID or Unique are automatically indexed by the salesforce. Whereas the custom indexes are to be requested from the salesforce support for optimising the performance on specific queries. Indexing is particularly important when large datasets are used and can improve the efficiency of SOQL queries by reducing the amount of scanned data [5].

Over-indexing may lead to degradation of performance when data write operations are done. For example indexing a large number of fields in a high transaction object (e.g transaction in financial application) may largely slow down the bulk data load. Hence, it is important for ISVs to consider the need of each index to be created so as to avoid having most of them in the database.

5. Security Considerations

Salesforce offers several layers of integrity: Object-level, Field-level, and Record-level [4].

5.1. Record-Level Security

ISVs have to ensure data integrity through every layer and should adhere to Salesforce's integrity framework. Like in any other typical project solution, Organization-Wide Defaults (OWD), Sharing Rules, and Role Hierarchies are key success factors for record access control within an ISV solution [9].

For example, in a human resources application that an ISV has created, the OWD may have been customized to be private, thus only the HR managers are allowed to view wage information. Sharing Rules can then be used to give particular users rights to particular records.

5.2. Field-Level Encryption

Salesforce comes with Shield Platform Encryption to enable organizations to protect sensitive data by Shield Platform Encryption encrypting fields. ISV solutions dealing with personal information including medical and financial records should employ encryption to help adhere to GDPR and HIPAA legal requirements [9].

For instance, a firm developing a healthcare management software must ensure the fields, which contain patient information are safe even if the hacker manages to penetrate through the system.

6. Salesforce Data Storage Calculations

Salesforce has unique limitations on data storage for each company, therefore it's critical for ISV partners to understand how storage capacity is calculated. This knowledge is particularly crucial when creating applications that may handle huge amounts of data. By improving the architecture of the stored information structure ISVs can assist clients in avoiding exceeding the storage constraints needlessly, otherwise this may result in reduced efficiency or higher storage costs.

6.1 How Salesforce Calculates Data Storage

The amount of entries stored in both standard and custom objects determines how much memory for data is needed. how much storage room every record requires.it is determines by the category of objects [9].

- **Standard Objects**: Every entry in typical entities like:Account, Contact, Opportunity, and Case utilize 2 KB of data capacity.
- **Custom Objects**: Every custom objects entry takes up 2 KB of data capacity. But the size may rise depending on the quantity and kind of objects records (such as rich text fields or long text areas) [8].

- **Big Objects**: Large-scale objects, crafted to manage vast amounts of data, are excluded from typical storage limits and have their own distinct storage capacity and restrictions.

For example, if an ISV application holds 100,000 Account records, they would use up about 200 MB of data storage, with each record taking up 2 KB. Similarly, same calculation methodology would apply to custom objects in an ISV's management store where 50,000 records of a custom object would take up around 100 MB of storage.

6.2. Large Data Volumes and Data Archiving

Data storage management is important in ISV applications that deal with enormous amounts of data especially in the finance or health sector since rules demand storage for an extended period [7]. To address these storage needs, then ISVs should apply data archiving techniques. This is made possible by salesforce, which migrates old, less accessed data to cloud solutions using third-party apps or custom developed solutions. Big Objects from Salesforce can also be used for ISV applications that store a significant amount of data which is not very frequently used such as logs or transaction records.

6.3. Data Storage Best Practices

During the creation of data scenarios, the ISV partners should follow the suggested procedure to utilize the salesforce data capacity [7].

Data Minimization: If possible the data must be normalized and unnecessary entities must be deleted.

Leverage External Storage: The external object feature must be used in the salesforce for linking with external data and preserving own storage capacity.

Data Retention Policies: Several automatic processes must be set up to remove unnecessary documents. Apex batch jobs or scheduled flows can be deployed to remove older data.

Big Objects: The ISV application with high data volume should use big objects to offload unnecessary data from the standard storage.

6.4. Impact of Data Storage on ISV Applications

Salesforce data storage limits is essential to be considered by the ISVs while designing solutions that are to be used by the customers with different capacities. If data model of an application will require excessive storage, additional costs may be incurred by the customer or the customer may face performance issues. Therefore, the data requirements should be considered carefully.

For example, an ISV developing a sales forecasting application may store forecast records across custom objects. To efficiently utilize the storage the ISV should limit the number of fields per record.

7. Best Practices for Data Model Design

Several best practices to be considered while developing the data model [10].

1. **Leverage Standard Objects:** It is always preferable to adhere to the standard Salesforce objects as to enable compatibility with other default Salesforce functions and use with third-party products.
2. **Performance Optimization**: It is necessary to avoid unnecessary joins between the objects to enhance the performance.
3. **Implement Security Early:** The salesforce security parameters must be followed strictly from the beginning till the end to address security concerns.
4. **Plan for Scalability**: It is vital to pay attention to the relationships while designing and then selectively denormalize where the need arises due to the expansion of data.
5. **Document the Model:** The information model should be documented well and should have definitions of the fields, and how the objects are related also the security setting. Such a degree of

specification ensures protection and reliability and helps towards the enhancement of understanding and management of the framework.

8. Conclusion

Careful planning is required for the efficient design of ISV applications. The utilization of Salesforce's standard objects and the creation of well-structured custom objects can enable the ISV partners to optimize the flexibility and interoperability. Data normalization, object relationships, and indexing can directly influence the performance and overall data management. Moreover, adherence to the best security practices like field and record level security, and encryption are crucial for delivering compliant and robust applications.

When ISV application grows in size, several strategies like archiving, SOQL optimization, bulk data operations, must be adopted. By considering these principles, the ISV partners can not only create solutions that can meet customer needs but the solutions will also align with the performance constraints of Salesforce.

References

1. Salesforce Documentation, "Data Modeling in Salesforce," [Online]. Available: https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/data_model_intro.htm.
2. C. Webber, "Salesforce Data Architecture and Management," *Salesforce Developer Documentation*, 2020.
3. J. Thompson, "Salesforce Data Modeling for Large-Scale ISV Applications," in *Proc. Salesforce Developer Conf.*, 2019.
4. L. Robinson, "Best Practices for Data Security in Managed Packages," *Salesforce AppExchange Partner Blog*, 2018.
5. G. Johnson, "Mastering Salesforce SOQL for Performance Optimization," *J. Cloud Comput.*, vol. 5, no. 2, pp. 102-110, 2017.
6. A. Richardson, "Salesforce Multitenant Architecture and Performance Considerations," *Salesforce Architects Guide*, 2019.
7. N. Patel, "Handling Large Data Volumes with Salesforce Platform," *Salesforce Technical Library*, 2016.
8. R. Davis, "Designing Effective Custom Objects and Relationships in Salesforce," *Cloud Appl. Design J.*, vol. 8, no. 1, pp. 34-45, 2017.
9. K. Harris, "Salesforce Security Model for ISVs: Implementing Field-Level and Object-Level Security," in *Proc. Salesforce Security Conf.*, 2018.
10. Sonar Software, "Learn About Salesforce Objects - FAQs, Best Practices & More," [Online]. Available: https://sonarsoftware.com/knowledge/salesforce/salesforce-objects/.