CT069-3-3-DBS

DATABASE SECURITY

APU3F2211CS(DA)

HAND OUT DATE : 13 May 2023

HAND IN DATE : 30 July 2023, 12 Midnight

Group Number: 12

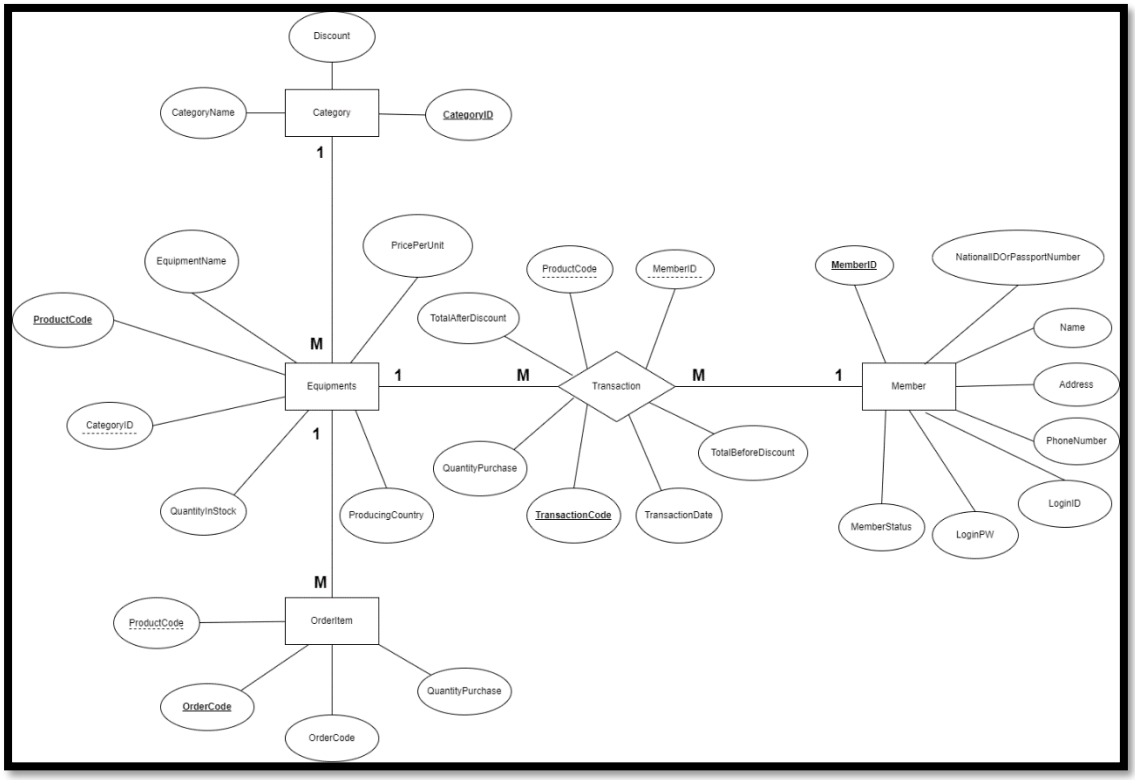
Group Members:

Name	Student Number
Ferdian Marcel	TP058072
Ferdinand Wilson	TP062635
Marcell Agung Wahyudi	TP058650
Michael Henry	TP058088

Table of Contents

1. Entity-Relationship Diagram, & Relational Model

1.1 ERD Model



The above figure shows the ERD model of the database used for this assignment. There are four entity and one relationship exist in the database which includes “Category”, “Equipments”, “OrderItem”, “Transaction”, and “Member”.

For “Category”, the primary key is the “CategoryID” and it has one to many relationships with equipment, which mean one category can be used by many equipment and one equipment can only have one category. Then, in the equipment table, “ProductCode” is the primary key while the “CategoryID” is a foreign key from “Category” table. Equipment table also has one to many relationships with “OrderItem” table, which mean one equipment can be used for many orders. Furthermore, “Equipment” table also have one to many relationships with “Transaction”, which mean one equipment can be transacted many times.

After that, for “OrderItem” entity, the primary key is ”OrderCode” and also one foreign key which is “ProductCode”. For “Transaction”, the primary key is “TransactionCode” while there are two foreign key which include “ProductCode” and “MemberID”. This is because “Transaction” table act as a relationship between “Equipments” and “Member” entity, therefore, both primary key from the two entities must exist in the “Transaction” table.

Finally, “Member” table primary key is “MemberID” and it has one to many relationships with “Transaction”, which mean one member can do many transactions.

1.2 Relational Model

a. Category Table

- Category (Primary Key): Unique identifier for each category.
- CategoryName: Name of the category.
- Discount: Discount applicable to products in this category.

The Category table represents different product categories. Each category has a unique CategoryID, and it includes the CategoryName and Discount applicable to products in that category.

b. Equipment Table

- ProductCode (Primary Key): Unique identifier for each equipment product.
- EquipmentName: Name of the equipment.
- PricePerUnit: Price per unit of the equipment.
- CategoryID (Foreign Key): References the Category table, linking equipment to its category.
- QuantityInStock: Quantity of the equipment available in stock.
- ProducingCountry: Country where the equipment is produced.

The Equipment table stores details about individual equipment products. Each product has a unique ProductCode, and it is associated with a specific CategoryID. PricePerUnit represents the cost, QuantityInStock tracks availability, and ProducingCountry indicates where the equipment is made.

c. Member Table

- MemberID (Primary Key): Unique identifier for each member.
- NationalIDOrPassportNumber: National ID or passport number of the member (encrypted in actual implementation).
- Name: Name of the member.
- Address: Address of the member (encrypted in actual implementation).
- PhoneNumber: Phone number of the member.
- MemberStatus: Status of the member.

- LoginID: Login ID of the member (encrypted in actual implementation).
- LoginPW: Login password of the member (encrypted in actual implementation).

The Member table holds information about members. MemberID is unique for each member. Sensitive data like NationalIDOrPassportNumber, Address, LoginID, and LoginPW are stored in encrypted form. PhoneNumber and MemberStatus provide additional details.

d. Transaction Table

- TransactionCode (Primary Key): Unique identifier for each transaction.
- ProductCode (Foreign Key): References the Equipment table, indicating the purchased product.
- MemberID (Foreign Key): References the Member table, identifying the member involved.
- TransactionDate: Date of the transaction.
- QuantityPurchase: Quantity of the product purchased.
- TotalBeforeDiscount: Total cost before any discounts.
- TotalAfterDiscount: Total cost after applying discounts.

The Transaction table records details of product purchases by members. TransactionCode is unique, and it references ProductCode and MemberID for the involved items. QuantityPurchase, TransactionDate, and Total costs are also tracked.

e. OrderItem Table

- OrderCode (Primary Key): Unique identifier for each order.
- ProductCode (Foreign Key): References the Equipment table, indicating the ordered product.
- OrderDate: Date of the order.
- QuantityPurchase: Quantity of the product ordered.

The OrderItem table stores information about product orders. OrderCode is unique, and it is associated with a specific ProductCode and QuantityPurchase. OrderDate tracks when the order was placed.

f. PurchaseCategory Table

- TransactionCode (Primary Key, Foreign Key): References the Transaction table.
- CategoryID (Primary Key, Foreign Key): References the Category table.

The PurchaseCategory table represents a many-to-many relationship between transactions and categories. It connects specific transactions with the categories of products involved.

2. Backup and Restore Strategy

2.1 Strategy 1

There are three types of backups in the SQL Server such as Full backup, differential backup, and Incremental backups. In this part, the student will explain about Full Backup Strategy.

Full Backup in SQL Server is the most fundamental and comprehensive sort of backup process. By the name of "Full", it copies all the data on the SQL database server which make this as the advantages of using the full backup strategy. Not only that, Full backup strategy also can reduce the loss of the data, particularly when paired with the transaction log backups, it assists to minimize the loss of the data by allowing the data recovery.

Finally, adding a thorough backup procedure into the SQL database security plan is an important practise that provides complete information protection, quicker recovery operations.

2.2 Strategy 2

A differential backup strategy is a strategy which uses a full backup strategy from the start, then periodically take the differential backup strategies in the end. The differential backup strategy is a backup strategy which captures the changes that are already occurring when compared to the last full backup, therefore, it is faster to make and restore when compared to the full backups. The strategy of differential backup creates a balance between the frequency of backups and the size for the backup files.

There are benefits in implementing the differential backup strategy. First things first, differential backups are smaller and easier to implement when compared to full

backups. Then, there is the faster restoration session using differential backup strategy. The differential backup strategy allows the user to only restore the last full backup and the latest differential backup, which in the end is going to reduce the downtime. Following that, there is the reduced storage space usage when implementing the differential backup strategy into the system of the project.

A differential backup strategy is going to take a longer time and time consuming to restore, when the time between full backups is long. The choice of the backup strategy is going to depend for the factors like the size of the data base itself and the frequency of changes.

2.3 Justification

In this database we use the Full backup Strategy because the protection of the backup is the safest backup than the others. By making full backups on the database, it will create a solid data recovery and reduce the possibility of the missing of the data, and the last is assure our database to deal with unanticipated issues.

3. Confidential & Integrity (Individual Work)

3.1 Ferdian Marcel [TP058072]

On this part, student managed to create the memberHiddenDetails feature which is for the Member Role, Store Clerk, and Management, which is not to show the concealed or sensitive information, which includes National ID Passport Number, Login ID, Login Password, and Address. By implementing this functionality, the student has contributed to enhancing the system's capabilities on the row-level security because for example user '1001' will not be able to check another user details.

3.2 Ferdinand Wilson [TP062635]

The student with the TP number of TP062635 has gained an understanding in the creation of several security features for confidentiality and integrity section. The data base gains a good amount of security with the help of the row level security integration for the student. Row level security is a vital step in guarding the confidentiality for the data in the data base of the organization. Each of the member role is only able to edit their own details. After the member has authenticated with their own correct password, the member should be able to edit their own details only.

3.3 Marcell Agung Wahyudi [TP058650]

For confidentiality and integrity, the student managed to create the row level security feature, which is a feature in the database that lets the administrator control which user can see which rows of data, in the case of the DBS project, it is used to give the 'Member' role permission to only see its own row of data, for example, when the user 'Member' wants to see the full details of the 'Member' Table, the only shown information in the Member table is the row of a member which has the MemberID that matches the current user's MemberID, another example is when the student wants to see 'Transaction' Table, the only transaction they can see is the one they themselves did.

3.4 Michael Henry [TP058088]

The work done by the student for Confidentiality and Integrity is encryption of member table. Encrypting the 'Member' table involves converting the stored data into a scrambled format that can only be deciphered with a proper decryption key. This step ensures that even if unauthorized access occurs, the exposed information remains

unreadable and meaningless. Encrypting the 'Member' table contributes to maintaining data integrity by preventing unauthorized modifications. Any tampering attempts on the encrypted data will render it useless without the appropriate decryption key.

4. Key Learnings (Individual Work)

4.1 Ferdian Marcel [TP058072]

In this group work, the student has learned many things in the SQL queries, such as the security of the database which is very important to company or business. The student learned new things such as encryption, back up the database, and row-level security.

On the individual part, this is the crucial part to them because they need to make the code suitable to another student because this is group task, and the queries also need to be synchronized.

In conclusion, this project significantly enhanced the student's understanding of the databases, especially practises, regulations, and SQL abilities. This lesson will surely help the future of the student in finding career in Data Analytics (SQL).

4.2 Ferdinand Wilson [TP062635]

The student (TP062635) has gained a decent understanding of the implementation method for data base security, the importance of data base security as well as how to ensure that the correct user is able to access the correct data that they are meant to be accessing. The student has succeeded in setting up a data base security for securing the details for member role as well as updating the member details for member role.

The student has learned how to create codes individually, that is able to be incorporated into the system of other members. This is a key section in this project as it enables the student to learn more about how to manage time and resources in a group project, especially in building a security system for a data base. Then, the student successfully learns about teamworking as well as how to prevent the code from other students in the project from interfering and interrupting each other.

The project has helped the student in the teamworking and communication skills, especially during the development and implementation of row level security part. The other group members have helped in the troubleshooting for the row level security part in how to manage and make the system more secure than ever before.

In short, the module Database Security has successfully helped the student in understanding the importance of security in a data base system as well as how to manage, create, and implement a security system into a data base system in Microsoft SQL Server program.

4.3 Marcell Agung Wahyudi [TP058650]

Throughout this course, the student has gained a solid grasp of database security principles and practices, from the importance of protecting sensitive information, to setting up access controls, and ensuring data accuracy. The student had also explored various ways to do data encoding, verify user, detect unauthorized activities, create row level securities and more.

The individual parts of the project helped the student the most since they can only rely on themselves, meaning that they need to not only make the code work for themselves, but also work with and alongside their teammate's codes, this not only teaches us more about teamworking but also about avoiding clashes in the code and work division.

The assignment also made us brainstorm in many offline meetings to figure out the system and making sure it works with each and every team member's functions, making sure the row level security works with other member's update function, making sure buying equipment works with transaction, and etc.

Overall, the project has helped the student tremendously from the standpoint of database understanding to database practices, disciplines, and SQL in general, which will undoubtedly improve the future career of the student as Data Analyst.

4.4 Michael Henry [TP058088]

A complete study of SQL queries and database security evolved in this collaborative group effort, highlighting the important significance of preserving information inside organizations. This module covered a wide range of subjects, with a particular emphasis on encryption techniques, database backups, and the strategic application of row-level security measures.

The project's personalized segment was very noteworthy. Participants were tasked with changing their code to fit neatly with the work of their peers. This required a rigorous approach to ensure that the code not only functioned optimally within its own environment but also seamlessly integrated with the larger project. This collaborative synchronization extended to the queries' selected information, ensuring a consistent and aligned approach throughout the project's execution.

In summary, the project resulted in significant advancements in database comprehension, with a focus on practical implementations, regulatory conformance, and the polishing of SQL capabilities. These new insights are likely to provide a

significant advantage in future professions, particularly in the field of Data Analytics, which is dependent on SQL competence. Participants gained not only technical knowledge but also valuable skills such as cooperation, code integration, and effective communication in a collaborative atmosphere. This diverse experience lays the groundwork for their continued development as skilled professionals in the dynamic field of database management and analytics.

5. References

- Datto. (2021, May 18). Incremental vs. Differential Backup. Datto Blog. https://www.datto.com/blog/incremental-vs-differential-backup

- Crocetti, R. F. and P. (2022, December 14). *Types of backup explained: Full, incremental, differential, etc.: TechTarget*. Data Backup. https://www.techtarget.com/searchdatabackup/feature/Full-incremental-or-differential-How-to-choose-the-correct-backup-type#:~:text=The%20most%20common%20backup%20types,synthetic%20full%20backups%20and%20mirroring.