

## Raport Proiect

Traistaru Alexandru Mihai

334AB

### Firma de service pentru dispozitive electronice

Aceasta aplicatie gestioneaza activitatea unei firme de service pentru dispozitive electronice. Un client poate face o comanda, specificand dispozitivul si problema acestuia. Comanda poate include mai multe tipuri de servicii (ex: inlocuire componente, curatare produs etc.). Clientii pot vedea si statusul comenzii (In derulare/ Asteptare pise / Finalizat). Dupa finalizarea serviciilor clientului i se face o factura.

- Un client poate aduce unul sau mai multe dispozitive la service si poate efectua una sau mai multe comenzi
- Pentru fiecare dispozitiv se pot efectua o comanda
- O comanda poate include mai multe piese de schimb, dar si un tip de piesa poate fi folosit in mai multe comenzi
- Fiecare comanda este atribuita unui angajat, un angajat poate avea mai multe comenzi atribuite
- Fiecare angajat face parte dintr un departament
- O comanda poate include mai multe servicii (in cazul in care un produs are mai multe defecte) dar si un serviciu poate fi prezent in mai multe comenzi
- Unei comenzi i se face o factura la final

#### Angajati (profilul unui angajat):

```
AngajatID (PK, int, not null)
DepartamentID (FK, int, not null)
Nume (nvarchar(50), null)
Prenume (nvarchar(50), null)
Email (varchar(100), null)
UserID (FK, int, null)
```

### Cienti (profilul unui client):

- ClientID (PK, int, not null)
- Nume (nvarchar(50), null)
- Prenume (nvarchar(50), null)
- Telefon (varchar(15), null)
- Email (varchar(100), null)
- CNP (nchar(13), null)
- Strada (nvarchar(50), null)
- Numar (char(10), null)
- Oras (nvarchar(50), null)
- Judet (nvarchar(50), null)
- UserID (FK, int, null)

### ComandaPiese (tabel de legatura comenzi-piese + cantitate folosita):

- ComandaID (PK, FK, int, not null)
- PiesaID (PK, FK, int, not null)
- Cantitate (int, null)

### ComandaServicii (tabel de legatura comenzi-servicii):

- ComandaID (PK, FK, int, not null)
- ServiciuID (PK, FK, int, not null)
- DataExecutie (date, not null)

### Comenzi (comenzile efective):

- ComandaID (PK, int, not null)
- ClientID (FK, int, not null)
- DispozitivID (FK, int, not null)
- AngajatID (FK, int, not null)
- DataPrimire (date, not null)
- DefectDispozitiv (nvarchar(max), null)
- Status (nvarchar(50), null)
- MetodaDePlata (nvarchar(20), not null)

### Departament (catalog pentru categorii):

- DepartamentID (PK, int, not null)
- Denumire (nvarchar(50), not null)

### Dispozitive (dispozitivele aduse de clienti):

- DispozitivID (PK, int, not null)
- ClientID (FK, int, not null)
- TipDispozitiv (nvarchar(50), not null)
- Producator (nvarchar(50), null)
- Model (nvarchar(50), null)
- Serie (varchar(50), null)
- DepartamentID (FK, int, not null)

### Facturi (factura emisa pentru o comanda):

- FacturalID (PK, int, not null)
- ComandaID (FK, int, null)
- Serie (varchar(100), null)
- Numar (int, null)
- DataEmitere (date, null)
- MetodaDePlata (nvarchar(50), null)
- PretFactura (decimal(10,2), not null)






### Piese (catalog piese + stoc):

- PiesaID (PK, int, not null)
- Denumire (nvarchar(100), null)
- Cod (varchar(50), null)
- Pret (decimal(10,2), not null)
- Furnizor (nvarchar(100), null)
- Stoc (int, null)

### Servicii (catalog de servicii):

- ServiciulID (PK, int, not null)
- Denumire (nvarchar(100), null)
- PretStandard (decimal(10,2), not null)

### Utilizatori (retine conturile de autentificare):

|   |   |
|---|---|
|  | UserID (PK, int, not null)              |
|  | NumeUtilizator (varchar(100), not null) |
|  | Parola (varchar(255), not null)         |
|  | Email (varchar(255), null)              |
|  | Rol (varchar(20), not null)             |

### Relatii:

Clienti – Dispozitive : 1:N

Clienti – Comenzi : 1:N

Dispozitive – Comenzi : 1:1

Angajati – Comenzi : 1:N

Departament – Angajati : 1:N

Comenzi – Servicii : N:N

Comenzi – Piese : N:N

Comenzi – Facturi : 1:1

Utilizatori – Clienti : 1:1

Utilizatori – Angajati : 1:1

|              | Clienti | Dispozitive | Departamente | Angajati | Comenzi | Servicii | Piese | Facturi | Utilizatori |
|--------------|---------|-------------|--------------|----------|---------|----------|-------|---------|-------------|
| Clienti      |         | 1:N         |              |          | 1:N     |          |       |         | 1:1         |
| Dispozitive  | 1:N     |             |              |          |         |          |       |         |             |
| Departamente |         |             |              | 1:N      |         |          |       |         |             |
| Angajati     |         |             | 1:N          |          | 1:N     |          |       |         | 1:1         |
| Comenzi      | 1:N     | 1:1         |              | 1:N      |         | N:N      | N:N   | 1:1     |             |
| Servicii     |         |             |              |          | N:N     |          |       |         |             |
| Piese        |         |             |              |          | N:N     |          |       |         |             |
| Facturi      |         |             |              |          | 1:1     |          |       |         |             |
| Utilizatori  | 1:1     |             |              | 1:1      |         |          |       |         |             |

### ComenziServicii

O comanda poate avea mai multe servicii executate, iar acelasi serviciu poate fi executat in multe comenzi.

De asta am tabela de legatura ComandaServicii. In plus, aici am si un atribut care tine de relatie, nu de o singura entitate: DataExecutie (cand s-a executat serviciul pentru comanda respectiva).

### ComenziPiese

O comanda poate folosi mai multe piese, iar o piesa poate fi folosita in mai multe comenzi.

De asta am tabela de legatura ComandaPiese. Si aici am si un atribut de relatie: Cantitate (cate bucati din piesa au fost folosite la comanda respectiva)

### Functionalitati:

#### 1) autentificare + roluri

- utilizatorul se poate inregistra cu: nume utilizator, parola, email (optional), rol (CLIENT / ANGAJAT)
  - parola (minim 4 caractere dintre care o cifra) si email ul se verifica sa fie valide
- utilizatorul se poate loga pe baza numelui de utilizator si parolei
- aplicatia separa accesul pe roluri:
  - rutele de client sunt accesibile doar de client
  - rutele de angajat sunt accesibile doar de angajat
- dupa login, utilizatorul este redirectionat automat la panoul corect (client/angajat)

#### 2) profil client (date personale)

- clientul isi poate completa/modifica profilul: nume, prenume, telefon (obligatorii), plus email/cnp/adresa (optionale)
- la salvare se face validare:
  - nume/prenume/telefon nu pot fi goale

- email (daca e completat) trebuie sa fie valid simplu
- cnp (daca e completat) trebuie sa aiba 13 cifre
- profilul se salveaza in tabela Clienti (update sau insert daca lipseste)

### 3) creare comanda (client)

- clientul poate crea o comanda doar daca are profil minim complet (nume + prenume + telefon)
- la creare comanda, clientul completeaza:
  - departament (categorie)
  - date dispozitiv (tip, producator, model, serie)
  - defect
  - metoda de plata
- aplicatia:
  - insereaza dispozitivul in Dispozitive
  - alege automat angajatul "cel mai disponibil" din departamentul ales
  - insereaza comanda in Comenzi cu status "Trimisa"

### 4) vizualizare comenzi (client)

- clientul poate vedea lista "comenzile mele"
- lista include date din mai multe tabele: date comanda + date dispozitiv + departament

### 5) editare si stergere comenzi (client)

- clientul poate edita sau sterge doar comenzile care sunt inca "Trimisa"
- daca statusul nu mai este "Trimisa", operatia este blocata

### 6) procesare comanda (angajat)

- angajatul vede:
  - "comenzile mele" (doar cele asignate lui)
  - "toate comenzile" (lista completa)
- angajatul poate:

- porni lucrul: schimba status "Trimisa" -> "In derulare"
- salva servicii executate pe comanda (in ComandaServicii, cu data executiei)
- adauga piese folosite (in ComandaPiese, cu cantitate) si scade stocul din Piese
- genera factura (in Facturi) si finalizeaza comanda ("Finalizata")
- angajatul nu poate modifica comenzi care nu ii apartin (se verifica pe AngajatID)

#### 7) facturi (client)

- clientul poate vedea lista "facturile mele"
- lista facturilor este bazata pe join Facturi + Comenzi + Dispozitive si este filtrata pe clientul logat

#### 8) gestiune catalog servicii (crud)

- angajatul poate adauga / edita / sterge servicii (denumire, pret)
- stergerea este blocata daca serviciul exista deja in ComandaServicii

#### 9) gestiune stoc piese (crud)

- angajatul poate adauga / edita / sterge piese (denumire, cod, pret, furnizor, stoc)
- stergerea este blocata daca piesa exista deja in ComandaPiese

#### 10) rapoarte (interogari complexe ale bazei de date)

- interfata pentru rapoarte cu parametri (ex: interval de date, prag stoc etc.)
- rapoarte generate cu subcereri:
  - angajat cel mai disponibil (pe departament)
  - comenzi fara factura pana la o data
  - top clienti in interval dupa total facturat
  - piese sub prag + consum in ultimele N zile

## 11) grafic

- pagina dedicata unde se vizualizeaza un grafic cu top client si facturile lor (pretul lor)

### Interogari simple:

1. Interogarea ia toate comenzile unui client si le afiseaza impreuna cu informatiile despre dispozitivul din comanda si departamentul din care face parte dispozitivul. Rezultatul este ordonat descrescator dupa data primirii si dupa id-ul comenzii. Parametru variabil: clientId

```
SELECT
    c.ComandaID,
    c.DataPrimire,
    c.DefectDispozitiv,
    c.Status,
    c.MetodaDePlata,
    d.DispozitivID,
    d.TipDispozitiv,
    d.Producator,
    d.Model,
    d.Serie,
    d.DepartamentID,
    dep.Denumire AS DepartamentDenumire
FROM Comenzi c
INNER JOIN Dispozitive d ON d.DispozitivID = c.DispozitivID
INNER JOIN Departament dep ON dep.DepartamentID =
d.DepartamentID
WHERE c.ClientID = ?
ORDER BY c.DataPrimire DESC, c.ComandaID DESC
```



2. Interogarea cauta o singura comanda, dar doar daca apartine clientului logat. Intoarce detaliile comenzii impreuna cu datele dispozitivului si departamentul dispozitivului. Este folosita cand clientul vrea sa editeze comanda. Parametri variabili: clientId, comandaId.

```
SELECT
    c.ComandaID,
    c.DataPrimire,
    c.DefectDispozitiv,
    c.Status,
    c.MetodaDePlata,
    d.DispozitivID,
    d.TipDispozitiv,
    d.Producator,
    d.Model,
    d.Serie,
    d.DepartamentID,
    dep.Denumire AS DepartamentDenumire
FROM Comenzi c
INNER JOIN Dispozitive d ON d.DispozitivID = c.DispozitivID
INNER JOIN Departament dep ON dep.DepartamentID =
d.DepartamentID
WHERE c.ClientID = ?
    AND c.ComandaID = ?
```

3. Interogarea ia toate comenzile asignate unui anumit angajat si le afiseaza cu detalii complete: date comanda, date client, date dispozitiv, departament si date angajat. Practic e lista “comenzile mele” pentru angajat. Parametru variabil: angajatId.

```
SELECT
    c.ComandaID, c.DataPrimire, c.DefectDispozitiv, c.Status,
    c.MetodaDePlata,
    cl.ClientID, cl.Nume AS ClientNume, cl.Prenume AS ClientPrenume,
    d.DispozitivID, d.TipDispozitiv, d.Producator, d.Model, d.Serie,
    dep.DepartamentID, dep.Denumire AS DepartamentDenumire,
```

```

a.AngajatID, a.Nume AS AngajatNume, a.Prenume AS AngajatPrenume
FROM Comenzi c
INNER JOIN Clienti cl ON cl.ClientID = c.ClientID
INNER JOIN Dispozitive d ON d.DispozitivID = c.DispozitivID
INNER JOIN Departament dep ON dep.DepartamentID =
d.DepartamentID
INNER JOIN Angajati a ON a.AngajatID = c.AngajatID

```

4. Interogarea afiseaza lista de servicii care au fost salvate ca executate pentru o comanda, adica serviciile din tabela de legatura dintre comenzi si servicii. Intoarce denumirea serviciului si pretul standard. Parametru variabil: comandaId.

```

SELECT s.ServiciuID, s.Denumire, s.PretStandard
FROM ComandaServicii cs
INNER JOIN Servicii s ON s.ServiciuID = cs.ServiciuID
WHERE cs.ComandaID = ?
ORDER BY s.Denumire ASC

```

5. Interogarea afiseaza piesele folosite la o comanda, adica piesele din tabela de legatura dintre comenzi si piese, impreuna cu cantitatea folosita. Rezultatul este afisat ca text de tipul "Denumire xCantitate". Parametru variabil: comandaId.

```

SELECT p.Denumire + ' x' + CAST(cp.Cantitate AS varchar(20)) AS Linie
FROM ComandaPiese cp
INNER JOIN Piese p ON p.PiesaID = cp.PiesaID
WHERE cp.ComandaID = ?
ORDER BY p.Denumire ASC

```

6. Interogarea afiseaza toate facturile emise pentru comenzile unui client. Pentru fiecare factura arata seria, numarul, data, pretul, metoda de plata, plus informatii despre dispozitivul din comanda facturata. Parametru variabil: clientId.

```
SELECT
    f.Serie, f.Numar, f.DataEmitere, f.PretFactura, f.MetodaDePlata,
    c.ComandaID,
    d.TipDispozitiv, d.Producator, d.Model, d.Serie AS SerieDispozitiv
FROM Facturi f
INNER JOIN Comenzi c ON c.ComandaID = f.ComandaID
INNER JOIN Dispozitive d ON d.DispozitivID = c.DispozitivID
WHERE c.ClientID = ?
ORDER BY f.DataEmitere DESC, f.FacturaID DESC
```

Interogari complexe:

1. Interogarea gaseste angajatul (sau angajatii) dintr-un departament care are cele mai putine comenzi active, adica in status "Trimisa" sau "In derulare". Foloseste subcereri ca sa numere comenzile active pentru fiecare angajat si ca sa compare cu minimul din departament. Parametru variabil: departamentId.

```
SELECT
    a.Nume,
    a.Prenume,
    d.Denumire AS Departament,
    (SELECT COUNT(*)
     FROM Comenzi c
     WHERE c.AngajatID = a.AngajatID
     AND c.Status IN (N'Trimisa', N'In derulare')) AS ComenziActive
```

```

FROM Angajati a
INNER JOIN Departament d ON d.DepartmentID = a.DepartmentID
WHERE a.DepartmentID = ?
AND (SELECT COUNT(*)
      FROM Comenzi c
      WHERE c.AngajatID = a.AngajatID
      AND c.Status IN (N'Trimisa', N'In derulare'))
=
(SELECT MIN(t.cnt)
 FROM (
      SELECT
        (SELECT COUNT(*)
         FROM Comenzi c2
         WHERE c2.AngajatID = a2.AngajatID
         AND c2.Status IN (N'Trimisa', N'In derulare')) AS cnt
      FROM Angajati a2
      WHERE a2.DepartmentID = ?
    ) t)
ORDER BY a.Nume ASC, a.Prenume ASC

```

2. Interogarea afiseaza comenzile primite pana la o data limita care nu au inca factura emisa. Verifica lipsa facturii printr-o subcerere de tip NOT EXISTS in tabela Facturi. Parametru variabil: dataLimita

```

SELECT
  c.DataPrimire,
  COALESCE(cl.Nume, N'') + N' ' + COALESCE(cl.Prenume, N'') AS

```

```

Client,
dep.Denumire AS Departament,
d.TipDispozitiv,
c.Status
FROM Comenzi c
INNER JOIN Clienti cl ON cl.ClientID = c.ClientID
INNER JOIN Dispozitive d ON d.DispozitivID = c.DispozitivID
INNER JOIN Departament dep ON dep.DepartamentID =
d.DepartamentID
WHERE c.DataPrimire <= ?
AND NOT EXISTS (
    SELECT 1
    FROM Facturi f
    WHERE f.ComandaID = c.ComandaID
)
ORDER BY c.DataPrimire ASC

```

3. Interogarea afiseaza clientii care au totalul facturat peste un prag minim intr-un interval de timp. Pentru fiecare client calculeaza numarul de facturi si suma totala folosind subcereri cu COUNT si SUM, apoi filtreaza clientii care trec pragul si ii ordoneaza descrescator dupa total. Parametri variabili: dataStart, dataEnd, pragMinim.

```

SELECT
cl.Nume,
cl.Prenume,
ISNULL((
    SELECT COUNT(*)
    FROM Facturi f
    INNER JOIN Comenzi c ON c.ComandaID = f.ComandaID
    WHERE c.ClientID = cl.ClientID
    AND f.DataEmitere BETWEEN ? AND ?
), 0) AS NrFacturi,
ISNULL((
    SELECT SUM(f.PretFactura)

```

```

FROM Facturi f
INNER JOIN Comenzi c ON c.ComandaID = f.ComandaID
WHERE c.ClientID = cl.ClientID
AND f.DataEmitere BETWEEN ? AND ?
), 0) AS Total
FROM Clienti cl
WHERE ISNULL((
SELECT SUM(f.PretFactura)
FROM Facturi f
INNER JOIN Comenzi c ON c.ComandaID = f.ComandaID
WHERE c.ClientID = cl.ClientID
AND f.DataEmitere BETWEEN ? AND ?
), 0) >= ?
ORDER BY Total DESC, NrFacturi DESC, cl.Nume ASC, cl.Prenume
ASC

```

4. Interogarea afiseaza piesele care au stoc sub sau egal cu un prag si calculeaza cat s-a consumat din ele in ultimele N zile. Consum inseamna suma cantitatilor din ComandaPiese pentru comenzile din perioada respectiva, calculata printr-o subcerere. Parametri variabili: pragStoc, zile.

```

SELECT
p.Denumire,
p.Stoc,
ISNULL((
SELECT SUM(cp.Cantitate)
FROM ComandaPiese cp
INNER JOIN Comenzi c ON c.ComandaID = cp.ComandaID
WHERE cp.PiesaID = p.PiesaID
AND c.DataPrimire >= DATEADD(day, -?, CAST(GETDATE() AS
date))
), 0) AS FolositeUltimeleZile
FROM Piese p
WHERE p.Stoc <= ?
ORDER BY p.Stoc ASC, FolositeUltimeleZile DESC, p.Denumire ASC

```