***PROIECT BAZE DE DATE***

**PLATFORMĂ DE GAMING**

***Jiglău Fabrizzio***

***GRUPA 133***

***2023 - 2024***

***CUPRINS***

[CERINȚE 4](#_Toc170509538)

[DESCRIEREA MODELULUI REAL 5](#_Toc170509539)

[PREZENTAREA CONSTRÂNGERILOR 5](#_Toc170509540)

[DESCRIEREA ENTITĂȚILOR 6](#_Toc170509541)

[DESCRIEREA RELAȚIILOR 7](#_Toc170509542)

[DESCRIEREA ATRIBUTELOR 9](#_Toc170509543)

[DIAGRAMA ENTITATE-RELAȚIE 13](#_Toc170509544)

[DIAGRAMA CONCEPTUALĂ 14](#_Toc170509545)

[SCHEME RELAȚIONALE 15](#_Toc170509546)

[REALIZAREA NORMALIZĂRII PÂNĂ LA FN3 15](#_Toc170509547)

[SECVENȚĂ UTILIZATĂ ÎN INSERAREA ÎNREGISTRĂRILOR ÎN TABEL 16](#_Toc170509548)

[CREAREA TABELELOR ȘI ÎNREGISTRAREA DATELOR 16](#_Toc170509549)

[UTILIZATOR 16](#_Toc170509550)

[NOTIFICARE 18](#_Toc170509551)

[MESAJ 20](#_Toc170509552)

[TARA 23](#_Toc170509553)

[LOCATIE 26](#_Toc170509554)

[DEVELOPER 29](#_Toc170509555)

[JOC 31](#_Toc170509556)

[RECENZIE 34](#_Toc170509557)

[COLECTIE 37](#_Toc170509558)

[COLECTIE\_JOC 40](#_Toc170509559)

[INVENTAR 42](#_Toc170509560)

[ITEM 43](#_Toc170509561)

[INVENTAR\_ITEM 47](#_Toc170509562)

[TRANZACTIE 49](#_Toc170509563)

[Cereri SQL complexe (12) 52](#_Toc170509564)

[CEREREA 1 52](#_Toc170509565)

[CEREREA 2 53](#_Toc170509566)

[CEREREA 3 54](#_Toc170509567)

[CEREREA 4 55](#_Toc170509568)

[CEREREA 5 56](#_Toc170509569)

[Actualizare și suprimare a datelor 57](#_Toc170509570)

[C1 57](#_Toc170509571)

[C2 59](#_Toc170509572)

[C3 61](#_Toc170509573)

[VIZUALIZARE COMPLEXĂ 63](#_Toc170509574)

[Crearea vizualizării 63](#_Toc170509575)

[Operatie permisa 63](#_Toc170509576)

[Operatie nepermisa 64](#_Toc170509577)

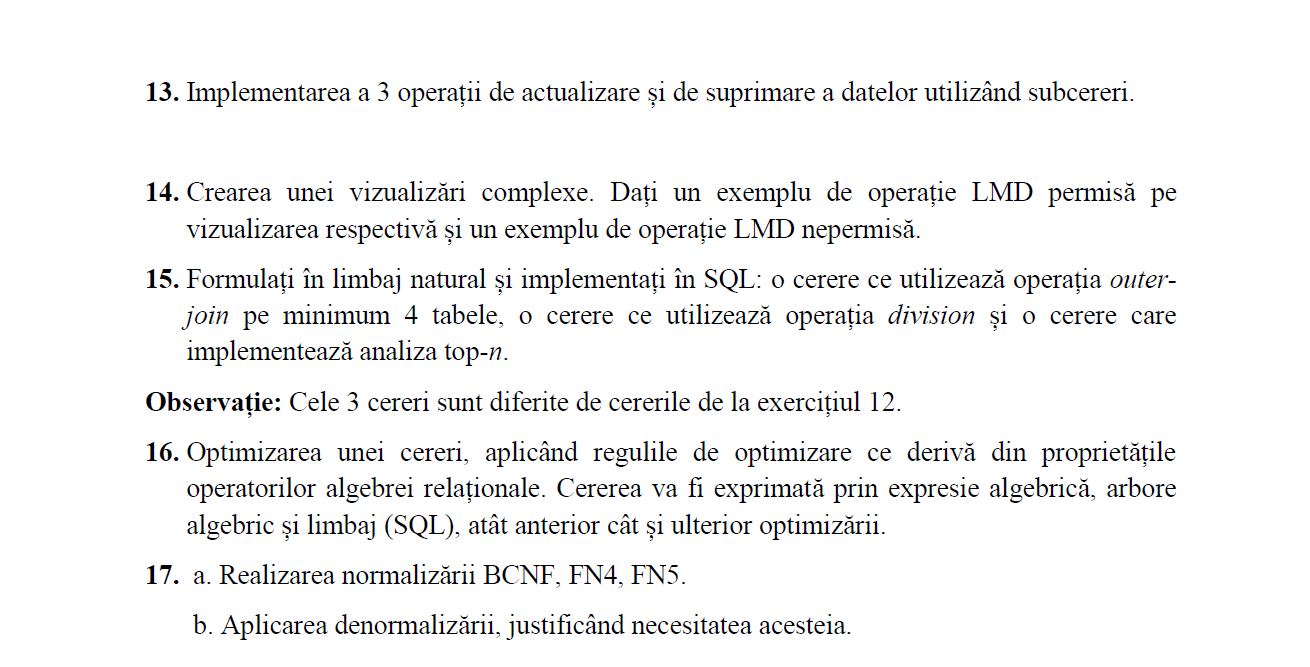
[Cereri SQL complexe (15) 65](#_Toc170509578)

[CEREREA 1 65](#_Toc170509579)

[CEREREA 2 66](#_Toc170509580)

[CEREREA 3 66](#_Toc170509581)

# CERINȚE

****

# DESCRIEREA MODELULUI REAL

Această bază de date este proiectată pentru a gestiona o platformă de gaming complexă, care include informații despre utilizatori, jocuri, dezvoltatori de jocuri, tranzacții, recenzii, notificări, mesaje, țări, locații, colecții, inventare și iteme. Scopul principal al acestei baze de date este de a organiza și stoca datele necesare pentru funcționarea eficientă și integrată a platformei, asigurând în același timp integritatea și consistența datelor.

Modelul este inspirat din celebrul Steam, un gigant al industriei de jocuri video. Baza de date permite utilizatorului să trăiască experiența adusă de original, adăugând și utilități extra. User-ul are acces atât la o gamă largă și variată de jocuri, cât și la un marketplace centrat în jurul itemelor valoroase ce apartin de jocurile platformei.

# PREZENTAREA CONSTRÂNGERILOR

* Un utilizator poate avea mai multe recenzii, iar o recenzie este asociată unui singur utilizator și unui singur joc.
* Un utilizator poate efectua mai multe tranzacții, iar o tranzacție este asociată unui singur utilizator.
* Un utilizator poate primi mai multe notificări, iar o notificare este asociată unui singur utilizator.
* Un utilizator poate avea mai multe colecții, iar o colecție este asociată unui singur utilizator.
* Un utilizator poate avea un singur inventar, iar un inventar este asociat unui singur utilizator.
* O notificare are un singur mesaj asociat, iar un mesaj aparține unei singure notificări.
* O colecție poate conține mai multe jocuri, iar un joc poate aparține mai multor colecții.
* Un inventar poate conține mai multe iteme, iar un item poate aparține mai multor inventare.
* Un dezvoltator poate crea mai multe jocuri, iar un joc este creat de un singur dezvoltator.
* O locație poate fi asociată mai multor dezvoltatori, iar un dezvoltator este localizat într-o singură locație.
* O țară poate avea mai multe locații, iar o locație este asociată unei singure țări.

# DESCRIEREA ENTITĂȚILOR

Pentru modelul de date al platformei de gaming, entitățile principale sunt: **Utilizator, Recenzie, Tranzactie, Notificare, Mesaj, Tara, Locatie, Developer, Colectie, Joc, Colectie\_Joc, Inventar, Item și Inventar\_Item**.

* UTILIZATOR – Reprezintă persoanele care își crează un cont pe platformă. Aceștia pot interacționa cu diverse funcționalități ale platformei, cum ar fi jocurile, recenziile și tranzacțiile. Cheia primară este id\_utilizator.
* RECENZIE – Recenziile sunt evaluările lăsate de utilizatori pentru diferite jocuri disponibile pe platformă. Fiecare recenzie este legată de un utilizator și un joc. Cheia primară este id\_recenzie.
* TRANZACTIE – Tranzacțiile reprezintă plățile efectuate de utilizatori. Fiecare tranzacție este asociată unui utilizator. Cheia primară este id\_tranzactie.
* NOTIFICARE – Notificările sunt texte scurte trimise utilizatorilor pentru a-i informa despre diverse evenimente sau promoții. Fiecare notificare este legată de un utilizator. Cheia primară este id\_notificare.
* MESAJ – Mesajele sunt conținutul individual al fiecărei notificări, fiecare mesaj fiind asociat unei notificări. Cheia primară este id\_mesaj.
* TARA – Țările sunt utilizate pentru a specifica locația dezvoltatorilor de jocuri și a altor entități. Cheia primară este id\_tara.
* LOCATIE – Locațiile sunt specificate pentru fiecare dezvoltator și sunt legate de o țară. Cheia primară este id\_locatie.
* DEVELOPER – Developerii sunt entitățile care creează jocurile. Fiecare dezvoltator este asociat unei locații. Cheia primară este id\_developer.
* COLECTIE – Colecțiile sunt seturi de jocuri organizate de utilizatori. Fiecare colecție este legată de un utilizator. Cheia primară este id\_colectie.
* JOC – Jocurile sunt produsele principale ale platformei, fiecare joc fiind creat de un dezvoltator și putând fi inclus în colecții, dar si evaluat de utilizatori prin recenzii. Cheia primară este id\_joc.
* COLECTIE\_JOC – Această entitate gestionează relația de tip many-to-many între jocuri și colecții. Cheile primare sunt id\_colectie și id\_joc.
* INVENTAR – Inventarele sunt seturi de iteme deținute de utilizatori. Fiecare inventar este asociat unui utilizator. Cheia primară este id\_inventar.
* ITEM – Itemele sunt obiecte virtuale care pot fi achiziționate sau câștigate în jocuri. Fiecare item este asociat unui joc. Cheia primară este id\_item.
* INVENTAR\_ITEM – Această entitate gestionează relația de tip many-to-many între iteme și inventare. Cheile primare sunt id\_item și id\_inventar.

# DESCRIEREA RELAȚIILOR

***Utilizator și Recenzie***

* Relație: Un utilizator poate avea mai multe recenzii, dar fiecare recenzie este asociată cu un singur utilizator.
* Cardinalitate: 1 la M(0) (Un utilizator, Multe recenzii)

Utilizator și Tranzactie

• Relație: Un utilizator poate efectua mai multe tranzacții, fiecare tranzacție fiind asociată cu un singur utilizator.

• Cardinalitate: 1 la M(0) (Un utilizator, Multe tranzacții)

***Utilizator și Notificare***

• Relație: Un utilizator poate primi mai multe notificări, fiecare notificare fiind asociată cu un singur utilizator.

• Cardinalitate: 1 la M(0) (Un utilizator, Multe notificări)

***Notificare și Mesaj***

• Relație: Fiecare notificare are un singur mesaj asociat, iar fiecare mesaj aparține unei singure notificări.

• Cardinalitate: 1 la 1 (O notificare, Un mesaj)

***Tara și Locatie***

• Relație: O țară poate avea mai multe locații, dar fiecare locație este asociată cu o singură țară.

• Cardinalitate: 1 la M(0) (O țară, Multe locații)

***Locatie și Developer***

• Relație: O locație poate găzdui mai mulți dezvoltatori, dar fiecare dezvoltator este asociat cu o singură locație.

• Cardinalitate: 1 la M(0) (O locație, Mulți dezvoltatori)

***Developer și Joc***

• Relație: Un dezvoltator poate crea mai multe jocuri, dar fiecare joc este asociat cu un singur dezvoltator.

• Cardinalitate: 1 la M(0) (Un dezvoltator, Multe jocuri)

***Utilizator și Colectie***

• Relație: Un utilizator poate avea mai multe colecții, dar fiecare colecție este asociată cu un singur utilizator.

• Cardinalitate: 1 la M(0) (Un utilizator, Multe colecții)

***Colectie și Joc (prin Colectie\_Joc)***

• Relație: O colecție poate conține mai multe jocuri și un joc poate fi inclus în mai multe colecții.

• Cardinalitate: M(0) la M(0) (Multe colecții, Multe jocuri)

***Utilizator și Inventar***

• Relație: Fiecare utilizator are un singur inventar asociat.

• Cardinalitate: 1 la 1 (Un utilizator, Un inventar)

***Inventar și Item (prin Inventar\_Item)***

• Relație: Un inventar poate conține mai multe iteme și un item poate fi inclus în mai multe inventare.

• Cardinalitate: M(0) la M(0) (Multe inventare, Multe iteme)

***Recenzie și Joc***

• Relație: O recenzie este asociată cu un singur joc, iar un joc poate avea mai multe recenzii.

• Cardinalitate: 1 la M(0) (Un joc, Multe recenzii)

***Item și Joc***

• Relație: Un item este asociat cu un singur joc, iar un joc poate avea mai multe iteme.

• Cardinalitate: 1 la M(0) (Un joc, Multe iteme)

***Utilizator și Tara***

• Relație: Un utilizator este asociat cu o singură țară, iar o țară poate avea mai mulți utilizatori.

• Cardinalitate: 1 la M(0) (O țară, Mulți utilizatori)

# DESCRIEREA ATRIBUTELOR

**Utilizator**

* id\_utilizator (PK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul unic pentru fiecare utilizator.
* nume\_utilizator (Not Null) – Variabilă de tip caracter de lungime maxim 50, reprezentând numele utilizatorului.
* parola (Not Null) – Variabilă de tip caracter de lungime maxim 50, reprezentând parola utilizatorului.
* email (Not Null, Unique) – Variabilă de tip caracter de lungime maxim 50, reprezentând emailul personal al utilizatorului.
* varsta (Not Null) – Variabilă de tip întreg de lungime maxim 3, reprezentând vârsta utilizatorului, cu o valoare minimă de 0.
* id\_tara (FK, Not Null) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul țării utilizatorului.

**Recenzie**

* id\_recenzie (PK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul unic pentru fiecare recenzie.
* id\_utilizator (FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul utilizatorului care a scris recenzia.
* id\_joc (FK, Not Null) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul jocului recenzat.
* text\_recenzie (Not Null) – Variabilă de tip caracter de lungime maxim 256, reprezentând textul recenziei.
* data\_recenzie (Not Null) – Variabilă de tip dată, reprezentând data la care a fost scrisă recenzia.

**Tranzactie**

* id\_tranzactie (PK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul unic pentru fiecare tranzacție.
* id\_utilizator (FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul utilizatorului care a efectuat tranzacția.
* metoda\_tranzactie (Not Null) – Variabilă de tip caracter de lungime maxim 25, reprezentând metoda de plată utilizată.
* suma\_tranzactie (Not Null) – Variabilă de tip întreg de lungime maxim 5, reprezentând suma tranzacției, cu o valoare minimă de 0.
* data\_tranzactie (Not Null) – Variabilă de tip dată, reprezentând data la care a avut loc tranzacția.

**Notificare**

* id\_notificare (PK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul unic pentru fiecare notificare.
* id\_utilizator (FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul utilizatorului care a primit notificarea.
* text\_notificare (Not Null) – Variabilă de tip caracter de lungime maxim 50, reprezentând textul notificării.

**Mesaj**

* id\_mesaj (PK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul unic pentru fiecare mesaj.
* id\_notificare (FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul notificării asociate.
* text\_mesaj (Not Null) – Variabilă de tip caracter de lungime maxim 256, reprezentând textul mesajului.
* este\_citit (Not Null) – Variabilă de tip întreg de lungime maxim 1, reprezentând dacă mesajul a fost citit (0 sau 1).

**Tara**

* id\_tara (PK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul unic pentru fiecare țară.
* nume\_tara (Not Null) – Variabilă de tip caracter de lungime maxim 50, reprezentând numele țării.
* salariu\_mediu (Not Null) – Variabilă de tip întreg de lungime maxim 5, reprezentând salariul mediu în țară, cu o valoare minimă de 0.

**Locatie**

* id\_locatie (PK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul unic pentru fiecare locație.
* id\_tara (FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul țării în care se află locația.
* oras (Not Null) – Variabilă de tip caracter de lungime maxim 25, reprezentând orașul în care se află locația.

**Developer**

* id\_developer (PK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul unic pentru fiecare dezvoltator.
* id\_locatie (FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul locației dezvoltatorului.
* nume\_developer (Not Null) – Variabilă de tip caracter de lungime maxim 50, reprezentând numele dezvoltatorului.
* data\_lansare\_studio (Not Null) – Variabilă de tip dată, reprezentând data lansării studioului de dezvoltare.
* salariu\_angajati (Not Null) – Variabilă de tip întreg de lungime maxim 5, reprezentând salariul mediu al angajaților, cu o valoare minimă de 0.

**Colectie**

* id\_colectie (PK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul unic pentru fiecare colecție.
* id\_utilizator (FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul utilizatorului care deține colecția.
* nume\_colectie (Not Null) – Variabilă de tip caracter de lungime maxim 50, reprezentând numele colecției.

**Joc**

* id\_joc (PK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul unic pentru fiecare joc.
* id\_developer (FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul dezvoltatorului care a creat jocul.
* nume\_joc (Not Null) – Variabilă de tip caracter de lungime maxim 50, reprezentând numele jocului.
* pret\_joc (Not Null) – Variabilă de tip întreg de lungime maxim 5, reprezentând prețul jocului, cu o valoare minimă de 0.
* data\_lansare\_joc (Not Null) – Variabilă de tip dată, reprezentând data lansării jocului.
* varsta\_recomandata – Variabilă de tip întreg de lungime maxim 3, reprezentând vârsta recomandată pentru joc.
* rating – Variabilă de tip întreg de lungime maxim 1, reprezentând rating-ul jocului.

**Colectie\_Joc**

* id\_colectie (PK, FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul colecției.
* id\_joc (PK, FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul jocului.

**Inventar**

* id\_inventar (PK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul unic pentru fiecare inventar.
* id\_utilizator (FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul utilizatorului care deține inventarul.
* capacitate (Not Null) – Variabilă de tip întreg de lungime maxim 2, reprezentând capacitatea inventarului.

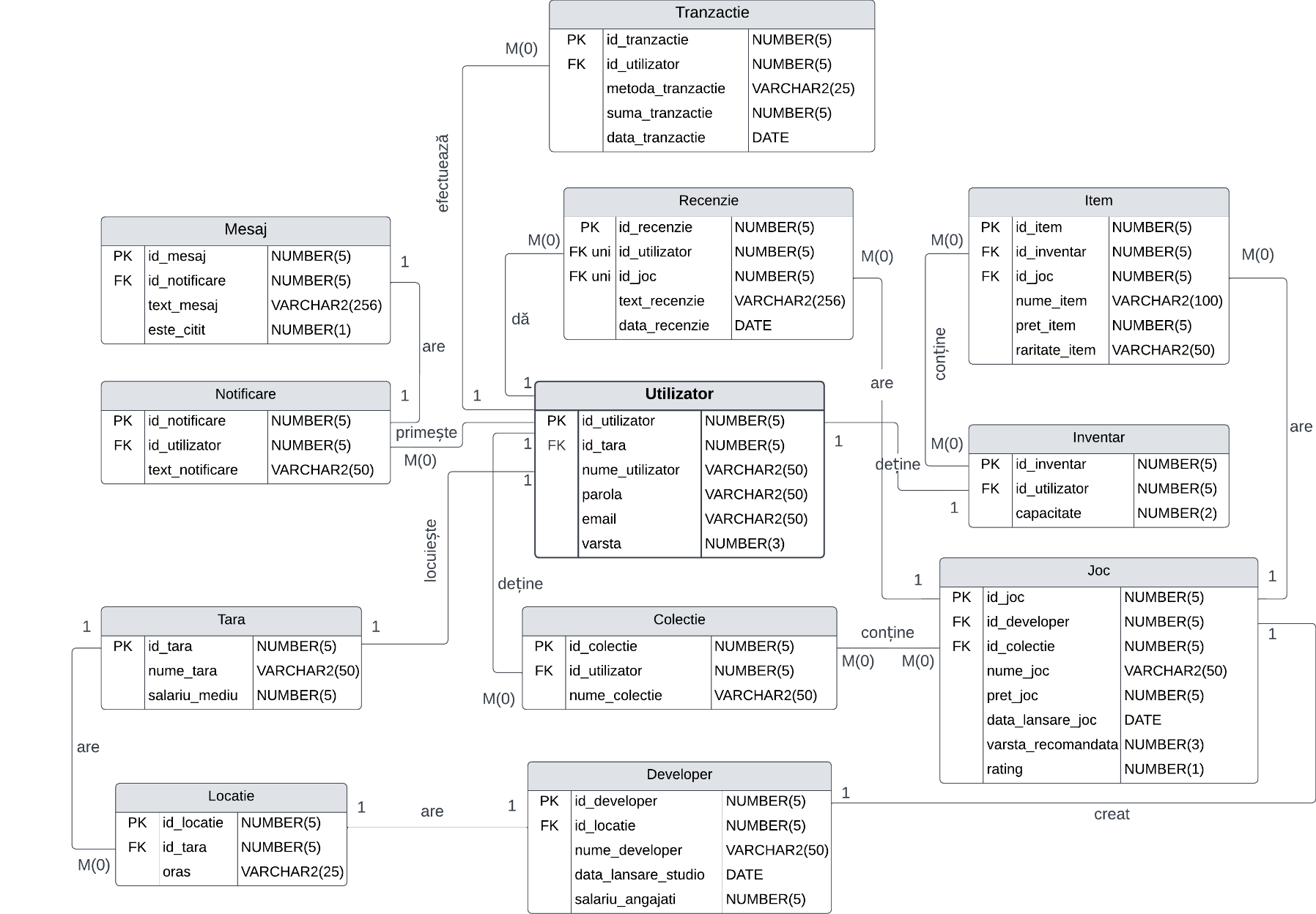
**Item**

* id\_item (PK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul unic pentru fiecare item.
* id\_joc (FK, Not Null) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul jocului asociat itemului.
* nume\_item (Not Null) – Variabilă de tip caracter de lungime maxim 100, reprezentând numele itemului.
* pret\_item – Variabilă de tip întreg de lungime maxim 10,2, reprezentând prețul itemului.
* raritate\_item (Not Null) – Variabilă de tip caracter de lungime maxim 50, reprezentând raritatea itemului.

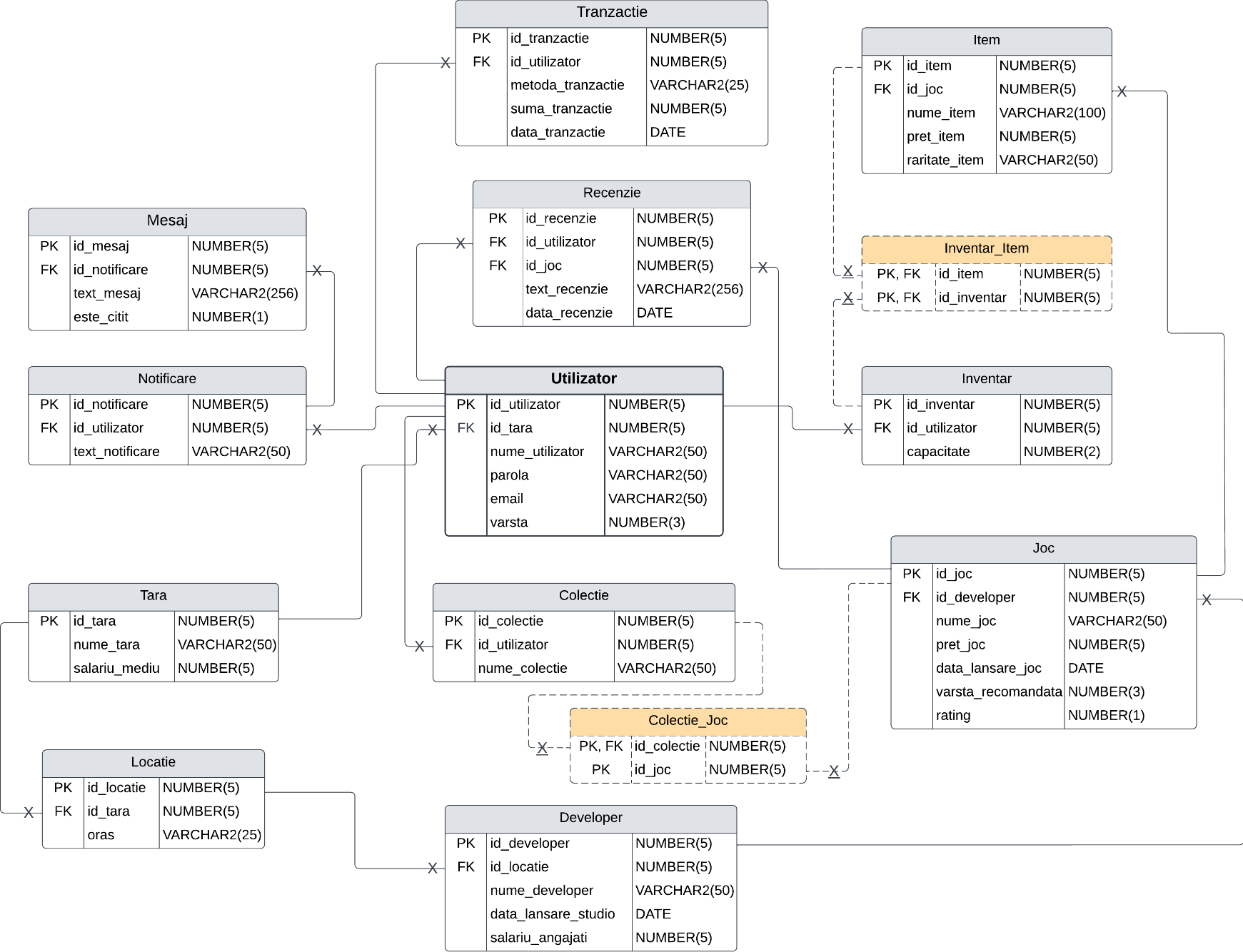
**Inventar\_Item**

* id\_item (PK, FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul itemului.
* id\_inventar (PK, FK) – Variabilă de tip întreg de lungime maxim 5, reprezentând identificatorul inventarului.

# DIAGRAMA ENTITATE-RELAȚIE



# DIAGRAMA CONCEPTUALĂ



# SCHEME RELAȚIONALE

**Utilizator** (**id\_utilizator#, id\_tara#**, *nume\_utilizator, parola, email, varsta*)

**Recenzie** (**id\_recenzie#, id\_utilizator#, id\_joc#,** *text\_recenzie, data\_recenzie*)

**Tranzactie** (**id\_tranzactie#, id\_utilizator#**, *metoda\_tranzactie, suma\_tranzactie, data\_tranzactie*)

**Notificare** (**id\_notificare#, id\_utilizator#,** *text\_notificare*)

**Mesaj** (**id\_mesaj#, id\_notificare#,** *text\_mesaj, este\_citit*)

**Tara** (**id\_tara#,** *nume\_tara, salariu\_mediu*)

**Locatie** (**id\_locatie#, id\_tara#**, *oras*)

**Developer** (**id\_developer#, id\_locatie*#,*** *nume\_developer, data\_lansare\_studio, salariu\_angajati*)

**Colectie** (**id\_colectie#, id\_utilizator#,** *nume\_colectie*)

**Joc** (**id\_joc#, id\_developer#**, *nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating*)

**Colectie\_Joc** (**id\_colectie#, id\_joc#**)

**Inventar** (**id\_inventar#, id\_utilizator#,** *capacitate*)

**Item** (**id\_item#, id\_joc#,** *nume\_item, pret\_item, raritate\_item*)

**Inventar\_Item** (**id\_item#, id\_inventar#**)

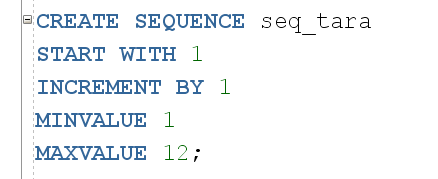
# REALIZAREA NORMALIZĂRII PÂNĂ LA FN3

Ne aflăm în FN1 deoarece fiecare tabelă prezintă cheie primară, fiecare coloană conține elemente de același tip, iar fiecare rând este unic. Pentru a contrazice FN1, ar fi trebuit sa avem două iteme cu aceleași atribute în inventarul unui utilizator, fiind imposbilă diferențierea între ele în lipsa unei chei (id\_item).

Normalizarea până la FN2 este asigurată de faptul că toate atributele unui tabel depind numai și numai de cheia primară a entității, iar tabelele asociative conțin doar chei primare. Greșit ar fi fost dacă în tabela JOC am fi avut informații despre studioul developer-ului, cum ar fi numele acestuia, întrucât aceasta nu are nicio legatură cu jocul în sine.

FN3 ține de faptul că atributele într-o entitate nu depind unele de celelalte. Dacă în tabela TRANZACTIE am fi avut *suma\_tranzactie* restricționată la *metoda\_tranzactie* (ex: folosind PayPal, poți avea o sumă de doar 20$), atunci nu am fi îndeplinit FN3.

# SECVENȚĂ UTILIZATĂ ÎN INSERAREA ÎNREGISTRĂRILOR ÎN TABEL



*CREATE SEQUENCE seq\_tara*

*START WITH 1*

*INCREMENT BY 1*

*MINVALUE 1*

*MAXVALUE 12;*

# CREAREA TABELELOR ȘI ÎNREGISTRAREA DATELOR

## UTILIZATOR

* CREATE

CREATE TABLE Utilizator (

id\_utilizator NUMBER(5) PRIMARY KEY,

id\_tara NUMBER(5) NOT NULL,

nume\_utilizator VARCHAR2(50) NOT NULL,

parola VARCHAR2(50) NOT NULL,

email VARCHAR2(50) NOT NULL UNIQUE,

varsta NUMBER(3) NOT NULL CHECK (varsta >= 0),

CONSTRAINT fk\_utilizator\_tara FOREIGN KEY (id\_tara) REFERENCES Tara(id\_tara)

);

DESCRIBE Utilizator;

* INSERT

INSERT INTO Utilizator (id\_utilizator, nume\_utilizator, parola, email, varsta, id\_tara)

VALUES (1, 'QuantumLeap', 'timeTravel42', 'alex.ramirez@protonmail.com', 18, 1); -- SUA

INSERT INTO Utilizator (id\_utilizator, nume\_utilizator, parola, email, varsta, id\_tara)

VALUES (2, 'GlitchWizard', 'codeMancer88', 'nina.schwartz@zoho.com', 19, 7); -- Suedia

INSERT INTO Utilizator (id\_utilizator, nume\_utilizator, parola, email, varsta, id\_tara)

VALUES (3, 'NeonRider', 'speedOfLight77', 'leon.tang@yandex.com', 21, 2); -- Japonia

INSERT INTO Utilizator (id\_utilizator, nume\_utilizator, parola, email, varsta, id\_tara)

VALUES (4, 'PixelHunter', 'sharpShooter89', 'marco.valenti@gmail.com', 12, 1); -- SUA

INSERT INTO Utilizator (id\_utilizator, nume\_utilizator, parola, email, varsta, id\_tara)

VALUES (5, 'FrostMage', 'chillFactor0', 'sarah.muller@web.de', 23, 10); -- Germania

INSERT INTO Utilizator (id\_utilizator, nume\_utilizator, parola, email, varsta, id\_tara)

VALUES (6, 'StormCaster', 'stormyWeather22', 'emily.dubois@yahoo.fr', 18, 4); -- Franta

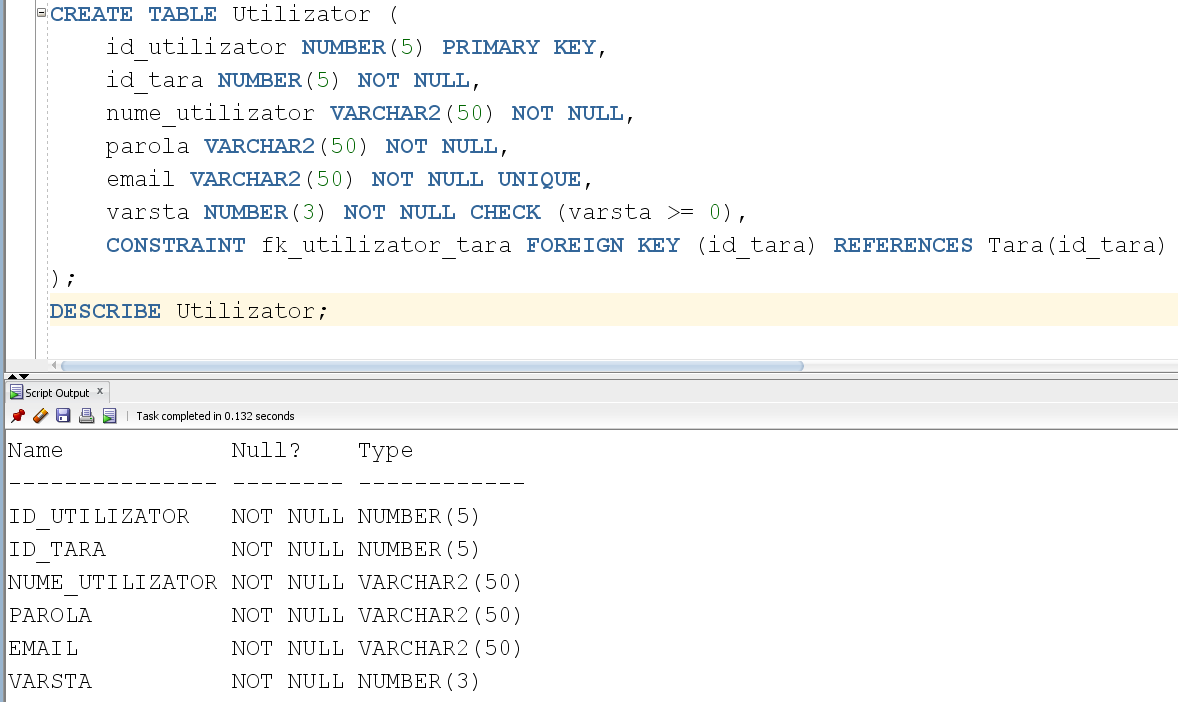
INSERT INTO Utilizator (id\_utilizator, nume\_utilizator, parola, email, varsta, id\_tara)

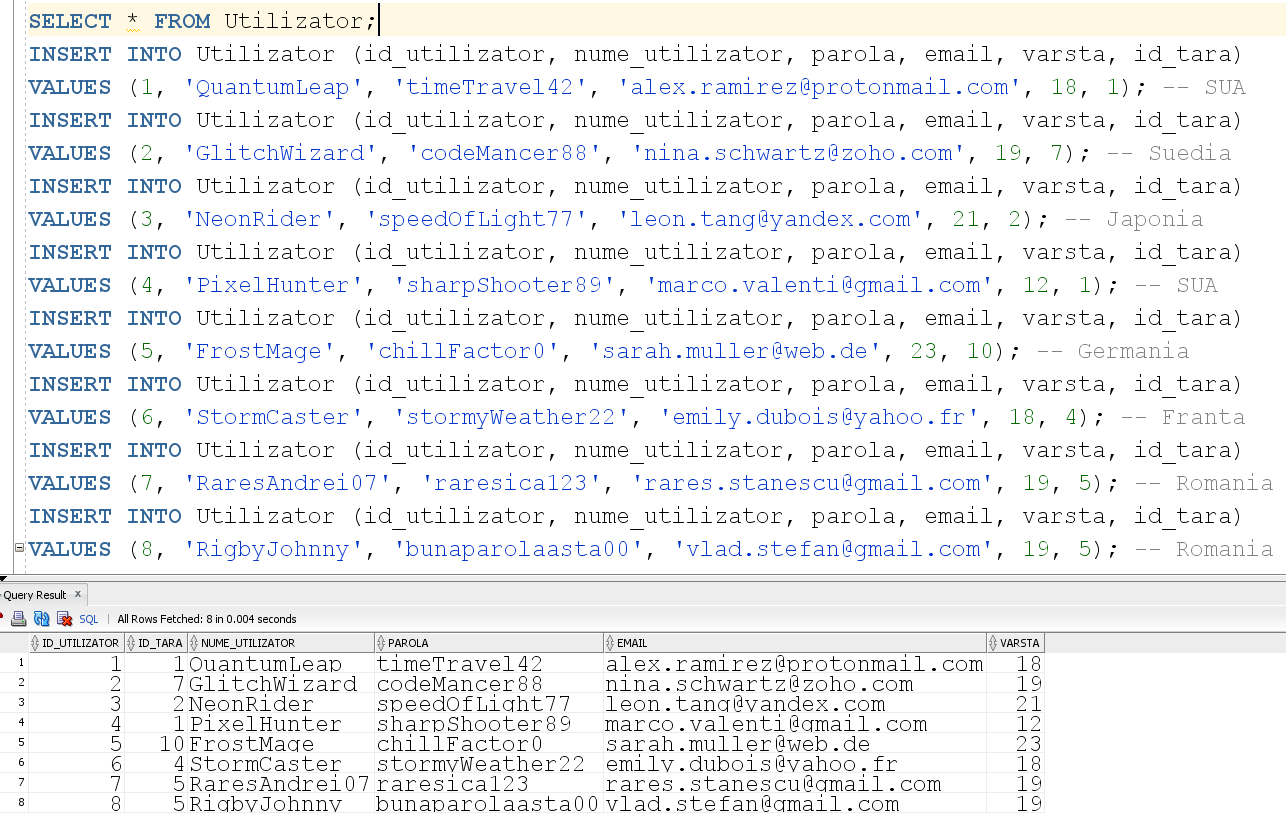
VALUES (7, 'RaresAndrei07', 'raresica123', 'rares.stanescu@gmail.com', 19, 5); -- Romania

INSERT INTO Utilizator (id\_utilizator, nume\_utilizator, parola, email, varsta, id\_tara)

VALUES (8, 'RigbyJohnny', 'bunaparolaasta00', 'vlad.stefan@gmail.com', 19, 5); -- Romania

SELECT \* FROM Utilizator;



****

## NOTIFICARE

* CREATE

CREATE TABLE Notificare (

id\_notificare NUMBER(5) PRIMARY KEY,

id\_utilizator NUMBER(5) NOT NULL,

text\_notificare VARCHAR2(50) NOT NULL,

CONSTRAINT fk\_notificare\_utilizator FOREIGN KEY (id\_utilizator) REFERENCES Utilizator(id\_utilizator)

);

DESCRIBE Notificare;

* INSERT

SELECT \* FROM Notificare;

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (1, 1, 'Noutati in cont. VERDE');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (2, 1, 'Actualizare importanta. PORTOCALIU');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (3, 2, 'Alerta de securitate. ROSU');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (4, 2, 'Promotii noi. VERDE');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (5, 3, 'Feedback procesat. VERDE');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (6, 4, 'Termeni actualizati. PORTOCALIU');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (7, 5, 'Cont expira curand. PORTOCALIU');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (8, 6, 'Eligibil pentru upgrade. VERDE');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (9, 7, 'Oferta speciala expira. PORTOCALIU');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (10, 8, 'Suspiciune de frauda. ROSU');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (11, 3, 'Joc preferat actualizat. VERDE');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (12, 3, 'Reduceri de weekend. PORTOCALIU');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (13, 4, 'Alerta securitate. ROSU');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (14, 4, 'Invitatie beta test. VERDE');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (15, 5, 'Abonament expira curand. PORTOCALIU');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (16, 6, 'Cadou in inventar. VERDE');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (17, 7, 'Sondaj comunitate. VERDE');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

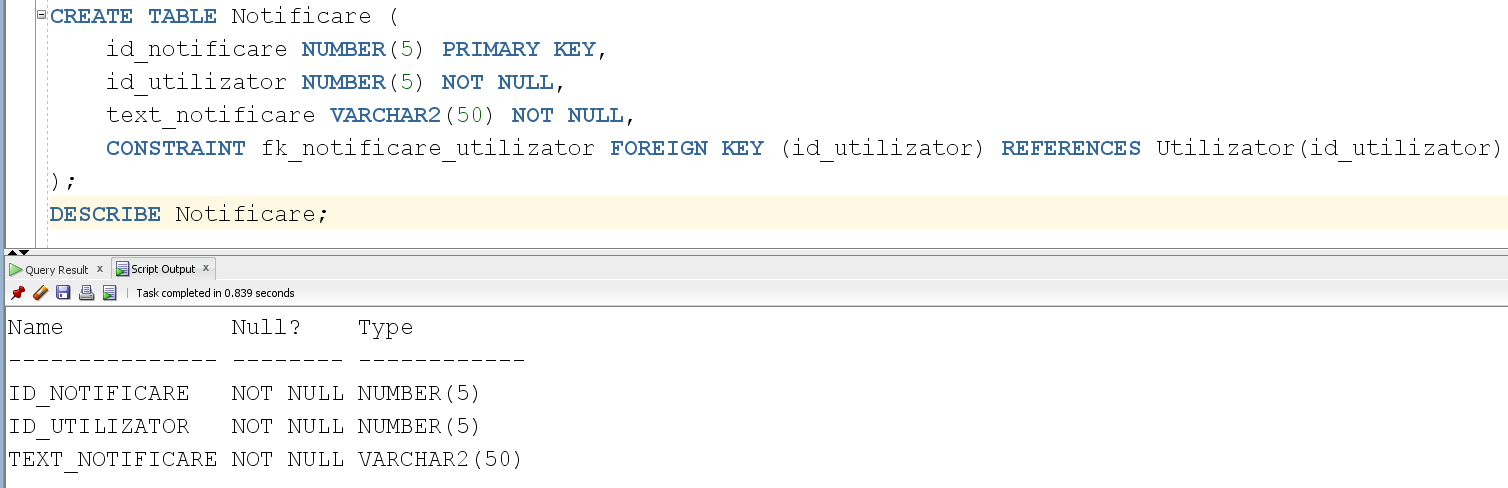
VALUES (18, 7, 'Update joc efectuat. VERDE');

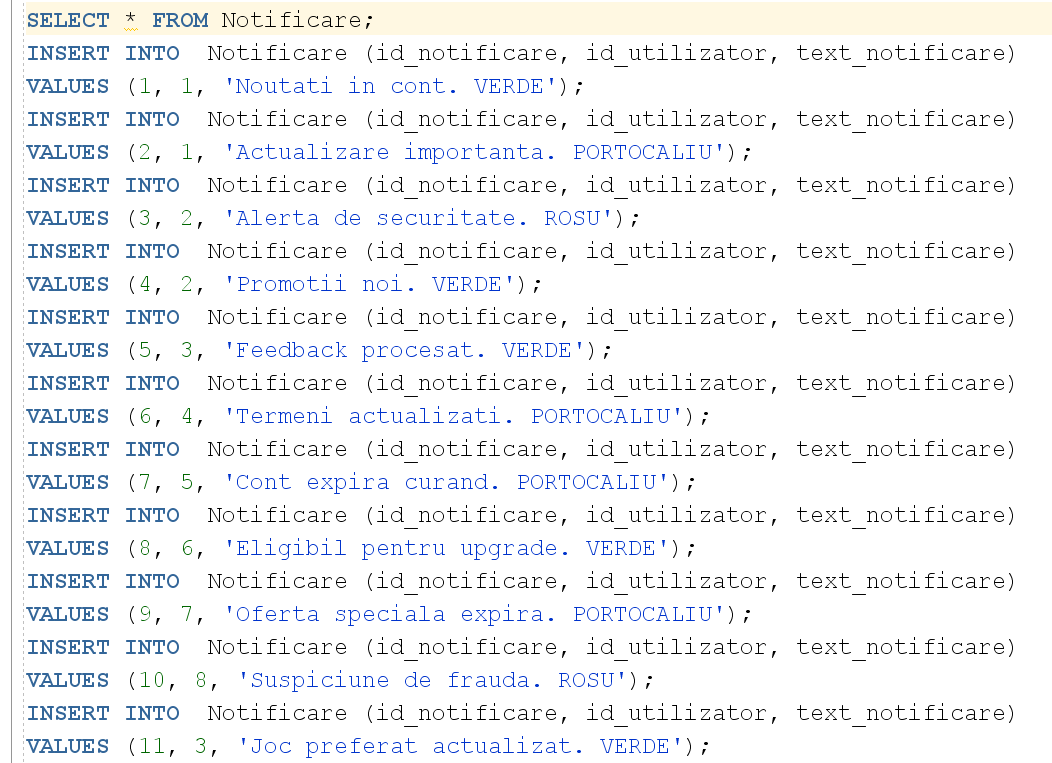
INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

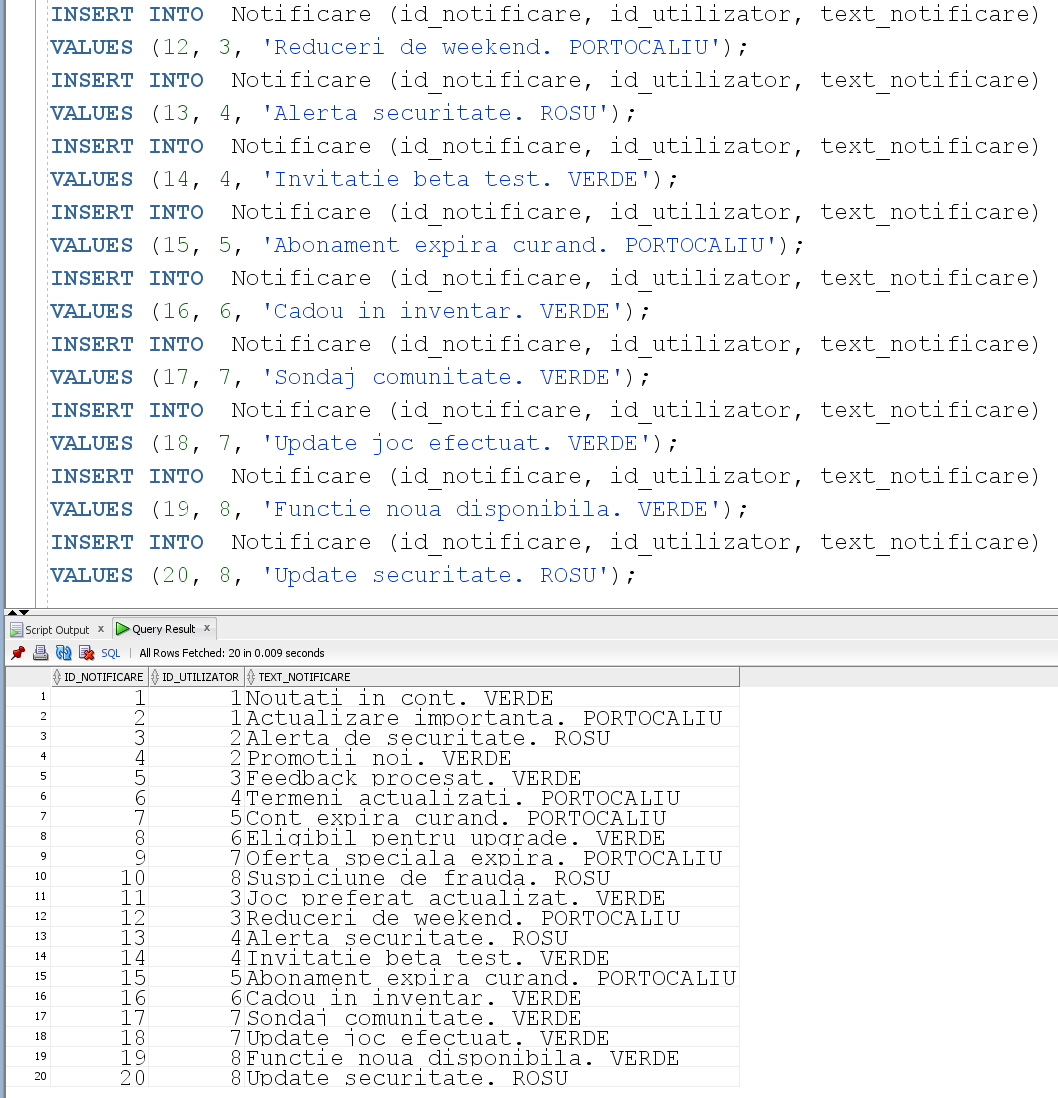
VALUES (19, 8, 'Functie noua disponibila. VERDE');

INSERT INTO Notificare (id\_notificare, id\_utilizator, text\_notificare)

VALUES (20, 8, 'Update securitate. ROSU');







## MESAJ

* CREATE

CREATE TABLE Mesaj (

id\_mesaj NUMBER(5) PRIMARY KEY,

id\_notificare NUMBER(5) NOT NULL,

text\_mesaj VARCHAR2(256) NOT NULL,

este\_citit NUMBER(1) NOT NULL CHECK (este\_citit IN (0, 1)),

CONSTRAINT fk\_mesaj\_notificare FOREIGN KEY (id\_notificare) REFERENCES Notificare(id\_notificare)

);

DESCRIBE Mesaj;

* INSERT

SELECT \* FROM Mesaj;

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (1, 1, 'Ai noutati in contul tau. Verifica-le acum!', 0);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (2, 2, 'O actualizare importanta este disponibila pentru tine.', 1);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (3, 3, 'Alerta de securitate! Verifica activitatea recenta.', 0);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (4, 4, 'Profita de promotiile noi adaugate recent.', 1);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (5, 5, 'Feedback-ul tau a fost procesat cu succes.', 0);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (6, 6, 'Termenii si conditiile au fost actualizati.', 1);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (7, 7, 'Atentie, contul tau va expira in curand.', 0);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (8, 8, 'Esti eligibil pentru un upgrade. Detalii in cont.', 1);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (9, 9, 'Ultima sansa sa profiti de oferta speciala.', 0);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (10, 10, 'Suspiciune de frauda detectata. Actioneaza imediat!', 1);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (11, 11, 'Jocul tau preferat a fost actualizat recent.', 0);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (12, 12, 'Reduceri de weekend disponibile acum.', 1);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (13, 13, 'Alerta de securitate majora! Verifica imediat.', 0);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (14, 14, 'Ai fost invitat la un beta test. Inscrie-te acum.', 1);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (15, 15, 'Abonamentul tau este pe cale sa expire. Reinnoieste-l.', 0);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (16, 16, 'Verifica inventarul tau pentru un cadou surpriza.', 1);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (17, 17, 'Participa la sondajul nostru si castiga puncte.', 0);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

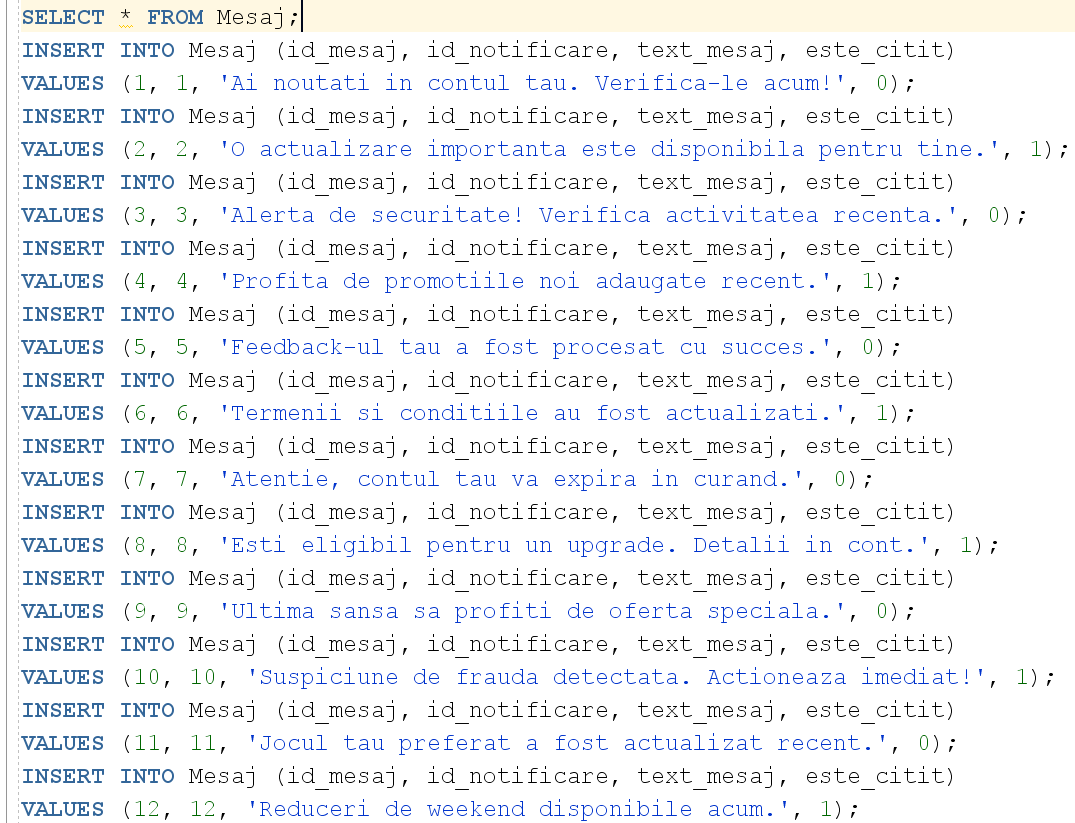
VALUES (18, 18, 'Update-ul jocului tau a fost efectuat cu succes.', 1);

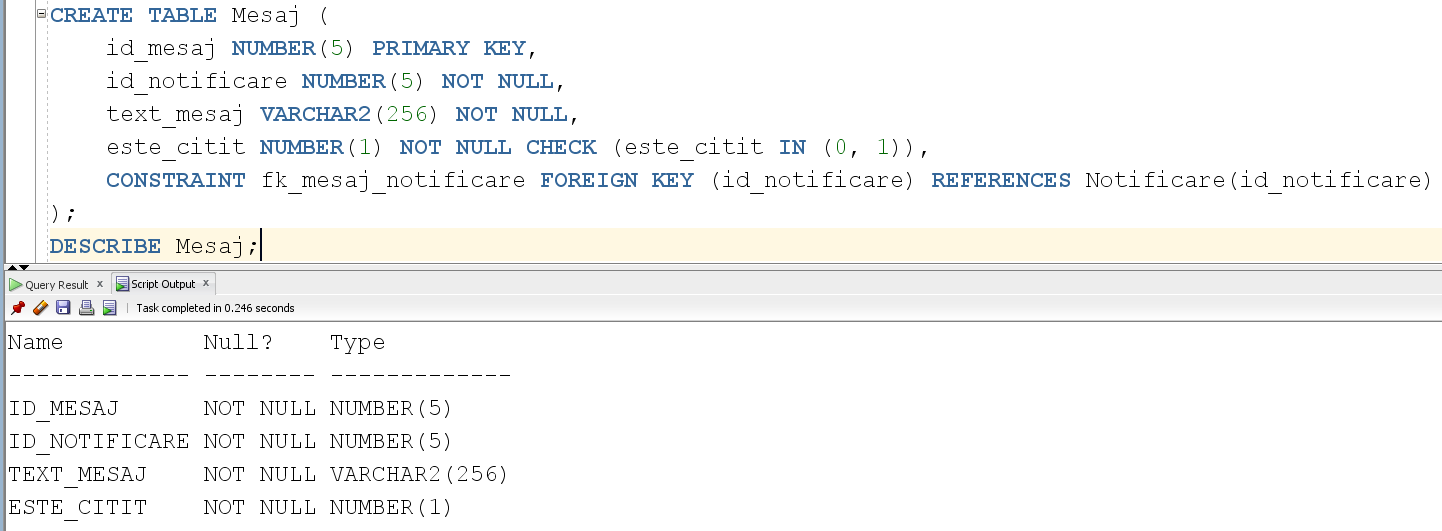
INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

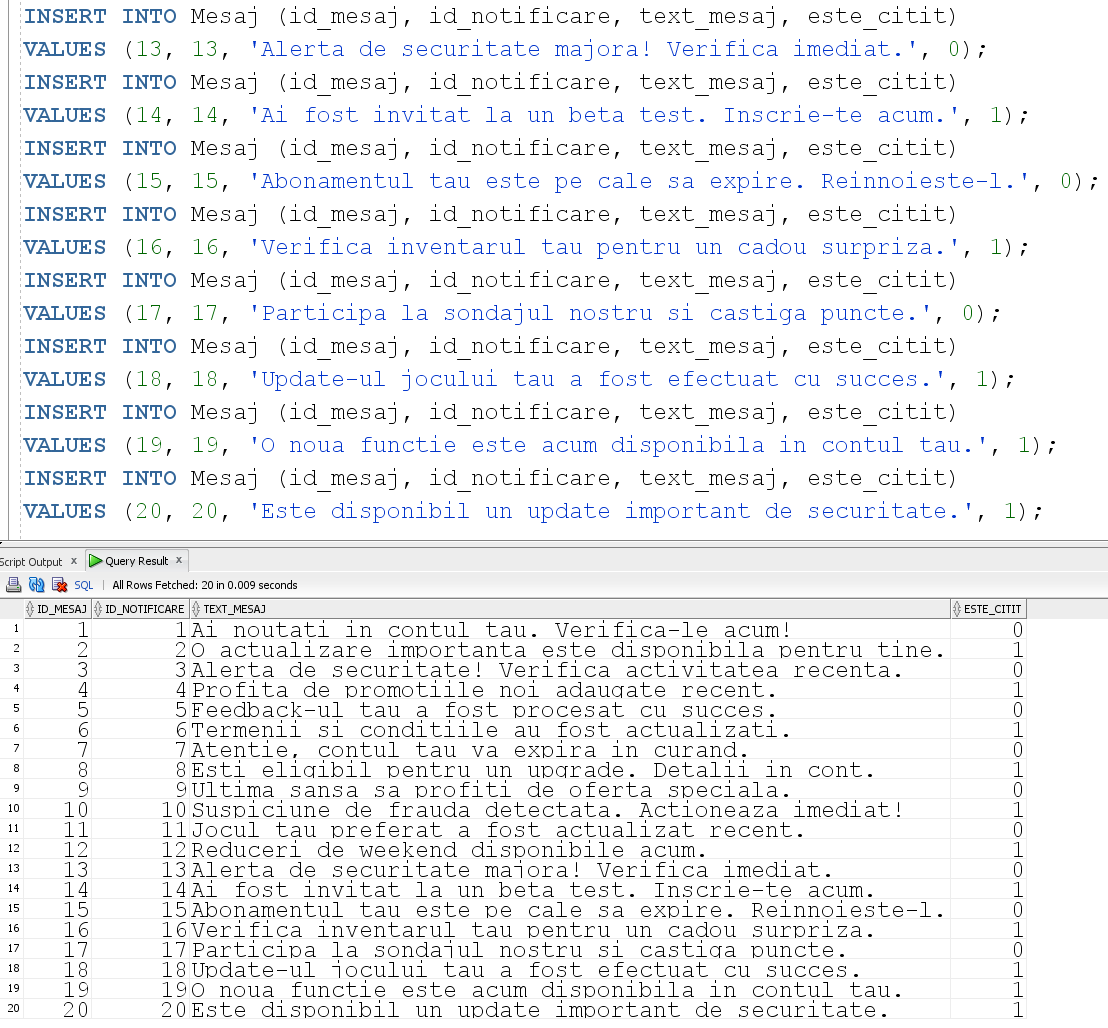
VALUES (19, 19, 'O noua functie este acum disponibila in contul tau.', 1);

INSERT INTO Mesaj (id\_mesaj, id\_notificare, text\_mesaj, este\_citit)

VALUES (20, 20, 'Este disponibil un update important de securitate.', 1);







## TARA

* CREATE

CREATE TABLE Tara (

id\_tara NUMBER(5) PRIMARY KEY,

nume\_tara VARCHAR2(50) NOT NULL UNIQUE,

salariu\_mediu NUMBER(5) NOT NULL CHECK (salariu\_mediu >= 0)

);

DESCRIBE Tara;

* INSERT

CREATE SEQUENCE seq\_tara

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 12;

INSERT INTO Tara (id\_tara, nume\_tara, salariu\_mediu)

VALUES (seq\_tara.nextval, 'SUA', 55000);

INSERT INTO Tara (id\_tara, nume\_tara, salariu\_mediu)

VALUES (seq\_tara.nextval, 'Japonia', 40000);

INSERT INTO Tara (id\_tara, nume\_tara, salariu\_mediu)

VALUES (seq\_tara.nextval, 'Canada', 45000);

INSERT INTO Tara (id\_tara, nume\_tara, salariu\_mediu)

VALUES (seq\_tara.nextval, 'Franta', 43000);

INSERT INTO Tara (id\_tara, nume\_tara, salariu\_mediu)

VALUES (seq\_tara.nextval, 'Romania', 12000);

INSERT INTO Tara (id\_tara, nume\_tara, salariu\_mediu)

VALUES (seq\_tara.nextval, 'Polonia', 27000);

INSERT INTO Tara (id\_tara, nume\_tara, salariu\_mediu)

VALUES (seq\_tara.nextval, 'Suedia', 42000);

INSERT INTO Tara (id\_tara, nume\_tara, salariu\_mediu)

VALUES (seq\_tara.nextval, 'Rusia', 15000);

INSERT INTO Tara (id\_tara, nume\_tara, salariu\_mediu)

VALUES (seq\_tara.nextval, 'Estonia', 23000);

INSERT INTO Tara (id\_tara, nume\_tara, salariu\_mediu)

VALUES (seq\_tara.nextval, 'Germania', 65000);

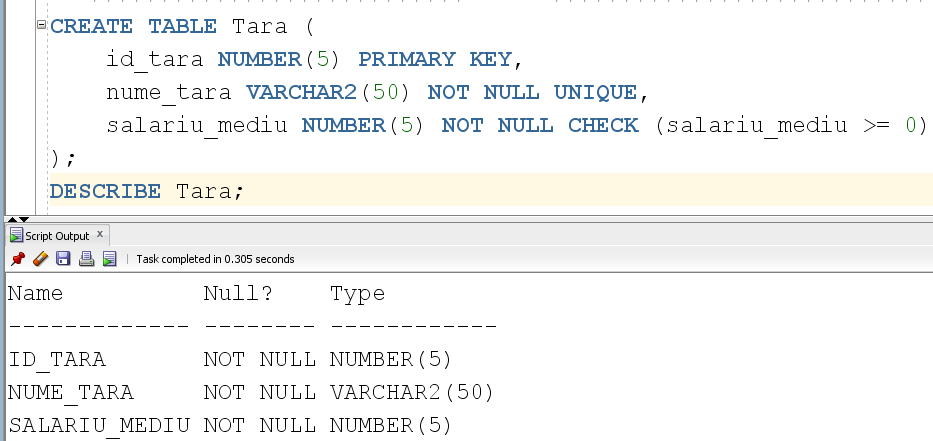
INSERT INTO Tara (id\_tara, nume\_tara, salariu\_mediu)

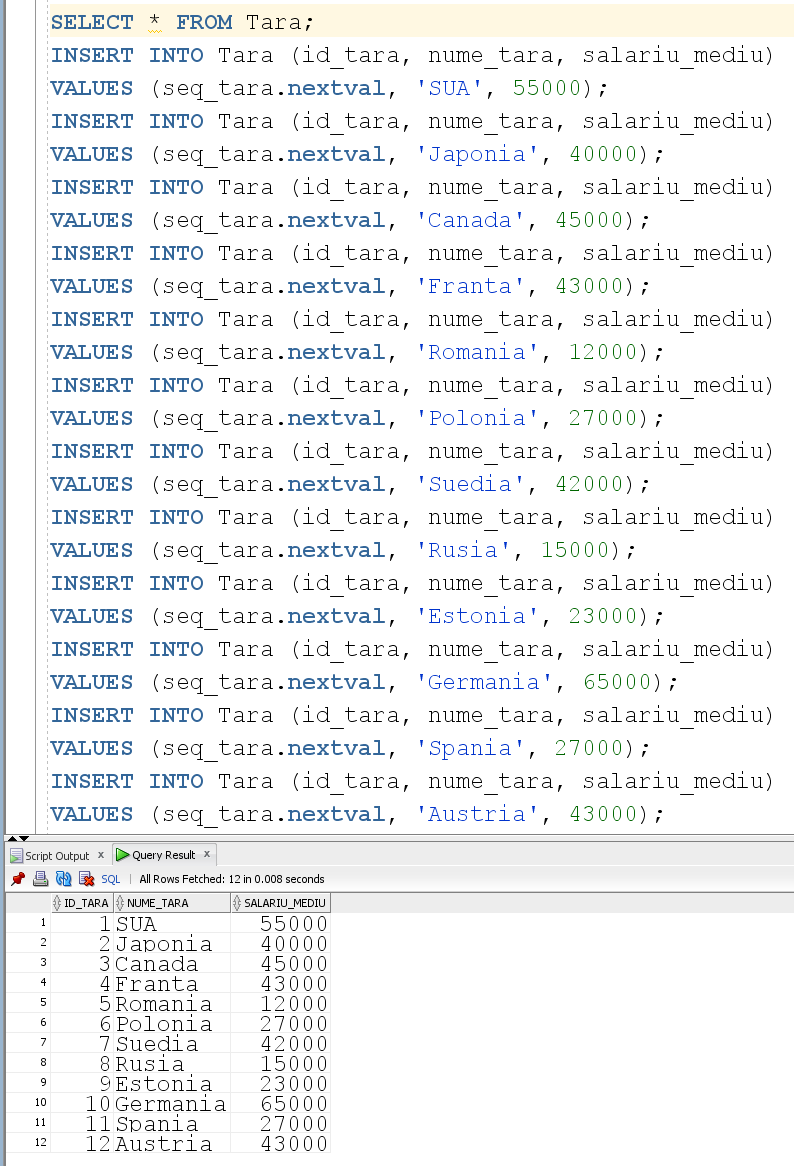
VALUES (seq\_tara.nextval, 'Spania', 27000);

INSERT INTO Tara (id\_tara, nume\_tara, salariu\_mediu)

VALUES (seq\_tara.nextval, 'Austria', 43000);

SELECT \* FROM Tara;





## LOCATIE

* CREATE

CREATE TABLE Locatie (

id\_locatie NUMBER(5) PRIMARY KEY,

id\_tara NUMBER(5) NOT NULL,

oras VARCHAR2(25) NOT NULL UNIQUE,

CONSTRAINT fk\_locatie\_tara FOREIGN KEY (id\_tara) REFERENCES Tara(id\_tara)

);

DESCRIBE Locatie;

* INSERT

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (1, 3, 'Montreal'); -- Canada, Ubisoft Montreal

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (2, 1, 'Redmond'); -- SUA, 343 Industries

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (3, 6, 'Varsovia'); -- Polonia, CD Projekt Red

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (4, 1, 'Santa Monica'); -- SUA, Naughty Dog

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (5, 2, 'Tokyo'); -- Japonia, Kojima Productions

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (6, 1, 'Bellevue'); -- SUA, Valve

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (7, 1, 'New York'); -- SUA, Rockstar Games

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (8, 1, 'Dallas'); -- SUA, id Software

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (9, 1, 'Frisco'); -- SUA, Gearbox Software

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (10, 7, 'Stockholm'); -- Suedia, Mojang Studios

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (11, 3, 'Vancouver'); -- Canada, EA Vancouver, FIFA23

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (12, 5, 'Bucuresti'); -- Romania, EAFC 24

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (13, 8, 'Moscova'); -- Rusia

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (14, 9, 'Tallinn'); -- Estonia

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (15, 10, 'Koln'); -- Germania

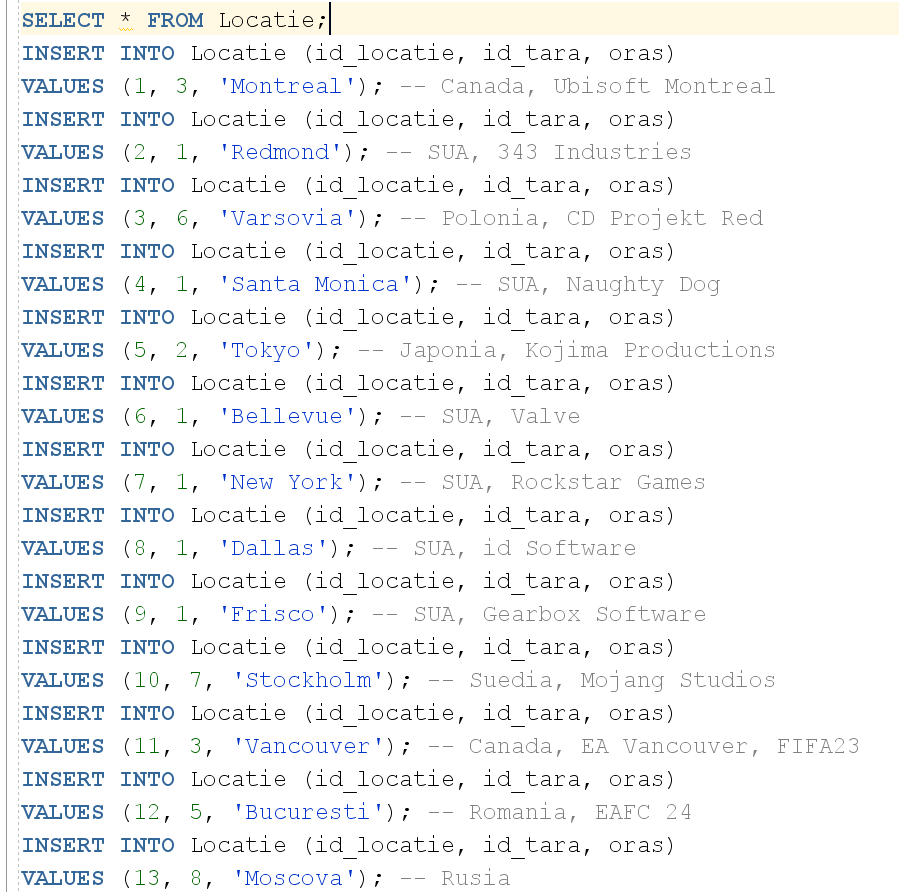
INSERT INTO Locatie (id\_locatie, id\_tara, oras)

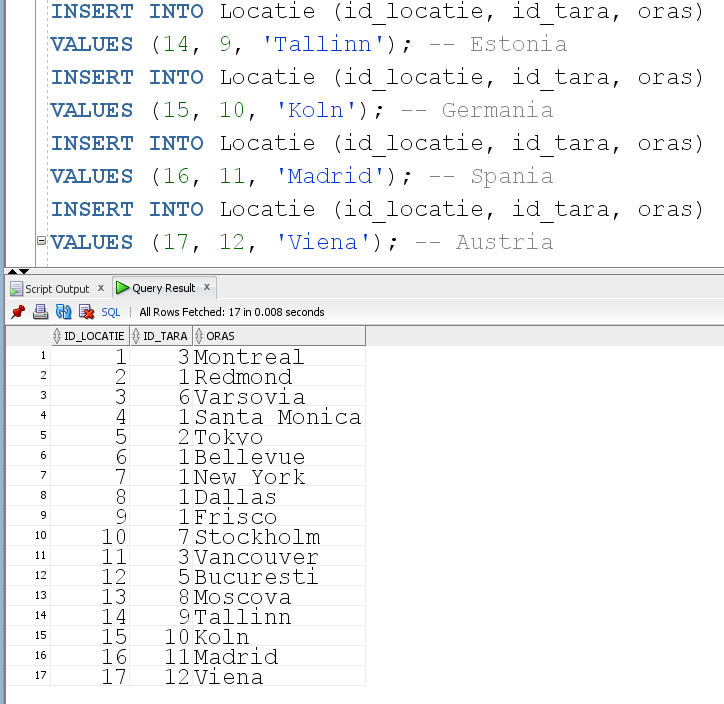
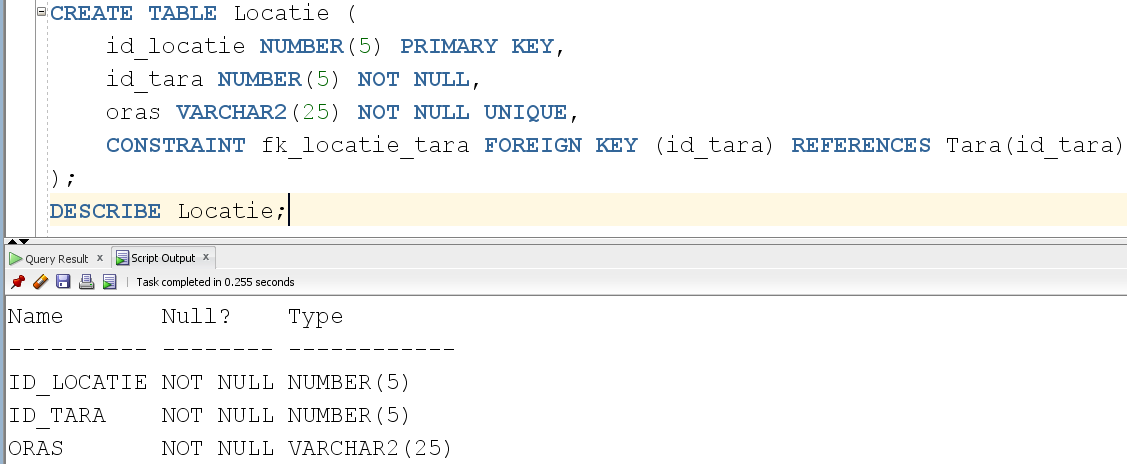
VALUES (16, 11, 'Madrid'); -- Spania

INSERT INTO Locatie (id\_locatie, id\_tara, oras)

VALUES (17, 12, 'Viena'); -- Austria

SELECT \* FROM Locatie;



## DEVELOPER

* CREATE

CREATE TABLE Developer (

id\_developer NUMBER(5) PRIMARY KEY,

id\_locatie NUMBER(5),

nume\_developer VARCHAR2(50) NOT NULL,

data\_lansare\_studio DATE NOT NULL,

salariu\_angajati NUMBER(5) NOT NULL CHECK (salariu\_angajati > 0),

CONSTRAINT fk\_developer\_locatie FOREIGN KEY (id\_locatie) REFERENCES Locatie(id\_locatie)

);

DESCRIBE Developer;

* INSERT

SELECT \* FROM Developer;

INSERT INTO Developer (id\_developer, id\_locatie, nume\_developer, data\_lansare\_studio, salariu\_angajati)

VALUES (1, 1, 'Ubisoft Montreal', TO\_DATE('1997-01-01', 'YYYY-MM-DD'), 44000); -- Sub medie in Canada

INSERT INTO Developer (id\_developer, id\_locatie, nume\_developer, data\_lansare\_studio, salariu\_angajati)

VALUES (2, 2, '343 Industries', TO\_DATE('2007-05-01', 'YYYY-MM-DD'), 53000); -- Sub medie in SUA

INSERT INTO Developer (id\_developer, id\_locatie, nume\_developer, data\_lansare\_studio, salariu\_angajati)

VALUES (3, 3, 'CD Projekt Red', TO\_DATE('2002-06-01', 'YYYY-MM-DD'), 26000); -- Sub medie in Polonia

INSERT INTO Developer (id\_developer, id\_locatie, nume\_developer, data\_lansare\_studio, salariu\_angajati)

VALUES (4, 4, 'Naughty Dog', TO\_DATE('1984-09-01', 'YYYY-MM-DD'), 60000); -- Peste medie in SUA

INSERT INTO Developer (id\_developer, id\_locatie, nume\_developer, data\_lansare\_studio, salariu\_angajati)

VALUES (5, 5, 'Kojima Productions', TO\_DATE('2005-12-01', 'YYYY-MM-DD'), 45000); -- Peste medie in Japonia

INSERT INTO Developer (id\_developer, id\_locatie, nume\_developer, data\_lansare\_studio, salariu\_angajati)

VALUES (6, 6, 'Valve Corporation', TO\_DATE('1996-08-01', 'YYYY-MM-DD'), 70000); -- Peste medie in SUA

INSERT INTO Developer (id\_developer, id\_locatie, nume\_developer, data\_lansare\_studio, salariu\_angajati)

VALUES (7, 7, 'Rockstar Games', TO\_DATE('1998-12-01', 'YYYY-MM-DD'), 65000); -- Peste medie in SUA

INSERT INTO Developer (id\_developer, id\_locatie, nume\_developer, data\_lansare\_studio, salariu\_angajati)

VALUES (8, 8, 'id Software', TO\_DATE('1991-02-01', 'YYYY-MM-DD'), 62000); -- Peste medie in SUA

INSERT INTO Developer (id\_developer, id\_locatie, nume\_developer, data\_lansare\_studio, salariu\_angajati)

VALUES (9, 9, 'Gearbox Software', TO\_DATE('1999-02-01', 'YYYY-MM-DD'), 58000); -- Peste medie in SUA

INSERT INTO Developer (id\_developer, id\_locatie, nume\_developer, data\_lansare\_studio, salariu\_angajati)

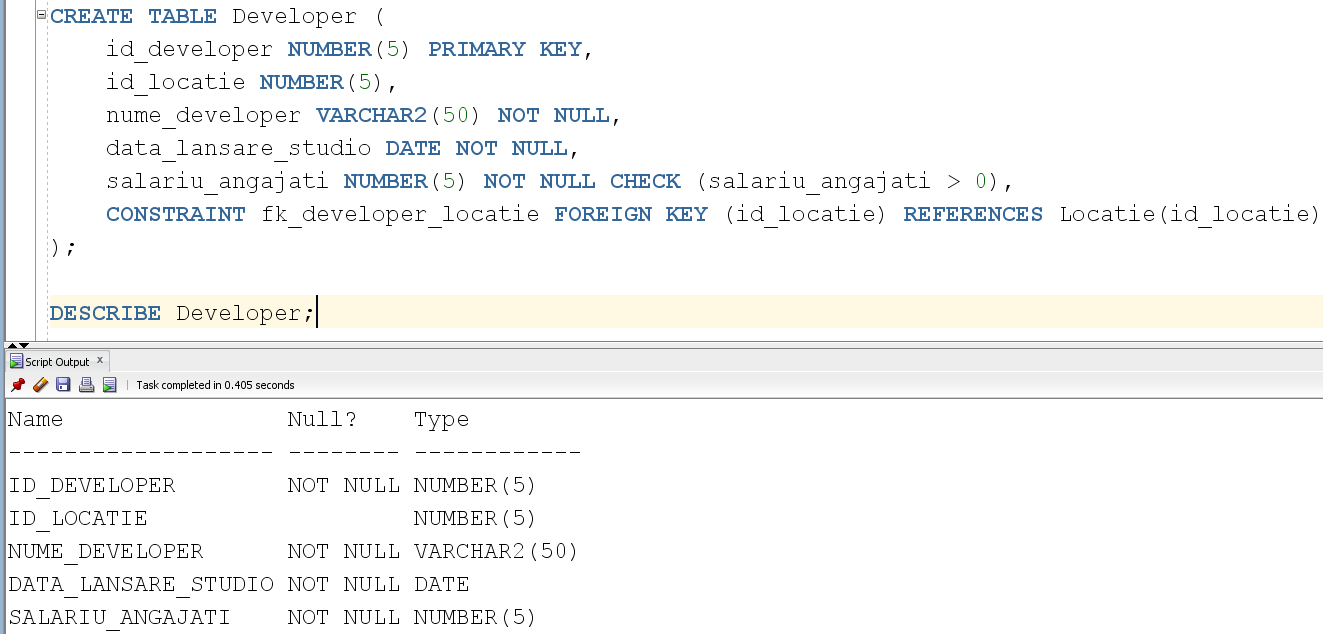
VALUES (10, 10, 'Mojang Studios', TO\_DATE('2009-05-01', 'YYYY-MM-DD'), 50000); -- Peste medie in Suedia

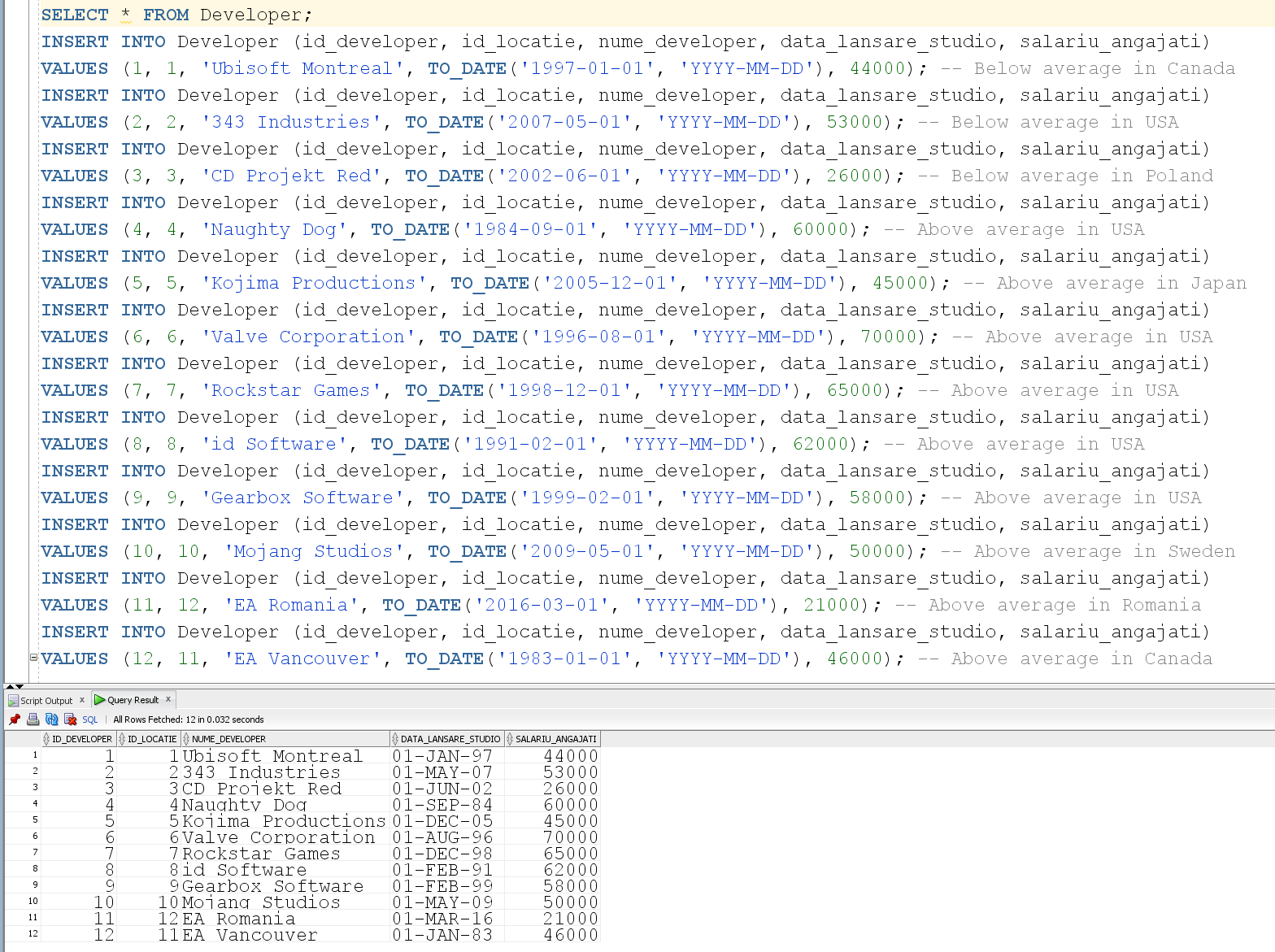
INSERT INTO Developer (id\_developer, id\_locatie, nume\_developer, data\_lansare\_studio, salariu\_angajati)

VALUES (11, 12, 'EA Romania', TO\_DATE('2016-03-01', 'YYYY-MM-DD'), 21000); -- Peste medie in Romania

INSERT INTO Developer (id\_developer, id\_locatie, nume\_developer, data\_lansare\_studio, salariu\_angajati)

VALUES (12, 11, 'EA Vancouver', TO\_DATE('1983-01-01', 'YYYY-MM-DD'), 46000); -- Peste medie in Canada





## JOC

* CREATE

CREATE TABLE Joc (

id\_joc NUMBER(5) PRIMARY KEY,

id\_developer NUMBER(5) NOT NULL,

nume\_joc VARCHAR2(50) NOT NULL UNIQUE,

pret\_joc NUMBER(5) NOT NULL CHECK (pret\_joc >= 0),

data\_lansare\_joc DATE NOT NULL,

varsta\_recomandata NUMBER(3) CHECK (varsta\_recomandata >= 0),

rating NUMBER(1) CHECK (rating BETWEEN 1 AND 5),

CONSTRAINT fk\_joc\_developer FOREIGN KEY (id\_developer) REFERENCES Developer(id\_developer)

);

DESCRIBE Joc;

* INSERT

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (1, 1, 'Assassins Creed Valhalla', 60, TO\_DATE('2020-11-10', 'YYYY-MM-DD'), 18, 4);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (2, 2, 'Halo Infinite', 60, TO\_DATE('2021-12-08', 'YYYY-MM-DD'), 16, 3);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (3, 3, 'Cyberpunk 2077', 60, TO\_DATE('2020-12-10', 'YYYY-MM-DD'), 18, 3);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (4, 4, 'The Last of Us Part II', 60, TO\_DATE('2020-06-19', 'YYYY-MM-DD'), 18, 5);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (5, 5, 'Death Stranding', 60, TO\_DATE('2019-11-08', 'YYYY-MM-DD'), 16, 4);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (6, 6, 'Ghost of Tsushima', 60, TO\_DATE('2020-07-17', 'YYYY-MM-DD'), 18, 4);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (7, 5, 'Monster Hunter World', 60, TO\_DATE('2018-01-26', 'YYYY-MM-DD'), 16, 4);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (8, 7, 'Red Dead Redemption 2', 60, TO\_DATE('2018-10-26', 'YYYY-MM-DD'), 18, 5);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (9, 8, 'Doom Eternal', 60, TO\_DATE('2020-03-20', 'YYYY-MM-DD'), 18, 4);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (10, 9, 'Borderlands 3', 60, TO\_DATE('2019-09-13', 'YYYY-MM-DD'), 18, 4);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (11, 10, 'Minecraft', 30, TO\_DATE('2011-11-18', 'YYYY-MM-DD'), 10, 5);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (12, 11, 'EAFC 24', 60, TO\_DATE('2030-09-30', 'YYYY-MM-DD'), 3, 4);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (13, 12, 'FIFA 23', 60, TO\_DATE('2022-09-30', 'YYYY-MM-DD'), 3, 3);

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (14, 6, 'CS:GO', 0, TO\_DATE('2012-08-21', 'YYYY-MM-DD'), 16, 5);

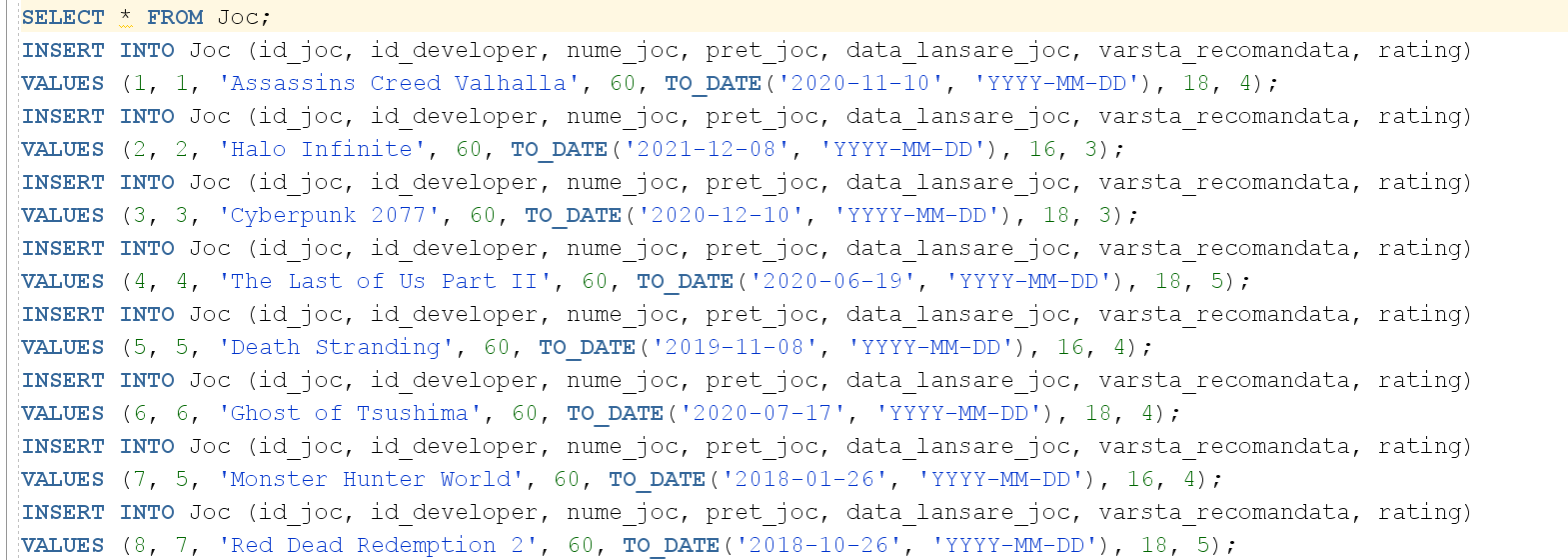
INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

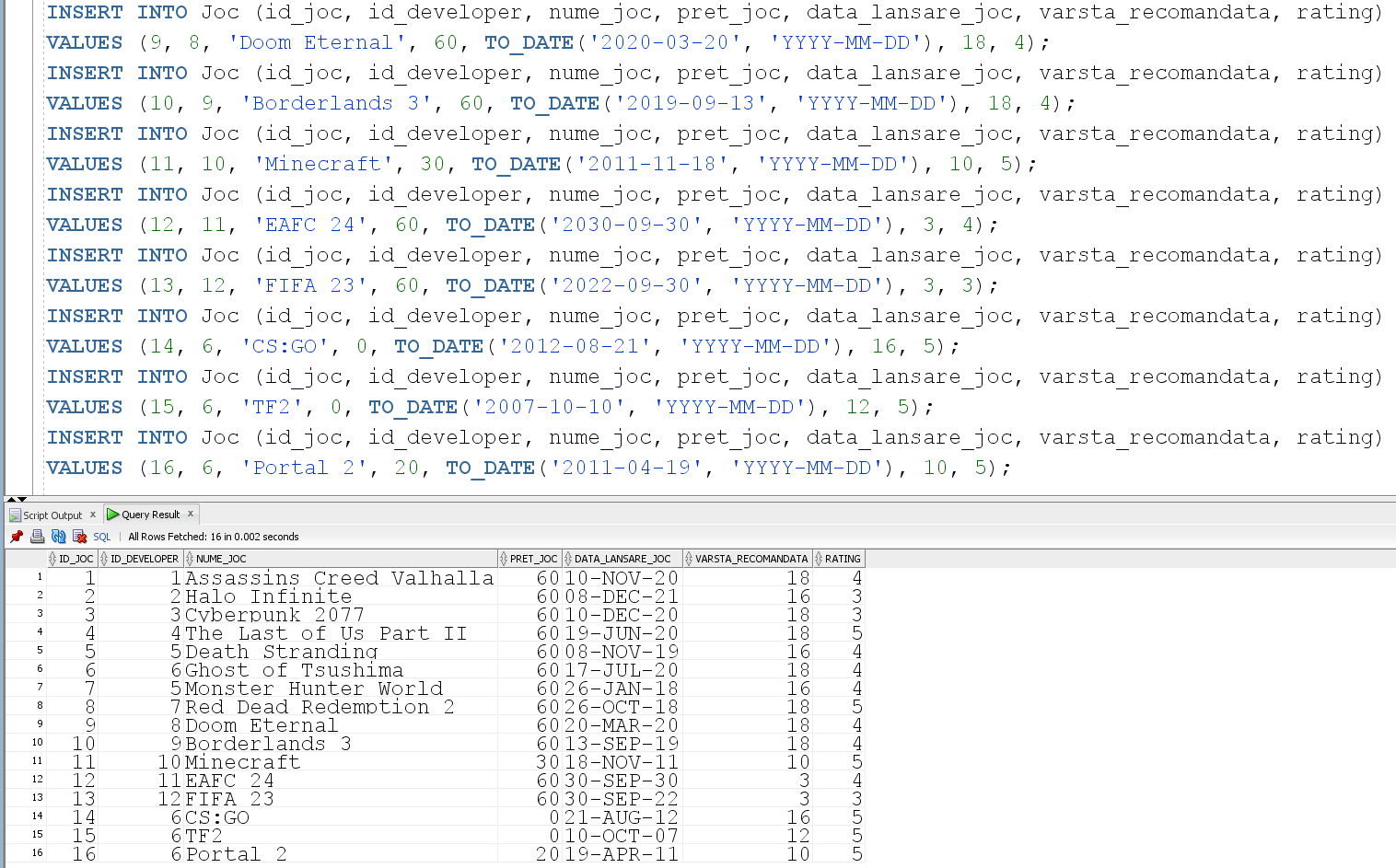
VALUES (15, 6, 'TF2', 0, TO\_DATE('2007-10-10', 'YYYY-MM-DD'), 12, 5);

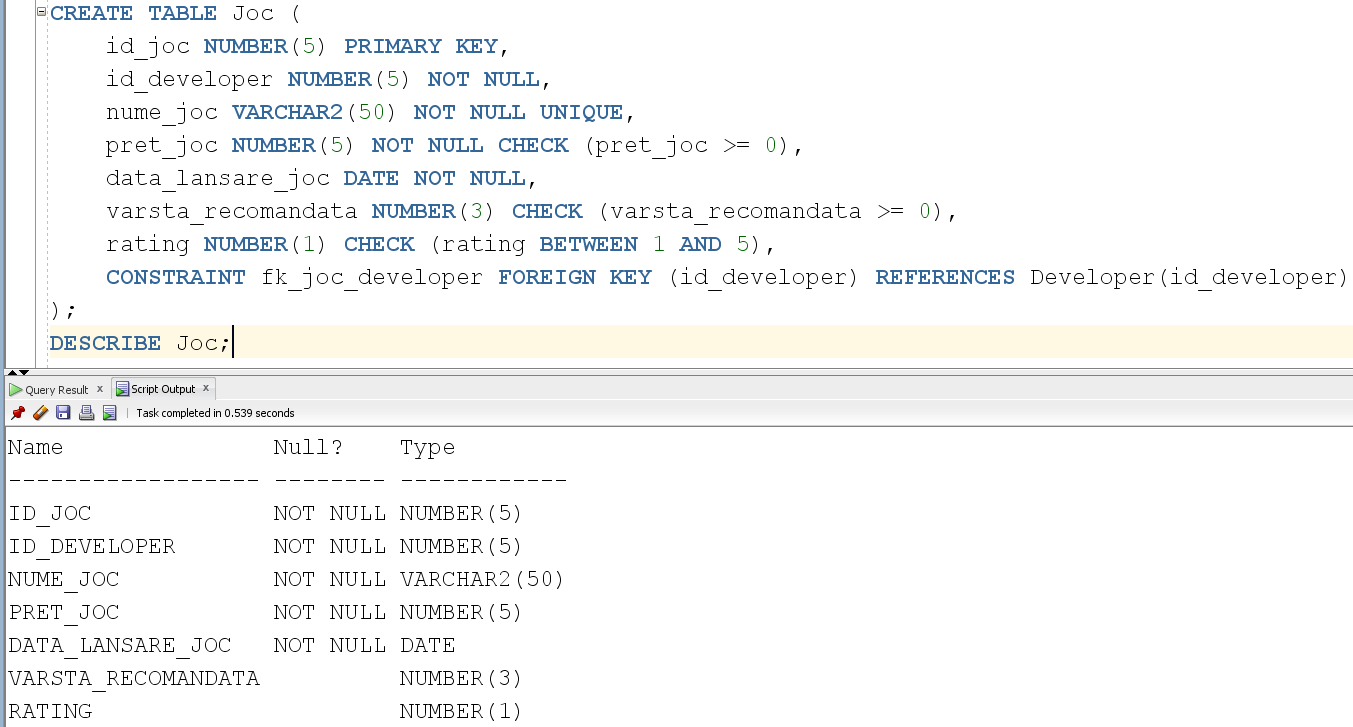
INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (16, 6, 'Portal 2', 20, TO\_DATE('2011-04-19', 'YYYY-MM-DD'), 10, 5);

SELECT \* FROM Joc;







## RECENZIE

* CREATE

CREATE TABLE Recenzie (

id\_recenzie NUMBER(5) PRIMARY KEY,

id\_utilizator NUMBER(5) NOT NULL,

id\_joc NUMBER(5) NOT NULL,

text\_recenzie VARCHAR2(256) NOT NULL,

data\_recenzie DATE NOT NULL,

CONSTRAINT fk\_recenzie\_utilizator FOREIGN KEY (id\_utilizator) REFERENCES Utilizator(id\_utilizator),

CONSTRAINT fk\_recenzie\_joc FOREIGN KEY (id\_joc) REFERENCES Joc(id\_joc)

);

DESCRIBE Recenzie;

* INSERT

SELECT \* FROM Recenzie;

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (1, 1, 1, 'Explorarea in stil viking este detaliata si captivanta. POZITIV', TO\_DATE('2020-08-10', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (2, 5, 2, 'Gameplay-ul este robust, dar campania este scurta. NEGATIV', TO\_DATE('2022-01-15', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (3, 8, 3, 'Viziunea futurista este impresionanta, dar plina de bug-uri. NEGATIV', TO\_DATE('2020-09-10', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (4, 3, 4, 'Poveste profunda si emotionanta. POZITIV', TO\_DATE('2020-12-20', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (5, 4, 5, 'Jocul redefineste genul, dar nu este pentru toata lumea. NEGATIV', TO\_DATE('2019-08-08', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (6, 2, 6, 'Un peisaj deschis superb si lupte detaliate. POZITIV', TO\_DATE('2020-10-17', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (7, 7, 7, 'Complex si detaliat, dar dificil pentru incepatori. NEGATIV', TO\_DATE('2018-07-26', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (8, 6, 8, 'O capodopera narativa cu un ritm lent. POZITIV', TO\_DATE('2018-04-26', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (9, 1, 9, 'Actiune non-stop si grafica impresionanta. POZITIV', TO\_DATE('2020-03-19', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (10, 5, 10, 'Lume uriasa ce asteapta sa fie explorata. POZITIV', TO\_DATE('2019-09-14', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (11, 1, 11, 'Creativitate fara limite si distractie continua. POZITIV', TO\_DATE('2023-10-01', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (12, 5, 12, 'Evolutie impresionanta in simularea fotbalului, gameplay rafinat. POZITIV', TO\_DATE('2030-09-29', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (13, 8, 13, 'Simulare realista a fotbalului, dar cu microtranzactii excesive. NEGATIV', TO\_DATE('2022-10-01', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

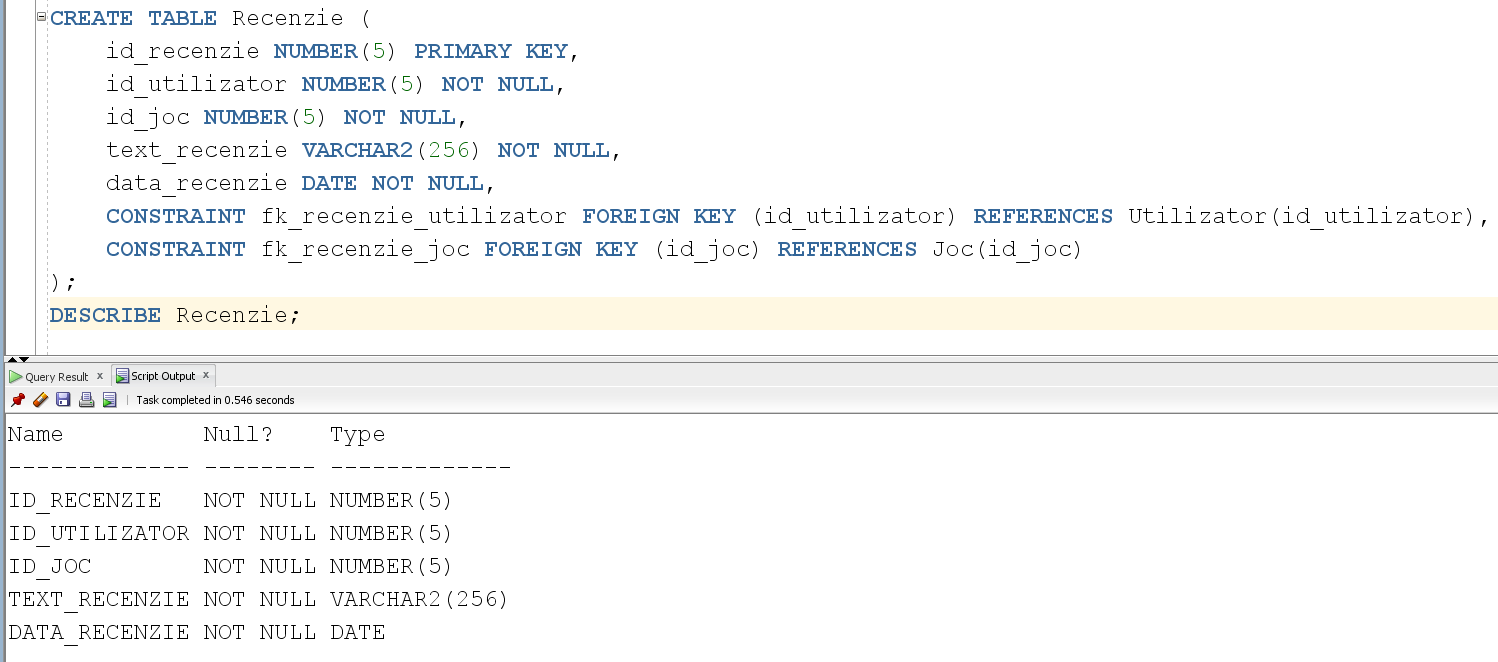
VALUES (14, 1, 14, 'Competitiv si captivant, un clasic al shooterelor. POZITIV', TO\_DATE('2012-09-01', 'YYYY-MM-DD'));

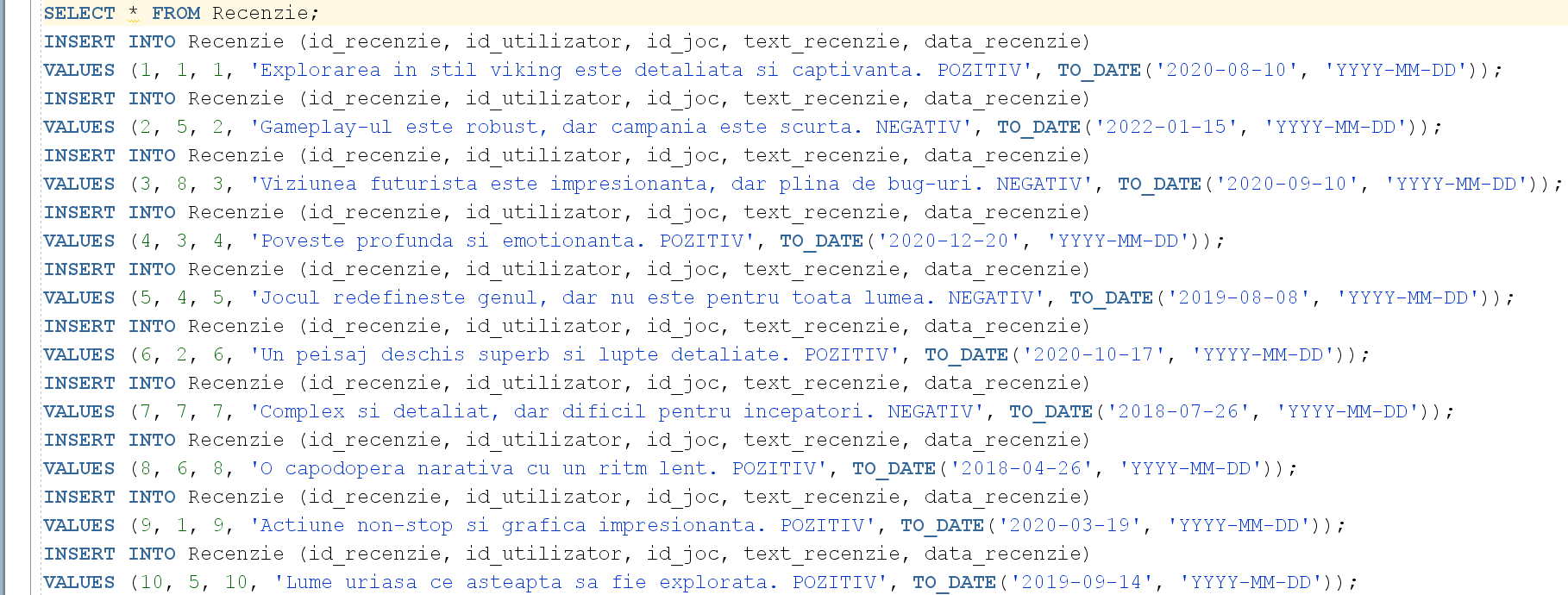
INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

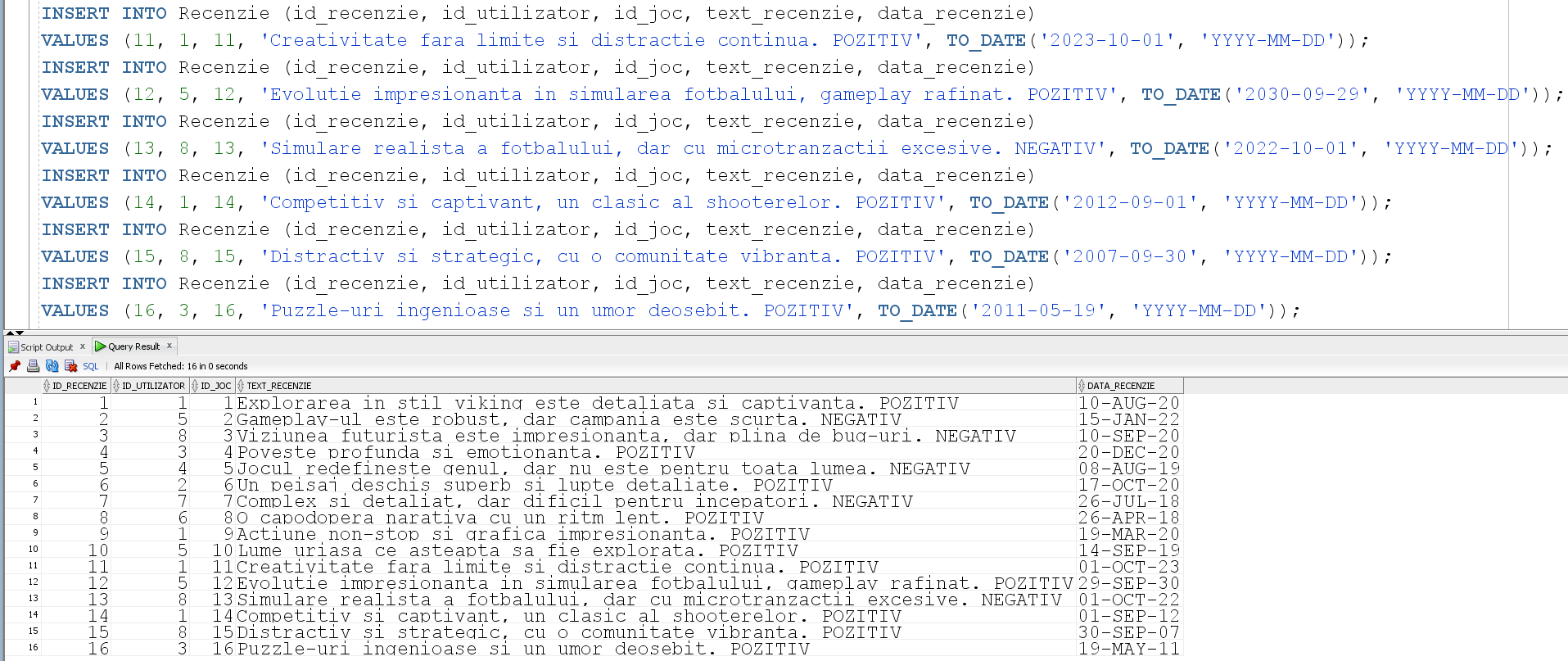
VALUES (15, 8, 15, 'Distractiv si strategic, cu o comunitate vibranta. POZITIV', TO\_DATE('2007-09-30', 'YYYY-MM-DD'));

INSERT INTO Recenzie (id\_recenzie, id\_utilizator, id\_joc, text\_recenzie, data\_recenzie)

VALUES (16, 3, 16, 'Puzzle-uri ingenioase si un umor deosebit. POZITIV', TO\_DATE('2011-05-19', 'YYYY-MM-DD'));







## COLECTIE

* CREATE

CREATE TABLE Colectie (

id\_colectie NUMBER(5) PRIMARY KEY,

id\_utilizator NUMBER(5) NOT NULL,

nume\_colectie VARCHAR2(50) NOT NULL,

CONSTRAINT fk\_colectie\_utilizator FOREIGN KEY (id\_utilizator) REFERENCES Utilizator(id\_utilizator)

);

DESCRIBE Colectie;

* INSERT

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (1, 1, 'Calatorii in Timp');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (2, 1, 'Exploarea spatiului');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (3, 2, 'Cosmar');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (4, 2, 'Lumi Deschise');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (5, 3, 'Actiune Intensa');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (6, 3, 'Cupe Mondiale');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (7, 4, 'RPG-uri Epice');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (8, 4, 'Shootere Tactice');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (9, 5, 'Paranormal');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (10, 5, 'Adrenalina la Maxim');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (11, 7, 'Povesti Epice');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (12, 7, 'Strategie si Tactici');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

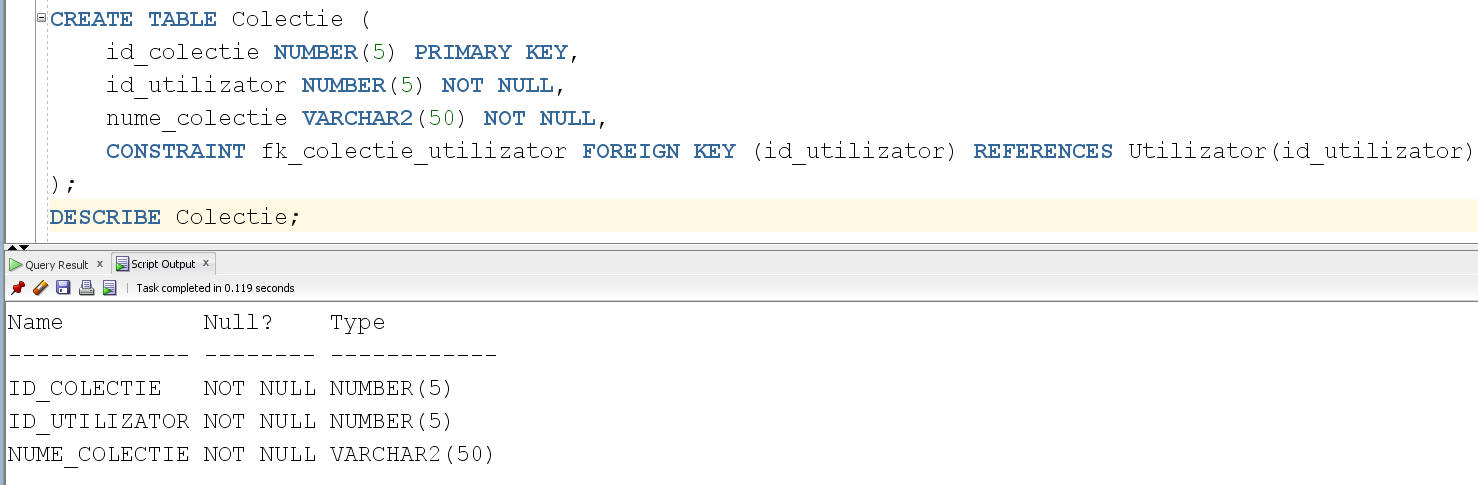
VALUES (13, 8, 'Sci-Fi si Fantezie');

INSERT INTO Colectie (id\_colectie, id\_utilizator, nume\_colectie)

VALUES (14, 8, 'Arena Sportiva');

SELECT \* FROM Colectie;





## COLECTIE\_JOC

* CREATE

CREATE TABLE Colectie\_Joc (

id\_colectie NUMBER(5) NOT NULL,

id\_joc NUMBER(5) NOT NULL,

PRIMARY KEY (id\_colectie, id\_joc),

CONSTRAINT fk\_colectie\_joc\_colectie FOREIGN KEY (id\_colectie) REFERENCES Colectie(id\_colectie),

CONSTRAINT fk\_colectie\_joc\_joc FOREIGN KEY (id\_joc) REFERENCES Joc(id\_joc)

);

DESCRIBE Colectie\_Joc;

* INSERT

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (1, 1);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (2, 2);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (3, 9);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (4, 8);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (5, 4);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (6, 13);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (7, 3);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (8, 14);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (9, 16);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (10, 5);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (11, 11);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (12, 11);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (13, 4);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (14, 2);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (1, 3);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (14, 13);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (4, 6);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (7, 7);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (10, 9);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (5, 10);

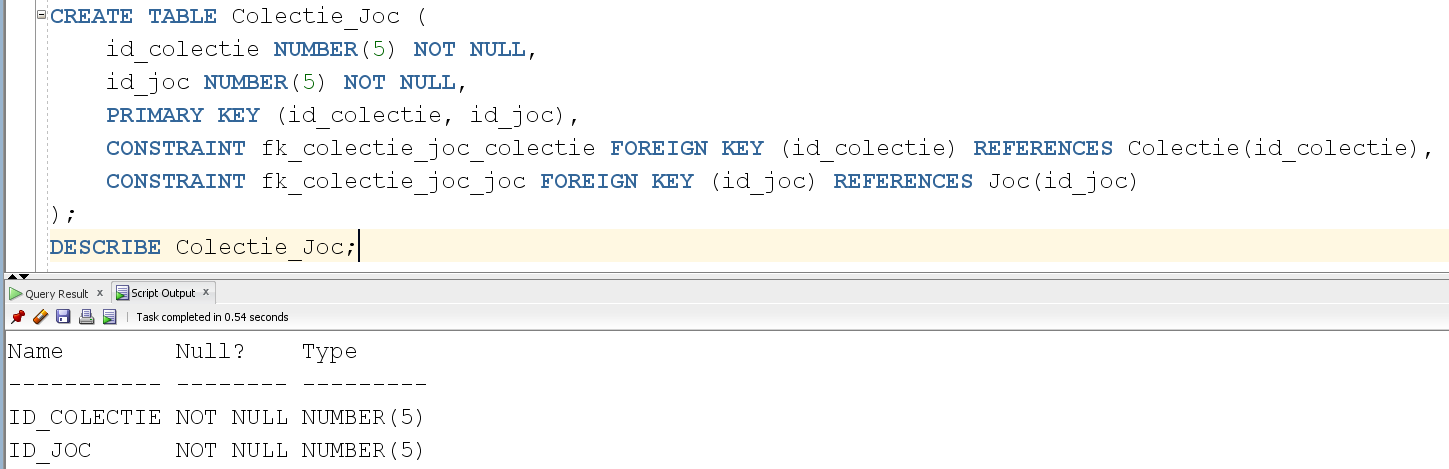
INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (8, 15);

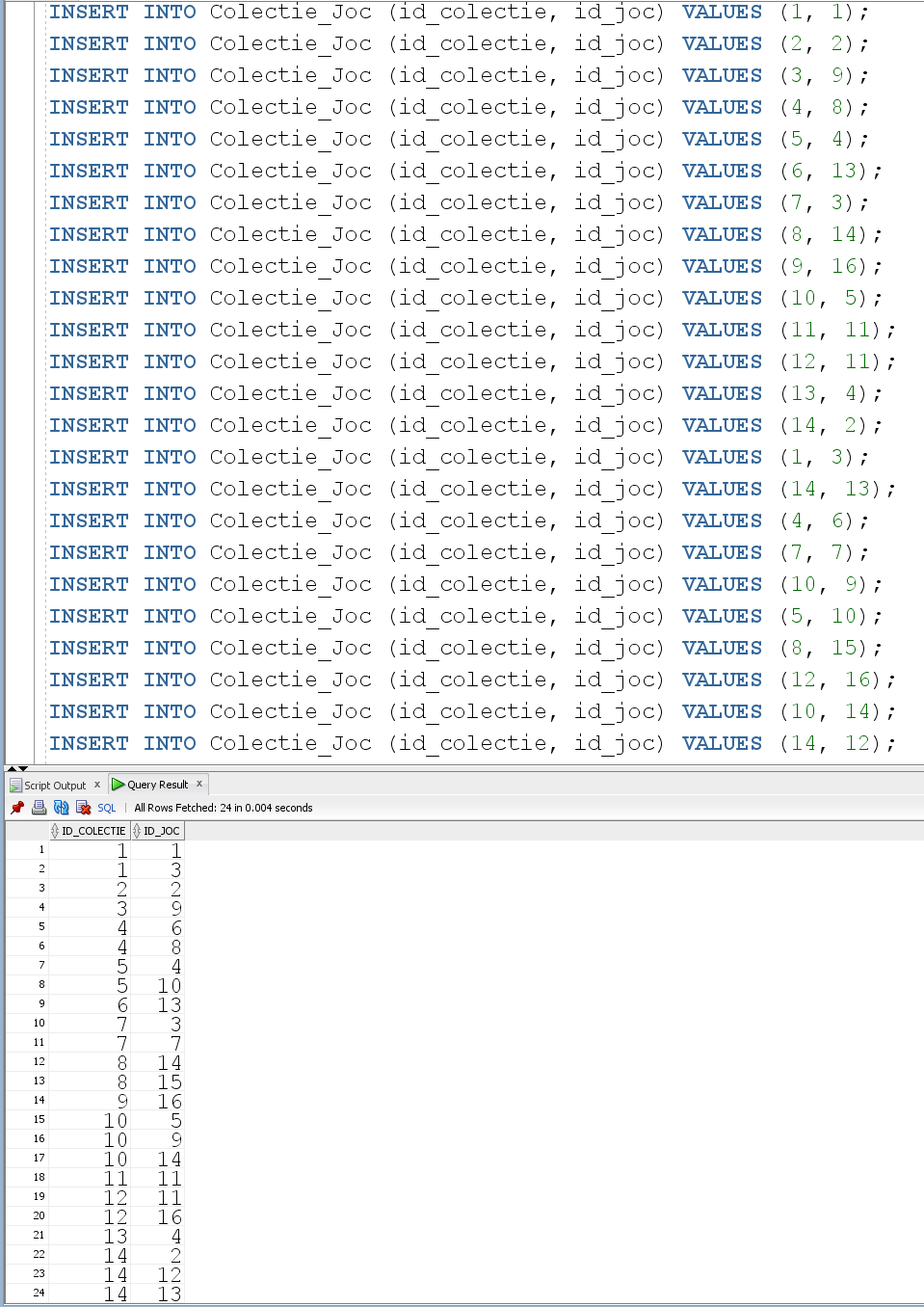
INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (12, 16);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (10, 14);

INSERT INTO Colectie\_Joc (id\_colectie, id\_joc) VALUES (14, 12);

SELECT \* FROM Colectie\_Joc;





## INVENTAR

* CREATE

CREATE TABLE Inventar (

id\_inventar NUMBER(5) PRIMARY KEY,

id\_utilizator NUMBER(5) NOT NULL,

capacitate NUMBER(2) NOT NULL CHECK (capacitate > 0),

CONSTRAINT fk\_inventar\_utilizator FOREIGN KEY (id\_utilizator) REFERENCES Utilizator(id\_utilizator)

);

DESCRIBE Inventar;

* INSERT

INSERT INTO Inventar (id\_inventar, id\_utilizator, capacitate) VALUES (1, 1, 10);

INSERT INTO Inventar (id\_inventar, id\_utilizator, capacitate) VALUES (2, 2, 3);

INSERT INTO Inventar (id\_inventar, id\_utilizator, capacitate) VALUES (3, 3, 5);

INSERT INTO Inventar (id\_inventar, id\_utilizator, capacitate) VALUES (4, 4, 6);

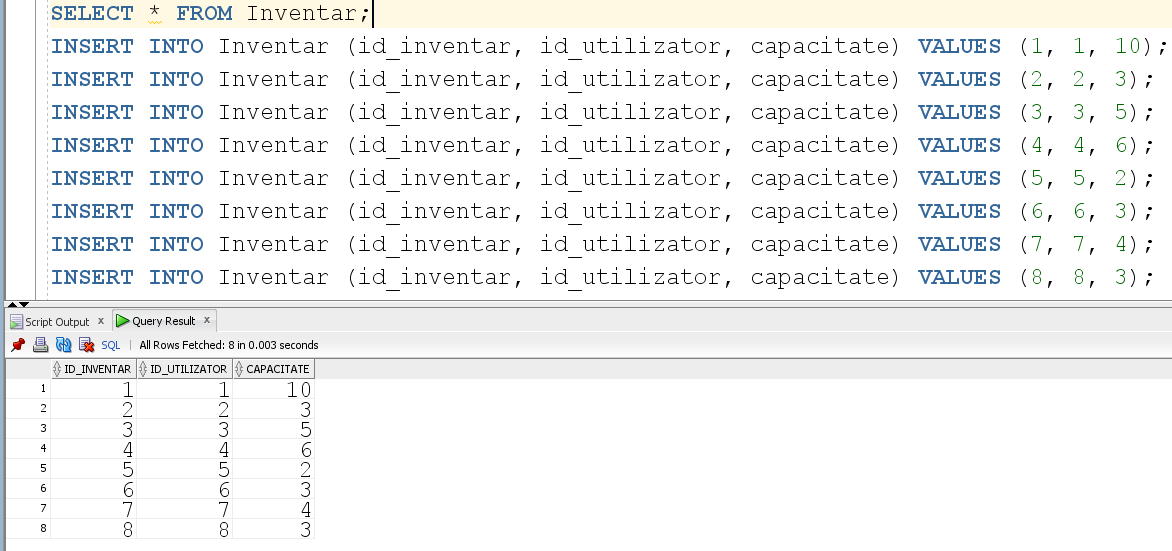
INSERT INTO Inventar (id\_inventar, id\_utilizator, capacitate) VALUES (5, 5, 2);

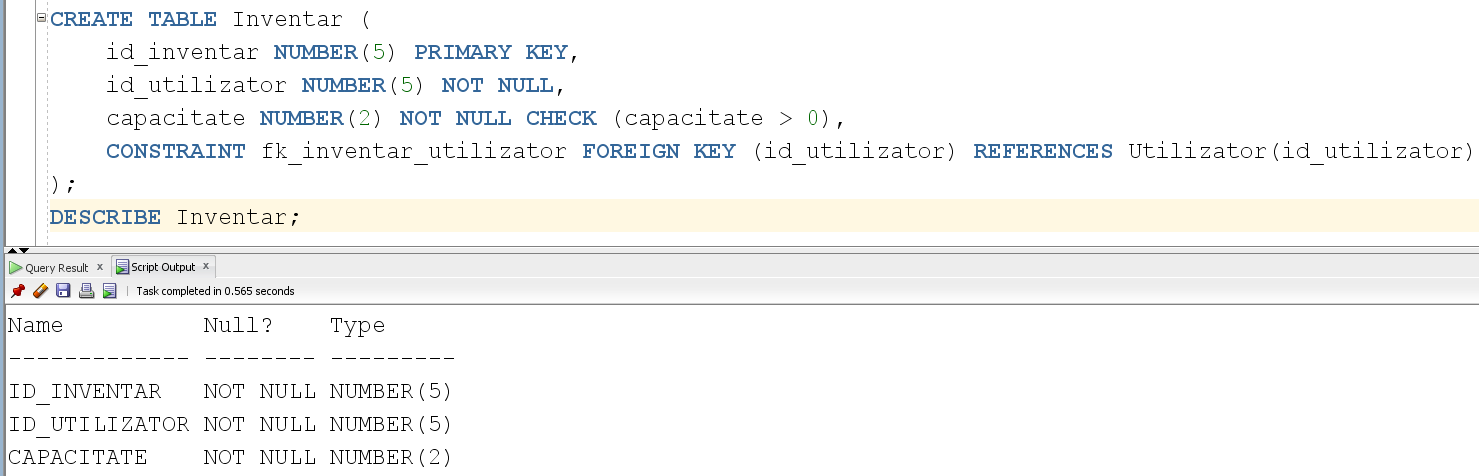
INSERT INTO Inventar (id\_inventar, id\_utilizator, capacitate) VALUES (6, 6, 3);

INSERT INTO Inventar (id\_inventar, id\_utilizator, capacitate) VALUES (7, 7, 4);

INSERT INTO Inventar (id\_inventar, id\_utilizator, capacitate) VALUES (8, 8, 3);

SELECT \* FROM Inventar;





## ITEM

* CREATE

CREATE TABLE Item (

id\_item NUMBER(5) PRIMARY KEY,

id\_joc NUMBER(5) NOT NULL,

nume\_item VARCHAR2(100) NOT NULL,

pret\_item NUMBER(10, 2),

raritate\_item VARCHAR2(50) NOT NULL,

CONSTRAINT fk\_item\_joc FOREIGN KEY (id\_joc) REFERENCES Joc(id\_joc)

);

DESCRIBE Item;

* INSERT

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (1, 'AK-47 | Redline', 25.00, 'CLASSIFIED', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (2, 'AWP | Dragon Lore', 1500.00, 'COVERT', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (3, 'Mann Co. Supply Crate Key', 2.50, 'COMMON', 15);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (4, 'Unusual Hat', 200.00, 'UNUSUAL', 15);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (5, 'Glock-18 | Fade', 300.00, 'COVERT', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (6, 'Strange Rocket Launcher', 50.00, 'STRANGE', 15);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (7, 'M4A4 | Howl', 1800.00, 'CONTRABAND', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (8, 'Strange Professional Killstreak Scattergun', 100.00, 'STRANGE', 15);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (9, 'Desert Eagle | Blaze', 600.00, 'COVERT', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (10, 'Unusual Taunt', 150.00, 'UNUSUAL', 15);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (11, 'Karambit | Doppler', 1200.00, 'COVERT', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (12, 'Australium Wrench', 100.00, 'AUSTRALIUM', 15);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (13, 'P90 | Asiimov', 50.00, 'CLASSIFIED', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (14, 'Strange Professional Killstreak Minigun', 200.00, 'STRANGE', 15);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (15, 'USP-S | Kill Confirmed', 100.00, 'COVERT', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (16, 'Unusual Rocket Launcher', 300.00, 'UNUSUAL', 15);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (17, 'AK-47 | Elite Build', 1.50, 'MIL-SPEC', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (18, 'AWP | Worm God', 2.00, 'MIL-SPEC', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (19, 'FAMAS | Pulse', 0.75, 'RESTRICTED', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (20, 'Galil AR | Chatterbox', 3.00, 'RESTRICTED', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (21, 'UMP-45 | Primal Saber', 1.20, 'RESTRICTED', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (22, 'CZ75-Auto | The Fuschia Is Now', 0.90, 'RESTRICTED', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (23, 'MP7 | Bloodsport', 1.50, 'RESTRICTED', 14);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (24, 'SG 553 | Pulse', 0.70, 'MIL-SPEC', 14);

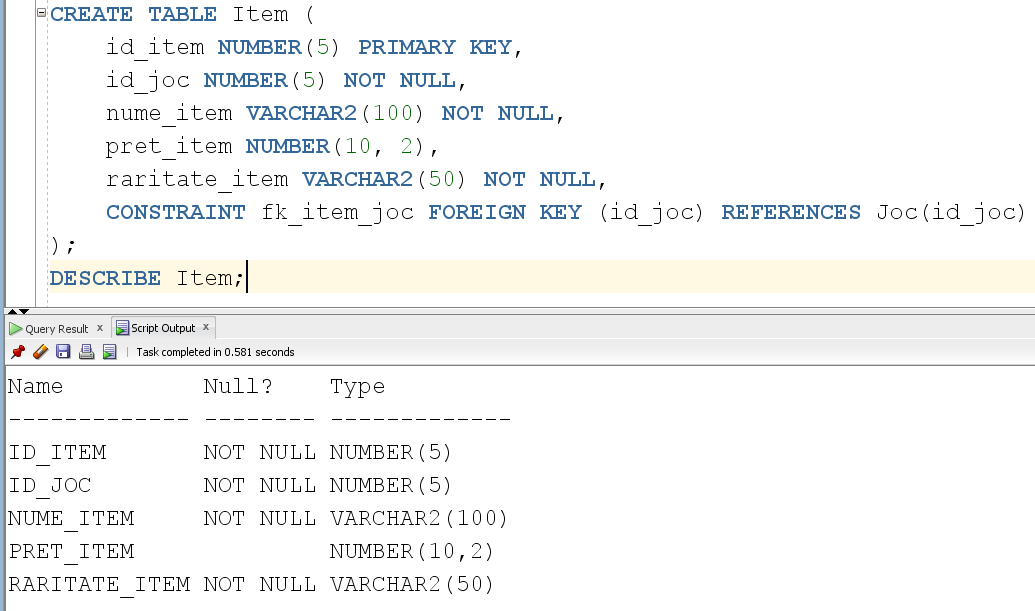
INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (25, 'P250 | Supernova', 0.65, 'MIL-SPEC', 14);

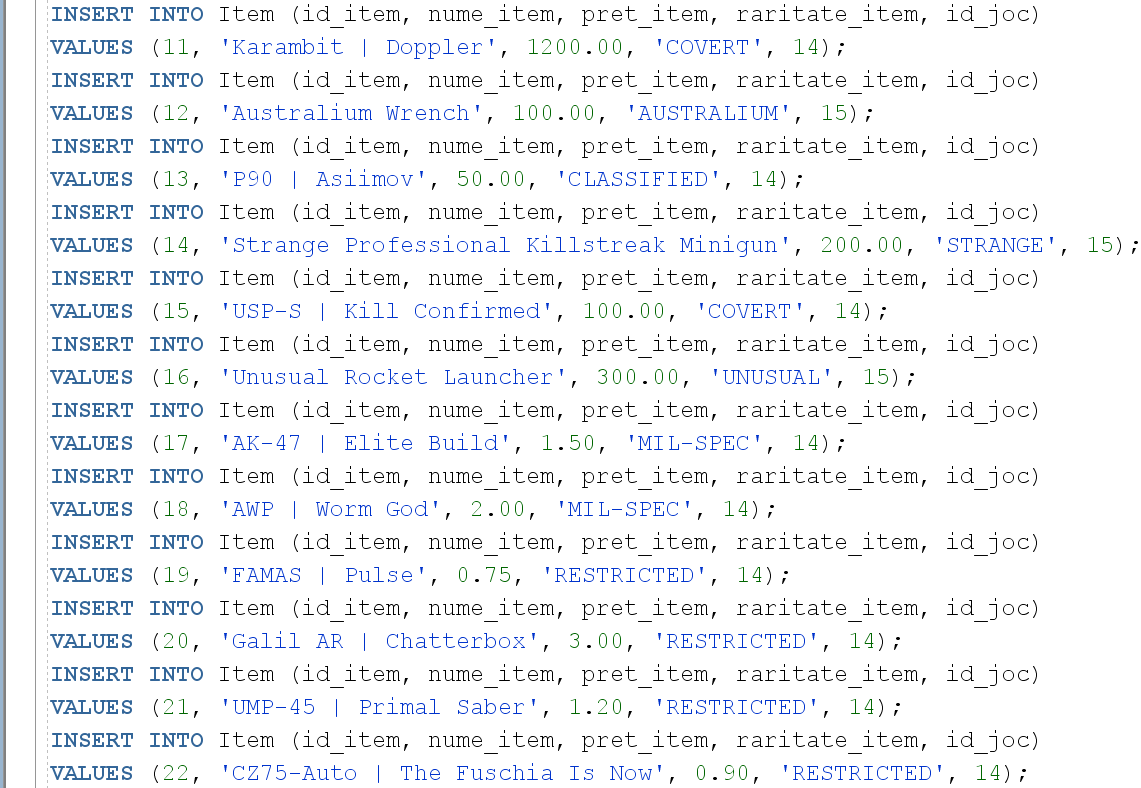
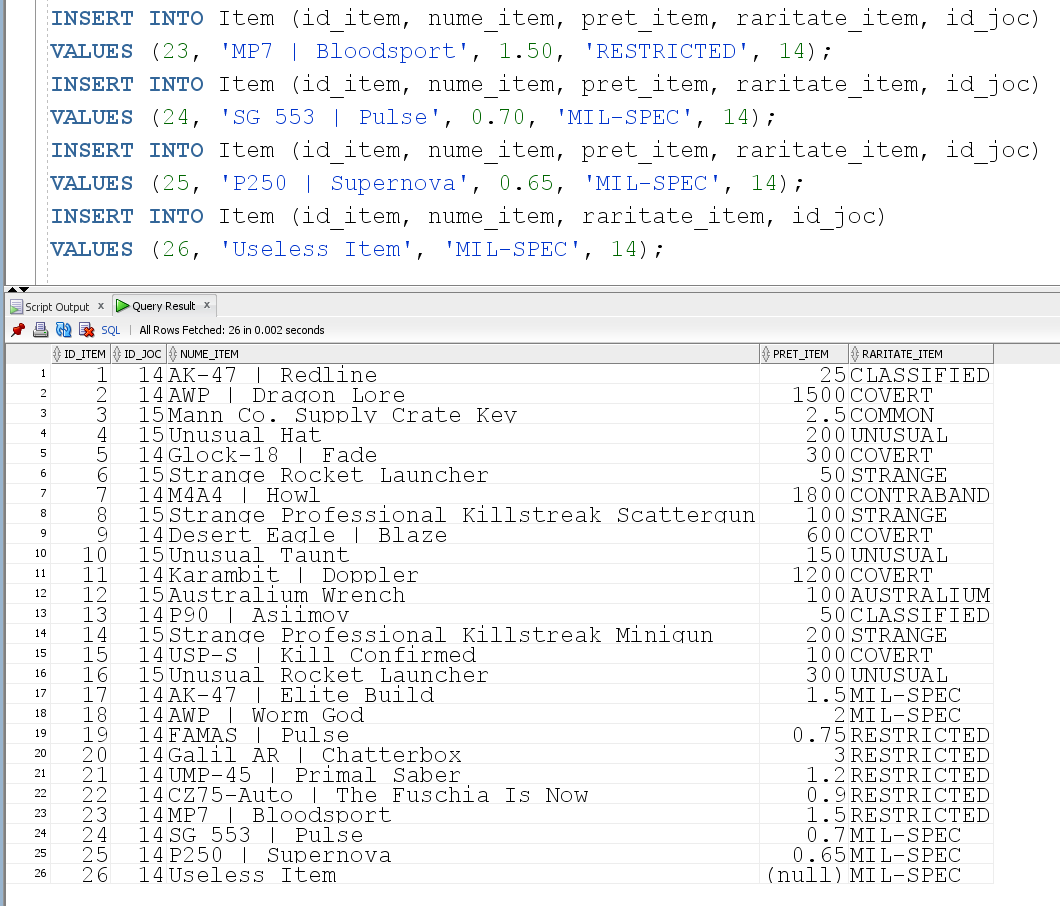
INSERT INTO Item (id\_item, nume\_item, raritate\_item, id\_joc)

VALUES (26, 'Useless Item', 'MIL-SPEC', 14);

SELECT \* FROM Item;





## INVENTAR\_ITEM

* CREATE

CREATE TABLE Inventar\_Item (

id\_item NUMBER(5) NOT NULL,

id\_inventar NUMBER(5) NOT NULL,

PRIMARY KEY (id\_item, id\_inventar),

CONSTRAINT fk\_inventar\_item\_item FOREIGN KEY (id\_item) REFERENCES Item(id\_item),

CONSTRAINT fk\_inventar\_item\_inventar FOREIGN KEY (id\_inventar) REFERENCES Inventar(id\_inventar)

);

DESCRIBE Inventar\_Item;

* INSERT

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (5, 1);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (12, 1);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (7, 1);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (19, 1);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (3, 1);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (26, 1);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (9, 2);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (15, 2);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (22, 2);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (1, 2);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (14, 3);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (6, 3);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (17, 3);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (20, 4);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (21, 4);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (24, 4);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (2, 5);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (11, 5);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (18, 5);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (25, 5);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (4, 6);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (8, 6);

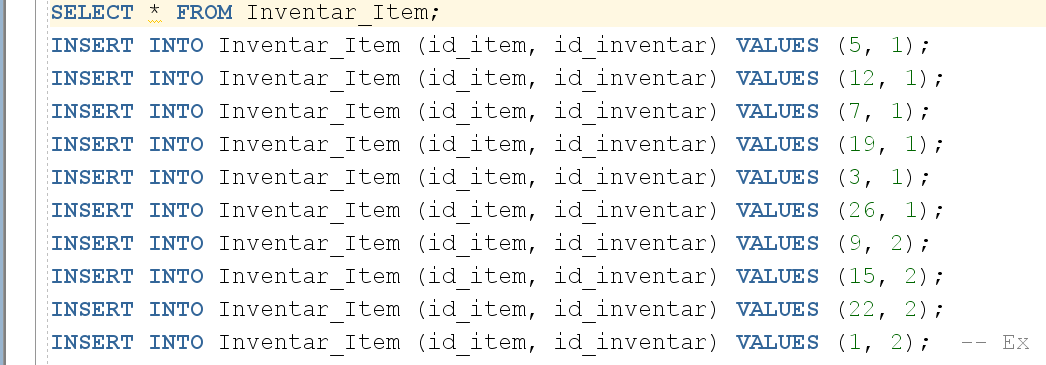
INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (10, 7);

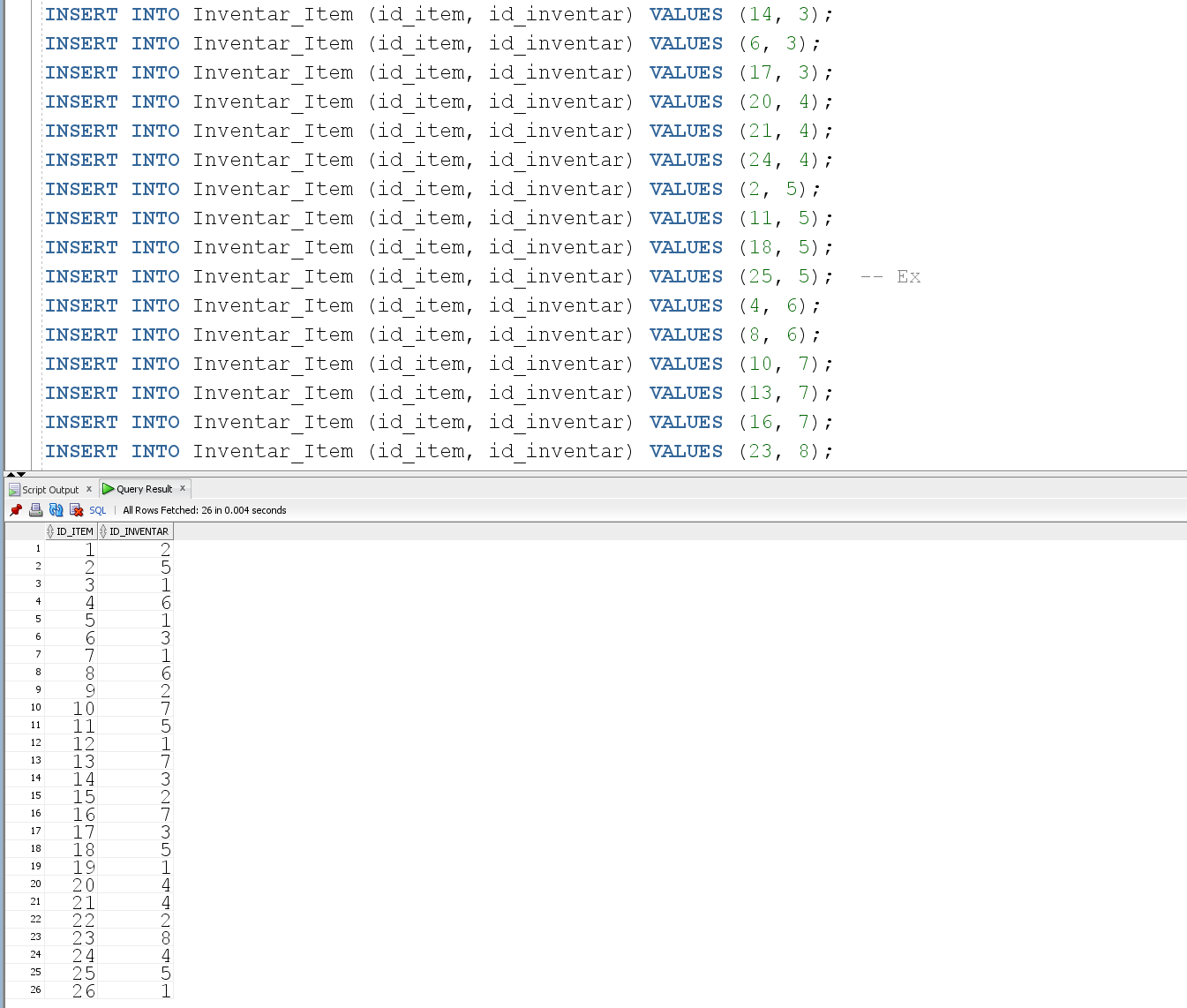
INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (13, 7);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (16, 7);

INSERT INTO Inventar\_Item (id\_item, id\_inventar) VALUES (23, 8);

SELECT \* FROM Inventar\_Item;





## TRANZACTIE

* CREATE

CREATE TABLE Tranzactie (

id\_tranzactie NUMBER(5) PRIMARY KEY,

id\_utilizator NUMBER(5) NOT NULL,

metoda\_tranzactie VARCHAR2(25) NOT NULL,

suma\_tranzactie NUMBER(5) NOT NULL CHECK (suma\_tranzactie >= 0),

data\_tranzactie DATE NOT NULL,

CONSTRAINT fk\_depozit\_utilizator FOREIGN KEY (id\_utilizator) REFERENCES Utilizator(id\_utilizator)

);

DESCRIBE Tranzactie;

* INSERT

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (1, 1, 'Card de credit', 100, TO\_DATE('2023-09-15', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (2, 1, 'PayPal', 50, TO\_DATE('2023-08-21', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (3, 2, 'Card de debit', 25, TO\_DATE('2023-07-30', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (4, 2, 'Transfer bancar', 200, TO\_DATE('2023-09-05', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (5, 3, 'Card de credit', 45, TO\_DATE('2023-10-11', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (6, 4, 'Card de debit', 150, TO\_DATE('2023-08-25', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (7, 5, 'PayPal', 30, TO\_DATE('2023-09-20', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (8, 6, 'Transfer bancar', 75, TO\_DATE('2023-10-02', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (9, 1, 'Card de credit', 120, TO\_DATE('2023-11-01', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (10, 1, 'PayPal', 60, TO\_DATE('2023-11-15', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (11, 2, 'Card de debit', 35, TO\_DATE('2023-12-05', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (12, 2, 'Transfer bancar', 210, TO\_DATE('2023-12-20', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (13, 3, 'Card de credit', 50, TO\_DATE('2023-11-10', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (14, 4, 'Card de debit', 160, TO\_DATE('2023-11-25', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

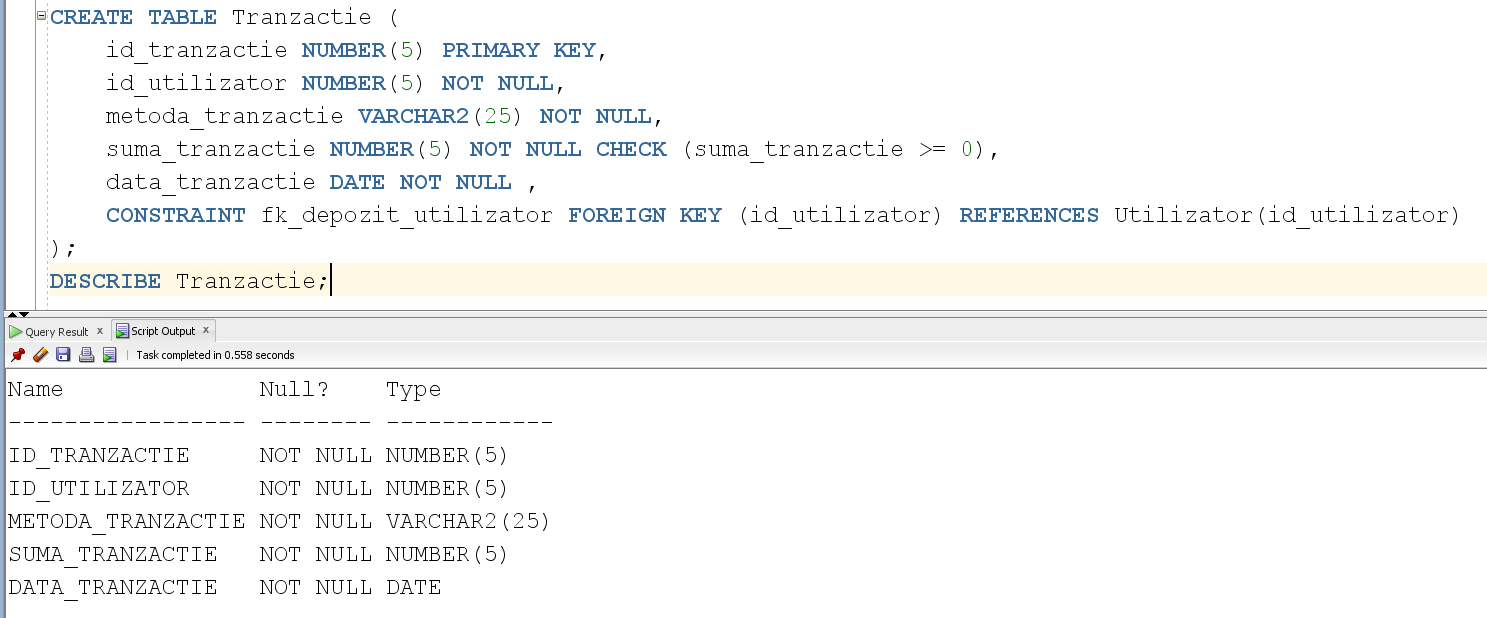
VALUES (15, 5, 'PayPal', 40, TO\_DATE('2023-12-01', 'YYYY-MM-DD'));

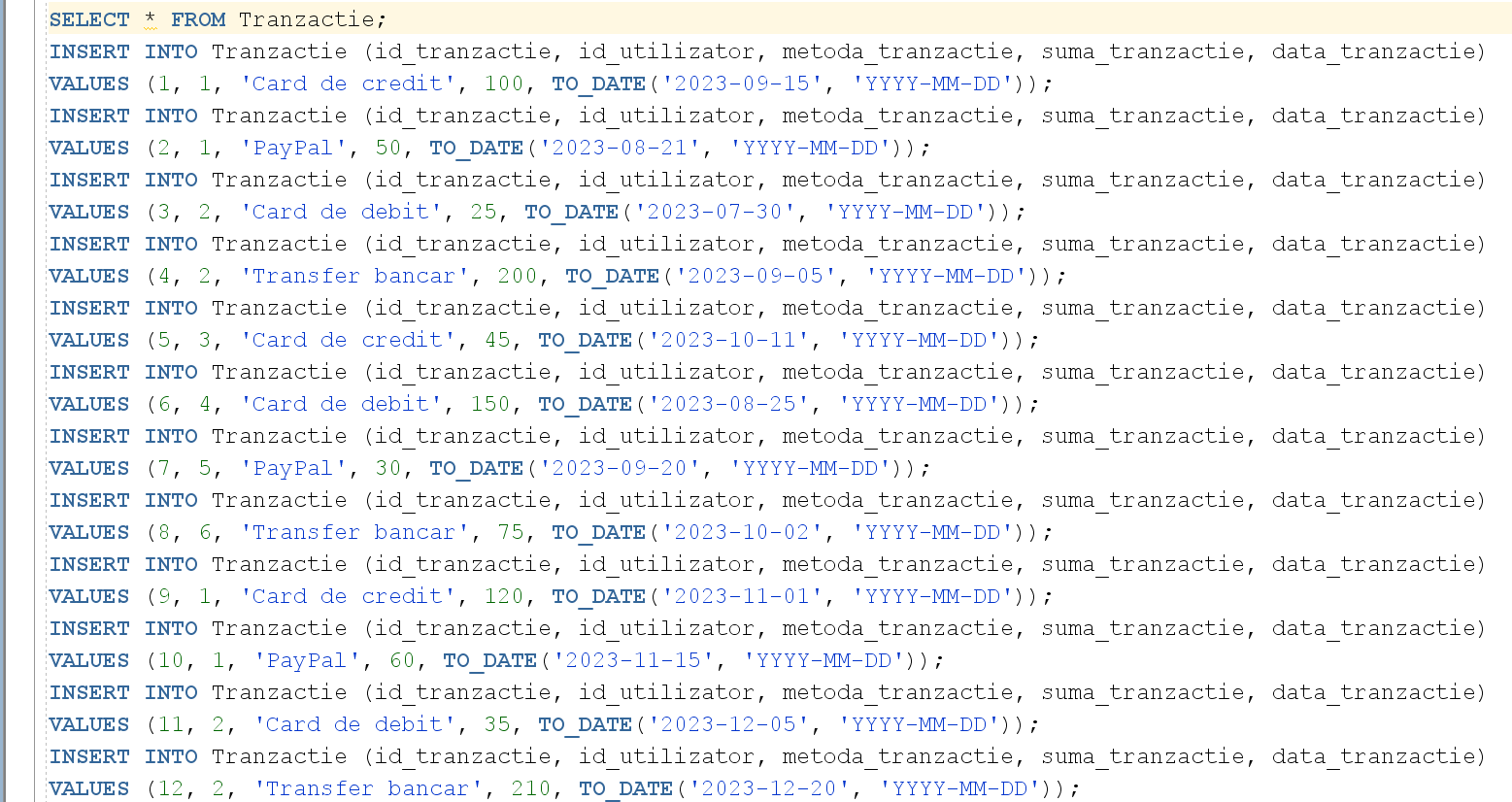
INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

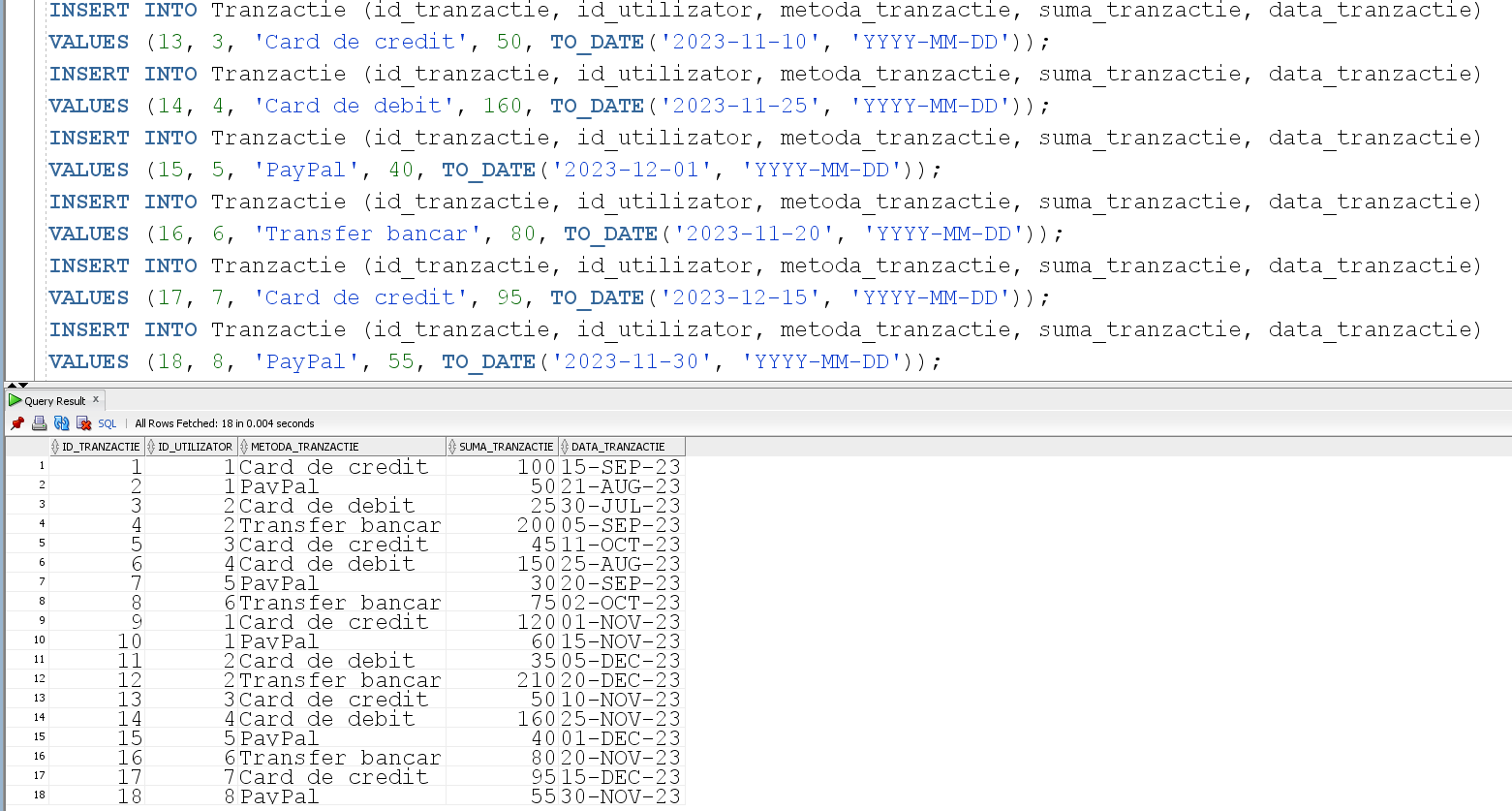
VALUES (16, 6, 'Transfer bancar', 80, TO\_DATE('2023-11-20', 'YYYY-MM-DD'));

INSERT INTO Tranzactie (id\_tranzactie, id\_utilizator, metoda\_tranzactie, suma\_tranzactie, data\_tranzactie)

VALUES (17, 7, 'Card de credit', 95, TO\_DATE('2023-12-15', 'YYYY-MM-DD'));







# Cereri SQL complexe (12)

## CEREREA 1

-- a) subcereri sincronizate în care intervin cel putin 3 tabele

-- b) subcereri nesincronizate în clauza FROM

-- Sa se afiseze numele si tara utilizatorilor cu varsta

-- pana in 20 de ani care locuiesc in tari cu studio-uri

-- de development.

SELECT u\_u20.nume, u\_u20.tara tara\_utilizator,

DECODE((

SELECT t.nume\_tara

FROM Developer d, Locatie l, Tara t

WHERE d.id\_locatie = l.id\_locatie

AND l.id\_tara = t.id\_tara

AND t.nume\_tara = u\_u20.tara

GROUP BY t.nume\_tara

), NULL, 'NU EXISTA', 'EXISTA') tara\_studio

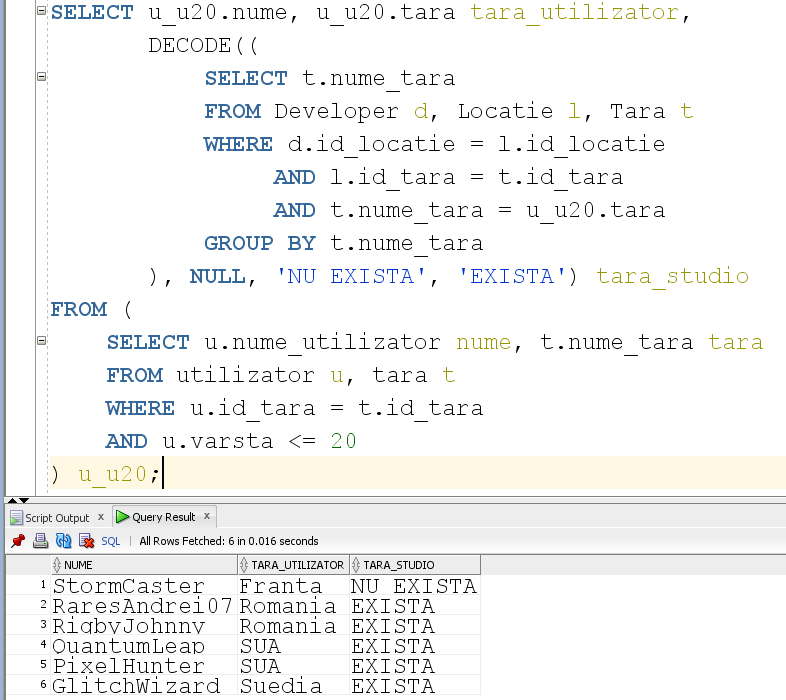
FROM (

SELECT u.nume\_utilizator nume, t.nume\_tara tara

FROM utilizator u, tara t

WHERE u.id\_tara = t.id\_tara

AND u.varsta <= 20

) u\_u20;

## CEREREA 2

-- c) grupari de date, functii grup, filtrare la nivel de grupuri

-- cu subcereri nesincronizate (în clauza de HAVING) în care intervin

-- cel putin 3 tabele (in cadrul aceleiasi cereri)

-- Sa se afiseze numele si valoarea totala a jocurilor utilizatorilor care

-- au valoarea medie a jocurilor detinute in librarie (reuniunea colectiilor)

-- peste valoarea medie a jocurilor dezvoltate de studio-uri din SUA.

SELECT u.nume\_utilizator, SUM(j.pret\_joc) AS valoare\_jocuri

FROM Utilizator u, Colectie c, Colectie\_Joc cj, Joc j

WHERE u.id\_utilizator = c.id\_utilizator

AND c.id\_colectie = cj.id\_colectie

AND cj.id\_joc = j.id\_joc

GROUP BY u.nume\_utilizator

HAVING AVG(j.pret\_joc) > (

SELECT AVG(j1.pret\_joc)

FROM Joc j1, Developer d, Locatie l, Tara t

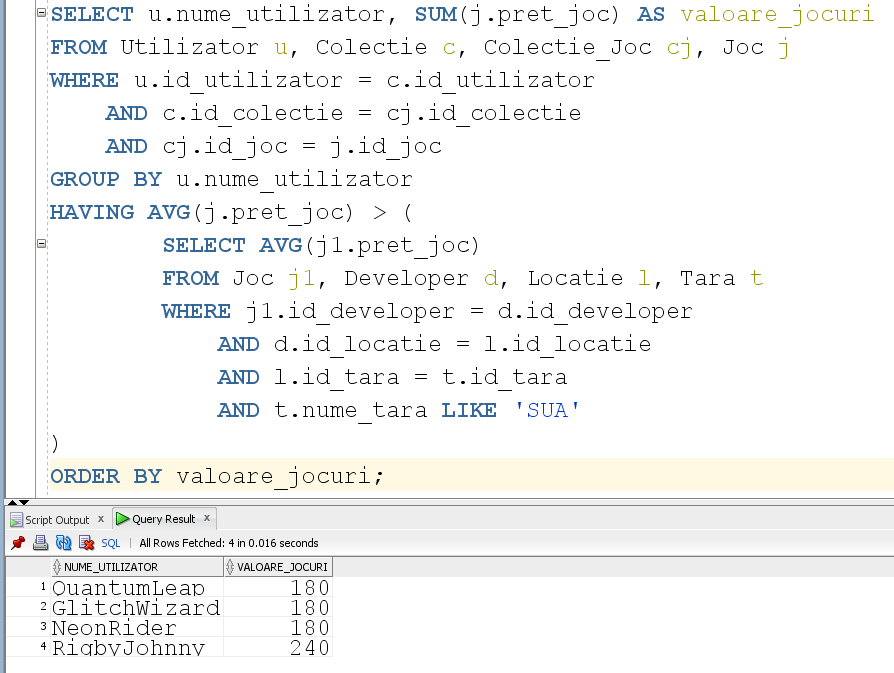
WHERE j1.id\_developer = d.id\_developer

AND d.id\_locatie = l.id\_locatie

AND l.id\_tara = t.id\_tara

AND t.nume\_tara LIKE 'SUA'

)

ORDER BY valoare\_jocuri;

## CEREREA 3

-- d) ordonari si utilizarea functiilor NVL si DECODE

-- (în cadrul aceleiasi cereri)

-- Afisati numele utilizatorilor impreuna cu valoarea inventarului lor.

-- Daca valoarea este sub 500, afisati "INCEPATOR", altfel "BOGAT".

-- Sortati descrescator dupa valoarea inventarului.

SELECT NumeUtiliz,

ValoareInv, DECODE(SIGN(500 - ValoareInv),

1, 'INCEPATOR',

'BOGAT') conditie

FROM (

SELECT u.nume\_utilizator AS NumeUtiliz,

SUM(NVL(i.pret\_item, 0)) AS ValoareInv

FROM Utilizator u, Inventar inv, Inventar\_Item ii, Item i

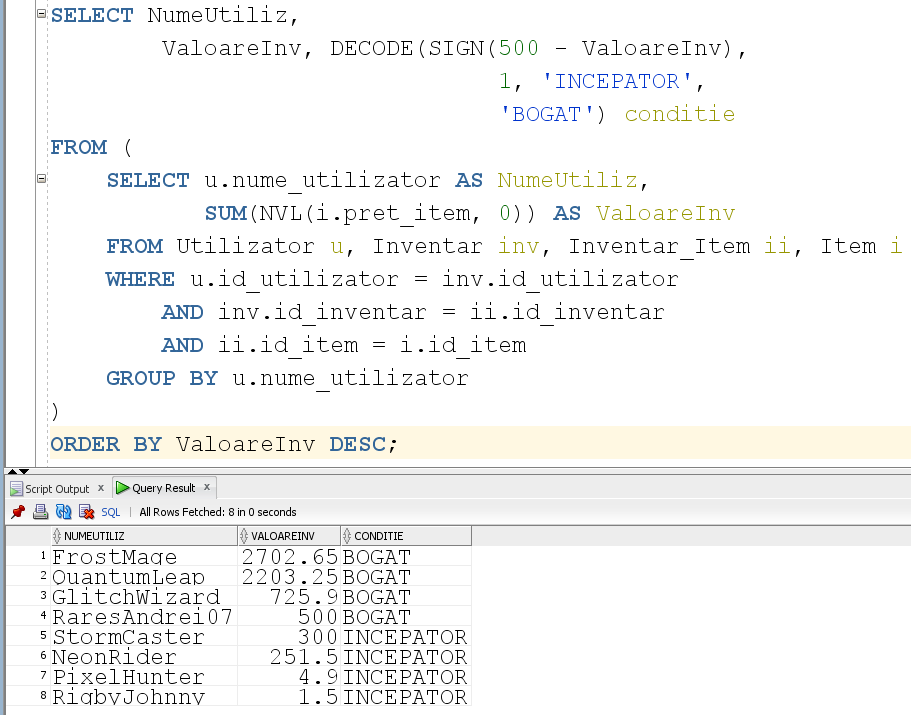
WHERE u.id\_utilizator = inv.id\_utilizator

AND inv.id\_inventar = ii.id\_inventar

AND ii.id\_item = i.id\_item

GROUP BY u.nume\_utilizator

)

ORDER BY ValoareInv DESC;

## CEREREA 4

-- e) utilizarea a cel putin 2 functii pe siruri de caractere,

-- 2 functii pe date calendaristice, a cel putin unei expresii CASE

-- Sa se afiseze numele utilizatorului, domeniul email-ului (doar pentru

-- utilizatorii care nu folosesc yandex.com) si numele jocurilor din colectii

-- (cu majuscule) pentru utilizatorii care detin jocuri lansate pana acum 3 ani,

-- cu recenzii date de acestia in ultimii 5 ani. Daca utilizatorul a lasat o

-- recenzie inainte de data lansarii jocului, este "EARLY TESTER", altfel

-- este un "UTILIZATOR OBISNUIT". Sortati dupa numele utilizatorului.

SELECT u.nume\_utilizator,

SUBSTR(u.email, INSTR(u.email, '@') + 1) AS domeniu\_mail,

UPPER(j.nume\_joc) AS joc\_upper,

CASE

WHEN MONTHS\_BETWEEN(r.data\_recenzie, j.data\_lansare\_joc) < 0

THEN 'EARLY TESTER'

ELSE

'UTILIZATOR OBISNUIT'

END AS tip\_utilizator

FROM Joc j, Colectie c, Colectie\_Joc cj, Utilizator u, Recenzie r

WHERE j.id\_joc = cj.id\_joc

AND c.id\_colectie = cj.id\_colectie

AND c.id\_utilizator = u.id\_utilizator

AND r.id\_utilizator = u.id\_utilizator

AND r.id\_joc = j.id\_joc

AND u.email NOT LIKE '%yandex.com'

AND j.data\_lansare\_joc <= ADD\_MONTHS(SYSDATE, -36) -- jocuri lansate pana acum 3 ani

AND r.data\_recenzie >= ADD\_MONTHS(SYSDATE, -60) -- recenzie din ultimii 4 ani

ORDER BY u.nume\_utilizator;



## CEREREA 5

-- f) utilizarea a cel putin 1 bloc de cerere (clauza WITH)

-- Afisati numele, tara si suma totala adunata din tranzactii pentru

-- utilizatorii platformei. Daca suma este mai mica de 75, afisati

-- "Intelept", intre 75 si 200, "Entuziast", altfel "Cheltuitor".

-- Ordonati descrescator dupa suma ceruta.

WITH Suma\_Tranzactii AS (

SELECT id\_utilizator, SUM(suma\_tranzactie) AS total

FROM Tranzactie

GROUP BY id\_utilizator

)

SELECT u.nume\_utilizator, t.nume\_tara, st.total,

CASE

WHEN st.total > 200 THEN 'Cheltuitor'

WHEN st.total > 75 THEN 'Entuziast'

ELSE 'Intelept'

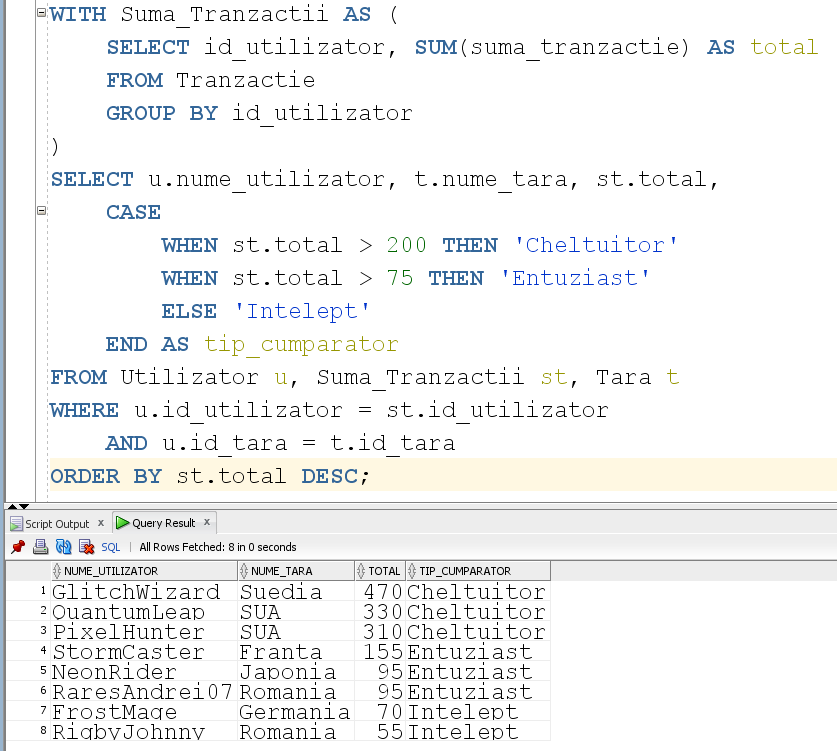
END AS tip\_cumparator

FROM Utilizator u, Suma\_Tranzactii st, Tara t

WHERE u.id\_utilizator = st.id\_utilizator

AND u.id\_tara = t.id\_tara

ORDER BY st.total DESC;



# Actualizare și suprimare a datelor

## C1

-- Stergeti din Inventar Si Item itemele care au raritatea "MEH"

-- si apartin jocului "Disliked Game".

DELETE FROM Item i

WHERE i.id\_item IN (

SELECT i1.id\_item

FROM Item i1, Joc j

WHERE i1.id\_joc = j.id\_joc

AND i1.raritate\_item LIKE 'MEH'

AND j.nume\_joc LIKE 'Disliked Game');

-- /// insert-urile necesare ///

INSERT INTO Joc (id\_joc, id\_developer, nume\_joc, pret\_joc, data\_lansare\_joc, varsta\_recomandata, rating)

VALUES (17, 11, 'Disliked Game', 5, TO\_DATE('2024-03-10', 'YYYY-MM-DD'), 10, 1);

SELECT \* FROM Joc;

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (27, 'Filler 1', 10, 'MEH', 17);

INSERT INTO Item (id\_item, nume\_item, pret\_item, raritate\_item, id\_joc)

VALUES (28, 'Filler 2', 1.2, 'MEH', 17);

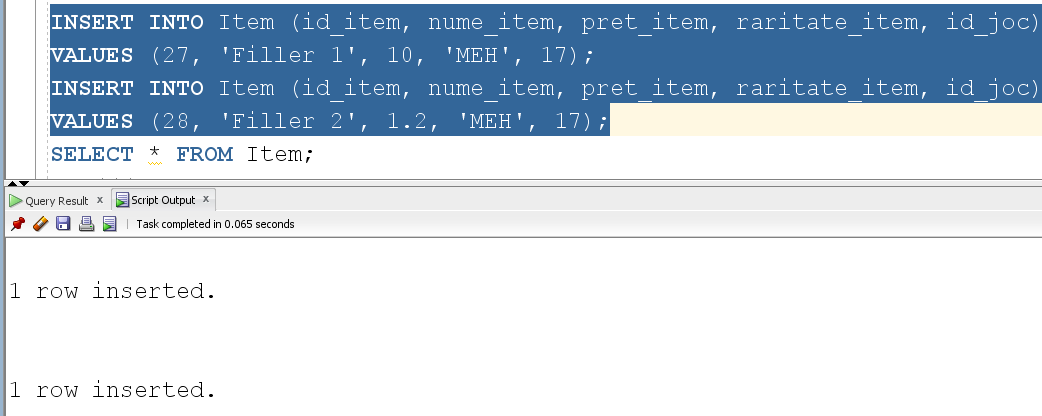
SELECT \* FROM Item;

-- /// delete jocul adaugat pentru c1

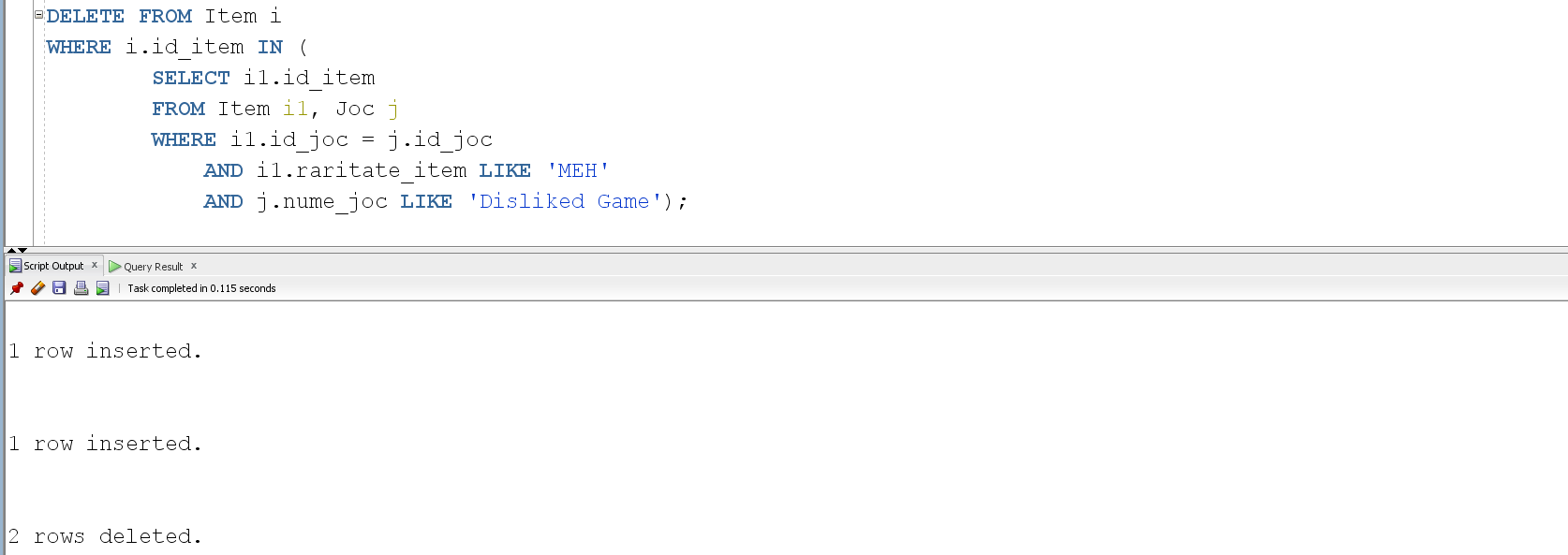
DELETE FROM Joc j

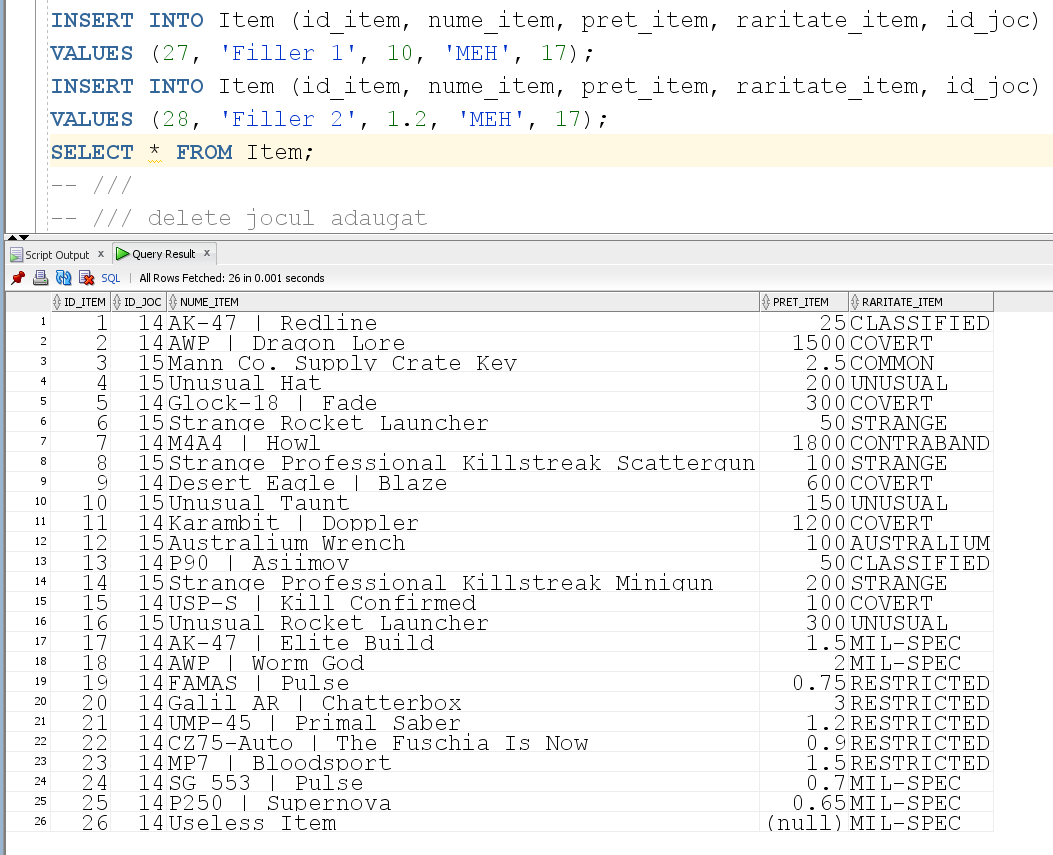
WHERE j.id\_joc = 17;

-- ///









## C2

-- Adaugati cate 10 u.m. la fiecare tranzactie a

-- utilizatorului cu suma tranzactiilor mai mare de 400 u.m.

UPDATE Tranzactie

SET suma\_tranzactie = suma\_tranzactie + 10

WHERE id\_utilizator IN (

SELECT id\_utilizator

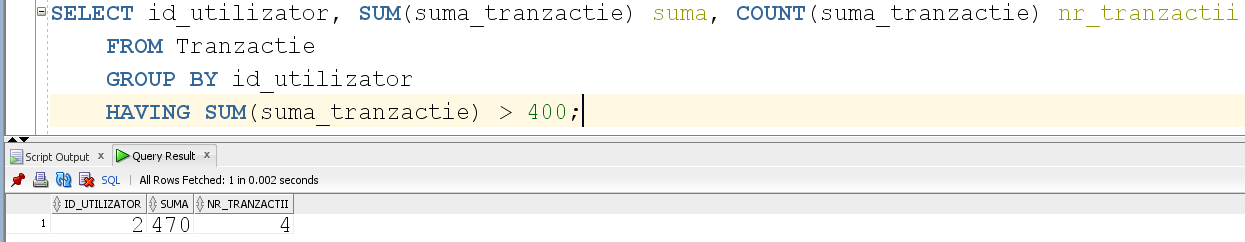
FROM Tranzactie

GROUP BY id\_utilizator

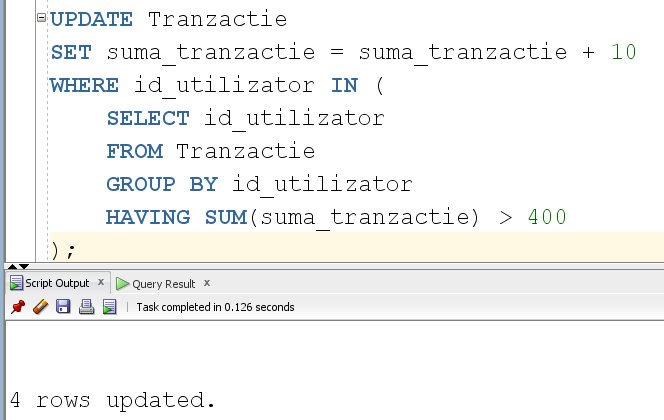
HAVING SUM(suma\_tranzactie) > 400

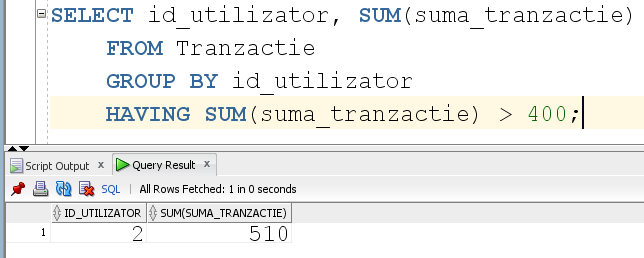
);

**INAINTE**



**DUPA**

****

****

## C3

-- Adaugati un nou atribut tabelei Developer numit "salariu\_bun",

-- unde acesta are valoarea “DA” daca angajatii au salariul mai mare decat

-- media salariilor tarii in care lucreaza, si "NU" in caz contrar.

ALTER TABLE Developer

ADD (salariu\_bun VARCHAR2(25));

UPDATE Developer d

SET d.salariu\_bun = (

CASE

WHEN d.salariu\_angajati > (

SELECT t.salariu\_mediu

FROM Locatie l, Tara t

WHERE l.id\_locatie = d.id\_locatie

AND l.id\_tara = t.id\_tara

) THEN 'DA'

ELSE 'NU'

END

);

-- /// verificari

describe developer;

select \* from developer;

-- ///

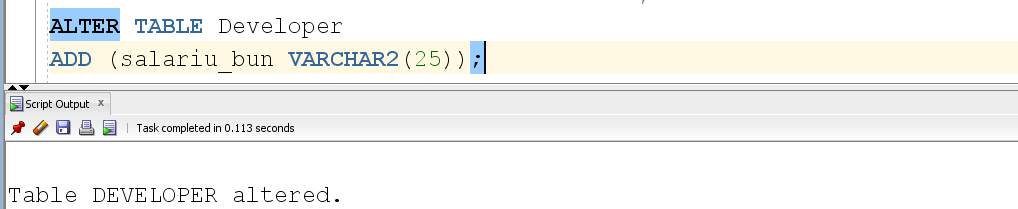
-- /// stergerea atributului

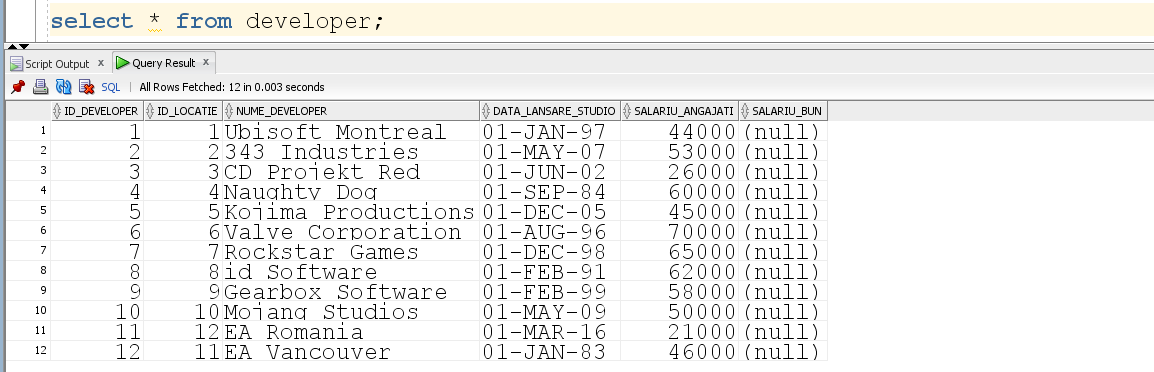
ALTER TABLE Developer

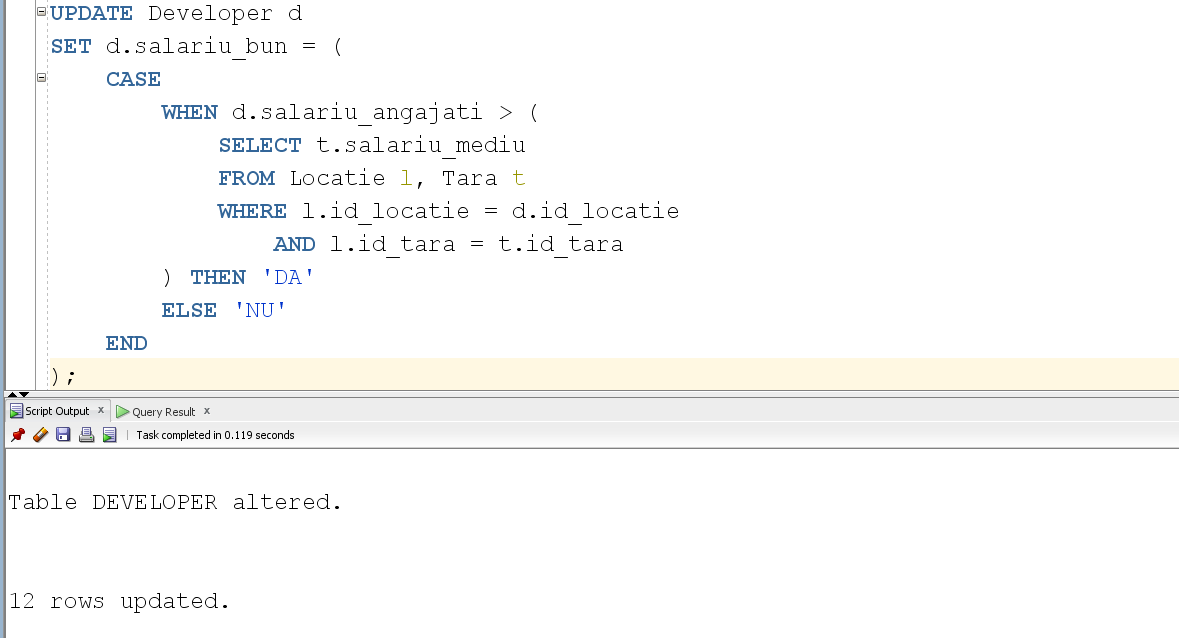
DROP COLUMN salariu\_bun;

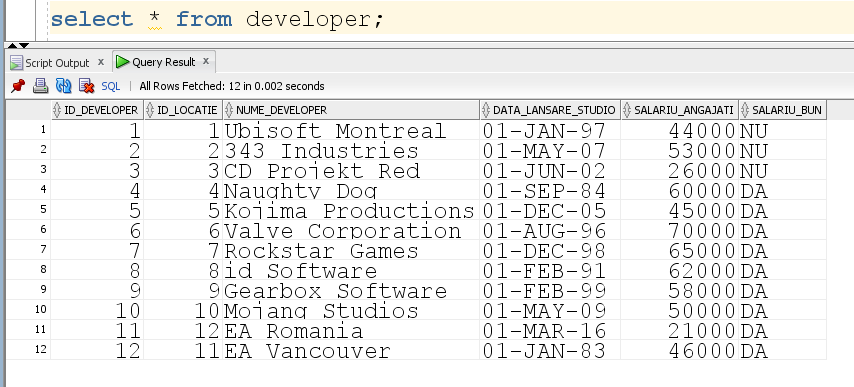
-- ///

**PAS CU PAS SCREENSHOT-URI**









# VIZUALIZARE COMPLEXĂ

## Crearea vizualizării

-- Vizualizare complexa care cuprinde id-ul si numele utilizatorilor,

-- notificarile si mesajele primite, precum si daca mesajul este citit.

CREATE VIEW Util\_Notif\_Mesaj AS

SELECT u.id\_utilizator, u.nume\_utilizator, n.text\_notificare,

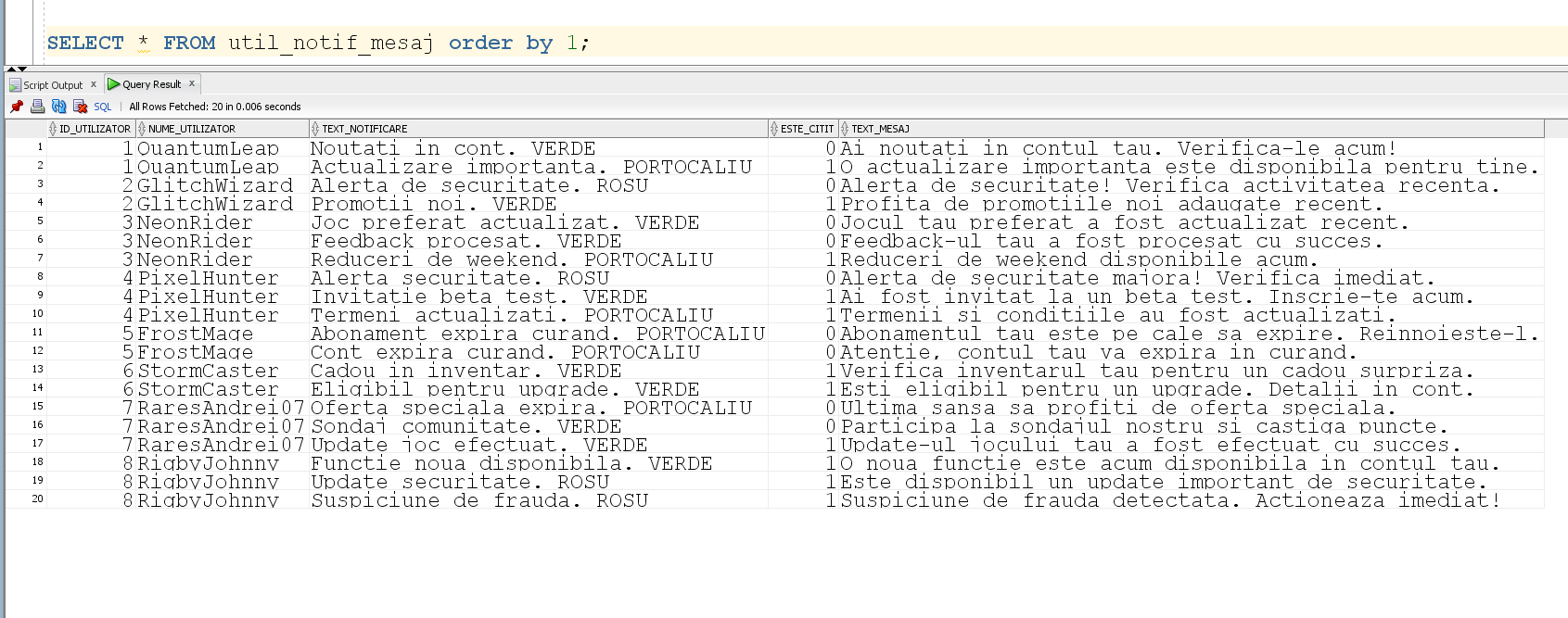
m.este\_citit, m.text\_mesaj

FROM Mesaj m, Notificare n, Utilizator u

WHERE m.id\_notificare = n.id\_notificare

AND u.id\_utilizator = n.id\_utilizator;

SELECT \* FROM util\_notif\_mesaj order by 1;



## Operatie permisa

-- Afisati utilizatorii care au toate mesajele citite. Adaugati si

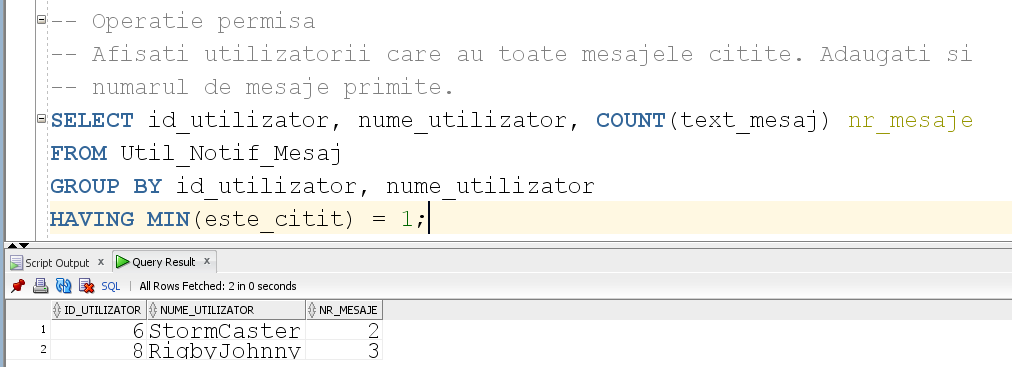
-- numarul de mesaje primite.

SELECT id\_utilizator, nume\_utilizator, COUNT(text\_mesaj) nr\_mesaje

FROM Util\_Notif\_Mesaj

GROUP BY id\_utilizator, nume\_utilizator

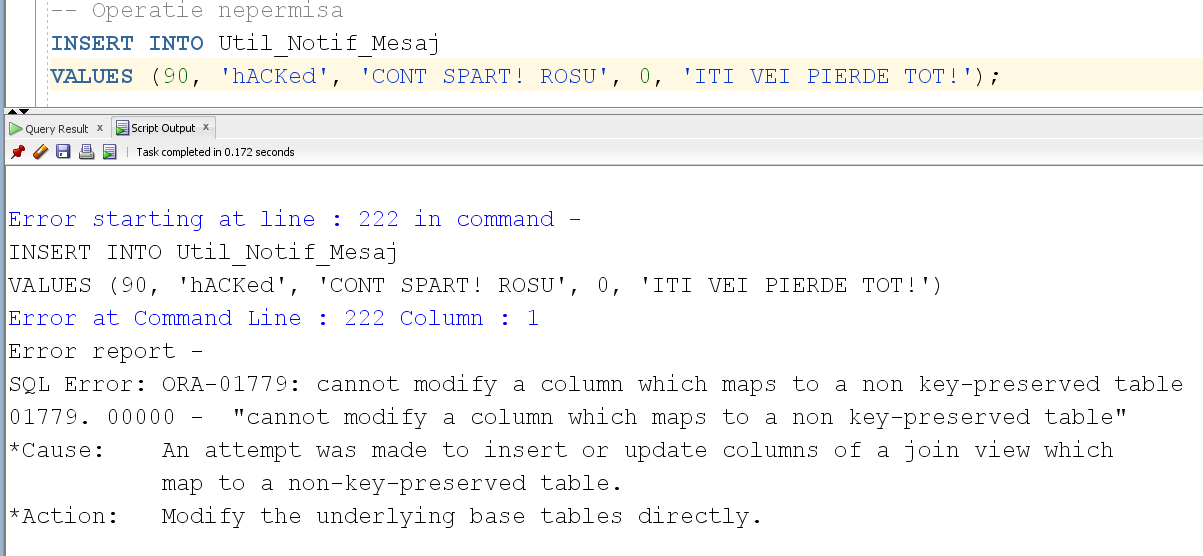
HAVING MIN(este\_citit) = 1;



## Operatie nepermisa

INSERT INTO Util\_Notif\_Mesaj

VALUES (90, 'hACKed', 'CONT SPART! ROSU', 0, 'ITI VEI PIERDE TOT!');



# Cereri SQL complexe (15)

## CEREREA 1

-- o cerere ce utilizeaza operatia **outer-join** pe minimum 4 tabele

-- (**LEFT OUTER JOIN**)

-- Afisati numele utilizatorilor, id-ul recenziilor date,

-- jocurile evaluate si colectiile utilizatorului, indiferent daca detine colectii.

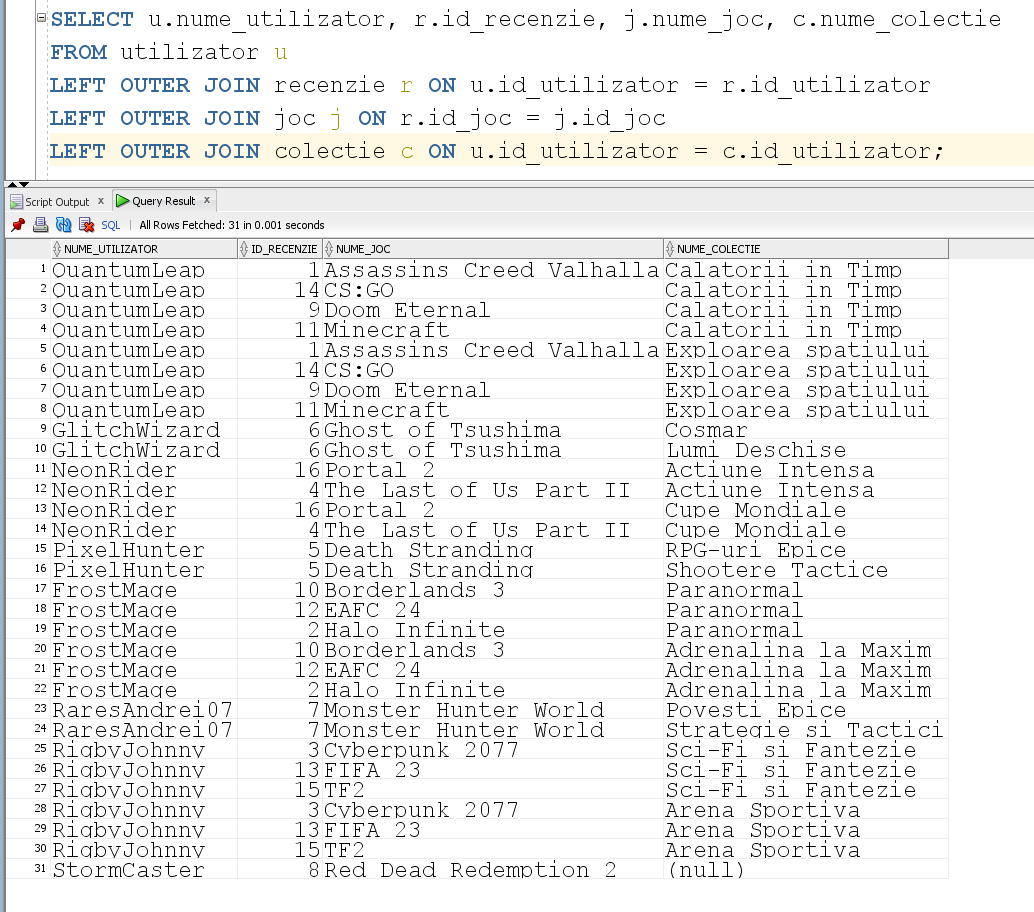
SELECT u.nume\_utilizator, r.id\_recenzie, j.nume\_joc, c.nume\_colectie

FROM utilizator u

LEFT OUTER JOIN recenzie r ON u.id\_utilizator = r.id\_utilizator

LEFT OUTER JOIN joc j ON r.id\_joc = j.id\_joc

LEFT OUTER JOIN colectie c ON u.id\_utilizator = c.id\_utilizator;



## CEREREA 2

-- o cerere ce utilizeaza operatia **division**

-- Afisati toate tarile cu numar par de litere

-- in care nu exista studio-uri de jocuri.

SELECT t.nume\_tara

FROM Tara t

WHERE MOD(LENGTH(t.nume\_tara), 2) = 0

AND NOT EXISTS (

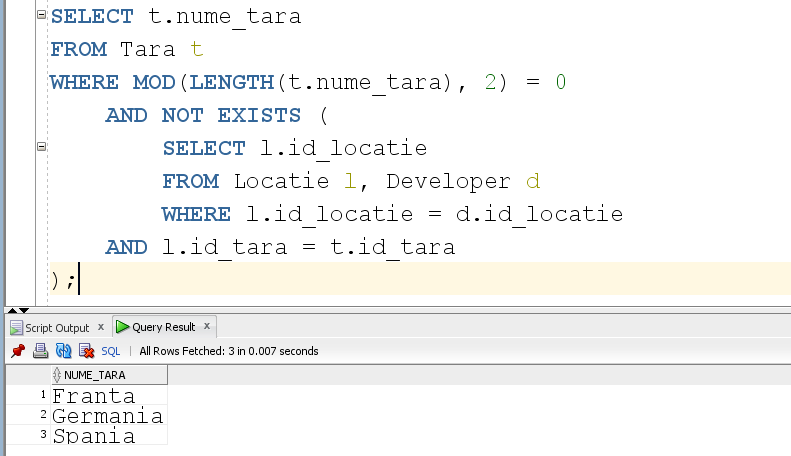
SELECT l.id\_locatie

FROM Locatie l, Developer d

WHERE l.id\_locatie = d.id\_locatie

AND l.id\_tara = t.id\_tara

);



## CEREREA 3

-- o cerere care implementeaza analiza **top-n**.

-- Afisati top 3 utilizatori cu cel mai mare numar de iteme.

-- Afisati si id-ul utilizatorului si numarul de iteme.

SELECT id\_utilizator, nume\_utilizator, numar\_iteme

FROM (

SELECT u.id\_utilizator, u.nume\_utilizator,

COUNT(ii.id\_item) AS numar\_iteme

FROM Utilizator u, Inventar i, Inventar\_Item ii

WHERE u.id\_utilizator = i.id\_utilizator

AND i.id\_inventar = ii.id\_inventar

GROUP BY u.id\_utilizator, u.nume\_utilizator

ORDER BY COUNT(ii.id\_item) DESC

)

WHERE ROWNUM <= 3;

