Platforma digitala pentru jocuri video

Realizat de

Neculae Andrei-Fabian

Grupa 252

Cuprins

[Exercitiul 1 1](#_Toc155912066)

[Exercitiul 2 1](#_Toc155912067)

[Exercitiul 3 2](#_Toc155912068)

[Exercitiul 4 2](#_Toc155912069)

[Exercitiul 5 6](#_Toc155912070)

[Exercitiul 6 22](#_Toc155912071)

[Exercitiul 7 27](#_Toc155912072)

[Exercitiul 8 30](#_Toc155912073)

[Exercitiul 9 35](#_Toc155912074)

[Exercitiul 10 39](#_Toc155912075)

[Exercitiul 11 41](#_Toc155912076)

[Exercitiul 12 46](#_Toc155912077)

[Exercitiul 13 49](#_Toc155912078)

[Exercitiul 14 62](#_Toc155912079)

# Exercitiul 1

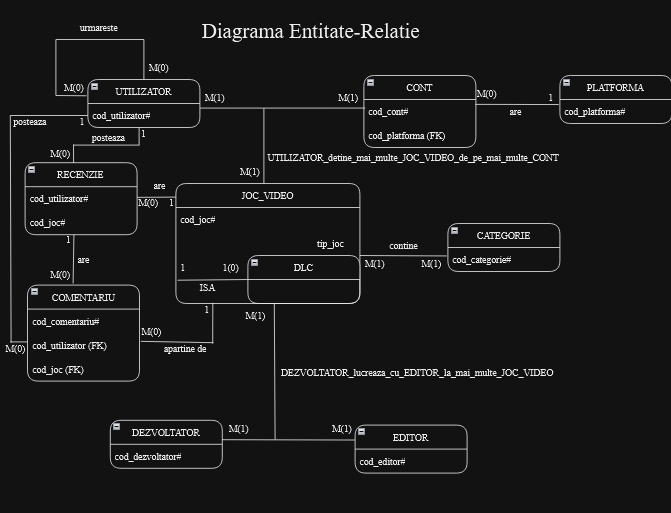
Cerinta: Prezentați pe scurt baza de date (utilitatea ei).

Rezolvare: Jocurile video fac parte din viata unui numar din ce in ce mai mare de oameni. Insa, de-a lungul anilor, odata cu evolutia jocurilor, dar si cu disputele financiare si competitiile dintre diverse mari companii, au aparut din ce in ce mai multe platforme digitale in aceasta industrie, fiecare detinand titlurile sale unice, ce nu pot fi gasite pe alte platforme. Din acest motiv, poate deveni incomod si chiar frustrant pentru utilizatori sa schimbe aplicatia de fiecare data cand doresc sa joace un anumit joc si chiar sa tina minte pe ce platforma se gaseste jocul pe care vor sa il deschida sau sa il cumpere. Aceste lucruri pot duce la o experienta destul de neplacuta ce nu ar trebui sa isi aiba locul in universul jocurilor video. Asadar, consider ca solutia este dezvoltarea unei noi platforme care sa le imbine pe toate celelalte la un loc. Utilizatorii se pot conecta cu fiecare cont in parte si isi pot importa toate librariile intr-un singur loc, pe care il pot configura dupa bunul plac. Astfel, jocurile pot fi accesate prin deschiderea unei singure aplicatii si pot fi gasite printr-o simpla cautare, ceea ce poate salva mult timp si multi nervi.

# Exercitiul 2

Cerinta: Realizati diagrama entitate-relatie (ERD): entitatile, relatiile si atributele trebuie definite in limba romana.

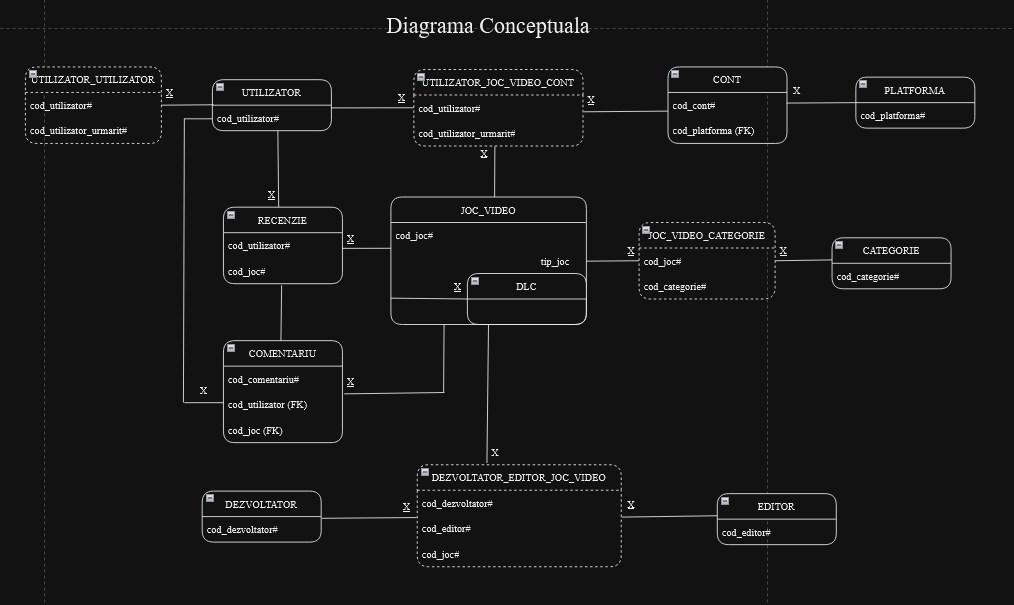
Rezolvare:



# Exercitiul 3

Cerinta: Pornind de la diagrama entitate-relatie realizati diagrama conceptuala a modelului propus, integrand toate atributele necesare: entitatile, relatiile si atributele trebuie definite in limba romana.

Rezolvare:

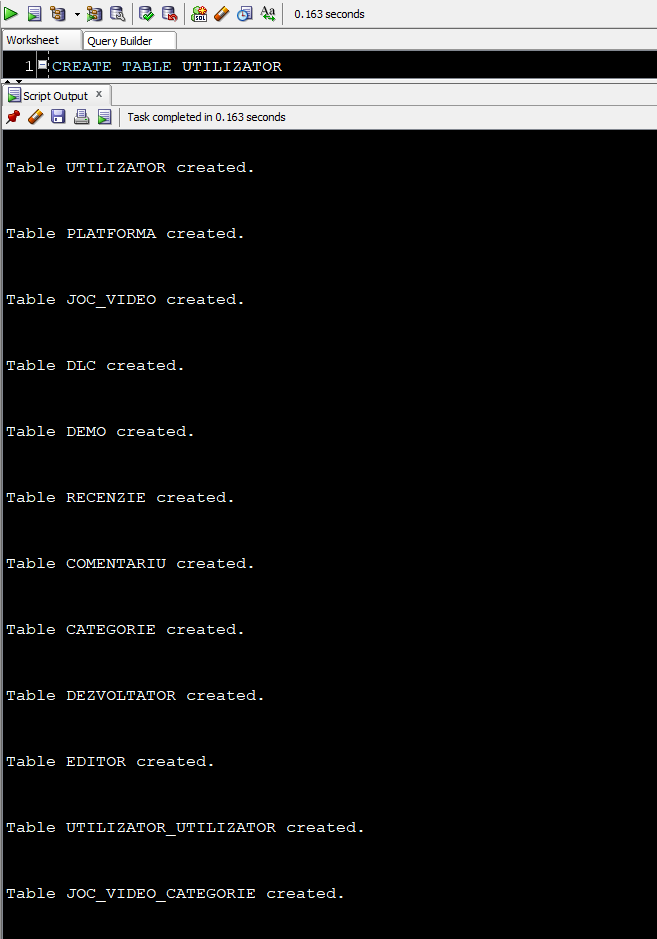


# Exercitiul 4

Cerinta: Implementati in Oracle diagrama conceptuala realizata: definiti toate tabelele, definind toate constrangerile de integritate necesare (chei primare, cheile externe etc).

Rezolvare:

|  |
| --- |
| **CREATE** **TABLE** UTILIZATOR  (  cod\_utilizator NUMBER(**7**) **constraint** pkey\_utilizator **PRIMARY** **KEY**,  nume VARCHAR2(**30**) **constraint** nume\_utilizator **NOT** **NULL**,  prenume VARCHAR2(**30**) **constraint** prenume\_utilizator **NOT** **NULL**,  parola VARCHAR(**30**) **constraint** parola\_utilizator **NOT** **NULL**,  email VARCHAR2(**30**) **constraint** email\_valid **CHECK**(email **LIKE** '%@%.%'),  data\_inregistrare DATE **DEFAULT** SYSDATE  );  **CREATE** **TABLE** PLATFORMA  (  cod\_platforma NUMBER(**3**) **constraint** pkey\_platforma **PRIMARY** **KEY**,  nume VARCHAR2(**30**) **constraint** nume\_platforma **NOT** **NULL**,  data\_lansare DATE,  site VARCHAR2(**50**)  );  **CREATE** **TABLE** CONT  (  cod\_cont NUMBER(**3**) **constraint** pkey\_cont **PRIMARY** **KEY**,  cod\_platforma NUMBER(**3**) **constraint** fkey\_cont\_platforma **REFERENCES** PLATFORMA(cod\_platforma),  nume VARCHAR2(**30**) **constraint** nume\_cont **NOT** **NULL**,  parola VARCHAR2(**30**) **constraint** parola\_cont **NOT** **NULL**  );  **CREATE** **TABLE** JOC\_VIDEO  (  cod\_joc NUMBER(**7**) **constraint** pkey\_joc\_video **PRIMARY** **KEY**,  nume VARCHAR2(**50**) **constraint** nume\_joc\_video **NOT** **NULL**,  data\_lansare DATE,  durata NUMBER(**5**, **1**),  tip VARCHAR2(**4**) **constraint** tip\_joc\_video **CHECK** (tip **IN** ('baza', 'dlc'))  );  **CREATE** **TABLE** DLC  (  cod\_joc NUMBER(**7**) **constraint** pkey\_dlc **REFERENCES** JOC\_VIDEO(cod\_joc),  cod\_joc\_baza NUMBER(**7**) **constraint** fkey\_joc\_dlc **REFERENCES** JOC\_VIDEO(cod\_joc),  **constraint** dlc\_valid **CHECK** (cod\_joc != cod\_joc\_baza)  );  **CREATE** **TABLE** RECENZIE  (  cod\_utilizator NUMBER(**7**) **constraint** fkey\_recenzie\_cod\_utilizator **REFERENCES** UTILIZATOR(cod\_utilizator),  cod\_joc NUMBER(**7**) **constraint** fkey\_recenzie\_cod\_joc **REFERENCES** JOC\_VIDEO(cod\_joc),  continut VARCHAR2(**2000**),  scor NUMBER(**1**) **constraint** scor\_valid **CHECK**(scor **BETWEEN** **1** **AND** **5**),  data\_postare DATE **DEFAULT** SYSDATE,  **constraint** pkey\_recenzie **PRIMARY** **KEY**(cod\_utilizator, cod\_joc)  );  **CREATE** **TABLE** COMENTARIU  (  cod\_comentariu NUMBER(**9**) **constraint** pkey\_comentariu **PRIMARY** **KEY**,  cod\_utilizator\_recenzie NUMBER(**7**),  cod\_joc NUMBER(**7**),  cod\_utilizator NUMBER(**7**) **constraint** fkey\_comentariu\_cod\_utilizator **REFERENCES** UTILIZATOR(cod\_utilizator),  continut VARCHAR2(**1000**) **constraint** continut\_comentariu **NOT** **NULL**,  data\_postare DATE **DEFAULT** SYSDATE,  **constraint** fkey\_comentariu\_recenzie **FOREIGN** **KEY**(cod\_utilizator\_recenzie, cod\_joc) **REFERENCES** RECENZIE(cod\_utilizator, cod\_joc)  );  **CREATE** **TABLE** CATEGORIE  (  cod\_categorie NUMBER(**3**) **constraint** pkey\_categorie **PRIMARY** **KEY**,  nume VARCHAR2(**30**) **constraint** nume\_categorie **NOT** **NULL**  );  **CREATE** **TABLE** DEZVOLTATOR  (  cod\_dezvoltator NUMBER(**5**) **constraint** pkey\_dezvoltator **PRIMARY** **KEY**,  nume VARCHAR2(**30**) **constraint** nume\_dezvoltator **NOT** **NULL**,  site VARCHAR2(**50**),  data\_infiintare DATE  );  **CREATE** **TABLE** EDITOR  (  cod\_editor NUMBER(**5**) **constraint** pkey\_editor **PRIMARY** **KEY**,  nume VARCHAR2(**30**) **constraint** nume\_editor **NOT** **NULL**,  site VARCHAR2(**50**),  data\_infiintare DATE  );  **CREATE** **TABLE** UTILIZATOR\_UTILIZATOR  (  cod\_utilizator NUMBER(**7**) **constraint** fkey\_urmarire\_utilizator1 **REFERENCES** UTILIZATOR(cod\_utilizator),  cod\_utilizator\_urmarit NUMBER(**7**) **constraint** fkey\_urmarire\_utilizator2 **REFERENCES** UTILIZATOR(cod\_utilizator),  **constraint** urmarire\_valid **CHECK**(cod\_utilizator != cod\_utilizator\_urmarit),  **constraint** pkey\_urmarire **PRIMARY** **KEY**(cod\_utilizator, cod\_utilizator\_urmarit)  );  **CREATE** **TABLE** JOC\_VIDEO\_CATEGORIE  (  cod\_joc NUMBER(**7**) **constraint** fkey\_continut\_joc **REFERENCES** JOC\_VIDEO(cod\_joc),  cod\_categorie NUMBER(**3**) **constraint** fkey\_continut\_categorie **REFERENCES** CATEGORIE(cod\_categorie),  **constraint** pkey\_continut **PRIMARY** **KEY**(cod\_joc, cod\_categorie)  );  **CREATE** **TABLE** UTILIZATOR\_JOC\_VIDEO\_CONT  (  cod\_utilizator NUMBER(**7**) **constraint** fkey\_detine\_utilizator **REFERENCES** UTILIZATOR(cod\_utilizator),  cod\_joc NUMBER(**7**) **constraint** fkey\_detine\_joc **REFERENCES** JOC\_VIDEO(cod\_joc),  cod\_cont NUMBER(**3**) **constraint** fkey\_detine\_cont **REFERENCES** CONT(cod\_cont),  pret NUMBER(**5**, **2**) **constraint** pret\_joc **NOT** **NULL**,  **constraint** pkey\_detine **PRIMARY** **KEY**(cod\_utilizator, cod\_joc, cod\_cont)  );  **CREATE** **TABLE** DEZVOLTATOR\_EDITOR\_JOC\_VIDEO  (  cod\_dezvoltator NUMBER(**5**) **constraint** fkey\_publica\_dezvoltator **REFERENCES** DEZVOLTATOR(cod\_dezvoltator),  cod\_editor NUMBER(**5**) **constraint** fkey\_publica\_editor **REFERENCES** EDITOR(cod\_editor),  cod\_joc NUMBER(**7**) **constraint** fkey\_publica\_joc **REFERENCES** JOC\_VIDEO(cod\_joc),  **constraint** pkey\_publica **PRIMARY** **KEY**(cod\_dezvoltator, cod\_editor, cod\_joc)  ); |

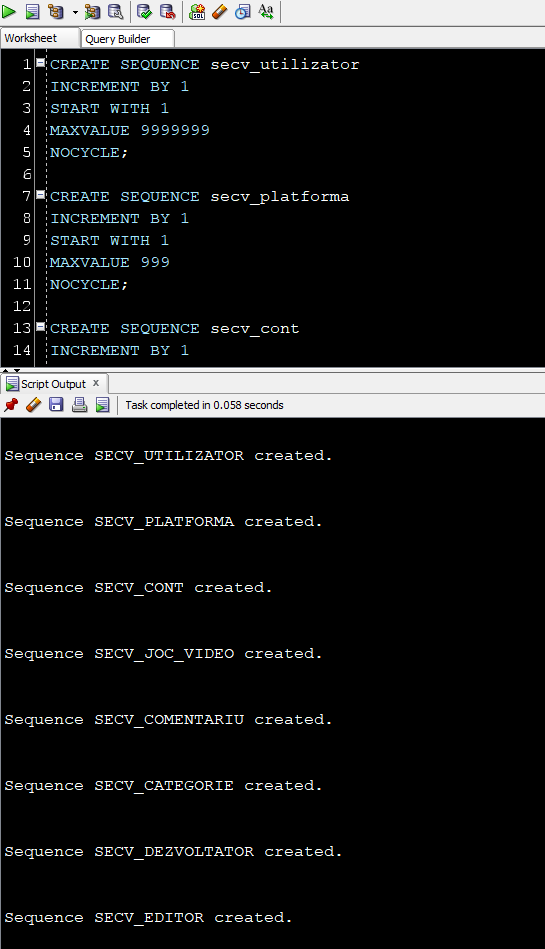


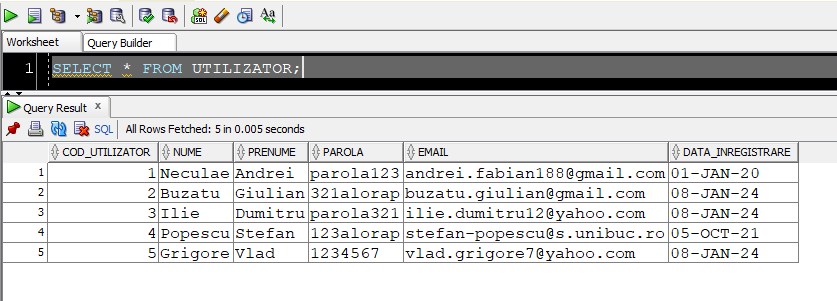
# Exercitiul 5

Cerinta: Adaugati informatii coerente in tabelele create (minim 5 inregistrari pentru fiecare entitate independenta; minim 10 inregistrari pentru tabela asociativa).

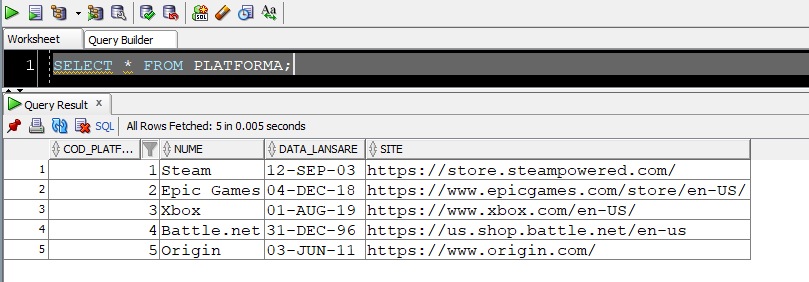
Rezolvare:

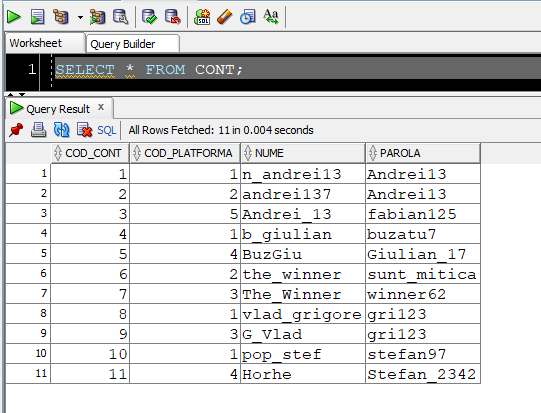
|  |
| --- |
| -- CREATE SEQUENCE  **CREATE** **SEQUENCE** secv\_utilizator  **INCREMENT** **BY** **1**  **START** **WITH** **1**  **MAXVALUE** **9999999**  NOCYCLE;  **CREATE** **SEQUENCE** secv\_platforma  **INCREMENT** **BY** **1**  **START** **WITH** **1**  **MAXVALUE** **999**  NOCYCLE;  **CREATE** **SEQUENCE** secv\_cont  **INCREMENT** **BY** **1**  **START** **WITH** **1**  **MAXVALUE** **9999**  NOCYCLE;  **CREATE** **SEQUENCE** secv\_joc\_video  **INCREMENT** **BY** **1**  **START** **WITH** **1**  **MAXVALUE** **9999999**  NOCYCLE;  **CREATE** **SEQUENCE** secv\_comentariu  **INCREMENT** **BY** **1**  **START** **WITH** **1**  **MAXVALUE** **9999999**  NOCYCLE;  **CREATE** **SEQUENCE** secv\_categorie  **INCREMENT** **BY** **1**  **START** **WITH** **1**  **MAXVALUE** **999**  NOCYCLE;  **CREATE** **SEQUENCE** secv\_dezvoltator  **INCREMENT** **BY** **1**  **START** **WITH** **1**  **MAXVALUE** **99999**  NOCYCLE;  **CREATE** **SEQUENCE** secv\_editor  **INCREMENT** **BY** **1**  **START** **WITH** **1**  **MAXVALUE** **99999**  NOCYCLE;  -- UTILIZATOR  **INSERT** **INTO** UTILIZATOR(cod\_utilizator, nume, prenume, parola, email, data\_inregistrare)  **VALUES** (secv\_utilizator**.**NEXTVAL, 'Neculae', 'Andrei', 'parola123', 'andrei.fabian188@gmail.com', '01-JAN-2020');  **INSERT** **INTO** UTILIZATOR(cod\_utilizator, nume, prenume, parola, email)  **VALUES** (secv\_utilizator**.**NEXTVAL, 'Buzatu', 'Giulian', '321alorap', 'buzatu.giulian@gmail.com');  **INSERT** **INTO** UTILIZATOR(cod\_utilizator, nume, prenume, parola, email)  **VALUES** (secv\_utilizator**.**NEXTVAL, 'Ilie', 'Dumitru', 'parola321', 'ilie.dumitru12@yahoo.com');  **INSERT** **INTO** UTILIZATOR(cod\_utilizator, nume, prenume, parola, email, data\_inregistrare)  **VALUES** (secv\_utilizator**.**NEXTVAL, 'Popescu', 'Stefan', '123alorap', 'stefan-popescu@s.unibuc.ro', '05-OCT-2021');  **INSERT** **INTO** UTILIZATOR(cod\_utilizator, nume, prenume, parola, email)  **VALUES** (secv\_utilizator**.**NEXTVAL, 'Grigore', 'Vlad', '1234567', 'vlad.grigore7@yahoo.com');  -- PLATFORMA  **INSERT** **INTO** PLATFORMA  **VALUES** (secv\_platforma**.**NEXTVAL, 'Steam', '12-SEP-2003', 'https://store.steampowered.com/');  **INSERT** **INTO** PLATFORMA  **VALUES** (secv\_platforma**.**NEXTVAL, 'Epic Games', '04-DEC-2018', 'https://www.epicgames.com/store/en-US/');  **INSERT** **INTO** PLATFORMA  **VALUES** (secv\_platforma**.**NEXTVAL, 'Xbox', '01-AUG-2019', 'https://www.xbox.com/en-US/');  **INSERT** **INTO** PLATFORMA  **VALUES** (secv\_platforma**.**NEXTVAL, 'Battle.net', '31-DEC-1996', 'https://us.shop.battle.net/en-us');  **INSERT** **INTO** PLATFORMA  **VALUES** (secv\_platforma**.**NEXTVAL, 'Origin', '03-JUN-2011', 'https://www.origin.com/');  -- CONT  **INSERT** **INTO** CONT  **VALUES** (secv\_cont**.**NEXTVAL, **1**, 'n\_andrei13', 'Andrei13');  **INSERT** **INTO** CONT  **VALUES** (secv\_cont**.**NEXTVAL, **2**, 'andrei137', 'Andrei13');  **INSERT** **INTO** CONT  **VALUES** (secv\_cont**.**NEXTVAL, **5**, 'Andrei\_13', 'fabian125');  **INSERT** **INTO** CONT  **VALUES** (secv\_cont**.**NEXTVAL, **1**, 'b\_giulian', 'buzatu7');  **INSERT** **INTO** CONT  **VALUES** (secv\_cont**.**NEXTVAL, **4**, 'BuzGiu', 'Giulian\_17');  **INSERT** **INTO** CONT  **VALUES** (secv\_cont**.**NEXTVAL, **2**, 'the\_winner', 'sunt\_mitica');  **INSERT** **INTO** CONT  **VALUES** (secv\_cont**.**NEXTVAL, **3**, 'The\_Winner', 'winner62');  **INSERT** **INTO** CONT  **VALUES** (secv\_cont**.**NEXTVAL, **1**, 'vlad\_grigore', 'gri123');  **INSERT** **INTO** CONT  **VALUES** (secv\_cont**.**NEXTVAL, **3**, 'G\_Vlad', 'gri123');  **INSERT** **INTO** CONT  **VALUES** (secv\_cont**.**NEXTVAL, **1**, 'pop\_stef', 'stefan97');  **INSERT** **INTO** CONT  **VALUES** (secv\_cont**.**NEXTVAL, **4**, 'Horhe', 'Stefan\_2342');  -- JOC\_VIDEO  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'Life is Strange 2', '27-SEP-2018', '16', 'baza');  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'Dead Space Remake', '27-JAN-2023', '11', 'baza');  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'Alan Wake', '14-MAY-2010', '11', 'baza');  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'Alan Wake The Signal', '12-OCT-2010', '1.5', 'dlc');  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'Alan Wake The Writer', '12-NOV-2010', '1.5', 'dlc');  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'Assassin''s Creed Syndicate', '23-OCT-2015', '18.5', 'baza');  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'Assassin''s Creed Syndicate Jack the Ripper', '15-DEC-2015', '3', 'dlc');  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'Assassin''s Creed Syndicate The Last Maharaja', '01-MAR-2016', '2.5', 'dlc');  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'Assassin''s Creed Syndicate The Dreadful Crimes', '11-APR-2016', '3.5', 'dlc');  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'Tell Me Why', '17-AUG-2020', '9.5', 'baza');  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'World of Warcraft', '23-NOV-2004', '250', 'baza');  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'Need For Speed Unbound', '29-NOV-2022', '23.5', 'baza');  **INSERT** **INTO** JOC\_VIDEO  **VALUES** (secv\_joc\_video**.**NEXTVAL, 'Pronty', '19-NOV-2021', '7.5', 'baza');  -- DLC  **INSERT** **INTO** DLC  **VALUES** (**4**, **3**);  **INSERT** **INTO** DLC  **VALUES** (**5**, **3**);  **INSERT** **INTO** DLC  **VALUES** (**7**, **6**);  **INSERT** **INTO** DLC  **VALUES** (**8**, **6**);  **INSERT** **INTO** DLC  **VALUES** (**9**, **6**);  -- RECENZIE  **INSERT** **INTO** RECENZIE(cod\_utilizator, cod\_joc, continut, scor)  **VALUES** (**1**, **2**, 'Bun jocul, dar nu e pentru mine.', **3**);  **INSERT** **INTO** RECENZIE(cod\_utilizator, cod\_joc, continut, scor, data\_postare)  **VALUES** (**4**, **3**, 'Mi-a placut foarte mult, recomand!', **5**, '09-JAN-2023');  **INSERT** **INTO** RECENZIE(cod\_utilizator, cod\_joc, continut, scor)  **VALUES** (**3**, **6**, 'Acest joc este o dezamagire, nu va pierdeti timpul si banii pe el', **1**);  **INSERT** **INTO** RECENZIE(cod\_utilizator, cod\_joc, continut, scor)  **VALUES** (**2**, **10**, 'Am pierdut prea mult timp in acest joc, mi-am facut foarte multi nervi, 10/10 recomand', **5**);  **INSERT** **INTO** RECENZIE(cod\_utilizator, cod\_joc, continut, scor, data\_postare)  **VALUES** (**3**, **13**, 'Jocul este foarte bun, dar este doar pentru persoanele mai rafinate', **4**, '01-JUN-2023');  **INSERT** **INTO** RECENZIE(cod\_utilizator, cod\_joc, continut, scor, data\_postare)  **VALUES** (**1**, **13**, 'Un joc excelent, abia astept sa se lanseze un dlc', **5**, '17-AUG-2020');  -- COMENTARIU  **INSERT** **INTO** COMENTARIU(cod\_comentariu, cod\_utilizator\_recenzie, cod\_joc, cod\_utilizator, continut)  **VALUES** (secv\_comentariu**.**NEXTVAL, **1**, **2**, **2**, 'Si mie mi s-a parut bun jocul, 100% il recomand mai departe');  **INSERT** **INTO** COMENTARIU(cod\_comentariu, cod\_utilizator\_recenzie, cod\_joc, cod\_utilizator, continut)  **VALUES** (secv\_comentariu**.**NEXTVAL, **1**, **2**, **3**, 'Nu inteleg de ce nu ti-a placut, mie mi s-a parut un joc foarte bun');  **INSERT** **INTO** COMENTARIU(cod\_comentariu, cod\_utilizator\_recenzie, cod\_joc, cod\_utilizator, continut)  **VALUES** (secv\_comentariu**.**NEXTVAL, **4**, **3**, **3**, 'Personal nu am gasit ceva care sa ma atraga la acest joc, mi s-a parut prea plictisitor');  **INSERT** **INTO** COMENTARIU(cod\_comentariu, cod\_utilizator\_recenzie, cod\_joc, cod\_utilizator, continut)  **VALUES** (secv\_comentariu**.**NEXTVAL, **3**, **6**, **1**, 'Mie nu mi s-a parut o dezamagire, ba din contra, il consider un joc foarte bun, desi inteleg de ce ar putea dezamagi unele persoane');  **INSERT** **INTO** COMENTARIU(cod\_comentariu, cod\_utilizator\_recenzie, cod\_joc, cod\_utilizator, continut)  **VALUES** (secv\_comentariu**.**NEXTVAL, **2**, **10**, **5**, 'Si eu am pierdut foarte mult timp in acest joc, nu am simtit efectiv cum trec orele! Foarte buna treaba din partea developerilor!');  **INSERT** **INTO** COMENTARIU(cod\_comentariu, cod\_utilizator\_recenzie, cod\_joc, cod\_utilizator, continut)  **VALUES** (secv\_comentariu**.**NEXTVAL, **3**, **13**, **2**, 'Nu ma consider o persoana ''rafinata'' si totusi nu mi s-a parut ca jocul ar fi prea slab, dar clar nu este de nota 10');  -- CATEGORIE  **INSERT** **INTO** CATEGORIE  **VALUES** (secv\_categorie**.**NEXTVAL, 'Actiune');  **INSERT** **INTO** CATEGORIE  **VALUES** (secv\_categorie**.**NEXTVAL, 'Aventura');  **INSERT** **INTO** CATEGORIE  **VALUES** (secv\_categorie**.**NEXTVAL, 'Science-fiction');  **INSERT** **INTO** CATEGORIE  **VALUES** (secv\_categorie**.**NEXTVAL, 'Horror');  **INSERT** **INTO** CATEGORIE  **VALUES** (secv\_categorie**.**NEXTVAL, 'Thriller');  **INSERT** **INTO** CATEGORIE  **VALUES** (secv\_categorie**.**NEXTVAL, 'Drama');  **INSERT** **INTO** CATEGORIE  **VALUES** (secv\_categorie**.**NEXTVAL, 'MMORPG');  **INSERT** **INTO** CATEGORIE  **VALUES** (secv\_categorie**.**NEXTVAL, 'Curse');  **INSERT** **INTO** CATEGORIE  **VALUES** (secv\_categorie**.**NEXTVAL, 'Metroidvania');  -- DEZVOLTATOR  **INSERT** **INTO** DEZVOLTATOR  **VALUES** (secv\_dezvoltator**.**NEXTVAL, 'DONTNOD Entertainment', 'https://dont-nod.com/en/', '01-MAY-2008');  **INSERT** **INTO** DEZVOLTATOR  **VALUES** (secv\_dezvoltator**.**NEXTVAL, 'Motive', 'https://www.ea.com/ea-studios/motive', '13-JULY-2015');  **INSERT** **INTO** DEZVOLTATOR  **VALUES** (secv\_dezvoltator**.**NEXTVAL, 'Remedy Entertainment', 'https://www.remedygames.com/', '18-AUG-1995');  **INSERT** **INTO** DEZVOLTATOR  **VALUES** (secv\_dezvoltator**.**NEXTVAL, 'Ubisoft Quebec', 'https://quebec.ubisoft.com/en/', '27-JUN-2005');  **INSERT** **INTO** DEZVOLTATOR  **VALUES** (secv\_dezvoltator**.**NEXTVAL, 'Blizzard Entertainment', 'https://www.blizzard.com/en-us/', '08-FEB-1991');  **INSERT** **INTO** DEZVOLTATOR  **VALUES** (secv\_dezvoltator**.**NEXTVAL, 'Criterion Games', 'https://www.ea.com/ea-studios/criterion-games', '01-JAN-1996');  **INSERT** **INTO** DEZVOLTATOR(cod\_dezvoltator, nume, site)  **VALUES** (secv\_dezvoltator**.**NEXTVAL, '18Light Game', 'https://18light.cc/en/');  **INSERT** **INTO** DEZVOLTATOR(cod\_dezvoltator, nume)  **VALUES** (secv\_dezvoltator**.**NEXTVAL, 'FunZone Games');  -- EDITOR  **INSERT** **INTO** EDITOR  **VALUES** (secv\_editor**.**NEXTVAL, 'Square Enix', 'https://square-enix-games.com/en\_US/home', '01-APR-2003');  **INSERT** **INTO** EDITOR  **VALUES** (secv\_editor**.**NEXTVAL, 'Electronic Arts', 'https://www.ea.com/', '27-MAY-1982');  **INSERT** **INTO** EDITOR  **VALUES** (secv\_editor**.**NEXTVAL, 'Remedy Entertainment', 'https://www.remedygames.com/', '18-AUG-1995');  **INSERT** **INTO** EDITOR  **VALUES** (secv\_editor**.**NEXTVAL, 'Ubisoft', 'https://www.ubisoft.com/en-us/', '28-MAR-1986');  **INSERT** **INTO** EDITOR  **VALUES** (secv\_editor**.**NEXTVAL, 'Xbox Game Studios', 'https://www.xbox.com/en-US/xbox-game-studios', '21-MAR-2000');  **INSERT** **INTO** EDITOR  **VALUES** (secv\_editor**.**NEXTVAL, 'Blizzard Entertainment', 'https://www.blizzard.com/en-us/', '08-FEB-1991');  **INSERT** **INTO** EDITOR(cod\_editor, nume, site)  **VALUES** (secv\_editor**.**NEXTVAL, '18Light Game', 'https://18light.cc/en/');  -- UTILIZATOR\_UTILIZATOR  **INSERT** **INTO** UTILIZATOR\_UTILIZATOR  **VALUES** (**1**, **2**);  **INSERT** **INTO** UTILIZATOR\_UTILIZATOR  **VALUES** (**1**, **3**);  **INSERT** **INTO** UTILIZATOR\_UTILIZATOR  **VALUES** (**1**, **5**);  **INSERT** **INTO** UTILIZATOR\_UTILIZATOR  **VALUES** (**2**, **1**);  **INSERT** **INTO** UTILIZATOR\_UTILIZATOR  **VALUES** (**2**, **3**);  **INSERT** **INTO** UTILIZATOR\_UTILIZATOR  **VALUES** (**2**, **4**);  **INSERT** **INTO** UTILIZATOR\_UTILIZATOR  **VALUES** (**3**, **1**);  **INSERT** **INTO** UTILIZATOR\_UTILIZATOR  **VALUES** (**3**, **2**);  **INSERT** **INTO** UTILIZATOR\_UTILIZATOR  **VALUES** (**3**, **5**);  **INSERT** **INTO** UTILIZATOR\_UTILIZATOR  **VALUES** (**4**, **2**);  **INSERT** **INTO** UTILIZATOR\_UTILIZATOR  **VALUES** (**5**, **1**);  **INSERT** **INTO** UTILIZATOR\_UTILIZATOR  **VALUES** (**5**, **4**);  -- JOC\_VIDEO\_CATEGORIE  -- Adaugam categoriile manual la jocurile de baza  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**1**, **2**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**1**, **6**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**2**, **1**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**2**, **2**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**2**, **3**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**2**, **4**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**3**, **1**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**3**, **2**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**3**, **5**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**3**, **6**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**6**, **1**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**6**, **2**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**10**, **2**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**10**, **6**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**11**, **7**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**12**, **8**);  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  **VALUES** (**13**, **9**);  -- Adaugam aceleasi categorii de la jocul de baza la dlc folosind o cerere  **INSERT** **INTO** JOC\_VIDEO\_CATEGORIE  (  **SELECT** d**.**cod\_joc, jvc**.**cod\_categorie  **FROM** JOC\_VIDEO jv  **JOIN** DLC d **ON** (d**.**cod\_joc\_baza = jv**.**cod\_joc)  **JOIN** JOC\_VIDEO\_CATEGORIE jvc **ON** (d**.**cod\_joc\_baza = jvc**.**cod\_joc)  );  -- UTILIZATOR\_JOC\_VIDEO\_CONT  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**2**, **1**, **4**, '29.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**5**, **2**, **8**, '19.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**4**, **3**, **10**, '59.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**4**, **4**, **11**, '0');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**1**, **6**, **2**, '24.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**1**, **7**, **2**, '5.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**1**, **8**, **2**, '5.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**1**, **9**, **2**, '5.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**3**, **6**, **7**, '19.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**1**, **10**, **2**, '114.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**1**, **12**, **1**, '39.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**1**, **13**, **1**, '9.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**3**, **13**, **7**, '14.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**5**, **10**, **9**, '0');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**2**, **10**, **5**, '129.99');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**4**, **11**, **11**, '0');  **INSERT** **INTO** UTILIZATOR\_JOC\_VIDEO\_CONT  **VALUES** (**1**, **2**, **3**, '24.99');  -- DEZVOLTATOR\_EDITOR\_JOC\_VIDEO  -- Adaugam dezvoltatori si editori manual la jocurile de baza  **INSERT** **INTO** DEZVOLTATOR\_EDITOR\_JOC\_VIDEO  **VALUES** (**1**, **1**, **1**);  **INSERT** **INTO** DEZVOLTATOR\_EDITOR\_JOC\_VIDEO  **VALUES** (**2**, **2**, **2**);  **INSERT** **INTO** DEZVOLTATOR\_EDITOR\_JOC\_VIDEO  **VALUES** (**3**, **3**, **3**);  **INSERT** **INTO** DEZVOLTATOR\_EDITOR\_JOC\_VIDEO  **VALUES** (**4**, **4**, **6**);  **INSERT** **INTO** DEZVOLTATOR\_EDITOR\_JOC\_VIDEO  **VALUES** (**1**, **5**, **10**);  **INSERT** **INTO** DEZVOLTATOR\_EDITOR\_JOC\_VIDEO  **VALUES** (**5**, **6**, **11**);  **INSERT** **INTO** DEZVOLTATOR\_EDITOR\_JOC\_VIDEO  **VALUES** (**6**, **2**, **12**);  **INSERT** **INTO** DEZVOLTATOR\_EDITOR\_JOC\_VIDEO  **VALUES** (**7**, **7**, **13**);  **INSERT** **INTO** DEZVOLTATOR\_EDITOR\_JOC\_VIDEO  **VALUES** (**8**, **7**, **13**);  -- Adaugam aceiasi dezvoltatori si aceiasi editori de la jocul de baza la dlc folosind o cerere  **INSERT** **INTO** DEZVOLTATOR\_EDITOR\_JOC\_VIDEO  (  **SELECT** dejv**.**cod\_dezvoltator, dejv**.**cod\_editor, d**.**cod\_joc  **FROM** JOC\_VIDEO jv  **JOIN** DLC d **ON** (d**.**cod\_joc\_baza = jv**.**cod\_joc)  **JOIN** DEZVOLTATOR\_EDITOR\_JOC\_VIDEO dejv **ON** (d**.**cod\_joc\_baza = dejv**.**cod\_joc)  ); |

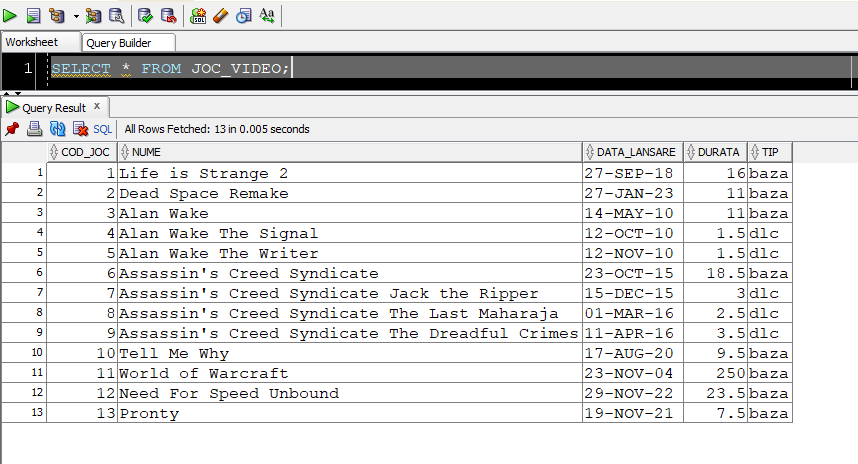


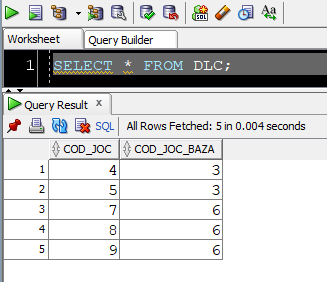


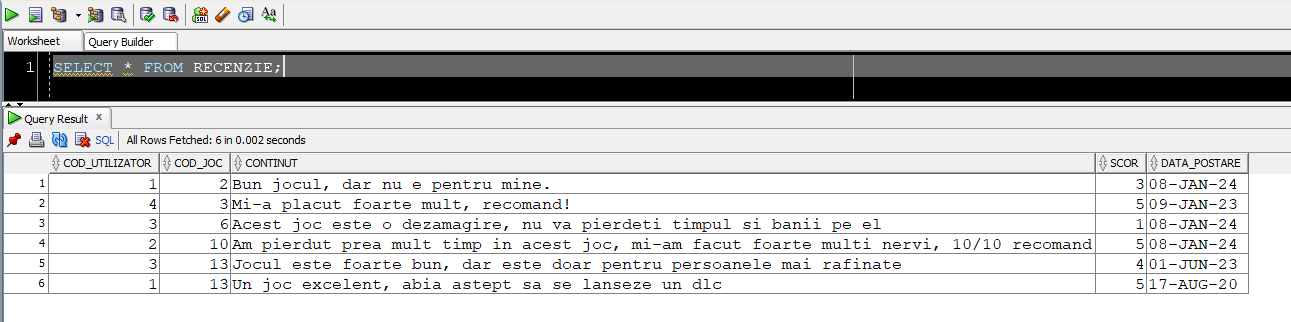
‚

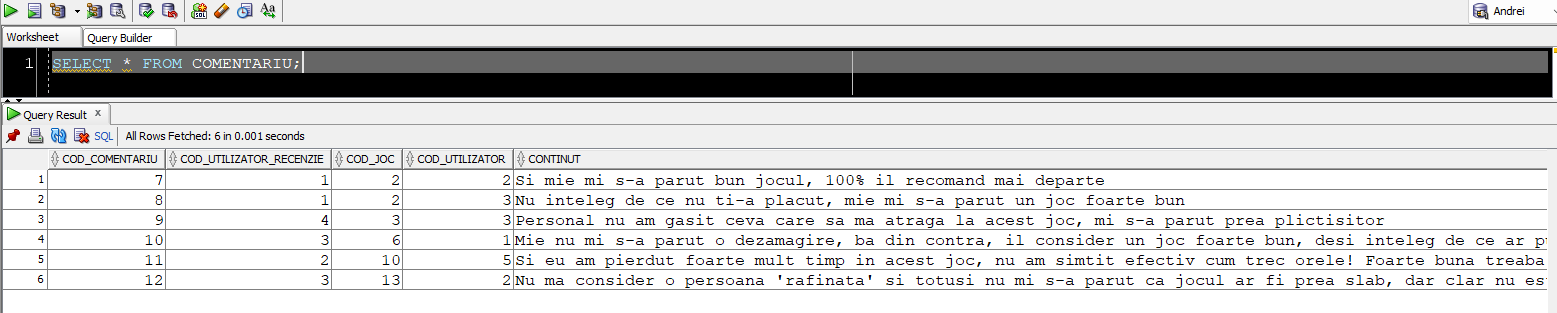


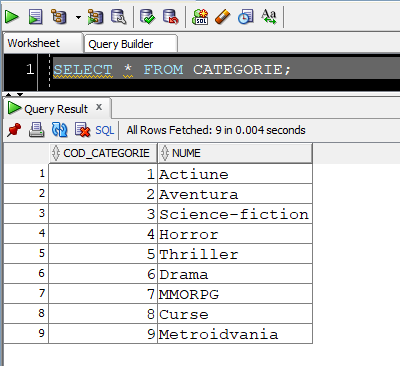


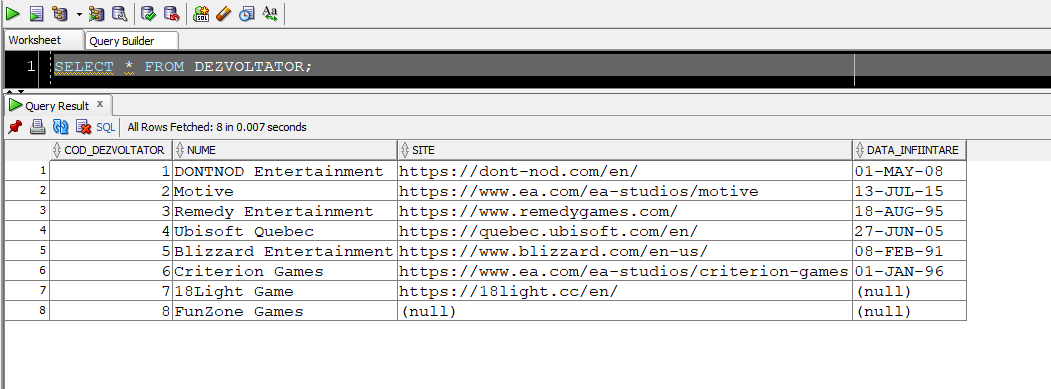


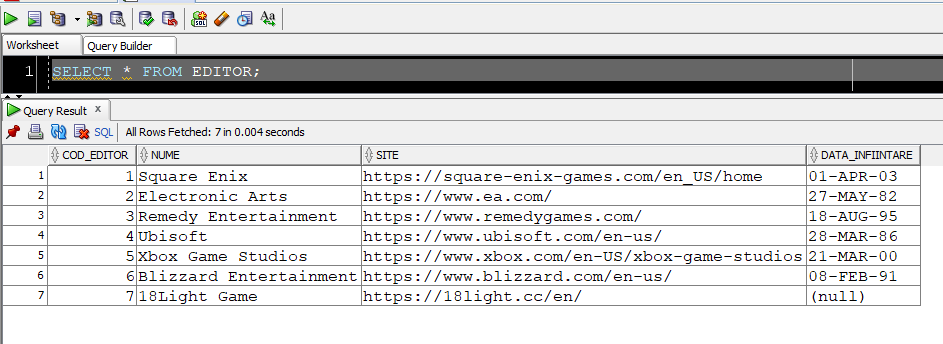


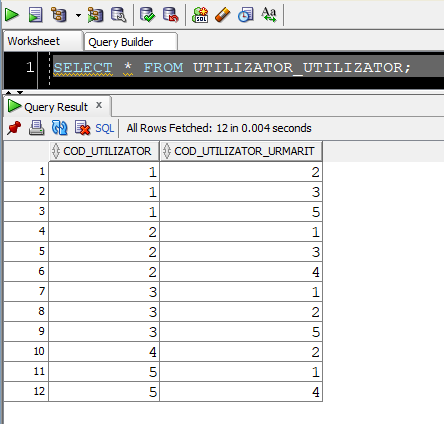


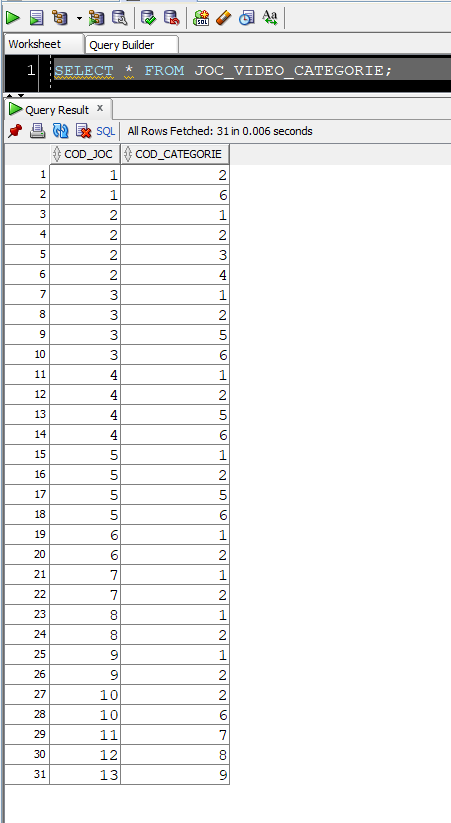


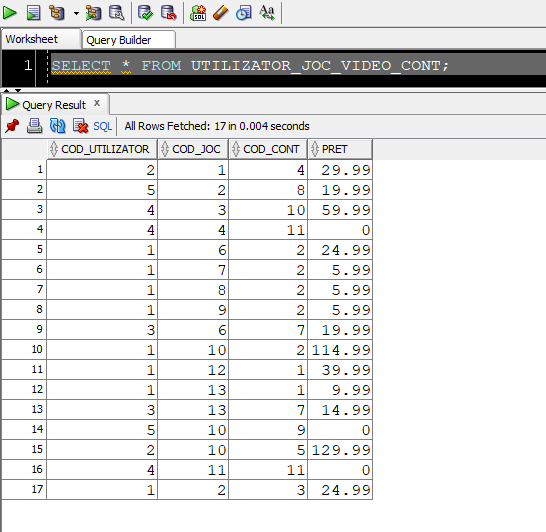


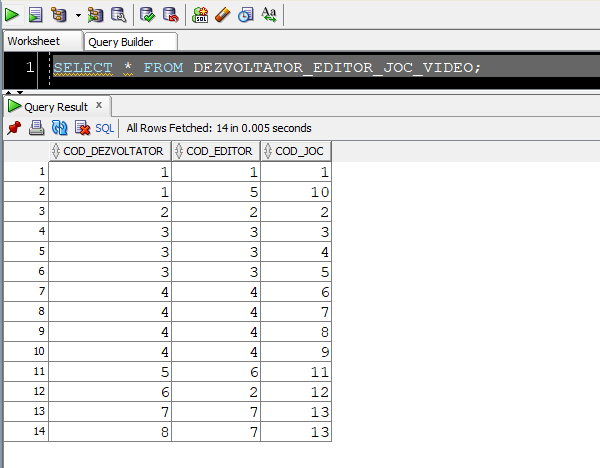










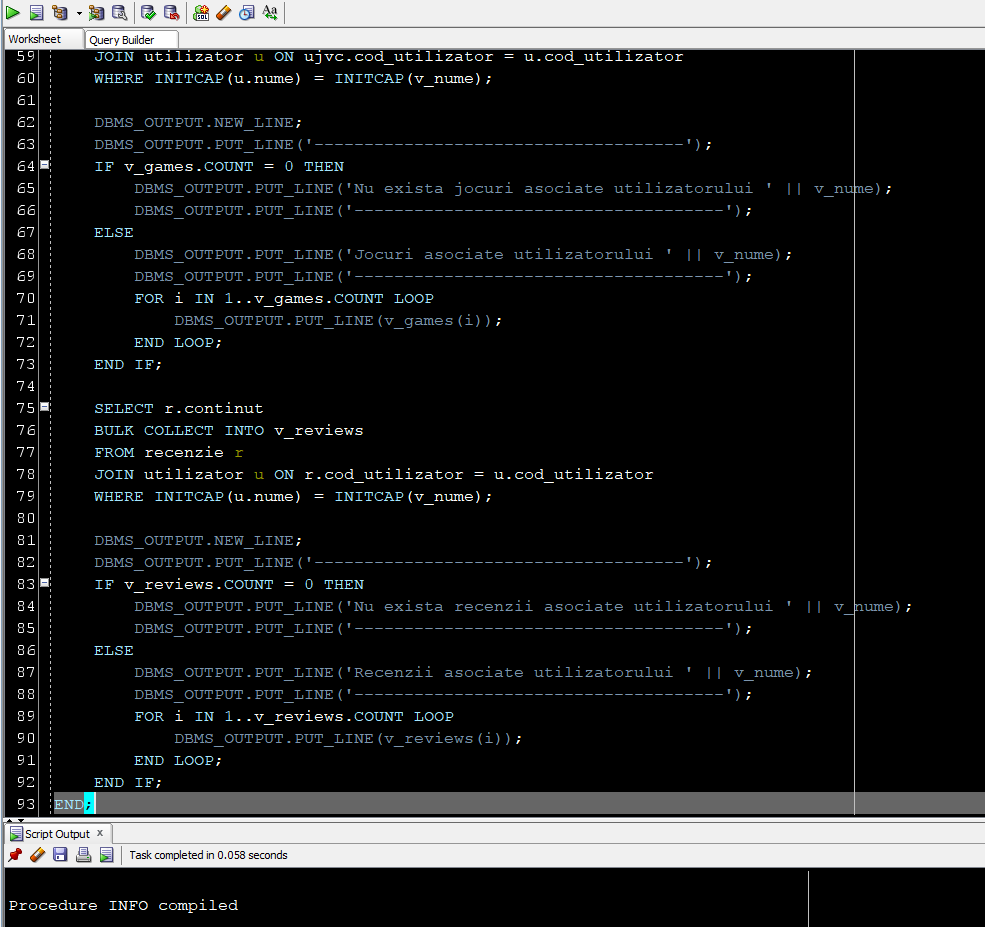


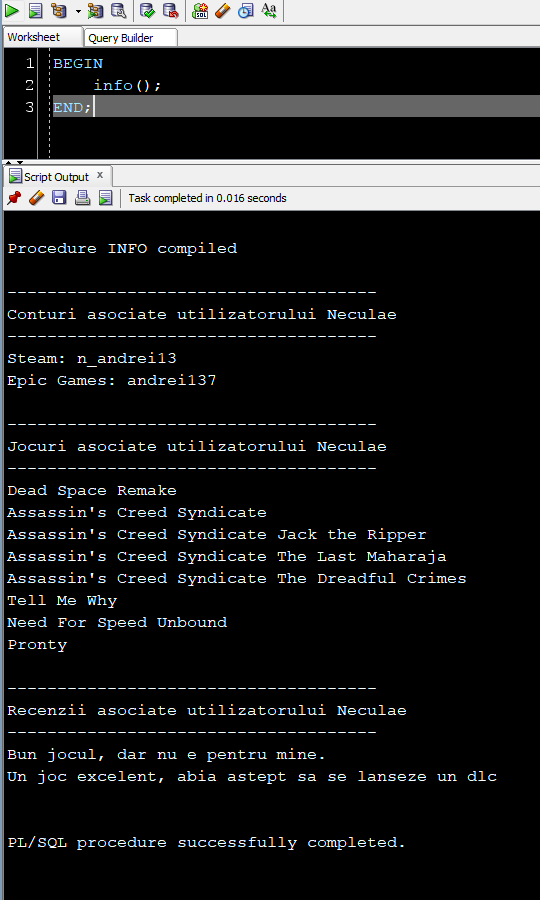
# Exercitiul 6

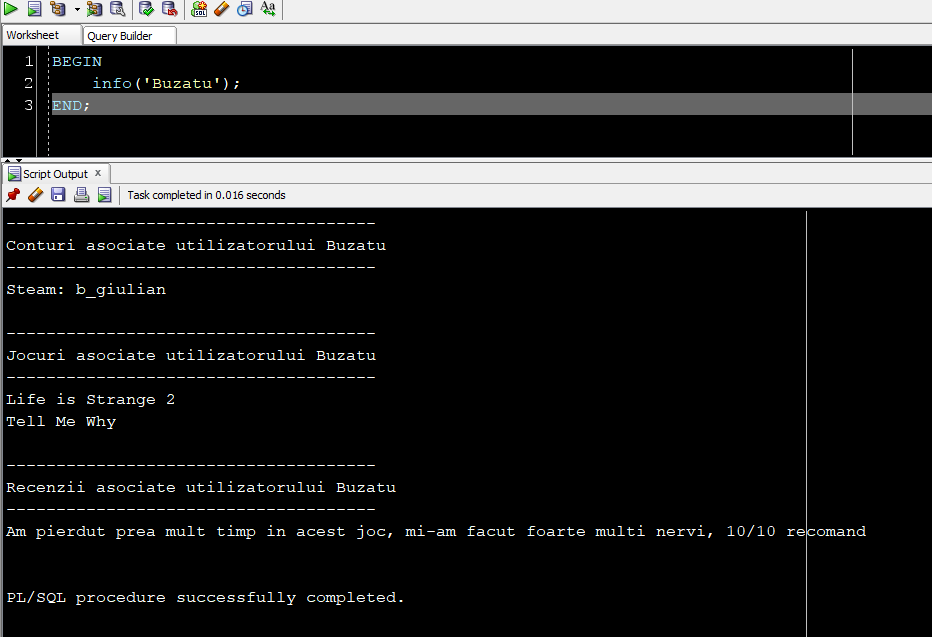
Cerinta: Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogrm stocat independent care sa utilizeze toate cele 3 tipuri de colectii studiate. Apelati subprogramul.

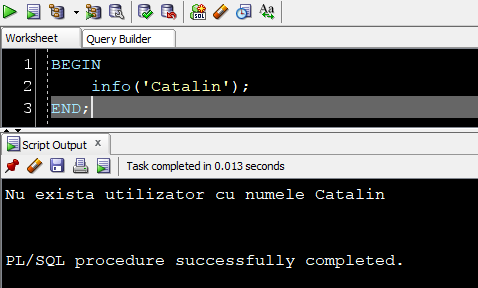
Rezolvare: Definiti un subprogram stocat independent care primeste un nume de utilizator ca parametru si afiseaza numele conturilor de Steam, Epic Games si Xbox ale acestuia, toate jocurile pe care le detine si toate recenziile lasate de acesta, folosind cate un tip de date diferit pentru fiecare. In cazul in care nu exista utilizator cu numele respectiv, se va afisa un mesaj de eroare. In cazul in care exista mai multi utilizatori cu acelasi nume, se vor afisa informatiile pentru fiecare.

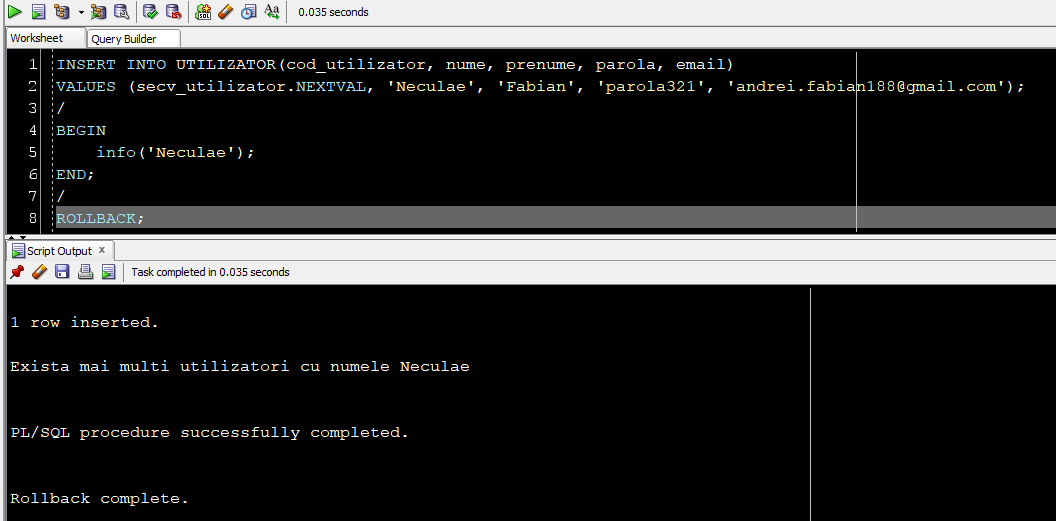
|  |
| --- |
| **CREATE** **OR** **REPLACE** **PROCEDURE** info  (  v\_nume utilizator**.**nume%TYPE **DEFAULT** 'Neculae'  )  **IS**  **TYPE** pair **IS** RECORD  (  platform platforma**.**nume%TYPE,  username cont**.**nume%TYPE  );  **TYPE** vector **IS** VARRAY(**3**) **OF** pair;  **TYPE** tablou\_indexat **IS** **TABLE** **OF** joc\_video**.**nume%TYPE **INDEX** **BY** PLS\_INTEGER;  **TYPE** tablou\_imbricat **IS** **TABLE** **OF** recenzie**.**continut%TYPE;  v\_accounts vector := vector();  v\_games tablou\_indexat;  v\_reviews tablou\_imbricat := tablou\_imbricat();  v\_nr\_utilizator NUMBER;  **BEGIN**  **SELECT** COUNT(\*)  **INTO** v\_nr\_utilizator  **FROM** utilizator  **WHERE** INITCAP(utilizator**.**nume) = INITCAP(v\_nume);  **IF** v\_nr\_utilizator = **0** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Nu exista utilizator cu numele ' || v\_nume);  **RETURN**;  **END** **IF**;  **IF** v\_nr\_utilizator > **1** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Exista mai multi utilizatori cu numele ' || v\_nume);  **RETURN**;  **END** **IF**;  **SELECT** **DISTINCT** p**.**nume, c**.**nume  BULK COLLECT **INTO** v\_accounts  **FROM** cont c  **JOIN** utilizator\_joc\_video\_cont ujvc **ON** c**.**cod\_cont = ujvc**.**cod\_cont  **JOIN** platforma p **ON** c**.**cod\_platforma = p**.**cod\_platforma  **JOIN** utilizator u **ON** ujvc**.**cod\_utilizator = u**.**cod\_utilizator  **WHERE** INITCAP(u**.**nume) = INITCAP(v\_nume) **AND** INITCAP(p**.**nume) **IN** ('Steam', 'Epic Games', 'Xbox');  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **IF** v\_accounts**.**COUNT = **0** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Nu exista conturi asociate utilizatorului ' || v\_nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **ELSE**  DBMS\_OUTPUT**.**PUT\_LINE('Conturi asociate utilizatorului ' || v\_nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **FOR** i **IN** **1..**v\_accounts**.**COUNT **LOOP**  DBMS\_OUTPUT**.**PUT\_LINE(v\_accounts(i)**.**platform || ': ' || v\_accounts(i)**.**username);  **END** **LOOP**;  **END** **IF**;  **SELECT** jv**.**nume  BULK COLLECT **INTO** v\_games  **FROM** joc\_video jv  **JOIN** utilizator\_joc\_video\_cont ujvc **ON** jv**.**cod\_joc = ujvc**.**cod\_joc  **JOIN** utilizator u **ON** ujvc**.**cod\_utilizator = u**.**cod\_utilizator  **WHERE** INITCAP(u**.**nume) = INITCAP(v\_nume);  DBMS\_OUTPUT**.**NEW\_LINE;  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **IF** v\_games**.**COUNT = **0** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Nu exista jocuri asociate utilizatorului ' || v\_nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **ELSE**  DBMS\_OUTPUT**.**PUT\_LINE('Jocuri asociate utilizatorului ' || v\_nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **FOR** i **IN** **1..**v\_games**.**COUNT **LOOP**  DBMS\_OUTPUT**.**PUT\_LINE(v\_games(i));  **END** **LOOP**;  **END** **IF**;    **SELECT** r**.**continut  BULK COLLECT **INTO** v\_reviews  **FROM** recenzie r  **JOIN** utilizator u **ON** r**.**cod\_utilizator = u**.**cod\_utilizator  **WHERE** INITCAP(u**.**nume) = INITCAP(v\_nume);  DBMS\_OUTPUT**.**NEW\_LINE;  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **IF** v\_reviews**.**COUNT = **0** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Nu exista recenzii asociate utilizatorului ' || v\_nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **ELSE**  DBMS\_OUTPUT**.**PUT\_LINE('Recenzii asociate utilizatorului ' || v\_nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **FOR** i **IN** **1..**v\_reviews**.**COUNT **LOOP**  DBMS\_OUTPUT**.**PUT\_LINE(v\_reviews(i));  **END** **LOOP**;  **END** **IF**;  **END**; |









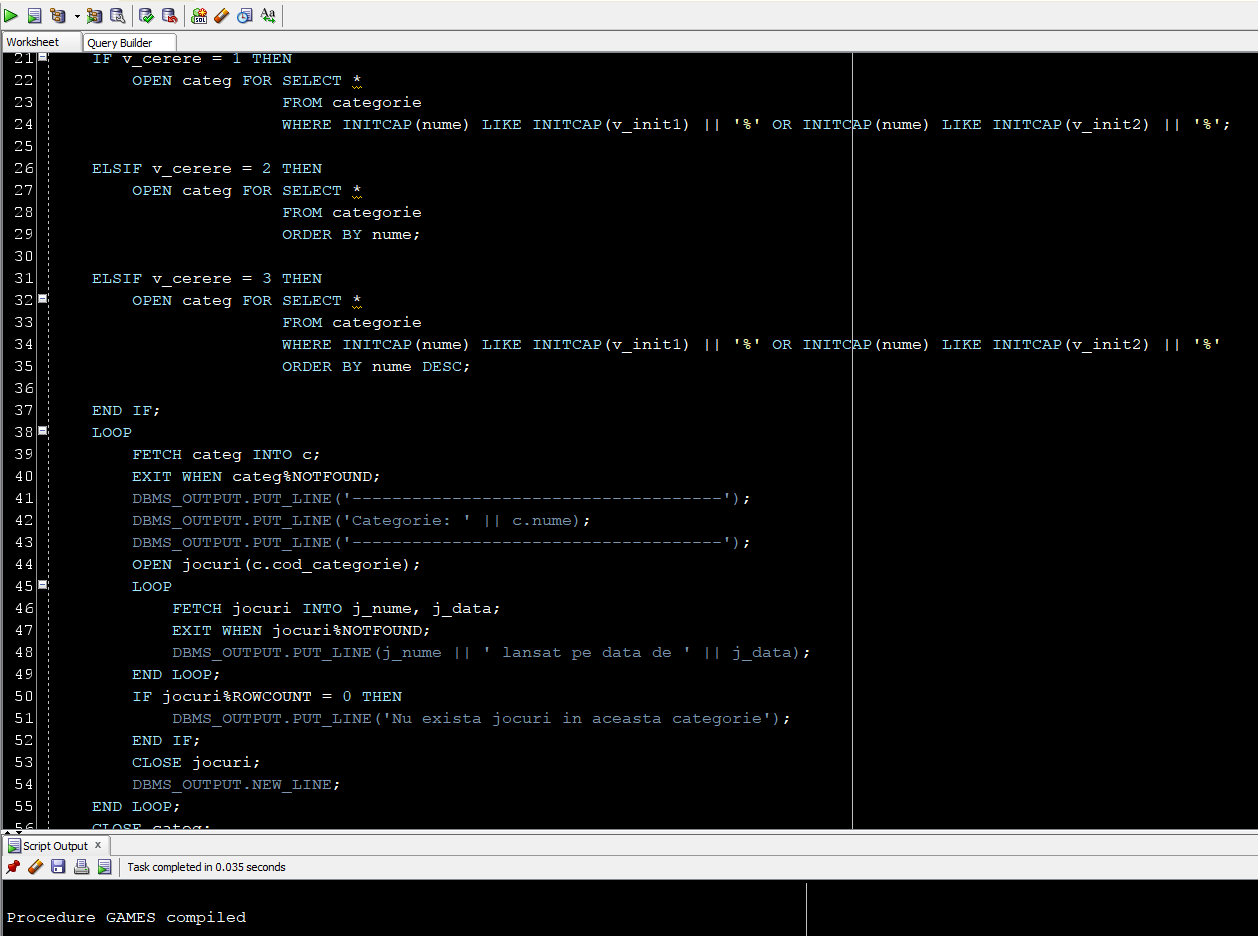


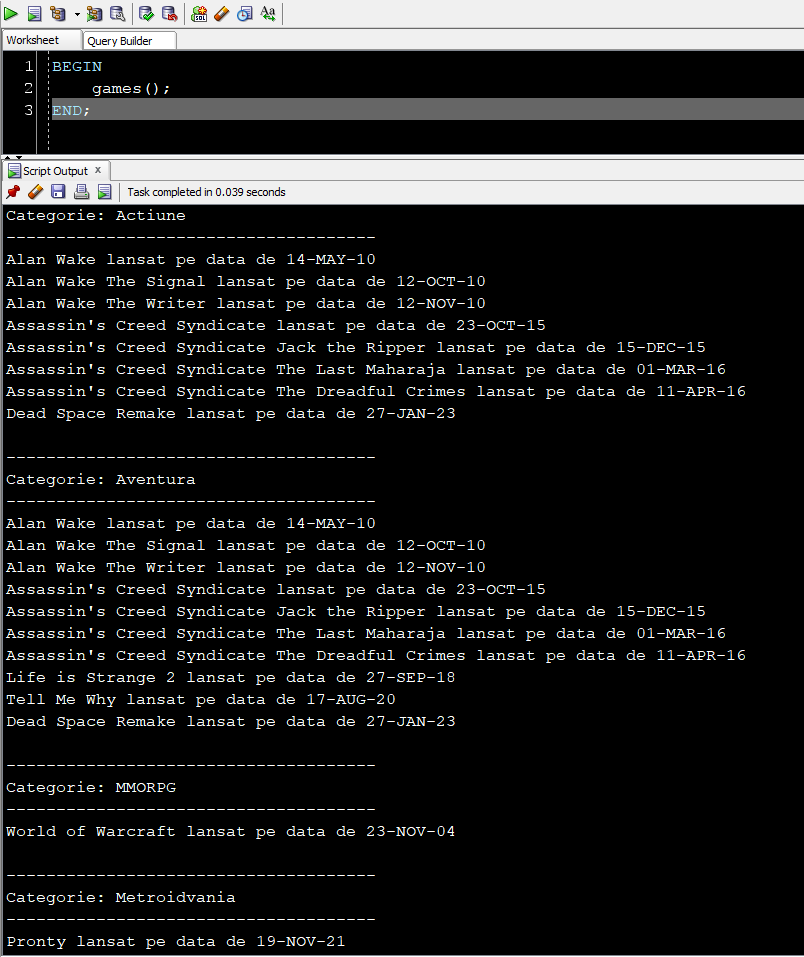
# Exercitiul 7

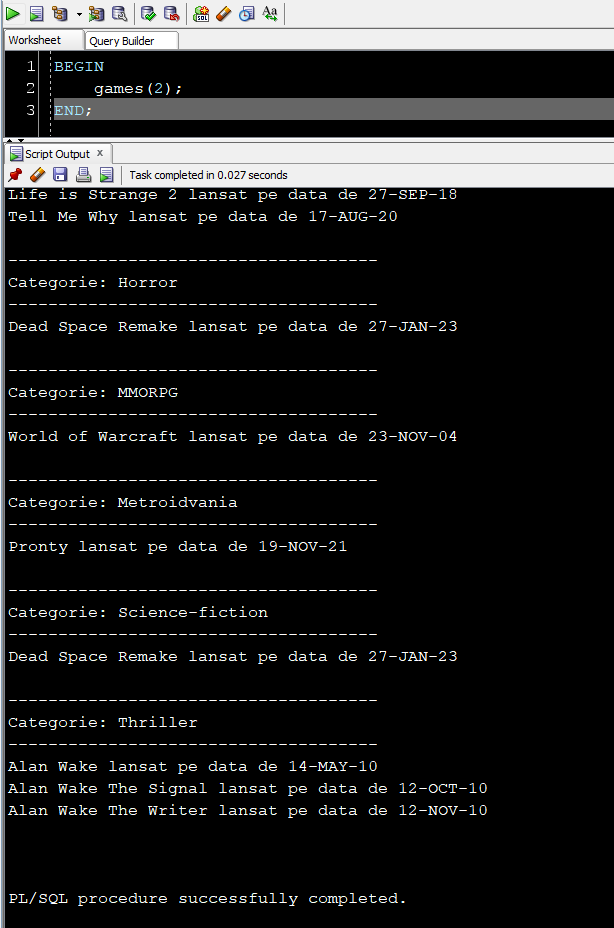
Cerinta: Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent care sa utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celalalt cursor. Apelati subprogramul.

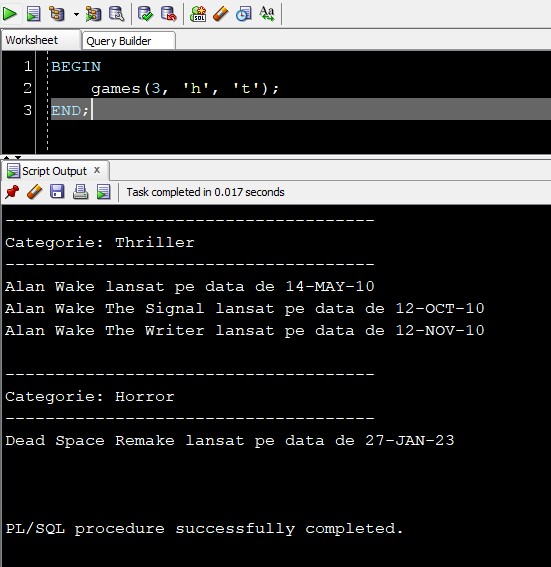
Rezolvare: Definiti un subprogram stocat independent care primeste o cerere si doua litere ca parametrii. Daca cererea este 1 se vor parcurge toate categoriile care incep cu literele date, daca este 2 se vor parcurge toate categoriile ordonate crescator dupa nume, si daca este 3 se vor parcurge categoriile care incep cu literele date, in ordine descrescatoare. Pentru fiecare categorie, se vor afisa toate jocurile ce apartin acesteia, in ordinea crescatoare a datei de lansare, sau un mesaj corespunzator in cazul in care nu exista jocuri in categoria respectiva.

|  |
| --- |
| **CREATE** **OR** **REPLACE** **PROCEDURE** games  (  v\_cerere NUMBER **DEFAULT** **1**,  v\_init1 VARCHAR2 **DEFAULT** 'A',  v\_init2 VARCHAR2 **DEFAULT** 'M'  )  **IS**  **TYPE** cursor\_dinamic **IS** **REF** **CURSOR** **RETURN** categorie%ROWTYPE;  categ cursor\_dinamic;  **CURSOR** jocuri(v\_categ categorie**.**cod\_categorie%TYPE) **IS**  **SELECT** jv**.**nume, data\_lansare  **FROM** joc\_video jv  **JOIN** joc\_video\_categorie jvc **ON** jv**.**cod\_joc = jvc**.**cod\_joc  **JOIN** categorie c **ON** jvc**.**cod\_categorie = c**.**cod\_categorie  **WHERE** c**.**cod\_categorie = v\_categ  **ORDER** **BY** data\_lansare;  c categorie%ROWTYPE;  j\_nume joc\_video**.**nume%TYPE;  j\_data joc\_video**.**data\_lansare%TYPE;  **BEGIN**  **IF** v\_cerere = **1** **THEN**  **OPEN** categ **FOR** **SELECT** \*  **FROM** categorie  **WHERE** INITCAP(nume) **LIKE** INITCAP(v\_init1) || '%' **OR** INITCAP(nume) **LIKE** INITCAP(v\_init2) || '%';  **ELSIF** v\_cerere = **2** **THEN**  **OPEN** categ **FOR** **SELECT** \*  **FROM** categorie  **ORDER** **BY** nume;  **ELSIF** v\_cerere = **3** **THEN**  **OPEN** categ **FOR** **SELECT** \*  **FROM** categorie  **WHERE** INITCAP(nume) **LIKE** INITCAP(v\_init1) || '%' **OR** INITCAP(nume) **LIKE** INITCAP(v\_init2) || '%'  **ORDER** **BY** nume **DESC**;  **END** **IF**;  **LOOP**  **FETCH** categ **INTO** c;  **EXIT** **WHEN** categ%NOTFOUND;  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  DBMS\_OUTPUT**.**PUT\_LINE('Categorie: ' || c**.**nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **OPEN** jocuri(c**.**cod\_categorie);  **LOOP**  **FETCH** jocuri **INTO** j\_nume, j\_data;  **EXIT** **WHEN** jocuri%NOTFOUND;  DBMS\_OUTPUT**.**PUT\_LINE(j\_nume || ' lansat pe data de ' || j\_data);  **END** **LOOP**;  **IF** jocuri%ROWCOUNT = **0** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Nu exista jocuri in aceasta categorie');  **END** **IF**;  **CLOSE** jocuri;  DBMS\_OUTPUT**.**NEW\_LINE;  **END** **LOOP**;  **CLOSE** categ;  **END**; |







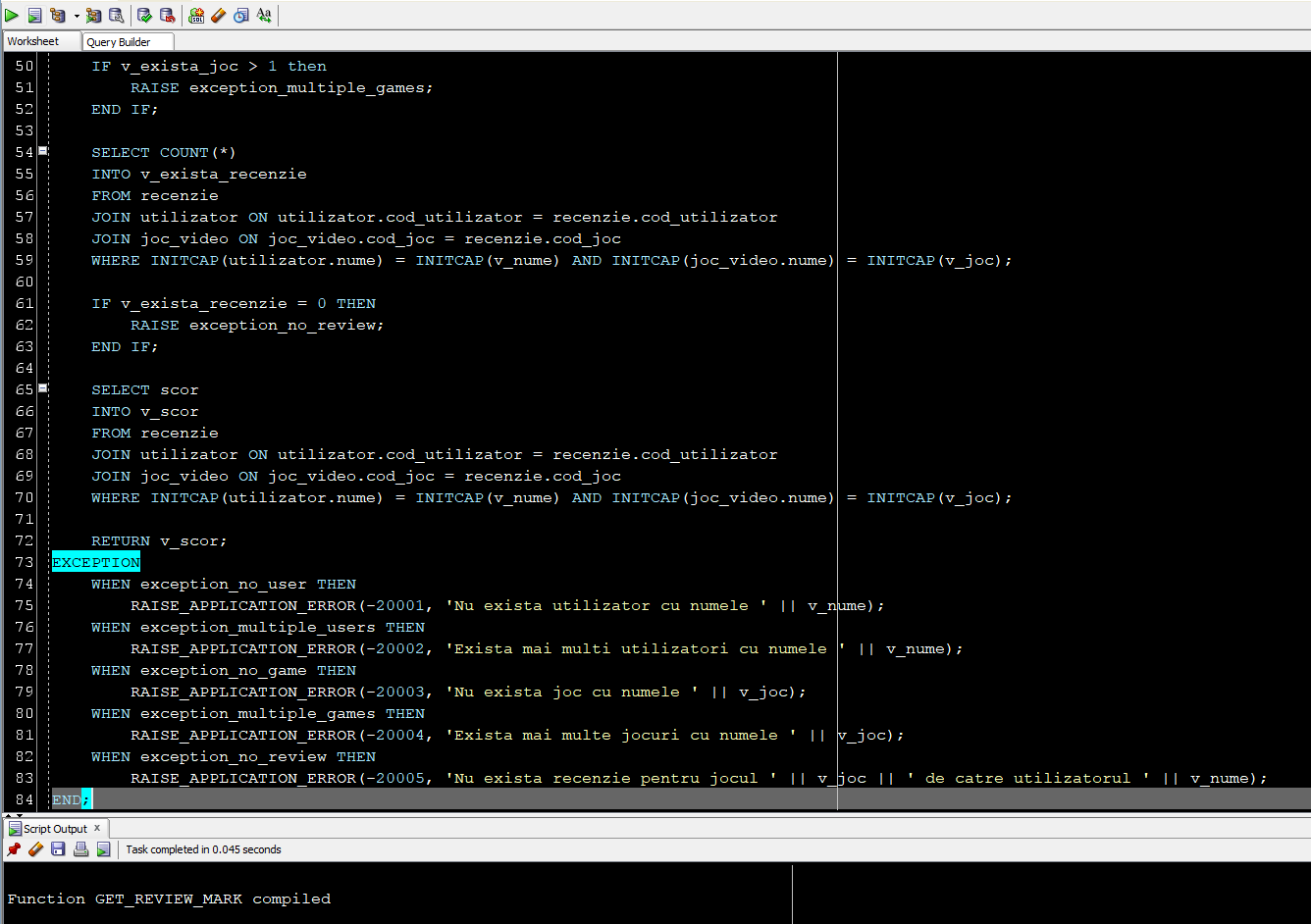


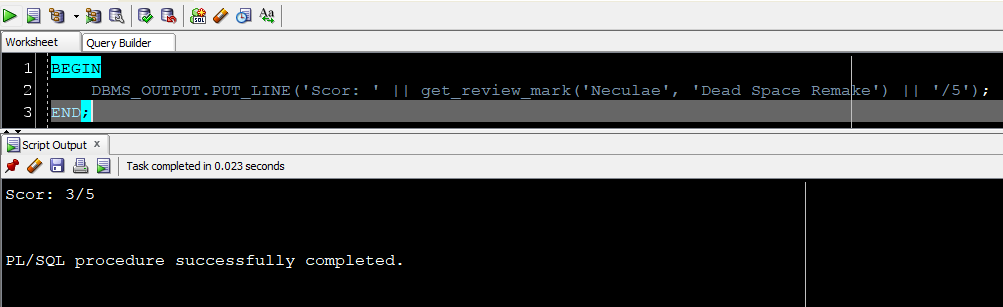
# Exercitiul 8

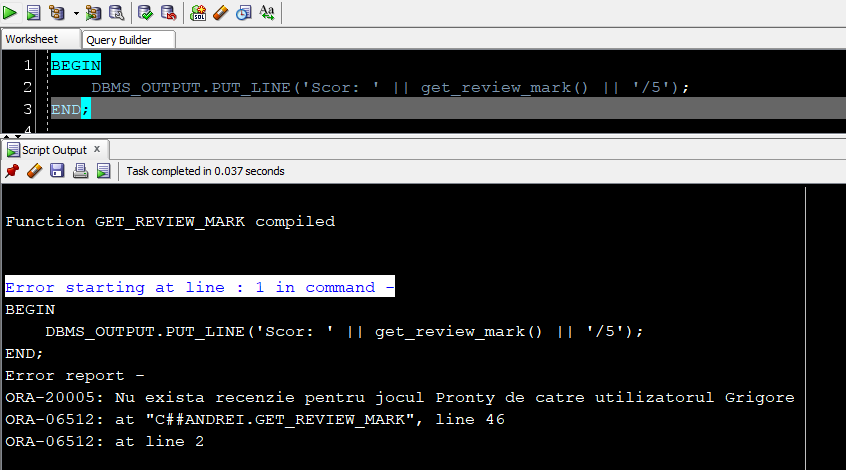
Cerinta: Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent de tip functie care sa utilizeze intr-o singura comanda SQL 3 dintre tabelele definite. Definiti minim 2 exceptii proprii. Apelati subprogramul astfel incat sa evidentiati toate cazurile definite si tratate.

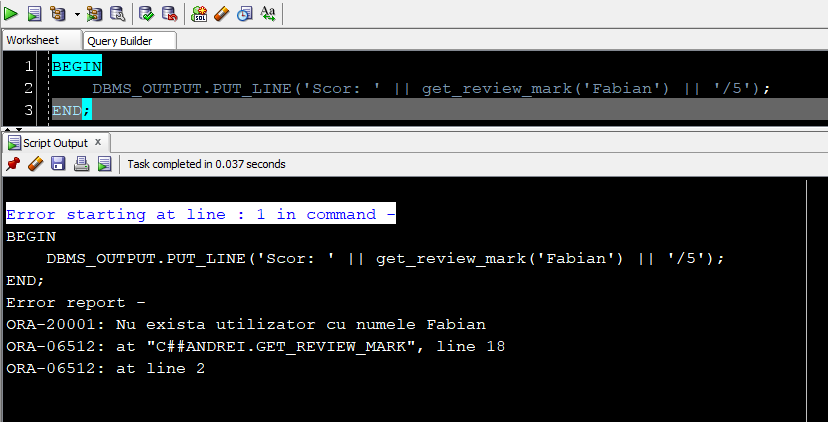
Rezolvare: Definiti un subprogram stocat independent de tip functie care primeste numele unui utilizator si numele unui joc ca parametrii si afiseaza nota pe care utilizatorul a dat-o jocului respectiv. Tratati cazurile in care nu exista numele utilizatorului sau al jocului, cazul in care exista mai multi utilizatori sau mai multe jocuri cu acelasi nume si cazul in care utilizatorul nu a lasat recenzie jocului respectiv (nu a dorit sa lase recenzie sau nu detine jocul).

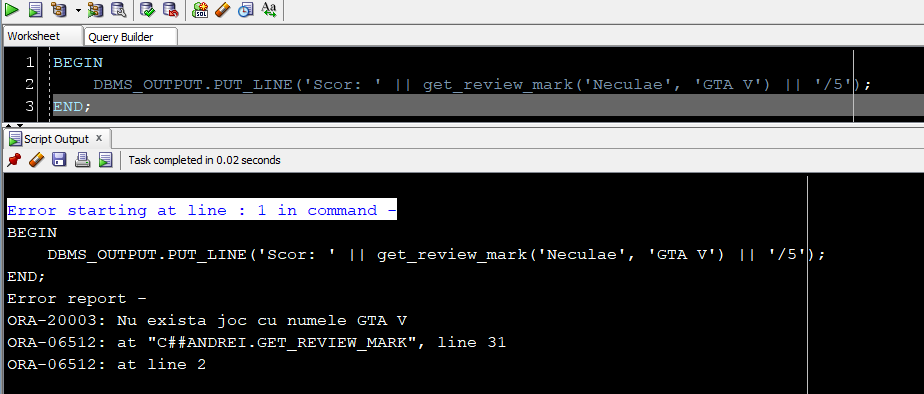
|  |
| --- |
| **CREATE** **OR** **REPLACE** **FUNCTION** get\_review\_mark  (  v\_nume utilizator**.**nume%TYPE **DEFAULT** 'Grigore',  v\_joc joc\_video**.**nume%TYPE **DEFAULT** 'Pronty'  ) **RETURN** recenzie**.**scor%TYPE  **IS**  v\_scor NUMBER;  v\_nr\_utilizator NUMBER;  v\_nr\_joc NUMBER;  v\_nr\_recenzie NUMBER;  exception\_no\_user **EXCEPTION**;  PRAGMA EXCEPTION\_INIT(exception\_no\_user, -**20001**);  exception\_multiple\_users **EXCEPTION**;  PRAGMA EXCEPTION\_INIT(exception\_multiple\_users, -**20002**);  exception\_no\_game **EXCEPTION**;  PRAGMA EXCEPTION\_INIT(exception\_no\_game, -**20003**);  exception\_multiple\_games **EXCEPTION**;  PRAGMA EXCEPTION\_INIT(exception\_multiple\_games, -**20004**);  exception\_no\_review **EXCEPTION**;  PRAGMA EXCEPTION\_INIT(exception\_no\_review, -**20005**);  **BEGIN**  **SELECT** COUNT(\*)  **INTO** v\_nr\_utilizator  **FROM** utilizator  **WHERE** INITCAP(utilizator**.**nume) = INITCAP(v\_nume);  **IF** v\_nr\_utilizator = **0** **THEN**  **RAISE** exception\_no\_user;  **END** **IF**;  **IF** v\_nr\_utilizator > **1** **then**  **RAISE** exception\_multiple\_users;  **END** **IF**;  **SELECT** COUNT(\*)  **INTO** v\_nr\_joc  **FROM** joc\_video  **WHERE** INITCAP(joc\_video**.**nume) = INITCAP(v\_joc);  **IF** v\_nr\_joc = **0** **THEN**  **RAISE** exception\_no\_game;  **END** **IF**;  **IF** v\_nr\_joc > **1** **then**  **RAISE** exception\_multiple\_games;  **END** **IF**;  **SELECT** COUNT(\*)  **INTO** v\_nr\_recenzie  **FROM** recenzie  **JOIN** utilizator **ON** utilizator**.**cod\_utilizator = recenzie**.**cod\_utilizator  **JOIN** joc\_video **ON** joc\_video**.**cod\_joc = recenzie**.**cod\_joc  **WHERE** INITCAP(utilizator**.**nume) = INITCAP(v\_nume) **AND** INITCAP(joc\_video**.**nume) = INITCAP(v\_joc);  **IF** v\_nr\_recenzie = **0** **THEN**  **RAISE** exception\_no\_review;  **END** **IF**;  **SELECT** scor  **INTO** v\_scor  **FROM** recenzie  **JOIN** utilizator **ON** utilizator**.**cod\_utilizator = recenzie**.**cod\_utilizator  **JOIN** joc\_video **ON** joc\_video**.**cod\_joc = recenzie**.**cod\_joc  **WHERE** INITCAP(utilizator**.**nume) = INITCAP(v\_nume) **AND** INITCAP(joc\_video**.**nume) = INITCAP(v\_joc);  **RETURN** v\_scor;  **EXCEPTION**  **WHEN** exception\_no\_user **THEN**  RAISE\_APPLICATION\_ERROR(-**20001**, 'Nu exista utilizator cu numele ' || v\_nume);  **WHEN** exception\_multiple\_users **THEN**  RAISE\_APPLICATION\_ERROR(-**20002**, 'Exista mai multi utilizatori cu numele ' || v\_nume);  **WHEN** exception\_no\_game **THEN**  RAISE\_APPLICATION\_ERROR(-**20003**, 'Nu exista joc cu numele ' || v\_joc);  **WHEN** exception\_multiple\_games **THEN**  RAISE\_APPLICATION\_ERROR(-**20004**, 'Exista mai multe jocuri cu numele ' || v\_joc);  **WHEN** exception\_no\_review **THEN**  RAISE\_APPLICATION\_ERROR(-**20005**, 'Nu exista recenzie pentru jocul ' || v\_joc || ' de catre utilizatorul ' || v\_nume);  **END**; |

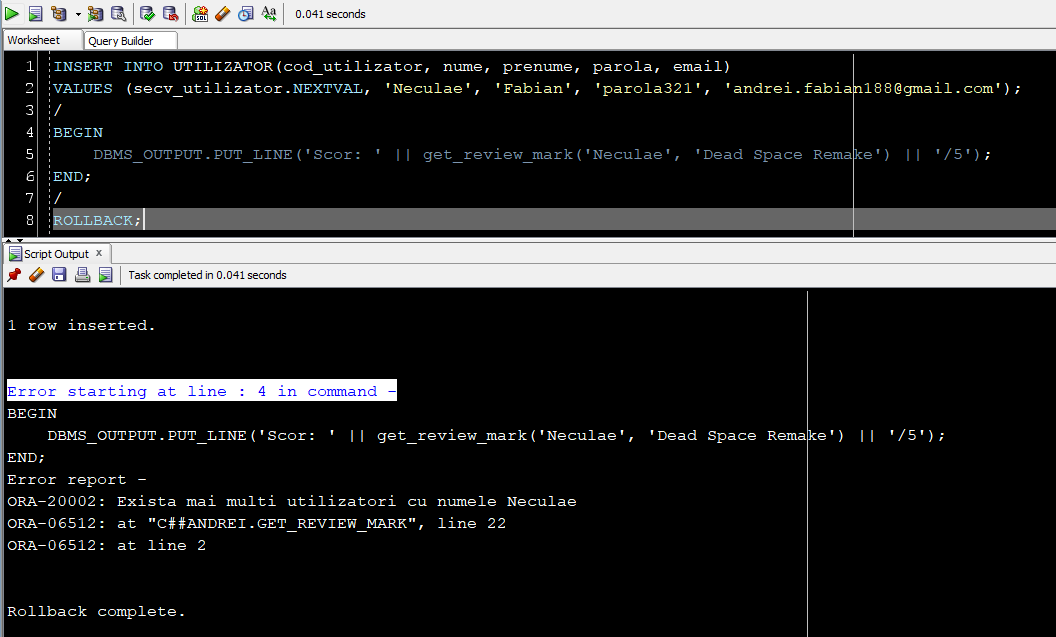


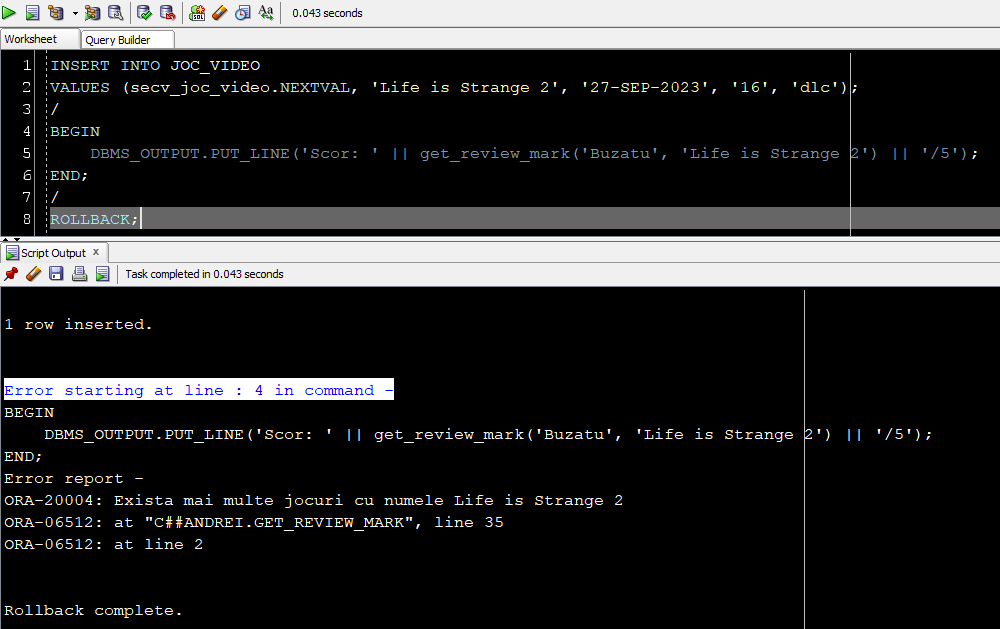










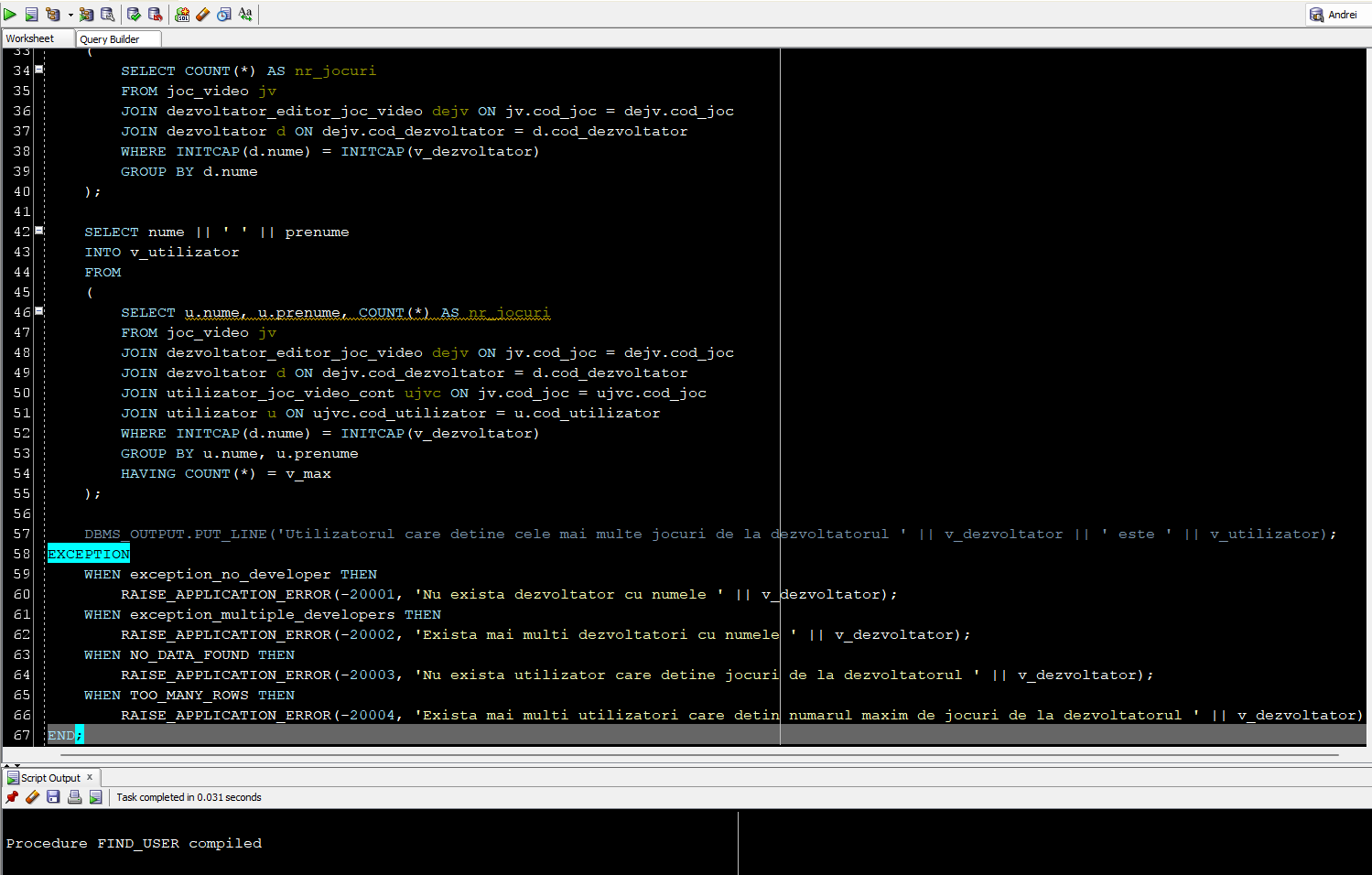


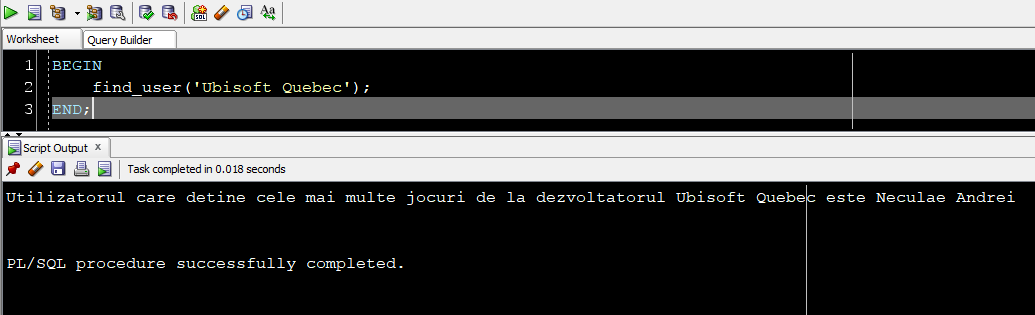
# Exercitiul 9

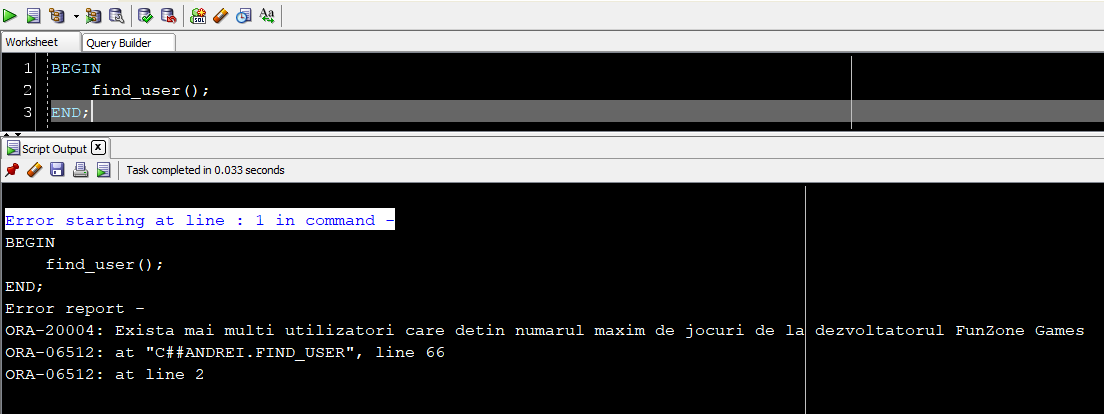
Cerinta: Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent de tip procedura care sa utilizeze intr-o singura comanda SQL 5 dintre tabelele definite. Tratati toate exceptiile care pot aparea, incluzand exceptiile NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelati subprogramul astfel incat sa evidentiati toate cazurile tratate.

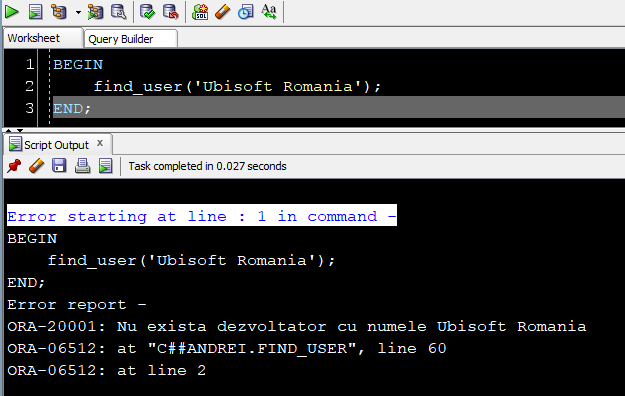
Rezolvare: Definiti un subprogram stocat independent de tip procedura care primeste un nume de dezvoltator ca parametru si afiseaza utilizatorul care detine cele mai multe jocuri de la acesta. Tratati cazurile in care nu exista dezvoltator cu numele dat, in care exista mai multi dezvoltatori cu numele dat, in care nu exista utilizator (nu detine nimeni joc de la dezvoltatorul dat) sau in care exista mai multi utilizatori.

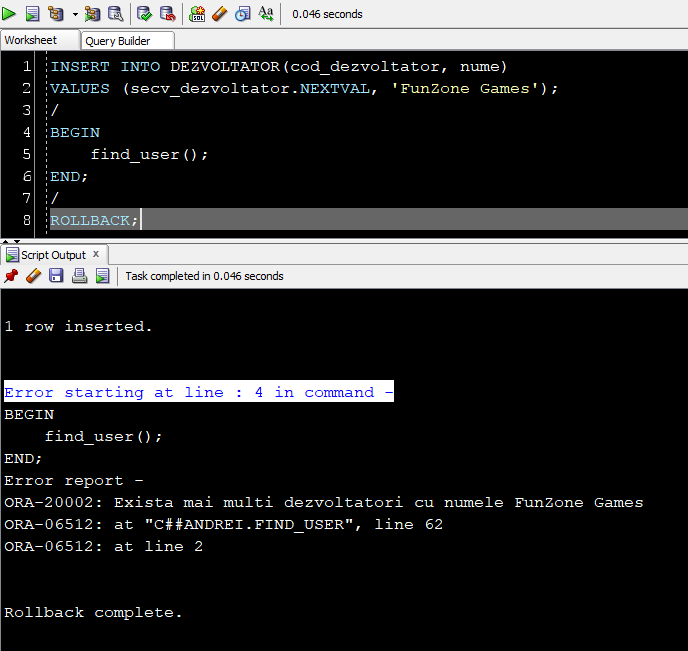
|  |
| --- |
| **CREATE** **OR** **REPLACE** **PROCEDURE** find\_user  (  v\_dezvoltator dezvoltator**.**nume%TYPE **DEFAULT** 'FunZone Games'  )  **IS**  v\_nr\_dezvoltator NUMBER;  v\_max NUMBER;  v\_utilizator VARCHAR2(**128**);  exception\_no\_developer **EXCEPTION**;  PRAGMA EXCEPTION\_INIT(exception\_no\_developer, -**20001**);  exception\_multiple\_developers **EXCEPTION**;  PRAGMA EXCEPTION\_INIT(exception\_multiple\_developers, -**20002**);  **BEGIN**  **SELECT** COUNT(\*)  **INTO** v\_nr\_dezvoltator  **FROM** dezvoltator  **WHERE** INITCAP(dezvoltator**.**nume) = INITCAP(v\_dezvoltator);  **IF** v\_nr\_dezvoltator = **0** **THEN**  **RAISE** exception\_no\_developer;  **END** **IF**;  **IF** v\_nr\_dezvoltator > **1** **THEN**  **RAISE** exception\_multiple\_developers;  **END** **IF**;  **SELECT** MAX(nr\_jocuri)  **INTO** v\_max  **FROM**  (  **SELECT** COUNT(\*) **AS** nr\_jocuri  **FROM** joc\_video jv  **JOIN** dezvoltator\_editor\_joc\_video dejv **ON** jv**.**cod\_joc = dejv**.**cod\_joc  **JOIN** dezvoltator d **ON** dejv**.**cod\_dezvoltator = d**.**cod\_dezvoltator  **WHERE** INITCAP(d**.**nume) = INITCAP(v\_dezvoltator)  **GROUP** **BY** d**.**nume  );  **SELECT** nume || ' ' || prenume  **INTO** v\_utilizator  **FROM**  (  **SELECT** u**.**nume, u**.**prenume, COUNT(\*) **AS** nr\_jocuri  **FROM** joc\_video jv  **JOIN** dezvoltator\_editor\_joc\_video dejv **ON** jv**.**cod\_joc = dejv**.**cod\_joc  **JOIN** dezvoltator d **ON** dejv**.**cod\_dezvoltator = d**.**cod\_dezvoltator  **JOIN** utilizator\_joc\_video\_cont ujvc **ON** jv**.**cod\_joc = ujvc**.**cod\_joc  **JOIN** utilizator u **ON** ujvc**.**cod\_utilizator = u**.**cod\_utilizator  **WHERE** INITCAP(d**.**nume) = INITCAP(v\_dezvoltator)  **GROUP** **BY** u**.**nume, u**.**prenume  **HAVING** COUNT(\*) = v\_max  );  DBMS\_OUTPUT**.**PUT\_LINE('Utilizatorul care detine cele mai multe jocuri de la dezvoltatorul ' || v\_dezvoltator || ' este ' || v\_utilizator);  **EXCEPTION**  **WHEN** exception\_no\_developer **THEN**  RAISE\_APPLICATION\_ERROR(-**20001**, 'Nu exista dezvoltator cu numele ' || v\_dezvoltator);  **WHEN** exception\_multiple\_developers **THEN**  RAISE\_APPLICATION\_ERROR(-**20002**, 'Exista mai multi dezvoltatori cu numele ' || v\_dezvoltator);  **WHEN** NO\_DATA\_FOUND **THEN**  RAISE\_APPLICATION\_ERROR(-**20003**, 'Nu exista utilizator care detine jocuri de la dezvoltatorul ' || v\_dezvoltator);  **WHEN** TOO\_MANY\_ROWS **THEN**  RAISE\_APPLICATION\_ERROR(-**20004**, 'Exista mai multi utilizatori care detin numarul maxim de jocuri de la dezvoltatorul ' || v\_dezvoltator);  **END**; |

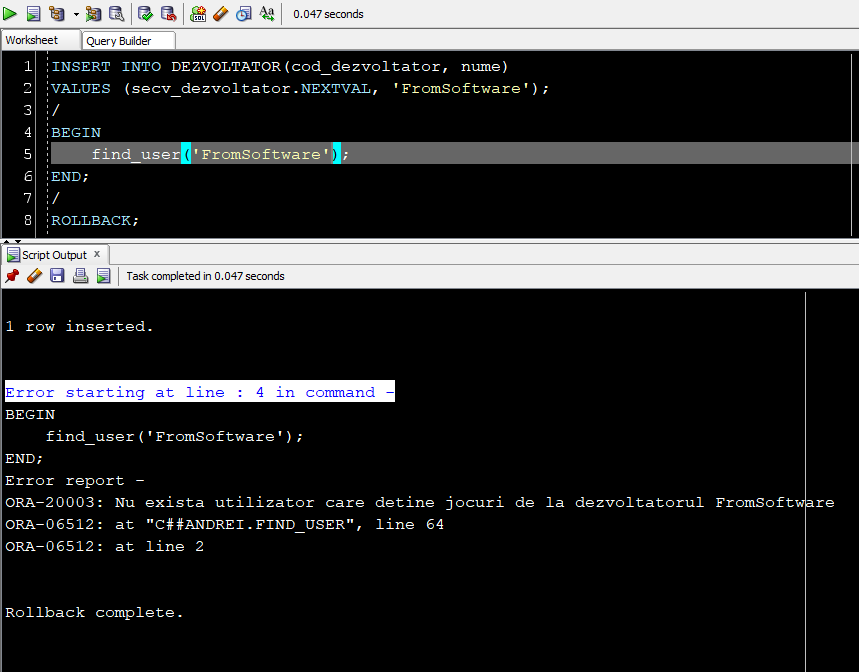










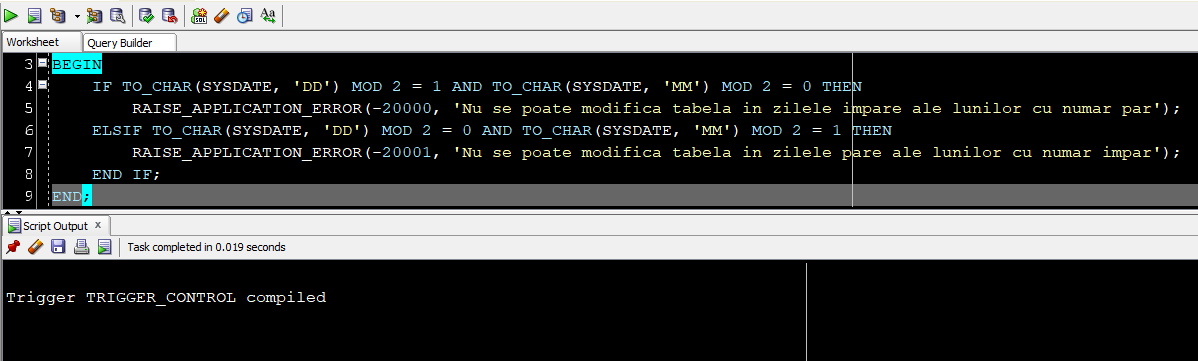


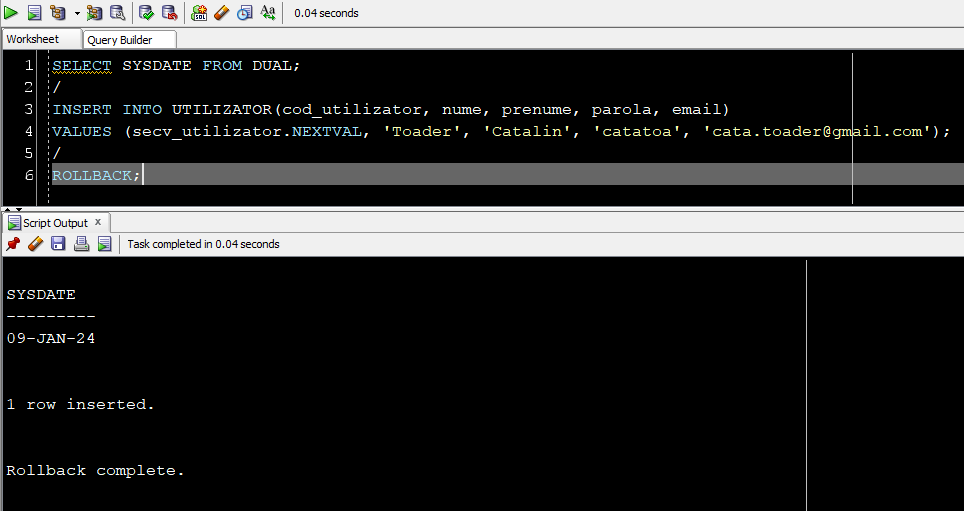
# Exercitiul 10

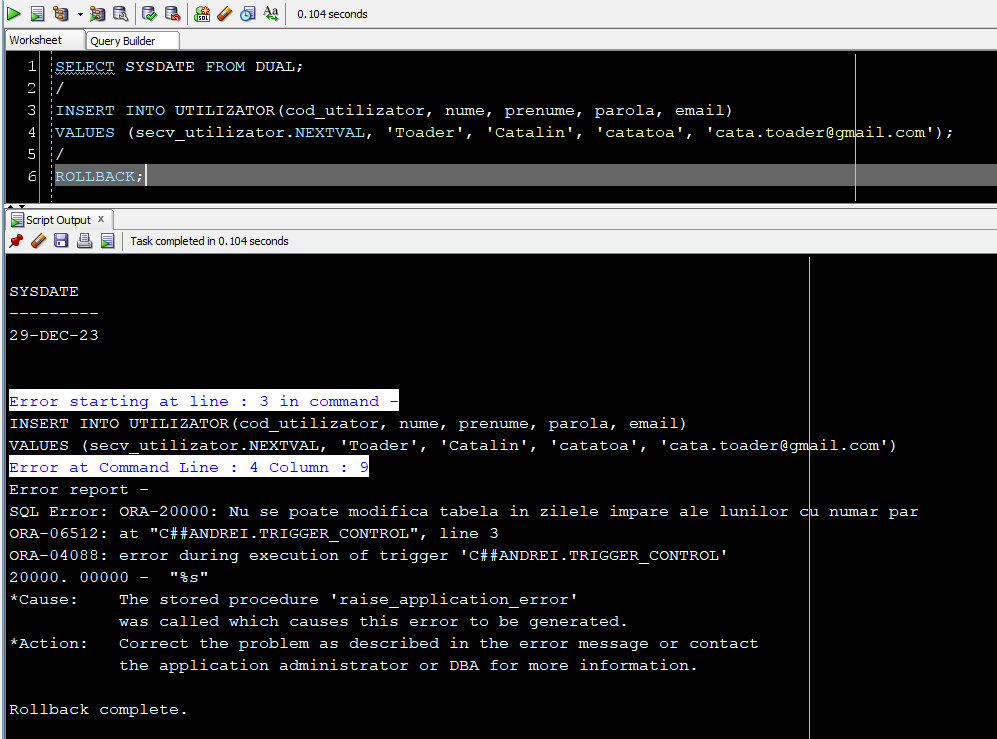
Cerinta: Definiti un trigger de tip LMD la nivel de comanda. Declansati trigger-ul.

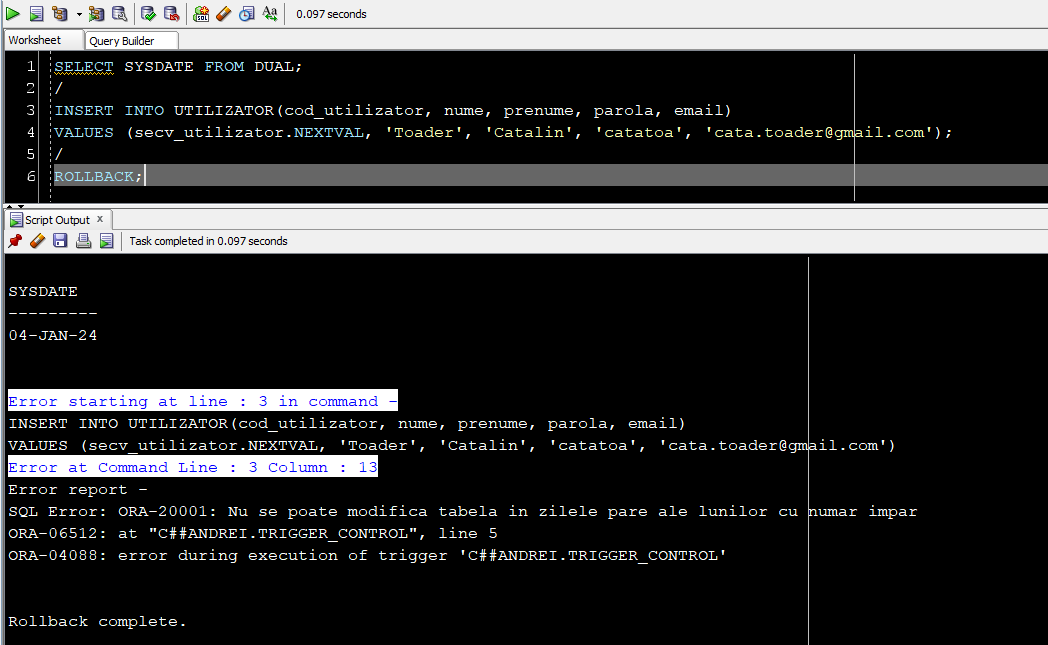
Rezolvare: Definiti un trigger care blocheaza lucrul asupra tabelei utilizator atat in zilele impare ale lunilor pare, cat si in zilele pare ale lunilor impare.

|  |
| --- |
| **CREATE** **OR** **REPLACE** **TRIGGER** trigger\_control  **BEFORE** **INSERT** **OR** **UPDATE** **OR** **DELETE** **ON** UTILIZATOR  **BEGIN**  **IF** TO\_CHAR(SYSDATE, 'DD') MOD **2** = **1** **AND** TO\_CHAR(SYSDATE, 'MM') MOD **2** = **0** **THEN**  RAISE\_APPLICATION\_ERROR(-**20000**, 'Nu se poate modifica tabela in zilele impare ale lunilor pare');  **ELSIF** TO\_CHAR(SYSDATE, 'DD') MOD **2** = **0** **AND** TO\_CHAR(SYSDATE, 'MM') MOD **2** = **1** **THEN**  RAISE\_APPLICATION\_ERROR(-**20001**, 'Nu se poate modifica tabela in zilele pare ale lunilor impare');  **END** **IF**;  **END**; |







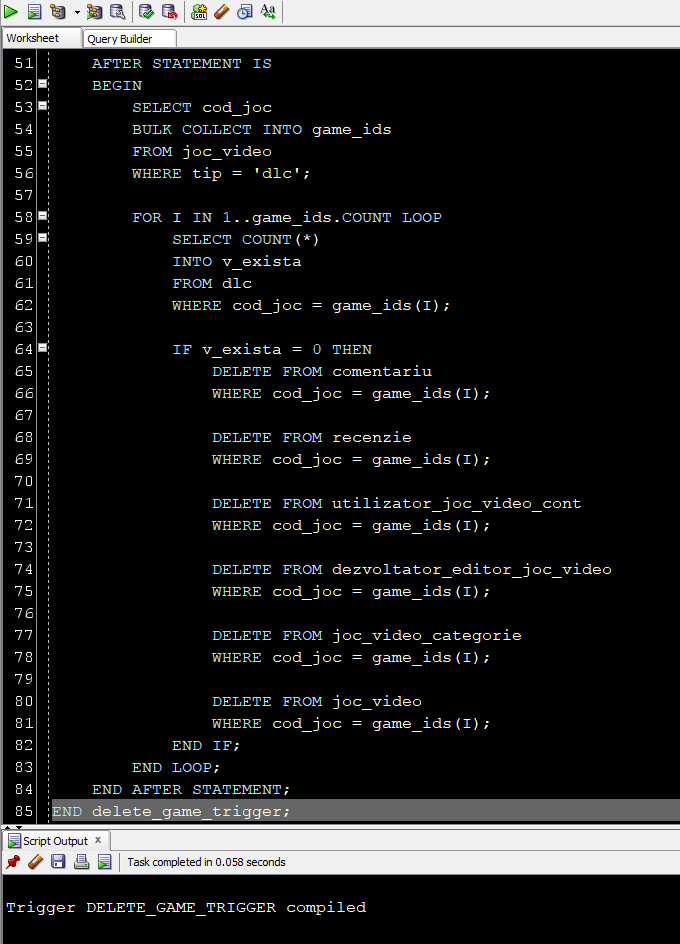


# Exercitiul 11

Cerinta: Definiti un trigger de tip LMD la nivel de linie. Declansati trigger-ul.

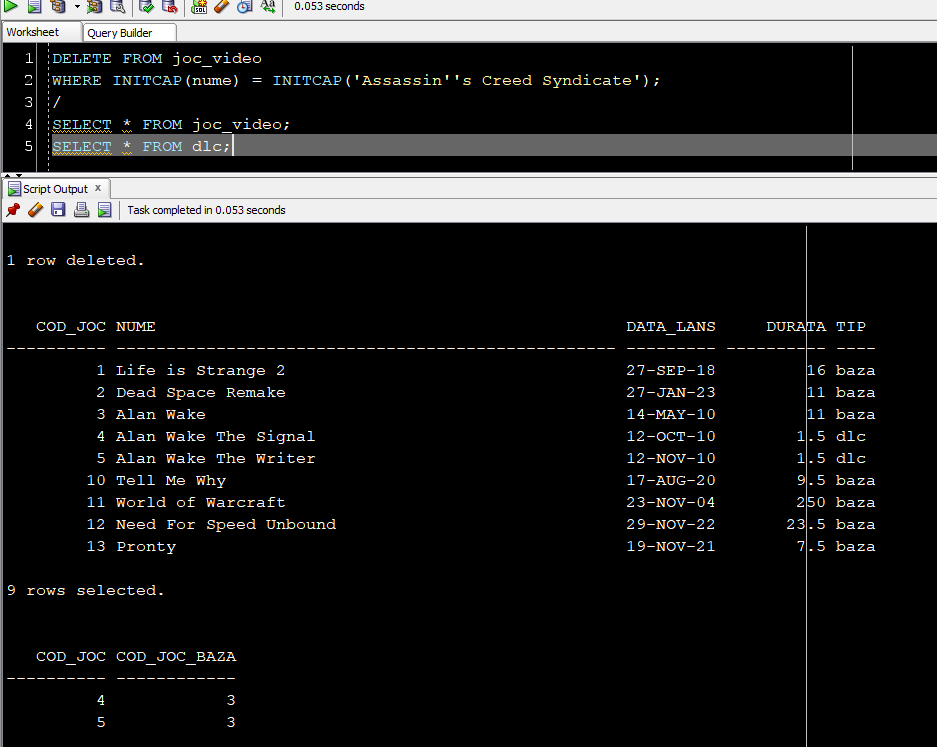
Rezolvare: Definiti un trigger care, la stergerea unui joc din tabela joc\_video, sterge atat toate dlc-urile acestuia, cat si toate instantele in care apar acestea (i.e. recenzii, comentarii si tabelele many-to-many)

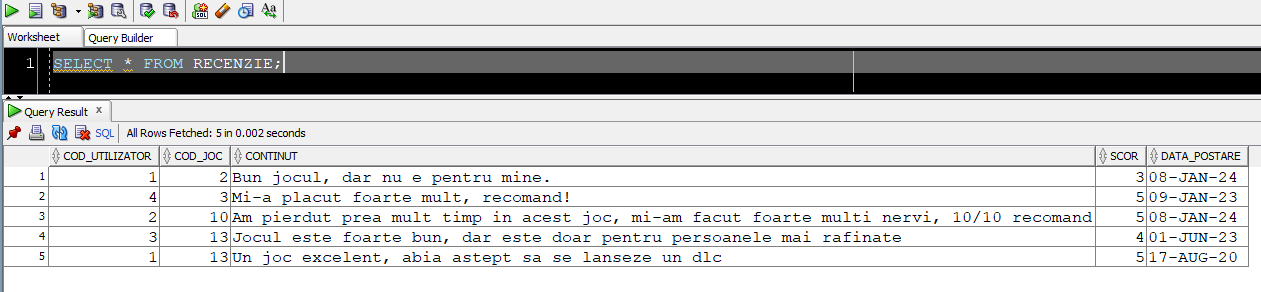
|  |
| --- |
| **CREATE** **OR** **REPLACE** **TRIGGER** delete\_game\_trigger  **FOR** **DELETE** **ON** joc\_video  COMPOUND **TRIGGER**  **TYPE** tablou\_imbricat **IS** **TABLE** **OF** joc\_video**.**cod\_joc%TYPE;  game\_ids tablou\_imbricat := tablou\_imbricat();  v\_exista NUMBER;  **BEFORE** **EACH** **ROW** **IS**  **BEGIN**  **DELETE** **FROM** comentariu  **WHERE** cod\_joc = :OLD**.**cod\_joc;    **DELETE** **FROM** recenzie  **WHERE** cod\_joc = :OLD**.**cod\_joc;  **DELETE** **FROM** utilizator\_joc\_video\_cont  **WHERE** cod\_joc = :OLD**.**cod\_joc;  **DELETE** **FROM** dezvoltator\_editor\_joc\_video  **WHERE** cod\_joc = :OLD**.**cod\_joc;  **DELETE** **FROM** joc\_video\_categorie  **WHERE** cod\_joc = :OLD**.**cod\_joc;  **SELECT** cod\_joc  BULK COLLECT **INTO** game\_ids  **FROM** dlc  **WHERE** cod\_joc\_baza = :OLD**.**cod\_joc;  **FOR** I **IN** **1..**game\_ids**.**COUNT **LOOP**  **DELETE** **FROM** comentariu  **WHERE** cod\_joc = game\_ids(I);    **DELETE** **FROM** recenzie  **WHERE** cod\_joc = game\_ids(I);  **DELETE** **FROM** utilizator\_joc\_video\_cont  **WHERE** cod\_joc = game\_ids(I);  **DELETE** **FROM** dezvoltator\_editor\_joc\_video  **WHERE** cod\_joc = game\_ids(I);  **DELETE** **FROM** joc\_video\_categorie  **WHERE** cod\_joc = game\_ids(I);  **DELETE** **FROM** dlc  **WHERE** cod\_joc = game\_ids(I);  **END** **LOOP**;  **END** **BEFORE** **EACH** **ROW**;  **AFTER** **STATEMENT** **IS**  **BEGIN**  **SELECT** cod\_joc  BULK COLLECT **INTO** game\_ids  **FROM** joc\_video  **WHERE** tip = 'dlc';  **FOR** I **IN** **1..**game\_ids**.**COUNT **LOOP**  **SELECT** COUNT(\*)  **INTO** v\_exista  **FROM** dlc  **WHERE** cod\_joc = game\_ids(I);  **IF** v\_exista = **0** **THEN**  **DELETE** **FROM** comentariu  **WHERE** cod\_joc = game\_ids(I);    **DELETE** **FROM** recenzie  **WHERE** cod\_joc = game\_ids(I);  **DELETE** **FROM** utilizator\_joc\_video\_cont  **WHERE** cod\_joc = game\_ids(I);  **DELETE** **FROM** dezvoltator\_editor\_joc\_video  **WHERE** cod\_joc = game\_ids(I);  **DELETE** **FROM** joc\_video\_categorie  **WHERE** cod\_joc = game\_ids(I);  **DELETE** **FROM** joc\_video  **WHERE** cod\_joc = game\_ids(I);  **END** **IF**;  **END** **LOOP**;  **END** **AFTER** **STATEMENT**;  **END** delete\_game\_trigger; |

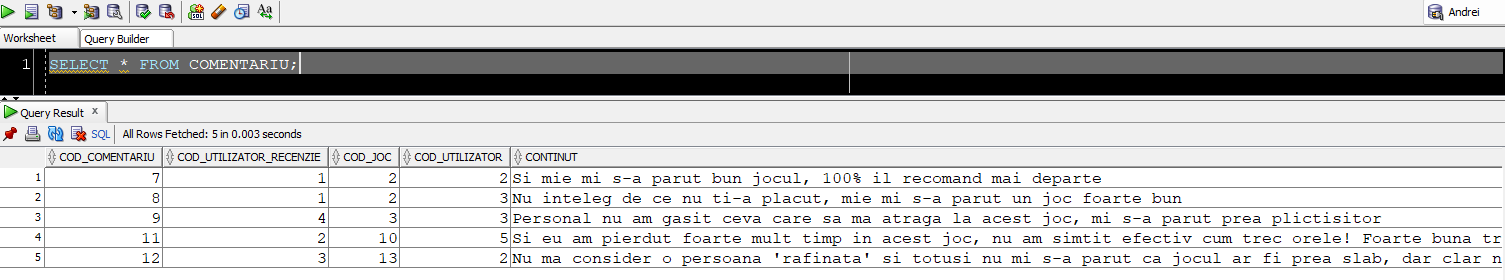


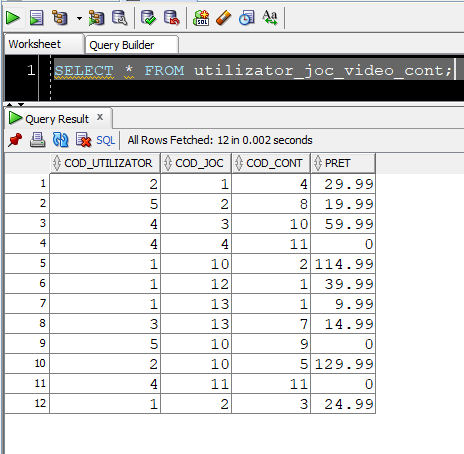
[Datele initiale, inainte de delete](#_Adaugarea_de_informatii)

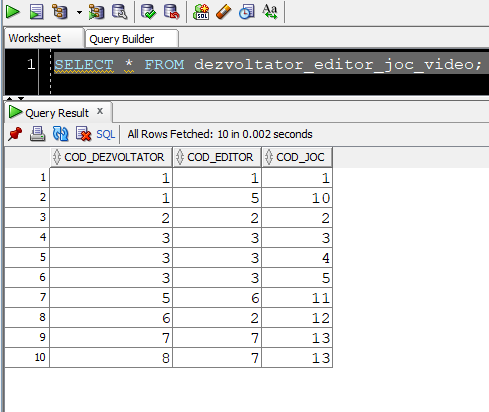
Datele dupa delete

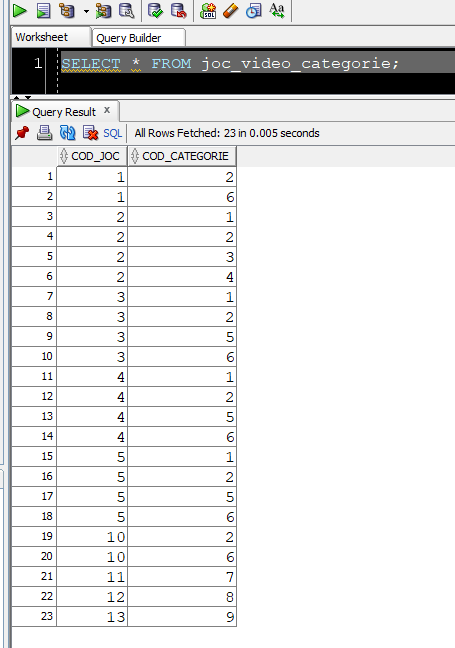










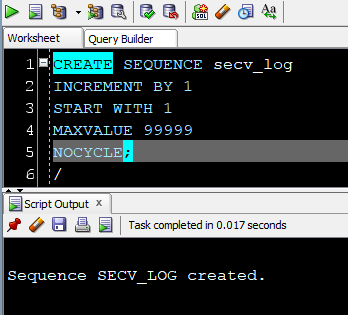


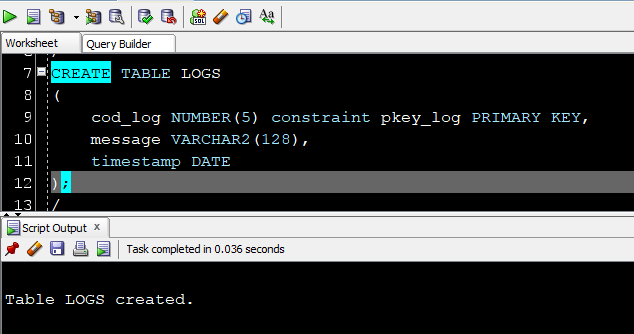
# Exercitiul 12

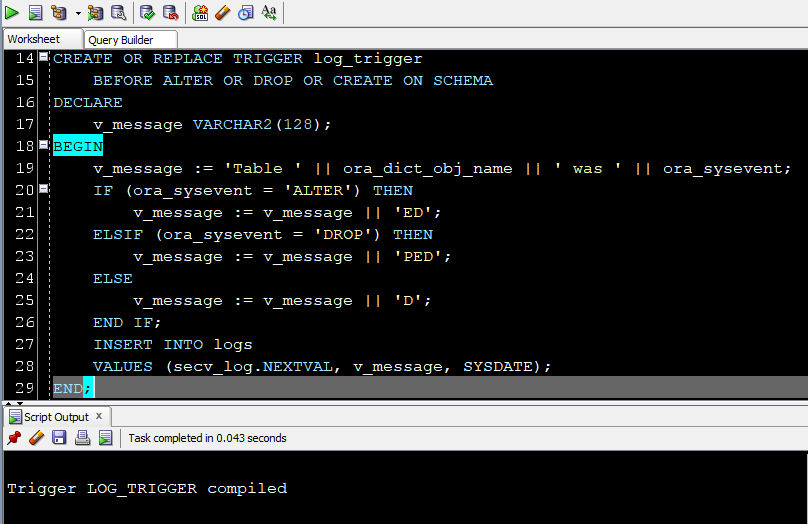
Cerinta: Definiti un trigger de tip LDD. Declansati trigger-ul.

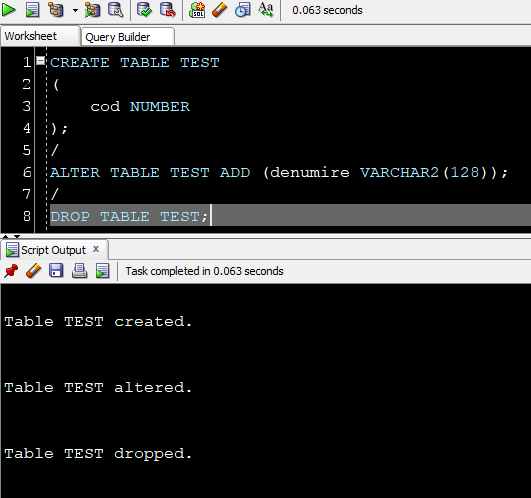
Rezolvare: Definiti un trigger care adauga intr-o tabela de log-uri toate actiunile de alter, create si drop care au loc, tabelul manipulat si timestamp-ul.

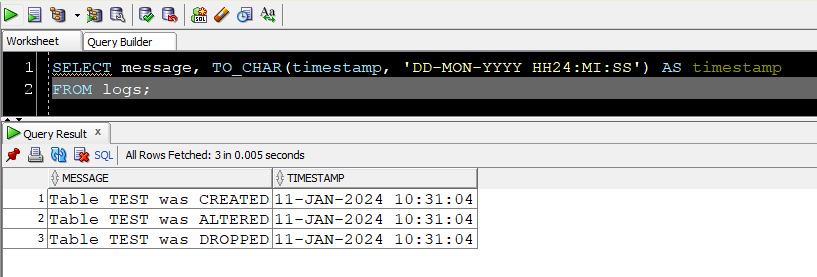
|  |
| --- |
| **CREATE** **SEQUENCE** secv\_log  **INCREMENT** **BY** **1**  **START** **WITH** **1**  **MAXVALUE** **99999**  NOCYCLE;  /  **CREATE** **TABLE** LOGS  (  cod\_log NUMBER(**5**) **constraint** pkey\_log **PRIMARY** **KEY**,  message VARCHAR2(**128**),  timestamp DATE  );  /  **CREATE** **OR** **REPLACE** **TRIGGER** log\_trigger  **BEFORE** **ALTER** **OR** **DROP** **OR** **CREATE** **ON** **SCHEMA**  **DECLARE**  v\_message VARCHAR2(**128**);  **BEGIN**  v\_message := 'Table ' || ora\_dict\_obj\_name || ' was ' || ora\_sysevent;  **IF** (ora\_sysevent = 'ALTER') **THEN**  v\_message := v\_message || 'ED';  **ELSIF** (ora\_sysevent = 'DROP') **THEN**  v\_message := v\_message || 'PED';  **ELSE**  v\_message := v\_message || 'D';  **END** **IF**;  **INSERT** **INTO** logs  **VALUES** (secv\_log**.**NEXTVAL, v\_message, SYSDATE);  **END**; |









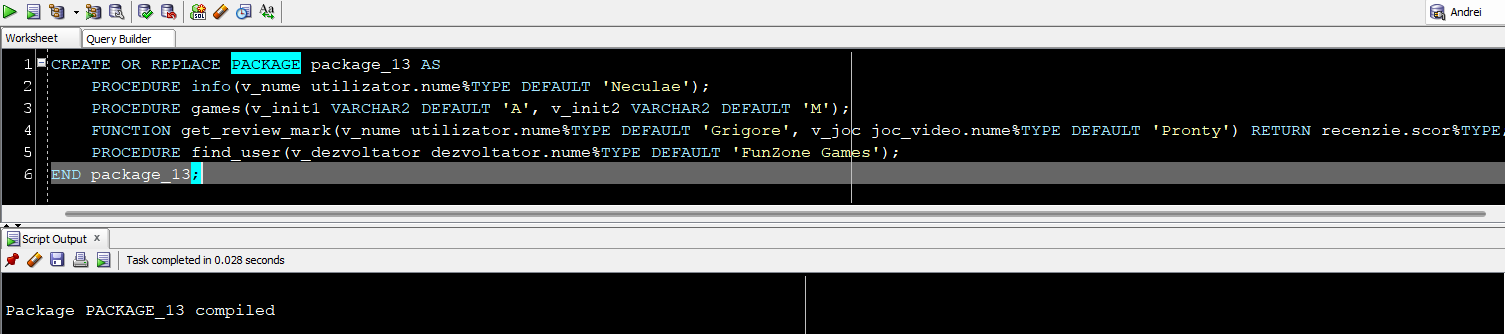


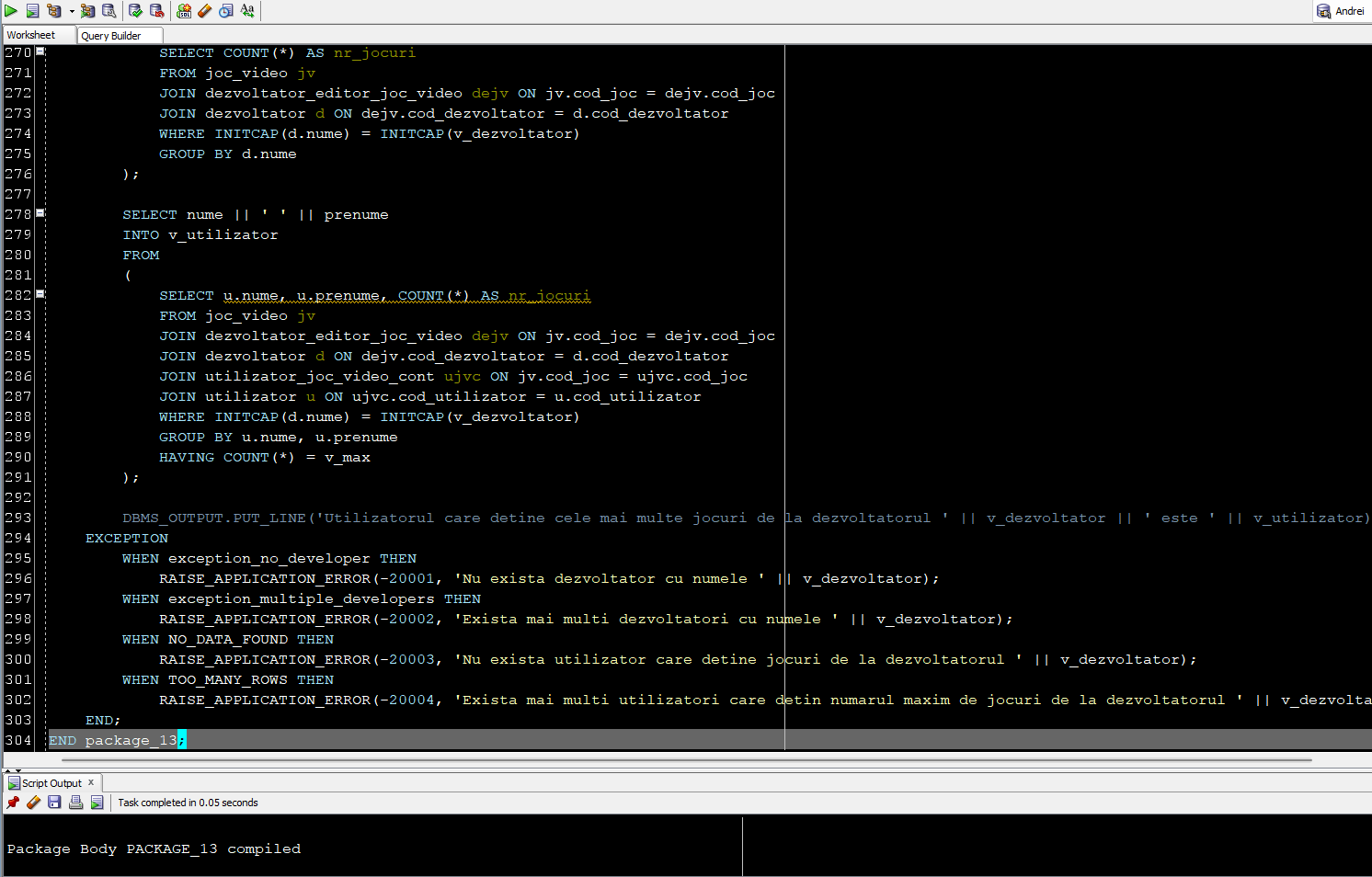
# Exercitiul 13

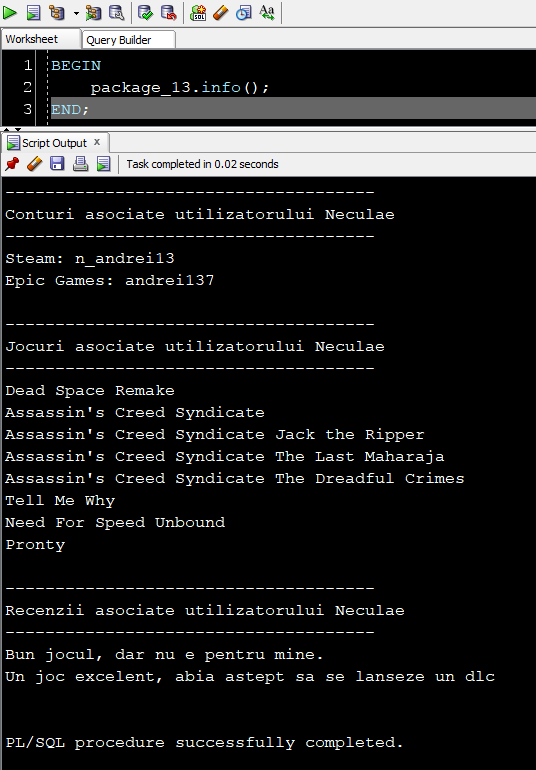
Cerinta: Definiti un pachet care sa contina toate obiectele definite in cadrul proiectului.

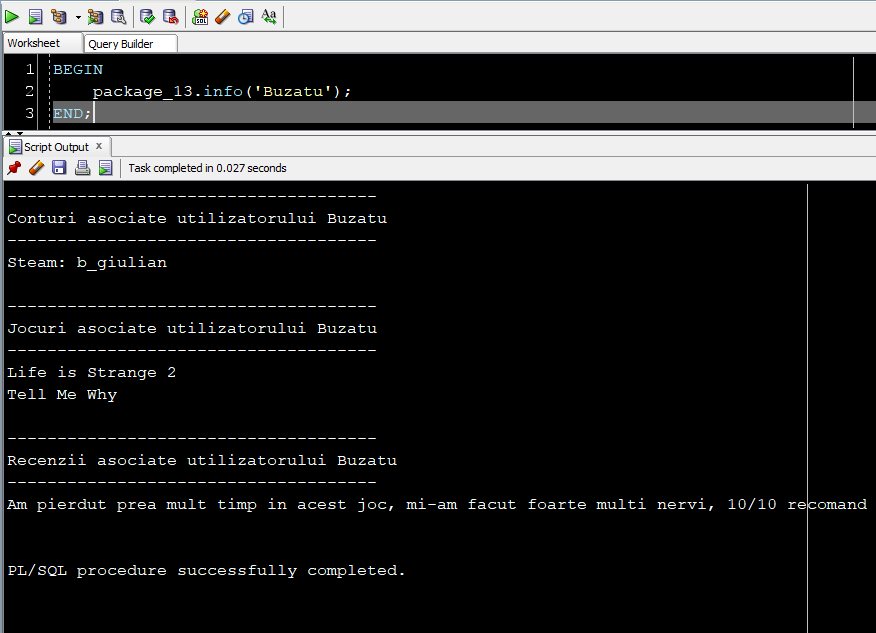
Rezolvare:

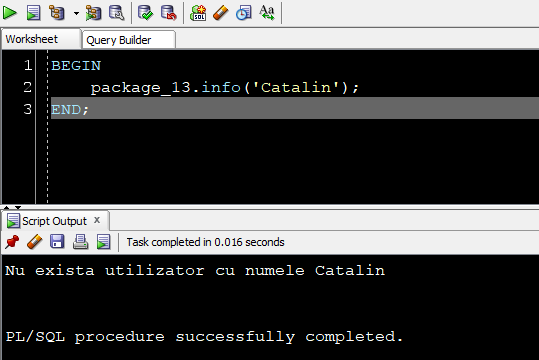
|  |
| --- |
| **CREATE** **OR** **REPLACE** PACKAGE package\_13 **AS**  **PROCEDURE** info(v\_nume utilizator**.**nume%TYPE **DEFAULT** 'Neculae');  **PROCEDURE** games(v\_init1 VARCHAR2 **DEFAULT** 'A', v\_init2 VARCHAR2 **DEFAULT** 'M');  **FUNCTION** get\_review\_mark(v\_nume utilizator**.**nume%TYPE **DEFAULT** 'Grigore', v\_joc joc\_video**.**nume%TYPE **DEFAULT** 'Pronty') **RETURN** recenzie**.**scor%TYPE;  **PROCEDURE** find\_user(v\_dezvoltator dezvoltator**.**nume%TYPE **DEFAULT** 'FunZone Games');  **END** package\_13;  /  **CREATE** **OR** **REPLACE** PACKAGE BODY package\_13 **AS**  **PROCEDURE** info  (  v\_nume utilizator**.**nume%TYPE **DEFAULT** 'Neculae'  )  **IS**  **TYPE** pair **IS** RECORD  (  platform platforma**.**nume%TYPE,  username cont**.**nume%TYPE  );  **TYPE** vector **IS** VARRAY(**3**) **OF** pair;  **TYPE** tablou\_indexat **IS** **TABLE** **OF** joc\_video**.**nume%TYPE **INDEX** **BY** PLS\_INTEGER;  **TYPE** tablou\_imbricat **IS** **TABLE** **OF** recenzie**.**continut%TYPE;  v\_accounts vector := vector();  v\_games tablou\_indexat;  v\_reviews tablou\_imbricat := tablou\_imbricat();  v\_exista NUMBER;  **BEGIN**  **SELECT** COUNT(\*)  **INTO** v\_exista  **FROM** utilizator  **WHERE** INITCAP(utilizator**.**nume) = INITCAP(v\_nume);  **IF** v\_exista = **0** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Nu exista utilizator cu numele ' || v\_nume);  **RETURN**;  **END** **IF**;    **IF** v\_exista > **1** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Exista mai multi utilizatori cu numele ' || v\_nume);  **RETURN**;  **END** **IF**;  **SELECT** **DISTINCT** p**.**nume, c**.**nume  BULK COLLECT **INTO** v\_accounts  **FROM** cont c  **JOIN** utilizator\_joc\_video\_cont ujvc **ON** c**.**cod\_cont = ujvc**.**cod\_cont  **JOIN** platforma p **ON** c**.**cod\_platforma = p**.**cod\_platforma  **JOIN** utilizator u **ON** ujvc**.**cod\_utilizator = u**.**cod\_utilizator  **WHERE** INITCAP(u**.**nume) = INITCAP(v\_nume) **AND** INITCAP(p**.**nume) **IN** ('Steam', 'Epic Games', 'Xbox');  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **IF** v\_accounts**.**COUNT = **0** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Nu exista conturi asociate utilizatorului ' || v\_nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **ELSE**  DBMS\_OUTPUT**.**PUT\_LINE('Conturi asociate utilizatorului ' || v\_nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **FOR** i **IN** **1..**v\_accounts**.**COUNT **LOOP**  DBMS\_OUTPUT**.**PUT\_LINE(v\_accounts(i)**.**platform || ': ' || v\_accounts(i)**.**username);  **END** **LOOP**;  **END** **IF**;  **SELECT** jv**.**nume  BULK COLLECT **INTO** v\_games  **FROM** joc\_video jv  **JOIN** utilizator\_joc\_video\_cont ujvc **ON** jv**.**cod\_joc = ujvc**.**cod\_joc  **JOIN** utilizator u **ON** ujvc**.**cod\_utilizator = u**.**cod\_utilizator  **WHERE** INITCAP(u**.**nume) = INITCAP(v\_nume);  DBMS\_OUTPUT**.**NEW\_LINE;  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **IF** v\_games**.**COUNT = **0** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Nu exista jocuri asociate utilizatorului ' || v\_nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **ELSE**  DBMS\_OUTPUT**.**PUT\_LINE('Jocuri asociate utilizatorului ' || v\_nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **FOR** i **IN** **1..**v\_games**.**COUNT **LOOP**  DBMS\_OUTPUT**.**PUT\_LINE(v\_games(i));  **END** **LOOP**;  **END** **IF**;    **SELECT** r**.**continut  BULK COLLECT **INTO** v\_reviews  **FROM** recenzie r  **JOIN** utilizator u **ON** r**.**cod\_utilizator = u**.**cod\_utilizator  **WHERE** INITCAP(u**.**nume) = INITCAP(v\_nume);  DBMS\_OUTPUT**.**NEW\_LINE;  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **IF** v\_reviews**.**COUNT = **0** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Nu exista recenzii asociate utilizatorului ' || v\_nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **ELSE**  DBMS\_OUTPUT**.**PUT\_LINE('Recenzii asociate utilizatorului ' || v\_nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **FOR** i **IN** **1..**v\_reviews**.**COUNT **LOOP**  DBMS\_OUTPUT**.**PUT\_LINE(v\_reviews(i));  **END** **LOOP**;  **END** **IF**;  **END**;  **PROCEDURE** games  (  v\_init1 VARCHAR2 **DEFAULT** 'A',  v\_init2 VARCHAR2 **DEFAULT** 'M'  )  **IS**  **TYPE** tip\_cursor **IS** **REF** **CURSOR** **RETURN** categorie%ROWTYPE;  categ tip\_cursor;  **CURSOR** jocuri(v\_categ categorie**.**cod\_categorie%TYPE) **IS**  **SELECT** jv**.**nume, data\_lansare  **FROM** joc\_video jv  **JOIN** joc\_video\_categorie jvc **ON** jv**.**cod\_joc = jvc**.**cod\_joc  **JOIN** categorie c **ON** jvc**.**cod\_categorie = c**.**cod\_categorie  **WHERE** c**.**cod\_categorie = v\_categ  **ORDER** **BY** data\_lansare;  c categorie%ROWTYPE;  j\_nume joc\_video**.**nume%TYPE;  j\_data joc\_video**.**data\_lansare%TYPE;  **BEGIN**  **OPEN** categ **FOR**  **SELECT** \*  **FROM** categorie  **WHERE** INITCAP(nume) **LIKE** INITCAP(v\_init1) || '%' **OR** INITCAP(nume) **LIKE** INITCAP(v\_init2) || '%';  **LOOP**  **FETCH** categ **INTO** c;  **EXIT** **WHEN** categ%NOTFOUND;  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  DBMS\_OUTPUT**.**PUT\_LINE('Categorie: ' || c**.**nume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **OPEN** jocuri(c**.**cod\_categorie);  **LOOP**  **FETCH** jocuri **INTO** j\_nume, j\_data;  **EXIT** **WHEN** jocuri%NOTFOUND;  DBMS\_OUTPUT**.**PUT\_LINE(j\_nume || ' lansat pe data de ' || j\_data);  **END** **LOOP**;  **IF** jocuri%ROWCOUNT = **0** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Nu exista jocuri in aceasta categorie');  **END** **IF**;  **CLOSE** jocuri;  DBMS\_OUTPUT**.**NEW\_LINE;  **END** **LOOP**;  **CLOSE** categ;  **END**;  **FUNCTION** get\_review\_mark  (  v\_nume utilizator**.**nume%TYPE **DEFAULT** 'Grigore',  v\_joc joc\_video**.**nume%TYPE **DEFAULT** 'Pronty'  ) **RETURN** recenzie**.**scor%TYPE  **IS**  v\_scor NUMBER;  v\_exista\_utilizator NUMBER;  v\_exista\_joc NUMBER;  v\_exista\_recenzie NUMBER;  **BEGIN**  **SELECT** COUNT(\*)  **INTO** v\_exista\_utilizator  **FROM** utilizator  **WHERE** INITCAP(utilizator**.**nume) = INITCAP(v\_nume);  **IF** v\_exista\_utilizator = **0** **THEN**  RAISE\_APPLICATION\_ERROR(-**20001**, 'Nu exista utilizator cu numele ' || v\_nume);  **END** **IF**;  **IF** v\_exista\_utilizator > **1** **then**  RAISE\_APPLICATION\_ERROR(-**20002**, 'Exista mai multi utilizatori cu numele ' || v\_nume);  **END** **IF**;  **SELECT** COUNT(\*)  **INTO** v\_exista\_joc  **FROM** joc\_video  **WHERE** INITCAP(joc\_video**.**nume) = INITCAP(v\_joc);  **IF** v\_exista\_joc = **0** **THEN**  RAISE\_APPLICATION\_ERROR(-**20003**, 'Nu exista joc cu numele ' || v\_joc);  **END** **IF**;  **IF** v\_exista\_joc > **1** **then**  RAISE\_APPLICATION\_ERROR(-**20004**, 'Exista mai multe jocuri cu numele ' || v\_joc);  **END** **IF**;  **SELECT** COUNT(\*)  **INTO** v\_exista\_recenzie  **FROM** recenzie  **JOIN** utilizator **ON** utilizator**.**cod\_utilizator = recenzie**.**cod\_utilizator  **JOIN** joc\_video **ON** joc\_video**.**cod\_joc = recenzie**.**cod\_joc  **WHERE** INITCAP(utilizator**.**nume) = INITCAP(v\_nume) **AND** INITCAP(joc\_video**.**nume) = INITCAP(v\_joc);  **IF** v\_exista\_recenzie = **0** **THEN**  RAISE\_APPLICATION\_ERROR(-**20005**, 'Nu exista recenzie pentru jocul ' || v\_joc || ' de catre utilizatorul ' || v\_nume);  **END** **IF**;  **SELECT** scor  **INTO** v\_scor  **FROM** recenzie  **JOIN** utilizator **ON** utilizator**.**cod\_utilizator = recenzie**.**cod\_utilizator  **JOIN** joc\_video **ON** joc\_video**.**cod\_joc = recenzie**.**cod\_joc  **WHERE** INITCAP(utilizator**.**nume) = INITCAP(v\_nume) **AND** INITCAP(joc\_video**.**nume) = INITCAP(v\_joc);  **RETURN** v\_scor;  **END**;  **PROCEDURE** find\_user  (  v\_dezvoltator dezvoltator**.**nume%TYPE **DEFAULT** 'FunZone Games'  )  **IS**  v\_exista NUMBER;  v\_max NUMBER;  v\_utilizator VARCHAR2(**128**);  **BEGIN**  **SELECT** COUNT(\*)  **INTO** v\_exista  **FROM** dezvoltator  **WHERE** INITCAP(dezvoltator**.**nume) = INITCAP(v\_dezvoltator);  **IF** v\_exista = **0** **THEN**  RAISE\_APPLICATION\_ERROR(-**20001**, 'Nu exista dezvoltator cu numele ' || v\_dezvoltator);  **END** **IF**;  **IF** v\_exista > **1** **THEN**  RAISE\_APPLICATION\_ERROR(-**20002**, 'Exista mai multi dezvoltator cu numele ' || v\_dezvoltator);  **END** **IF**;  **SELECT** MAX(nr\_jocuri)  **INTO** v\_max  **FROM**  (  **SELECT** COUNT(\*) **AS** nr\_jocuri  **FROM** joc\_video jv  **JOIN** dezvoltator\_editor\_joc\_video dejv **ON** jv**.**cod\_joc = dejv**.**cod\_joc  **JOIN** dezvoltator d **ON** dejv**.**cod\_dezvoltator = d**.**cod\_dezvoltator  **WHERE** INITCAP(d**.**nume) = INITCAP(v\_dezvoltator)  **GROUP** **BY** d**.**nume  );  **SELECT** nume || ' ' || prenume  **INTO** v\_utilizator  **FROM**  (  **SELECT** u**.**nume, u**.**prenume, COUNT(\*) **AS** nr\_jocuri  **FROM** joc\_video jv  **JOIN** dezvoltator\_editor\_joc\_video dejv **ON** jv**.**cod\_joc = dejv**.**cod\_joc  **JOIN** dezvoltator d **ON** dejv**.**cod\_dezvoltator = d**.**cod\_dezvoltator  **JOIN** utilizator\_joc\_video\_cont ujvc **ON** jv**.**cod\_joc = ujvc**.**cod\_joc  **JOIN** utilizator u **ON** ujvc**.**cod\_utilizator = u**.**cod\_utilizator  **WHERE** INITCAP(d**.**nume) = INITCAP(v\_dezvoltator)  **GROUP** **BY** u**.**nume, u**.**prenume  **HAVING** COUNT(\*) = v\_max  );  DBMS\_OUTPUT**.**PUT\_LINE('Utilizatorul care detine cele mai multe jocuri de la dezvoltatorul ' || v\_dezvoltator || ' este ' || v\_utilizator);  **EXCEPTION**  **WHEN** NO\_DATA\_FOUND **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Nu exista utilizator care detine jocuri de la dezvoltatorul ' || v\_dezvoltator);  **WHEN** TOO\_MANY\_ROWS **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Exista mai multi utilizatori care detin numarul maxim de jocuri de la dezvoltatorul ' || v\_dezvoltator);  **END**;  **END** package\_13; |

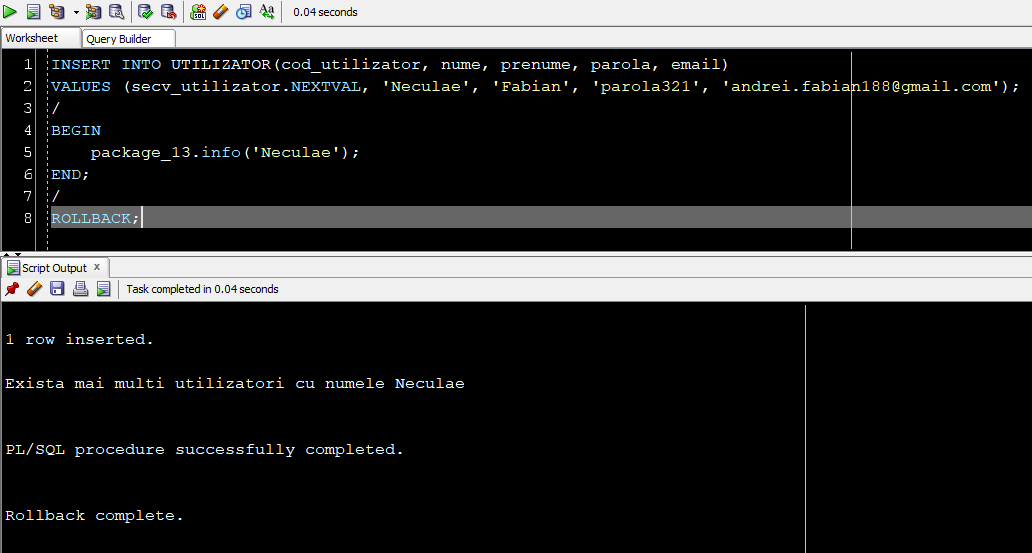


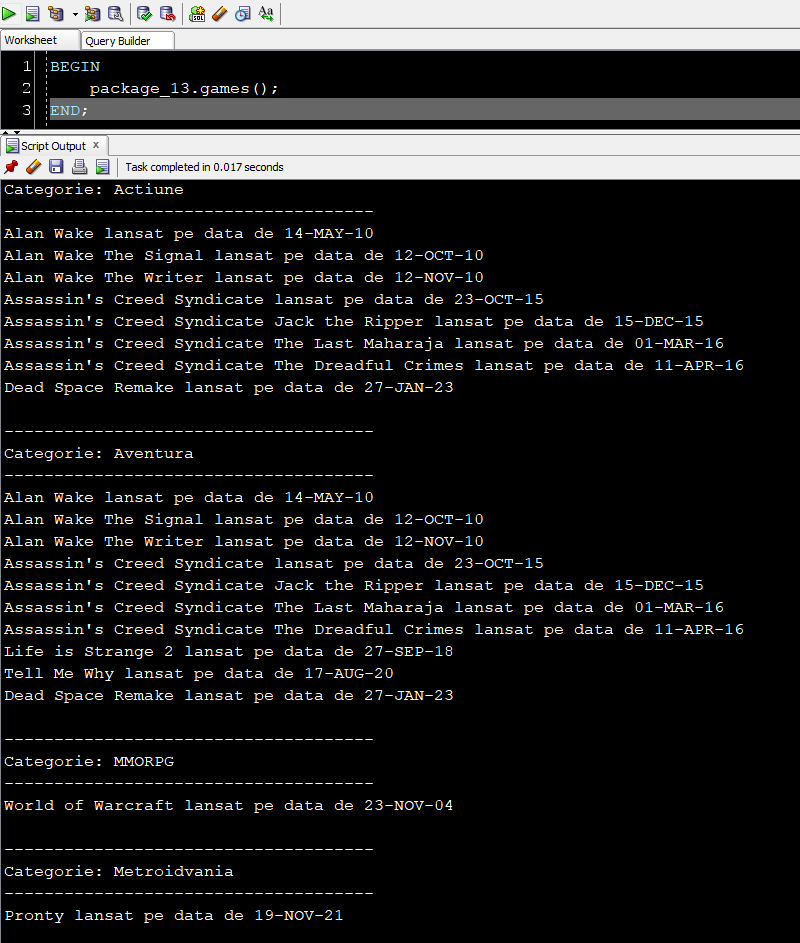


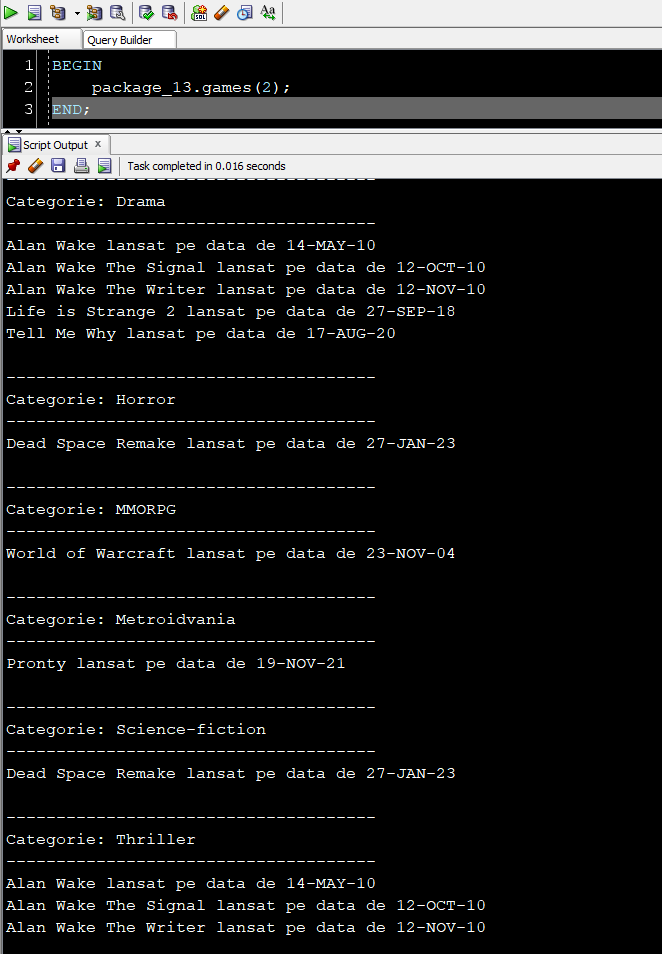


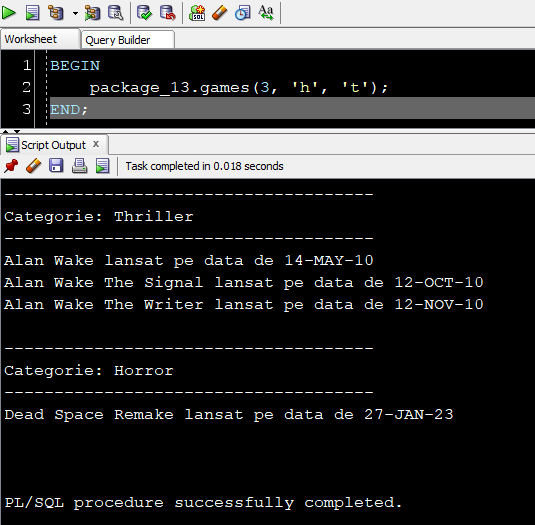


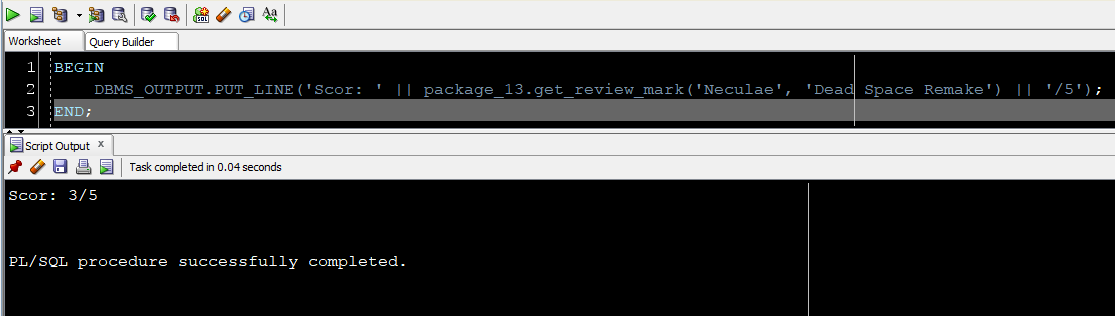


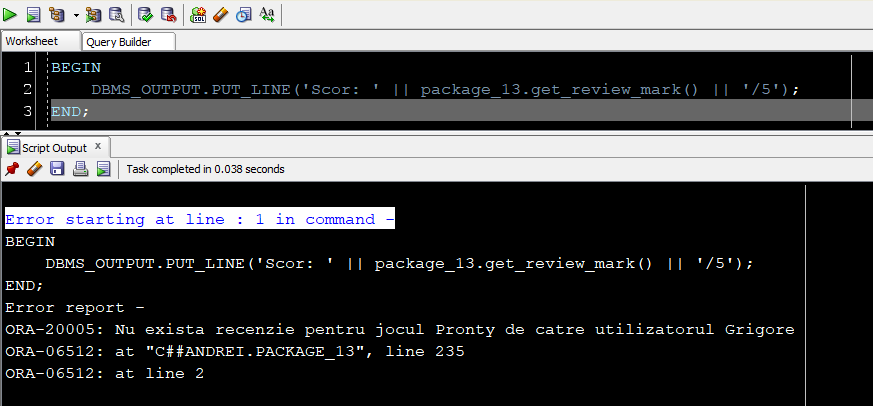


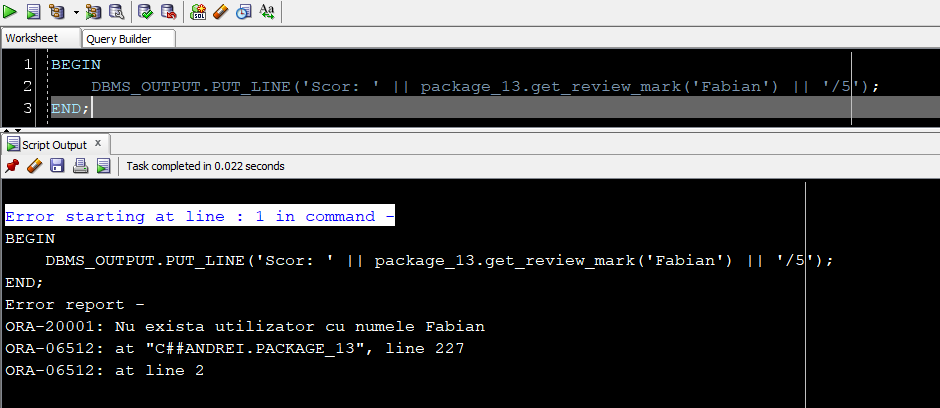


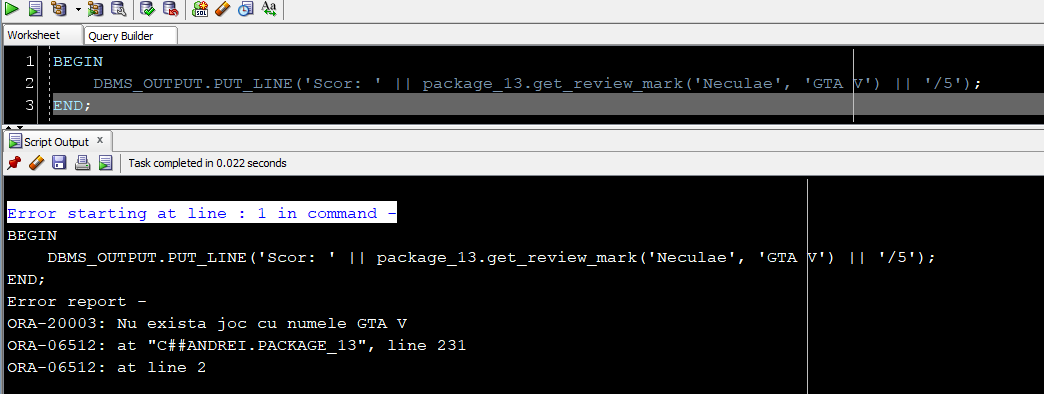


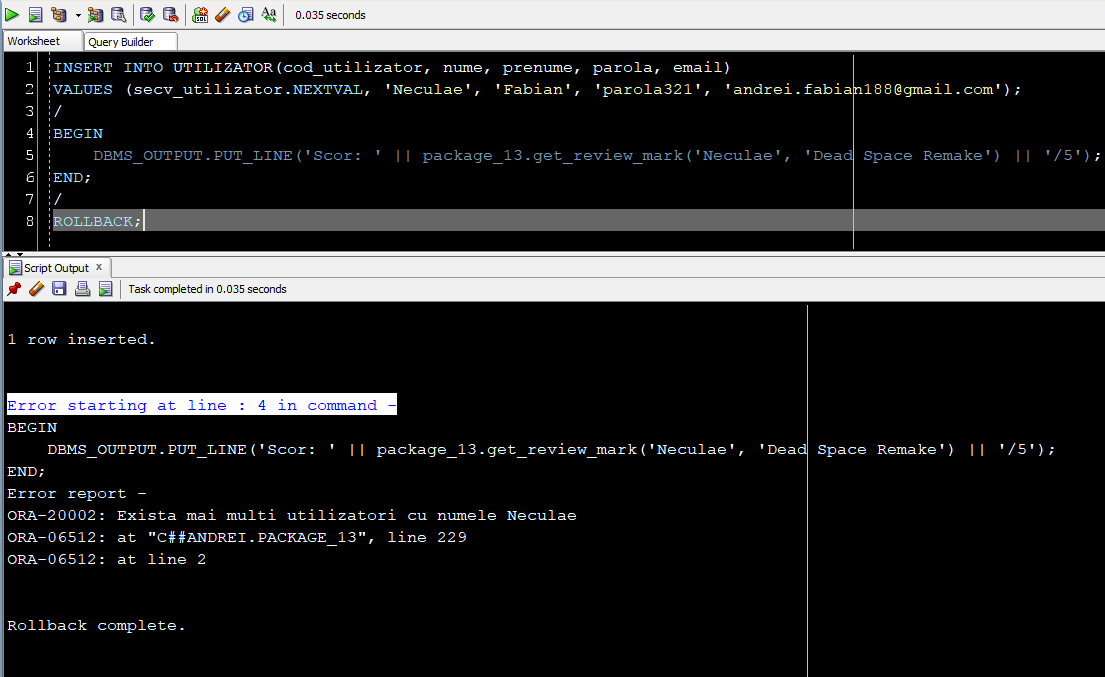


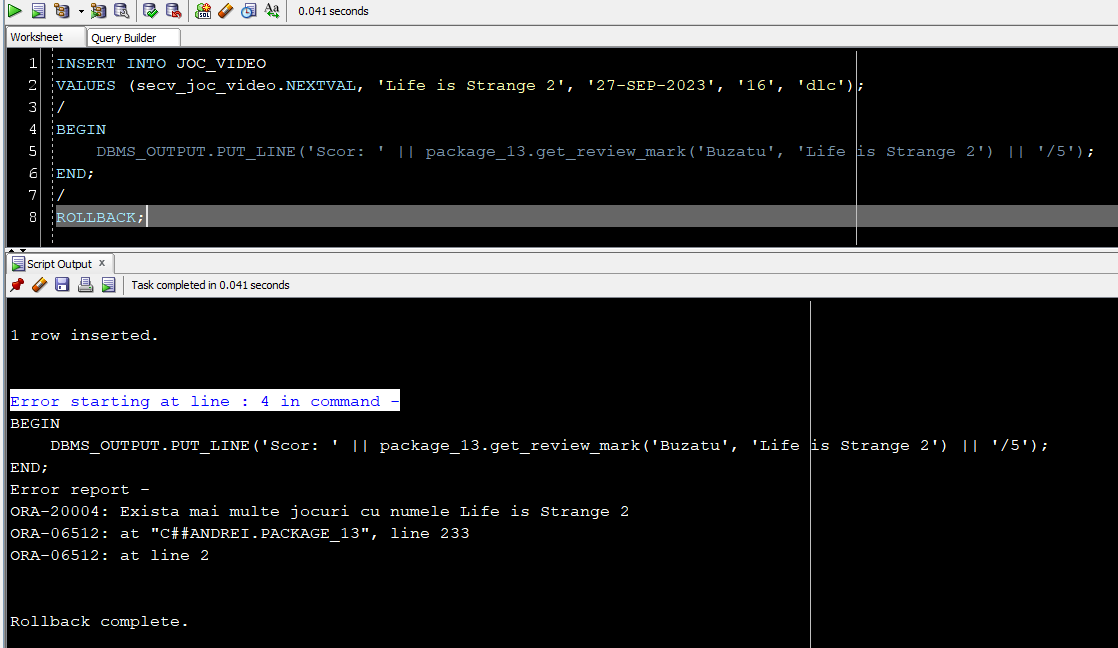


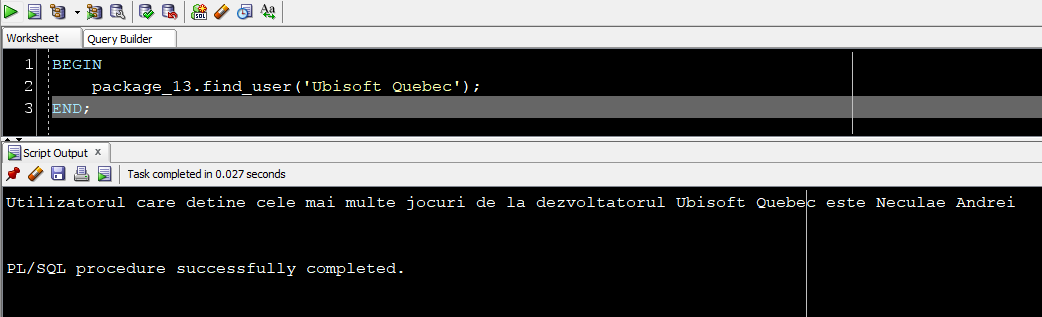


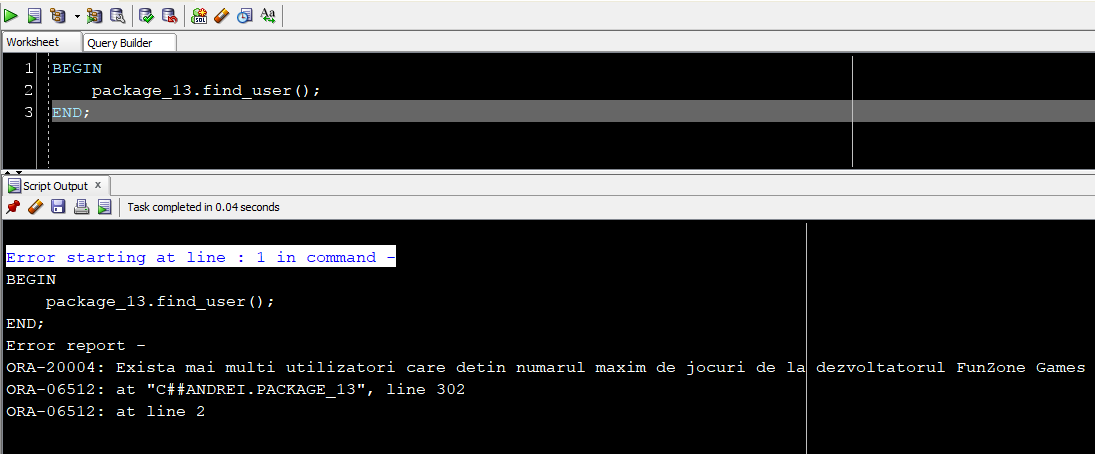


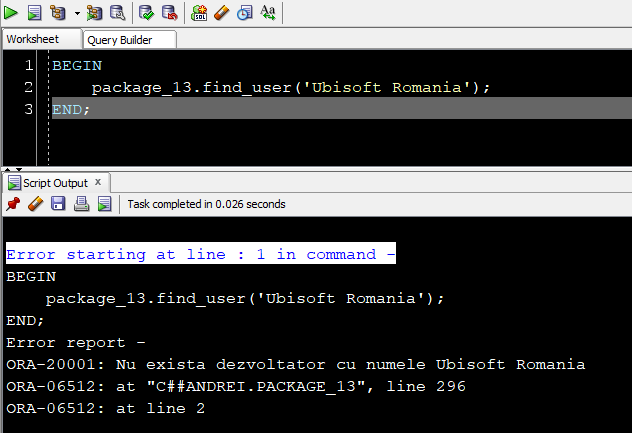


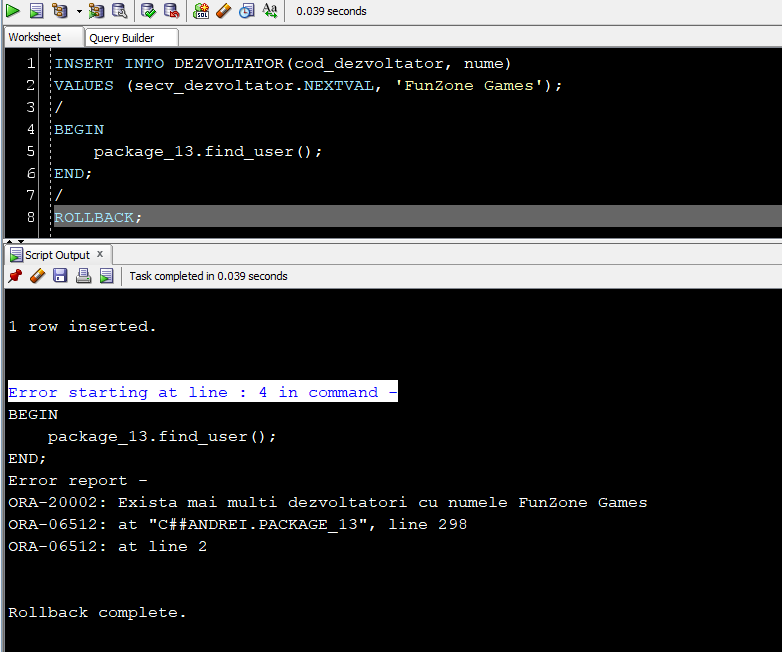


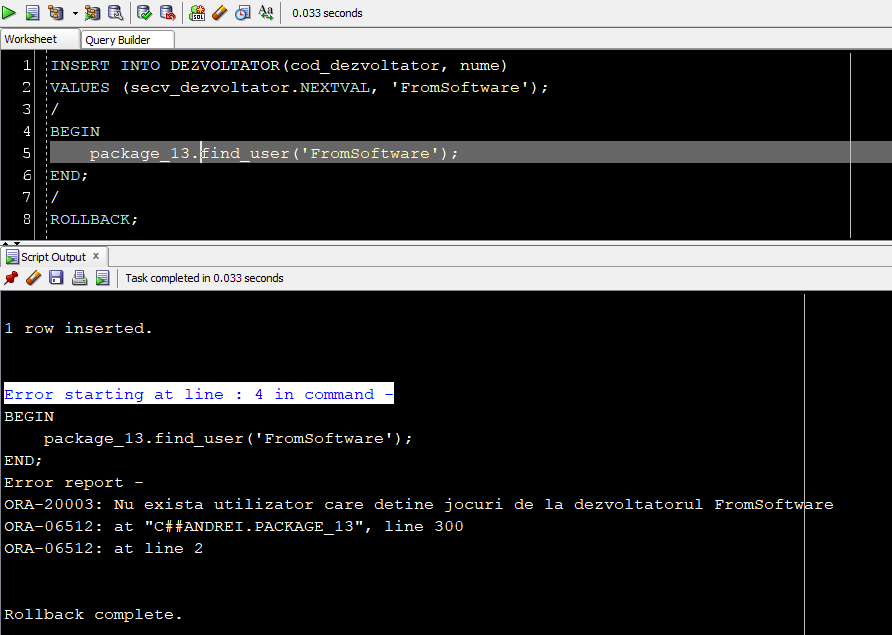












# Exercitiul 14

Cerinta: Definiti un pachet care sa includa tipuri de date complexe si obiecte necesare unui flux de actiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 functii, minim 2 proceduri).

Rezolvare: Definiti un pachet care sa contina urmatoarele:

* O procedura care sa afiseze numele, tipul (baza/dlc), dezvoltatorul, editorul si scorul tuturor jocurilor care fac parte dintr-o categorie data ca parametru;
* O procedura care insereaza intr-un tabel numele, prenumele si jocurile pe care le detine un utilizator. Numele si prenumele sunt date ca parametru;
* O functie care primeste ca parametrii numele unui utilizator si returneaza numele, prenumele si toate conturile acestuia, sub forma Platforma-Username-Parola si un bool care determina daca parolele sunt cenzurate sau nu.
* O functie care returneaza numarul de jocuri detinute de cel putin n utilizatori, unde n este dat ca prim parametru, si produse de catre dezvoltatorul al carui nume este dat ca al doilea parametru.

|  |
| --- |
| **CREATE** **OR** **REPLACE** **TYPE** tablou\_imbricat\_joc **AS** **TABLE** **OF** VARCHAR2(**50**);  /  **CREATE** **TABLE** info\_utilizator  (  nume VARCHAR2(**30**),  prenume VARCHAR2(**30**),  jocuri tablou\_imbricat\_joc  ) NESTED **TABLE** jocuri STORE **AS** tablou\_jocuri;  /  **CREATE** **OR** **REPLACE** PACKAGE package\_14 **AS**  **TYPE** tuplu\_joc **IS** RECORD  (  nume\_joc joc\_video**.**nume%TYPE,  tip\_joc joc\_video**.**tip%TYPE,  nume\_dezvoltator dezvoltator**.**nume%TYPE,  nume\_editor editor**.**nume%TYPE  );  **TYPE** tablou\_indexat\_joc **IS** **TABLE** **OF** tuplu\_joc **INDEX** **BY** PLS\_INTEGER;  **TYPE** tuplu\_info **IS** RECORD  (  username cont**.**nume%TYPE,  parola cont**.**parola%TYPE,  nume\_platforma platforma**.**nume%TYPE  );  **TYPE** tablou\_imbricat\_info **IS** **TABLE** **OF** tuplu\_info;  **TYPE** info **IS** RECORD  (  nume utilizator**.**nume%TYPE,  prenume utilizator**.**prenume%TYPE,  conturi tablou\_imbricat\_info  );  **PROCEDURE** category\_games\_details(v\_categorie categorie**.**nume%TYPE **DEFAULT** 'Drama');  **PROCEDURE** insert\_values(v\_nume utilizator**.**nume%TYPE **DEFAULT** 'Neculae', v\_prenume utilizator**.**prenume%TYPE **DEFAULT** 'Andrei');  **FUNCTION** get\_info(v\_nume utilizator**.**nume%TYPE **DEFAULT** 'Neculae', v\_cenzura NUMBER **DEFAULT** **1**) **RETURN** info;  **FUNCTION** nr\_games(v\_min NUMBER **DEFAULT** **1**, v\_dezvoltator dezvoltator**.**nume%TYPE **DEFAULT** 'DONTNOD Entertainment') **RETURN** NUMBER;  **END**;  /  **CREATE** **OR** **REPLACE** PACKAGE BODY package\_14 **AS**  **PROCEDURE** category\_games\_details  (  v\_categorie categorie**.**nume%TYPE **DEFAULT** 'Drama'  )  **IS**  v\_jocuri tablou\_indexat\_joc;  v\_scor\_mediu NUMBER;  **BEGIN**  **SELECT** j**.**nume, INITCAP(j**.**tip), d**.**nume, e**.**nume  BULK COLLECT **INTO** v\_jocuri  **FROM** joc\_video j  **JOIN** dezvoltator\_editor\_joc\_video dejv **ON** j**.**cod\_joc = dejv**.**cod\_joc  **JOIN** dezvoltator d **ON** dejv**.**cod\_dezvoltator = d**.**cod\_dezvoltator  **JOIN** editor e **ON** dejv**.**cod\_editor = e**.**cod\_editor  **JOIN** joc\_video\_categorie jvc **ON** j**.**cod\_joc = jvc**.**cod\_joc  **JOIN** categorie c **ON** jvc**.**cod\_categorie = c**.**cod\_categorie  **WHERE** INITCAP(c**.**nume) = INITCAP(v\_categorie);  **FOR** I **IN** **1..**v\_jocuri**.**COUNT **LOOP**  **SELECT** AVG(scor)  **INTO** v\_scor\_mediu  **FROM** recenzie r  **JOIN** joc\_video j **ON** r**.**cod\_joc = j**.**cod\_joc  **WHERE** j**.**nume = v\_jocuri(i)**.**nume\_joc;  DBMS\_OUTPUT**.**PUT\_LINE('Nume joc : ' || v\_jocuri(i)**.**nume\_joc || chr(**10**) ||  'Tip joc : ' || v\_jocuri(i)**.**tip\_joc || chr(**10**) ||  'Dezvoltator : ' || v\_jocuri(i)**.**nume\_dezvoltator || chr(**10**) ||  'Editor : ' || v\_jocuri(i)**.**nume\_editor);  **IF** v\_scor\_mediu **IS** **NULL** **THEN**  DBMS\_OUTPUT**.**PUT\_LINE('Scor : nu are recenzii' || chr(**10**));  **ELSE**  DBMS\_OUTPUT**.**PUT\_LINE('Scor : ' || v\_scor\_mediu || '/5' || chr(**10**));  **END** **IF**;    DBMS\_OUTPUT**.**NEW\_LINE;  **END** **LOOP**;  **EXCEPTION**  **WHEN** NO\_DATA\_FOUND **THEN**  RAISE\_APPLICATION\_ERROR(-**20001**, 'Nu exista categorie cu numele ' || v\_categorie);  **END**;  **PROCEDURE** insert\_values  (  v\_nume utilizator**.**nume%TYPE **DEFAULT** 'Neculae',  v\_prenume utilizator**.**prenume%TYPE **DEFAULT** 'Andrei'  )  **IS**  v\_jocuri tablou\_imbricat\_joc := tablou\_imbricat\_joc();  v\_exista NUMBER(**1**);  **BEGIN**  **SELECT** j**.**nume  BULK COLLECT **INTO** v\_jocuri  **FROM** joc\_video j  **JOIN** utilizator\_joc\_video\_cont ujvc **ON** j**.**cod\_joc = ujvc**.**cod\_joc  **JOIN** utilizator u **ON** ujvc**.**cod\_utilizator = u**.**cod\_utilizator  **WHERE** INITCAP(u**.**nume) = INITCAP(v\_nume) **AND** INITCAP(u**.**prenume) = INITCAP(v\_prenume);  **SELECT** COUNT(\*)  **INTO** v\_exista  **FROM** info\_utilizator  **WHERE** nume = v\_nume **AND** prenume = v\_prenume;  **IF** v\_exista = **1** **THEN**  **UPDATE** info\_utilizator  **SET** jocuri = v\_jocuri  **WHERE** nume = v\_nume **AND** prenume = v\_prenume;  **ELSE**  **INSERT** **INTO** info\_utilizator  **VALUES** (v\_nume, v\_prenume, v\_jocuri);  **END** **IF**;  **EXCEPTION**  **WHEN** NO\_DATA\_FOUND **THEN**  RAISE\_APPLICATION\_ERROR(-**20001**, 'Nu exista utilizator cu numele ' || v\_nume || ' ' || v\_prenume);  **WHEN** TOO\_MANY\_ROWS **THEN**  RAISE\_APPLICATION\_ERROR(-**20002**, 'Exista mai multi utilizatori cu numele ' || v\_nume || ' ' || v\_prenume);  **END**;  **FUNCTION** get\_info  (  v\_nume utilizator**.**nume%TYPE **DEFAULT** 'Neculae',  v\_cenzura NUMBER **DEFAULT** **1**  ) **RETURN** info  **IS**  v\_info info;  v\_conturi tablou\_imbricat\_info := tablou\_imbricat\_info();  **BEGIN**  **SELECT** u**.**nume, u**.**prenume  **INTO** v\_info**.**nume, v\_info**.**prenume  **FROM** utilizator u  **WHERE** INITCAP(u**.**nume) = INITCAP(v\_nume);  **SELECT** **DISTINCT** c**.**nume, c**.**parola, p**.**nume  BULK COLLECT **INTO** v\_conturi  **FROM** cont c  **JOIN** utilizator\_joc\_video\_cont ujvc **ON** c**.**cod\_cont = ujvc**.**cod\_cont  **JOIN** platforma p **ON** c**.**cod\_platforma = p**.**cod\_platforma  **JOIN** utilizator u **ON** ujvc**.**cod\_utilizator = u**.**cod\_utilizator  **WHERE** INITCAP(u**.**nume) = INITCAP(v\_nume);  **IF** v\_cenzura = **1** **THEN**  **FOR** I **IN** **1..**v\_conturi**.**COUNT **LOOP**  v\_conturi(i)**.**parola := RPAD('\*', LENGTH(v\_conturi(i)**.**parola), '\*');  **END** **LOOP**;  **END** **IF**;  v\_info**.**conturi := v\_conturi;  **RETURN** v\_info;  **EXCEPTION**  **WHEN** NO\_DATA\_FOUND **THEN**  RAISE\_APPLICATION\_ERROR(-**20001**, 'Nu exista utilizator cu numele ' || v\_nume);  **WHEN** TOO\_MANY\_ROWS **THEN**  RAISE\_APPLICATION\_ERROR(-**20002**, 'Exista mai multi utilizatori cu numele ' || v\_nume);  **END**;  **FUNCTION** nr\_games  (  v\_min NUMBER **DEFAULT** **1**,  v\_dezvoltator dezvoltator**.**nume%TYPE **DEFAULT** 'DONTNOD Entertainment'  ) **RETURN** NUMBER  **IS**  **CURSOR** jocuri(v\_dezvoltator dezvoltator**.**nume%TYPE) **IS**  **SELECT** j**.**cod\_joc  **FROM** joc\_video j  **JOIN** dezvoltator\_editor\_joc\_video dejv **ON** j**.**cod\_joc = dejv**.**cod\_joc  **JOIN** dezvoltator d **ON** dejv**.**cod\_dezvoltator = d**.**cod\_dezvoltator  **WHERE** INITCAP(d**.**nume) = INITCAP(v\_dezvoltator);  j joc\_video**.**cod\_joc%TYPE;  v\_nr\_utilizaor NUMBER := **0**;  v\_nr\_joc NUMBER := **0**;  **BEGIN**  **OPEN** jocuri(v\_dezvoltator);  **LOOP**  **FETCH** jocuri **INTO** j;  **EXIT** **WHEN** jocuri%NOTFOUND;  **SELECT** COUNT(\*)  **INTO** v\_nr\_utilizaor  **FROM** utilizator\_joc\_video\_cont ujvc  **JOIN** joc\_video j **ON** ujvc**.**cod\_joc = j**.**cod\_joc  **WHERE** ujvc**.**cod\_joc = j;  **IF** v\_nr\_utilizaor >= v\_min **THEN**  v\_nr\_joc := v\_nr\_joc + **1**;  **END** **IF**;  **END** **LOOP**;  **CLOSE** jocuri;  **RETURN** v\_nr\_joc;  **EXCEPTION**  **WHEN** NO\_DATA\_FOUND **THEN**  RAISE\_APPLICATION\_ERROR(-**20001**, 'Nu exista dezvoltator cu numele ' || v\_dezvoltator);  **WHEN** TOO\_MANY\_ROWS **THEN**  RAISE\_APPLICATION\_ERROR(-**20002**, 'Exista mai multi dezvoltatori cu numele ' || v\_dezvoltator);  **END**;  **END**;  /  **DECLARE**  informatii package\_14**.**info := package\_14**.**get\_info(**0**, 'Toader');  **BEGIN**  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  DBMS\_OUTPUT**.**PUT\_LINE('Conturile utilizatorului ' || informatii**.**nume || ' ' || informatii**.**prenume);  DBMS\_OUTPUT**.**PUT\_LINE('-------------------------------------');  **FOR** i **IN** **1..**informatii**.**conturi**.**COUNT **LOOP**  DBMS\_OUTPUT**.**PUT\_LINE('Store : ' || informatii**.**conturi(i)**.**nume\_platforma || chr(**10**) ||  'Username: ' || informatii**.**conturi(i)**.**username || chr(**10**) ||  'Password: ' || informatii**.**conturi(i)**.**parola || chr(**10**));  DBMS\_OUTPUT**.**NEW\_LINE;  **END** **LOOP**;  **END**; |

