



DATABASE MULTITENANCY FEATURE ENABLE MECHANISM

Vidyasagar yadav¹ & Abhay Kumar Agarwal²

Abstract – As being started of Mankind humans are in habit of collecting the data, storing the data for reusable purposes. Invent of computer systems has provided us the capability of storing the data and reusable things. A Database Management System allows a person to organize, store, and retrieve data from a computer. It is a way of communicating with a computer's —stored memory. Edgar Codd worked for IBM in the development of hard disk systems, and he was not happy with the lack of a search engine in the CODASYL approach, and the IMS model. He wrote a series of papers, in 1970, outlining novel ways to construct databases. His ideas eventually evolved into a paper titled, A Relational Model of Data for Large Shared Data Banks, which described new method for storing data and processing large databases. Records would not be stored in a free-form list of linked records, as in CODASYL navigational model, but instead used a —table with fixed-length records. We are trying to use same schema for different clients of having same set of requirements.

Keywords – Schema, Multitenant

1. INTRODUCTION

In the very early years of computers, —punch cards were used for input, output, and data storage. Punch cards offered a fast way to enter data, and to retrieve it. Herman Hollerith is given credit for adapting the punch cards used for weaving looms to act as the memory for a mechanical tabulating machine, in 1890. Much later, databases came along.

Databases (or DBs) have played a very important part in the recent evolution of computers. The first computer programs were developed in the early 1950s, and focused almost completely on coding languages and algorithms. At the time, computers were basically giant calculators and data (names, phone numbers) was considered the leftovers of processing information. Computers were just starting to become commercially available, and when business people started using them for real-world purposes, this leftover data suddenly became important.

2. RELATED WORK

Multi tenancy: it is an architecture where a single instance serves a number of tenants at a time. There are three different approaches to manage a multi tenant data— 1. Separate databases. 2. Shared database, separate schemas. 3. Shared database, shared schema Which approach is to be chosen depends in the economic considerations, security considerations, number of tenants and size of each tenant. Here, the last approach is of our interest. Shared database, shared schema: in this approach the database contains a set of tables that store the data of all the tenants. No separate database is deployed for every tenant. Also, a single instance is created to serve all the requests from the client. Traditional applications can be easily converted to multi tenant at low cost with less effort. But care should be taken to build a strong reliable architecture otherwise, a poor architecture could lead maintenance nightmare. [1]

Windows azure provides platform for developing multi tenant applications. The tenant's perspective is: ●Isolation● ●Availability ●Scalability ●Costs ●Customizability The provider's perspective is: ●Meeting needs and requirements of the tenants. ●Multiple service level ●Profitability ●Maintainability The architecture could be such as: ●multiple instance, single tenant ●single instance, multi tenant or ●multiple instance, multi tenant The architecture considerations are: ●Application stability: a multi tenant application is more vulnerable to instance failure and the failure could lead to total disruption affecting all the tenants. However, Windows Azure help to mitigate this risk by enabling users to deploy multiple, identical instances of the Windows Azure roles that make up the application. ●Windows Azure offers various instance sizes that enable users to scale up or scale down individual instances. ●Resource Limitations and Throttling: the application limits the maximum number of transactions per second supported by the data storage system used in your application. ●It provides you with a feature to provide you the geographical location where the resource will be hosted. ●Service level agreement: the application provides different service levels for different subscription levels of the service. ●Authorisation and authentication: own customised schemes for authorisation and authentication could be used, or the existing systems like Gmail, Microsoft account could be used.

¹ Student, M.Tech, KNIT Sultanpur, India

² Assistant Professor, KNIT Sultanpur, India

3. PROPOSED WORK

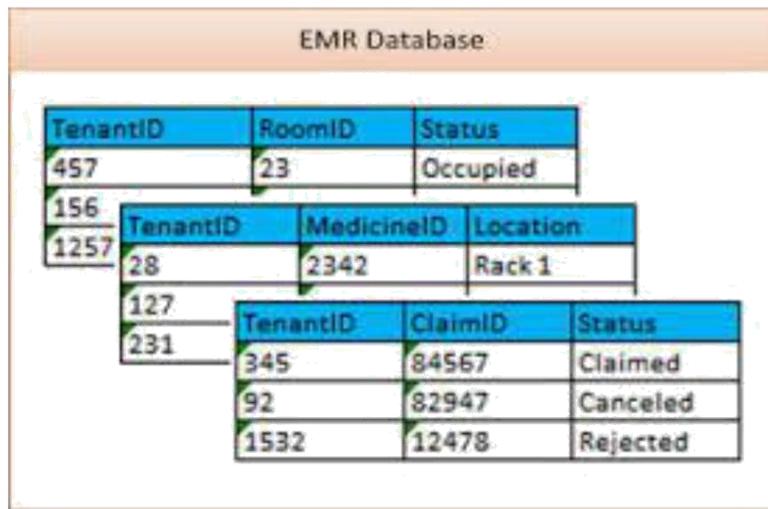
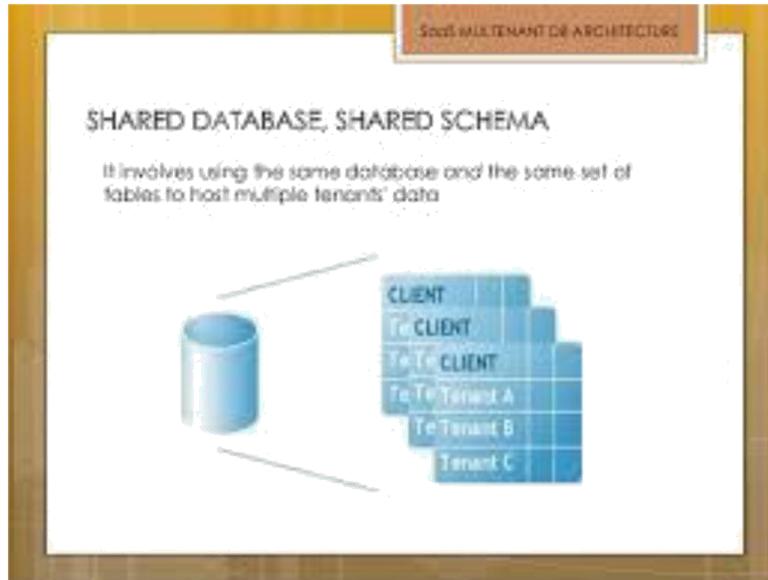
What we are trying to do is to abstract schema design for various target clients. It can be easily identified in following steps

1. Observe Pattern
2. Select common attributes
3. Design schema
4. Identify constraints
5. Create referential integrity
6. Identify keys
7. Open terminal for input
8. Target schema for current input
9. Assign tenant id for the new client
10. Give abstraction to the new client as being alone user of schema
11. Incorporate a security key in the communication of tenant



The image shows a screenshot of a web-based "User Registration Form". The form is contained within a window-like border. It includes the following fields and controls:

- First Name:** A text input field containing the letter "A".
- Last Name:** An empty text input field.
- Username:** An empty text input field.
- Password:** An empty text input field.
- Confirm Password:** An empty text input field.
- Gender:** Two radio button options: "Male" and "Female".
- Address:** A text input field with a small "x" icon on the right side.
- Email ID:** An empty text input field.
- Sign Up:** A blue button located at the bottom right of the form.



4. CONCLUSION AND FUTURE WORK

The proposed architecture provides a cloud based Multi tenant SAAS application. The application designed ensure high commitment by the cloud company and a safe, secure and reliable cloud to multi tenants cutting the major cost involved in purchasing licensed products and services and paying for the space on the server for storing data. The database has been secured using AES encryption. The architecture proposed has put its best efforts to provide a business model for small and medium enterprizes (SME) to grow. Many traditional applications can also be easily converted to multi tenant type.

REFERENCES

- [1] Cor-Paul Bezemer, Andy Zaidman , Multi-Tenant SaaS Applications :Maintenance Dream or Nightmare?, position paper.
- [2] James McCaffrey, AES keeping data secure with, msdn magazine, <http://msdn.microsoft.com/enus/magazine/cc164055.aspx>
- [3] William Stallng, Ch-5, 5.2, Cryptography and Network Security, 4th ed, Pearson, 2006.
- [4] Abdel-Karim Al Tamimi, Performance Analysis of Data Encryption Algorithms, http://www.cs.wustl.edu/~jain/cse56706/ftp/enc_ryption_perf/
- [5] William Stallng, Ch-9 , 9.2, Cryptography and Network Security, 4th ed, Pearson, 2006.