

Importarea datelor din Excel într-o bază de date

Ana-Maria Comorașu

December 2020

Contents

1	Introducere	2
2	Aplicații folosite	2
3	Introducerea fizică a datelor într-un Excel	2
4	Pașii importării datelor din Excel într-un tabel din baza de date	3
5	Introducerea datelor prin Pattern Matching	6
6	Introducerea datelor utilizând formule	8
7	Concluzii	11

1 Introducere

În referatul trecut am reușit să creez local o bază de date utilizând *Oracle Express Edition*, dar următoarea parte importantă pentru a putea începe exercițiile de laborator pentru proiect este popularea tabelor cu informații. Cea mai simplă metodă generală de a putea introduce informații într-un tabel este folosind un document Excel, întrucât putem face *pattern matching* pe celule, sau le putem copia și formata foarte ușor.

Scopul acestui referat este de a găsi o metodă ușoară și la îndemână pentru a popula baza de date cu cât mai multe informații, folosind o aplicație deja cunoscută de toată lumea încă din liceu, sau chiar gimnaziu.

De asemenea, Excel este mai accesibil pentru tabele pentru că merge pe ideea **WYSI-WYG**, adică *What You See Is What You Get*.

2 Aplicații folosite

Pentru acest referat am utilizat:

- Microsoft Office Excel (alternativă free puteți descărca de aici).
- Oracle SQL Developer

Pentru exemplificare, voi folosi baza de date creată local de la referatul anterior, având entitățile ce vor fi utilizate ulterior pentru proiectul final.

3 Introducerea fizică a datelor într-un Excel

În exemplul de mai jos, este prezentat un exemplu de date ce pot fi inserate în tabelul "zi" din baza de date deja creat.

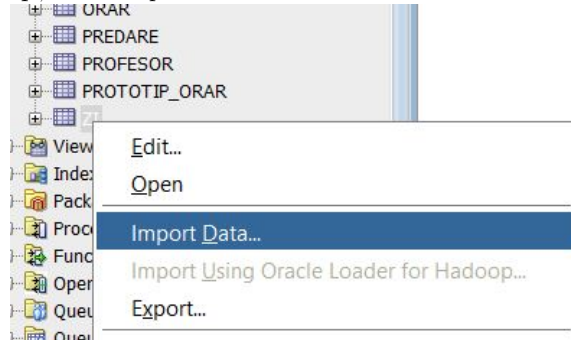
```
1 -- ZI --
2 CREATE TABLE zi(
3     zi_id NUMBER(10) PRIMARY KEY,
4     nume VARCHAR2(20)
5 );
```

id_zi	denumire
1	Luni
2	Marti
3	Miercuri
4	Joi
5	Vineri
6	Sambata
7	Duminica

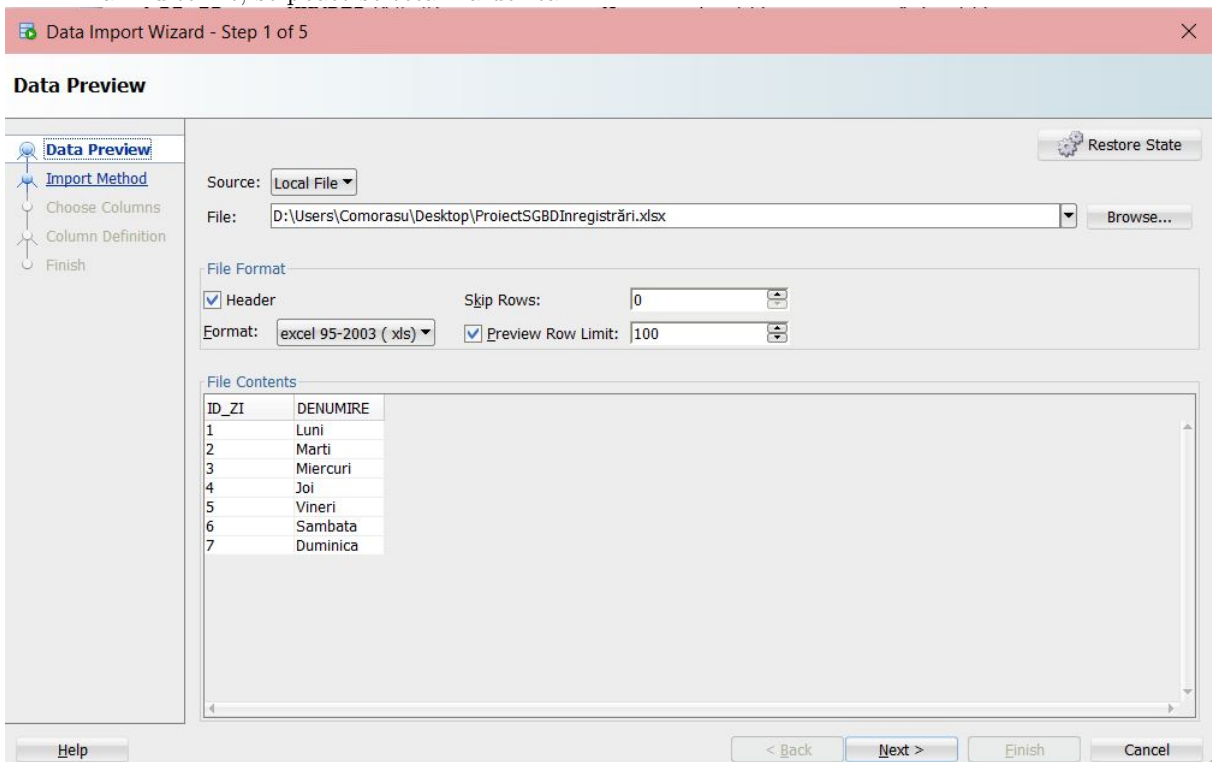
4 Pași importării datelor din Excel într-un tabel din baza de date

Mai departe vom urmări pașii ce trebuie parcurși pentru a putea introduce datele din tabelul nostru în entitatea "zi" din baza de date.

Primul pas (evident) este salvarea documentului Excel. Se va deschide în SQL Developer baza de date unde se va accesa click dreapta pe tabela în care dorim să introducem datele. Apoi, vom selecta opțiunea *Import Data*.



Mai departe, se va selecta sursa din care vrem să importăm înregistrări. Pentru tabelul din exemplu, a fost selectat și formatul cu *Header*, precum s-a putut observa în tabel. În cadrul *File Contents*, se poate verifica și conținutul filei. Dacă documentul Excel conține mai multe file, se poate selecta fila dorită.



Odată ce ați selectat opțiunile dorite, apăsați *Next*. În etapa a 2-a, vor fi prezentate două metode de importare: *Insert* sau *Insert Script*. Pentru primul exemplu am selectat *Insert*, care va insera înregistrările. În schimb, *Insert Script* generează codul SQL pentru inserarea valorilor în tabelă.

Data Import Wizard - Step 2 of 4

Import Method

Specify the method for importing data. For Staging External Table method, an external table will be created as a staging table for importing the Target Table. For other import methods, data is imported directly into the table.

Import Method: **Insert**

☐ Send Create Script to SQL Worksheet

Table Name: **ZI**

☐ Import Row Limit: **100**

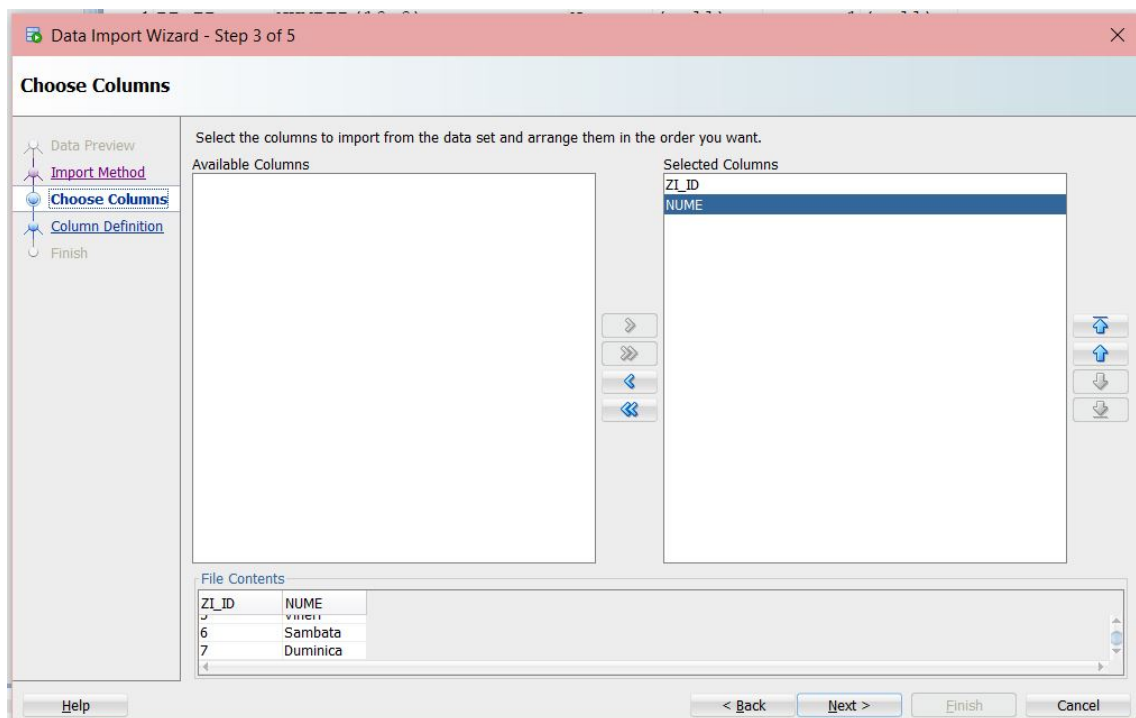
File Contents

ID_ZI	DENUMIRE
1	Luni
2	Marti
3	Miercuri
4	Joi
5	Vineri
6	Sambata
7	Duminica

Help **< Back** **Next >** **Finish** **Cancel**

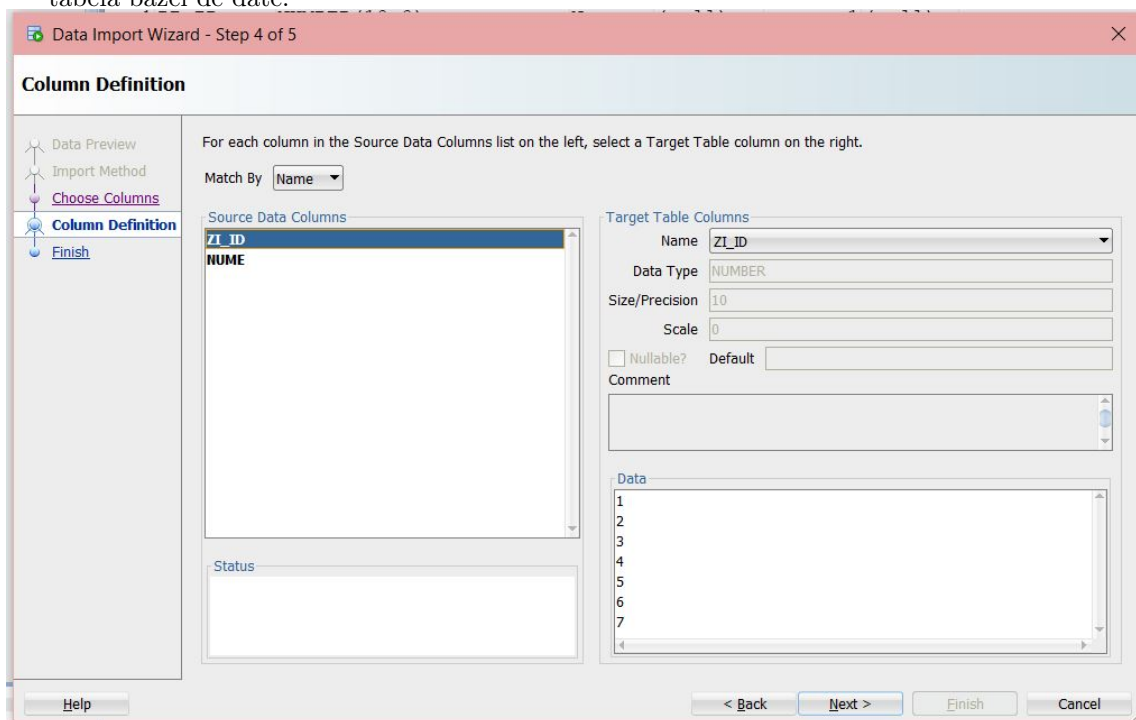
Click *Next*.

Mai departe veți putea selecta numele coloanelor din care doriți să importați datele. Acestea vor fi selectate automat, în schimb le puteți muta în partea stângă prin intermediul săgeților dacă nu doriți ca acestea să fie utilizate. Când ați terminat, **Click** *Next*, bineînțeles.

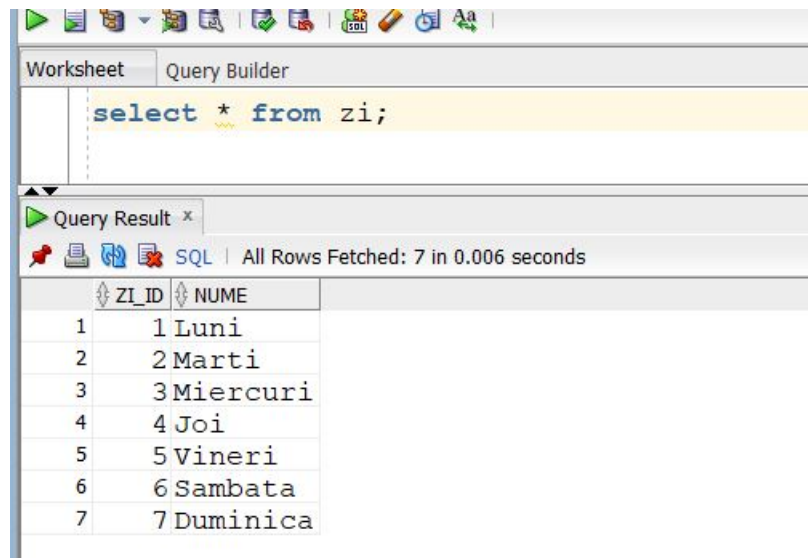


La pasul următor, veți putea face match între coloanele din fila din documentul Excel (sursa) și coloanele din baza de date (destinația). Pentru a diminua confuzia, am denumit coloanele identic.

Atenție! Datele din tabelă trebuie să se potrivească și cu tipul de date din coloana din tabela bazei de date.



După ce ați terminat parcurgerea tuturor coloanelor, puteți apăsa *Finish*.
Am verificat ulterior dacă inserarea s-a realizat cu succes.



Perfect! A funcționat!

5 Introducerea datelor prin Pattern Matching

Pentru cel de-al doilea exemplu, doresc să utilizez pattern matching pe celule, care poate fi foarte folositor atunci când înregistrările din baza de date merg după o regulă foarte simplă. Pentru cel de-al doilea exemplu, voi folosi tabela *Ora*, care este structurată:

```

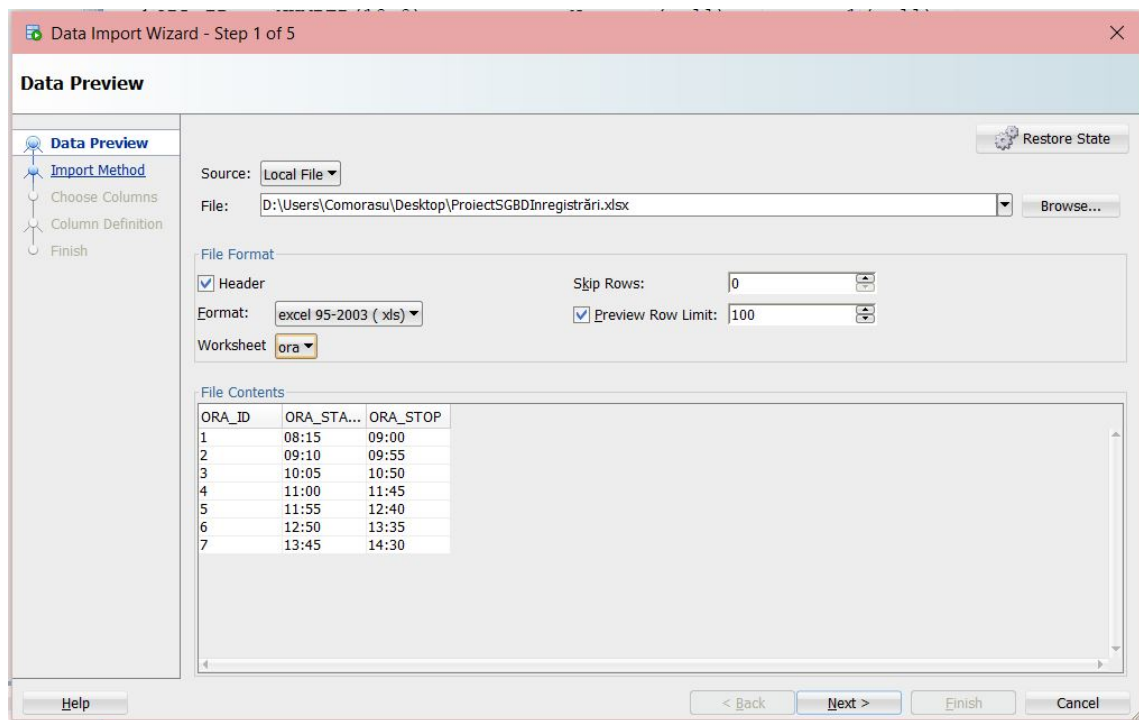
1 -- ORA --
2 CREATE TABLE ora(
3     ora_id NUMBER(10) PRIMARY KEY,
4     ora_start VARCHAR2(20) NOT NULL,
5     ora_stop VARCHAR2(20)
6 );

```

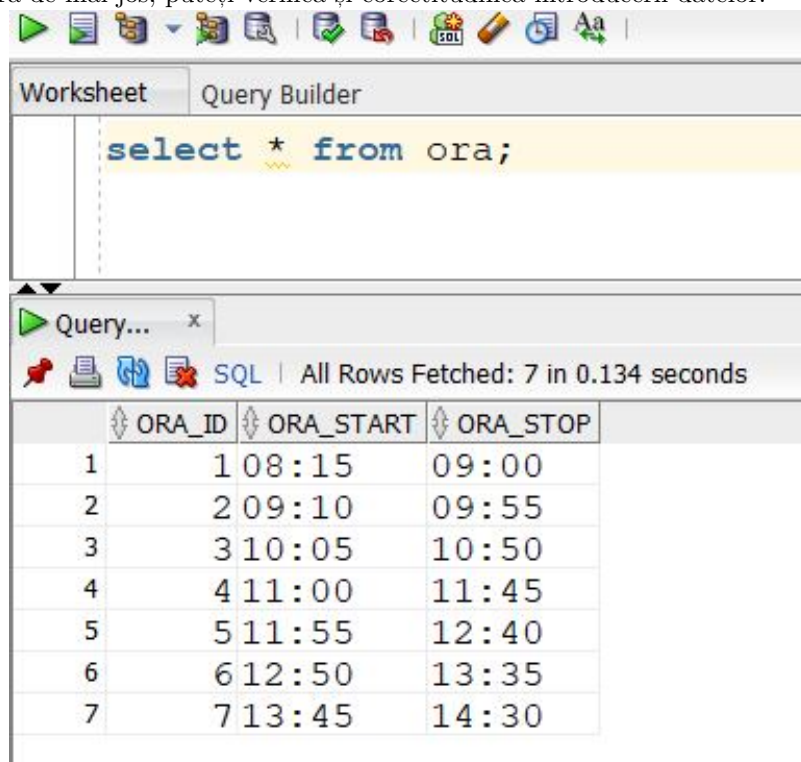
Voi presupune că prima oră va începe la ora 8:15 și durează 45 de minute. Pauza dintre ore va fi de 10 minute. Introducând datele pentru primele 2 exemple, am selectat celulele și am *tras* valorile pentru mai multe linii. Am realizat acest lucru lucrând atât cu numere, cât și cu date orare.

	A	B	C	D
1	ora_id	ora_start	ora_stop	
2		8:15	9:00	
3		9:10	9:55	
4		10:05	10:50	
5		11:00	11:45	
6		11:55	12:40	
7		12:50	13:35	
8		13:45	14:30	
9				
10				

Ulterior, am urmărit aceiași pași ca la [exemplul anterior](#).



În figura de mai jos, puteți verifica și corectitudinea introducerii datelor.



6 Introducerea datelor utilizând formule

Până acum am lucrat cu text simplu pentru a putea introduce date în tabel, dar de multe ori nu va fi suficient doar să introducem datele de mână. Uneori poate dorim să folosim o formulă mai complicată, unde nu ne poate ajuta un simplu pattern matching.

Din ce am testat în plus, SQL Developer se comportă foarte bine și cu funcțiile cu numere, if-uri și altele. În schimb, am sesizat o problemă interesantă atunci când am dorit să generez cheia primară concatenând 3 coloane.

A2					=CONCAT(B2,C2,D2)				
	A	B	C	D					
1	predare_id	profesor_id	clasa_id	materie_id					
2	1291MAT	12	91	MAT					
3	2792MAT	27	92	MAT					
4	21101MAT	21	101	MAT					
5	12102MAT	12	102	MAT					
6	21111MAT	21	111	MAT					
7	27121MAT	27	121	MAT					
8	2192INF	21	92	INF					
9	25101INF	25	101	INF					
10	25111INF	25	111	INF					
11	21121INF	21	121	INF					
12	2391LRO	23	91	LRO					
13	2392LRO	23	92	LRO					
14	23101LRO	23	101	LRO					
15	23102LRO	23	102	LRO					
16	23111LRO	23	111	LRO					
17	23121LRO	23	121	LRO					
18	1591LEN	15	91	LEN					
19	1592LEN	15	92	LEN					
20	15101LEN	15	101	LEN					
21	15102LEN	15	102	LEN					
22	15111LEN	15	111	LEN					
23	15121LEN	15	121	LEN					
24	1591LFR	15	91	LFR					
25	1592LFR	15	92	LFR					

Din păcate, în SQL Developer nu s-a putut realiza importarea datelor folosind funcția **CONCAT** (o funcție mai nouă în Excel), în schimb a funcționat atunci când am utilizat **CONCATENATE** (variante sa mai veche). De asemenea, trebuie să menționez că am selectat și varianta de format 2003+, având același rezultat.

Data Preview

Source: Local File

File: D:\Users\Comorasu\Desktop\ProiectSGBDInregistrări.xlsx

File Format

☒ Header

Format: excel 95-2003 (.xls)

Worksheet: predare

Skip Rows: 0

☒ Preview Row Limit: 100

File Contents

PREDARE...	PROFESO...	CLASA_ID	MATERIE_ID	DESCRIERE
------------	------------	----------	------------	-----------

Pentru exemplul meu, am concatenat valorile astfel încât să aibă și aceeași lungime.

A2						
=CONCATENATE(B2,IF(C2<100,CONCATENATE("0",C2),C2),D2)						
	A	B	C	D	E	F
1	predare_id	profesor_id	clasa_id	materie_id	descriere	
2	12091MAT	12	91	MAT	1	
3	27092MAT	27	92	MAT	3	
4	21101MAT	21	101	MAT	4	
5	12102MAT	12	102	MAT	1	
6	21111MAT	21	111	MAT	5	
7	27121MAT	27	121	MAT	4	
8	21092INF	21	92	INF	1	
9	25101INF	25	101	INF	3	
10	25111INF	25	111	INF	3	
11	21121INF	21	121	INF	1	
12	23091LRO	23	91	LRO	3	
13	23092LRO	23	92	LRO	3	

Data Preview

Import Method

Choose Columns

Column Definition

Finish

Source: Local File

File: D:\Users\Comorasu\Desktop\ProiectSGBDInregistrări.xlsx

File Format

☒ Header

Format: excel 95-2003 (.xls)

Worksheet: Predare2

Skip Rows: 0

☒ Preview Row Limit: 100

File Contents

PREDARE...	PROFESO...	CLASA_ID	MATERIE_ID	DESCRIERE
12091MAT	12	91	MAT	1
12091MAT	27	92	MAT	3
12091MAT	21	101	MAT	4
12091MAT	12	102	MAT	1
12091MAT	21	111	MAT	5
12091MAT	27	121	MAT	4
12091MAT	21	92	INF	1
12091MAT	25	101	INF	3
12091MAT	25	111	INF	3
12091MAT	21	121	INF	1
12091MAT	23	91	LRO	3
12091MAT	23	92	LRO	3
12091MAT	23	101	LRO	3

Help

< Back

Next >

Fin

După cum puteți observa mai sus, acum nu mai există probleme de citire a datelor. Mai jos, este un exemplu de *Insert Script*, cea de-a doua variantă de inserare a înregistrărilor.

0.47400001 seconds

Worksheet Query Builder

```

SET DEFINE OFF

INSERT INTO PREDARE (PREDARE_ID, PROFESOR_ID, CLASA_ID, MATERIE_ID, DESCRIERE)
VALUES ('1291MAT', 12, 91, 'MAT', NULL);

INSERT INTO PREDARE (PREDARE_ID, PROFESOR_ID, CLASA_ID, MATERIE_ID, DESCRIERE)
VALUES ('2792MAT', 27, 92, 'MAT', NULL);

INSERT INTO PREDARE (PREDARE_ID, PROFESOR_ID, CLASA_ID, MATERIE_ID, DESCRIERE)
VALUES ('21101MAT', 21, 101, 'MAT', NULL);

INSERT INTO PREDARE (PREDARE_ID, PROFESOR_ID, CLASA_ID, MATERIE_ID, DESCRIERE)
VALUES ('12102MAT', 12, 102, 'MAT', NULL);

INSERT INTO PREDARE (PREDARE_ID, PROFESOR_ID, CLASA_ID, MATERIE_ID, DESCRIERE)
VALUES ('21111MAT', 21, 111, 'MAT', NULL);

INSERT INTO PREDARE (PREDARE_ID, PROFESOR_ID, CLASA_ID, MATERIE_ID, DESCRIERE)
VALUES ('27121MAT', 27, 121, 'MAT', NULL);

INSERT INTO PREDARE (PREDARE_ID, PROFESOR_ID, CLASA_ID, MATERIE_ID, DESCRIERE)
VALUES ('2192INF', 21, 92, 'INF', NULL);

```

7 Concluzii

După cum s-a observat în exemplele anterioare, este mult mai ușor și mai convenabil să introducem datele (chiar și de mână, dacă nu se depășesc 100 de înregistrări) decât să scriem fiecare insert de mână, întrucât putem folosi pattern matching în unele locuri, formule, sau chiar să copiem datele dintr-un alt tabel.

Pentru proiectul acestei materii ar fi foarte accesibilă această metodă, dacă nu se folosesc lucruri foarte complicate.

Totuși, așa cum s-a observat și din ultimul exemplu, SQL Developer s-ar putea să aibă niște limitări în această privință, pentru cazurile în care vom dori să folosim niște funcții mai noi adăugate în aplicația Excel (apărute după 2003?).

Per total, nu consider acest obstacol un dezavantaj, întrucât scopul utilizării Excel a fost în principal o încercare de **Copy-Paste**, care s-a dovedit un succes.