

- În acest rezumat se va prezenta modalitatea de populare a entităților bazată pe feed-ul utilizatorului, însemnând că acesta va alege departamentul a cărui entitate trebuie populată cu date. Crearea sa a fost făcută în etapele anterioare, dar și crearea va fi condiționată de utilizator în produsul final pentru a face parcurge totul organizat și pentru a nu încărcă memoria inutil, lucru care poate duce la închiderea neașteptată a programului, dar și a calculatorului pe care operează aplicația dacă anumite limite privind memoria nu sunt respectate. În stadiul în care este rularea programului durează aproximativ 30 de secunde, iar asta înseamnă folosirea activă a tuturor metodelor de interes major implementate. De asemenea, instanțele și, în general, instanțierea de obiecte de tip Excel trebuie administrată corect, există riscul ca după închiderea programului să mai rămână instanțe de Excel deschise în background.
- Inserția datelor în entități este făcută cu ajutorul unui obiect Schedule.cs, folosirea acestuia are rolul de-a menține o lizibilitate a programului, dar și o bună coeziune a datelor deoarece toate acestea vor ține de obiectul respectiv, adică vor fi proprietăți, dar mai important vor fi grupate. Declarația de primitive putea și ea să constituie o soluție, însă am optat pentru varianta cu inițializare de obiect. În fapt, va fi o singură instanță care-și va schimba proprietățile pentru a introduce date în entitatea aleasă de utilizator. Avem posibilitatea de a folosi "setteri" sau constructorul obiectului pentru salvarea temporară a datelor.
- Câteva observații privitoare la sheet-ul orarului:
- Semigrupa A este întotdeauna pe linie cu numele/codul grupei, iar aceasta înseamnă că semigrupa B este pe linie cu o celulă nulă.
- Când găsim o celulă nenulă, îi extragem datele incluse, rămânând să găsim și ziua, locul și semigrupa.
- Fiecare grupă are negreșit două semigrupe, A și B, ceea ce înseamnă că putem folosi o structură fixă pentru a le găsi. Asta înseamnă că atunci când găsim pe A, trebuie să căutăm și cealaltă jumătate, pe B.
- Prelucrarea celulei se face în felul următor:

```
0 references
internal string[] CellSplit(string str)
{
    // e. g.: Graf., L, NII8, Musuroi_C_L
    string[] arr = { str.Split(',')[0].Trim(), str.Split(',')[1].Trim(),
        str.Split(',')[2].Trim(), str.Split(',')[3].Trim() };
    return arr;
}
//
1 reference
```

```

if (dep_i != 0)
{
    schedule.setTab(dep);
    for (i = ones_nones[dep_i]; i <= ones_nones[dep_i + 1] - 1; dep_i++) // dep_i
    {
        c1++;
        for (j = 4; j <= 38; j++)
        {
            c2++; // c2 will be from 1 to 35
            // another value for dep different from the previous one
            dep = excel.ReadCell(i, j);
            // here we take each relevant cell and insert it into the table
            if (dep != "Null cell!")
            {
                // this is what is in a cell
                schedule.setAbvr(CellSplit(dep)[0]); // course, e.g. PCLP2
                schedule.setType(CellSplit(dep)[1]); // type, e.g. L or C
                schedule.setPlace(CellSplit(dep)[2]); // e.g. NII8
                schedule.setProf(CellSplit(dep)[3]); // e.g. Musuroi_C_L
                //excel.ReadCell(i, j);
            }
        }
        c2 = 0;
    }
}
else

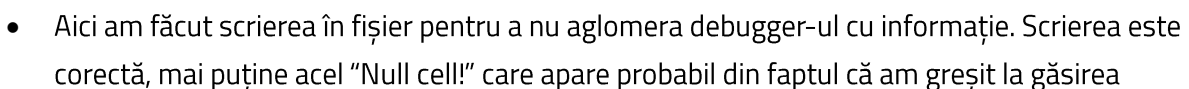
```

- Pentru a putea ajunge la ziua în care se desfășoară activitatea trebuie să scădem o linie și mai multe linii după caz. Următoarea imagine demonstrează acest aspect:

```

// also the line is always i - 1, so one line back
switch (time)
{
    case "8,00-9,50":
    {
        return 0;
    }
    case "10,00-11,50":
    {
        return 1;
    }
    case "12,00-13,50":
    {
        return 2;
    }
    case "14,00-15,50":
    {
        return 3;
    }
    case "16,00-17,50":
    {
        return 4;
    }
    case "18,00-19,50":
    {
        return 5;
    }
    case "20,00-21,50":
    {
        return 6;
    }
    default:
    {
        return -1;
    }
}

```



semigrupului/grupului. După această metodă prin apelarea unei metode de introducere în tabel, se va popula tabelul cu date.

- Din cauza faptului că apar prea multe instanțe ale aceluiași obiect Excel, trebuie conceput un sistem de verificare a inițializării, cel mai probabil voi folosi o variabilă globală pe baza căreia să se verifice dacă obiectul a fost instanțiat sau nu. O voi implementa cu ajutorul unei metode pentru a încapsula tot acest conținut repetitiv.

```

8 references
void ExcelInstantiation(string path, int sheet)
{
    if (!excel_inst)
    {
        excel = new Excel(path, sheet);
        excel_inst = true;
    }
}
8 references
void ExcelDeinstantiation()
{
    if (excel_inst)
    {
        excel.Close();
        excel_inst = false;
    }
}
//
1 reference

```

- O problemă întâlnită este faptul că atunci când programul caută o șir anume, căutarea se parcurge la fiecare iterație, deci această căutare trebuie făcută anterior pentru a elimina efectul său negativ.
- Rularea a funcționat, în final am reușit să scriu într-un fișier tot ce ține de departamentul ales. Următoarele imagini vor ilustra cum atributele obiectului ajung în tabelul corespondent departamentului.

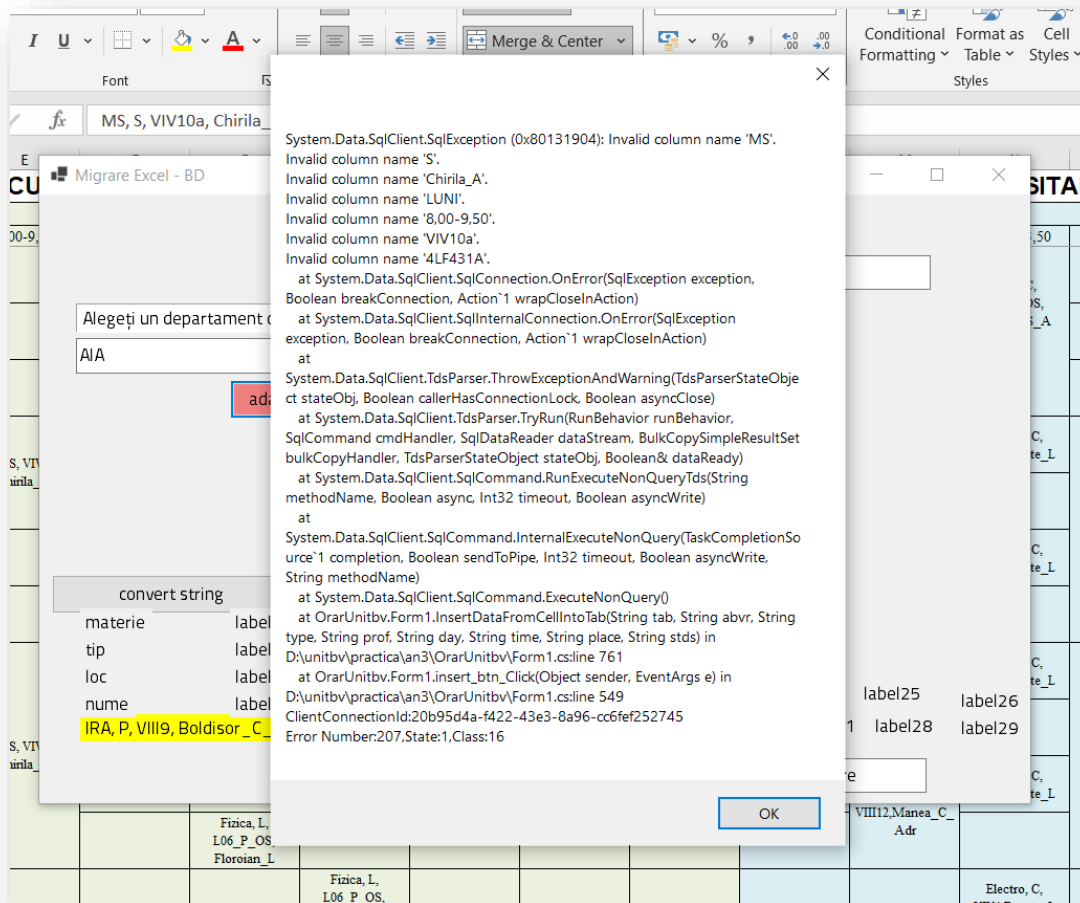
```

        schedule.setStds(sgr);
    }
    // here we must try to insert data into the table
    InsertDataFromCellIntoTab(
        schedule.getTab(),
        schedule.getAbvr(),
        schedule.getType(),
        schedule.getProf(),
        schedule.getDay(),
        schedule.getTime(),
        schedule.getPlace(),
        schedule.getStds());
}

```

```
internal string InsertDataFromCellIntoTab(string tab, string abvr, string type, string prof, string
{
    // con.Open();
    // SqlCommand cmd = new SqlCommand("insert intol", con);
    // con.Close();
    // e.g. Graf., L, NII8, Musuroi_C_L
    // using CellSplit(str)[] method
    con.Open();
    SqlCommand insert_cmd = new SqlCommand("'" +
        $"insert into [{tab}]([Materie abreviere], Tip, [Cadru didactic], " +
        "Ziua, [Interval orar], [Locație], [Semigrupă]) values " +
        $"([{abvr}], [{type}], [{prof}], [{day}], [{time}], [{place}], [{stds}]);", con);
    insert_cmd.CommandTimeout = 45;
    insert_cmd.ExecuteNonQuery();
    con.Close();
    return "ceva";
}
```

- O să înșiruiesc în acest rezumat rezultatele inserării datelor în tabelul departamentului.



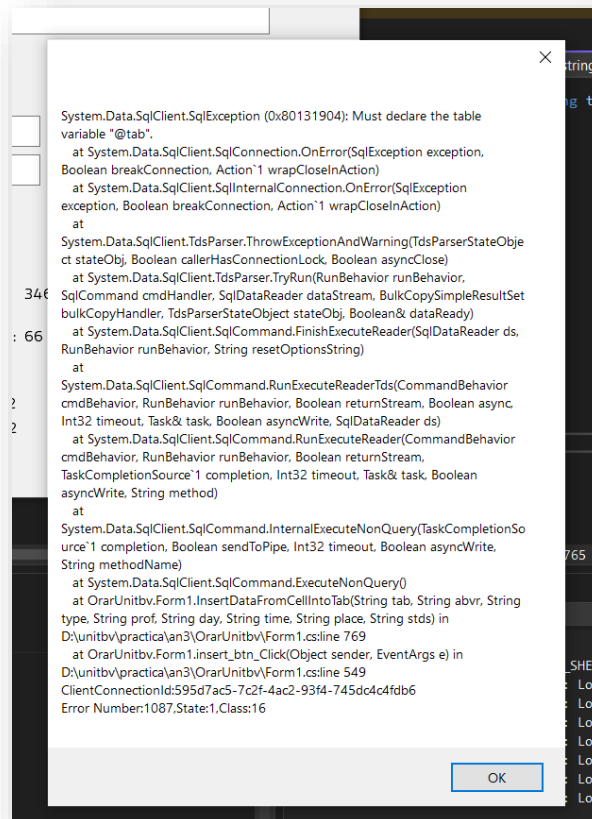
- Eroarea se poate remedia ușor. Am uitat câteva linii de cod esențiale în inserția datelor în tabel.

```

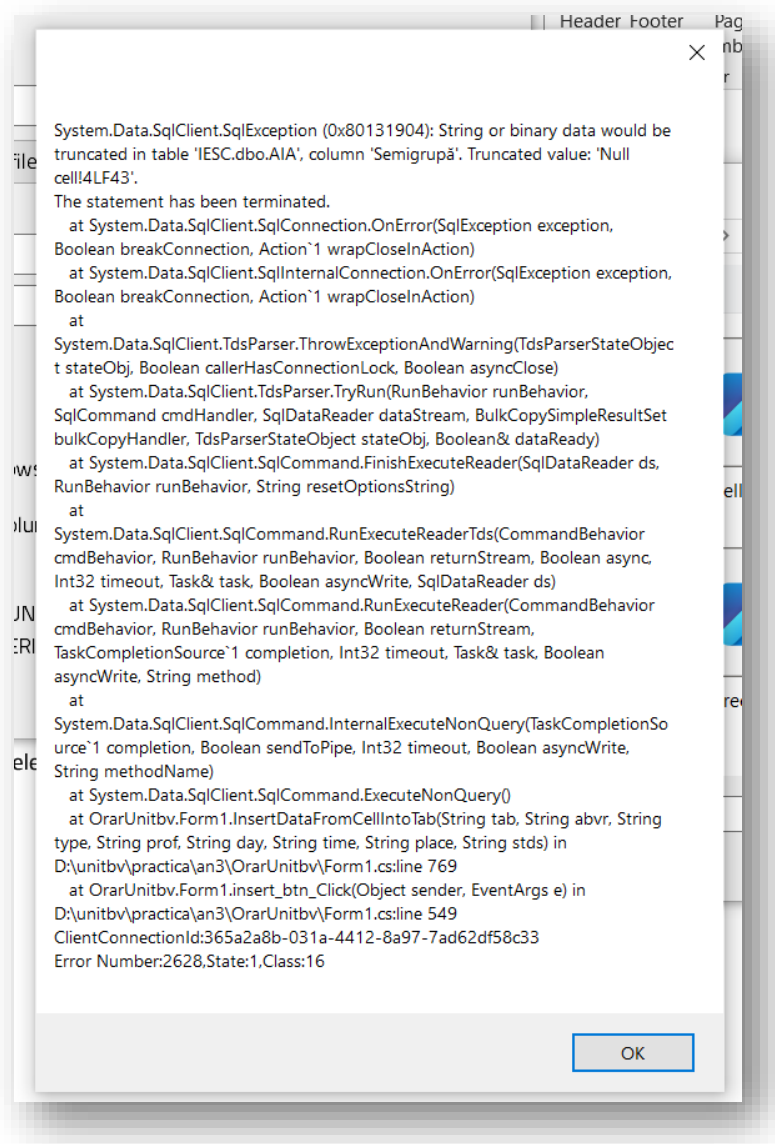
internal string InsertDataFromCellIntoTab(string tab, string abvr, string type, string prof, string
{
    // con.Open();
    // SqlCommand cmd = new SqlCommand("insert into", con);
    // con.Close();
    // e.g. Graf., L, NII8, Musuroi_C_L
    // using CellSplit(str)[] method
    con.Open();
    SqlCommand insert_cmd = new SqlCommand(" " +
        "insert into @tab ([Materie abreviere], Tip, [Cadru didactic], " +
        "Ziua, [Interval orar], [Locatie], [Semigrupa]) values " +
        "(@abvr, @type, @prof, @day, @time, @place, @stds);", con);
    insert_cmd.Parameters.AddWithValue("@tab", tab);
    insert_cmd.Parameters.AddWithValue("@abvr", abvr);
    insert_cmd.Parameters.AddWithValue("@type", type);
    insert_cmd.Parameters.AddWithValue("@prof", prof);
    insert_cmd.Parameters.AddWithValue("@day", day);
    insert_cmd.Parameters.AddWithValue("@time", time);
    insert_cmd.Parameters.AddWithValue("@place", place);
    insert_cmd.Parameters.AddWithValue("@stds", stds);
    insert_cmd.CommandTimeout = 45;
    insert_cmd.ExecuteNonQuery();
    con.Close();
    return "ceva";
}

```

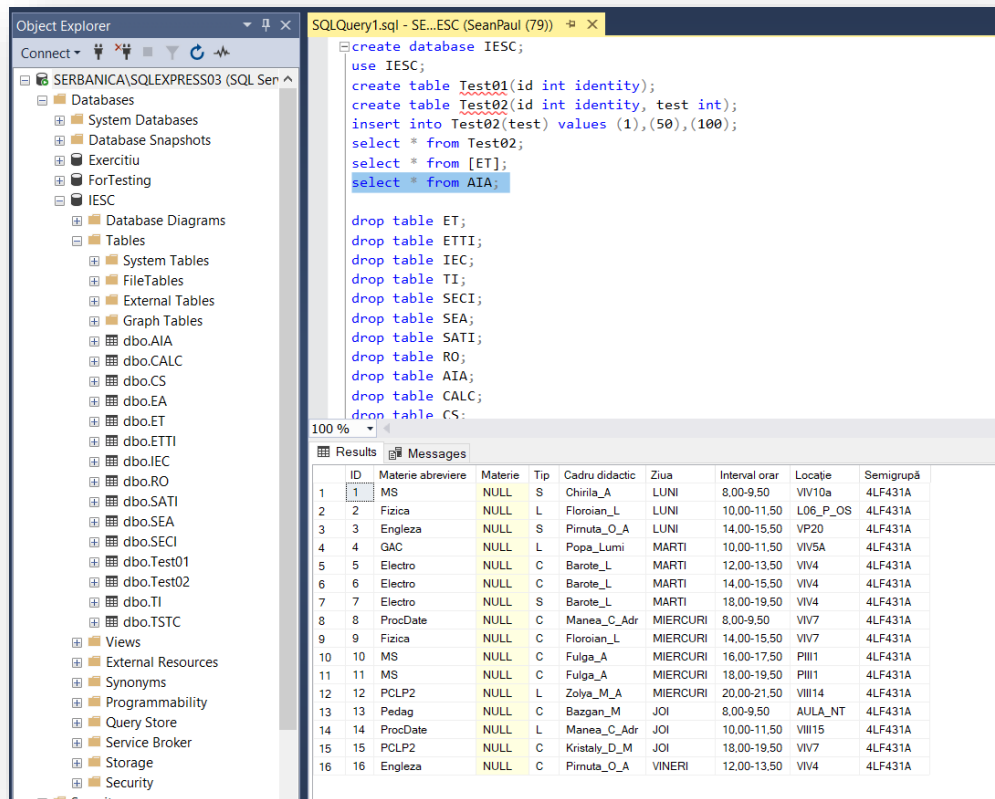
- A doua eroare:



- Numele de entitate nu poate fi introdus în textul comenzii precum se introduc datele, variabilele, deci va fi necesar pentru acesta o altă sintaxă.



- Deși am întâmpinat o excepție, date au fost introduse în tabelul AIA, următoarea imagine este reprezentativă.



- Trebuie adăugat în textul excepției și linia/coloana la care a apărut acest caz excepțional.
- Zilele următoare o să văd ce mai pot să îmbunătățesc.