شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Database Design and programming

Tahaluf Training Center 2021

Day 6

1. **Databases and Security (Authorization Vs Authentication)**
2. SQL Injection
3. Data Integrity with Constraints

Database authentication is the process or act of confirming that a user who is attempting to log in to a database is authorized to do so, and is only accorded the rights to perform activities that he or she has been authorized to do.

- SQL Server supports two authentication modes:

1. Windows authentication is the default, and is often referred to as integrated security because this SQL Server security model is tightly integrated with Windows.

2. Mixed mode supports authentication both by Windows and by SQL Server. User name and password pairs are maintained within SQL Server.

Authorization

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Authorization is the process of giving necessary privileges to the user to access specific resources such as files, databases, locations, funds, files, information, almost anything within an application.

In simple terms, authorization evaluates a user's ability to access the system and up to what extent.

real-world example :

1. If you forgot your password, they may ask you some security questions that only you know, or they may email you a password reset token. This is authentication.

2. Once you have successfully logged in your user account, you can access your profile, download your bank statement, make transactions, and do many other banking-related activities. All of these activities are authorized. You are granted the privilege to perform them.

Permissions are the types of access granted to specific securable.

At the server level, permissions are assigned to SQL Server logins and server roles. At the database level, they are assigned to database users and database roles.

You have three main ways to control permissions:

Grant — The GRANT statement enables principals to access specified securables.

Deny — The DENY statement prevents principals from accessing specified securables.

Revoke — The REVOKE statement eliminates permissions that were previously granted for specific securables.

Permissions in SQL


```
GRANT SELECT, INSERT, UPDATE, DELETE ON
Person.Person TO Ayman;

GRANT ALL ON Person.Person  TO Ayman;

REVOKE DELETE ON Person.Person FROM Ayman;

REVOKE ALL ON Person.Person FROM Ayman;
```

Day 6

1. Databases and Security (Authorization Vs Authentication)
2. **SQL Injection**
3. Data Integrity with Constraints

SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.

It generally allows an attacker to view data that they are not normally able to retrieve.

SQL injection examples:

1. Retrieving hidden data, where you can modify an SQL query to return additional results.

2. Subverting application logic, where you can change a query to interfere with the application's logic.

3. UNION attacks, where you can retrieve data from different database tables.

4. Examining the database, where you can extract information about the version and structure of the database.

SQL injection usually occurs when you ask a user for input, like their username/userId, and instead of a name/id, the user gives you an SQL statement that you will unknowingly run on your database.

If there is nothing to prevent a user from entering "wrong" input, the user can enter some "smart" input .

UserId: 202 OR 1=1

Then, the SQL statement will look like this:

```
SELECT * FROM Users WHERE  EmployeeID = 202 OR 1=1;
```

```
SELECT  EmployeeID, Name,
Password FROM Employee WHERE EmployeeID = 202 or 1=1;
```

Day 6

1. Databases and Security (Authorization Vs Authentication)
2. SQL Injection
3. **Data Integrity with Constraints**

Integrity Constraints are used to apply business rules for the database tables.

The constraints available in SQL are Foreign Key, Not Null, Unique, Check.

Constraints can be defined in two ways:

1) The constraints can be specified immediately after the column definition. This is called column-level definition.

2) The constraints can be specified after all the columns are defined. This is called table-level definition.

1. SQL Primary key:

This constraint defines a column or combination of columns which uniquely identifies each row in the table.

Primary Key at column level:

```
CREATE TABLE TahalufEmp (
     ID      INTEGER NOT NULL PRIMARY KEY,
     name          CHAR (30),
     city          CHAR (20)
) ;
```

Primary Key at table level:

```
CREATE TABLE TahalufEmp2 (
     ID      INTEGER NOT NULL UNIQUE ,
     name          CHAR (30),
     city          CHAR (20)

) ;
ALTER TABLE TahalufEmp2 ADD PRIMARY KEY (ID);
```

2. SQL Foreign key or Referential Integrity :

This constraint identifies any column referencing the PRIMARY KEY in another table. It establishes a relationship between two columns in the same table or between different tables.

3. SQL Not Null Constraint :

This constraint ensures all rows in the table contain a definite value for the column which is specified as not null. Which means a null value is not allowed.

4. SQL Unique Key:

This constraint ensures that a column or a group of columns in each row have a distinct value. A column(s) can have a null value but the values cannot be duplicated.

5. SQL Check Constraint :

This constraint defines a business rule on a column. All the rows must satisfy this rule. The constraint can be applied for a single column or a group of columns.

```
gender char(1) CHECK (gender in ('M','F')),
```

On the E-Learning Portal