MIS 301 Relational Database Management System

DATABASE MANAGEMENT SYSTEM

Integrity and Security: concept of triggers, stored procedures

Lecture 17 & 18

TRIGGERS IN DBMS

- Triggers are the SQL statements that are automatically executed when there is any change in the database.
- The triggers are executed **in response to certain events** in a particular table like INSERT, UPDATE or DELETE .
- Syntax:

```
create trigger Trigger_name
(before | after)
[insert | update | delete]
on [table_name]
[for each row] [trigger_body]
```

TRIGGERS IN DBMS

- **BEFORE | AFTER:** It specifies when the trigger will be initiated i.e. before the ongoing event or after the ongoing event.
- **FOR EACH ROW:** Row-level trigger gets executed when any row value of any column changes.
- **TRIGGER BODY:** It consists of queries that need to be executed when the trigger is called.
- Example:

```
CREATE TRIGGER new_marks
BEFORE INSERT
ON Student
FOR EACH ROW
SET new.Marks = new.Marks + 5;
```

- *The **new** keyword refers to the row that is getting affected.*

ADVANTAGES OF TRIGGERS

1. Triggers provide a way to check the integrity of the data. When there is a change in the database the triggers can adjust the entire database.

2. Instead of putting the same function call all over the application a trigger can be put and it will be executed when the concerned action takes place.

DISADVANTAGES OF TRIGGERS

1. Triggers may be difficult to troubleshoot as they execute automatically in the database. If there is some error then it is hard to find the logic of trigger because they are fired before or after updates/inserts happen.

2. The triggers may increase the overhead of the database as they are executed every time any field is updated.

Stored Procedures

- A stored procedure in SQL is a type of code in SQL that can be stored for later use and can be used many times.
- Values can be passed to stored procedures.
- A stored procedure is a subroutine available to applications that access a relational database management system.
- Such procedures are stored in the database data dictionary.
- Uses for stored procedures include data-validation and access-control.
- Syntax:

```
CREATE PROCEDURE procedure_name
 AS
sql_statement
GO;
```

- Syntax for execution of the stored procedure:

```
EXEC procedure_name;
```

STORED PROCEDURES

- Example:

```
CREATE PROCEDURE Durgapur_residents
AS
SELECT * FROM students WHERE city=‘Durgapur’
GO;

EXEC Durgapur_residents;
```

Stored Procedure with Parameters

- Example:

```
CREATE PROCEDURE city_residents @city char(10)
AS
SELECT * FROM students WHERE city=@city
GO;

EXEC city_residents city=‘Durgapur’;
```

Stored Procedure with Parameters

- Example:

```
        CREATE PROCEDURE city_streamwise @city char(10),
@stream     char(15)
        AS
        SELECT * FROM students WHERE city=@city and
stream=@stream
        GO;


        EXEC city_streamwise city=‘Durgapur’, stream=‘Marketing’;
```

- Till we meet again in the next class……….