

- Seminar 1 - Limbojui SQL

Crearea tabelului *Utilizatori* din baza de date *Problema Filme*

```
CREATE TABLE Utilizatori (  
    cod_u INT PRIMARY KEY IDENTITY,  
    nume_u VARCHAR(100) NOT NULL UNIQUE,  
    parola VARCHAR(100),  
    email VARCHAR(100)  
);
```

constrângere PRIMARY KEY
proprietate IDENTITY

denumirea
coloanei

tip de date

NVARCHAR = tip
de date care poate
stoca și caractere
uniocode

! Mai putem scrie constrângerile și astfel:

```
CONSTRAINT pk-Utilizatori PRIMARY KEY (cod-u)  
CONSTRAINT uq-nume-u UNIQUE (nume-u)
```

- ! OBS.** - fiecare tabel are un **identificator unic**, asigurăt prin def. constrângerii **PRIMARY KEY** (pe una/mai multe coloane)
- putem folosi **IDENTITY** pt. a genera **valori unice** în mod automat pt. oarecăr coloană, în cazul în care e **doar o coloană** de constrângere **PRIMARY KEY** !!
 - constrângerea **UNIQUE** asigură unicitatea valorilor la nivelul coloanei / combinației de coloane
 - putem avea mai multe constrângeri **UNIQUE** pe un tabel, dar doar o constrângere **PRIMARY KEY**
 - proprietatea **IDENTITY (seed, increment)**:
 - ° **seed** repr. valoarea de la care se începe
 - ° **increment** repr. val. care se adaugă de fiecare dată la imediată față de ultima înregistrare

Avem tabelul *Filme*

```
CREATE TABLE Filme (  
    cod_f INT PRIMARY KEY IDENTITY  
    titlu NVARCHAR(200)  
    durata TIME  
    limba NVARCHAR(100)  
    cod-c INT FOREIGN KEY REFERENCES Compozii (cod-c) ON UPDATE CASCADE  
    ON DELETE CASCADE  
);
```

constrângere FOREIGN KEY

! Mai putem scrie și așa constrângerea:

CONSTRAINT fk_Bomponii_Filme FOREIGN KEY (cod-c) REFERENCES Bomponii (cod-c)
ON UPDATE CASCADE ON DELETE CASCADE

! OBS. - nu se pot face modificări în tabelul care conține constrângerea **PRIMARY KEY** dacă sunt modificări distrugătoare pe date din tabelul care conține constrângerea **FOREIGN KEY**
- la crearea unei constrângeri **FOREIGN KEY** se pot specifica acțiuni care să aibă loc în cazul operațiilor de **modificare / ștergere**, a valorilor din coloana pe care este def. constr. **PRIMARY KEY** din celălalt tabel.

- NO ACTION (default)
- CASCADE
- SET NULL
- SET DEFAULT

Avem tabelul de legătură dintre Utilizatori și Filme, numit **Note**, pentru relația **M-M**.

```
CREATE TABLE Note(  
  cod-u INT FOREIGN KEY REFERENCES Utilizatori (cod-u),  
  cod-f INT FOREIGN KEY REFERENCES Filme (cod-f),  
  nota INT,  
  CONSTRAINT pk_note PRIMARY KEY (cod-u, cod-f)  
);
```

! Dacă am omis specificarea unei constrângeri **UNIQUE** la crearea tabelului, aceasta se poate crea ulterior cu ajutorul instrucțiunii **ALTER TABLE**:

```
ALTER TABLE Utilizatori  
ADD CONSTRAINT uq_email UNIQUE (email);
```

Și ca să punem constrângerea notei de a fi între 1 și 10, scriem așa:

```
ALTER TABLE Note  
ADD CONSTRAINT ch_note CHECK (nota >= 1 AND nota <= 10);
```

- asigură dacă în tabel există valori care nu respectă constrângerea !

Și să adăugăm o coloană nouă scriem

```
ALTER TABLE Note  
ADD data-aj-ora-adăugării DATETIME;
```

Ca să setăm o valoare implicită pt. o coloană, folosim DEFAULT.

```
ALTER TABLE Note  
ADD CONSTRAINT df_data_si_oro_colegii DEFAULT GETDATE() FOR data_si_oro_colegii
```

data si ora curentă
la momentul adăugării înre-
gistrării în tabel

Dacă dorim să modificăm numele bazei de date, folosim ALTER DATABASE

```
ALTER DATABASE ProblemeFilme  
MODIFY Name = NoteFilme;
```