Feta Almar-Eran

Grupa 133

**Gestionarea unui aeroport**

1. **Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.**

Mi-am propus sa realizez o baza de date care sa faciliteze gestiunea unui aeroport. Acesta va contine date despre angajatii din aeroport si joburile lor, atat cei care lucreaza in departamente terestre, cat si cei care zboara (pentru acestia se poate tine si evidenta avioanelor in care a mai lucrat pana atunci). De asemenea, baza de date va oferi informatii despre zborurile care pleaca din acel aeroport, avioanele implicate in zboruri, rutele si pistele pe care vor pleca, precum si hangarele si portile unde se afla. Se pot furniza informatii si despre clientii care vor pleca din acel aeroport, alturi de optiunile lor: pot cere sau nu transport pana la aeroport, parcare pentru timp indelungat, sau pot avea bagaje de diferite dimensiuni. Consider ca o astfel de baza de date este utila unui aeroport, intrucat exista foarte multe date care trebuiesc procesate cat mai rapid pentru a facilita fluiditatea activitatilor desfasurate.

1. **Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.**

* Trebuie sa existe minim un angajat care sa lucreze in fiecare job disponibil.
* Un angajat nu poate avea mai multe joburi in cadrul aeroportului.
* Pot exista departamente in care sa nu lucreze nimeni.
* Un angajat poate face parte dintr-un singur departament.
* O parte din angajati, cei care lucreaza in avioane, vor fi repartizati unui anumit avion.
* Pentru angajatii care lucreaza in avioane, se va tine evidenta avioanelor in care au lucrat pana atunci.
* Un angajat poate lucra intr-un singur avion la un moment dat.
* Avioanele pot sau nu sa fie depozitate in hangare, un hangar avand spatiu pentru mai multe avioane.
* Zborurile au atribuite rute, piste, porti si avioane.
* Nu este neaparat ca fiecare ruta, pista, poarta, avion sa aiba un zbor in acel moment.
* Fiecarui zbor ii este atribuit un anumit numar de clienti, dar zborul nu poate exista fara sa aiba macar un client.
* Fiecare client poate avea bilet la mai multe zboruri, dar nu poate fi inregistrat drept client daca nu are minim un bilet.
* Clientii pot alege daca vor avea sau nu bagaj si ce fel de bagaj (de mana, de cala, etc.).
* Clientii pot sa aleaga si daca vor transport pana la aeroport sau daca vor parcare pentru termen lung.
* Fiecare client poate avea maxim un loc de parcare.

1. **Descrierea entităților, incluzând precizarea cheii primare.**

* JOB: (PK: id\_job) contine lista joburilor disponibile in cadrul aeroportului (atat cei cu joburi in aeroport, cat si cei ce zboara: piloti, insotitori de zbor.
* ANGAJAT: (PK: id\_angajat) reprezinta o lista cu angajatii si ofera detalii in legatura cu acestia (numele, salariul, CNP, telefon, etc.).
* DEPARTAMENT: (PK: id\_departament) este lista departamentelor existente in aeroport.
* ISTORIC: (PK: id\_istoric) reprezinta istoricul avioanelor in care au mai lucrat pilotii sau insotitorii de zbor.
* AVION: (PK: id\_avion) lista avioanelor prezente in momentul actual in aeroport.
* HANGAR: (PK: id\_hangar) hangarele disponibile pentru depozitarea avioanelor.
* ZBOR: (PK: id\_zbor) contine toate zborurile cu plecare din aeroportul nostru si informatii despre preturi si date.
* RUTA: (PK: id\_ruta) este lista rutelor cu plecare din aeroport si ofera informatii despre distanta si destinatie.
* PISTE: (PK: id\_pista) lista pistelor de pe care vor pleca avioanele.
* PORTI: (PK: id\_poarta) lista portilor de imbarcare disponibile.
* CLIENT: (PK: id\_client) este lista persoanelor care au cel putin un bilet catre un zbor si ofera informatii generale despre acestia.
* PARCARE: (PK: id\_parcare) lista locurilor de parcare disponibile pentru termen indelungat.
* TRANSPORT: (PK: id\_transport) este lista mijloacelor de transport public destinate aduceri clientilor la aeroport.
* BAGAJE: (PK: id\_bagaj) contine lista bagajelor.

1. **Descrierea relațiilor, incluzând precizarea cardinalității acestora.**

* JOB-ANGAJAT: (One to many) fiecare job are mai multi angajati (minim unul) si fiecare angajat are un singur job.
* DEPARTAMENT-ANGAJAT: (One to many) fiecare departament are mai multi angajati (pot exista departamente fara angajati) si fiecare angajat face parte dintr-un singur departament.
* ANGAJAT-ISTORIC: (One to many) un angajat care lucreaza ca pilot sau insotitor de zbor poate avea mai multe avioane in care a lucrat inainte, iar un astfel de istoric poate sa fie al unui singur angajat.
* ANGAJAT-AVION: (Many to one) unii angajati pot face parte dintr-o echipa a unui singur avion (un avion are minim un angajat).
* HANGAR-AVION: (One to many) fiecare hangar poate depozita mai multe avioane (nu e neaparat sa depoziteze un avion), iar fiecare avion poate poate sta intr-un singur hangar la un moment dat.
* AVION-ZBOR: (One to many) un avion poate sa fie folosit pentru mai multe zboruri (pot exista avioane fara zboruri), iar fiecare zbor are un singur avion.
* ZBOR-RUTA: (Many to one) un zbor poate sa fie pe o singura ruta, iar o ruta poate fi folosita de mai multe zboruri (pot exista rute necirculate).
* ZBOR-PISTE: (Many to one) un zbor poate sa porneasca de pe o singura pista, iar o pista poate sa fie folosita pentru mai multe zboruri (pot exista piste nefolosite).
* ZBOR-PORTI: (Many to one) un zbor poate sa porneasca de la o singura poarta, iar o poarta poate sa fie folosita pentru mai multe zboruri (pot exista porti nefolosite).
* ZBOR-CLIENT: (Many to many) un zbor poate sa aiba mai multi clienti, iar un client poate sa aiba bilete catre mai multe zboruri.
* CLIENT-PARCARE: (One to one) un client poate sa aiba un singur loc de parcare pe termen lung.
* CLIENT-TRANSPORT: (One to many) un client poate solicita un singur mijloc de transport in comun pentru a ajunge la aeroport, iar fiecare mijloc de transport poate avea mai multi clienti.
* CLIENT-BAGAJE: (One to many) un client poate avea mai multe bagaje (sau niciunul), iar un bagaj poate sa fie al unui singur client.

1. **Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.**
   * + JOBS: -id\_job: (Number, Not Null) este cheia primara a tabelei.

-nume\_job: (Char, Not Null) reprezinta denumirea unui anumit job. (Valori posibile: pilot, insotitor de zbor, controlor de zbor, etc.)

-sal\_min: (Number) reprezinta valoarea minima a salariului din acel job.

-sal\_max: (Number) reprezinta valoarea maxima a salariului din acel job.

* + - ANGAJAT: -id\_angajat: (Number, Not Null) cheia primara a tabelei.

-nume: (Char, Not Null) numele angajatilor.

-prenume: (Char, Not Null) prenumele angajatilor.

-CNP: (Number, Not Null) codul numeric al angajatilor.

-data\_angajare: (Date, Not Null) data in care au fost angajati salariatii.

-salariu: (Number) salariul pe care il primeste un anumit angajat.

-email: (Char) emailul unui angajat.

-telefon: (Number) numarul de telefon al unui angajat.

-id\_departament: (Number) id-ul departamentului in care se afla un angajat.

-id\_job: (Number, Not Null) id-ul jobului in care lucreaza un angajat.

-id\_avion: (Char) id-ul avionului in care lucreaza in present un angajat.

* + - DEPARTAMENT: -id\_departament: (Number, Not Null) cheia primara a tabelei.

-nume\_departament: (Char, Not Null) numele departamentului.

* + - ISTORIC: -id\_istoric: (Number, Not Null) cheia primara a tabelei.

-id\_angajat: (Number, Not Null) id-ul angajatului din istoric.

-id\_avion: (Char, Not Null) id-ul avionului in care a mai lucrat un angajat.

-data\_inceput: (Date) data de inceput.

-data\_sfarsit: (Date) data de sfarsit.

* + - AVION: -id\_avion: (Char, Not Null) cheia primara a tabelei.

-tip\_avion: (Char, Not Null) denumirea tipului de avion. (Ex. Boeing, AirBus)

-companie\_aeriana: (Char, Not Null) denumirea companiei aeriene. (Ex. Wizz Air, Blue Air)

-id\_hangar: (Number) id-ul hangarului in care este depozitat avionul.

* + - HANGAR: -id\_hangar: (Number, Not Null) cheia primara a tabelei.

-cod\_hangar: (Char, Not Null) codul hangarului.

* + - ZBOR: -id\_zbor: (Char, Not Null) cheia primara a tabelei.

-data\_plecare: (Date, Not Null) data in care va pleca avionul din aeroport.

-data\_sosire: (Date, Not Null) data in care va ajunge la destinatie.

-pret\_clasa1: (Number) pretul zborului la clasa I.

-pret\_clasa2: (Number) pretul zborului la clasa a II-a.

-id\_avion: (Char, Not Null) id-ul avionului care va efectua zborul.

-id\_ruta: (Number, Not Null) id-ul rutei pe care va pleca avionul.

-id\_pista: (Number, Not Null) id-ul pistei de pe care va pleca avionul.

-id\_poarta: (Number, Not Null) id-ul portii de unde va pleca avionul.

* + - RUTA: -id\_ruta: (Number, Not Null) cheia primara a tabelei.

-destinatie: (Char, Not Null) denumirea aeroportului in care va ajunge.

-distanta: (Number) distanta in km a zborului.

* + - PISTE: -id\_pista: (Number, Not Null) cheia primara a tabelei.

-cod\_pista: (Char, Not Null) codul pistei.

* + - PORTI: -id\_poarta: (Number, Not Null) cheia primara a tabelei.

-numar: (Char, Not Null) codul portii de unde pleaca un avion.

* + - CLIENT: -id\_client: (Number, Not Null) cheia primara a tabelei.

-nume: (Char, Not Null) numele clientului.

-prenume: (Char, Not Null) prenumele clientului.

-CNP: (Number, Not Null) codul numeric al clientului.

-telefon: (Number) numarul de telefon al clientului.

-id\_zbor: (Char, Not Null) id-ul zborului la care are bilet un client.

-id\_parcare: (Number) id-ul locului de parcare.

-id\_transport: (Number) id-ul mijlocului de transport.

* + - PARCARE: -id\_parcare: (Number, Not Null) cheia primara a tabelei.

-numar\_loc: (Char, Not Null) codul locului de parcare.

* + - TRANSPORT: -id\_transport: (Number, Not Null) cheia primara a tabelei.

-tip\_vehicul: (Char, Not Null) tipul vehiculului. (Ex. Autobuz, microbuz, etc.)

-nr\_locuri: (Number) numarul de locuri al vehiculului.

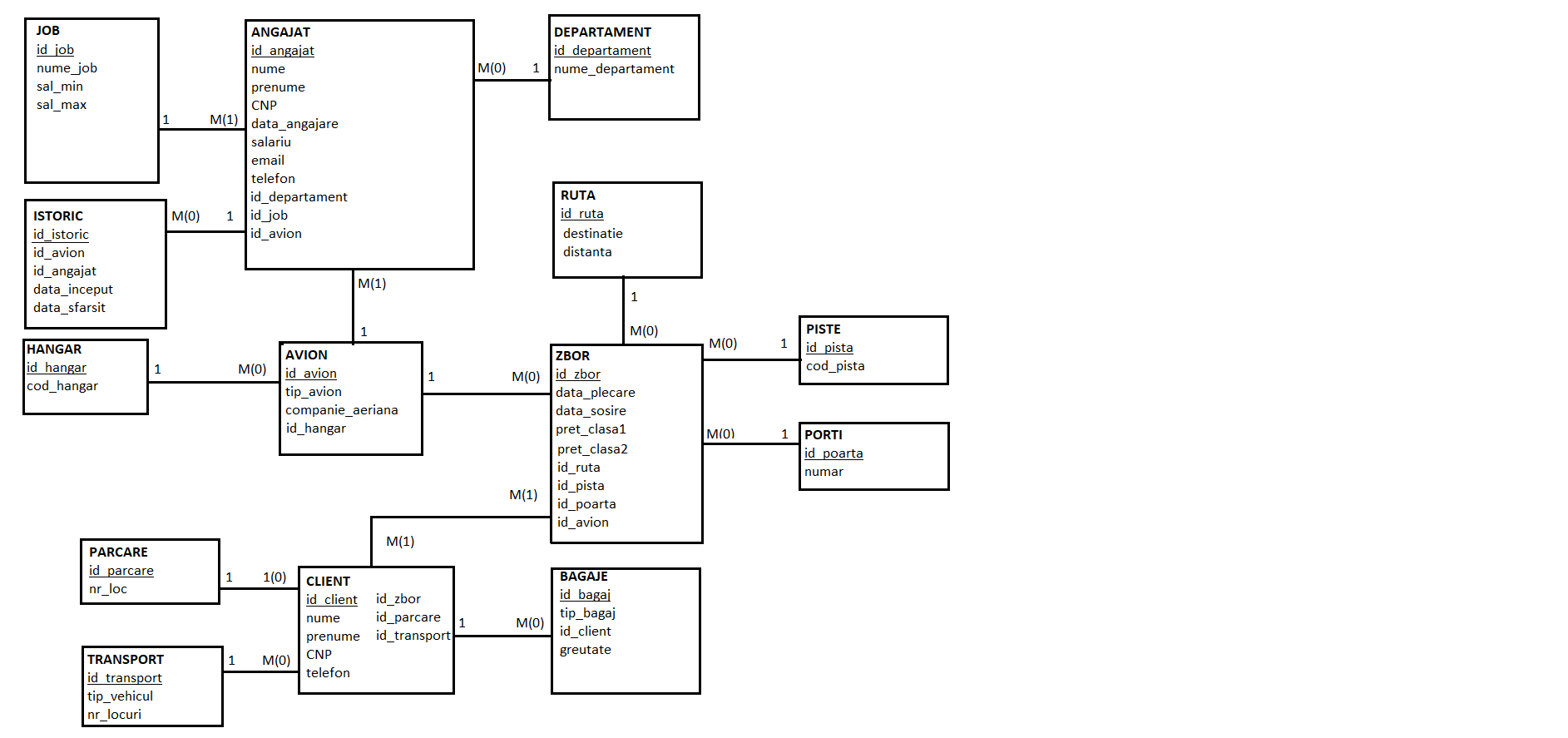
* + - BAGAJE: -id\_bagaj: (Number, Not Null) cheia primara a tabelei.

-tip\_bagaj: (Char, Not Null) tipul bagajului. (Ex. Bagaj de mana, de cala, etc.)

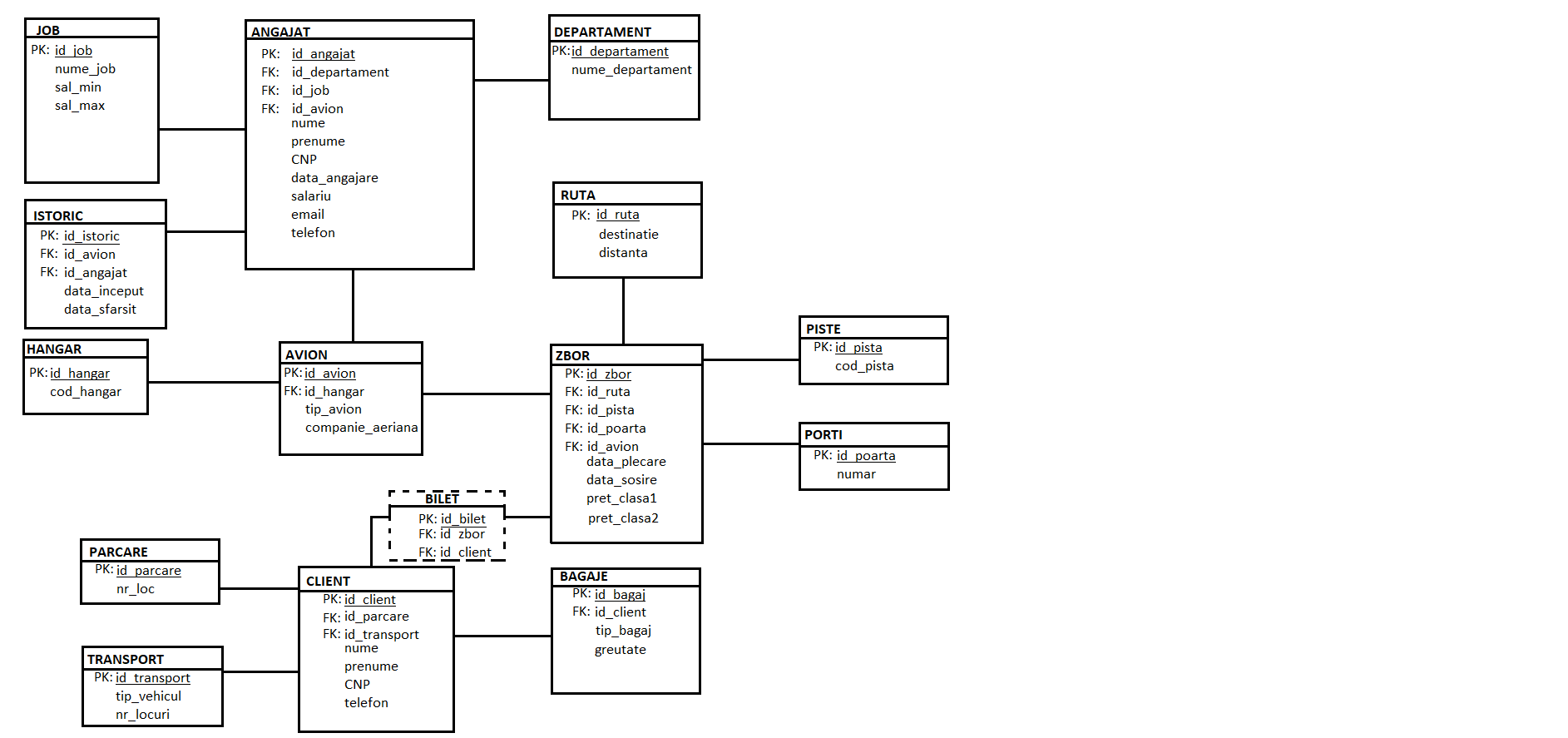
-id\_client: (Number, Not Null) id-ul clientului.

-greutate: (Number) greutatea in kg a bagajului.

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



1. **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.**

****

1. **Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.**

* JOB(id\_job#, nume\_job, sal\_min, sal\_max);
* ANGAJAT(id\_angajat#, id\_departament, id\_job, id\_avion, nume, prenume, CNP, data\_angajare, salariu, email, telefon);
* DEPARTAMENT(id\_departament#, nume\_departament);
* ISTORIC(id\_istoric#, id\_avion, id\_angajat, data\_inceput, data\_sfarsit);
* AVION(id\_avion#, id\_hangar, tip\_avion, companie\_aeriana);
* HANGAR(id\_hangar#, cod\_hangar);
* ZBOR(id\_zbor#, id\_ruta, id\_pista, id\_poarta, id\_avion, data\_plecare, data\_sosire, pret\_clasa1, pret\_clasa2);
* RUTA(id\_ruta#, destinatie, distanta);
* PISTE(id\_pista#, cod\_pista);
* PORTI(id\_poarta#, numar);
* BILET(id\_bilet#, id\_zbor, id\_client);
* CLIENT(id\_client#, id\_parcare, id\_transport, nume, prenume, CNP, telefon);
* PARCARE(id\_parcare#, nr\_loc);
* TRANSPORT(id\_transport#, tip\_vehicul, nr\_locuri);
* BAGAJE(id\_bagaj#, id\_client, tip\_bagaj, greutate);

1. **Realizarea normalizării până la forma normală 3 (FN1-FN3).**

Diagrama conceptuala prezentata la exercitiul 7 reflecta deja baza de date in FN3, deoarece fiecare atribut din ea are valoare indivizibila, iar acele atribute care nu sunt chei primare depind de intreaga cheie si doar de cheie. In continuare am realizat cateva exemple de tabele ce puteau sa nu fie in FN3 si transformarea pe care as fi facut-o.

* In FN1:

Tabela ISTORIC putea contine in atributul avion o lista cu avioanele in care au lucrat pilotii inainte. Pentru a o transforma in FN1 este necesar ca fiecare avion sa fie scris pe o linie separata, chiar daca un pilot a lucrat in mai multe avioane pana atunci.

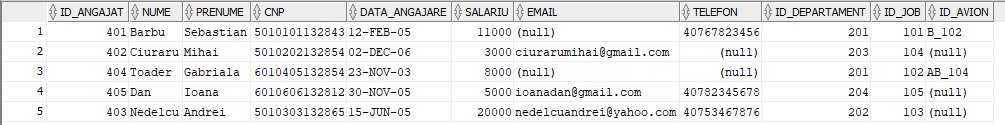
* Din FN1 in FN2:

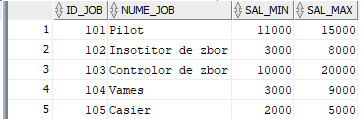
Tabela ANGAJAT putea avea atributele tabelei DEPARTAMENT si 2 chei primare: id\_angajat si id\_departament. In acest caz nume\_departament ar fi depins doar de id\_departament, nu si de id\_angajat. Pentru a aduce tabela la forma normala 2 a fost necesara crearea a 2 tabele separate: ANGAJAT si DEPARTAMENT.

* Din FN2 in FN3:

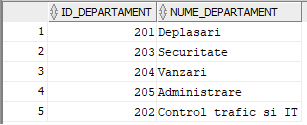
In tabela ANGAJAT puteam avea atributele tabelei JOB, avand o singura cheie primara: id\_angajat. In aceasta situatie, atributele nume\_job, sal\_min si sal\_max nu ar fi depins de cheia primara a tabelei (id\_angajat), ci de id\_job. Pentru a aduce tabela la forma normala 3 a fost necesara crearea a 2 tabele separate: JOB si ANGAJAT.

1. **Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).**

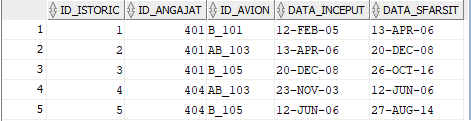
Select \* from angajat;

Select \* from job;

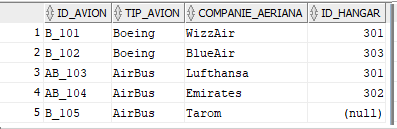
Select \* from departament;



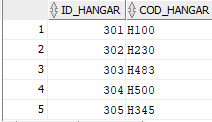
Select \* from istoric;



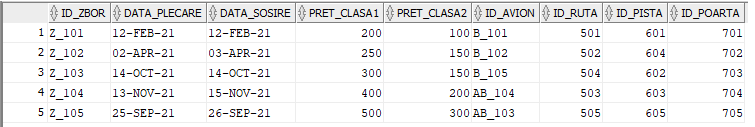
Select \* from avion;



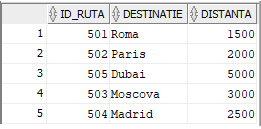
Select \* from hangar;



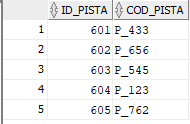
Select \* from zbor;



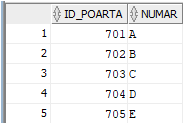
Select \* from ruta;



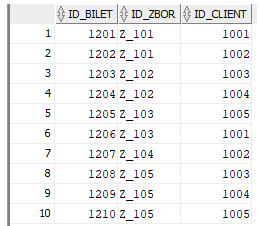
Select \* from piste;



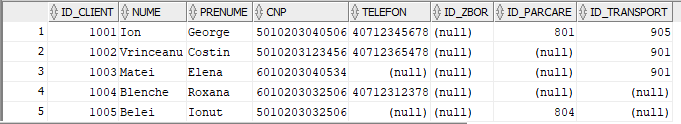
Select \* from porti;



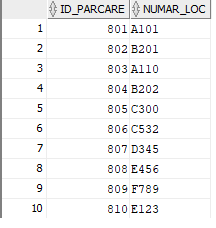
Select \* from bilet;



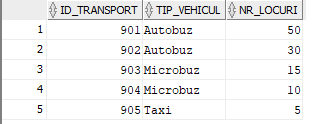
Select \* from client;



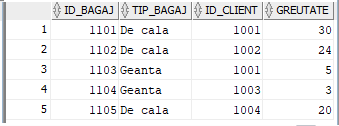
Select \* from parcare;



Select \* from transport;



Select \* from bagaje;



**11. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:**

**• operație join pe cel puțin 4 tabele**

**• filtrare la nivel de linii**

**• subcereri sincronizate în care intervin cel puțin 3 tabele**

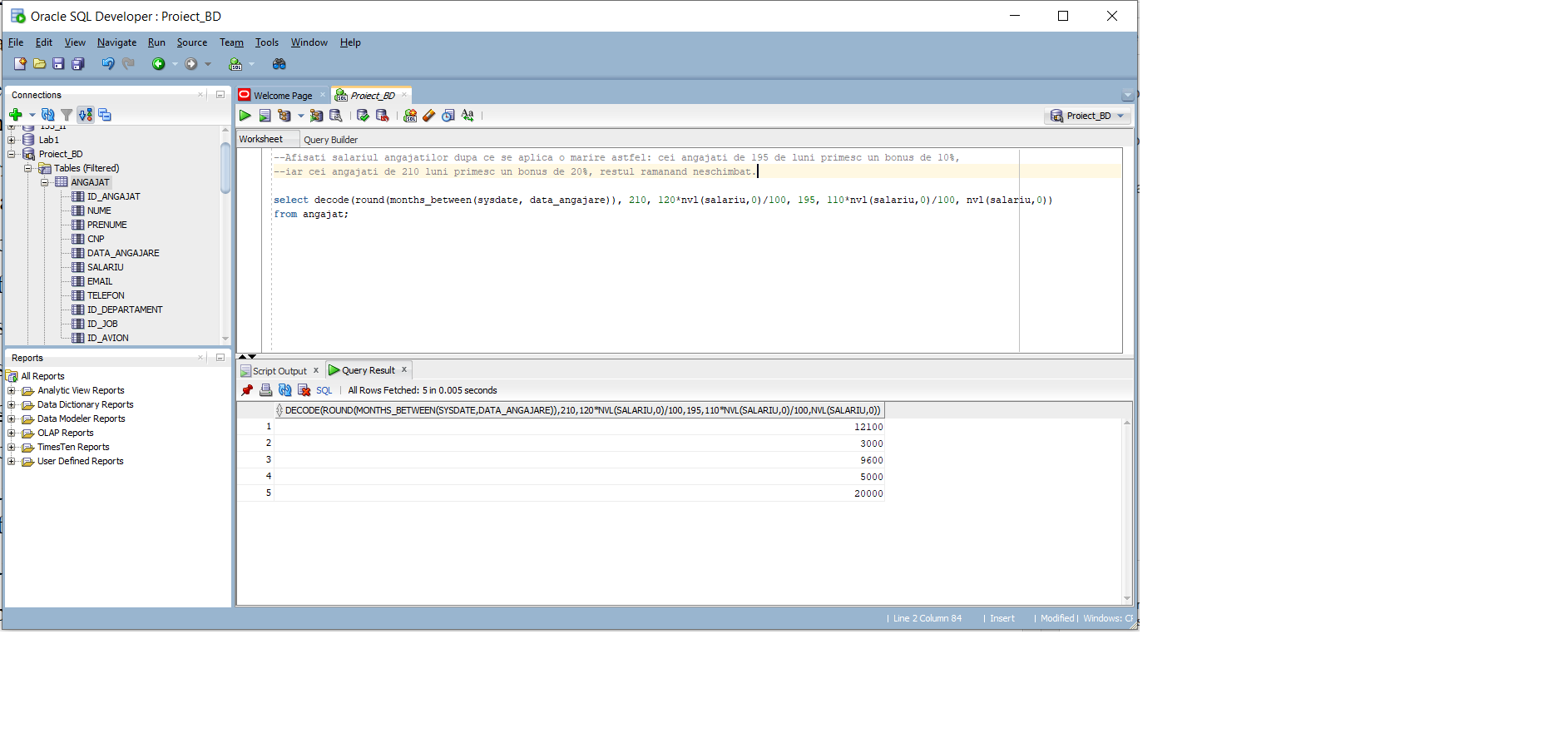
**• subcereri nesincronizate în care intervin cel puțin 3 tabele**

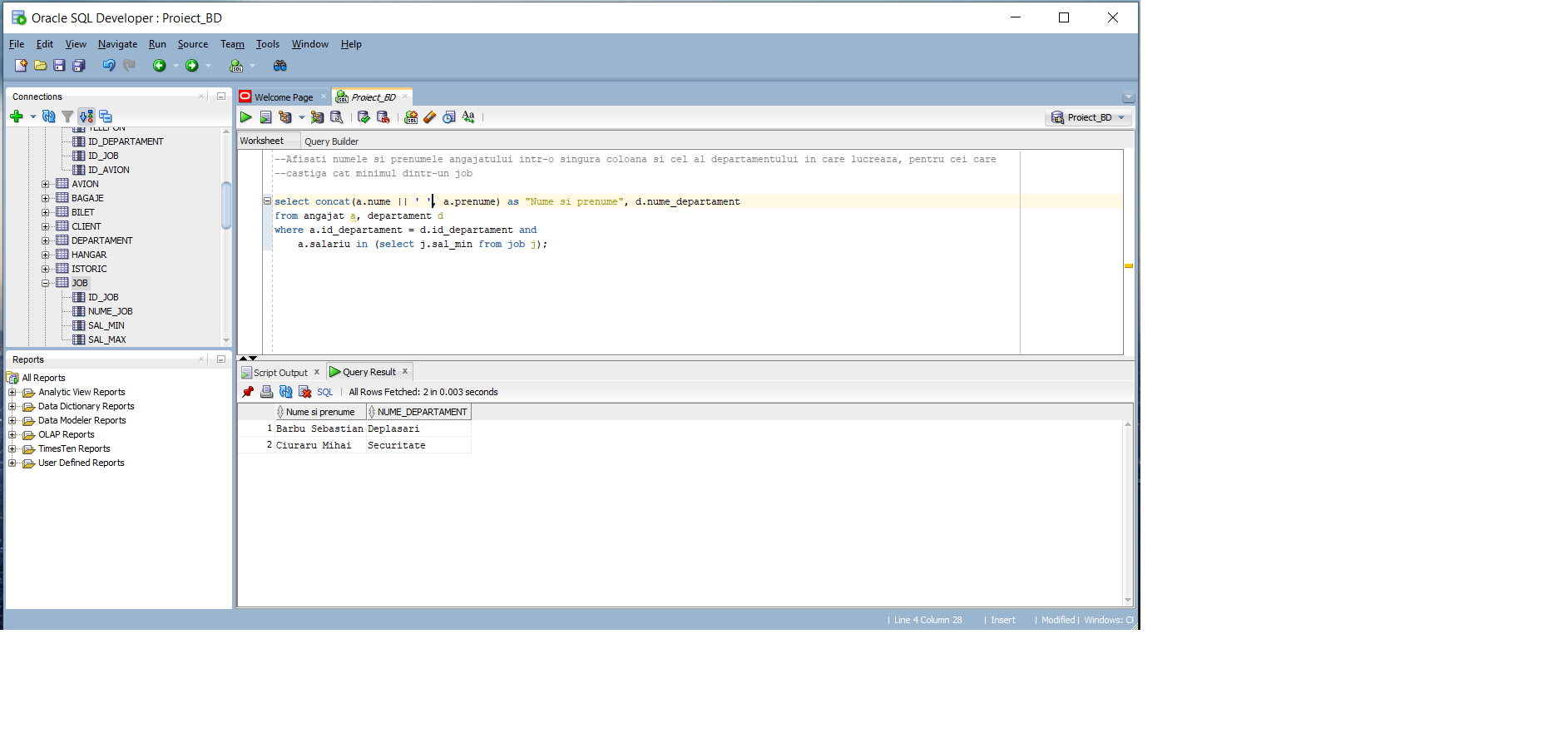
**• grupări de date, funcții grup, filtrare la nivel de grupuri**

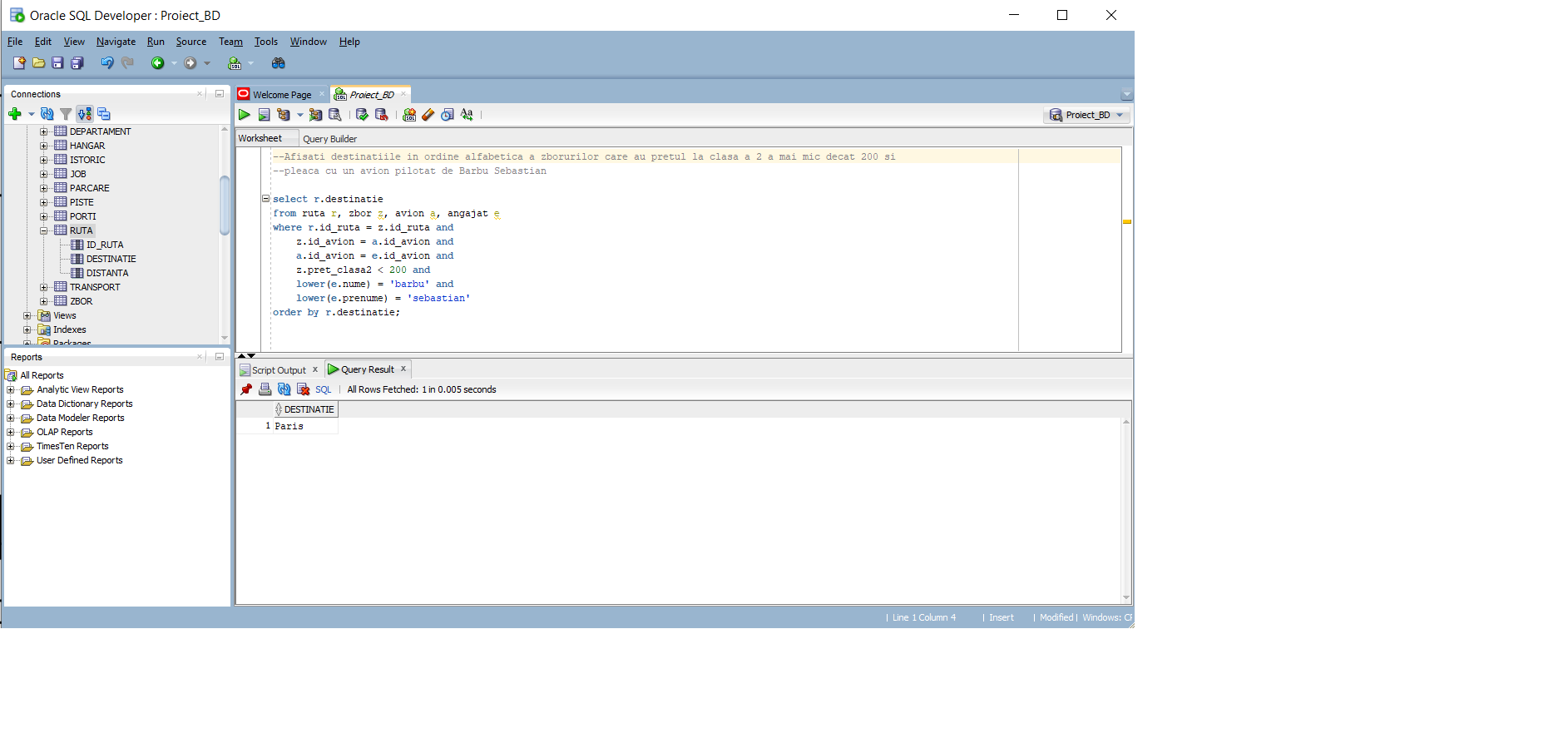
**• ordonări**

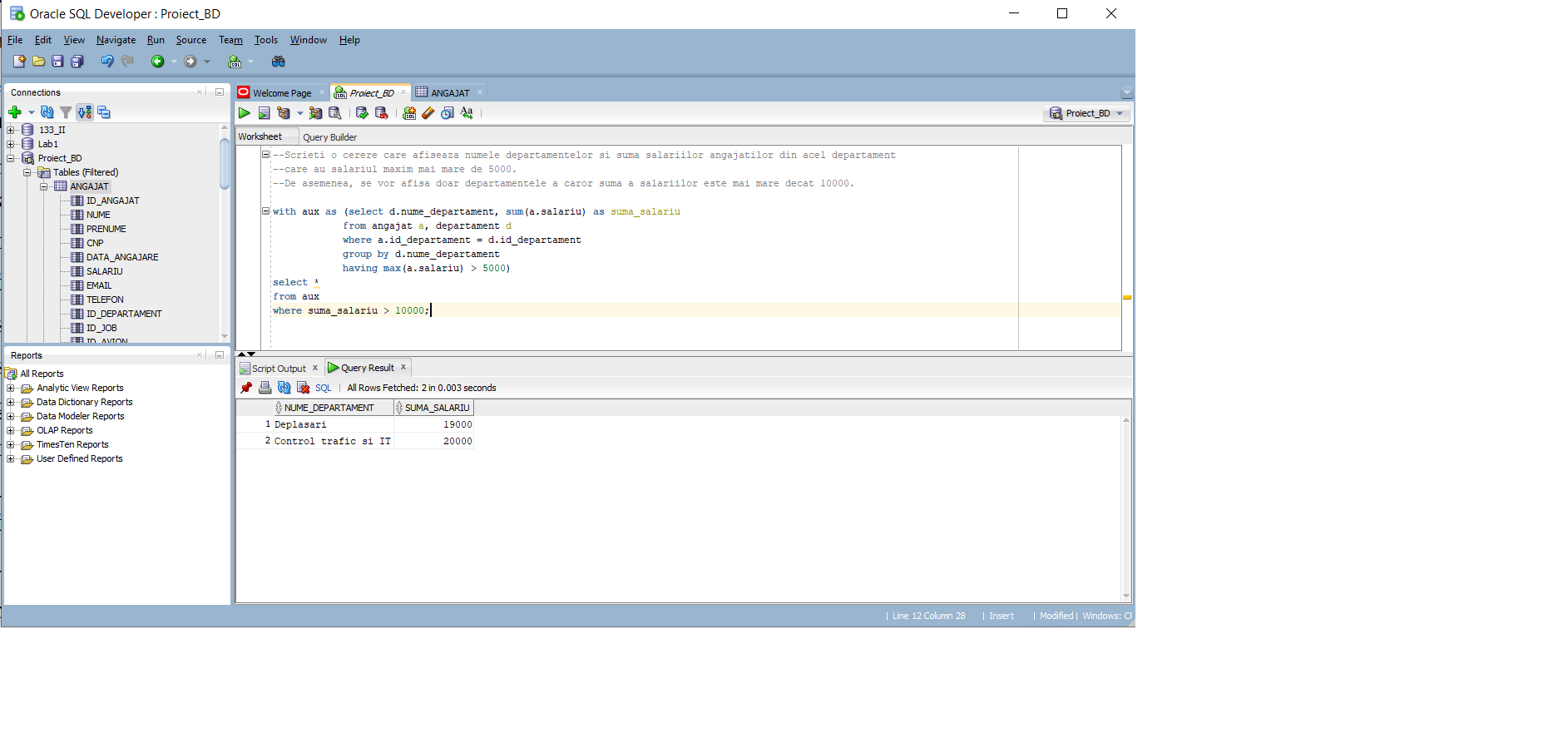
**• utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a funcțiilor NVL și DECODE, a cel puțin unei expresii CASE**

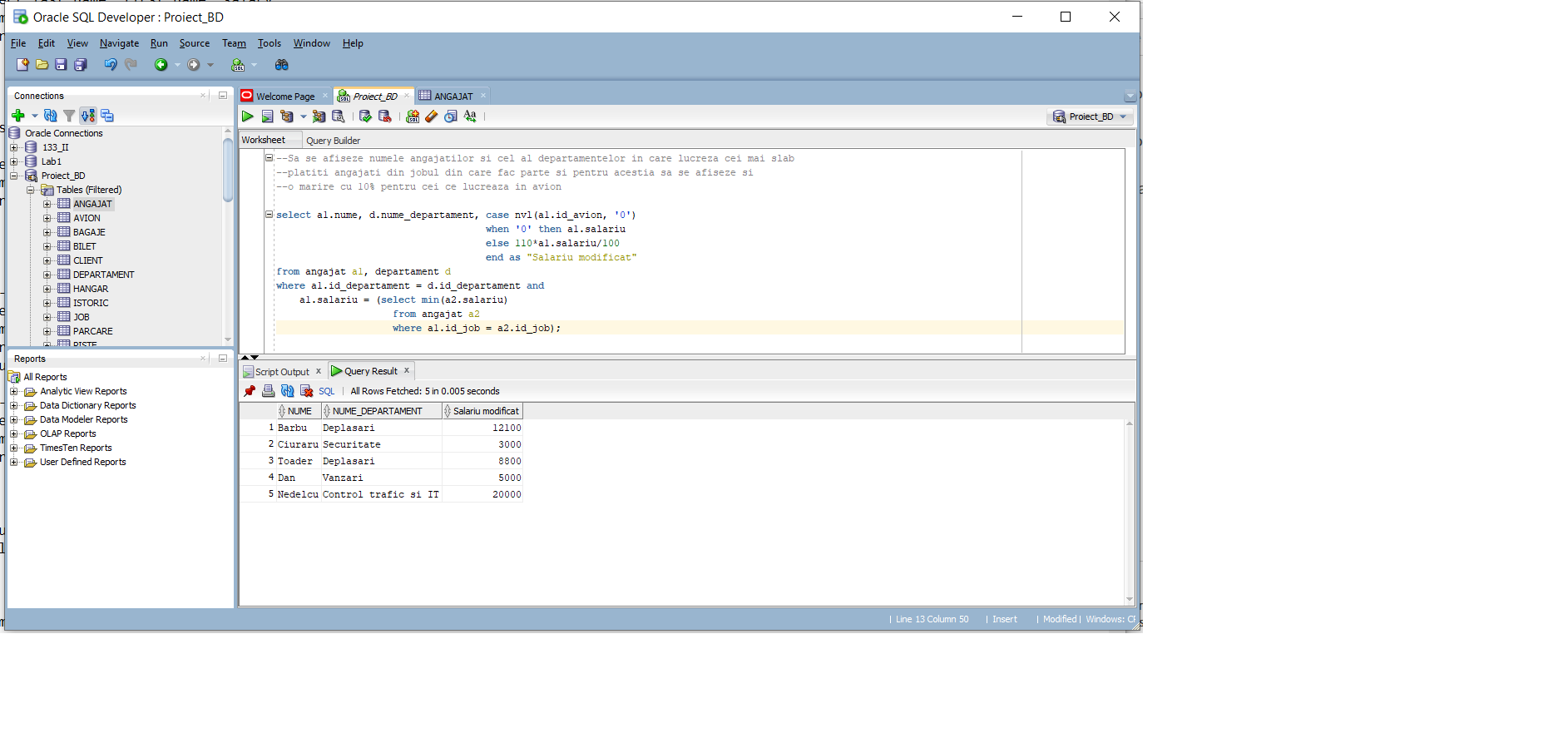
**• utilizarea a cel puțin 1 bloc de cerere (clauza WITH)**



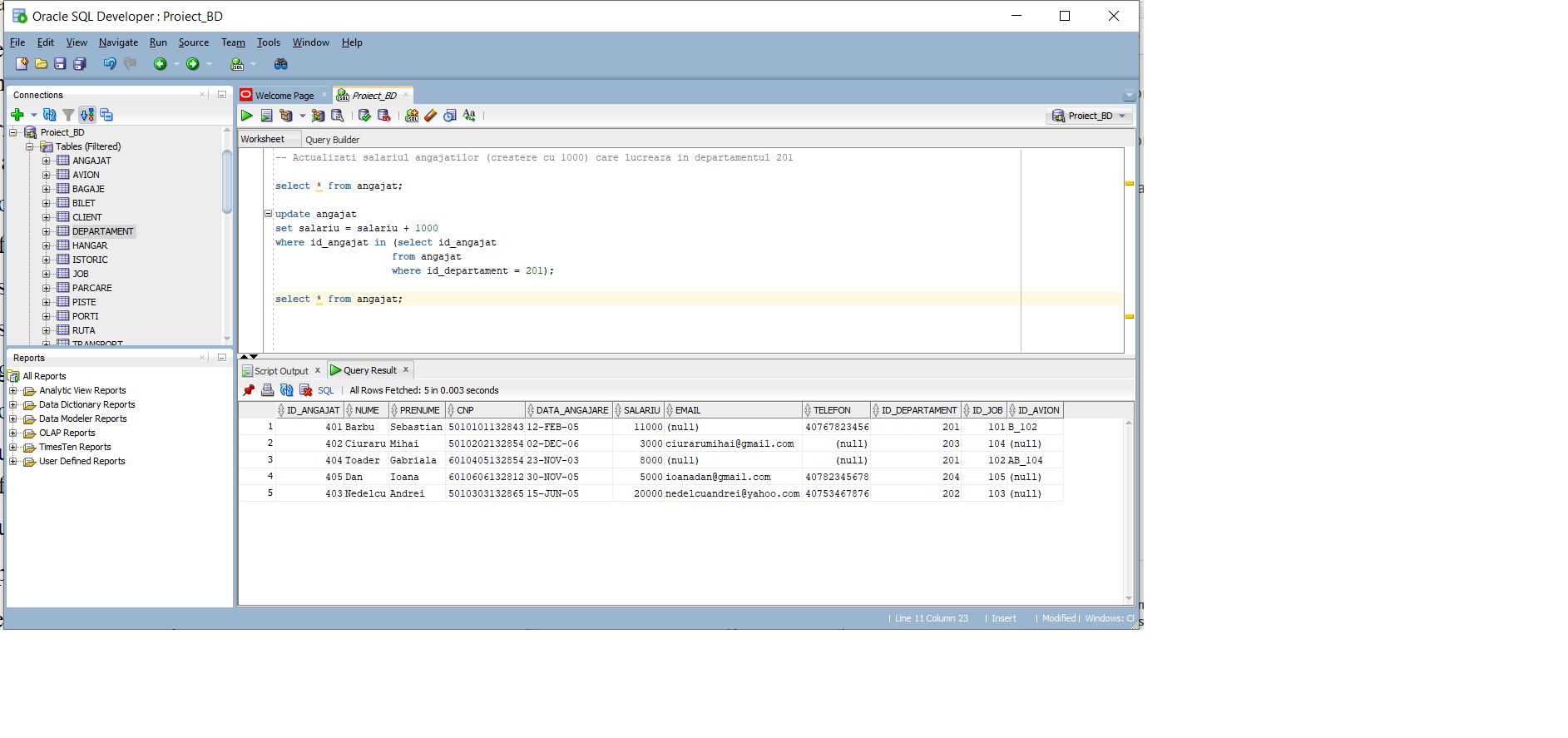


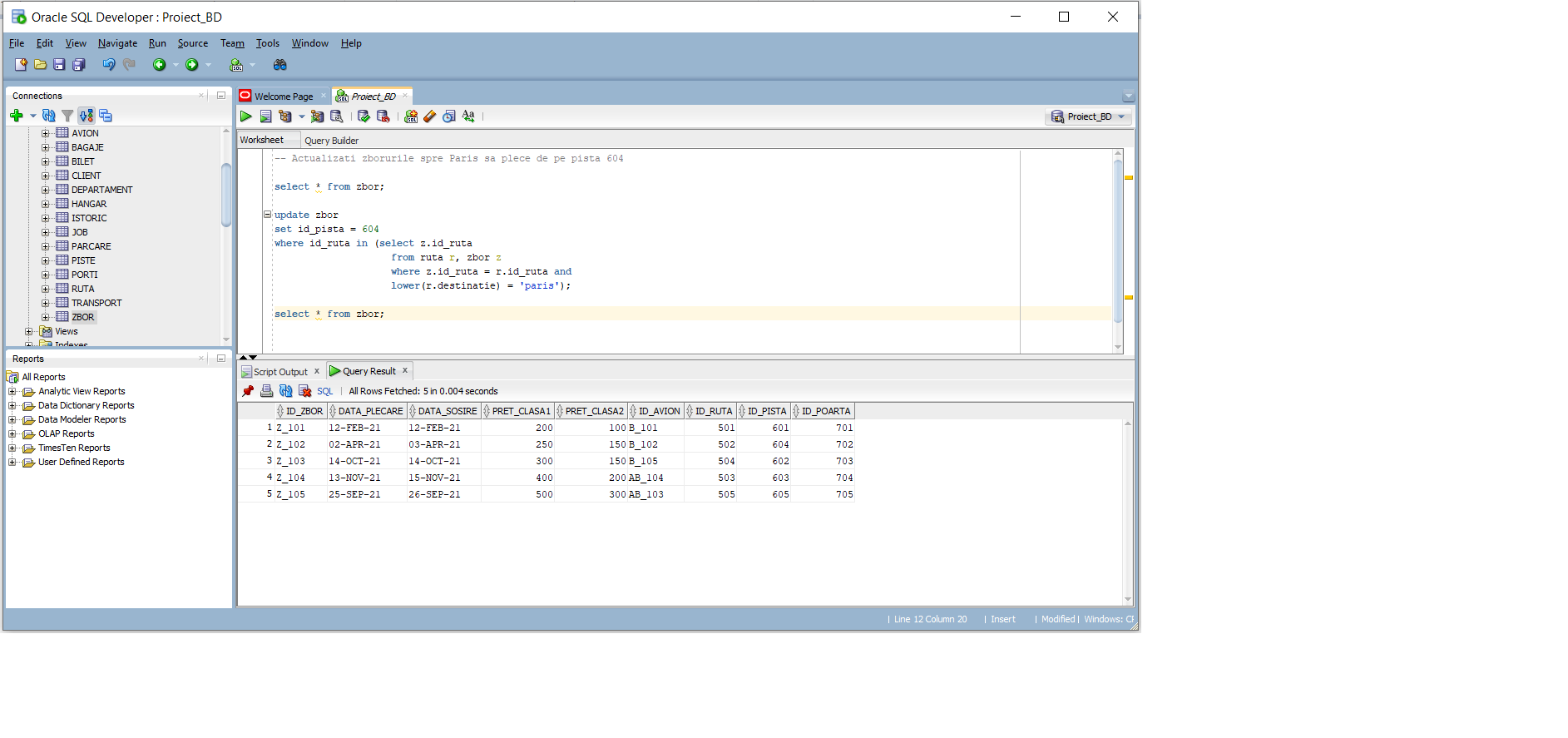


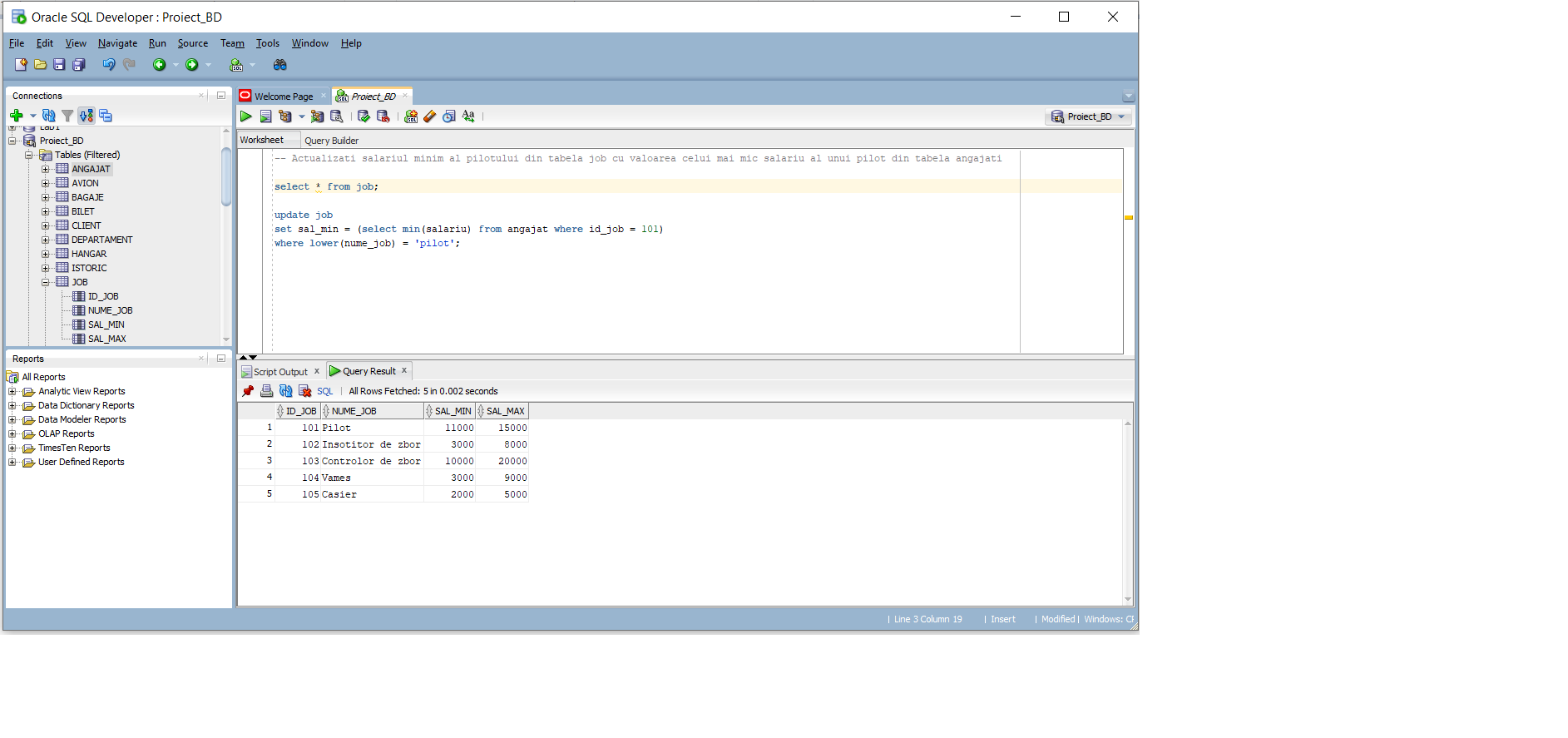




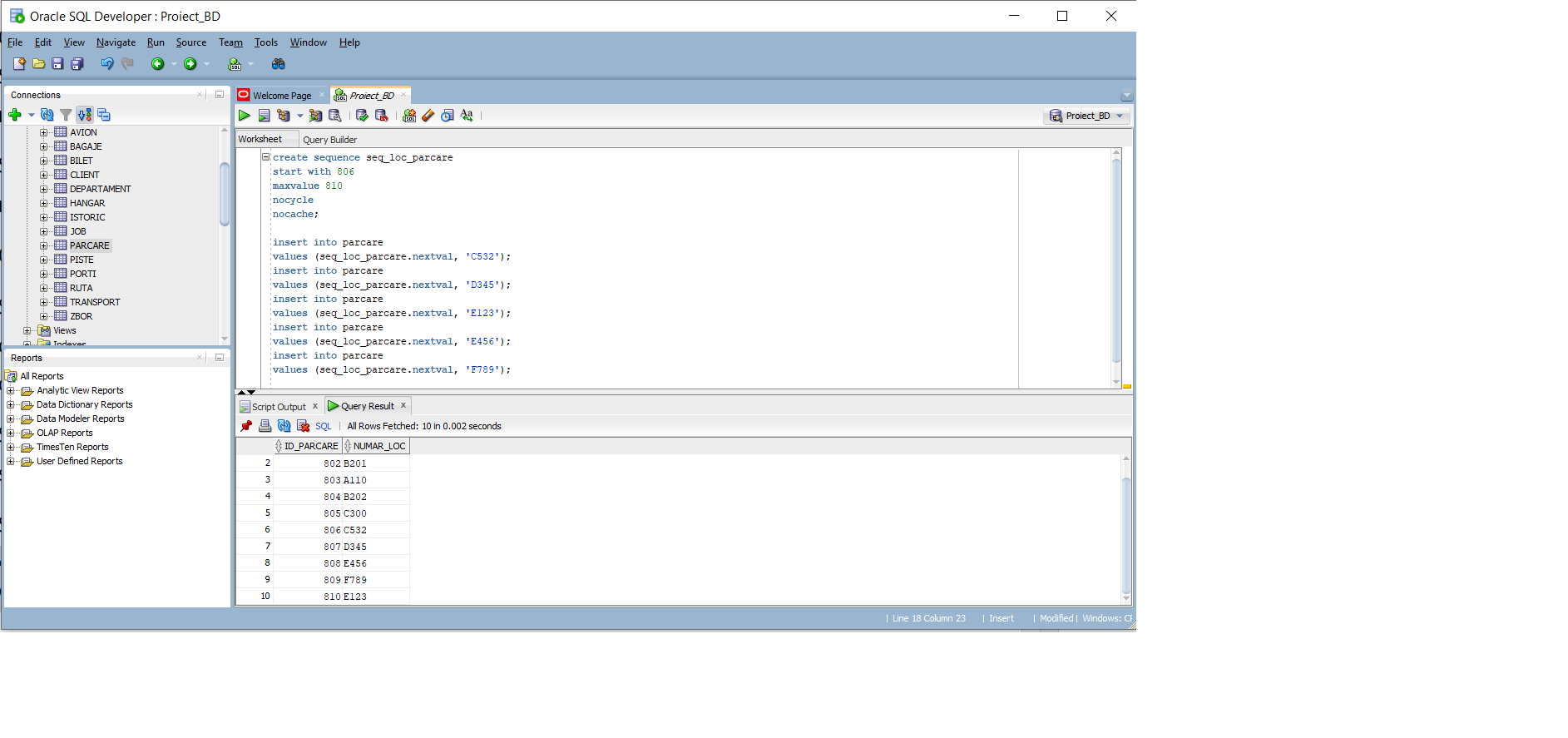
1. **Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.**



****

****

1. **Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 10).**



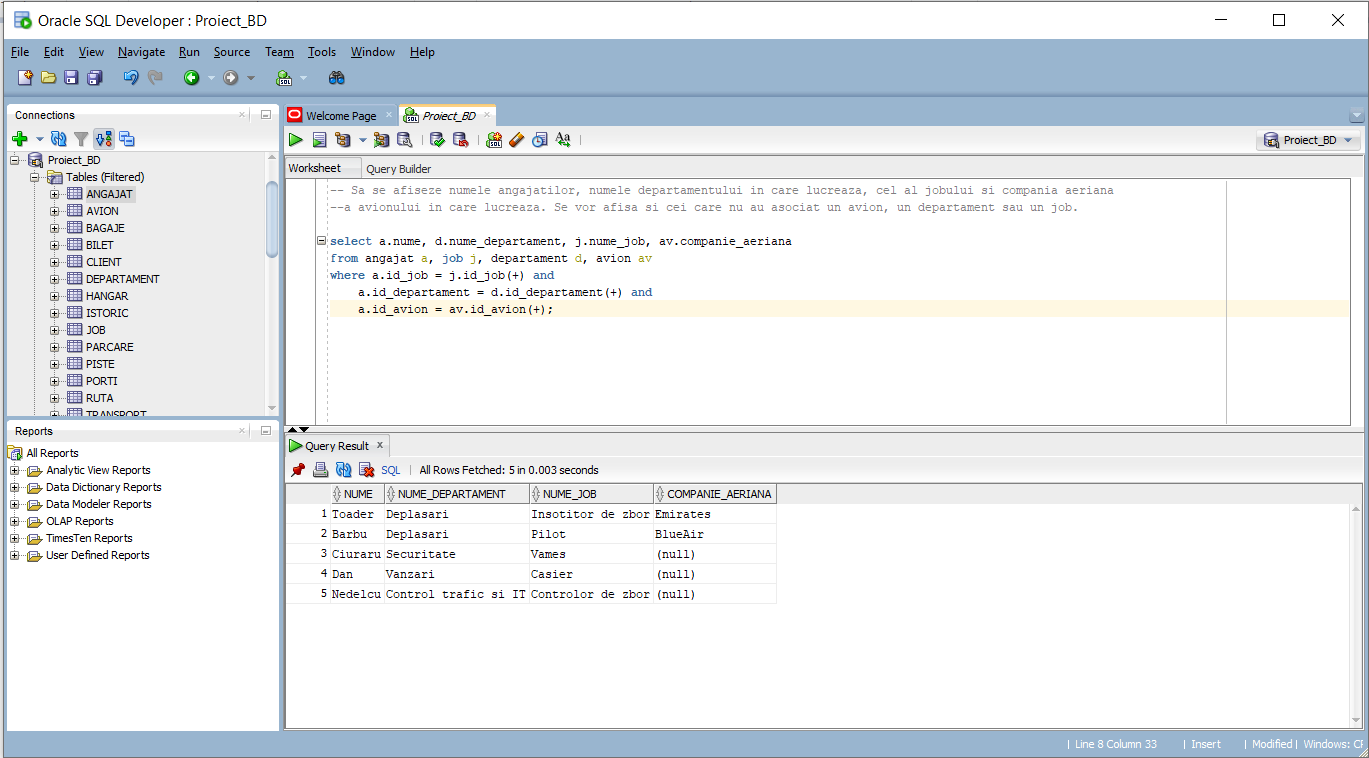
**16. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele și două cereri ce utilizează operația division.**

**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.**

**18. a. Realizarea normalizării BCNF, FN4, FN5.**

**b. Aplicarea denormalizării, justificând necesitatea acesteia.**

Baza de date prezentata pana acum este in BCNF, deoarece pentru orice dependenta intre doua atribute A si B (A->B), nu exista A non-cheie primara daca B este cheie primara.

****