The Hong Kong Polytechnic University

COMP 3335 – Database Security (Semester 1, 2023/2024)

Assignment

This is an individual assignment. You may use the course material and Internet resources to answer the questions. However, you should not post the questions online and ask for help. Discussion among your peers is encouraged; however, you must produce answers by yourself and in your own words. Any suspicion of plagiarism will be thoroughly investigated. Copying answers from GenAI tools into your assignment is a form of plagiarism. This assignment is due before **Saturday, November 4, 23:59**.
Late submissions will be subjected to a 15% penalty per day, starting at 00:01.
Total: 100 points. Course weight: 11%.

Exercise 1: Database Normalization [28 pts]

The aim of this exercise is to give you hands-on experience with database normalization. We are going to normalize a 1NF table to 2NF and 3NF, which can improve data integrity and facilitate the management of access controls in databases.

The Sales Transaction Table below serves as a comprehensive record-keeping system for a store, capturing every sale transaction (tx) along with associated products, prices, quantity ordered and customer details. The table is given in 1NF (First Normal Form). With a composite primary key of `(TxID, ProductID)`, the table allows a single `TxID` to be linked to multiple `ProductID`s, thereby enabling the logging of multiple products within the same transaction. This feature ensures that all products in a single transaction share the same `Salesperson` and `CustomerName`, providing a consolidated view of each sales interaction. Each salesperson takes a commission on the sale at a rate (`CommRate`) proportional to their rank. Refer to the attached `transactions.sql` MySQL file for the table schema.

Table 1: Sales Transaction Table in 1NF

TxID	ProductID	ProductName	UnitPrice	Qty	TotalPrice	CustomerName	Salesperson	CommRate
1	101	Laptop	1000	1	1000	Alice	Bob	5%
1	102	Mouse	50	2	100	Alice	Bob	5%
2	101	Laptop	1000	1	1000	Carol	Dave	3%
3	103	Keyboard	100	1	100	Carol	Eve	5%
3	104	Monitor	200	1	200	Carol	Eve	5%

Question 1 [10 pts]. Identify the attributes that have a *partial* dependency on the composite primary key. A partial dependency is a condition in which an attribute is dependent on only a portion (subset) of the primary key. [-2 pts for each incorrect dependency]

Question 2 [9 pts]. Create new tables that remove the partial dependencies identified in Q1. Give the table schemas, and identify the primary key(s).

Info. The resulting tables are in Second Normal Form (2NF), which requires that there is no partial dependencies in a table.

You can refer to the process of normalization we used in Lecture 2 for this task (however, we only care about partial dependencies here), as well as the book *Database systems: design, implementation and management*[1] Part 1, Chapter 6, Section 6-3 for more details into the conversion process.

Question 3 [3 pts]. Identify any transitive dependencies in the 2NF tables and list them. A transitive dependency exists when a *nonkey* attribute can determine another nonkey attribute. Here, nonkey means the attribute is not a primary key.

[1]To access the book, go to http://ezproxy.lb.polyu.edu.hk/login?url=https://bc.vitalsource.com/tenants/the_hong_kong_polytechnic_university_explore/books/9780357673096. Login with your NetID. Then, click "Continue without an Account" on VitalSource and accept the terms of use.

Question 4 [6 pts]. Create new tables to eliminate these transitive dependencies. Give the tables description, and identify the primary key(s).

> *Info.* The resulting tables are in Third Normal Form (3NF), which requires that there is no transitive dependencies in a table. Moreover, these tables are also in Boyce-Codd Normal Form (BCNF) because there is no other dependency involving a primary key as a dependent. Therefore, for each functional dependency, every determinant (i.e., the attributes that compose the left side of a functional dependency) is a primary or candidate key to some table.

Exercise 2: Access Control [30 pts]

In Exercise 1, you have normalized a sales table. The next step is to grant access rights to different users. Here are the requirements:

1. Alice, the Sales Manager, should be able to perform CRUD (Create, Read, Update, Delete) operations related to salespeople and product information. Furthermore, she should be able to grant/revoke these access rights to sales executives.
2. Catherine, the Inventory Clerk, maintains the inventory of products. She should be able to perform CRUD operations related to product information.
3. David, a Marketing Executive, is organizing sales campaigns. His role requires extracting information to understand which products are the most popular. Grant him read access to the relevant information.
4. Ewing, the Database Administrator, should have the ability to perform any operations across all tables, create new users, and grant/revoke any of CRUD rights to them. Additionally, ensure Ewing can connect to the DBMS only from IP addresses within the range 192.168.123.x.

Use the root account to create users, Alice, Bob, Catherine, David and Ewing, and grant access privileges to them using SQL statements. Justify your solutions, which are typed as comments in the SQL file.

Exercise 3: Securing Data-in-motion in MySQL [42 pts]

This exercise aims to familiarize students with securing data transmission between a MySQL client and server using encrypted connections. By the end of this exercise, students should be able to create, implement, and verify encrypted connections in MySQL.

> *Info.* The TLS protocol (formerly named SSL in older versions) secures communications by encrypting network traffic between two endpoints. It also supports the authentication of the server through certificates, and can optionally be used to authenticate clients.
>
> Certificates are digital documents that attest the ownership of a public key by the holder of the certificate. They contain information like the name of the certificate holder, the serial number, expiration dates, the copy of the certificate holder's public key, and the digital signature of the certificate-issuing authority. A Certificate Authority (CA) is a trusted entity that issues digital certificates, typically for domain names like `google.com` after verification of ownership.

Preliminaries. You need to install OpenSSL and MySQL Community Edition on your computer.

1. **Installing OpenSSL** (version 1.1.x or 3.x):
   - **Windows**: Download the compiled x64 binaries from Blackboard (taken from Git).
   - **macOS**: Try first the command `openssl version` in a terminal. If the command fails, install OpenSSL using Homebrew with the command: `brew install openssl`.
   - **Ubuntu**: Similarly, try the command `openssl version` in a terminal. If it fails, install OpenSSL using the command: `sudo apt-get install openssl`.

Refer to the online documentation at https://www.openssl.org/docs/manpages.html according to the version you are using for the questions below.

2. **Installing MySQL Community Edition**: Follow the instructions on this page to install MySQL Community Edition 8.0.16+. Download the binaries from this link. Note that you do not need an account to download, simply click on "*No thanks, just start my download.*"

   Connect to your MySQL Server using the root account.[2] Then, execute the command `SHOW VARIABLES LIKE '%ssl%'` and look at the line `has_ssl`. If not `YES`, then your MySQL installation does not support SSL/TLS as it should. Please make sure to upgrade your installation to version 8.0.16 or above.

Question 1 [5 pts]. Connect to your MySQL Server using `mysql -u root -p -h localhost --ssl-mode DISABLED`. Then, execute the command `STATUS` (or `\s` for short) and give a screenshot showing the lines "SSL" and "Server version."

Question 2 [5 pts]. In MySQL's documentation, read about the `ssl-mode` option, and change the mode from `DISABLED` to a mode that forces the use of TLS but does not verify the server certificate. Which mode do you choose? Check the connection status again and report a new screenshot showing the value for "SSL."

Question 3 [5 pts]. Using the command `SHOW VARIABLES LIKE '%ssl%'`, find the TLS server certificate filename that the server is configured to use. Locate this file on disk, then use OpenSSL to parse the certificate and show you its content. Give the value of the `Issuer` of the certificate. Hint: The certificate follows the X.509 standard, so look into `openssl x509`.

Question 4 [12 pts]. If the client does not verify the server's certificate, an adversary could actively interpose on the connection and pretend to be the server. The client would still report that the connection is encrypted. By enforcing certificate validation, we can prevent this attack.

Utilize OpenSSL to create your own CA certificate (and corresponding private key), along with **two** server certificates (and private keys) issued by your CA: one with the Common Name = IAmAHacker.com, the other with Common Name = localhost. Note down the exact commands you used.

- **Hint 1**: Use `openssl ecparam -out <outfile> -name prime256v1 -genkey` command to create an elliptic curve (EC) private key file.
- **Hint 2**: Use `openssl req -new -key <keyfile> -out <CSR>` command to create a certificate signing request (CSR).
- **Hint 3**: Use `openssl req -x509 -key <keyfile> -in <CSR> -out <certificate> -days 365` command to create a self-signed certificate using the provided private key. Note that the CA certificate is self-signed, i.e., it is signed by its own private key.
- **Hint 4**: Use `openssl x509 -req -in <CSR> -CA <CA> -CAkey <keyfile> -CAcreateserial -out <CRT> -days 365` to sign a CSR by the CA's private key and create a certificate.
- **Hint 5**: Use the parameter `-config openssl.cnf` to point to the config file attached to this assignment with `openssl req` commands.

Question 5 [10 pts]. Configure explicitly the MySQL server and client for encrypted connections using the certificates created in the previous step (select the first server certificate created) by writing configurations to `my.cnf` (or `my.ini` on Windows) under `[mysqld]` for the server side and under `[mysql]` for the client side. The server should mandate TLS encryption. Show the lines you added to the config file.

Question 6 [5 pts]. For each server certificate created in Q4, connect to your MySQL Server using `--ssl-mode VERIFY_CA` and `VERIFY_IDENTITY` and report on the success or failure. Explain the importance of the `--ssl-mode` setting on the client side and how it affects the verification process.

[2] Note that the root password, if not empty, is located in the server logs upon the first installation, see in `mysql-error.log` for Windows, `/var/log/mysql/error.log` in Linux. If not, try connecting using `sudo mysql`, then change the root password.