

Perfect 👍

Vă formulez **exact** ca într-o lucrare practică pentru elevi:

● LUCRARE PRACTICĂ

Tema: Crearea și utilizarea indexurilor (Clustered și Nonclustered)

◆ SARCINA 1

Crearea unui CLUSTERED INDEX pe coloana Check_in_data

✚ Cerință pentru elevi:

1. Creați tabelul Rezervari.
 2. Creați un **CLUSTERED INDEX** pe coloana Check_in_data.
 3. Introduceți 5 înregistrări.
 4. Afișați rezervările din martie 2026.
-

☑ Rezolvare

1 Crearea tabelului

```
CREATE TABLE Rezervari (  
    RezervatieId INT,  
    Check_in_data DATE,  
    Check_out_data DATE,  
    CameraId INT  
);
```

2 Crearea Clustered Index

```
CREATE CLUSTERED INDEX IX_Rezervari_CheckIn  
ON Rezervari(Check_in_data);
```

✓ Datele se vor ordona fizic după Check_in_data.

3 Inserare date

```
INSERT INTO Rezervari VALUES
(1, '2026-03-01', '2026-03-05', 2),
(2, '2026-02-15', '2026-02-20', 1),
(3, '2026-03-10', '2026-03-15', 3),
(4, '2026-01-05', '2026-01-08', 4),
(5, '2026-03-20', '2026-03-25', 2);
```

4 Interogare

```
SELECT * FROM Rezervari
WHERE Check_in_data BETWEEN '2026-03-01' AND '2026-03-31';
```

◆ SARCINA 2

Observarea Clustered Index creat automat prin PRIMARY KEY

✚ Cerință pentru elevi:

1. Creați tabelul Camere cu CameraId PRIMARY KEY.
 2. Introduceți 4 camere.
 3. Verificați ce tip de index a fost creat automat.
-

☑ Rezolvare

```
CREATE TABLE Camere (
    CameraId INT PRIMARY KEY,
    Tipul VARCHAR(50),
    Etajul INT,
    Disponibilitatea VARCHAR(20)
);
```

- ✓ SQL Server creează automat **Clustered Index** pe CameraId.
-

Inserare date:

```
INSERT INTO Camere VALUES
(1, 'Single', 1, 'Disponibila'),
(2, 'Dubla', 1, 'Ocupata'),
(3, 'Suite', 2, 'Disponibila'),
(4, 'Single', 3, 'Ocupata');
```

Test:

```
SELECT * FROM Camere WHERE CameraId = 3;
```

◆ SARCINA 3

Crearea unui NONCLUSTERED INDEX pe Nume

✚ Cerință pentru elevi:

1. Creați tabelul `Cienti`.
 2. Introduceți 5 clienți.
 3. Creați un NONCLUSTERED INDEX pe coloana `Nume`.
 4. Căutați un client după nume.
-

☑ Rezolvare

1 Creare tabel

```
CREATE TABLE Cienti (  
    ClientId INT PRIMARY KEY,  
    Nume VARCHAR(50),  
    Prenume VARCHAR(50),  
    Email VARCHAR(100),  
    Telefon VARCHAR(20),  
    RezervatieId INT  
);
```

2 Inserare date

```
INSERT INTO Cienti VALUES  
(1, 'Popescu', 'Ion', 'ion@mail.com', '069111111', 1),  
(2, 'Ionescu', 'Ana', 'ana@mail.com', '069222222', 2),  
(3, 'Popescu', 'Maria', 'maria@mail.com', '069333333', 3),  
(4, 'Rusu', 'Dan', 'dan@mail.com', '069444444', 4),  
(5, 'Munteanu', 'Elena', 'elena@mail.com', '069555555', 5);
```

3 Creare Nonclustered Index

```
CREATE NONCLUSTERED INDEX IX_Cienti_Nume  
ON Cienti(Nume);
```

- ✓ Datele NU se reordonează fizic.
 - ✓ Se creează o structură separată pentru căutare rapidă.
-

4 Interogare

```
SELECT * FROM Clienti  
WHERE Nume = 'Popescu';
```

◆ SARCINA 4

Crearea unui index compus

📌 Cerință pentru elevi:

1. Creați un NONCLUSTERED INDEX pe coloanele:
 - Email
 - Telefon
 2. Căutați clientul folosind ambele coloane.
-

☑ Rezolvare

```
CREATE NONCLUSTERED INDEX IX_Clienti_Email_Telefon  
ON Clienti (Email, Telefon);
```

Test:

```
SELECT * FROM Clienti  
WHERE Email = 'ana@mail.com'  
AND Telefon = '069222222';
```

● Concluzie pentru elevi

- ✓ CLUSTERED → ordonează fizic tabelul
 - ✓ NONCLUSTERED → creează o structură separată
 - ✓ Un tabel poate avea:
 - 1 Clustered Index
 - mai multe Nonclustered Index
-

💡 Recomandarea mea didactică:

Rulați fiecare SELECT cu **Actual Execution Plan activat (Ctrl + M)** pentru a vedea diferența reală.

Dacă doriți, pot formula și o variantă de evaluare practică cu punctaj.

Perfect 👍

Formulez **doar condiția clară pentru elevi + rezolvarea directă**, fără creare tabele sau inserări.



EXERCITII – INDEXURI

◆ SARCINA 1 – CLUSTERED INDEX

📌 Condiție:

Optimizați căutarea rezervărilor după data de check-in.

Creați un **CLUSTERED INDEX** pe coloana `Check_in_data` din tabelul `Rezervari`.

☑ Rezolvare:

```
CREATE CLUSTERED INDEX IX_Rezervari_CheckIn  
ON Rezervari (Check_in_data);
```

- ✓ Datele din tabel vor fi ordonate fizic după `Check_in_data`.
 - ✓ Interogările pe interval de dată vor fi mai rapide.
-

◆ SARCINA 2 – NONCLUSTERED INDEX

📌 Condiție:

Optimizați căutarea clienților după nume.

Creați un **NONCLUSTERED INDEX** pe coloana `Nume` din tabelul `Cienti`.

☑ Rezolvare:

```
CREATE NONCLUSTERED INDEX IX_Cienti_Nume  
ON Cienti (Nume);
```

- ✓ Datele NU se reordonează fizic.
 - ✓ Se creează o structură separată pentru căutare rapidă.
-

◆ SARCINA 3 – INDEX COMPUS

✚ Condiție:

Optimizați căutarea clienților folosind simultan Email și Telefon.
Creați un index compus pe aceste două coloane.

☑ Rezolvare:

```
CREATE NONCLUSTERED INDEX IX_Clienti_Email_Telefon  
ON Clienti (Email, Telefon);
```

- ✓ Indexul este eficient când ambele coloane sunt folosite în condiția WHERE.
 - ✓ Ordinea coloanelor este importantă.
-

◆ SARCINA 4 – INDEX PE COLOANĂ FRECVENȚĂ ÎN FILTRARE

✚ Condiție:

Optimizați afișarea camerelor disponibile.
Creați un index pe coloana Disponibilitatea din tabelul Camere.

☑ Rezolvare:

```
CREATE NONCLUSTERED INDEX IX_Camere_Disponibilitate  
ON Camere (Disponibilitatea);
```

- ✓ Recomandat pentru coloane folosite frecvent în WHERE.
-

◆ SARCINA 5 – Ștergerea unui index

✚ Condiție:

Ștergeți indexul creat pentru Disponibilitatea.

☑ Rezolvare:

```
DROP INDEX IX_Camere_Disponibilitate  
ON Camere;
```

Concluzie simplă pentru elevi

- CLUSTERED → modifică ordinea fizică
 - NONCLUSTERED → nu modifică ordinea fizică
 - 1 singur clustered per tabel
 - Mai multe nonclustered permise
-

Dacă doriți, pot formula și 2–3 sarcini mai complexe (nivel avansat, cu JOIN și index pe chei străine).