

University of Wales

Trinity Saint David

MODULE CODE/ COD Y MODIWL: ACCC5001

DATABASE MANAGEMENT SYSTEMS

MODULE TUTOR: Koushik Modak

Antoci Bianca

1817784

Task 2

Table of Contents

Report	2
Additional task	9
References	15

Report

a) "Include in the report a screenshot of the sales and audit data table and how they can be used to detect any DDL and DML transaction occurring." [1]

SQL statements are split into two significant and meaningful groups: DDL (the language of data definition) and DML (data manipulation language).

DDL is used to construct and adapt our tables (CREATE, Change OR DROP) and DML is used to deal with the tables' data (DELETE, INSERT OR UPDATE) [2].

With the following tables below we can see how they can be used.

Sales Table:

In the current database schema, the Lease table corresponds to the Sales table.

Below is the screenshot of the Lease table

leaseNumber	clientId	ownerID	propertyID	methodofpay	depositPaidSta...	startDate	endDate
1	1	1	1	card	yes	2020-01-01 00:0...	2020-01-01 00:0...
2	2	2	2	paypal	no	2020-02-01 00:0...	2020-01-01 00:0...
3	3	3	3	card	no	2020-05-01 00:0...	2020-01-01 00:0...
4	4	4	4	card	yes	2020-02-01 00:0...	2020-01-01 00:0...
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Screenshot 1 - sales table

Audit data table:

The database contains two audit data tables as follows:

1. PRGPropertyAudit table

This table stores the audit data retrieved from the Insert and delete DML transactions.

Below is the screenshot of the PRGPropertyAudit table

	Audit_id	CustomerID	VisitID	PropertyID	EventType	UserResponsi...	TransactionDate
▶*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Screenshot 2 - audit data table

2. PRGPropertyUpdateAudit table

This table stores the audit data retrieved from the update DML transaction. The table holds the details of @old and @new data items.

Below is the screenshot of PRGPropertyUpdateAudit table:

	Audit_id	Old_CustomerID	Old_VisitiD	Old_PropertyID	New_Custom...	New_VisitiD	New_PropertyID	UserResponsi...	TransactionDate
▶*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Screenshot 3 - PRGPropertyUpdateAudit table

b) "Identify the advantages to be gained from your proposed solution and use appropriate examples from your solution (in Task 1) to support your discussion. Show queries that can be run using the database." [1]

To maintain data integrity in SQL we use triggers. They are automatically executed in response to certain events on a particular table [3].

From the proposed solution exemplified above, we shall test the four triggers created and their advantages :

1. trg_create_alter_drop trigger

This trigger keeps track of all the changes that take place on the database tables. These changes include table creation, table alteration and table dropping.

Testing with table creation

SQL CODE:

```
Create table NewCustomer (  
CustomerID INT IDENTITY PRIMARY KEY,  
FName VARCHAR(50));
```

Contents of PRGDatabaseAudit table

Audit_id	event_data	LoginName
1	<EVENT_INSTANCE><EventType>CREATE TABLE</EventTy...	dbo
2	<EVENT_INSTANCE><EventType>DROP TABLE</EventType...	dbo
3	<EVENT_INSTANCE><EventType>CREATE TABLE</EventTy...	dbo
4	<EVENT_INSTANCE><EventType>CREATE TABLE</EventTy...	dbo
5	<EVENT_INSTANCE><EventType>CREATE TABLE</EventTy...	dbo

Screenshot 4 - Contents of PRGDatabaseAudit table

Based on the last record, details of event data with the login name of the user who created the table have been appended in the audit table. The event data consists of the following details:

```

<EVENT_INSTANCE>
  <EventType>CREATE_TABLE</EventType>
  <PostTime>2020-10-30T06:23:37.207</PostTime>
  <SPID>52</SPID>
  <ServerName>DESKTOP-EEBUP0T\SQLEXPRESS</ServerName>
  <LoginName>DESKTOP-EEBU P0T\Gmail Softy</LoginName>
  <UserName>dbo</UserName>
  <DatabaseName>Bianca_Estate_Agent</DatabaseName>
  <SchemaName>dbo</SchemaName>
  <ObjectName>NewCustomer</ObjectName>
  <ObjectType>TABLE</ObjectType>
  <TSQLCommand>
    <SetOptions ANSI_NULLS="ON" ANSI_NULL_DEFAULT="ON"
ANSI_PADDING="ON" QUOTED_IDENTIFIER="ON" ENCRYPTED="FALSE" />
    <CommandText>Create table NewCustomer (
CustomerID INT IDENTITY PRIMARY KEY,
FName VARCHAR(50))</CommandText>
  </TSQLCommand>
</EVENT_INSTANCE>

```

Testing with table alteration

SQL CODE:

```
ALTER TABLE NewCustomer add LName VARCHAR(50);
```

Contents of PRGDatabaseAudit table

Audit_id	event_data	LoginName
1	<EVENT_INSTANCE><EventType>CREATE TABLE</EventTy...	dbo
2	<EVENT_INSTANCE><EventType>DROP TABLE</EventType...	dbo
3	<EVENT_INSTANCE><EventType>CREATE TABLE</EventTy...	dbo
4	<EVENT_INSTANCE><EventType>CREATE TABLE</EventTy...	dbo
5	<EVENT_INSTANCE><EventType>CREATE TABLE</EventTy...	dbo
6	<EVENT_INSTANCE><EventType>ALTER TABLE</EventType...	dbo

Screenshot 5 - Contents of PRGDatabaseAudit table

A new record of Audit ID 6 has been populated in the PRGDatabaseAudit table. The event data field contains the following information:

```
<EVENT_INSTANCE>
  <EventType>ALTER_TABLE</EventType>
  <PostTime>2020-10-30T06:33:30.870</PostTime>
  <SPID>52</SPID>
  <ServerName>DESKTOP-EEBUP0T\SQLEXPRESS</ServerName>
  <LoginName>DESKTOP-EEBU P0T\Gmail Softy</LoginName>
  <UserName>dbo</UserName>
  <DatabaseName>Bianca_Estate_Agent</DatabaseName>
  <SchemaName>dbo</SchemaName>
  <ObjectName>NewCustomer</ObjectName>
  <ObjectType>TABLE</ObjectType>
  <Alter Table ActionList>
    <Create>
      <Columns>
        <Name>LName</Name>
      </Columns>
    </Create>
  </Alter Table ActionList>
  <TSQLCommand>
    <SetOptions ANSI_NULLS="ON" ANSI_NULL_DEFAULT="ON"
ANSI_PADDING="ON" QUOTED_IDENTIFIER="ON" ENCRYPTED="FALSE" />
```

```

    <CommandText>ALTER TABLE NewCustomer add LName
    VARCHAR(50)</CommandText>
  </TSQLCommand> </EVENT_INSTANCE>

```

Testing with table dropping

SQL CODE

```
DROP TABLE NewCustomer;
```

Contents of PRGDatabaseAudit table

Audit_id	event_data	LoginName
1	<EVENT_INSTANCE><EventType>CREATE TABLE</EventTy...	dbo
2	<EVENT_INSTANCE><EventType>DROP TABLE</Event Type...	dbo
3	<EVENT_INSTANCE><EventType>CREATE TABLE</Event Ty...	dbo
4	<EVENT_INSTANCE><EventType>CREATE TABLE</Event Ty...	dbo
5	<EVENT_INSTANCE><EventType>CREATE TABLE</Event Ty...	dbo
6	<EVENT_INSTANCE><EventType>ALTER TABLE</Event Type...	dbo
7	<EVENT_INSTANCE><EventType>DROP TABLE</Event Type...	dbo

Screenshot 6 - Contents of PRGDatabaseAudit table

A new record with Audit id 7 has been added to the table. Below are the contents of the event data field.

```

<EVENT_INSTANCE>
  <EventType>DROP_TABLE</EventType>
  <PostTime>2020-10-30T06:38:13.173</PostTime>
  <SPID>52</SPID>
  <ServerName>DESKTOP-EEBUP0T\SQLEXPRESS</ServerName>
  <LoginName>DESKTOP-EEBU P0T\Gmail Softy</LoginName>
  <UserName>dbo</UserName>
  <DatabaseName>Bianca_Estate_Agent</DatabaseName>
  <SchemaName>dbo</SchemaName>
  <ObjectName>NewCustomer</ObjectName>
  <ObjectType>TABLE</ObjectType>
  <TSQLCommand>
    <SetOptions ANSI_NULLS="ON" ANSI_NULL_DEFAULT="ON"
    ANSI_PADDING="ON" QUOTED_IDENTIFIER="ON" ENCRYPTED="FALSE" />
    <CommandText>DROP TABLE NewCustomer</CommandText>
  </TSQLCommand>
</EVENT_INSTANCE>

```

2. trg_after_insert Trigger

This trigger keeps track of all the records inserted in the Lease table.

DML Statement for data insertion

```
INSERT INTO Lease (leaseNumber, clientID, ownerID, propertyID,  
methodofpay, depositPaidStatus, startDate, endDate) VALUES  
(5, 4, 4, 4, 'card', 'yes', (convert(datetime,'01-02-20 00:00:00 AM',5)),  
(convert(datetime,'01-01-20 00:00:00 AM')));
```

Contents of the PRGPropertyAudit

Audit_id	CustomerID	VisitID	PropertyID	EventType	UserResponsible	TransactionDate
1	4	5	4	INSERT	dbo	2020-10-30 06:45:29.760

Screenshot 7 - Contents of the PRGPropertyAudit

A new record has been inserted in the PRGPropertyAudit table bearing the customer id, visit id, property id, event type, user responsible and the transaction date of the newly created record. In this case, the event type is INSERT.

3. trg_after_delete Trigger

This trigger keeps track of all the records deleted in the lease table.

DML Statement for data deletion

```
Delete from Lease WHERE leaseNumber = 5;
```

Contents of the PRGPropertyAudit

Audit_id	CustomerID	VisitID	PropertyID	EventType	UserResponsible	TransactionDate
1	4	5	4	INSERT	dbo	2020-10-30 06:45:29.760
2	4	5	4	DELETE	dbo	2020-10-30 06:52:46.077

Screenshot 8 - Contents of the PRGPropertyAudit

A new record has been inserted in the PRGPropertyAudit table bearing the customer id, visit id, property id, event type, user responsible and the transaction date of the deleted record. In this case, the event type is DELETE.

4. trg_after_update Trigger

This trigger keeps a report of all the records updated. Both the old data items and new/updated data items are stored.

DML Statement for testing the trigger:

UPDATE Lease SET ClientID = 2 WHERE leaseNumber = 4;

Contents of the PRGPropertyUpdateAudit Table

Audit_id	Old_CustomerID	Old_VisitID	Old_PropertyID	New_Custome...	New_VisitID	New_PropertyID	UserResponsi...	TransactionDate
1	4	4	4	2	4	4	dbo	2020-10-30
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Screenshot 9 - Contents of the PRGPropertyAudit

From the screenshot above, the customer number has been updated from 4 to 2 while the visit id and property id remain unchanged even after the update.

Additional task

a) "Discuss the term Transaction and Concurrency as used in the DBMS."
[1]

In DBMS, the Transaction is a set of logically related operations that contains a group of tasks performed by a single user in order to access the database contents. In other words, we say that a transaction is an operation or list of actions that are being achieved by an individual user, which reads or renews the essence of the database [4]. The action notion in DBMS is completed as an individual entity. If the database action doesn't renew the database but just brings back info, this category of action is named a read-only transaction. A good transaction can modify databases from dependable state to a different one [5].

A database is a common property accessed and it is handled by a lot of users and used concurrently. With Concurrency, it is possible to have executions of several transactions at the same time that can try to access the same rows of the database during overlapping periods of time. Such concurrent logins, called collisions, can cause inconsistencies or errors if not handled properly [5].

b) "ACID properties of Transaction and why they are necessary." [1]

ACID (Atomicity, Accuracy, Separation, Durability) implies a typical set of properties in database systems that guarantee that database transactions are correctly processed. [6].

Atomicity

Atomicity states determine that all transaction activities must take place at once or that the transaction is interrupted. An atomic transaction guarantees that any commit has successfully finished the total process or that the database is rolled back to its state before the commit is proposed in situations where the connection is broken during the middle of an operation.

In atomicity, there are two actions: abort and commit.

Aborting a transaction means that changes will not be made and therefore all the changes made will not be visible meanwhile committing entails permanently saving the changes and all the changes will be made visible.

A transaction must be executed in its entirety in order to achieve the correctness of the database state[7].

Consistency

Consistency involves leaving a database in a consistent state before and after a transaction has taken place. It is also assured that a transaction can only bring the database from one valid state to a special one by preventing data corruption [7].

It is then the duty of the user to control the consistent state of the database [8].

Isolations

It displays that the data which is used at the time of execution of a transaction can not be utilized by another transaction until the primary one is concluded. The isolation property of a DBMS is imposed by the concurrency control subsystem of the DBMS [7].

Durability

Durability states that a transaction has made a lasting improvement to the database, so the durability property is used to illustrate the accomplishment of the stable state of the database.

Durability cannot be lost by the malfunction of the device or the wrong execution of an invalid transaction. When a transaction is completed, the stable state of a database is normally entered and the consistent state should not be missing, except in the case of a machine breakdown.

Usually, the Durability property of a DBMS is maintained by the recovery subsystem of the DBMS [7].

C) “Discussion on the three anomalies such as Dirty Read, Unrepeatable Read and Lost Update caused by interleaved schedule.”[1]

When different transactions are completed concurrently, then it might lead to a few problems. These problems are called concurrency problems. [8]

The following diagram illustrates the three Concurrency Problems [8].

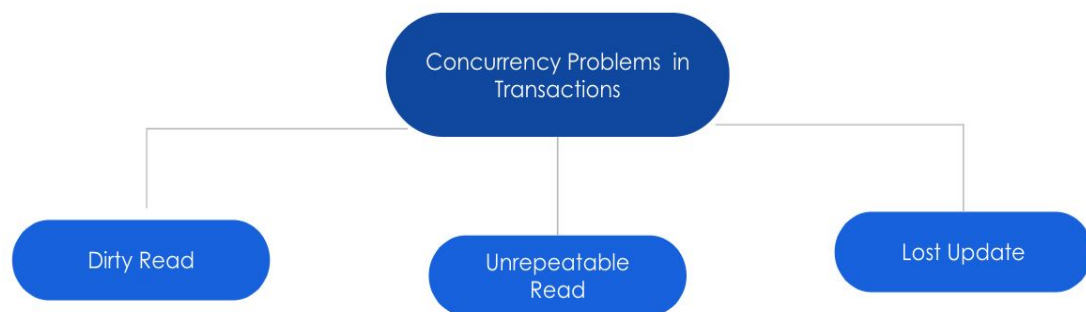


Diagram 1

Dirty Read

The dirty read refers to a situation where one transaction updates an item of the database, and for some reason, the transaction then fails. Another transaction then accesses the modified ledger entry before authentically changing it back to the original value. For instance, if transaction A updates a database item that is read and accessed by transaction B and transaction A aborts, then transaction B will have the values of the unstable database [9].

Unrepeatable Read

Unrepeatable read is also known as Inconsistent Retrievals Problem.

Unrepeatable read occurs when a transaction is updating some piece of data while another transaction is calculating some summary function over the same piece of data.

For example, while transaction A is updating a record, transaction B is reading the same record and also doing some further processing on it. Now, when transaction B reads the record, the new value will be inconsistent with the previous value [9].

Lost Update

Lost Update occurs where the behavior of two transactions accessing the same database objects do not correct the value of other database items.

For example, if 2 transactions A and B read a record and so update it, the primary update effect is going to be overwritten by the second update effect [9].

d) "Two-phase locking and how it can avoid the problem caused by Dirty Read and how two-phase locking would help to avoid problem caused by Dirty Read"[1]

The Two-Phase Locking Protocol (2PL) may be a DBMS mechanism that provides consistency by enforcing a lock on transaction data that prevents other transactions to concurrently match the same data.

In the first section, before the execution of the transaction begins, a transaction requests permission for the lock it needs.

The transaction hits all the locks in the second part, and the third stage is also started as soon as the transaction drops its first lock. In the third level, no new locks can be used for the transaction. Only the acquired locks are released [10] .

The Two-Phase Locking protocol grants all transactions to come up with a lock or unlock request in two parts: Growing Phase and Shrinking Phases.

Growing Phase

In this phase, a transaction obtains locks but may not release any locks [10].

Shrinking Phases

In this phase, a transaction may release locks but may not obtain any lock [10].

In Dirty Read one transaction updates an item of the database, and for some logic, the transaction then fails. Another transaction accesses the updated database item before it is changed back to the initial value. Two-phase locking will help to prevent Dirty Read by ensuring a lock is only released when a transaction is complete.

References

- [1] University of Wales, Trinity Saint David, "Database Management System(ACCC5001_LON)"[Online].
Available:https://moodle.uwtsd.ac.uk/pluginfile.php/359993/mod_resource/content/12/ACCC5001-%20Database%20Mgt%20System%20A2_October2020%20term.pdf
- [2] Jignesh Raiyani. (2019). Database Level DDL Triggers on Tables.
Available:
<https://www.sqlshack.com/database-level-ddl-trigger-over-the-table/>.
- [3] MySql Tutorial . (2019). MySQL Triggers. Available:
<https://www.mysqltutorial.org/mysql-triggers.aspx/>.
- [4] Atluri, V., & Samarati, . (2012). *Security of Data and Transaction Processing*. Berlin/Heidelberg: Springer Science & Business Media.
- [5] guru99. (2020). *DBMS Transaction Management: What are ACID Properties?*. Available:
<https://www.guru99.com/dbms-transaction-management.html#:~:text=A%20transaction%20is%20a%20program,called%20a%20read%20Only%20transaction.>
- [6] tutorialgateway. (2020). *ACID Properties in SQL Server*. Available:
<https://www.tutorialgateway.org/acid-properties-in-sql-server/>.
- [7] geeksforgeeks. (2019). *ACID Properties in DBMS*. Available:
<https://www.geeksforgeeks.org/acid-properties-in-dbms/>.

[8] Thomasian, A. (2013). Database Concurrency Control. New York City: Springer Science & Business Media.

[9] Graefe, G. (2019). On Transactional Concurrency Control. San Rafael: Morgan & Claypool Publishers.

[10] guru99. (2020). DBMS Concurrency Control: Timestamp & Lock-Based Protocols. Available:
<https://www.guru99.com/dbms-concurrency-control.html#6>.