PROIECT BAZE DE DATE  
   
  
 MOTOGP  
  
  
   
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
 FURDUI VLAD RAREȘ  
 GRUPA 141  
  
   
  
  
  
  
  
   
  
  
                           
**1. Descrierea modelului real,a utilității acestuia și a regulilor de funcționare**  
   
Pentru o cât mai bună organizare a campionatului mondial de motocilcism viteza MotoGP, compania Dorna își propune o gestionare cât mai ușoară a celor 22 de etape, numite Grand Prix-uri, precum și a curselor ce au loc de-a lungul acestora.   
   
Fiecare Grand Prix se poate desfășura pe un singur circuit din motive de egalitate față de categoriile tinere. Însă se pot desfășura mai multe evenimente de acest fel într-o țară, dacă aceasta dispune de mai multe circuite omologate cu categoria A.  În cadrul curselor desfășurate participă echipe care pot fi de uzină, ce își produc singure motoarele, dar și cele de tip privat. Acestea sunt reprezentate de un singur pilot, prin încheierea unui contract ce durează minim jumătate de sezon. Totodată, se intenționează transmiterea acestor curse world wide, fie printr-un abonament sau prin sistemul pay-per-view.    
   
Fiecare cursă trebuie să se desfășoare în parametrii optimi, astfel că acestea sunt supervizate de stewarzi. De asemenea, pentru ca fiecare motocicletă să ruleze în concordanță cu dorințele echipei, acestea au dreptul de a aduce ingineri .    
   
**2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.**  
   
1. Un Grand Prix poate avea mai multe curse.   
   
2. Un spectator poate participa la mai multe Grand Prix-uri.   
   
3. Un Grand Prix se poate desfășura într-un singur loc la un moment de timp, dar turneul poate ajunge de mai multe ori în aceeași țară dacă sunt mai multe circuite acolo.   
   
4. Un Grand Prix poate avea un singur sponsor.   
   
5. Cursele pot fi transmise ori prin abonament ori prin PPV.   
   
6.O cursă nu se poate avea loc fără stewarzi.   
   
7. În cadrul mai multor curse pot lua parte mai multe echipe.   
   
8. Fiecare echipă poate aduce niciunul, unul sau mai mulți ingineri.   
   
9. Fiecare pilot trebuie să aibă un contract activ cu o echipă pentru a concura.   
   
**3. Descrierea entităților, incluzând precizarea cheii primare.**   
   
Toate entitățile sunt independente, în afară de contract.   
   
*Grand\_Prix -* reprezintă una dintre etapele campionatului mondial de motociclism viteză. Acesta se desfășoară de-a lungul mai multor zile în fața a zeci de mii de oameni pe viu si a sutelor de mii din fața ecranelor. Are ca și cheie primară cod\_gp.   
   
*Curse-* reprezintă apogeul Grand Prix-ului, momentul care decide care dintre cei mai abili piloți din lume este cel mai rapid pe circuitul din cadrul evenimentului. Are ca și cheie primară cod\_cursa.   
   
        *MotoGP-* subentitate a Curse, cateogorie unde motorul are cea mai mare capacitate, 1000cm^3 și unde se ating viteze de până la 360 km/h. Are ca și cheie primară cod\_cursa.   
   
*Moto2-* subentitate a Curse, categorie unde motorul are o capacitate de 756 cm^3 și unde se întrec viitorii piloți MotoGp, dar și unii experimentați care nu au făcut față la clasa regină. Are ca și cheie primară cod\_cursa.   
   
*Moto3-* subentitate a Curse, categorie unde motorul nu depășește 250 cm^3 și unde se întrec juniorii, cu vârste în medie de 19-20 de ani. Are ca și cheie primară cod\_cursa.   
   
*Sponsor-* în general o companie ce are legătură cu fenomenul, fie producător de anvelope sau băutură energizantă și care este inclus în numele Grand Prix-ului. Are ca și cheie primară cod\_sponsor.   
   
*Fani-*plătitori de bilet care vizualizează live cursele, fie din tribune sau din zonele de camping ce fac parte din alcătuirea circuitului. Are ca și cheie primară cod\_fan.   
   
*Circuit-* locul unde se întrec motocicletele, fiecare diferă prin numărul de viraje, dificultate sau elevație. Are ca și cheie primară cod\_circuit.   
   
*Transmisiune-* modul prin care întreaga lume poate vedea spectacolul creat de motociclete. Are ca și cheie primară cod\_transmisiune.   
   
       *PPV – subentitate a Transimisiune,* mod de a vedea doar anumite curse, plătind pentru vizionarea fiecăreia în parte. Are ca și cheie primară cod\_transmisiune.   
   
       *Abonament – subentitate a Transimisiune,* mod de a vedea fiecare cursă, este plătit lunar. Are ca și cheie primară cod\_transmisiune.   
   
*Steward-* persoană întrebuințată de a se asigura că siguranța piloților și a fanilor nu este pusă în pericol, dar și că fiecare cursă decurge în parametrii optimi și legali. Are ca și cheie primară cod\_steward.   
   
*Echipe-* precum în oricare alt sport, acestea participă fiecare cu motocicleta și specificațiile sale în dorința de a se clasa cât mai sus posibil. Are ca și cheie primară cod\_echipa.   
   
      *Uzină* - subentitate a Echipe, tip de echipă, de obicei cu un buget mare și care își face propriile motociclete. Are ca și cheie primară cod\_echipa.   
   
      *Privată -* subentitate a Echipe , spre deosebire de cele de uzină, acestea cumpără motociclete de la cele de uzină. Are ca și cheie primară cod\_echipa.   
   
*Ingineri -* persoane convocate de către fiecare echipă în parte cu scopul de a se asigura că motocicleta merge în parametrii optimi și, în caz contrar, de a repara cât mai repede defecțiunile tehnice. Are ca și cheie primară cod\_inginer.   
   
*Contract -* deține informații despre înțelegerea dintre echipă și pilot. Are ca și cheie primară cod\_echipa, cod\_pilot, pentru a permite echipelor să schimbe pilotul din mers dacă le permite contractul.   
   
*Pilot –* atlet desăvârșit care poate concura doar pentru o singură echipă, cea cu care are contractul încheiat. Are ca și cheie primară cod\_pilot.   
   
   
   
   
   
   
   
   
 **4 Descrierea relațiilor, incluzând precizarea cardinalității acestora.**   
   
SPONSOR sponsorizează GRAND\_PRIX : relație de tip one-to-many care leagă entitățile SPONSOR și GRAND\_PRIX, reprezentând care este numele sponsorului pentru fiecare grand prix. Aceasta are cardinalitate minimă 0:1( un Grand Prix poate să nu aibă un sponsor) și cardinalitate maximă 1 : n (fiecare Grand Prix poate să aibă un sponsor).   
   
FANI participă\_la GRAND\_PRIX : relație de tip many-to-many care leagă entitățile FANI și GRAND\_PRIX, reprezentând care fani participă la care Grand Prix-uri. Aceasta are cardinalitate minimă 0:1( un Grand Prix poate să nu aibă fani) și cardinalitate maximă m : n (fiecare Grand Prix poate să aibă mai mulți fani).   
   
GRAND\_PRIX are\_loc\_la CIRCUIT : relație de tip many-to-one care leagă entitățile CIRCUIT și GRAND\_PRIX, reprezentând care este circuitul pentru fiecare grand prix. Aceasta are cardinalitate minimă 1:1( un Grand Prix se poate desfășura pe un singur circuit) și cardinalitate maximă 1 : n (mai multe Grand Prix-uri se pot desfășura pe un circuit).   
   
GRAND\_PRIX  desfășoară CURSE : relație de tip one-to-many care leagă entitățile CURSE și GRAND\_PRIX, reprezentând care sunt cursele pentru fiecare grand prix. Aceasta are cardinalitate minimă 1:1( un Grand Prix poate să aibă o cursă) și cardinalitate maximă 1 : n (nu se poate desfășura o cursă pentru mai multe grand prix-uri).   
   
CURSE sunt\_transmise\_de TRANSMISIUNE : relație de tip many-to-many care leagă entitățile TRANSMISIUNE și CURSE, reprezentând care sunt transimisiunile ce prezintă cursele . Aceasta are cardinalitate minimă 1:1( o cursă poate fi transmisă de o transmisiune) și cardinalitate maximă m : n (mai multe curse pot fi transmise de mai multe transmisiuni).   
   
ECHIPE încheie CONTRACT : relație de tip many-to-one care leagă entitățile ECHIPE și CONTRACT, reprezentând contractul încheiat de o echipă. Aceasta are cardinalitate minimă 1:1( o echipă poate încheia un contract) și cardinalitate maximă  n : 1 (o echipă poate încheia mai multe contracte).   
   
PILOT încheie CONTRACT : relație de tip one-to-many care leagă entitățile PILOTși CONTRACT, reprezentând contractul încheiat cu un pilot. Aceasta are cardinalitate minimă 1:1( un pilot poate încheia un contract) și cardinalitate maximă 1 : n (mai multe contracte pot fi încheiate cu un pilot).   
   
STEWARD asigură CURSE : relație de tip many-to-many care leagă entitățile STEWARD și CURSE, reprezentând care sunt stewarzii ce asigură cursele . Aceasta are cardinalitate minimă n:1( o cursă trebuie asigurată de mai mulți stewarzi) și cardinalitate maximă m : n (mai multe curse pot fi asigurate de mai mulți stewarzi).   
   
   
INGINERI lucrează\_la ECHIPE concurează\_la CURSE : relație de tip 3 care leagă entitățiile INGINERI, ECHIPE, și CURSE și care arată  la care echipe lucrează inginerii și în care curse concurează acestea.    
   
**5 Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.** 

Entitatea independenta SPONSOR are urmatoarele atribute

      cod\_sponsor: variabilă de tip întreg, de lungime maximă 5, care reprezintă codul sponsorului

      nume: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă numele sponsorului

      suma: variabilă de tip întreg, de lungime maximă 5, care reprezintă suma platita de sponsor

Entitatea independenta GRAND\_PRIX are urmatoarele atribute

      cod\_gp: variabilă de tip întreg, de lungime maximă 5, care reprezintă codul grand prix-ului   
   
      cod\_circuit: variabilă de tip întreg, de lungime maximă 5, care reprezintă codul circuitului unde se desfășoară grand prix-ul. Atributul trebuie să corespundă la o valoare a cheii primare din tabela CIRCUIT   
   
  cod\_sponsor: variabilă de tip întreg, de lungime maximă 5, care reprezintă codul sponsorului pentru respectivul grand prix. Atributul trebuie să corespundă la o valoare a cheii primare din tabela SPONSOR. 

      nume: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă numele grand prix-ului

      data: variabilă de tip dată calendaristică ce reprezintă când are loc grand prix-ul

      vechime: variabilă de tip întreg, de lungime maximă 5, care reprezintă vechimea grand prix-ului

Entitatea independenta FANI are urmatoarele atribute

      cod\_fan: variabilă de tip întreg, de lungime maximă 5, care reprezintă codul fanilor   
   
  cod\_gp: variabilă de tip întreg, de lungime maximă 5, care reprezintă codul grand prix-ului la care merge fanul. atributul trebuie să corespundă la o valoare a cheii primare din tabela GRAND\_PRIX.

      nume: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă numele fanului

      prenume: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă prenumele fanului

      nationalitate: variabila de tip șir de caractere, de lungime maximă 23, care reprezintă țara de unde este fanul

Entitatea independenta CIRCUIT are urmatoarele atribute

      cod\_circuit: variabilă de tip întreg, de lungime maximă 5, care reprezintă codul circuitului

      nume: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă numele circuitului

      lungime: variabilă de tip întreg, de lungime maximă 6, 2 fiind zecimale, care reprezintă lungimea circuitului

      viraje: variabilă de tip întreg, de lungime maximă 2, care reprezintă numărul virajelor

Entitatea independenta CURSE are urmatoarele atribute

      cod\_cursa: variabilă de tip întreg, de lungime maximă 5, care reprezintă codul cursei   
    
  cod\_gp: variabilă de tip întreg, de lungime maximă 5, care reprezintă codul grand prix-ului unde au loc cursele. Atributul trebuie să corespundă la o valoare a cheii primare din tabela GRAND\_PRIX.

      lap: variabilă de tip întreg, de lungime maximă 2, care reprezintă numărul de ture al cursei

Subentitatea MotoGP are urmatoarele atribute

      cod\_cursa: variabilă de tip întreg, de lungime maximă 5, care reprezintă codul cursei de clasa MotoGP

      castigator: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă numele ultimului câștigător

      ora: lungime: variabilă de tip întreg, de lungime maximă 4, 2 fiind zecimale, care reprezintă ora la care începe cursa

Subentitatea Moto2 are urmatoarele atribute

      cod\_cursa - variabilă de tip întreg, de lungime maximă 5, care reprezintă codul cursei de clasa Moto2

      castigator: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă numele ultimului câștigător

      ora: lungime: variabilă de tip întreg, de lungime maximă 4, 2 fiind zecimale, care reprezintă ora la care începe cursa

Subentitatea Moto3 are urmatoarele atribute

      cod\_cursa - variabilă de tip întreg, de lungime maximă 5, care reprezintă codul cursei de clasa Moto3

      castigator: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă numele ultimului câștigător

      ora: lungime: variabilă de tip întreg, de lungime maximă 4, 2 fiind zecimale, care reprezintă ora la care începe cursa

Entitatea independenta STEWARD are urmatoarele atribute

      cod\_steward - variabilă de tip întreg, de lungime maximă 5, care reprezintă codul stewardului

      nume: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă numele fanului

      prenume: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă prenumele fanului

      viraj: variabilă de tip întreg, de lungime maximă 2, care reprezintă numărul virajului la care acesta activează

Entitatea independenta TRANSMISIUNE are urmatoarele atribute

      cod\_transmisiune - variabilă de tip întreg, de lungime maximă 5, care reprezintă codul transmisiunii

      nume: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă numele transmisiunii

      tara: variabilă de tip șir de caractere, de lungime maximă 30 care reprezintă locul unde se transmite cursa

Subentitatea PPV are urmatoarele atribute

      cod\_transmisiune - variabilă de tip întreg, de lungime maximă 5, care reprezintă codul evenimentului pay-per-view

      pret: variabilă de tip întreg, de lungime maximă 5, care reprezintă pretul evenimentului

Subentitatea ABONAMENT are urmatoarele atribute

      cod\_transmisiune - variabilă de tip întreg, de lungime maximă 5, care reprezintă codul abonamentului pentru fiecare cursă

      perioada: variabilă de tip dată calendaristică ce reprezintă când se termină abonamentul

Entitatea independenta ECHIPE are urmatoarele atribute

      cod\_echipa - variabilă de tip întreg, de lungime maximă 5, care reprezintă codul echipei   
       nume: variabilă de tip șir de caractere, de lungime maximă 30, care reprezintă numele echipei

      tip: variabilă de tip șir de caractere, de lungime maximă 10, care reprezintă tipul motorului

Subentitatea UZINĂ are urmatoarele atribute

      cod\_echipa - variabilă de tip întreg, de lungime maximă 5, care reprezintă codul echipei de uzină

      locatie: variabilă de tip șir de caractere, de lungime maximă 20, care reprezintă locul sediului echipei

Subentitatea PRIVAT are urmatoarele atribute

      cod\_echipa - variabilă de tip întreg, de lungime maximă 5, care reprezintă codul echipei de profil privat

      motorizare: variabilă de tip șir de caractere, care reprezintă producătorul motorului

Entitatea independenta INGINERI are urmatoarele atribute

      cod\_inginer - variabilă de tip întreg, de lungime maximă 5, care reprezintă codul inginerului

      nume: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă numele inginerului

      prenume: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă prenumele inginerului

      post: variabilă de tip șir de caractere, de lungime maximă 30, care reprezintă specializarea acestuia

Entitatea dependenta CONTRACT are urmatoarele atribute

       cod\_echipa: variabilă de tip întreg, de lungime maximă 5, care reprezintă

codul echipei care încheie contractele; atributul trebuie să fie în concordanță cu o

valoare a cheii primare din tabela ECHIPE;

      cod\_pilot: variabilă de tip întreg, de lungime maximă 5, care reprezintă

codul pilotului care încheie contractul; atributul trebuie să fie în concordanță cu o

valoare a cheii primare din tabela PILOT;

      suma: variabilă de tip întreg, de lungime maximă 7, care reprezintă valoarea contractului

     perioada: variabilă de tip dată calendaristică ce reprezintă când se termină contractul

Entitatea independenta PILOT are urmatoarele atribute

      cod\_pilot- variabilă de tip întreg, de lungime maximă 5, care reprezintă codul pilotului

      nume: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă numele pilotului

      prenume: variabilă de tip șir de caractere, de lungime maximă 46 care reprezintă prenumele pilotului

      podium: variabilă de tip întreg, de lungime maximă 3, care reprezintă numărul de podiumuri al pilotului

Relația CURSE\_sunt\_transmise\_TRANSMISIUNE:

        cod\_cursa: variabilă de tip întreg, de lungime maximă 5, care reprezintă

codul cursei care este transmisă; atributul trebuie să fie în concordanță cu o

valoare a cheii primare din tabela CURSE;

      cod\_transmisiune: variabilă de tip întreg, de lungime maximă 5, care reprezintă

codul transmisiunii cursei; atributul trebuie să fie în concordanță cu o

valoare a cheii primare din tabela TRANSMISIUNE;

Relația STEWARD\_asigura\_CURSE:

        cod\_cursa: variabilă de tip întreg, de lungime maximă 5, care reprezintă

codul cursei care este asigurată; atributul trebuie să fie în concordanță cu o

valoare a cheii primare din tabela CURSE;

      cod\_steward: variabilă de tip întreg, de lungime maximă 5, care reprezintă

codul stewardului ce asigura cursa; atributul trebuie să fie în concordanță cu o

valoare a cheii primare din tabela STEWARD;

       nr: variabilă de tip întreg, cu lungime maximă 3, care reprezintă numărul de stewarzi

Relația FANI\_participa\_la\_GRAND\_PRIX:

      cod\_gp: variabilă de tip întreg, de lungime maximă 5, care reprezintă

codul grand prix-ului la care merg fanii; atributul trebuie să fie în concordanță cu o

valoare a cheii primare din tabela GRAND PRIX;

    cod\_fan: variabilă de tip întreg, de lungime maximă 5, care reprezintă

codul fanului ce participă la grand prix; atributul trebuie să fie în concordanță cu o

valoare a cheii primare din tabela FANI;

      nr: variabilă de tip întreg, cu lungime maximă 3, care reprezintă numărul de fani

Relația INGINERI lucrează\_la ECHIPE concurează\_la CURSE :

      cod\_inginer: variabilă de tip întreg, de lungime maximă 5, care reprezintă

codul inginerului din cadrul unei echipe; atributul trebuie să fie în concordanță cu o

valoare a cheii primare din tabela INGINERI;

    cod\_echipa: variabilă de tip întreg, de lungime maximă 5, care reprezintă

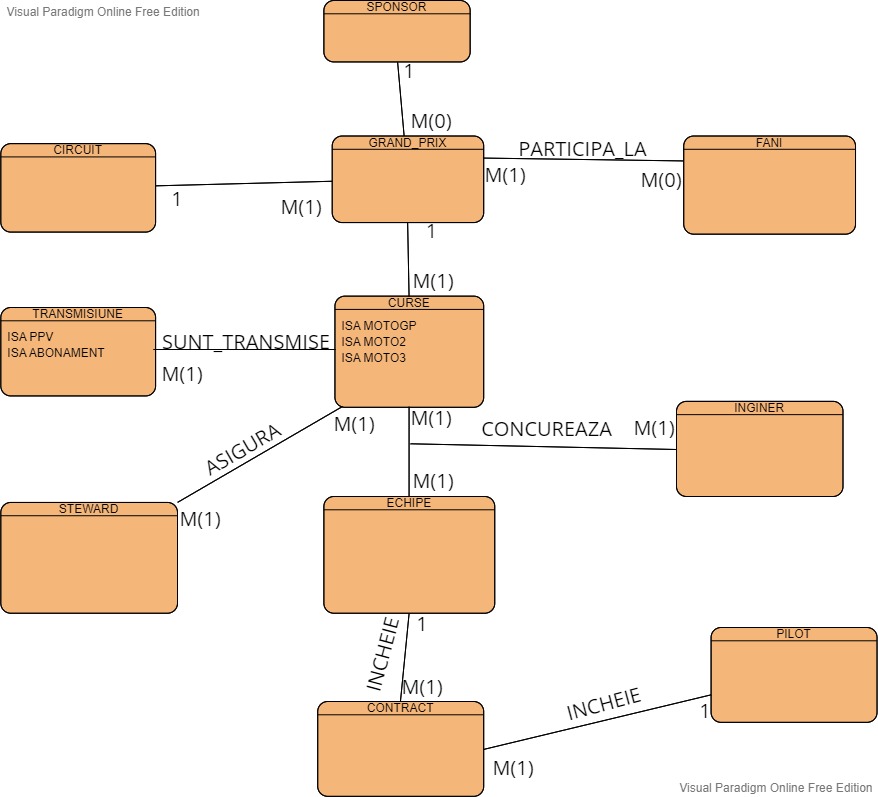
codul echipei ce participă la cursă; atributul trebuie să fie în concordanță cu o

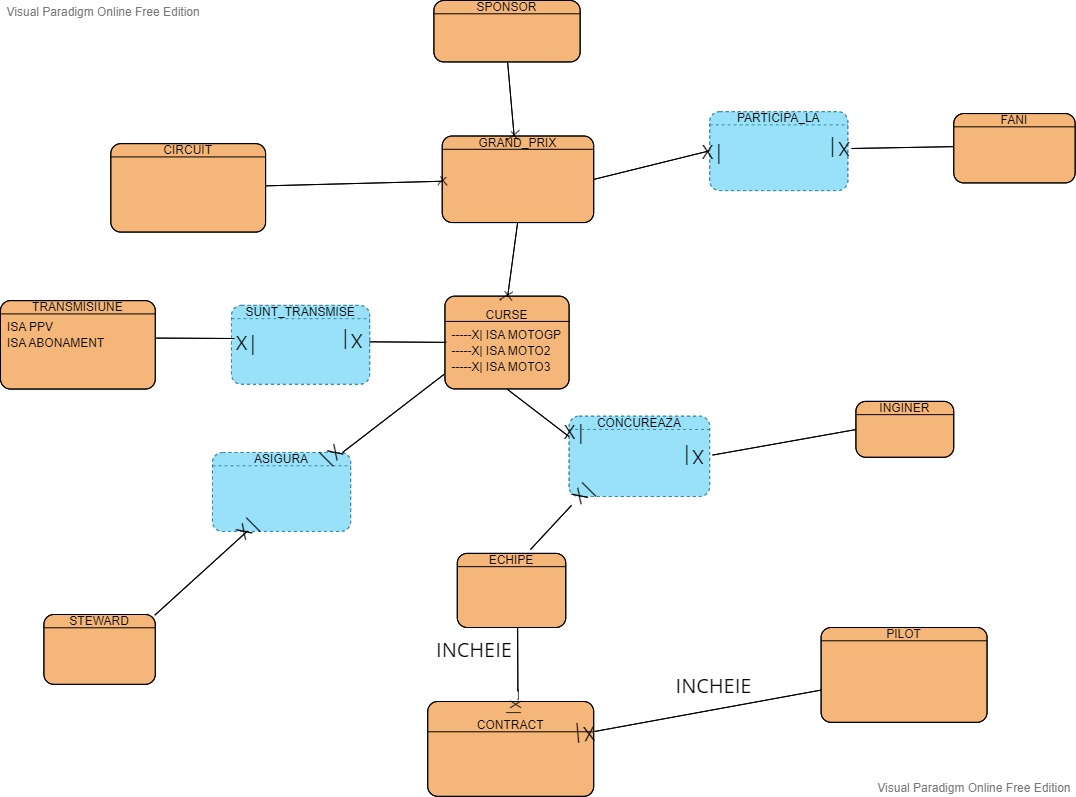
valoare a cheii primare din tabela ECHIPE;

    cod\_cursa: variabilă de tip întreg, de lungime maximă 5, care reprezintă

codul cursei la care participă echipele cu inginerii; atributul trebuie să fie în concordanță cu o

valoare a cheii primare din tabela CURSE;

   
   
**6 Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.**

**7.Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.**  
  
  
  
  
  
  
**8 Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.**SPONSOR(cod\_sponsor#, nume, suma)  
  
GRAND\_PRIX(cod\_gp#, cod\_circuit, cod\_sponsor, nume, data)  
  
FANI( cod\_fan#, cod\_gp, nume, prenume, nationalitate)  
  
CIRCUIT(cod\_circuit#, nume, lungime, viraje)  
  
CURSE(cod\_cursa#, cod\_gp, lap)  
  
MotoGP(cod\_cursa#, castigator, ora)   
  
Moto2(cod\_cursa#, castigator, ora)  
  
Moto3(cod\_cursa#, castigator, ora)  
  
STEWARD(cod\_steward#, nume, prenume, viraj)  
  
TRANSMISIUNE(cod\_transmisiune#, nume, tara)  
  
PPV(cod\_transmisiune#, pret)  
  
ABONAMENT(cod\_transmisiune#, perioada)  
  
ECHIPE(cod\_echipa#, nume, tip)  
  
UZINĂ(cod\_echipa#, locatie)  
  
PRIVAT(cod\_echipa#, motorizare)  
  
INGINERI(cod\_inginer#, nume, prenume, post)  
  
CONTRACT(cod\_echipa#, cod\_pilot#, suma, perioada)  
  
PILOT(cod\_pilot#, nume, prenume, podium)  
  
SUNT\_TRANSMISE(cod\_cursa#, cod\_transmisiune#)  
  
ASIGURA(cod\_cursa#, cod\_steward#, nr)  
  
PARTICIPA\_LA(cod\_gp#, cod\_fan#, nr)  
  
LUCREAZA\_LA(cod\_inginer#, cod\_echipa#, cod\_cursa#)

**9 Realizarea normalizării până la forma normală 3(FN1-FN3).***Formă normala 1*  
  
Ca o relație să se regăsească în FN1, trebuie ca fiecare atribut care o compune să îi corespundă o valoare indivizibilă .  
  
Exemplu de relație care nu se află în FN1, bazată pe faptul că un inginer poate lucra pentru mai multe echipe în paralel în cadrul aceluiași grand prix.

|  |  |
| --- | --- |
| cod\_inginer# | cod\_echipa# |
| 12 | 3 |
| 17 | 2, 6 |
| 20 | 1, 3 |
| 30 | 4 |

|  |  |
| --- | --- |
| cod\_inginer# | cod\_echipa# |
| 12 | 3 |
| 17 | 2 |
| 17 | 6 |
| 20 | 1 |
| 20 | 3 |
| 30 | 4 |

O variantă de normalizare a acestui tabel este următoarea:

*Formă normala II*  
  
Știm că o relație este în forma normală II dacă este, în primul rând, în FN1, iar fiecare atribut care nu este cheie primară să fie dependent de întreaga cheie primară .  
  
Un exemplu de relatie FN1, dar care nu este FN2 este acest tabel:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| cod\_echipa# | cod\_pilot# | cod\_pilot# | tip | nume | prenume | victorii |
| 32 | 63 | 63 | V4 | Bagnaia | Pecco | 6 |
| 16 | 43 | 43 | V4-Linie | Miller | Jack | 3 |

|  |  |  |
| --- | --- | --- |
| cod\_echipa# | cod\_pilot# | victorii |
| 32 | 63 | 6 |
| 16 | 43 | 3 |

Pentru a avea această relație în forma normală II, observăm următoarele dependențe:  
  
{cod\_echipa#} -> {tip}  
{cod\_pilot#} -> {nume, prenume}  
{cod\_echipa#, cod\_pilot#} -> {victorii}  
Așadar, de aici reies 3 tabele:

|  |  |
| --- | --- |
| cod\_echipa# | tip |
| 32 | V4 |
| 16 | V4-Linie |

|  |  |  |
| --- | --- | --- |
| cod\_pilot# | nume | prenume |
| 63 | Bagnaia | Pecco |
| 43 | Miller | Jack |

Acum acestea se regăsesc in FN2.  
  
*Forma normala III*Ca o relație să se regăsească în FN3, aceasta trebuie să fie în forma normală 2 și fiecare atribut care nu este cheie să depindă de cheia primară.  
  
Astfel, tabelul GRAND\_PRIX inițial arată astfel:

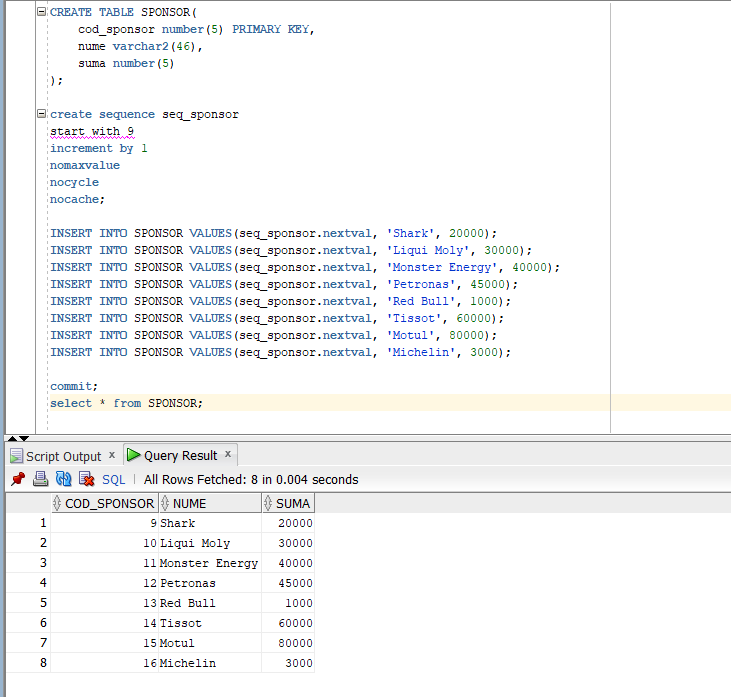
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cod\_gp# | nume | Data | nume(sponsor) | Tip\_nume(sponsor) |
| 10 | Bahrain GP | 17-05-2022 | Barwa | Real estate |
| 20 | Mugello GP | 18-03-2022 | Pirelli | Cauciucuri |
| 30 | Le Mans GP | 19-08-2022 | Michelin | Cauciucuri |
| 40 | Jerez GP | 20-10-2022 | Red Bull | Băutură |

Observăm că Tip\_nume(sponsor) depinde tranzitiv de cheia primară a atributului nume(sponsor)  
{cod\_gp#}->{nume, data, nume(sponsor)} – cod\_gp determină funcțional atributele nume, data și nume(sponsor)  
  
{cod\_gp#}->{nume, data, nume(sponsor)}->{Tip\_nume(sponsor)}  
  
Folosind regula Casey-Delobel descompunem relația dată în două proiecții:   
  
GRAND\_PRIX(cod\_gp#, nume, data, nume(sponsor))  
SPONSOR(cod\_sponsor#, Tip\_nume(sponsor)).  
  
Apoi, am adăugat la sponsor “nume(sponsor)” pentru a respecta regula formei normale 3 ca fiecare atribut să depindă de cheia primară.  
  
  
GRAND\_PRIX(cod\_gp#, nume, data, cod\_sponsor)  
SPONSOR(cod\_sponsor#, , nume(sponsor), Tip\_nume(sponsor)).  
  
Tabelul GRAND\_PRIX are această formă în urma normalizării:

|  |  |  |  |
| --- | --- | --- | --- |
| cod\_gp# | nume | Data | cod\_sponsor |
| 10 | Bahrain GP | 17-05-2022 | 100 |
| 20 | Mugello GP | 18-03-2022 | 200 |
| 30 | Le Mans GP | 19-08-2022 | 300 |
| 40 | Jerez GP | 20-10-2022 | 400 |

|  |  |  |
| --- | --- | --- |
| cod\_sponsor# | nume(sponsor) | Tip\_nume(sponsor) |
| 10 | Barwa | Real estate |
| 20 | Pirelli | Cauciucuri |
| 30 | Michelin | Cauciucuri |
| 40 | Red Bull | Băutură |

Iar SPONSOR are această formă:

****  
  
**10 Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare table neasociativ; minimum 10 înregistrări în tabelele asociative). + 13. Secvente  
  
  
CREATE TABLE SPONSOR**(

cod\_sponsor number(5) PRIMARY KEY,

nume varchar2(46),

suma number(5)

);

create sequence seq\_sponsor

start with 9

increment by 1

nomaxvalue

nocycle

nocache;

INSERT INTO SPONSOR VALUES(seq\_sponsor.nextval, 'Shark', 20000);

INSERT INTO SPONSOR VALUES(seq\_sponsor.nextval, 'Liqui Moly', 30000);

INSERT INTO SPONSOR VALUES(seq\_sponsor.nextval, 'Monster Energy', 40000);

INSERT INTO SPONSOR VALUES(seq\_sponsor.nextval, 'Petronas', 45000);

INSERT INTO SPONSOR VALUES(seq\_sponsor.nextval, 'Red Bull', 1000);

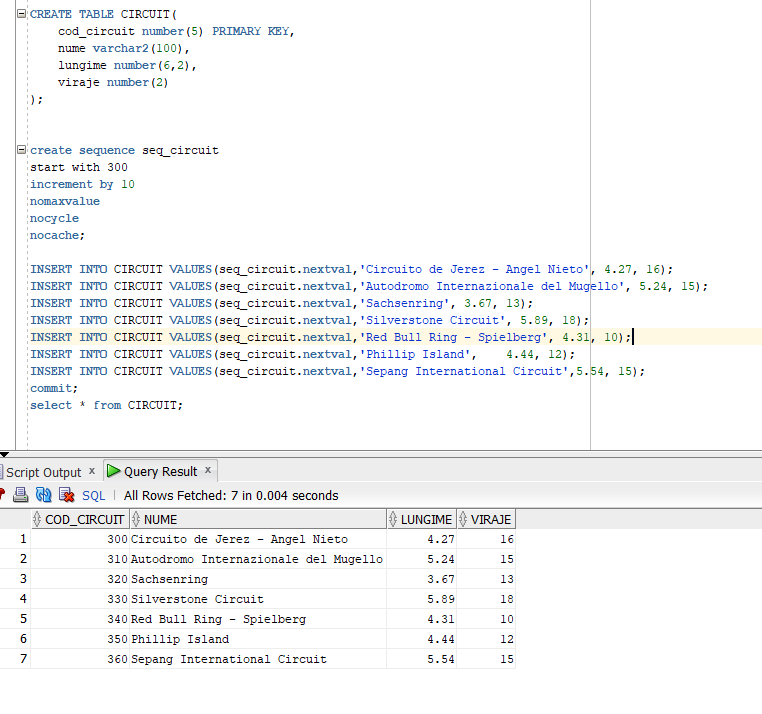
INSERT INTO SPONSOR VALUES(seq\_sponsor.nextval, 'Tissot', 60000);

INSERT INTO SPONSOR VALUES(seq\_sponsor.nextval, 'Motul', 80000);

INSERT INTO SPONSOR VALUES(seq\_sponsor.nextval, 'Michelin', 3000);

**CREATE TABLE CIRCUIT**(

cod\_circuit number(5) PRIMARY KEY,

 nume varchar2(100),

lungime number(6,2),

viraje number(2)

);

create sequence seq\_circuit

start with 300

increment by 10

nomaxvalue

nocycle

nocache;

INSERT INTO CIRCUIT VALUES(seq\_circuit.nextval,'Circuito de Jerez - Angel Nieto', 4.27, 16);

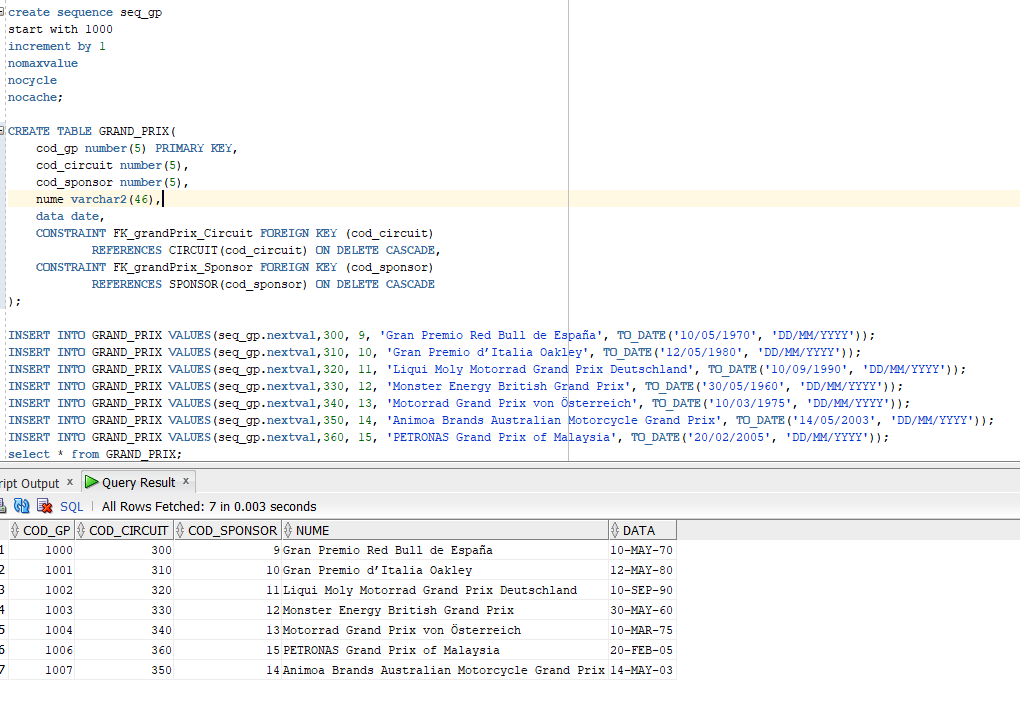
INSERT INTO CIRCUIT VALUES(seq\_circuit.nextval,'Autodromo Internazionale del Mugello', 5.24, 15);

INSERT INTO CIRCUIT VALUES(seq\_circuit.nextval,'Sachsenring', 3.67, 13);

INSERT INTO CIRCUIT VALUES(seq\_circuit.nextval,'Silverstone Circuit', 5.89, 18);

INSERT INTO CIRCUIT VALUES(seq\_circuit.nextval,'Red Bull Ring - Spielberg', 4.31, 10);

INSERT INTO CIRCUIT VALUES(seq\_circuit.nextval,'Phillip Island', 4.44, 12);

****INSERT INTO CIRCUIT VALUES(seq\_circuit.nextval,'Sepang International Circuit',5.54, 15);

**CREATE TABLE GRAND\_PRIX**(

cod\_gp number(5) PRIMARY KEY,

cod\_circuit number(5),

cod\_sponsor number(5),

nume varchar2(46),

data date,

CONSTRAINT FK\_grandPrix\_Circuit FOREIGN KEY (cod\_circuit)

REFERENCES CIRCUIT(cod\_circuit) ON DELETE CASCADE,

CONSTRAINT FK\_grandPrix\_Sponsor FOREIGN KEY (cod\_sponsor)

REFERENCES SPONSOR(cod\_sponsor) ON DELETE CASCADE

);

create sequence seq\_gp

start with 1000

increment by 1

nomaxvalue

nocycle

nocache;

INSERT INTO GRAND\_PRIX VALUES(seq\_gp.nextval,300, 9, 'Gran Premio Red Bull de España', TO\_DATE('10/05/1970', 'DD/MM/YYYY'));

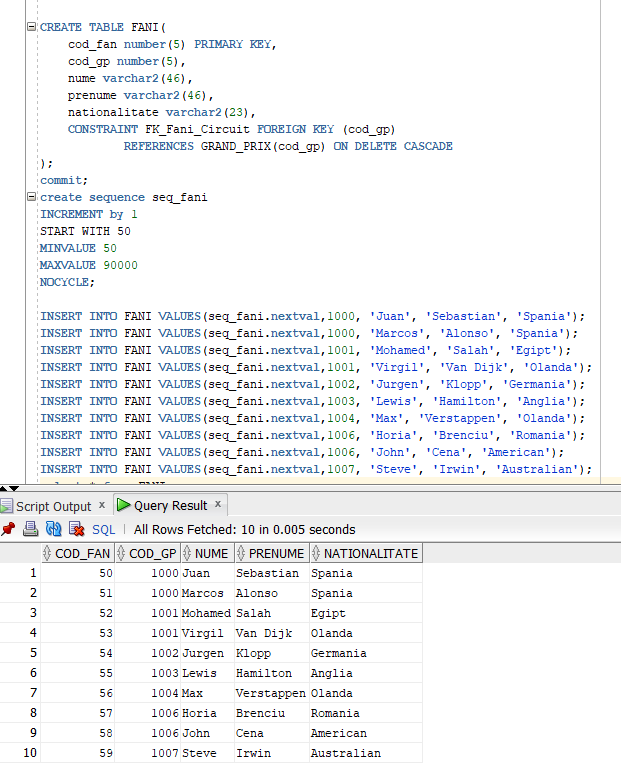
INSERT INTO GRAND\_PRIX VALUES(seq\_gp.nextval,310, 10, 'Gran Premio d’Italia Oakley', TO\_DATE('12/05/1980', 'DD/MM/YYYY'));

INSERT INTO GRAND\_PRIX VALUES(seq\_gp.nextval,320, 11, 'Liqui Moly Motorrad Grand Prix Deutschland', TO\_DATE('10/09/1990', 'DD/MM/YYYY'));

INSERT INTO GRAND\_PRIX VALUES(seq\_gp.nextval,330, 12, 'Monster Energy British Grand Prix', TO\_DATE('30/05/1960', 'DD/MM/YYYY'));

INSERT INTO GRAND\_PRIX VALUES(seq\_gp.nextval,340, 13, 'Motorrad Grand Prix von Österreich', TO\_DATE('10/03/1975', 'DD/MM/YYYY'));

INSERT INTO GRAND\_PRIX VALUES(seq\_gp.nextval,350, 14, 'Animoa Brands Australian Motorcycle Grand Prix', TO\_DATE('14/05/2003', 'DD/MM/YYYY'));

INSERT INTO GRAND\_PRIX VALUES(seq\_gp.nextval,360, 15, 'PETRONAS Grand Prix of Malaysia', TO\_DATE('20/02/2005', 'DD/MM/YYYY'));  
  
**CREATE TABLE FANI**(

cod\_fan number(5) PRIMARY KEY,

cod\_gp number(5),

nume varchar2(46),

prenume varchar2(46),

nationalitate varchar2(23),

CONSTRAINT FK\_Fani\_Circuit FOREIGN KEY (cod\_gp)

REFERENCES GRAND\_PRIX(cod\_gp) ON DELETE CASCADE

);

commit;

create sequence seq\_fani

INCREMENT by 1

START WITH 50

MINVALUE 50

MAXVALUE 90000

NOCYCLE;

INSERT INTO FANI VALUES(seq\_fani.nextval,1000, 'Juan', 'Sebastian', 'Spania');

INSERT INTO FANI VALUES(seq\_fani.nextval,1000, 'Marcos', 'Alonso', 'Spania');

INSERT INTO FANI VALUES(seq\_fani.nextval,1001, 'Mohamed', 'Salah', 'Egipt');

INSERT INTO FANI VALUES(seq\_fani.nextval,1001, 'Virgil', 'Van Dijk', 'Olanda');

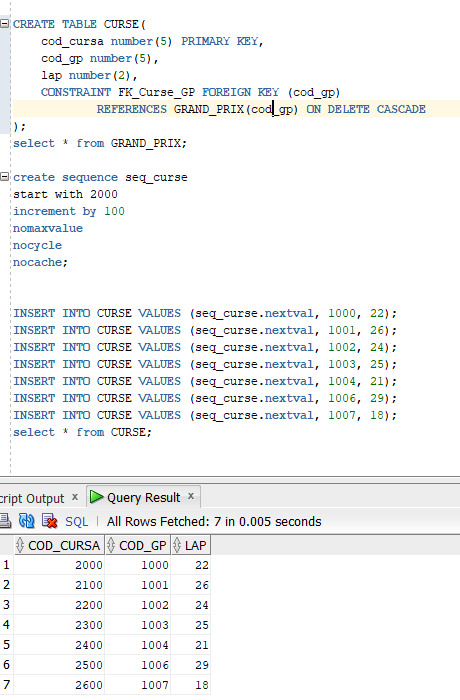
INSERT INTO FANI VALUES(seq\_fani.nextval,1002, 'Jurgen', 'Klopp', 'Germania');

INSERT INTO FANI VALUES(seq\_fani.nextval,1003, 'Lewis', 'Hamilton', 'Anglia');

INSERT INTO FANI VALUES(seq\_fani.nextval,1004, 'Max', 'Verstappen', 'Olanda');

INSERT INTO FANI VALUES(seq\_fani.nextval,1006, 'Horia', 'Brenciu', 'Romania');

INSERT INTO FANI VALUES(seq\_fani.nextval,1006, 'John', 'Cena', 'American');

INSERT INTO FANI VALUES(seq\_fani.nextval,1007, 'Steve', 'Irwin', 'Australian');  
  
  
**CREATE TABLE CURSE**(

cod\_cursa number(5) PRIMARY KEY,

cod\_gp number(5),

lap number(2),

CONSTRAINT FK\_Curse\_GP FOREIGN KEY (cod\_gp)

REFERENCES GRAND\_PRIX(cod\_gp) ON DELETE CASCADE

);

select \* from GRAND\_PRIX;

create sequence seq\_curse

start with 2000

increment by 100

nomaxvalue

nocycle

nocache;

INSERT INTO CURSE VALUES (seq\_curse.nextval, 1000, 22);

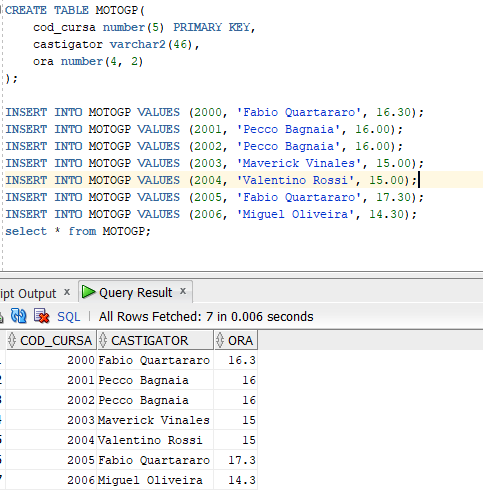
INSERT INTO CURSE VALUES (seq\_curse.nextval, 1001, 26);

INSERT INTO CURSE VALUES (seq\_curse.nextval, 1002, 24);

INSERT INTO CURSE VALUES (seq\_curse.nextval, 1003, 25);

INSERT INTO CURSE VALUES (seq\_curse.nextval, 1004, 21);

INSERT INTO CURSE VALUES (seq\_curse.nextval, 1006, 29);

INSERT INTO CURSE VALUES (seq\_curse.nextval, 1007, 18);  
  
  
**CREATE TABLE MOTOGP**(

cod\_cursa number(5) PRIMARY KEY,

castigator varchar2(46),

ora number(4, 2)

);

INSERT INTO MOTOGP VALUES (2000, 'Fabio Quartararo', 16.30);

INSERT INTO MOTOGP VALUES (2001, 'Pecco Bagnaia', 16.00);

INSERT INTO MOTOGP VALUES (2002, 'Pecco Bagnaia', 16.00);

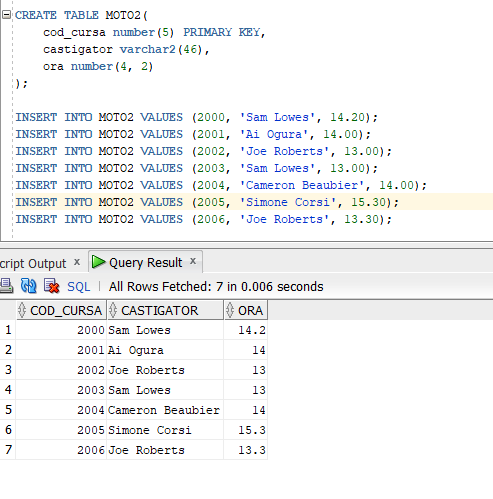
INSERT INTO MOTOGP VALUES (2003, 'Maverick Vinales', 15.00);

INSERT INTO MOTOGP VALUES (2004, 'Valentino Rossi', 15.00);

INSERT INTO MOTOGP VALUES (2005, 'Fabio Quartararo', 17.30);

INSERT INTO MOTOGP VALUES (2006, 'Miguel Oliveira', 14.30);

select \* from MOTOGP;

**CREATE TABLE MOTO2**(

cod\_cursa number(5) PRIMARY KEY,

castigator varchar2(46),

ora number(4, 2)

);

INSERT INTO MOTO2 VALUES (2000, 'Sam Lowes', 14.20);

INSERT INTO MOTO2 VALUES (2001, 'Ai Ogura', 14.00);

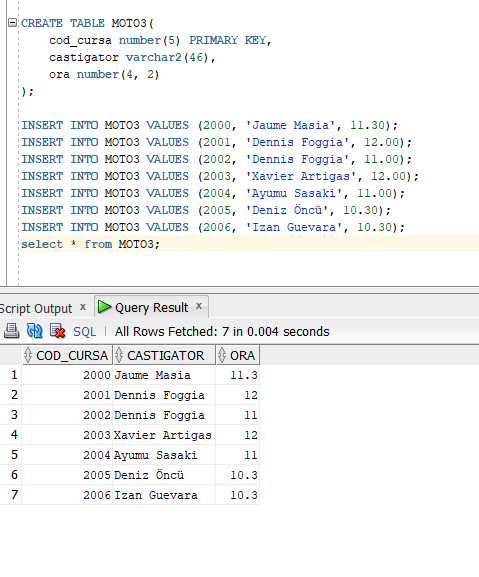
INSERT INTO MOTO2 VALUES (2002, 'Joe Roberts', 13.00);

INSERT INTO MOTO2 VALUES (2003, 'Sam Lowes', 13.00);

INSERT INTO MOTO2 VALUES (2004, 'Cameron Beaubier', 14.00);

INSERT INTO MOTO2 VALUES (2005, 'Simone Corsi', 15.30);

INSERT INTO MOTO2 VALUES (2006, 'Joe Roberts', 13.30);

select \* from MOTO2;  
  
  
  
  
**CREATE TABLE MOTO3**(

cod\_cursa number(5) PRIMARY KEY,

castigator varchar2(46),

ora number(4, 2)

);

INSERT INTO MOTO3 VALUES (2000, 'Jaume Masia', 11.30);

INSERT INTO MOTO3 VALUES (2001, 'Dennis Foggia', 12.00);

INSERT INTO MOTO3 VALUES (2002, 'Dennis Foggia', 11.00);

INSERT INTO MOTO3 VALUES (2003, 'Xavier Artigas', 12.00);

INSERT INTO MOTO3 VALUES (2004, 'Ayumu Sasaki', 11.00);

INSERT INTO MOTO3 VALUES (2005, 'Deniz Öncü', 10.30);

INSERT INTO MOTO3 VALUES (2006, 'Izan Guevara', 10.30);

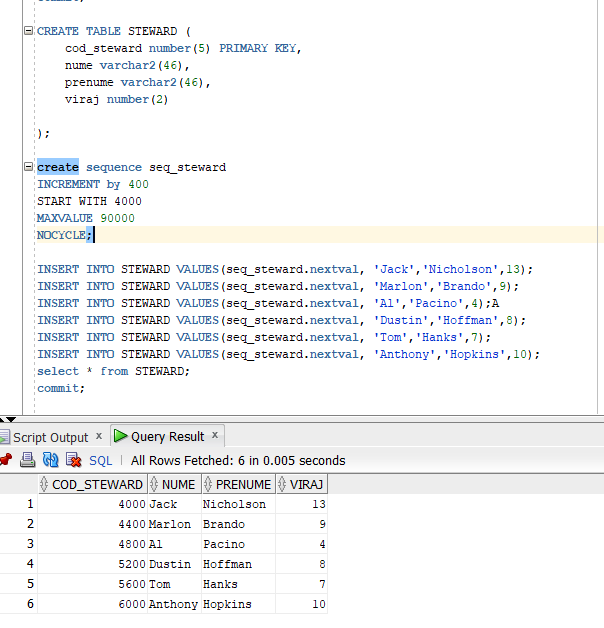
select \* from MOTO3;  
  
  
  
**CREATE TABLE STEWARD** (

cod\_steward number(5) PRIMARY KEY,

nume varchar2(46),

prenume varchar2(46),

viraj number(2)



);

create sequence seq\_steward

INCREMENT by 400

START WITH 4000

MAXVALUE 90000

NOCYCLE;

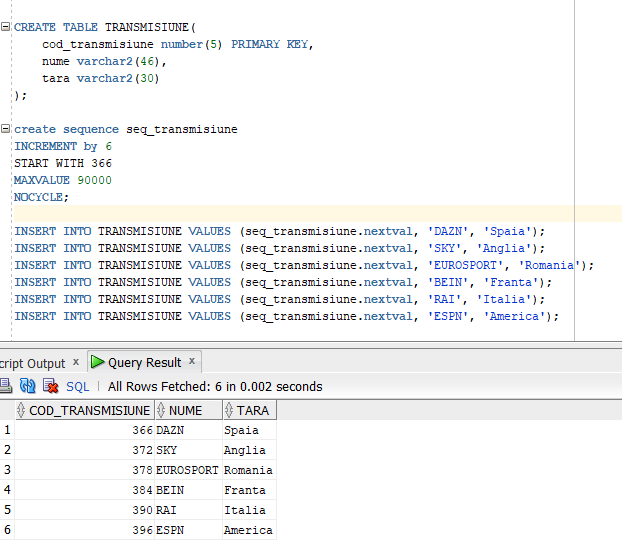
INSERT INTO STEWARD VALUES(seq\_steward.nextval, 'Jack','Nicholson',13);

INSERT INTO STEWARD VALUES(seq\_steward.nextval, 'Marlon','Brando',9);

INSERT INTO STEWARD VALUES(seq\_steward.nextval, 'Al','Pacino',4);

INSERT INTO STEWARD VALUES(seq\_steward.nextval, 'Dustin','Hoffman',8);

INSERT INTO STEWARD VALUES(seq\_steward.nextval, 'Tom','Hanks',7);

INSERT INTO STEWARD VALUES(seq\_steward.nextval, 'Anthony','Hopkins',10);  
  
  
**CREATE TABLE TRANSMISIUNE**(

cod\_transmisiune number(5) PRIMARY KEY,

nume varchar2(46),

tara varchar2(30)

);

create sequence seq\_transmisiune

INCREMENT by 6

START WITH 366

MAXVALUE 90000

NOCYCLE;

INSERT INTO TRANSMISIUNE VALUES (seq\_transmisiune.nextval, 'DAZN', 'Spaia');

INSERT INTO TRANSMISIUNE VALUES (seq\_transmisiune.nextval, 'SKY', 'Anglia');

INSERT INTO TRANSMISIUNE VALUES (seq\_transmisiune.nextval, 'EUROSPORT', 'Romania');

INSERT INTO TRANSMISIUNE VALUES (seq\_transmisiune.nextval, 'BEIN', 'Franta');

INSERT INTO TRANSMISIUNE VALUES (seq\_transmisiune.nextval, 'RAI', 'Italia');

INSERT INTO TRANSMISIUNE VALUES (seq\_transmisiune.nextval, 'ESPN', 'America');  
  
**CREATE TABLE ABONAMENT**(

cod\_transmisiune number(5) PRIMARY KEY,

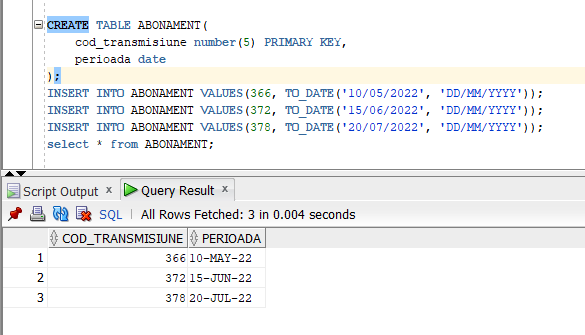
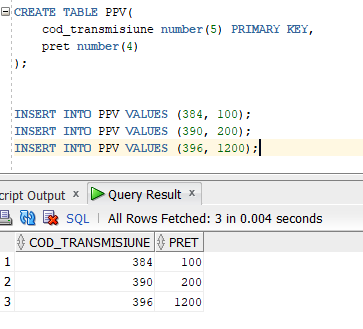
perioada date

);

INSERT INTO ABONAMENT VALUES(366, TO\_DATE('10/05/2022', 'DD/MM/YYYY'));

INSERT INTO ABONAMENT VALUES(372, TO\_DATE('15/06/2022', 'DD/MM/YYYY'));

INSERT INTO ABONAMENT VALUES(378, TO\_DATE('20/07/2022', 'DD/MM/YYYY'));

select \* from ABONAMENT;  
  
  
  
  
CREATE TABLE PPV(

cod\_transmisiune number(5) PRIMARY KEY,

pret number(4)

);

INSERT INTO PPV VALUES (384, 100);

INSERT INTO PPV VALUES (390, 200);

INSERT INTO PPV VALUES (396, 1200);  
  
  
  
**CREATE TABLE ECHIPE**(

cod\_echipa number(5) PRIMARY KEY,

nume varchar2(100)

);

create sequence seq\_echipe

start with 7000

increment by 20

nomaxvalue

nocycle

nocache;

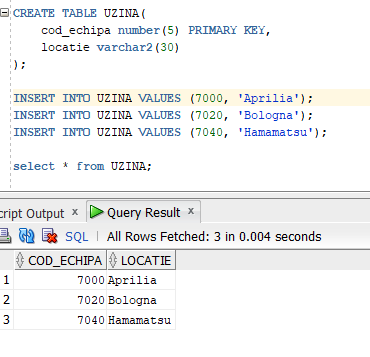
INSERT INTO ECHIPE VALUES(seq\_echipe.nextval, 'Aprilia Racing');

INSERT INTO ECHIPE VALUES(seq\_echipe.nextval, 'Ducati Lenovo Team');

INSERT INTO ECHIPE VALUES(seq\_echipe.nextval, 'Monster Energy Yamaha MotoGP');

INSERT INTO ECHIPE VALUES(seq\_echipe.nextval, 'Gresini Racing MotoGP');

INSERT INTO ECHIPE VALUES(seq\_echipe.nextval, 'Tech3 KTM Factory Racing');

INSERT INTO ECHIPE VALUES(seq\_echipe.nextval, 'LCR Honda');  
  
**CREATE TABLE UZINA**(

cod\_echipa number(5) PRIMARY KEY,

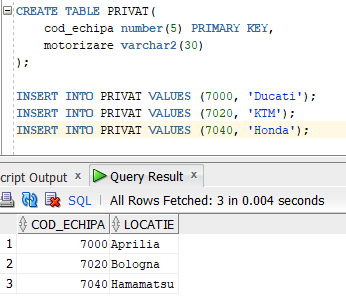
locatie varchar2(30)

);

INSERT INTO UZINA VALUES (7000, 'Aprilia');

INSERT INTO UZINA VALUES (7020, 'Bologna');

INSERT INTO UZINA VALUES (7040, 'Hamamatsu');

select \* from UZINA;  
  
  
**CREATE TABLE PRIVAT**(

cod\_echipa number(5) PRIMARY KEY,

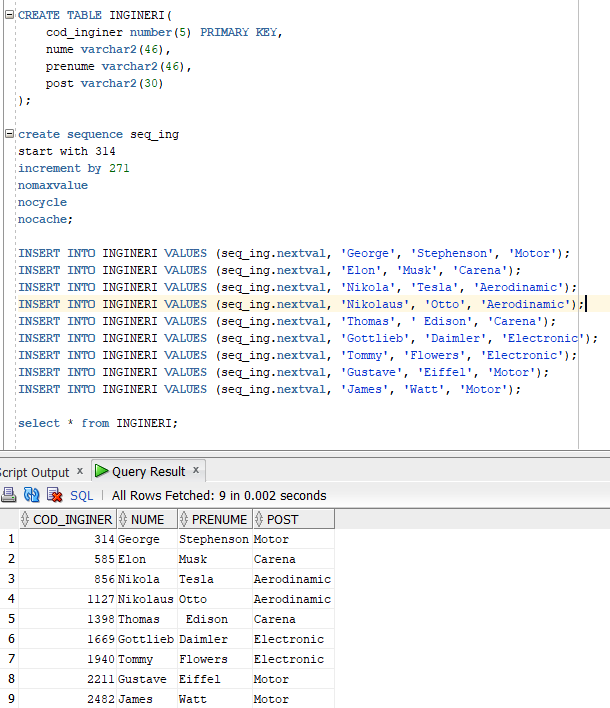
motorizare varchar2(30)

);

INSERT INTO PRIVAT VALUES (7000, 'Ducati');

INSERT INTO PRIVAT VALUES (7020, 'KTM');

INSERT INTO PRIVAT VALUES (7040, 'Honda');  
  
  
  
  
**CREATE TABLE INGINERI**(

 cod\_inginer number(5) PRIMARY KEY,

nume varchar2(46),

prenume varchar2(46),

post varchar2(30)

);

create sequence seq\_ing

start with 314

increment by 271

nomaxvalue

nocycle

nocache;

INSERT INTO INGINERI VALUES (seq\_ing.nextval, 'George', 'Stephenson', 'Motor');

INSERT INTO INGINERI VALUES (seq\_ing.nextval, 'Elon', 'Musk', 'Carena');

INSERT INTO INGINERI VALUES (seq\_ing.nextval, 'Nikola', 'Tesla', 'Aerodinamic');

INSERT INTO INGINERI VALUES (seq\_ing.nextval, 'Nikolaus', 'Otto', 'Aerodinamic');

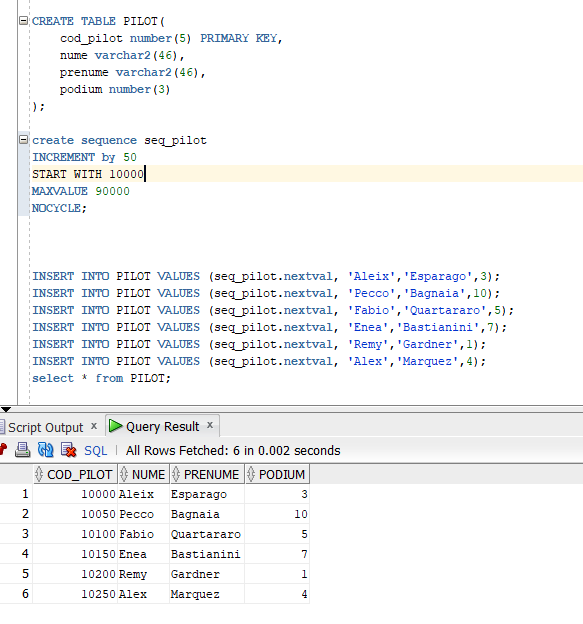
INSERT INTO INGINERI VALUES (seq\_ing.nextval, 'Thomas', ' Edison', 'Carena');

INSERT INTO INGINERI VALUES (seq\_ing.nextval, 'Gottlieb', 'Daimler', 'Electronic');

INSERT INTO INGINERI VALUES (seq\_ing.nextval, 'Tommy', 'Flowers', 'Electronic');

INSERT INTO INGINERI VALUES (seq\_ing.nextval, 'Gustave', 'Eiffel', 'Motor');

INSERT INTO INGINERI VALUES (seq\_ing.nextval, 'James', 'Watt', 'Motor');  
  
**CREATE TABLE PILOT**(

 cod\_pilot number(5) PRIMARY KEY,

prenume varchar2(46),  
  
 nume varchar2(46),

podium number(3)

);

create sequence seq\_pilot

INCREMENT by 50

START WITH 10000

MAXVALUE 90000

NOCYCLE;

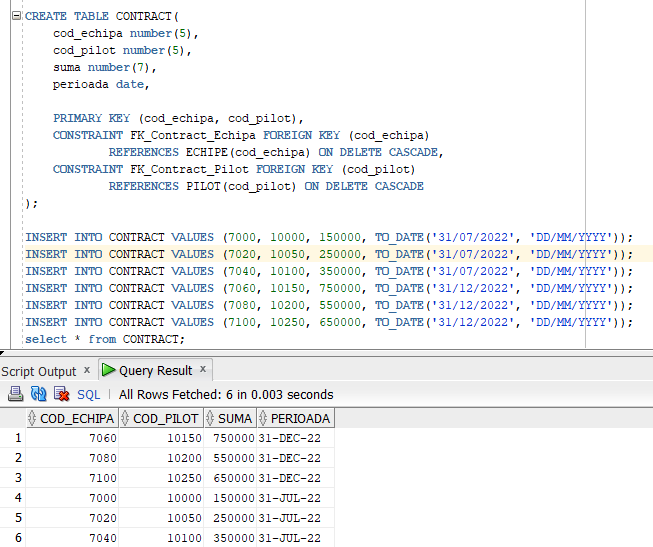
INSERT INTO PILOT VALUES (seq\_pilot.nextval, 'Aleix','Esparago',3);

INSERT INTO PILOT VALUES (seq\_pilot.nextval, 'Pecco','Bagnaia',10);

INSERT INTO PILOT VALUES (seq\_pilot.nextval, 'Fabio','Quartararo',5);

INSERT INTO PILOT VALUES (seq\_pilot.nextval, 'Enea','Bastianini',7);

INSERT INTO PILOT VALUES (seq\_pilot.nextval, 'Remy','Gardner',1);

INSERT INTO PILOT VALUES (seq\_pilot.nextval, 'Alex','Marquez',4);  
  
**CREATE TABLE CONTRACT**(

cod\_echipa number(5),

cod\_pilot number(5),

suma number(7),

perioada date,

PRIMARY KEY (cod\_echipa, cod\_pilot),

CONSTRAINT FK\_Contract\_Echipa FOREIGN KEY (cod\_echipa)

REFERENCES ECHIPE(cod\_echipa) ON DELETE CASCADE,

CONSTRAINT FK\_Contract\_Pilot FOREIGN KEY (cod\_pilot)

REFERENCES PILOT(cod\_pilot) ON DELETE CASCADE

);

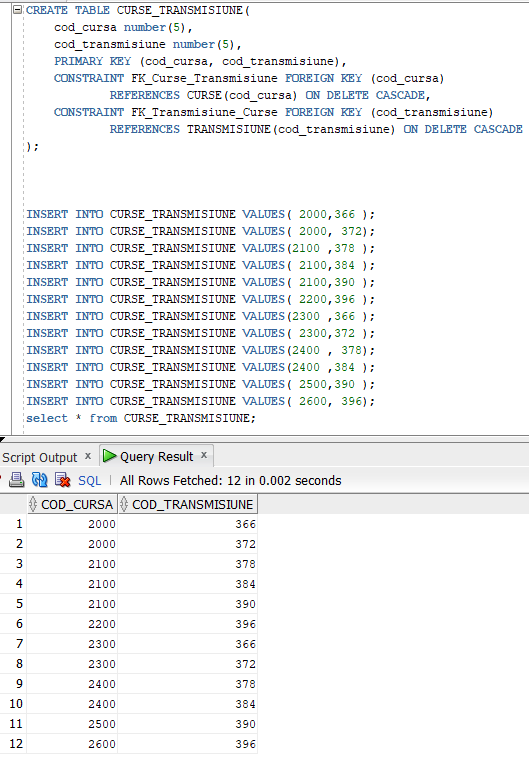
INSERT INTO CONTRACT VALUES (7000, 10000, 150000, TO\_DATE('31/07/2022', 'DD/MM/YYYY'));

INSERT INTO CONTRACT VALUES (7020, 10050, 250000, TO\_DATE('31/07/2022', 'DD/MM/YYYY'));

INSERT INTO CONTRACT VALUES (7040, 10100, 350000, TO\_DATE('31/07/2022', 'DD/MM/YYYY'));

INSERT INTO CONTRACT VALUES (7060, 10150, 750000, TO\_DATE('31/12/2022', 'DD/MM/YYYY'));

INSERT INTO CONTRACT VALUES (7080, 10200, 550000, TO\_DATE('31/12/2022', 'DD/MM/YYYY'));

INSERT INTO CONTRACT VALUES (7100, 10250, 650000, TO\_DATE('31/12/2022', 'DD/MM/YYYY'));  
  
  
  
**CREATE TABLE CURSE\_TRANSMISIUNE**(

cod\_cursa number(5),

cod\_transmisiune number(5),

PRIMARY KEY (cod\_cursa, cod\_transmisiune),

CONSTRAINT FK\_Curse\_Transmisiune FOREIGN KEY (cod\_cursa)

REFERENCES CURSE(cod\_cursa) ON DELETE CASCADE,

CONSTRAINT FK\_Transmisiune\_Curse FOREIGN KEY (cod\_transmisiune)

REFERENCES TRANSMISIUNE(cod\_transmisiune) ON DELETE CASCADE

);

INSERT INTO CURSE\_TRANSMISIUNE VALUES( 2000,366 );

INSERT INTO CURSE\_TRANSMISIUNE VALUES( 2000, 372);

INSERT INTO CURSE\_TRANSMISIUNE VALUES(2100 ,378 );

INSERT INTO CURSE\_TRANSMISIUNE VALUES( 2100,384 );

INSERT INTO CURSE\_TRANSMISIUNE VALUES( 2100,390 );

INSERT INTO CURSE\_TRANSMISIUNE VALUES( 2200,396 );

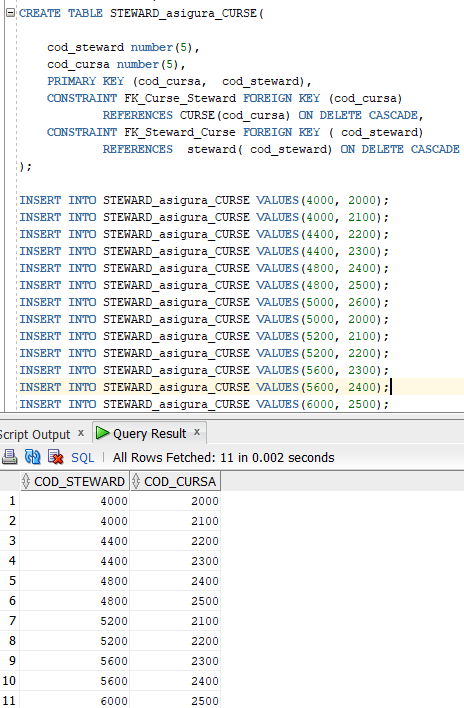
INSERT INTO CURSE\_TRANSMISIUNE VALUES(2300 ,366 );

INSERT INTO CURSE\_TRANSMISIUNE VALUES( 2300,372 );

INSERT INTO CURSE\_TRANSMISIUNE VALUES(2400 , 378);

INSERT INTO CURSE\_TRANSMISIUNE VALUES(2400 ,384 );

INSERT INTO CURSE\_TRANSMISIUNE VALUES( 2500,390 );

INSERT INTO CURSE\_TRANSMISIUNE VALUES( 2600, 396);  
  
  
**CREATE TABLE STEWARD\_asigura\_CURSE**(

cod\_steward number(5),

cod\_cursa number(5),

PRIMARY KEY (cod\_cursa, cod\_steward),

CONSTRAINT FK\_Curse\_Steward FOREIGN KEY (cod\_cursa)

REFERENCES CURSE(cod\_cursa) ON DELETE CASCADE,

CONSTRAINT FK\_Steward\_Curse FOREIGN KEY ( cod\_steward)

REFERENCES steward( cod\_steward) ON DELETE CASCADE

);

INSERT INTO STEWARD\_asigura\_CURSE VALUES(4000, 2000);

INSERT INTO STEWARD\_asigura\_CURSE VALUES(4000, 2100);

INSERT INTO STEWARD\_asigura\_CURSE VALUES(4400, 2200);

INSERT INTO STEWARD\_asigura\_CURSE VALUES(4400, 2300);

INSERT INTO STEWARD\_asigura\_CURSE VALUES(4800, 2400);

INSERT INTO STEWARD\_asigura\_CURSE VALUES(4800, 2500);

INSERT INTO STEWARD\_asigura\_CURSE VALUES(5000, 2600);

INSERT INTO STEWARD\_asigura\_CURSE VALUES(5000, 2000);

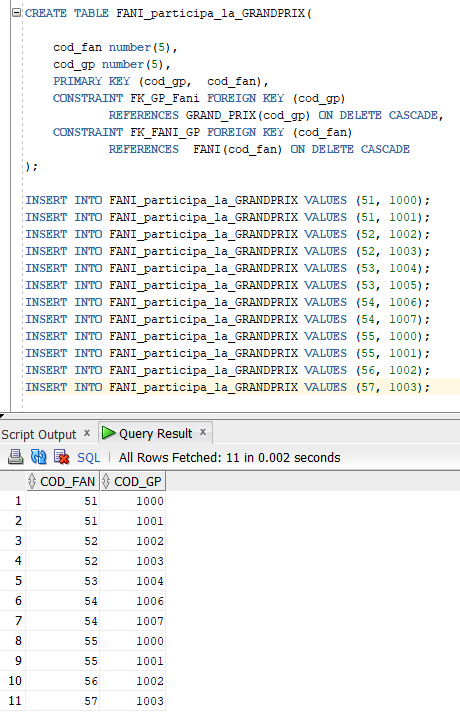
INSERT INTO STEWARD\_asigura\_CURSE VALUES(5200, 2100);

INSERT INTO STEWARD\_asigura\_CURSE VALUES(5200, 2200);

INSERT INTO STEWARD\_asigura\_CURSE VALUES(5600, 2300);

INSERT INTO STEWARD\_asigura\_CURSE VALUES(5600, 2400);

INSERT INTO STEWARD\_asigura\_CURSE VALUES(6000, 2500);  
  
**CREATE TABLE FANI\_participa\_la\_GRANDPRIX**(



cod\_fan number(5),

cod\_gp number(5),

PRIMARY KEY (cod\_gp, cod\_fan),

CONSTRAINT FK\_GP\_Fani FOREIGN KEY (cod\_gp) REFERENCES GRAND\_PRIX(cod\_gp) ON DELETE CASCADE,

CONSTRAINT FK\_FANI\_GP FOREIGN KEY (cod\_fan) REFERENCES FANI(cod\_fan) ON DELETE CASCADE);

INSERT INTO FANI\_participa\_la\_GRANDPRIX VALUES (51, 1000);

INSERT INTO FANI\_participa\_la\_GRANDPRIX VALUES (51, 1001);

INSERT INTO FANI\_participa\_la\_GRANDPRIX VALUES (52, 1002);

INSERT INTO FANI\_participa\_la\_GRANDPRIX VALUES (52, 1003);

INSERT INTO FANI\_participa\_la\_GRANDPRIX VALUES (53, 1004);

INSERT INTO FANI\_participa\_la\_GRANDPRIX VALUES (53, 1005);

INSERT INTO FANI\_participa\_la\_GRANDPRIX VALUES (54, 1006);

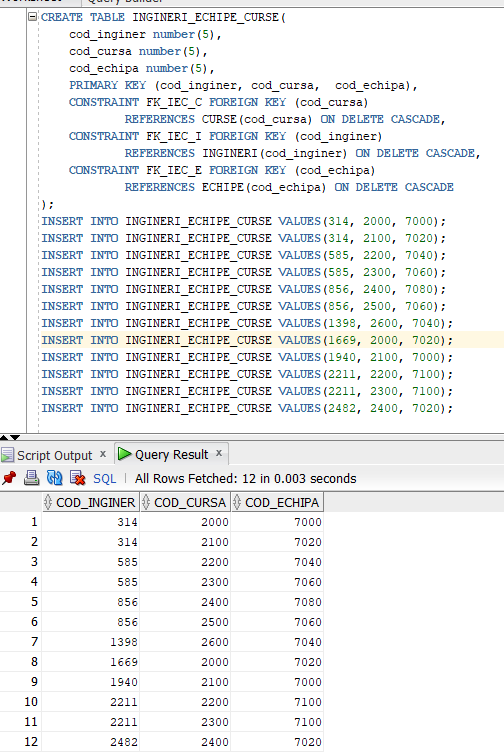
INSERT INTO FANI\_participa\_la\_GRANDPRIX VALUES (54, 1007);

INSERT INTO FANI\_participa\_la\_GRANDPRIX VALUES (55, 1000);

INSERT INTO FANI\_participa\_la\_GRANDPRIX VALUES (55, 1001);

INSERT INTO FANI\_participa\_la\_GRANDPRIX VALUES (56, 1002);

INSERT INTO FANI\_participa\_la\_GRANDPRIX VALUES (57, 1003);

**CREATE TABLE INGINERI\_ECHIPE\_CURSE**(

cod\_inginer number(5),

cod\_cursa number(5),

cod\_echipa number(5),

PRIMARY KEY (cod\_inginer, cod\_cursa, cod\_echipa),

CONSTRAINT FK\_IEC\_C FOREIGN KEY (cod\_cursa)

REFERENCES CURSE(cod\_cursa) ON DELETE CASCADE,

CONSTRAINT FK\_IEC\_I FOREIGN KEY (cod\_inginer)

REFERENCES INGINERI(cod\_inginer) ON DELETE CASCADE,

CONSTRAINT FK\_IEC\_E FOREIGN KEY (cod\_echipa)

REFERENCES ECHIPE(cod\_echipa) ON DELETE CASCADE

);

INSERT INTO INGINERI\_ECHIPE\_CURSE VALUES(314, 2000, 7000);

INSERT INTO INGINERI\_ECHIPE\_CURSE VALUES(314, 2100, 7020);

INSERT INTO INGINERI\_ECHIPE\_CURSE VALUES(585, 2200, 7040);

INSERT INTO INGINERI\_ECHIPE\_CURSE VALUES(585, 2300, 7060);

INSERT INTO INGINERI\_ECHIPE\_CURSE VALUES(856, 2400, 7080);

INSERT INTO INGINERI\_ECHIPE\_CURSE VALUES(856, 2500, 7060);

INSERT INTO INGINERI\_ECHIPE\_CURSE VALUES(1398, 2600, 7040);

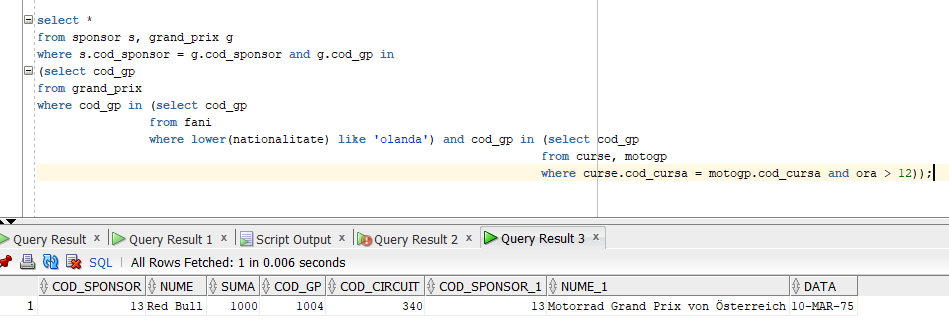
INSERT INTO INGINERI\_ECHIPE\_CURSE VALUES(1669, 2000, 7020);

INSERT INTO INGINERI\_ECHIPE\_CURSE VALUES(1940, 2100, 7000);

INSERT INTO INGINERI\_ECHIPE\_CURSE VALUES(2211, 2200, 7100);

INSERT INTO INGINERI\_ECHIPE\_CURSE VALUES(2211, 2300, 7100);

INSERT INTO INGINERI\_ECHIPE\_CURSE VALUES(2482, 2400, 7020);

**11 5 CERERI COMPLEXE SQL**  
  
**1.SĂ SE SELECTEZE SPONSORII GRAND PRIX-URILOR CARE AU CURSA DE MOTOGP CE ÎNCEPE DUPĂ 12.00 ȘI ARE FANI DIN OLANDA(FILTRARE LA NIVEL DE LINII, SUBCERERE NESINCRONIZATĂ, FUNCȚIE SIR)**

select \*

from sponsor s, grand\_prix g

where s.cod\_sponsor = g.cod\_sponsor and g.cod\_gp in

(select cod\_gp

from grand\_prix

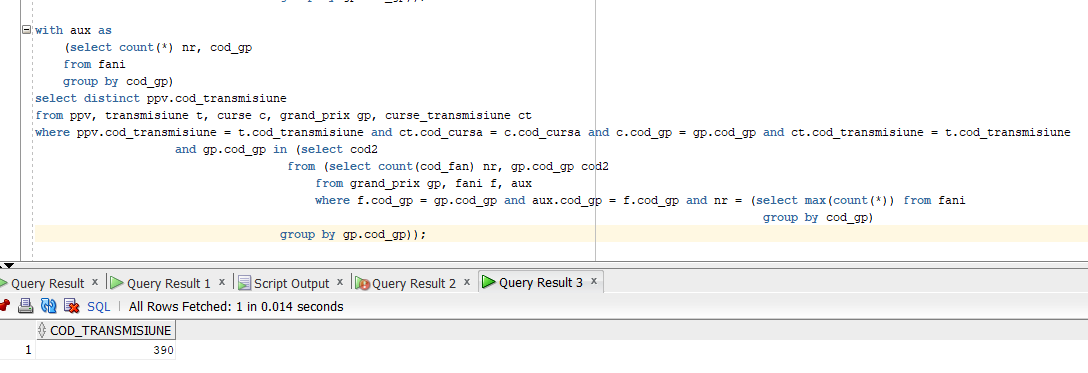
where cod\_gp in (select cod\_gp

from fani

where lower(nationalitate) like 'olanda') and cod\_gp in (select cod\_gp

from curse, motogp

where curse.cod\_cursa = motogp.cod\_cursa and ora > 12));  
  
  
  
  
  
  
  
  
  
  
**2.SĂ SE SELECTEZE TRANSMISIUNILE PPV ALE CURSELOR CE AU AVUT CEL MAI MARE NUMĂR DE FANI(GRUPĂRI DE DATE, FUNCȚII GRUP, CLAUZA WITH, JOIN PE 4 TABELE)**  
  
with aux as

 (select count(\*) nr, cod\_gp

from fani

group by cod\_gp)

select distinct ppv.cod\_transmisiune

from ppv, transmisiune t, curse c, grand\_prix gp, curse\_transmisiune ct

where ppv.cod\_transmisiune = t.cod\_transmisiune and ct.cod\_cursa = c.cod\_cursa and c.cod\_gp = gp.cod\_gp and ct.cod\_transmisiune = t.cod\_transmisiune

and gp.cod\_gp in (select cod2

from (select count(cod\_fan) nr, gp.cod\_gp cod2

from grand\_prix gp, fani f, aux

where f.cod\_gp = gp.cod\_gp and aux.cod\_gp = f.cod\_gp and nr = (select max(count(\*)) from fani

group by cod\_gp)

group by gp.cod\_gp));  
  
  
  
  
  
  
  
  
  
  
  
  
  
**3.SĂ SE SELECTEZE CURSELE CARE AU RAPORTUL LUNGIME/LAP MAI MARE DECÂT MEDIA ȘI NUMĂR IMPAR DE VOCALE ÎN NUME  
(NVL, CERERE SINCRONIZATĂ PE 3 TABELE)**

select c.cod\_cursa, cr.nume

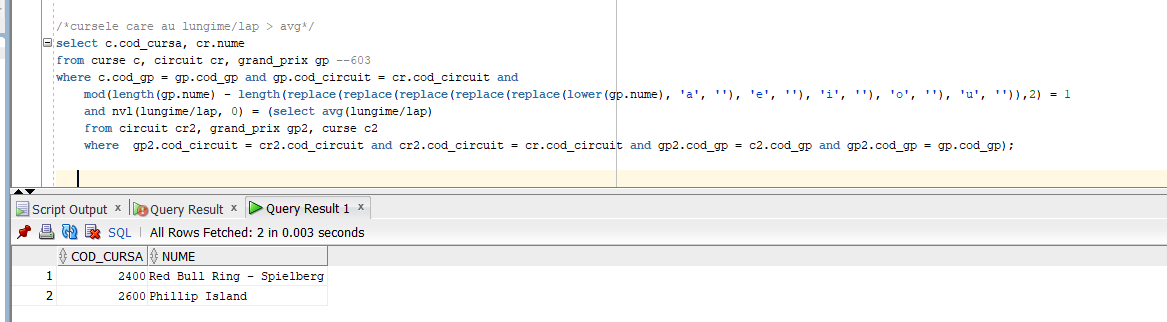
from curse c, circuit cr, grand\_prix gp --603

where c.cod\_gp = gp.cod\_gp and gp.cod\_circuit = cr.cod\_circuit and

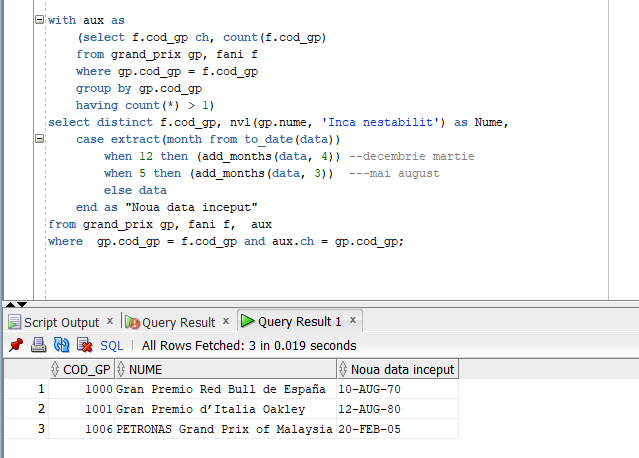
mod(length(gp.nume) - length(replace(replace(replace(replace(replace(lower(gp.nume), 'a', ''), 'e', ''), 'i', ''), 'o', ''), 'u', '')),2) = 1

and nvl(lungime/lap, 0) = (select avg(lungime/lap)

from circuit cr2, grand\_prix gp2, curse c2

 where gp2.cod\_circuit = cr2.cod\_circuit and cr2.cod\_circuit = cr.cod\_circuit and gp2.cod\_gp = c2.cod\_gp and gp2.cod\_gp = gp.cod\_gp);

**4.SĂ SE SELECTEZE GRAND PRIX-URILE CARE NUMĂRUL DE FANI MAI MARE DECÂT 1, CELE CARE SE DESFĂȘOARĂ ÎN DECEMBRIE VOR FI AMÂNATE PENTRU MARTIE; IAR CELE DIN MAI PENTRU AUGUST  
(CASE, FILTRARE LA NIVEL DE GRUPURI, FUNCȚII CALENDARISTICE )**

with aux as

(select f.cod\_gp ch, count(f.cod\_gp)

from grand\_prix gp, fani f

where gp.cod\_gp = f.cod\_gp

group by gp.cod\_gp

having count(\*) > 1)

select distinct f.cod\_gp, nvl(gp.nume, 'Inca nestabilit') as Nume,

case extract(month from to\_date(data))

when 12 then (add\_months(data, 4)) --decembrie martie

when 5 then (add\_months(data, 3)) ---mai august

else data

end as "Data modificata"

from grand\_prix gp, fani f, aux

where gp.cod\_gp = f.cod\_gp and aux.ch = gp.cod\_gp;

**5.SĂ SE NUMERE CÂȚI FANI ARE FIECARE ȚARĂ(DECODE, FUNCȚIE ȘIR)**  
  
  
select nationalitate ,

decode(nationalitate, 'Spania', (select count(f.cod\_fan) from fani f where nationalitate = 'Spania'),

'Egipt', (select count(f.cod\_fan) from fani f where nationalitate = 'Egipt'),

'Olanda', (select count(f.cod\_fan) from fani f where nationalitate = 'Olanda'),

'Germania', (select count(f.cod\_fan) from fani f where nationalitate = 'Germania'),

'Romania', (select count(f.cod\_fan) from fani f where nationalitate = 'Romania'),

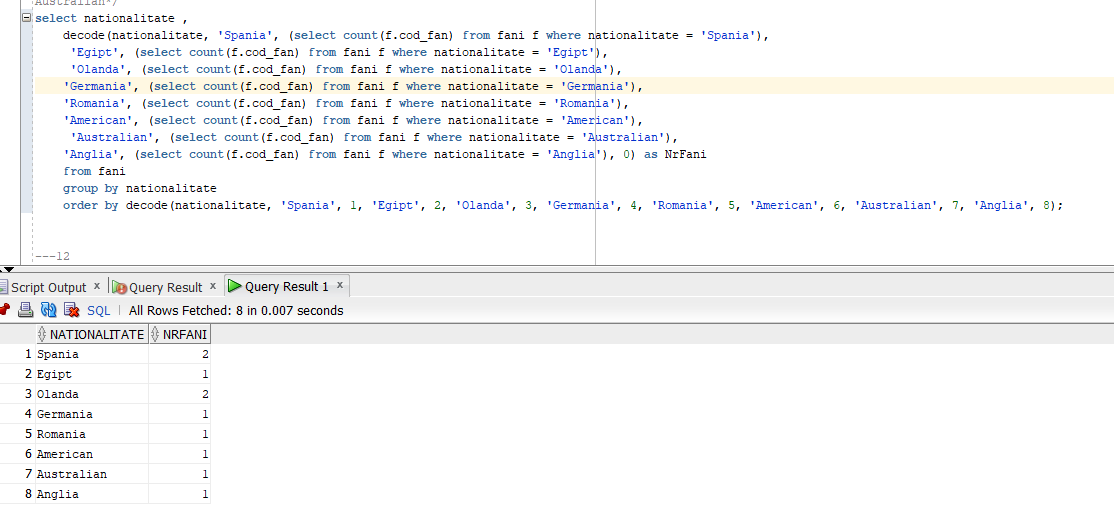
'American', (select count(f.cod\_fan) from fani f where nationalitate = 'American'),

'Australian', (select count(f.cod\_fan) from fani f where nationalitate = 'Australian'),

'Anglia', (select count(f.cod\_fan) from fani f where nationalitate = 'Anglia'), 0) as NrFani

from fani

group by nationalitate

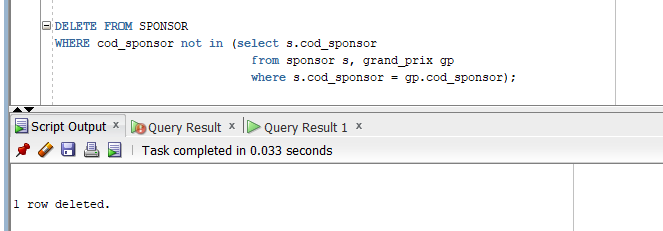
order by decode(nationalitate, 'Spania', 1, 'Egipt', 2, 'Olanda', 3, 'Germania', 4, 'Romania', 5, 'American', 6, 'Australian', 7, 'Anglia', 8);  
  
  
  
**12** **Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.**  
  
**1.Să se șteargă toți sponsorii care nu sponsorizează un grand prix.**

DELETE FROM SPONSOR

WHERE cod\_sponsor not in (select s.cod\_sponsor

from sponsor s, grand\_prix gp

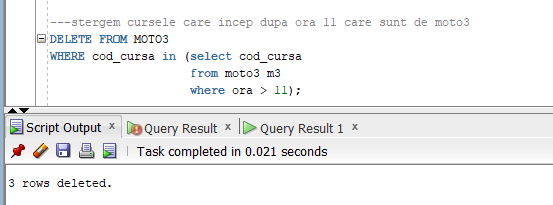
where s.cod\_sponsor = gp.cod\_sponsor);

  
  
  
  
  
  
  
  
  
**2.Să se șteargă toate cursele de MOTO3 ce încep după ora 11.00.**  
  
DELETE FROM MOTO3

WHERE cod\_cursa in (select cod\_cursa

from moto3 m3

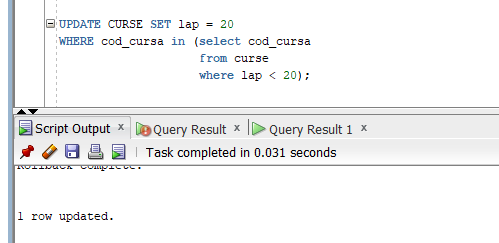
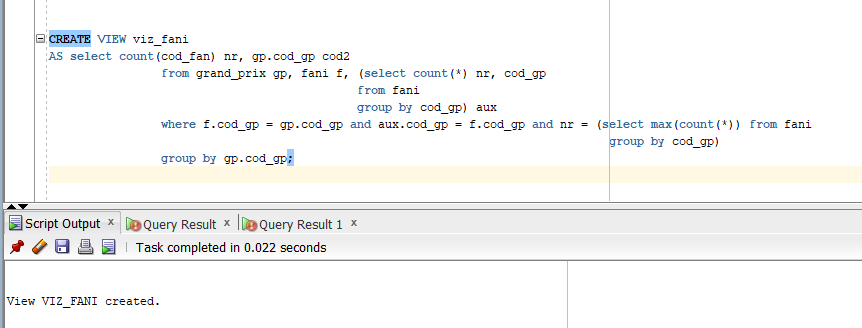
where ora > 11);



**3.Să se modifice numărul de lapuri al fiecărei curse astfel încat lap count>20.**  
  
UPDATE CURSE SET lap = 20

WHERE cod\_cursa in (select cod\_cursa

from curse

 where lap < 20);  
  
  
  
  
 **14 Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.**  
  
  
  
CREATE VIEW viz\_fani

AS select count(cod\_fan) nr, gp.cod\_gp cod2

from grand\_prix gp, fani f, (select count(\*) nr, cod\_gp

from fani   
 group by cod\_gp) aux

where f.cod\_gp = gp.cod\_gp and aux.cod\_gp = f.cod\_gp and nr = (select max(count(\*)) from fani

group by cod\_gp)

group by gp.cod\_gp;  
  
---Nu putem realiza operatii LMD in vizualizarea propusa pentru ca aceasta are atat functii grup (count), cat si clauze GROUP BY.

CREATE VIEW viz\_pilot

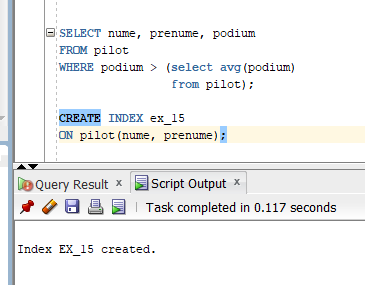
AS select \* from pilot where mod(length(nume) - length(replace(replace(replace(replace(replace(lower(nume), 'a', ''), 'e', ''), 'i', ''), 'o', ''), 'u', '')),2) = 0;

---operatie lmd permisa

UPDATE viz\_pilot SET nume = 'nume\_nou'

WHERE podium = (select max(podium)

from pilot

 );  
  
**15 Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii. Specificați cererea**.SELECT nume, prenume, podium

FROM pilot

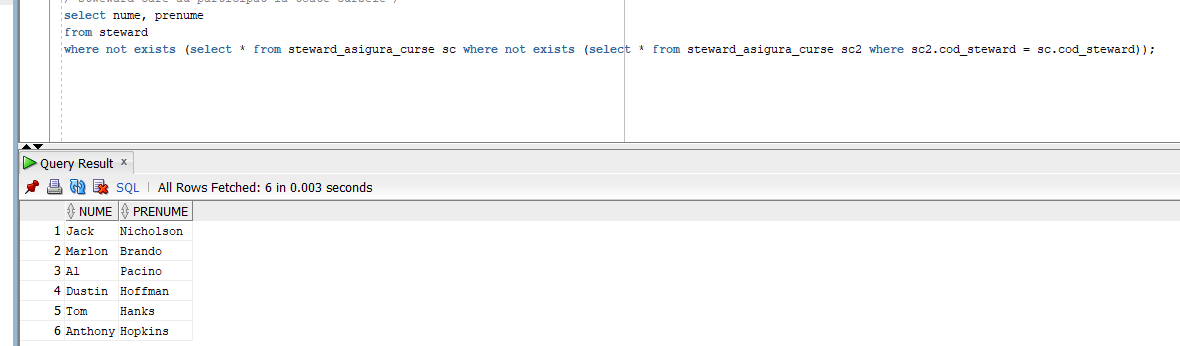
WHERE podium > (select avg(podium)

from pilot);

CREATE INDEX ex\_15

ON pilot(nume, prenume);  
  
  
  
  
   
**16 Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele și două cereri ce utilizează operația division.**  
 **1.SĂ SE SELECTEZE TOȚI STEWARZII CE AU ASIGURAT FIECARE CURSĂ**  
  
  
  
  
  
 select nume, prenume

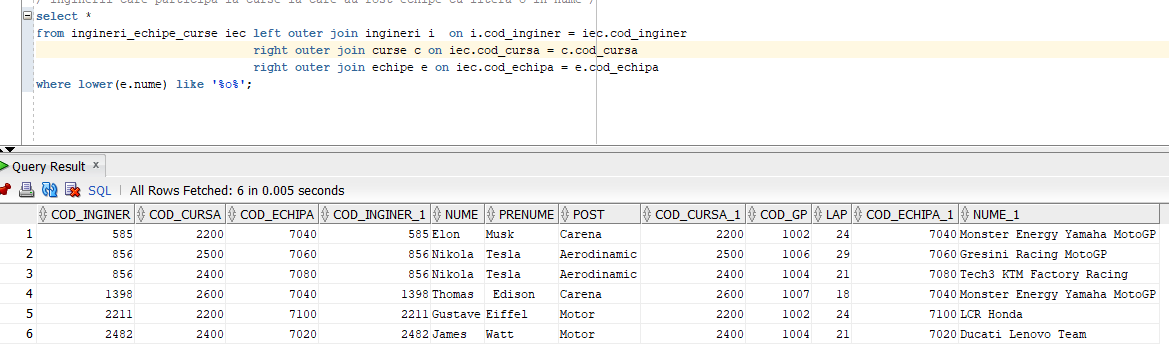
from steward

where not exists (select \* from steward\_asigura\_curse sc where not exists (select \* from steward\_asigura\_curse sc2 where sc2.cod\_steward = sc.cod\_steward));  
  
  
  
  
  
  
  
  
  
 **2.SĂ SE SELECTEZE TOȚI INGINERII CE AU CONCURAT ÎN CADRUL UNEI CURSE PENTRU O ECHIPĂ CE CONȚINE LITERA „O” ÎN NUME**  
  
  
select \*

from ingineri\_echipe\_curse iec left outer join ingineri i on i.cod\_inginer = iec.cod\_inginer

right outer join curse c on iec.cod\_cursa = c.cod\_cursa

right outer join echipe e on iec.cod\_echipa = e.cod\_echipa

where lower(e.nume) like '%o%';  
  
  
17.Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.  
  
Să se afișeze numele și codul echipelor ale căror piloți au 3 sau 5 podiumuri.  
  
VARIANTĂ NEOPTIMIZATĂ  
R1 = SELECT(PILOT, PODIUM=3)  
R2 = SELECT(PILOT, PODIUM=5)  
R3 = UNION (R1, R2)  
R4 = SEMIJOIN (ECHIPA, R3)  
REZULTAT = R5 = PROJECT(R4, COD\_ECHIPA)  
  
select c.cod\_echipa, e.nume

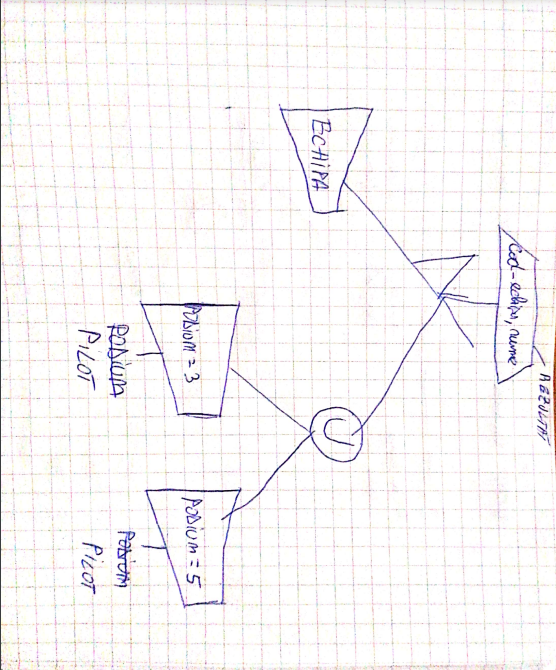
from contract c ,echipe e, ( select p.\* from PILOT p

where p.podium=3

union

select p.\* from PILOT p

where p.podium=5) aux

where c.cod\_pilot=aux.cod\_pilot and c.cod\_echipa=e.cod\_echipa;  
  
  
  
  
  
  
  
  
VARIANTĂ OPTIMIZATĂ  
  
R1 =SELECT (PILOT, PODIUM=3 OR PODIUM = 5)  
R2 = PROJECT (R1, COD\_PILOT)  
R3 = PROJECT(ECHIPA, NUME, COD\_ECHIPE)  
REZULTAT = SEMIJOIN (R1, R3)  
  
select c.cod\_echipa, e.nume

from contract c ,echipe e, ( select p.\* from PILOT p

where p.podium=3 or p.podium=5) aux

where c.cod\_pilot=aux.cod\_pilot and c.cod\_echipa=e.cod\_echipa;  
  
