

## Capitolul 3 – Tehnologii folosite în realizarea aplicației

### 3.2. SQL Server

SQL (Structured Query Language) este un limbaj structura pentru interogarea bazelor de date relaționale. Modul de folosirea în aplicațiile de tip client/server presupune că aplicația client este cea care generează instrucțiunile SQL.

A fost lansat inițial de IBM, după care a fost standardizat pentru prima dată de ANSI, iar apoi ISO.

SQL operează asupra datelor normalizate, fiind un limbaj neprocedural. Conceptele necesare pentru lucrul cu acest limbaj sunt următoarele: tabelă, cheie primară, coloană, rând, index, bază de date relațională etc.

*Tabela sau relația* este un ansamblu format din  $n$  coloane și  $m$  rânduri (numite și tupluri/linii) care respectă următoarele condiții minime: valorile aflate la intersecția rândului cu coloane trebuie să fie de nivel elementar, liniile să nu se repete, iar descrierile coloanelor să nu fie repetitive.

*Coloana* tabelii este formată din valorile pe care le ia atributul respectiv în liniile tabelii.

*Rândul/tuplul* este format din valorile coloanelor ce se referă la o entitate a tabelii respective.

Accesul la date se face folosind indexarea, deoarece este rapidă. Un *index* poate fi reprezentat de o cheie pe una sau mai multe coloane. Această indexare se face automat, deoarece se pot adăuga sau șterge entități oricând, fără ca datele memorate să fie afectate.

*Cheia primară* este un atribut care nu permite valori duplicate, el este unic pe coloană, deci fiecare linie se identifică în mod unic.

*Baza de date relațională* este un ansamblu de tabele normalizate, grupate în jurul unui subiect, în principiu, bine definit. Într-o bază de date relațională, entitățile și legăturile sunt transpuse în tabele.

SQL Server este un sistem de gestionare a bazelor de date relaționale produs de către Microsoft. Principalele limbaje de interogare sunt MS-SQL și T-SQL. Este cel mai răspândit limbaj pentru bazele de date.

T-SQL (Transact- SQL) este o implementare a standardului SQL-92 (standard ISO

### Capitolul 3 – Tehnologii folosite în realizarea aplicației

pentru SQL), dar mai facilitează și unele extensii.

Principalele tipuri de dată din SQL Server sunt:

Tip de dată	Capacitate
bigint	8 octeți
int	4 octeți
smallint	2 octeți
tinyint	1 octet
decimal, cu precizie între 1-38 zecimale	5 - 17 octeți
bit	1 octet
date	3 octeți
datetime	8 octeți
varchar(lungime)	max $2^{31}-1$ octeți
varbinary	max $2^{31}-1$ octeți
image	max $2^{31}-1$ octeți

**Tabelul 3.1.** *Principalele tipuri de dată în SQL Server*

Pentru a crea o bază de date în SQL Server trebuie într-un query nou să scriem următoarea secvență de cod:

```
CREATE DATABASE <numele bazei de date>
```

Această secvență provoacă apariția unei baze de date noi.

Pentru a adăuga o nouă tabelă utilizăm următoarea secvență de cod:

```
CREATE TABLE <numele tabelii>
```

```
(  
    nume_coloană1 tip_de_dată,  
    nume_coloană2 tip_de_dată,  
    nume_coloană3 tip_de_dată,  
    .....  
)
```

### Capitolul 3 – Tehnologii folosite în realizarea aplicației

Unde *tip\_de\_dată* este un tip de dată din **Tabelul 3.1**, iar *nume\_coloană* este numele atributului.

Adăugarea unui index se face folosind comanda CREATE INDEX:

```
CREATE UNIQUE INDEX nume_index ON nume_tabelă (nume_coloană)
```

Este de preferat să adăugăm și proprietatea de unicitate a indexului deoarece el trebuie să existe o singură dată într-o tabelă pentru se evita eventualele anomalii care pot să apară. Acest index va permite căutarea rapidă a datelor fără a fi nevoie să citim toate datele din tabelă.

Tabela mai are nevoie și de o cheie primară, este foarte importantă pentru indentificarea unei entități într-o tabelă cu multe date. Adăugare cheii primare trebuie menționată în comanda de creare a tabelului. Astfel comanda pentru crearea unei tabeli ce conține și o cheie primară devine:

```
CREATE TABLE <numele tabelii>  
(  
    coloană_primară INT PRIMARY KEY IDENTITY  
    nume_coloană2 tip_de_dată,  
    nume_coloană3 tip_de_dată,  
    .....  
)
```

O cheie primară trebuie să fie de tip întreg și trebuie setată ca fiind auto-incrementabilă prin adăugarea declarației IDENTITY (aceasta asigură și unicitatea cheii), iar adăugarea de *PRIMARY KEY* specifică faptul că acest atribut este cheie primară.

Mai trebuie menționat și faptul că pentru a efectua operații de cuplare între tabele trebuie să creăm și un atribut numit *FOREIGN KEY*, cheie străină. Acest se realizează tot în declarația de creare a tabelului.

### Capitolul 3 – Tehnologii folosite în realizarea aplicației

```
CREATE TABLE <numele tabelii>
(
    coloană_primară INT PRIMARY KEY IDENTITY,
    nume_coloană2 tip_de_dată,
    nume_coloană3 tip_de_dată,
    cheie_străină INT FOREIGN KEY REFERENCES tabelă(cheie_primară)
    .....
)
```

Cheia străină este specificată printr-o referință către cheia primară a altei tabeli. Această dependență impune ca orice valoare a atributului care este cheia străină, să existe în tabela cu care se află în relație ca și cheia primară.

Vom exemplifica crearea unei baze de date *Gestiune* și a unei tabeli *Produse* ce are ca cheia primară un atribut numit *ID*.

```
CREATE DATABASE Gestiune
```

```
CREATE TABLE PRODUS
(
    ID INT PRIMARY KEY IDENTITY,
    NUME_PRODUS VARCHAR(50),
    PRET_PRODUS DECIMAL
)
```

Continuăm exemplul prin adăugarea unei noi tabeli STOC care va conține și un atribut de tip numeric ce va fi cheia străină cu referință către produs pentru a putea reține cantitatea din fiecare produs.

```
CREATE TABLE STOC
(
    ID INT NOT NULL PRIMARY KEY,
    CANTITATE DECIMAL,
    ID_PRODUS INT FOREIGN KEY REFERENCES PRODUS(ID)
)
```

### Capitolul 3 – Tehnologii folosite în realizarea aplicației

Pentru adăuarea, modificarea și ștergerea datelor dintr-o tabelă SQL Server ne pune la dispoziție *Procedurile Stocate*, acestea reprezintă bucăți de cod ce sunt salvate în baza de date, iar la dorința utilizatorului ele pot fi apelate. Procedurile stocate permit și parametrării pentru codul ce îl execută. Există mult prea faimoasele operații CRUD (Create, Read, Update și Delete), adică creare, citire, modificare și ștergere, care sunt necesare pentru fiecare tabelă, mai puțin operația de ștergere, care în general nu este recomandată, rolul bazei de date fiind să stocheze date.

Operația de creare într-o bază de date se face folosind declarația INSERT, această declarație inserează un rând nou în tabelă.

```
INSERT INTO nume_tabelă (colana1, colana2, ...) VALUES (valoare1, valoare2, ...)
```

Exemplu de creare pe tabela produs creată anterior:

```
INSERT INTO  
PRODUS(NUME_PRODUS, PRET_PRODUS)  
VALUES('Zahar', 10)
```

Trebuie să menționăm faptul că nu mai trebuie să precizăm o valoare pentru atributul ID, acesta având specificatorul *IDENTITY* are proprietatea ca la inserarea datelor se adaugă automat, valoarea nouă fiind valoare veche plus o unitate.

Exemplu de citire a datelor din tabela PRODUS:

```
SELECT ID, NUME_PRODUS,PRET_PRODUS FROM PRODUS  
sau  
SELECT * FROM PRODUS
```

Aceste două variante de citire a datelor dintr-o tabelă sunt relativ la fel, dar a doua variantă nu reflectă în mod clar datele care ne sunt aduse din tabelă.

Pentru exemplificarea operației de modificare într-o tabelă este bine să introducem și două concepte noi, cel de procedură stocată și cel de parametru. O procedură stocată se

### Capitolul 3 – Tehnologii folosite în realizarea aplicației

crează folosind declarația *CREATE PROCEDURE*, iar parametrii se specifică folosind semnul ”@”.

Exemplu de procedură stocată ce modifică un rând din tabela produs folosind ca parametru id-ul rândului ce trebuie modificat, numele nou al produsului și prețul nou al produsului.

```
CREATE PROCEDURE Modifica_produs
@id int,
@nume_nou varchar(50),
@pret_nou decimal
AS
BEGIN
UPDATE PRODUS SET
NUME_PRODUS=@nume_nou,
PRET_PRODUS=@pret_nou
WHERE ID=@id
END
```

Pentru ștergerea datelor dintr-o tabelă se poate face o procedură stocată, dar de data asta nu va mai fii nevoie decât de id-ul rândului pe care vrem să-l ștergem.

```
CREATE PROCEDURE Sterge_produs
@id int
AS
BEGIN
DELETE FROM PRODUS WHERE ID=@id
END
```

Declarațiile necesare pentru operațiile de tip CRUD sunt: *SELECT*, *INSERT*, *UPDATE* și *DELETE*.