



## Practical NO : 5

Aim : Implementing SQL commands for triggers

### Theory :

#### SQL triggers :

A trigger is a set of SQL statements that reside in system memory with unique names. It is a specialized category of stored procedure that is called automatically when a database server event occurs. Each trigger is always associated with a table.

A trigger is called a special procedure because it cannot be called directly like a stored procedure.

The key distinction between the trigger and procedure is that a trigger is called automatically when a data modification events occurs against a table. A stored procedure, on the other hand, must be invoked directly.

#### Types of SQL triggers :

These are three main types of SQL triggers

1]

#### Data Definition language (DDL) triggers :

DDL triggers are fired in response to the DDL events such as CREATE, ALTER, and



Chhatrapati Shahu Maharaj Shikshan Sanstha's  
**CHH. SHAHU COLLEGE OF ENGINEERING**

Kanchanwadi, Paithan Road, Aurangabad.

Date :

DROP statements.

We can create these triggers at the database level or server level, depending on the type of DML events.

It can also be executed in response to certain system-defined stored procedure that do DDL-like operations.

## 2] DML triggers

DML triggers are fired in response to DML events like INSERT, UPDATE, and DELETE statements in the user's table or view.

It can also be executed in response to DML-like operations performed by system-defined stored procedures.

The DML triggers can be classified into two types :

- After triggers
- Instead of triggers

After triggers :

After triggers fires, when SQL Server completes the triggering action successfully, that fired it, generally, this trigger is executed when a table completes an insert, update or delete operation.



It is not supported in views sometimes  
It is known as FOR triggers

we can classify this trigger further into three types

- 1] AFTER INSERT Triggers
- 2] AFTER UPDATE Triggers
- 3] AFTER DELETE Triggers

### 3] Logon triggers :

Logon triggers are fires in response to a LOGON event. The LOGON event occurs when a user session is generated with an SQL Server instance, which is made after the authentication process of logging is completed but before establishing a user session.

As a result, the SQL Server error log will display all messages created by the triggers including error messages and the PRINT statement messages.

If authentication fails, logon triggers do not execute. These triggers may be used to audit and control server sessions, such as tracking login activity or limiting the number of sessions for a particular login.



Chhatrapati Shahu Maharaj Shikshan Sanstha's

**CHH. SHAHU COLLEGE OF ENGINEERING**

Kanchanwadi, Paithan Road, Aurangabad.

Date :

## Practical No : 6

Aim : Normalization ( 1NF, 2NF, 3NF, BCNF ) in database.

Theory : Normalization :

Normalization is the process of organizing the data in the database.

- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like insertion, update, and deletion anomalies.
- Normalization divides the larger table into smaller and links them using relationships.
- The normal form is used to reduce redundancy from the database table.

### 1NF [ First Normal Form ]

- A relation will be 1NF if it contains an atomic value.
- It states that an attribute of a table cannot hold multiple values. It must hold only single-valued attribute.
- First normal form disallows the multivalued attribute, composite attribute, and their combinations.



### 2NF (Second Normal form)

In the 2NF, relational must be in 1NF

- In the second normal form, all non-key attributes are fully functional dependent on the primary key.

### 3NF (Third Normal form)

A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency

- 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form
- A relation is in third normal form if it holds at least one of the following conditions for every non-trivial function dependency  $x \rightarrow y$ 
  1.  $x$  is a super key.
  2.  $y$  is a prime attribute, i.e. each element of  $y$  is part of some candidate key.

### Boyce Codd normal form (BCNF)

BCNF is the advance version of 3NF.

It is stronger than 3NF

- A table is in BCNF if every functional dependency  $x \rightarrow y$ ,  $x$  is super key of the table



- for BCNF, the table should be in 3NF, and for every FD, LHS is super key.

### Advantages of Normalization.

- Normalization helps to minimize data redundancy.
- Greater overall database organization.
- Data consistency within the database.
- Much more flexible database design.
- Enforces the concept of relational integrity.

### Disadvantage Normalization.

- you cannot start building the database before knowing what the user needs.
- The performance degrades when normalizing the normalizing the relations to higher normal form i.e 4NF, 5NF.
- It is very time-consuming and difficult to normalize relation of a higher degree.
- careless decomposition may lead to a bad database design, leading to serious problems.



## Practical No : 7

Aim : Implementing Transactions in SQL.

Theory :

### Transaction :

A transaction can be defined as a group of tasks. A single task is the minimum processing unit which cannot be divided further.

A transaction usually means that the data in the database has changed. One of the major uses of DBMS is to protect the user's data from system failures.

The transaction is any one execution of the user program in DBMS.

Executing the same program multiple times will generate multiple transaction.

e.g Transaction to be performed to withdraw cash from an ATM vestibule.

### Transaction property

The transaction has the four properties. These are used to maintain consistency in a database, before and after the transaction.



### property of transaction :

- 1] Atomicity
- 2] Consistency
- 3] Isolation
- 4] Durability.

#### 1] Atomicity :

It states that all operations of the transaction take place at once if not the transaction is aborted.

There is no midway, i.e. the transaction cannot occur partially. Each transaction is treated as one unit and either run to completion or is not executed at all.

Atomicity involves the following two operations:

- 1] Abort : If a transaction aborts then all the changes made are not visible.
- 2) Commit : If a transaction commits then all the changes made are visible.
- 3] Consistency : The integrity constraints are maintained so that the database is consistent.



Chhatrapati Shahu Maharaj Shikshan Sanstha's

**CHH. SHAHU COLLEGE OF ENGINEERING**

Kanchanwadi, Paithan Road, Aurangabad.

Date :

before and after the transaction

- The execution of a transaction will leave a database in either its prior stable state or a new stable state.
- The consistent property of database states that every transaction seen a consistent database instance.
- The transaction is used to transform the database from one consistent state to another consistent state

### 37 Isolation:

It shows that the data which is used at the time of execution of a transaction cannot be used by the second transaction until the first one is completed.

In isolation, if the transaction  $T_1$  is being executed and using the data item  $x$ , then that data item can't be accessed by any other transaction  $T_2$  until the transaction  $T_1$  ends.

The concurrency control subsystem of the DBMS enforced the isolation property.



## 4] Durability :

The durability property is used to indicate the performance of the database's consistent state.

It states that the transaction made the permanent changes.

They cannot be lost by the erroneous operation of a faulty transaction or by the system failure.

When a transaction is completed, then the database reaches a state known as the consistent state.

The consistent state cannot be lost, even in the event of a system's failure.

The recovery subsystem of the DBMS has the responsibility of durability property.

## # Commands of transactions in SQL

## 1] Commit Command :

It is also known as Transactional command. The command is used by the database to save changes.

Syntax : COMMIT ;

**CHH. SHAHU COLLEGE OF ENGINEERING**

Kanchanwadi, Paithan Road, Aurangabad.

Date :

2]

**Rollback command :**

The `ROLLBACK` command is the transactional command used to undo transaction that have not already been saved to the database. Rollback command is used to undo the set of transaction.

Syntax : `ROLLBACK`

3]

**Savepoint command :**

Savepoint is the point in a transaction when we roll the transaction back to a certain point without rolling the entire operation.

Syntax : `SAVE TRANSACTION savepoint_name;`

4]

**Set transaction command :**

The `SET TRANSACTION` command will be used to initiate database transaction. The command is used to generate the characteristics of transaction.

Syntax : `SET TRANSACTION ISOLATION LEVEL <level name>`



Chhatrapati Shahu Maharaj Shikshan Sanstha's  
**CHH. SHAHU COLLEGE OF ENGINEERING**

Kanchanwadi, Paithan Road, Aurangabad.

Date :

## Practical No : 8

Aim : Design database for Bank management System

Theory :

E-R diagram for database Bank management system.



Chhatrapati Shahu Maharaj Shikshan Sanstha's

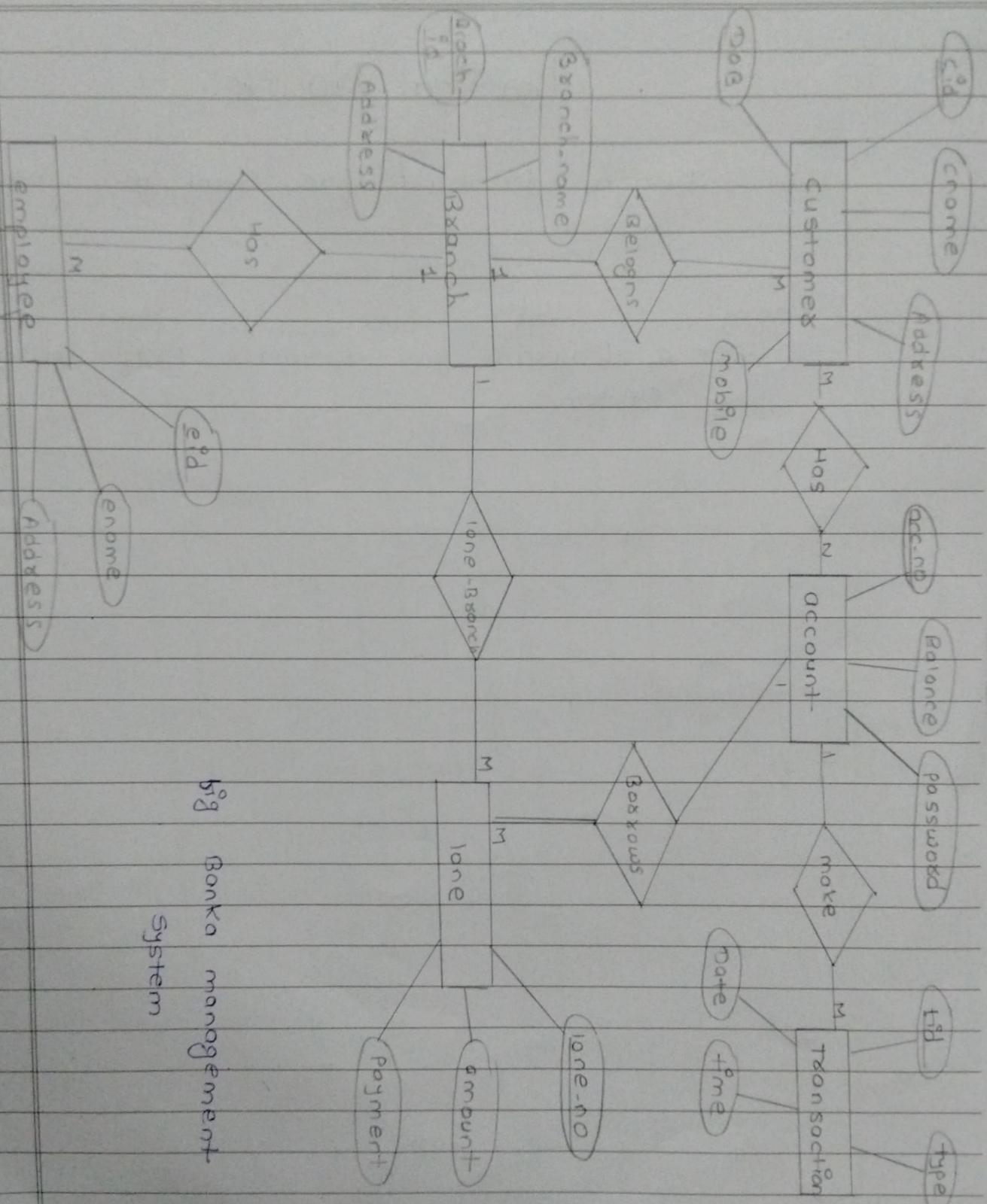
## CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date :



CHH. SHAHU COLLEGE OF ENGINEERING  
Kanchanwadi, Paithan Road, Aurangabad.





Chhatrapati Shahu Maharaj Shikshan Sanstha's

# CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.



Chhatrapati Shahu Maharaj Shikshan Sanstha's

# CHH. SHAHU COLLEGE OF ENGINEERING

Kanchanwadi, Paithan Road, Aurangabad.

Date : 17

Account

acc-no	Balance	password
--------	---------	----------

Transaction

tran-id	type	Date	time	acc-no
---------	------	------	------	--------

Customer

c-id	mobile	DoB	cname	Address	Bran-id
------	--------	-----	-------	---------	---------

Branch

Bran-id	Bran-name	address
---------	-----------	---------

Tone

toneid	amount	payment	acc-no	Bran-id
--------	--------	---------	--------	---------

Employee

e-id	ename	address	DoB	Bran-id
------	-------	---------	-----	---------

Customer - account

c-id	mobile
------	--------