# CSCI5408 – ASSIGNMENT 5

Dhrumil Amish Shah (B00857606)
dh416386@dal.ca

# Assignment 5 - Database Administration and Security Reading
# Multi-tenant Database Access Control

The research paper focuses on outlining a multi-tenant access control model and algorithm for efficiently managing the now emerging multi-tenant database solutions. The use of cloud services for hosting databases and performing database operations is a recent development that allows users to store and use databases over the internet. This multi-tenant database solution serves multiple tenants and multiple users per tenant. The central idea of the research paper is proposing a database schema namely Elastic Extension Tables (EET), its architecture and the access to the databases and schemas that can be granted to the tenants and its multiple users based on the groups the users belong to and the roles assigned to the users[1].

Further, the study describes the security challenges posed due to resource sharing in a cloud-based multi-tenant database solution. Hosting databases and storing data on the cloud is one of the crucial security problems as the data is shared and accessed by multiple users and is at risk of unwanted leakage and unauthorized access by others. The authors further elaborate on three data isolation approaches – Separate Database, Shared Database-Separate Schema, and Shared Database-Shared Schema. The Separate Database data isolation approach stores data for every tenant in a separate database, the Shared Database-Separate Schema data isolation approach lets users store data in the same database instance while every tenant having his schema stored separately and the Shared Database-Shared Schema approach allows data from all tenants to be stored in the same database and same schema. The third approach requires a higher degree of security and isolation as the data for all tenants will be stored in the same database differentiating the data for each of them with tenant ID[1]. The study details the third approach which manages data in two types – tenant's data that is in the shared pool and tenant's data that is isolated. The multi-tenant data isolation approach leads to challenges like ensuring that the tenant has access to his own data only, the tenant's data is stored robustly and with high security, enhancing database performance, and designing a database solution that works for all business domains[1].

The study proposes a multi-tenant database schema to get control over the security risks caused due to the data isolation approaches by launching Elastic Extension Tables (EET) which incorporates Common Tenant Tables (CTT), Extension Tables (ET), and Virtual Extension Tables (VET). It allows the tenants to select from three models where the first one is a multi-tenant relational database, the second one is a combination of multi-tenant relational database and virtual relational database, and the third model is a virtual relational database. The EET model is explained using the EETs, Elastic Extension Tables Proxy Service, Elastic Extension Tables Access Control method and Elastic Extension Tables Access Control algorithm.

The Common Tenant Tables in EET are physical relational tables which is a shared resource between users of a multi-tenant database. These tables fit all business domains such as HR, CRM, accounting, or any other domain. The Virtual Extension Tables let the tenants extend an existing business database or set up a new virtual database by configuring virtual database tables, columns, rows, indexing the columns, primary key to the tables, establishing relationships between the tables, and assigning constraints to the database columns. The third type is Extension Tables (ET) which is made up of eight tables managing each of the tasks mentioned above.

The Elastic Extension Tables Proxy Service (EETPS) works on the multi-tenants' queries by transforming these queries to traditional relational database queries using programs. This proxy service aims at achieving two targets – the first is to retrieve rows from CTTs, VETs, a combination of CTT and VET, and to reduce the cost of building Structured Query Language queries, managing their processing and backend by calling functions of this proxy service. The EETPS provides the tenants to work with their queries using models that simplify database operations for a multi-tenant database solution[1].

The Elastic Extension Tables Access Control (EETAC) describes the access control mechanism that grants access to the tenants using three tables – Access Control Main Entity Tables, Access Control Join Tables, and Information-Schema column view. The Access Control Main Entity Tables encompasses Tenant table (stores the tenant information), User table (stores details of three types of users that can be stored per tenant), Group table (defines the tenant's groups and their levels), Role table (stores the roles and permissions for users and groups), Capacity table (lets tenants authorize their permission to perform operations). The Access Control Join Tables are used to create many-to-many relationships between these main entity tables, and lastly, the Information_Schema view provides information about fields of tables and views of Oracle, MySQL, PostgreSQL databases.

The Elastic Extension Tables Access Control algorithm deals with granting access to the tenants to the columns and rows of the multi-tenant database solution. The algorithm is further divided into three algorithms. The first algorithm validates if the columns the tenant is trying to access is authorized by checking for columns in SELECT and WHERE clauses. The second algorithm deals with fetching roles of the tenant as granted to CTT and VET. The third algorithm is used to fetch columns and their rules only for the ones the tenant has access to.

The research paper details two experiments conducted using the various components such as Elastic Extension Tables Proxy Service, Elastic Extension Tables Access Control methods and algorithms. The main goal of the experiment is to analyse the response time and costs for executing multi-tenant queries in a multi-tenant database setup. The experiments proved that the cost and effort for executing a query for a user having access to some columns or a percentage of columns of the VET are less as compared to that of a user having access to all columns of a VET[1].

## References

[1] Y. Haitham and M. Goyal, “Multi-tenant Database Access Control.” https://ieeexplore-ieee-org.ezproxy.library.dal.ca/document/6755311 (Accessed Jul. 29, 2021).