

Proiect Sisteme de Gestiune a Bazelor de Date

Gestiunea unui lanț de grădini zoologice

Atasie Oana-Andreea

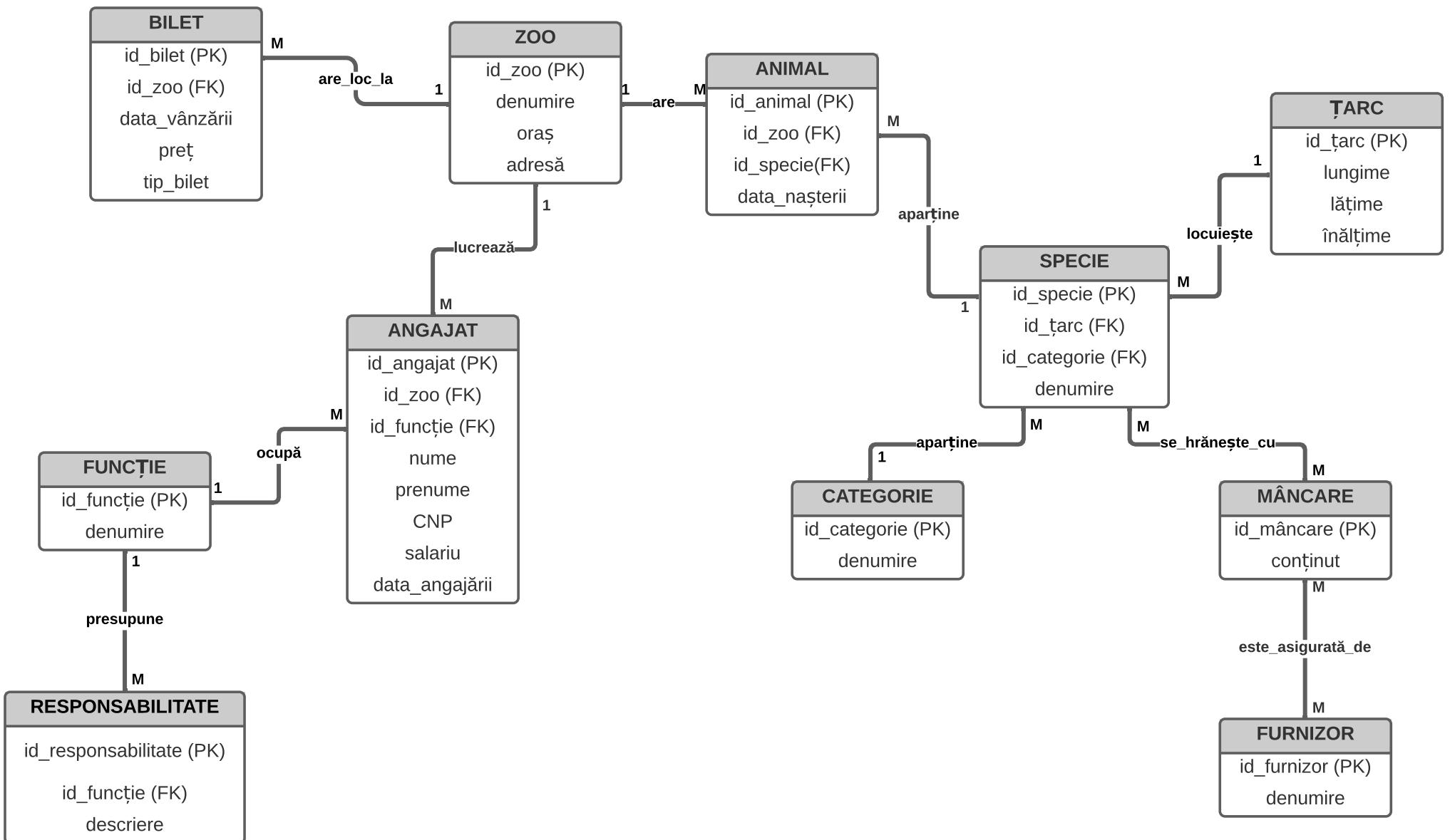
Grupa 241

1. Prezentați pe scurt baza de date (utilitatea ei).

Pentru un lanț de grădini zoologice se dorește crearea unei baze de date cu ajutorul căreia, pentru fiecare grădină zoologică din lanț, să se țină evidența biletelor vândute, a angajaților, a animalelor și a tipului de țarc în care acestea locuiesc, precum și mâncarea cu care se hrănesc animalele, asigurată de diversi furnizori. Vizitatorii unui zoo pot cumpăra bilete, în baza de date reținându-se data la care biletul a fost vândut și prețul acestuia. Totodată, în cadrul unui zoo lucrează mai mulți angajați, fiecare angajat fiind repartizat unei singure grădini zoologice. Fiecare angajat are o anumită funcție, putând exista mai mulți angajați cu aceeași funcție. O funcție presupune mai multe responsabilități, acestea fiind diferite de la o funcție la alta. În ceea ce privește animalele, grădina zoologică deține mai multe animale, fiecare animal aparținând unui singur zoo, nefiind transferabil de la o grădină la alta. De asemenea, fiecare animal aparține unei specii, fiecare specie cuprinzând mai multe animale. O specie se încadrează într-o singură categorie, iar o categorie presupune mai multe specii. Animalele aparținând unei specii locuiesc într-un anumit tip de țarc, însă țarcurile sunt construite astfel încât mai multe specii de animale să locuiască în ele. Fiecare specie de animale se hrănește cu mai multe tipuri de mâncare, iar un tip de mâncare reprezintă hrana mai multor specii de animale. Lanțul de grădini zoologice are hrana animalelor asigurată de furnizori, în aşa fel încât un furnizor asigură mai multe feluri de mâncare, iar un tip de mâncare este adus de mai mulți furnizori.

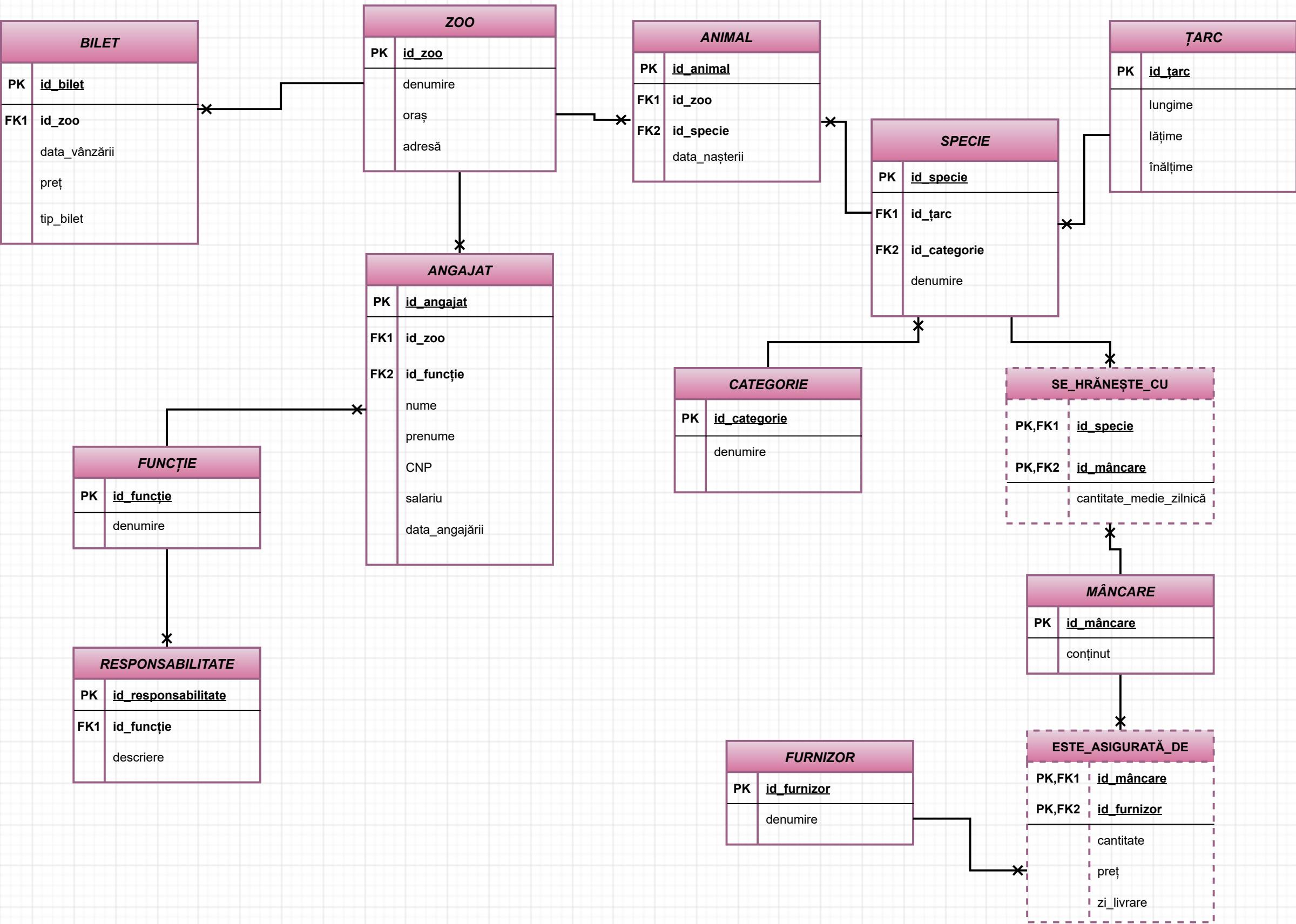
2. Realizați diagrama entitate-relație (ERD).

DIAGRAMA ENTITATE - RELAȚIE PENTRU UN LANȚ DE GRĂDINI ZOOLOGICE



3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.

DIAGRAMA CONCEPTUALĂ PENTRU UN LANT DE GRĂDINI ZOOLOGICE



4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

-- CREARE TABELE

--> TABEL ZOO

```
create table zoo
(
    id_zoo number(5) constraint pk_zoo primary key,
    denumire varchar2(50) not null,
    oras varchar2(20) not null,
    adresa varchar2(50) not null,
    constraint zoo_adresa_unique unique (adresa)
);
```

--> TABEL BILET

```
create table bilet
(
    id_bilet number(5) constraint pk_bilet primary key,
    id_zoo number(5) constraint fk_bilet_zoo references zoo(id_zoo),
    data_vanzarii date not null,
    pret number(5) not null,
    tip_bilet varchar2(50) not null
);
```

--> TABEL TARC

```
create table tarc
(
    id_tarc number(5) constraint pk_tarc primary key,
    lungime number(10,5) not null,
    latime number(10,5) not null,
    inaltime number(10,5) not null
);
```

--> TABEL CATEGORIE

```
create table categorie
(
    id_categorie number(5) constraint pk_categorie primary key,
    denumire varchar2(50) not null,
```

```

constraint categorie_denumire_unique unique (denumire)
);

--> TABEL SPECIE
create table specie
(
id_specie number(5) constraint pk_specie primary key,
id_tarc number(5) constraint fk_specie_tarc references tarc(id_tarc),
id_categorie      number(5)      constraint      fk_specie_categorie      references
categorie(id_categorie),
denumire varchar2(50) not null,
constraint specie_denumire_unique unique (denumire)
);

--> TABEL ANIMAL
create table animal
(
id_animal number(5) constraint pk_animal primary key,
id_zoo number(5) constraint fk_animal_zoo references zoo(id_zoo),
id_specie number(5) constraint fk_animal_specie references specie(id_specie),
data_nasterii date
);

--> TABEL MANCARE
create table mancare
(
id_mancare number(5) constraint pk_mancare primary key,
continut varchar2(50) not null,
constraint mancare_continut_unique unique (continut)
);

--> TABEL SE_HRANESTE_CU
create table se_hraneste_cu
(
id_specie number(5) constraint fk_se_hraneste_cu_specie references specie(id_specie),
id_mancare      number(5)      constraint      fk_se_hraneste_cu_mancare      references
mancare(id_mancare),
cantitate_medie_zilnica number(10,5) not null,
primary key(id_specie, id_mancare)
);

```

```
--> TABEL FURNIZOR
create table furnizor
(
    id_furnizor number(5) constraint pk_furnizor primary key,
    denumire varchar2(20) not null
);

--> TABEL ESTE_ASIGURATA_DE
create table este_asigurata_de
(
    id_mancare number(5) constraint fk_este_asigurata_de_mancare references
mancare(id_mancare),
    id_furnizor number(5) constraint fk_este_asigurata_de_furnizor references
furnizor(id_furnizor),
    cantitate number(5) not null,
    pret number(5) not null,
    zi_livrare number(2) not null,
    primary key(id_mancare, id_furnizor)
);

--> TABEL FUNCTIE
create table functie
(
    id_functie number(5) constraint pk_functie primary key,
    denumire varchar2(20) not null,
    constraint functie_denumire_unique unique (denumire)
);

--> TABEL ANGAJAT
create table angajat
(
    id_angajat number(5) constraint pk_angajat primary key,
    id_zoo number(5) constraint fk_angajat_zoo references zoo(id_zoo),
    id_functie number(5) constraint fk_angajat_functie references functie(id_functie),
    nume varchar2(20) not null,
    prenume varchar2(20) not null,
    CNP number(13) not null,
    constraint angajat_CNP_unique unique(CNP),
);
```

```
    salariu number(5) not null,  
    data_angajarii date not null  
);
```

--> TABEL RESPONSABILITATE

```
create table responsabilitate  
(  
    id_responsabilitate number(5) constraint pk_responsabilitate primary key,  
    id_functie      number(5)      constraint fk_responsabilitate_functie      references  
    functie(id_functie),  
    descriere varchar2(60) not null  
);
```

The screenshot shows the Oracle SQL Developer interface with a script editor window containing SQL code for creating three tables: ZOO, BILET, and TARC. The code includes constraints like primary keys and foreign keys, and data types like NUMBER and VARCHAR2. The 'Script Output' pane at the bottom shows the results of the execution.

```
-- CREARE TABLE  
--> TABEL ZOO  
create table zoo  
(  
    id_zoo number(5) constraint pk_zoo primary key,  
    denumire varchar2(50) not null,  
    oras varchar2(20) not null,  
    adresa varchar2(50) not null,  
    constraint zoo_adresa_unique unique (adresa)  
);  
  
--> TABEL BILET  
create table bilet  
(  
    id_bilet number(5) constraint pk_bilet primary key,  
    id_zoo number(5) constraint fk_bilet_zoo references zoo(id_zoo),  
    data_vanzare date not null,  
    pret number(5) not null,  
    tip_bilet varchar2(50) not null  
);  
  
--> TABEL TARC  
create table tarc
```

Script Output X | Task completed in 0.105 seconds

Table ZOO created.
Table BILET created.

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

--> TABEL TARC
create table tarc
(
    id_tarc number(5) constraint pk_tarc primary key,
    lungime number(10,5) not null,
    latime number(10,5) not null,
    inaltime number(10,5) not null
);

--> TABEL CATEGORIE
create table categorie
(
    id_categoria number(5) constraint pk_categoria primary key,
    denumire varchar2(50) not null,
    constraint categoria_denumire_unique unique (denumire)
);

--> TABEL SPECIE
create table specie
(
    id_specie number(5) constraint pk_specie primary key,
    id_tarc number(5) constraint fk_specie_tarc references tarc(id_tarc),
    id_categoria number(5) constraint fk_specie_categoria references categorie(id_categoria),
    denumire varchar2(50) not null,
    constraint specie_denumire_unique unique (denumire)
);

```

Script Output x | Task completed in 0.101 seconds

```

Table TARC created.

Table CATEGORIE created.

Table SPECIE created.

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

--> TABEL ANIMAL
create table animal
(
    id_animal number(5) constraint pk_animal primary key,
    id_zoo number(5) constraint fk_animal_zoo references zoo(id_zoo),
    id_specie number(5) constraint fk_animal_specie references specie(id_specie),
    data_nasterii date
);

--> TABEL MANCARE
create table mancare
(
    id_mancare number(5) constraint pk_mancare primary key,
    continut varchar2(50) not null,
    constraint mancare_continut_unique unique (continut)
);

--> TABEL SE_HRANESTE CU
create table se_hrane_cu
(
    id_specie number(5) constraint fk_se_hrane_cu_specie references specie(id_specie),
    id_mancare number(5) constraint fk_se_hrane_cu_mancare references mancare(id_mancare),
    cantitate_medie_zilnica number(10,5) not null,
    primary key(id_specie, id_mancare)
);

```

Script Output x | Task completed in 0.12 seconds

```

Table ANIMAL created.

Table MANCARE created.

Table SE_HRANESTE CU created.

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

File Edit View Navigate Run Source Team Tools Window Help
File Project Database Schema Services
SQL Worksheet History
Atasie_Oana_SQL.sql
Worksheet Query Builder
project

id_mancare number(5) constraint fk_se_hraneste_cu_mancare references mancare(id_mancare),
cantitate_medie_zilnica number(10,5) not null,
primary key(id_specie, id_mancare)
);

--> TABEL FURNIZOR
create table furnizor
(
id_furnizor number(5) constraint pk_furnizor primary key,
denumire varchar2(20) not null
);

--> TABEL ESTE_ASIGURATA_DE
create table este_asigurata_de
(
id_mancare number(5) constraint fk_este_asigurata_de_mancare references mancare(id_mancare),
id_furnizor number(5) constraint fk_este_asigurata_de_furnizor references furnizor(id_furnizor),
cantitate number(5) not null,
pret number(5) not null,
zi_livrare number(2) not null,
primary key(id_mancare, id_furnizor)
);

--> TABEL FUNCTIE
create table functie
;

Script Output X | Task completed in 0.064 seconds
Table FURNIZOR created.

Table ESTE_ASIGURATA_DE created.


```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

File Edit View Navigate Run Source Team Tools Window Help
File Project Database Schema Services
SQL Worksheet History
Atasie_Oana_SQL.sql
Worksheet Query Builder
project

create table functie
(
id_functie number(5) constraint pk_functie primary key,
denumire varchar2(20) not null,
constraint functie_denumire_unique unique (denumire)
);

--> TABEL ANGAJAT
create table angajat
(
id_angajat number(5) constraint pk_angajat primary key,
id_zoo number(5) constraint fk_angajat_zoo references zoo(id_zoo),
id_functie number(5) constraint fk_angajat_functie references functie(id_functie),
nume varchar2(20) not null,
prenume varchar2(20) not null,
CNP number(13) not null,
constraint angajat_CNP_unique unique(CNP),
salarii number(5) not null,
data_angajarii date not null
);

--> TABEL RESPONSABILITATE
create table responsabilitate
(
id_responsabilitate number(5) constraint pk_responsabilitate primary key,
id_functie number(5) constraint fk_responsabilitate_functie references functie(id_functie),
descriere varchar2(60) not null
);

Script Output X | Task completed in 0.115 seconds
Table FUNCTIE created.

Table ANGAJAT created.

Table RESPONSABILITATE created.


```

5. Adăugați informații coerente în tabelele create (minim 3-5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

-- INSERARE VALORI

-- TABEL ZOO

```
INSERT INTO ZOO VALUES(1,'Gradina Zoologica din Bucuresti','Bucuresti','Strada Vadul Moldovei 4');
```

```
INSERT INTO ZOO VALUES(2,'Gradina Zoologica din Timisoara','Timisoara','Strada Avram Imbroane 90');
```

```
INSERT INTO ZOO VALUES(3,'Gradina Zoologica din Oradea','Oradea','Calea Matei Basarab 1');
```

```
INSERT INTO ZOO VALUES(4,'Gradina Zoologica din Resita','Resita','Strada Alexandru Nedelea 7');
```

```
INSERT INTO ZOO VALUES(5,'Gradina Zoologica din Targu Mures','Targu Mures','Strada Verii 57');
```

-- TABEL BILET

```
INSERT INTO BILET VALUES(1,5,to_date('08/04/2020','dd/mm/yyyy'),12,'Persoane cu dizabilitati');
```

```
INSERT INTO BILET VALUES(2,2,to_date('11/04/2020','dd/mm/yyyy'),20,'Adulti');
```

```
INSERT INTO BILET VALUES(3,3,to_date('17/03/2020','dd/mm/yyyy'),14,'Studenti');
```

```
INSERT INTO BILET VALUES(4,4,to_date('02/12/2020','dd/mm/yyyy'),15,'Grup minim 15 adulți');
```

```
INSERT INTO BILET VALUES(5,4,to_date('22/02/2020','dd/mm/yyyy'),7,'Grup minim 15 copii și adulți');
```

```
INSERT INTO BILET VALUES(6,1,to_date('06/09/2020','dd/mm/yyyy'),0,'Copii pana la 3 ani');
```

```
INSERT INTO BILET VALUES(7,4,to_date('13/02/2020','dd/mm/yyyy'),8,'Persoane cu dizabilitati');
```

```
INSERT INTO BILET VALUES(8,2,to_date('26/04/2020','dd/mm/yyyy'),9,'Grup minim 15 copii și adulți');
```

```
INSERT INTO BILET VALUES(9,3,to_date('14/11/2020','dd/mm/yyyy'),14,'Elevi');
```

```
INSERT INTO BILET VALUES(10,2,to_date('28/06/2020','dd/mm/yyyy'),15,'Grup minim 15 adulți');
```

```
INSERT INTO BILET VALUES(11,1,to_date('20/03/2020','dd/mm/yyyy'),18,'Adulti');
```

```
INSERT INTO BILET VALUES(12,3,to_date('28/11/2020','dd/mm/yyyy'),5,'Persoane peste 75 ani');
```

```
INSERT INTO BILET VALUES(13,5,to_date('08/09/2020','dd/mm/yyyy'),12,'Studenti');
```

```
INSERT INTO BILET VALUES(14,2,to_date('20/01/2020','dd/mm/yyyy'),20,'Adulti');
INSERT INTO BILET VALUES(15,4,to_date('05/07/2020','dd/mm/yyyy'),17,'Adulti');
INSERT INTO BILET VALUES(16,5,to_date('22/07/2020','dd/mm/yyyy'),21,'Adulti');
INSERT INTO BILET VALUES(17,1,to_date('25/05/2020','dd/mm/yyyy'),0,'Persoane
peste 75 ani');
INSERT INTO BILET VALUES(18,1,to_date('15/11/2020','dd/mm/yyyy'),9,'Studenti');
INSERT INTO BILET VALUES(19,4,to_date('24/10/2020','dd/mm/yyyy'),8,'Persoane cu
dizabilitati');
INSERT INTO BILET VALUES(20,2,to_date('01/01/2020','dd/mm/yyyy'),10,'Studenti');
INSERT INTO BILET VALUES(21,1,to_date('23/05/2020','dd/mm/yyyy'),9,'Persoane cu
dizabilitati');
INSERT INTO BILET VALUES(22,5,to_date('24/04/2020','dd/mm/yyyy'),3,'Persoane
peste 75 ani');
INSERT INTO BILET VALUES(23,1,to_date('16/07/2020','dd/mm/yyyy'),9,'Studenti');
INSERT INTO BILET VALUES(24,2,to_date('07/07/2020','dd/mm/yyyy'),15,'Grup
minim 15 adulți');
INSERT INTO BILET VALUES(25,1,to_date('08/05/2020','dd/mm/yyyy'),0,'Copii pana
la 3 ani');
INSERT INTO BILET VALUES(26,2,to_date('30/03/2020','dd/mm/yyyy'),0,'Persoane
peste 75 ani');
INSERT INTO BILET VALUES(27,4,to_date('23/09/2020','dd/mm/yyyy'),17,'Adulti');
INSERT INTO BILET VALUES(28,1,to_date('15/06/2020','dd/mm/yyyy'),8,'Grup
minim 15 copii si adulți');
INSERT INTO BILET VALUES(29,3,to_date('05/06/2020','dd/mm/yyyy'),13,'Grup
minim 15 copii si adulți');
INSERT INTO BILET VALUES(30,4,to_date('08/01/2020','dd/mm/yyyy'),7,'Grup
minim 15 copii si adulți');
INSERT INTO BILET VALUES(31,3,to_date('11/01/2020','dd/mm/yyyy'),23,'Adulti');
INSERT INTO BILET VALUES(32,1,to_date('16/02/2020','dd/mm/yyyy'),16,'Grup
minim 15 adulți');
INSERT INTO BILET VALUES(33,1,to_date('01/12/2020','dd/mm/yyyy'),0,'Studenti');
INSERT INTO BILET VALUES(34,3,to_date('29/11/2020','dd/mm/yyyy'),5,'Copii pana
la 3 ani');
INSERT INTO BILET VALUES(35,2,to_date('02/01/2020','dd/mm/yyyy'),9,'Grup
minim 15 copii si adulți');
INSERT INTO BILET VALUES(36,4,to_date('01/06/2020','dd/mm/yyyy'),0,'Copii intre
3 si 18 ani');
INSERT INTO BILET VALUES(37,4,to_date('01/06/2020','dd/mm/yyyy'),0,'Copii intre
3 si 18 ani');
```

```
INSERT INTO BILET VALUES(38,4,to_date('28/02/2020','dd/mm/yyyy'),6,'Copii intre  
3 si 18 ani');  
INSERT INTO BILET VALUES(39,1,to_date('15/06/2020','dd/mm/yyyy'),0,'Copii pana  
la 3 ani');  
INSERT INTO BILET VALUES(40,1,to_date('08/05/2020','dd/mm/yyyy'),16,'Grup  
minim 15 adulți');  
INSERT INTO BILET VALUES(41,1,to_date('08/05/2020','dd/mm/yyyy'),16,'Grup  
minim 15 adulți');  
INSERT INTO BILET VALUES(42,2,to_date('20/01/2020','dd/mm/yyyy'),20,'Adulti');  
INSERT INTO BILET VALUES(43,4,to_date('08/02/2020','dd/mm/yyyy'),7,'Grup  
minim 15 copii si adulți');  
INSERT INTO BILET VALUES(44,2,to_date('20/01/2020','dd/mm/yyyy'),20,'Adulti');  
INSERT INTO BILET VALUES(45,5,to_date('08/04/2020','dd/mm/yyyy'),12,'Persoane  
cu dizabilitati');  
INSERT INTO BILET VALUES(46,3,to_date('29/11/2020','dd/mm/yyyy'),5,'Copii pana  
la 3 ani');  
INSERT INTO BILET VALUES(47,5,to_date('08/04/2020','dd/mm/yyyy'),12,'Persoane  
cu dizabilitati');  
INSERT INTO BILET VALUES(48,3,to_date('29/11/2020','dd/mm/yyyy'),5,'Copii pana  
la 3 ani');  
INSERT INTO BILET VALUES(49,3,to_date('28/11/2020','dd/mm/yyyy'),5,'Persoane  
peste 75 ani');  
INSERT INTO BILET VALUES(50,3,to_date('28/11/2020','dd/mm/yyyy'),5,'Persoane  
peste 75 ani');  
INSERT INTO BILET VALUES(51,3,to_date('28/11/2020','dd/mm/yyyy'),5,'Persoane  
peste 75 ani');  
INSERT INTO BILET VALUES(52,1,to_date('03/05/2020','dd/mm/yyyy'),6,'Elevi');  
INSERT INTO BILET VALUES(53,1,to_date('03/05/2020','dd/mm/yyyy'),16,'Grup  
minim 15 adulți');
```

-- TABEL FUNCTIE

```
INSERT INTO FUNCTIE VALUES(1,'Vanzator');  
INSERT INTO FUNCTIE VALUES(2,'Ingrijitor');  
INSERT INTO FUNCTIE VALUES(3,'Securitate');  
INSERT INTO FUNCTIE VALUES(4,'Director General');  
INSERT INTO FUNCTIE VALUES(5,'Veterinar');
```

-- TABEL ANGAJAT

```
INSERT INTO ANGAJAT
VALUES(1,5,3,'Dobre','Elena',2811025074180,4091,to_date('04/09/2018','dd/mm/yyyy'))
);
INSERT INTO ANGAJAT
VALUES(2,5,2,'Fratila','Mihai',1811015074130,4082,to_date('16/07/2020','dd/mm/yyyy'))
);
INSERT INTO ANGAJAT
VALUES(3,3,2,'Voinea','Teodor',1811015074138,4910,to_date('03/05/2015','dd/mm/yyyy'));
);
INSERT INTO ANGAJAT
VALUES(4,2,2,'Iancu','Marcel',1811015074167,4277,to_date('29/12/2019','dd/mm/yyyy'));
);
INSERT INTO ANGAJAT
VALUES(5,4,3,'Dumitrescu','Valentina',2811025074184,5936,to_date('24/06/2019','dd/m
m/yyyy'));
);
INSERT INTO ANGAJAT
VALUES(6,4,2,'Rusu','Andreea',2811025074149,3689,to_date('29/10/2017','dd/mm/yyyy'
));
);
INSERT INTO ANGAJAT
VALUES(7,3,2,'Ifrim','Ionela',2811025074173,3999,to_date('25/08/2017','dd/mm/yyyy'))
;
);
INSERT INTO ANGAJAT
VALUES(8,2,2,'Cristea','Olivia',2811025074142,5039,to_date('06/09/2017','dd/mm/yyyy'
));
);
INSERT INTO ANGAJAT
VALUES(9,5,2,'Dima','Ioan',1811015074175,5345,to_date('17/12/2016','dd/mm/yyyy'));
);
INSERT INTO ANGAJAT
VALUES(10,2,3,'Nistor','Viorel',1811015074181,5740,to_date('19/07/2014','dd/mm/yyy
y'));
);
INSERT INTO ANGAJAT
VALUES(11,5,1,'Nita','Diana',2811025074115,5484,to_date('26/05/2017','dd/mm/yyyy'))
;
);
INSERT INTO ANGAJAT
VALUES(12,1,2,'Sava','Viviana',2811025074156,4518,to_date('23/08/2016','dd/mm/yyy
y'));
);
INSERT INTO ANGAJAT
VALUES(13,4,2,'Florea','Mihaela',2811025074129,3342,to_date('20/10/2018','dd/mm/yy
y'));
```

```
INSERT INTO ANGAJAT
VALUES(14,2,2,'Sava','Diana',2811025074147,3974,to_date('15/12/2016','dd/mm/yyyy'))
);
INSERT INTO ANGAJAT
VALUES(15,5,3,'Dima','Viviana',2811025074185,4627,to_date('03/06/2018','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(16,3,3,'Marin','Andreea',2811025074178,5349,to_date('10/06/2020','dd/mm/yyyy'));
;
INSERT INTO ANGAJAT
VALUES(17,3,2,'Stan','Maria',2811025074118,3291,to_date('09/10/2016','dd/mm/yyyy')) ;
INSERT INTO ANGAJAT
VALUES(18,3,2,'Moldoveanu','Viorel',1811015074153,4045,to_date('03/03/2016','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(19,1,2,'Calinescu','Mihaela',2811025074161,3935,to_date('05/04/2019','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(20,4,2,'Marin','Viviana',2811025074124,3520,to_date('16/12/2019','dd/mm/yyyy'));
;
INSERT INTO ANGAJAT
VALUES(21,2,2,'Constantinescu','Andreea',2811025074117,5237,to_date('15/07/2016','dd/mm/yyyy'));
;
INSERT INTO ANGAJAT
VALUES(22,2,5,'Nistor','Cristian',1811015074191,9719,to_date('04/05/2018','dd/mm/yyyy'));
;
INSERT INTO ANGAJAT
VALUES(23,2,4,'Ionita','Iulian',1811015074187,9594,to_date('26/07/2014','dd/mm/yyyy')) ;
;
INSERT INTO ANGAJAT
VALUES(24,3,2,'Teodorescu','Mircea',1811015074168,3403,to_date('17/01/2019','dd/mm/yyyy'));
;
INSERT INTO ANGAJAT
VALUES(25,5,2,'Sava','Daniel',1811015074165,4287,to_date('02/04/2016','dd/mm/yyyy')) ;
;
INSERT INTO ANGAJAT
VALUES(26,1,2,'Diaconescu','Valentin',1811015074136,4764,to_date('14/10/2020','dd/mm/yyyy')) ;
```

```
INSERT INTO ANGAJAT
VALUES(27,4,2,'Stoica','Cosmina',2811025074134,4723,to_date('05/01/2018','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(28,2,3,'Ionita','Oana',2811025074177,3057,to_date('18/01/2018','dd/mm/yyyy'))
);
INSERT INTO ANGAJAT
VALUES(29,2,2,'Gheorghiu','Valentin',1811015074111,4580,to_date('13/07/2015','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(30,3,2,'Ionita','Valentina',2811025074123,3677,to_date('13/10/2014','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(31,5,2,'Moisescu','Cristina',2811025074160,3239,to_date('21/04/2016','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(32,2,2,'Negoita','Elena',2811025074172,4834,to_date('09/10/2018','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(33,3,2,'Ciobanu','Oana',2811025074148,4333,to_date('22/08/2020','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(34,4,2,'Stanescu','Ionela',2811025074144,5602,to_date('10/10/2016','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(35,1,2,'Nitu','Oana',2811025074116,3011,to_date('10/04/2016','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(36,5,2,'Calinescu','Iulia',2811025074170,3112,to_date('17/02/2019','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(37,2,2,'Munteanu','Veronica',2811025074152,3658,to_date('30/04/2014','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(38,2,2,'Barbu','Cristian',1811015074117,3989,to_date('17/10/2020','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(39,4,5,'Marin','Mihaela',2811025074194,6567,to_date('20/06/2018','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(40,4,1,'Pop','Ioan',1811015074114,5126,to_date('08/04/2018','dd/mm/yyyy'));
```

```
INSERT INTO ANGAJAT
VALUES(41,1,5,'Barbu','Mihai',1811015074178,5949,to_date('03/06/2017','dd/mm/yyyy'))
));
INSERT INTO ANGAJAT
VALUES(42,1,3,'Barbu','Veronica',2811025074181,4922,to_date('08/11/2018','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(43,5,2,'Diaconu','Viorica',2811025074135,3672,to_date('03/08/2015','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(44,4,2,'Teodorescu','Cristian',1811015074159,3721,to_date('19/05/2017','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(45,5,5,'Tudor','Ioana',2811025074195,5225,to_date('18/05/2016','dd/mm/yyyy'));
);
INSERT INTO ANGAJAT
VALUES(46,3,5,'Ionita','Cristina',2811025074193,8507,to_date('18/05/2016','dd/mm/yyyy'));
);
INSERT INTO ANGAJAT
VALUES(47,2,2,'Mihăilescu','Doina',2811025074157,4216,to_date('20/04/2015','dd/mm/yyyy'));
);
INSERT INTO ANGAJAT
VALUES(48,5,2,'Ursu','Maria',2811025074150,3123,to_date('20/07/2016','dd/mm/yyyy'));
);
INSERT INTO ANGAJAT
VALUES(49,1,2,'Manole','Roxana',2811025074141,3626,to_date('07/11/2014','dd/mm/yyyy'));
);
INSERT INTO ANGAJAT
VALUES(50,5,4,'Dobre','Mihaela',2811025074190,6004,to_date('18/11/2017','dd/mm/yyyy'));
);
INSERT INTO ANGAJAT
VALUES(51,1,1,'Dobre','Iulian',1811015074116,3662,to_date('14/04/2020','dd/mm/yyyy'));
);
INSERT INTO ANGAJAT
VALUES(52,1,4,'Dumitrescu','Ion',1811015074174,4449,to_date('02/10/2020','dd/mm/yyyy'));
);
INSERT INTO ANGAJAT
VALUES(53,1,5,'Dinu','Vladimir',1811015074115,4680,to_date('11/08/2015','dd/mm/yyyy'));
```

```
INSERT INTO ANGAJAT
VALUES(54,1,3,'Lupu','Elena',2811025074151,5641,to_date('16/11/2017','dd/mm/yyyy'))
);
INSERT INTO ANGAJAT
VALUES(55,1,5,'Iancu','Iulian',1811015074158,3343,to_date('09/08/2017','dd/mm/yyyy'))
));
INSERT INTO ANGAJAT
VALUES(56,2,1,'Dinescu','Vlad',1811015074131,4563,to_date('06/04/2014','dd/mm/yyyy'));
)
INSERT INTO ANGAJAT
VALUES(57,2,1,'Dumitrescu','Veronica',2811025074120,3469,to_date('17/01/2016','dd/mm/yyyy'));
)
INSERT INTO ANGAJAT
VALUES(58,2,4,'Andreescu','Valentina',2811025074155,4084,to_date('12/02/2020','dd/mm/yyyy'));
)
INSERT INTO ANGAJAT
VALUES(59,2,5,'Gheorghiu','Diana',2811025074176,5700,to_date('27/01/2015','dd/mm/yyyy'));
)
INSERT INTO ANGAJAT
VALUES(60,3,1,'Tabacu','Ionut',1811015074171,3739,to_date('17/08/2017','dd/mm/yyyy'));
)
INSERT INTO ANGAJAT
VALUES(61,3,5,'Moisescu','Catinca',2811025074169,4248,to_date('08/03/2019','dd/mm/yyyy'));
)
INSERT INTO ANGAJAT
VALUES(62,3,3,'Tudor','Doina',2811025074125,3871,to_date('28/02/2020','dd/mm/yyyy'));
)
INSERT INTO ANGAJAT
VALUES(63,3,4,'Tudor','Cristina',2811025074189,9966,to_date('19/05/2018','dd/mm/yyyy'));
)
INSERT INTO ANGAJAT
VALUES(64,3,4,'Mihailescu','Dumitru',1811015074166,4394,to_date('21/01/2015','dd/mm/yyyy'));
)
INSERT INTO ANGAJAT
VALUES(65,4,4,'Marin','Cristian',1811015074188,6320,to_date('03/02/2016','dd/mm/yyyy'));
)
INSERT INTO ANGAJAT
VALUES(66,4,4,'Nistor','Cristina',2811025074128,5060,to_date('08/05/2020','dd/mm/yyyy'));
```

```
INSERT INTO ANGAJAT
VALUES(67,4,1,'Stanciu','Elena',2811025074119,3653,to_date('25/08/2017','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(68,4,5,'Muresan','Valentin',1811015074154,5288,to_date('21/09/2016','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(69,4,1,'Popescu','Elena',2961025074112,5814,to_date('14/05/2017','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(70,4,3,'Petrescu','Rodica',2811025074140,5842,to_date('28/07/2020','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(71,4,3,'Georgescu','Cosmin',1811015074133,4064,to_date('30/11/2016','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(72,5,3,'Dobre','Valentin',1811015074183,3921,to_date('06/02/2017','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(73,5,4,'Ionescu','Ionela',2811025074112,5223,to_date('28/12/2018','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(74,5,1,'Dima','Viorel',1811015074121,5030,to_date('05/06/2015','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(75,5,5,'Toma','Elena',2811025074143,5621,to_date('23/04/2020','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(76,5,5,'Mocanu','Tudor',1811015074137,3109,to_date('03/04/2019','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(77,5,3,'Puscasu','Ion',1811015074145,4457,to_date('27/03/2020','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(78,1,5,'Ionescu','Valentin',2913025074141,3050,to_date('07/11/2014','dd/mm/yyyy'));
INSERT INTO ANGAJAT
VALUES(79,5,2,'Vladimirescu','Ionela',2811017079121,4000,to_date('05/06/2015','dd/mm/yyyy'));
```

```
INSERT INTO ANGAJAT  
VALUES(80,4,1,'Prisacariu','Cristian',1822015074188,7132,to_date('03/02/2016','dd/mm/  
yyyy'));  
INSERT INTO ANGAJAT  
VALUES(81,4,2,'Hitanu','Erika',2922015074165,5721,to_date('03/02/2016','dd/mm/yyyy'  
));
```

-- TABEL RESPONSABILITATE

```
INSERT INTO RESPONSABILITATE VALUES(1,1,'relationarea cu clientii');  
INSERT INTO RESPONSABILITATE VALUES(2,1,'manipulare bilete');  
INSERT INTO RESPONSABILITATE VALUES(3,1,'prezentarea ofertei');  
INSERT INTO RESPONSABILITATE VALUES(4,2,'curatarea tarcurilor');  
INSERT INTO RESPONSABILITATE VALUES(5,2,'inspectarea tarcurilor');  
INSERT INTO RESPONSABILITATE VALUES(6,2,'hranirea animalelor');  
INSERT INTO RESPONSABILITATE VALUES(7,3,'supravegherea camerelor de luat  
vederi');  
INSERT INTO RESPONSABILITATE VALUES(8,3,'supravegherea zonelor de acces in  
gradina zoologica');  
INSERT INTO RESPONSABILITATE VALUES(9,3,'interventia in caz de nerespectare  
a regulilor');  
INSERT INTO RESPONSABILITATE VALUES(10,4,'gestionarea tuturor situatiilor  
neprevazute');  
INSERT INTO RESPONSABILITATE VALUES(11,4,'gestionarea bugetului');  
INSERT INTO RESPONSABILITATE VALUES(12,4,'time management');  
INSERT INTO RESPONSABILITATE VALUES(13,5,'controlul saptamanal al  
animalelor');  
INSERT INTO RESPONSABILITATE VALUES(14,5,'tratarea animalelor in caz de  
boala');
```

-- TABEL MANCARE

```
INSERT INTO MANCARE VALUES(1,'radacini, fructe de padure, oua, larve');  
INSERT INTO MANCARE VALUES(2,'carne');  
INSERT INTO MANCARE VALUES(3,'furaje');  
INSERT INTO MANCARE VALUES(4,'fructe, frunze');  
INSERT INTO MANCARE VALUES(5,'ierburi,frunze verzi');  
INSERT INTO MANCARE VALUES(6,'carne, fructe de padure');  
INSERT INTO MANCARE VALUES(7,'carne, furaje');  
INSERT INTO MANCARE VALUES(8,'granule');  
INSERT INTO MANCARE VALUES(9,'cereale');
```

```
-- TABEL FURNIZOR
INSERT INTO FURNIZOR VALUES(1,'Food for Animals');
INSERT INTO FURNIZOR VALUES(2,'Animal Food');
INSERT INTO FURNIZOR VALUES(3,'Ferma Ionescu');
INSERT INTO FURNIZOR VALUES(4,'TetraMin');
INSERT INTO FURNIZOR VALUES(5,'Global Food');
INSERT INTO FURNIZOR VALUES(6,'Animal Love');
INSERT INTO FURNIZOR VALUES(7,'American Zoo');
INSERT INTO FURNIZOR VALUES(8,'Ferma Corog');
INSERT INTO FURNIZOR VALUES(9,'TetraMax');
INSERT INTO FURNIZOR VALUES(10,'Ferma Alpin');
INSERT INTO FURNIZOR VALUES(11,'Global Food');
INSERT INTO FURNIZOR VALUES(12,'Food Food');
```

```
-- TABEL ESTE_ASIGURATA_DE
INSERT INTO ESTE_ASIGURATA_DE VALUES(1,1,22500,6,11);
INSERT INTO ESTE_ASIGURATA_DE VALUES(2,2,67500,14,10);
INSERT INTO ESTE_ASIGURATA_DE VALUES(3,3,22500,12,13);
INSERT INTO ESTE_ASIGURATA_DE VALUES(4,4,13500,11,9);
INSERT INTO ESTE_ASIGURATA_DE VALUES(5,5,11250,6,13);
INSERT INTO ESTE_ASIGURATA_DE VALUES(6,6,45000,6,14);
INSERT INTO ESTE_ASIGURATA_DE VALUES(7,7,22500,8,11);
INSERT INTO ESTE_ASIGURATA_DE VALUES(8,8,67500,10,8);
INSERT INTO ESTE_ASIGURATA_DE VALUES(9,9,1350,13,17);
INSERT INTO ESTE_ASIGURATA_DE VALUES(7,10,450,8,13);
INSERT INTO ESTE_ASIGURATA_DE VALUES(1,6,9000,6,20);
INSERT INTO ESTE_ASIGURATA_DE VALUES(2,7,7500,14,1);
INSERT INTO ESTE_ASIGURATA_DE VALUES(3,8,30000,8,16);
INSERT INTO ESTE_ASIGURATA_DE VALUES(4,9,15000,15,19);
INSERT INTO ESTE_ASIGURATA_DE VALUES(5,10,45000,10,10);
INSERT INTO ESTE_ASIGURATA_DE VALUES(4,7,0,7,3);
```

```
-- TABEL TARC
INSERT INTO TARC VALUES(1,500,250,300);
INSERT INTO TARC VALUES(2,600,600,400);
INSERT INTO TARC VALUES(3,500,450,400);
INSERT INTO TARC VALUES(4,200,300,500);
INSERT INTO TARC VALUES(5,300,300,300);
INSERT INTO TARC VALUES(6,450,300,300);
INSERT INTO TARC VALUES(7,400,400,300);
```

```
INSERT INTO TARC VALUES(8,500,180,250);
INSERT INTO TARC VALUES(9,150,150,200);
INSERT INTO TARC VALUES(10,300,400,400);
INSERT INTO TARC VALUES(11,400,150,150);
INSERT INTO TARC VALUES(12,300,300,100);
INSERT INTO TARC VALUES(13,250,240,150);
INSERT INTO TARC VALUES(14,200,250,90);
```

-- TABEL CATEGORIE

```
INSERT INTO CATEGORIE VALUES(1, 'Animal omnivor salbatic');
INSERT INTO CATEGORIE VALUES(2, 'Animale carnivore periculoase');
INSERT INTO CATEGORIE VALUES(3, 'Cabaline');
INSERT INTO CATEGORIE VALUES(4, 'Primate');
INSERT INTO CATEGORIE VALUES(5, 'Animale erbivore Africa');
INSERT INTO CATEGORIE VALUES(6, 'Ecvidee africane');
INSERT INTO CATEGORIE VALUES(7, 'Reptile');
INSERT INTO CATEGORIE VALUES(8, 'Pesti');
INSERT INTO CATEGORIE VALUES(9, 'Pasari');
INSERT INTO CATEGORIE VALUES(10,'Insectivore');
```

-- TABEL SPECIE

```
INSERT INTO SPECIE VALUES(1,10,9,'Papagal cenusiu');
INSERT INTO SPECIE VALUES(2,1,1,'Porc mistret');
INSERT INTO SPECIE VALUES(3,8,7,'Crocodil');
INSERT INTO SPECIE VALUES(4,10,9,'Perus');
INSERT INTO SPECIE VALUES(5,9,8,'Sanitari');
INSERT INTO SPECIE VALUES(6,9,8,'Killi');
INSERT INTO SPECIE VALUES(7,2,2,'Leu');
INSERT INTO SPECIE VALUES(8,13,7,'Testoasa terestra');
INSERT INTO SPECIE VALUES(9,3,3,'Ponei');
INSERT INTO SPECIE VALUES(10,4,4,'Cimpanzeu');
INSERT INTO SPECIE VALUES(11,9,8,'Discus');
INSERT INTO SPECIE VALUES(12,3,5,'Camila');
INSERT INTO SPECIE VALUES(13,7,6,'Zebra');
INSERT INTO SPECIE VALUES(14,3,5,'Dromader');
INSERT INTO SPECIE VALUES(15,3,3,'Armasar');
INSERT INTO SPECIE VALUES(16,6,1,'Urs brun');
INSERT INTO SPECIE VALUES(17,12,7,'Aigialosauridae');
INSERT INTO SPECIE VALUES(18,11,7,'Piton');
INSERT INTO SPECIE VALUES(19,9,7,'Cameleon');
```

```
INSERT INTO SPECIE VALUES(20,11,7,'Viperide');
INSERT INTO SPECIE VALUES(21,2,2,'Tigru');
INSERT INTO SPECIE VALUES(22,5,4,'Gorila');
INSERT INTO SPECIE VALUES(23,4,4,'Capucin');
INSERT INTO SPECIE VALUES(24,14,7,'Iguana');
INSERT INTO SPECIE VALUES(25,null,10,'Arici');
```

-- TABEL ANIMAL

```
INSERT INTO ANIMAL VALUES(1,3,1,to_date('03/03/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(2,4,2,to_date('20/11/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(3,3,3,to_date('06/02/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(4,2,4,to_date('31/07/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(5,1,5,to_date('14/05/2018','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(6,3,6,to_date('17/09/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(7,1,3,to_date('14/04/2020','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(8,5,7,to_date('16/06/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(9,1,8,to_date('20/02/2019','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(10,2,4,to_date('13/03/2019','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(11,3,2,to_date('05/08/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(12,5,9,to_date('09/04/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(13,1,7,to_date('03/11/2018','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(14,3,10,to_date('17/08/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(15,3,11,to_date('27/03/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(16,1,1,to_date('24/05/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(17,4,9,to_date('14/08/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(18,2,11,to_date('10/05/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(19,5,11,to_date('11/10/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(20,2,2,to_date('01/06/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(21,5,12,to_date('31/12/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(22,2,8,to_date('01/09/2018','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(23,3,13,to_date('23/03/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(24,1,5,to_date('05/01/2018','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(25,4,14,to_date('04/05/2020','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(26,3,4,to_date('06/04/2020','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(27,4,13,to_date('23/11/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(28,3,1,to_date('08/12/2020','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(29,1,16,to_date('04/01/2019','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(30,4,6,to_date('30/03/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(31,3,15,to_date('17/11/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(32,2,16,to_date('25/05/2020','dd/mm/yyyy'));
```

```
INSERT INTO ANIMAL VALUES(33,2,3,to_date('22/05/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(34,5,17,to_date('01/06/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(35,4,18,to_date('27/03/2020','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(36,1,3,to_date('07/08/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(37,5,16,to_date('12/09/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(38,4,19,to_date('01/11/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(39,5,4,to_date('11/12/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(40,5,3,to_date('05/06/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(41,3,4,to_date('19/10/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(42,3,9,to_date('10/05/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(43,2,20,to_date('08/07/2018','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(44,4,15,to_date('14/02/2020','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(45,4,12,to_date('25/12/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(46,4,16,to_date('04/04/2019','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(47,5,13,to_date('16/10/2019','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(48,3,21,to_date('07/11/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(49,3,13,to_date('14/09/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(50,2,14,to_date('06/03/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(51,5,14,to_date('05/07/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(52,2,7,to_date('20/11/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(53,1,2,to_date('19/08/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(54,3,12,to_date('24/11/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(55,2,22,to_date('11/03/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(56,2,4,to_date('11/03/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(57,3,12,to_date('29/12/2013','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(58,4,4,to_date('14/04/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(59,1,14,to_date('01/03/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(60,3,4,to_date('23/08/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(61,1,13,to_date('15/11/2020','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(62,5,2,to_date('26/11/2019','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(63,1,21,to_date('23/11/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(64,4,3,to_date('30/05/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(65,3,13,to_date('16/07/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(66,4,21,to_date('24/09/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(67,5,9,to_date('20/10/2020','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(68,3,23,to_date('28/05/2019','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(69,1,10,to_date('07/12/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(70,1,2,to_date('28/12/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(71,1,12,to_date('20/07/2019','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(72,2,22,to_date('02/05/2014','dd/mm/yyyy'));
```

```
INSERT INTO ANIMAL VALUES(73,2,13,to_date('22/08/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(74,5,13,to_date('11/10/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(75,3,2,to_date('08/10/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(76,4,8,to_date('01/05/2015','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(77,3,6,to_date('25/03/2018','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(78,4,3,to_date('16/11/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(79,4,17,to_date('02/10/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(80,5,2,to_date('09/06/2019','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(81,2,23,to_date('14/08/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(82,5,10,to_date('07/09/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(83,5,1,to_date('02/07/2017','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(84,3,11,to_date('09/03/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(85,5,18,to_date('02/07/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(86,3,15,to_date('10/05/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(87,3,12,to_date('10/05/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(88,2,19,to_date('25/03/2016','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(89,2,11,to_date('09/03/2014','dd/mm/yyyy'));
INSERT INTO ANIMAL VALUES(90,5,24,to_date('11/10/2016','dd/mm/yyyy'));
```

-- TABEL SE_HRANESTE CU

```
INSERT INTO SE_HRANESTE CU VALUES(1,9,50);
INSERT INTO SE_HRANESTE CU VALUES(2,1,100);
INSERT INTO SE_HRANESTE CU VALUES(2,6,150);
INSERT INTO SE_HRANESTE CU VALUES(2,3,100);
INSERT INTO SE_HRANESTE CU VALUES(3,7,300);
INSERT INTO SE_HRANESTE CU VALUES(4,9,50);
INSERT INTO SE_HRANESTE CU VALUES(4,8,20);
INSERT INTO SE_HRANESTE CU VALUES(5,8,5);
INSERT INTO SE_HRANESTE CU VALUES(6,8,3);
INSERT INTO SE_HRANESTE CU VALUES(7,2,500);
INSERT INTO SE_HRANESTE CU VALUES(8,4,100);
INSERT INTO SE_HRANESTE CU VALUES(8,6,150);
INSERT INTO SE_HRANESTE CU VALUES(8,5,100);
INSERT INTO SE_HRANESTE CU VALUES(9,3,400);
INSERT INTO SE_HRANESTE CU VALUES(10,4,100);
INSERT INTO SE_HRANESTE CU VALUES(12,5,50);
INSERT INTO SE_HRANESTE CU VALUES(13,6,400);
INSERT INTO SE_HRANESTE CU VALUES(14,6,100);
INSERT INTO SE_HRANESTE CU VALUES(15,3,500);
INSERT INTO SE_HRANESTE CU VALUES(16,1,100);
```

```

INSERT INTO SE_HRANESTE_CU VALUES(16,8,110);
INSERT INTO SE_HRANESTE_CU VALUES(17,7,20);
INSERT INTO SE_HRANESTE_CU VALUES(18,7,40);
INSERT INTO SE_HRANESTE_CU VALUES(19,1,30);
INSERT INTO SE_HRANESTE_CU VALUES(20,7,30);
INSERT INTO SE_HRANESTE_CU VALUES(21,2,100);
INSERT INTO SE_HRANESTE_CU VALUES(22,4,80);
INSERT INTO SE_HRANESTE_CU VALUES(22,1,30);
INSERT INTO SE_HRANESTE_CU VALUES(23,4,30);
INSERT INTO SE_HRANESTE_CU VALUES(24,7,30);

```

The screenshot shows the Oracle SQL Developer interface with a script being run. The script contains SQL code for creating a table, dropping it, and inserting data into two tables: ZOO and BILET. The execution output shows five rows inserted into the BILET table.

```

-- drop table functie;
-----
-- EXERCITIU 5
-- INSERARE VALORI
-- TABEL ZOO
INSERT INTO ZOO VALUES(1,'Gradina Zoologica din Bucuresti','Bucuresti','Strada Vadul Moldovei 4');
INSERT INTO ZOO VALUES(2,'Gradina Zoologica din Timisoara','Timisoara','Strada Avram Imbroane 90');
INSERT INTO ZOO VALUES(3,'Gradina Zoologica din Oradea','Oradea','Calea Matei Basarab 1');
INSERT INTO ZOO VALUES(4,'Gradina Zoologica din Resita','Resita','Strada Alexandru Nedelea 7');
INSERT INTO ZOO VALUES(5,'Gradina Zoologica din Targu Mures','Targu Mures','Strada Verii 57');

-- TABEL BILET
INSERT INTO BILET VALUES(1,5,to_date('08/04/2020','dd/mm/yyyy'),12,'Persoane cu dizabilitati');
INSERT INTO BILET VALUES(2,2,to_date('11/04/2020','dd/mm/yyyy'),20,'Adulti');
INSERT INTO BILET VALUES(3,3,to_date('17/03/2020','dd/mm/yyyy'),14,'Studenti');
INSERT INTO BILET VALUES(4,4,to_date('07/12/2020','dd/mm/yyyy'),15,'Grupe minim 15 adulti');

```

Script Output:

```

1 row inserted.

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Project\Atasie_Oana_SQL.sql

```

File Edit View Navigate Run Source Team Tools Window Help
Co... Pr... Oracle Connections
+ exam57
+ gr241
+ grupa41_01g
+ grupa44_01g
+ Oana
+ project
+ project_principal
+ recap_01g
+ TEST
Oracle NoSQL Connect Database Schema Ser...
SQL Worksheet History
Worksheet Query Builder
Atasie_Oana_SQL.sql
INSERT INTO BILET VALUES (36,4,to_date('01/06/2020','dd/mm/yyyy'),0,'Copii intre 3 si 18 ani');
INSERT INTO BILET VALUES (37,4,to_date('01/06/2020','dd/mm/yyyy'),0,'Copii intre 3 si 18 ani');
INSERT INTO BILET VALUES (38,4,to_date('08/02/2020','dd/mm/yyyy'),6,'Copii intre 3 si 18 ani');
INSERT INTO BILET VALUES (39,1,to_date('15/06/2020','dd/mm/yyyy'),0,'Copii pana la 3 ani');
INSERT INTO BILET VALUES (40,1,to_date('08/05/2020','dd/mm/yyyy'),16,'Grup minim 15 adulti');
INSERT INTO BILET VALUES (41,1,to_date('08/05/2020','dd/mm/yyyy'),16,'Grup minim 15 adulti');
INSERT INTO BILET VALUES (42,2,to_date('20/01/2020','dd/mm/yyyy'),20,'Adulti');
INSERT INTO BILET VALUES (43,4,to_date('08/02/2020','dd/mm/yyyy'),7,'Grup minim 15 copii si adulti');
INSERT INTO BILET VALUES (44,2,to_date('20/01/2020','dd/mm/yyyy'),20,'Adulti');
INSERT INTO BILET VALUES (45,5,to_date('08/04/2020','dd/mm/yyyy'),12,'Persoane cu dizabilitati');
INSERT INTO BILET VALUES (46,3,to_date('29/11/2020','dd/mm/yyyy'),5,'Copii pana la 3 ani');
INSERT INTO BILET VALUES (47,5,to_date('08/04/2020','dd/mm/yyyy'),12,'Persoane cu dizabilitati');
INSERT INTO BILET VALUES (48,3,to_date('29/11/2020','dd/mm/yyyy'),5,'Copii pana la 3 ani');
INSERT INTO BILET VALUES (49,3,to_date('28/11/2020','dd/mm/yyyy'),5,'Persoane peste 75 ani');
INSERT INTO BILET VALUES (50,3,to_date('28/11/2020','dd/mm/yyyy'),5,'Persoane peste 75 ani');
INSERT INTO BILET VALUES (51,3,to_date('28/11/2020','dd/mm/yyyy'),5,'Persoane peste 75 ani');
INSERT INTO BILET VALUES (52,1,to_date('03/05/2020','dd/mm/yyyy'),6,'Elevi');
INSERT INTO BILET VALUES (53,1,to_date('03/05/2020','dd/mm/yyyy'),16,'Grup minim 15 adulti');

Script Output X | Task completed in 0.417 seconds
1 row inserted.

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Project\Atasie_Oana_SQL.sql

```

File Edit View Navigate Run Source Team Tools Window Help
Co... Pr... Oracle Connections
+ exam57
+ gr241
+ grupa41_01g
+ grupa44_01g
+ Oana
+ project
+ project_principal
+ recap_01g
+ TEST
Oracle NoSQL Connect Database Schema Ser...
SQL Worksheet History
Worksheet Query Builder
Atasie_Oana_SQL.sql
INSERT INTO BILET VALUES (49,3,to_date('29/11/2020','dd/mm/yyyy'),5,'Copii pana la 3 ani');
INSERT INTO BILET VALUES (49,3,to_date('28/11/2020','dd/mm/yyyy'),5,'Persoane peste 75 ani');
INSERT INTO BILET VALUES (50,3,to_date('28/11/2020','dd/mm/yyyy'),5,'Persoane peste 75 ani');
INSERT INTO BILET VALUES (50,3,to_date('28/11/2020','dd/mm/yyyy'),5,'Persoane peste 75 ani');
INSERT INTO BILET VALUES (51,3,to_date('28/11/2020','dd/mm/yyyy'),5,'Persoane peste 75 ani');
INSERT INTO BILET VALUES (52,1,to_date('03/05/2020','dd/mm/yyyy'),6,'Elevi');
INSERT INTO BILET VALUES (53,1,to_date('03/05/2020','dd/mm/yyyy'),16,'Grup minim 15 adulti');

-- TABEL FUNCTIE
INSERT INTO FUNCTIE VALUES (1,'Vanzator');
INSERT INTO FUNCTIE VALUES (2,'Ingrujitor');
INSERT INTO FUNCTIE VALUES (3,'Securitate');
INSERT INTO FUNCTIE VALUES (4,'Director General');
INSERT INTO FUNCTIE VALUES (5,'Veterinar');

-- TABEL ANGAJAT
INSERT INTO ANGAJAT VALUES (1,5,3,'Dobre','Elena',2811025074180,4091,to_date('04/09/2018','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES (2,5,2,'Fratila','Mihai',1811015074130,4082,to_date('16/07/2020','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES (3,3,2,'Voinea','Teodor',1811015074138,4910,to_date('03/05/2015','dd/mm/yyyy'));

Script Output X | Task completed in 0.116 seconds
1 row inserted.

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

File Edit View Navigate Run Source Team Tools Window Help
+ Project SQL Worksheet History
Worksheet Query Builder
Oracle Connections exam57 g241 grupa11_011g grupa44_011g Oana project project_principal recap_011g TEST Oracle NoSQL Connector Database Schema Service
Atasie_Oana_SQL.sql
Script Output X Query Result X Task completed in 0.447 seconds

INSERT INTO ANGAJAT VALUES(60,3,1,'Tabacu','Ionut',1811015074171,3739,to_date('17/08/2017','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(61,3,5,'Moisescu','Catinca',2811025074169,4248,to_date('08/03/2019','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(62,3,3,'Tudor','Doina',2811025074125,3871,to_date('28/02/2020','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(63,3,4,'Tudor','Cristina',2811025074189,9966,to_date('19/05/2018','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(64,3,4,'Mihailescu','Dumitru',1811015074166,4394,to_date('01/01/2015','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(65,4,4,'Marin','Cristian',1811015074154,5288,to_date('03/02/2016','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(66,4,4,'Nistor','Cristina',2811025074128,5060,to_date('08/05/2020','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(67,4,1,'Stanciu','Elena',2811025074119,3653,to_date('25/08/2017','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(68,4,5,'Muresan','Valentin',1811015074188,6320,to_date('21/09/2016','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(69,4,1,'Popescu','Rodica',2811025074140,5842,to_date('28/07/2020','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(71,4,3,'Georgescu','Cosmin',1811015074133,4064,to_date('30/11/2016','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(72,5,3,'Dobre','Valentin',1811015074183,3921,to_date('06/02/2017','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(73,5,4,'Ionescu','Ionela',2811025074112,5223,to_date('28/12/2018','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(74,5,1,'Dima','Viorel',1811015074121,5030,to_date('05/06/2015','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(75,5,5,'Toma','Elena',2811025074143,5621,to_date('23/04/2020','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(76,5,5,'Mocanu','Tudor',1811015074137,3109,to_date('03/04/2019','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(77,5,3,'Puscasu','Ion',1811015074145,4457,to_date('27/03/2020','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(78,1,5,'Ionescu','Valentin',2913025074141,3050,to_date('07/11/2014','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(79,5,2,'Vladimirescu','Ionela',2811017079121,4000,to_date('05/06/2015','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(80,4,1,'Prisacariu','Cristian',1822015074188,7132,to_date('03/02/2016','dd/mm/yyyy'));
INSERT INTO ANGAJAT VALUES(81,4,2,'Hitaru','Erika',2922015074165,5721,to_date('03/02/2016','dd/mm/yyyy'));

1 row inserted.

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

File Edit View Navigate Run Source Team Tools Window Help
+ Project SQL Worksheet History
Worksheet Query Builder
Oracle Connections exam57 g241 grupa11_011g grupa44_011g Oana project project_principal recap_011g TEST Oracle NoSQL Connector Database Schema Service
Atasie_Oana_SQL.sql
Script Output X Query Result X Task completed in 0.204 seconds

-- TABEL RESPONSABILITATE
INSERT INTO RESPONSABILITATE VALUES(1,1,'relatia cu clientii');
INSERT INTO RESPONSABILITATE VALUES(2,1,'manipulare biletelor');
INSERT INTO RESPONSABILITATE VALUES(3,1,'prezentarea ofertei');
INSERT INTO RESPONSABILITATE VALUES(4,2,'curatarea tarcurilor');
INSERT INTO RESPONSABILITATE VALUES(5,2,'inspectarea tarcurilor');
INSERT INTO RESPONSABILITATE VALUES(6,2,'hranirea animalelor');
INSERT INTO RESPONSABILITATE VALUES(7,3,'supravegherea camerelor de luat vederi');
INSERT INTO RESPONSABILITATE VALUES(8,3,'supravegherea zonelor de acces in gradina zoologica');
INSERT INTO RESPONSABILITATE VALUES(9,3,'interventia in caz de nerespectare a regulilor');
INSERT INTO RESPONSABILITATE VALUES(10,4,'gestionarea tuturor situatiilor neprevazute');
INSERT INTO RESPONSABILITATE VALUES(11,4,'gestionarea bugetului');
INSERT INTO RESPONSABILITATE VALUES(12,4,'time management');
INSERT INTO RESPONSABILITATE VALUES(13,5,'controlul saptamanal al animalelor');
INSERT INTO RESPONSABILITATE VALUES(14,5,'tratarea animalelor in caz de boala');

1 row inserted.

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

File Edit View Navigate Run Source Team Tools Window Help

Oracle Connections

Atasie_Oana_SQL.sql

SQL Worksheet History

Worksheet Query Builder

```
-- TABEL MANCARE
INSERT INTO MANCARE VALUES(1,'radacini, fructe de padure, oua, larve');
INSERT INTO MANCARE VALUES(2,'carne');
INSERT INTO MANCARE VALUES(3,'furaje');
INSERT INTO MANCARE VALUES(4,'fructe, frunze');
INSERT INTO MANCARE VALUES(5,'ierburi,frunze verzi');
INSERT INTO MANCARE VALUES(6,'carne, fructe de padure');
INSERT INTO MANCARE VALUES(7,'carne, furaje');
INSERT INTO MANCARE VALUES(8,'granule');
INSERT INTO MANCARE VALUES(9,'cereale');
```

Script Output X | Task completed in 0.145 seconds

1 row inserted.

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

File Edit View Navigate Run Source Team Tools Window Help

Oracle Connections

Atasie_Oana_SQL.sql

SQL Worksheet History

Worksheet Query Builder

```
INSERT INTO MANCARE VALUES(9,'cereale');

-- TABEL FURNIZOR
INSERT INTO FURNIZOR VALUES(1,'Food for Animals');
INSERT INTO FURNIZOR VALUES(2,'Animal Food');
INSERT INTO FURNIZOR VALUES(3,'Ferma Ionescu');
INSERT INTO FURNIZOR VALUES(4,'TetraMin');
INSERT INTO FURNIZOR VALUES(5,'Global Food');
INSERT INTO FURNIZOR VALUES(6,'Animal Love');
INSERT INTO FURNIZOR VALUES(7,'American Zoo');
INSERT INTO FURNIZOR VALUES(8,'Ferma Corog');
INSERT INTO FURNIZOR VALUES(9,'TetraMax');
INSERT INTO FURNIZOR VALUES(10,'Ferma Alpin');
INSERT INTO FURNIZOR VALUES(11,'Global Food');
INSERT INTO FURNIZOR VALUES(12,'Food Food');
```

Script Output X | Task completed in 0.142 seconds

1 row inserted.

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Project\Atasie_Oana_SQL.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections Atasie_Oana_SQL.sql SQL Worksheet History

Worksheet Query Builder

```
-- TABEL ESTE_ASIGURATA_DE
INSERT INTO ESTE_ASIGURATA_DE VALUES (1, 1, 22500, 6, 11);
INSERT INTO ESTE_ASIGURATA_DE VALUES (2, 2, 67500, 14, 10);
INSERT INTO ESTE_ASIGURATA_DE VALUES (3, 3, 22500, 12, 13);
INSERT INTO ESTE_ASIGURATA_DE VALUES (4, 4, 13500, 11, 9);
INSERT INTO ESTE_ASIGURATA_DE VALUES (5, 5, 11250, 6, 13);
INSERT INTO ESTE_ASIGURATA_DE VALUES (6, 6, 45000, 6, 14);
INSERT INTO ESTE_ASIGURATA_DE VALUES (7, 7, 22500, 8, 11);
INSERT INTO ESTE_ASIGURATA_DE VALUES (8, 8, 67500, 10, 8);
INSERT INTO ESTE_ASIGURATA_DE VALUES (9, 9, 1350, 13, 17);
INSERT INTO ESTE_ASIGURATA_DE VALUES (7, 10, 450, 8, 13);
INSERT INTO ESTE_ASIGURATA_DE VALUES (1, 6, 9000, 6, 20);
INSERT INTO ESTE_ASIGURATA_DE VALUES (2, 7, 7500, 14, 1);
INSERT INTO ESTE_ASIGURATA_DE VALUES (3, 8, 30000, 8, 16);
INSERT INTO ESTE_ASIGURATA_DE VALUES (4, 9, 15000, 15, 19);
INSERT INTO ESTE_ASIGURATA_DE VALUES (5, 10, 45000, 10, 10);
INSERT INTO ESTE_ASIGURATA_DE VALUES (4, 7, 0, 7, 3);
```

Script Output Task completed in 0.142 seconds

1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Project\Atasie_Oana_SQL.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections Atasie_Oana_SQL.sql SQL Worksheet History

Worksheet Query Builder

```
-- TARC
INSERT INTO TARC VALUES (1, 500, 250, 300);
INSERT INTO TARC VALUES (2, 600, 600, 400);
INSERT INTO TARC VALUES (3, 500, 450, 400);
INSERT INTO TARC VALUES (4, 200, 300, 500);
INSERT INTO TARC VALUES (5, 300, 300, 300);
INSERT INTO TARC VALUES (6, 450, 300, 300);
INSERT INTO TARC VALUES (7, 400, 400, 300);
INSERT INTO TARC VALUES (8, 500, 180, 250);
INSERT INTO TARC VALUES (9, 150, 150, 200);
INSERT INTO TARC VALUES (10, 300, 400, 400);
INSERT INTO TARC VALUES (11, 400, 150, 150);
INSERT INTO TARC VALUES (12, 300, 300, 100);
INSERT INTO TARC VALUES (13, 250, 240, 150);
INSERT INTO TARC VALUES (14, 200, 250, 90);
```

Script Output Task completed in 0.158 seconds

1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Project\Atasie_Oana_SQL.sql

```
-- TABEL CATEGORIE
INSERT INTO CATEGORIE VALUES(1, 'Animal omnivor salbatic');
INSERT INTO CATEGORIE VALUES(2, 'Animale carnivore periculoase');
INSERT INTO CATEGORIE VALUES(3, 'Cabaline');
INSERT INTO CATEGORIE VALUES(4, 'Primate');
INSERT INTO CATEGORIE VALUES(5, 'Animale erbivore Africa');
INSERT INTO CATEGORIE VALUES(6, 'Ecvidee africane');
INSERT INTO CATEGORIE VALUES(7, 'Reptile');
INSERT INTO CATEGORIE VALUES(8, 'Pesti');
INSERT INTO CATEGORIE VALUES(9, 'Pasari');
INSERT INTO CATEGORIE VALUES(10,'Insectivore');

-- TABEL SPECIE
INSERT INTO SPECIE VALUES(1,10,9, 'Papagal cenusiu');
```

Script Output: x | Task completed in 0.139 seconds

```
1 row inserted.

1 row inserted.
```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Project\Atasie_Oana_SQL.sql

```
INSERT INTO SPECIE VALUES(1,1,1, 'Foc mastic');
INSERT INTO SPECIE VALUES(3,8,7,'Crocodil');
INSERT INTO SPECIE VALUES(4,10,9,'Perus');
INSERT INTO SPECIE VALUES(5,9,8,'Sanitarul');
INSERT INTO SPECIE VALUES(6,9,8,'Killi');
INSERT INTO SPECIE VALUES(7,2,2,'Leu');
INSERT INTO SPECIE VALUES(8,13,7, 'Testoasa terestra');
INSERT INTO SPECIE VALUES(9,3,3,'Ponei');
INSERT INTO SPECIE VALUES(10,4,4,'Cimpanzeu');
INSERT INTO SPECIE VALUES(11,9,8,'Discus');
INSERT INTO SPECIE VALUES(12,3,5,'Camila');
INSERT INTO SPECIE VALUES(13,7,6,'Zebra');
INSERT INTO SPECIE VALUES(14,3,5,'Dromader');
INSERT INTO SPECIE VALUES(15,3,3,'Armasar');
INSERT INTO SPECIE VALUES(16,6,1,'Urs brun');
INSERT INTO SPECIE VALUES(17,12,7,'Aigialosauridae');
INSERT INTO SPECIE VALUES(18,11,7,'Piton');
INSERT INTO SPECIE VALUES(19,9,7,'Cameleon');
INSERT INTO SPECIE VALUES(20,11,7,'Viperide');
INSERT INTO SPECIE VALUES(21,2,2,'Tigru');
INSERT INTO SPECIE VALUES(22,5,4,'Gorila');
INSERT INTO SPECIE VALUES(23,4,4,'Capucin');
INSERT INTO SPECIE VALUES(24,14,7,'Iguana');
INSERT INTO SPECIE VALUES(25,null,10,'Arici');
```

Script Output: x | Task completed in 0.137 seconds

```
1 row inserted.

1 row inserted.

1 row inserted.
```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Project\Atasie_Oana_SQL.sql

```
File Edit View Navigate Run Source Team Tools Window Help
+ Oracle Connections
  + exam57
  + gr241
  + grupa41_olig
  + grupa44_olig
  + Oana
  + project
  + project_principal
  + recap_olig
  + TEST
  + Oracle NoSQL Connect
  + Database Schema Services
Atasie_Oana_SQL.sql
Worksheet Query Builder
Script Output x
Task completed in 0.577 seconds
1 row inserted.

1 row inserted.
```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Project\Atasie_Oana_SQL.sql

```
File Edit View Navigate Run Source Team Tools Window Help
+ Oracle Connections
  + exam57
  + gr241
  + grupa41_olig
  + grupa44_olig
  + Oana
  + project
  + project_principal
  + recap_olig
  + TEST
  + Oracle NoSQL Connect
  + Database Schema Services
Atasie_Oana_SQL.sql
Worksheet Query Builder
Script Output x
Task completed in 0.183 seconds
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.
```

6. Definiți un subprogram stocat care să utilizeze un tip de colecție studiat. Apelați subprogramul.

Patronul dorește să crească cu 10% salariul tuturor angajaților cu cea mai mare vechime de la fiecare zoo, indiferent de numărul de angajați care îndeplinește această condiție. De asemenea, se dorește afișarea acestor angajați, alături de id-ul grădinii zoologice la care lucrează, cât și a salariului vechi și a salariului nou.

```
CREATE OR REPLACE PROCEDURE exercitiu6 IS
```

```

TYPE record_exercitiu6 IS RECORD(
    idul_zoo zoo.id_zoo%TYPE,
    idul_angajat angajat.id_angajat%TYPE
);
TYPE tablou_indexat_ex6 IS TABLE OF record_exercitiu6 INDEX BY
PLS_INTEGER;
colectie_ex6 tablou_indexat_ex6;
salariuV_ang_curent angajat.salariu%TYPE;
salariuN_ang_curent angajat.salariu%TYPE;

BEGIN
    SELECT id_zoo, id_angajat
    BULK COLLECT INTO colectie_ex6
    FROM angajat a
    WHERE data_angajarii = (SELECT MIN(data_angajarii)
        FROM angajat
        WHERE id_zoo = a.id_zoo
    )
    GROUP BY id_zoo, id_angajat
    ORDER BY 1;

    FOR i IN colectie_ex6.FIRST .. colectie_ex6.LAST LOOP
        IF colectie_ex6.EXISTS(colectie_ex6.PRIOR(i)) AND
            colectie_ex6(i).idul_zoo = colectie_ex6(colectie_ex6.PRIOR(i)).idul_zoo
        THEN
            DBMS_OUTPUT.PUT_LINE('Angajatul ' || colectie_ex6(i).idul_angajat || '
este unul dintre angajatii cu'
            || ' cea mai mare vechime de la zoo-ul cu id-ul ' ||
            colectie_ex6(i).idul_zoo || '.');
        END IF;
    END LOOP;
END;

```

```

ELSIF colectie_ex6.EXISTS(colectie_ex6.NEXT(i)) AND
    colectie_ex6(i).idul_zoo = colectie_ex6(colectie_ex6.NEXT(i)).idul_zoo
THEN
    DBMS_OUTPUT.PUT_LINE('Angajatul ' || colectie_ex6(i).idul_angajat || '
este unul dintre angajatii cu'
    || ' cea mai mare vechime de la zoo-ul cu id-ul ' ||
colectie_ex6(i).idul_zoo || '.');
ELSE
    DBMS_OUTPUT.PUT_LINE('Angajatul ' || colectie_ex6(i).idul_angajat || ' are
cea mai mare vechime de la'
    || ' zoo-ul cu id-ul ' || colectie_ex6(i).idul_zoo || '.');
END IF;

SELECT salariu
INTO salariuV_ang_curent
FROM angajat
WHERE id_angajat = colectie_ex6(i).idul_angajat;

UPDATE angajat
SET salariu = salariu + ROUND(salariu * 0.1)
WHERE id_angajat = colectie_ex6(i).idul_angajat
RETURNING salariu INTO salariuN_ang_curent;

DBMS_OUTPUT.PUT_LINE('Angajatul avea salariul ' || salariuV_ang_curent || ',
iar acum are salariul '
    || salariuN_ang_curent || '.');
DBMS_OUTPUT.NEW_LINE;

END LOOP;

END exercitiu6;
/

```

Apelarea subprogramului cu bloc PL/SQL:

```

BEGIN
    exercitiu6();
END;
/

```

```

CREATE OR REPLACE PROCEDURE exercitiu6 IS
    id_angajat number;
    salariuN_ang_curent number;
    salariuV_ang_curent number;
    CURSOR colectie_ex6 IS
        SELECT id_angajat, salariu
        FROM angajati
        WHERE id_angajat = collectie_ex6(i).idul_angajat
        RETURNING salariu INTO salariuN_ang_curent;
    DBMS_OUTPUT.PUT_LINE('Angajatul avea salariul ' || salariuV_ang_curent ||
                         ' || salariuN_ang_curent || '.' );
    DBMS_OUTPUT.NEW_LINE;
END LOOP;
END exercitiu6;
/
-- Apelarea subprogramului cu bloc PL/SQL
BEGIN
    exercitiu6();
END;
/

```

-- EXERCITIU 7
-- 7. Definiți un subprogram stocat care să utilizeze un tip de cursor studiat. /
-- Veterinarii de la un zoo pot examina maxim trei animale pe zi, de aceea, se definește o cursor studiat.
-- din noul an, 1 februarie 2021, prioritatea să fie oferită animalelor cu o vîrstă mai mare de șase ani. Deoarece veterinarii nu pot examina mai mult de 3 animale pe zi, se dorește obținerea pentru fiecare zoo, a top trei cele mai în vîrstă animale cu vîrstă peste 6 ani. Se dorește afișarea acestor informații astfel: numele și orașul zoo-ului, urmat de detaliile animalelor care fac parte din topul menționat mai sus (id-ul animalului, data nașterii și denumirea speciei).

```

Procedure EXERCITIU6 compiled
PL/SQL procedure successfully completed.

```

7. Definiți un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul.

Veterinarii de la un zoo pot examina maxim trei animale pe zi, de aceea, se dorește ca, în prima zi de examinare din noul an, 1 februarie 2021, prioritatea să fie oferită animalelor cu o vîrstă mai mare de șase ani. Deoarece veterinarii nu pot examina mai mult de 3 animale pe zi, se dorește obținerea pentru fiecare zoo, a top trei cele mai în vîrstă animale cu vîrstă peste 6 ani. Se dorește afișarea acestor informații astfel: numele și orașul zoo-ului, urmat de detaliile animalelor care fac parte din topul menționat mai sus (id-ul animalului, data nașterii și denumirea speciei).

Dacă există animale cu aceeași vîrstă, ele vor fi plasate pe același loc în top, și fiind un caz special, veterinarii vor verifica toate animalele din top. Dacă există un zoo care nu conține animale cu vîrstă mai mare de 6 ani se va afișa un mesaj sugestiv. Dacă la un zoo nu există un număr de animale care îndeplinește condiția de mai sus, se vor afișa animalele din top, dar și un mesaj care să sugereze că nu se poate forma un top trei, ci unul mai mic.

```

CREATE OR REPLACE PROCEDURE exercitiu7 IS
    idul_zoo zoo.id_zoo%TYPE;
    nume_zoo zoo.denumire%TYPE;
    oras_zoo zoo.oras%TYPE;
    idul_anim animal.id_animal%TYPE;
    anim_data_nasterii animal.data_nasterii%TYPE;
    denumire_specie specie.denumire%TYPE;

```

```

numar_top number :=0;
data_veche animal.data_nasterii%TYPE;

CURSOR cursor_zoo IS
    SELECT id_zoo, denumire, oras
    FROM zoo;

CURSOR cursor_animal(parametru_zoo_id zoo.id_zoo%TYPE) IS
    SELECT a.id_animal, data_nasterii, s.denumire
    FROM animal a JOIN specie s ON (a.id_specie = s.id_specie)
    WHERE id_zoo = parametru_zoo_id AND MONTHS_BETWEEN('01-FEB-
2021',data_nasterii) >= 6*12
    ORDER BY data_nasterii;

BEGIN
    OPEN cursor_zoo;
    LOOP
        FETCH cursor_zoo INTO idul_zoo, nume_zoo, oras_zoo;
        EXIT WHEN cursor_zoo%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Gradina zoologica "' || nume_zoo || " din orasul ' ||
oras_zoo
                                || ' are urmatorul top trei: ');
        DBMS_OUTPUT.PUT_LINE('-----');
        numar_top :=0;
        SELECT SYSDATE
        INTO data_veche
        FROM dual;
        OPEN cursor_animal(idul_zoo);
        LOOP
            FETCH cursor_animal INTO idul_anim, anim_data_nasterii, denumire_specie;
            IF anim_data_nasterii <> data_veche THEN
                numar_top := numar_top + 1;
            END IF;
            data_veche := anim_data_nasterii;
            EXIT WHEN numar_top > 3 OR cursor_animal%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE(numar_top || '. Animalul cu id-ul ' || idul_anim || ' si
denumirea speciei ' || denumire_specie
                                || ' are data de nastere ' || anim_data_nasterii || '.');
        END LOOP;
        IF numar_top = 0 THEN

```

```

DBMS_OUTPUT.PUT_LINE( 'Nu exista animale cu varsta mai mare de 6 ani la
zoo-ul "' || nume_zoo || '".');
ELSIF numar_top < 3 THEN
    DBMS_OUTPUT.PUT_LINE( 'Nu se poate ajunge la top 3 la zoo-ul "' || 
nume_zoo || '".');
    END IF;
    CLOSE cursor_animal;
    DBMS_OUTPUT.NEW_LINE;
END LOOP;
CLOSE cursor_zoo;
END exercitiu7;
/

```

Apelarea subprogramului cu bloc PL/SQL:

```

BEGIN
exercitiu7();
END;
/

```

```

-- Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Project\Atasie_Oana_SQL.sql
File Edit View Navigate Run Source Team Tools Window Help
[Oracle Connections]
[project]
[Oracle SQL*Plus Connector]
[Database Schema Services]
[Worksheet] [Query Builder]
[Script Output] [Task completed in 0.065 seconds]
[Output] [Buffer Size: 20000]
[Output]
Gradina zoologica "Gradina Zoologica din Bucuresti" din orasul Bucuresti are urmatorul top trei:
-----
Nu exista animale cu varsta mai mare de 6 ani la zoo-ul "Gradina Zoologica din Bucuresti".
Gradina zoologica "Gradina Zoologica din Timisoara" din orasul Timisoara are urmatorul top trei:
-----
1. Animalul cu id-ul 89 si denumirea speciei Discus are data de nastere 09-MAR-14.
2. Animalul cu id-ul 55 si denumirea speciei Gorila are data de nastere 11-MAR-14.
3. Animalul cu id-ul 56 si denumirea speciei Perus are data de nastere 11-MAR-14.
Gradina zoologica "Gradina Zoologica din Oradea" din orasul Oradea are urmatorul top trei:
-----
1. Animalul cu id-ul 57 si denumirea speciei Camila are data de nastere 29-DEC-13.
2. Animalul cu id-ul 23 si denumirea speciei Zebra are data de nastere 23-MAR-14.
3. Animalul cu id-ul 87 si denumirea speciei Ponoi are data de nastere 10-MAY-14.
3. Animalul cu id-ul 86 si denumirea speciei Armasar are data de nastere 10-MAY-14.
Gradina zoologica "Gradina Zoologica din Resita" din orasul Resita are urmatorul top trei:
-----
1. Animalul cu id-ul 64 si denumirea speciei Crocodil are data de nastere 30-MAY-14.
2. Animalul cu id-ul 75 si denumirea speciei Aligatosauridae are data de nastere 02-OCT-14.
3. Animalul cu id-ul 78 si denumirea speciei Crocodil are data de nastere 16-NOV-14.
Gradina zoologica "Gradina Zoologica din Targu Mures" din orasul Targu Mures are urmatorul top trei:
-----
1. Animalul cu id-ul 40 si denumirea speciei Crocodil are data de nastere 05-JUN-14.
2. Animalul cu id-ul 8 si denumirea speciei Leu are data de nastere 16-JUN-14.
Nu se poate ajunge la top 3 la zool-ul "Gradina Zoologica din Targu Mures".

```

8. Definiți un subprogram stocat de tip funcție care să utilizeze 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

*Se dorește crearea unui nou zoo pe o suprafață cu o dimensiune prestabilită, iar pentru a putea vedea ce categorii de animale sunt mai potrivite pentru spațiul acestuia, se dorește cunoașterea tuturor dimensiunilor țarcurilor distincte, necesare speciilor dintr-o anumită categorie, astfel: pentru o denumire de categorie primită ca parametru se dorește afișarea unei colecții care conține toate dimensiunile (lungime*latime* înălțime) țarcurilor distincte (adică cele cu id diferit) în care locuiesc speciile de animale din categoria respectivă (sub forma "dimensiune1", "dimensiune2", ... "dimensiuneN") ordonate crescător.*

Dacă se întâlnesc pe parcurs dimensiuni identice ale unor țarcuri diferite, dimensiunea identică nu va reprezenta un nou element în colecție, ci elementul care se repetă va avea forma "dimensiuneK X nr_de_aparitii".

```
CREATE OR REPLACE TYPE exercitiu8_colectie IS TABLE OF VARCHAR2(50);
/
```

```
CREATE OR REPLACE FUNCTION exercitiu8
(denum_categ IN categorie.denumire%TYPE)
RETURN exercitiu8_colectie IS
    colectie exercitiu8_colectie;
    i INTEGER;
    j INTEGER;
    aux VARCHAR2(50);
    exceptie_categ_gresita EXCEPTION;
    verificare VARCHAR2(10);
```

```
BEGIN
```

```
    SELECT 'Exista'
    INTO verificare
    FROM categorie
    WHERE LOWER(denumire) = LOWER(denum_categ); -- pentru a verifica daca
exista categoria data
```

```
    SELECT latime*inaltime*lungime
    BULK COLLECT INTO colectie
    FROM categorie c JOIN specie s ON (c.id_categorie = s.id_categorie)
        JOIN tarc t ON(s.id_tarc = t.id_tarc)
    WHERE LOWER(c.denumire) = LOWER(denum_categ)
    GROUP BY latime*inaltime*lungime, t.id_tarc
    ORDER BY 1;
```

```

IF colectie.COUNT = 0 -- pentru a verifica daca se poate ajunge de la categorii la tarc
    THEN RAISE exceptie_categ_gresita;
END IF;

i := colectie.FIRST;
WHILE i <= colectie.LAST LOOP
    IF i > colectie.FIRST THEN
        j := colectie.PRIOR(i);
        aux := colectie(j);
        IF colectie(i) = aux OR (INSTR(aux,'X') <> 0 AND SUBSTR(aux, 0,
LENGTH(aux)-2) = colectie(i)) THEN
            IF INSTR(aux,'X') <> 0 THEN
                aux      :=      CONCAT(SUBSTR(aux,          0,      LENGTH(aux)-1),
TO_CHAR(TO_NUMBER(SUBSTR(aux, LENGTH(aux))+1));
            ELSE
                aux := CONCAT(aux,'X2');
            END IF;
            colectie(j) := aux;
            colectie.DELETE(i);
            i := colectie.NEXT(i);
        ELSE
            i := colectie.NEXT(i);
        END IF;
    ELSE
        i := colectie.NEXT(i);
    END IF;
END LOOP;

RETURN colectie;

EXCEPTION
    WHEN no_data_found
        THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista categoria data ca
parametru.');
    WHEN exceptie_categ_gresita
        THEN RAISE_APPLICATION_ERROR(-20001, 'Pentru categoria data nu se pot
parcurge tablele.');
    WHEN OTHERS
        THEN RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');

```

```
END exercitiu8;
```

```
/
```

Apelare:

```
SELECT exercitiu8('Animal omnivor salbatic') FROM DUAL; --
```

```
PROIECT.EXERCITIU8_COLECTIE('37500000', '40500000')
```

```
SELECT exercitiu8('Reptile') FROM DUAL; --
```

```
PROIECT.EXERCITIU8_COLECTIE('4500000X2', '9000000X3', '22500000')
```

SELECT exercitiu8('Mamifer') FROM DUAL; -- ORA-20000: Nu exista categoria data ca parametru.

SELECT exercitiu8('Insectivore') FROM DUAL; -- ORA-20001: Pentru categoria data nu se pot parcurge tablele.

The screenshot shows the Oracle SQL Developer interface with the following details:

- File Menu:** Edit, View, Navigate, Run, Source, Team, Tools, Window, Help.
- Toolbars:** Standard, Object Navigator, Database Navigator, Object Inspector, Database Inspector, Script Output.
- Connections:** Oracle Connections (exam57, gr241, grupa41_oilg, grupa44_oilg, Oana, proiect, proiect_principal, recap_oilg, TEST).
- Project:** Atasie_Oana_SQL.sql.
- Worksheet:** The main area contains the PL/SQL code for the EXERCITIU8_COLECTIE function. The code includes logic for deleting from a collection, moving to the next item, handling exceptions for data not found or invalid category, and raising application errors for invalid parameters or unable to iterate over tables.
- Script Output:** Shows the results of the compilation command: "Type EXERCITIU8_COLECTIE compiled" and "Function EXERCITIU8 compiled".

```
colectie.DELETE(i);
i := colectie.NEXT(i);
ELSE
    i := colectie.NEXT(i);
END IF;
ELSE
    i := colectie.NEXT(i);
END IF;
END LOOP;

RETURN colectie;

EXCEPTION
    WHEN no_data_found
        THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista categoria data ca parametru.');
    WHEN exceptie_categ_gresita
        THEN RAISE_APPLICATION_ERROR(-20001, 'Pentru categoria data nu se pot parcurge tablele.');
    WHEN OTHERS
        THEN RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');

END exercitiu8;
/
```

Type EXERCITIU8_COLECTIE compiled
Function EXERCITIU8 compiled

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

File Edit View Navigate Run Source Team Tools Window Help
+ Oracle Connections
  exam57
  gr241
  grupa41_o1lg
  grupa44_o1lg
  Oana
  project
  project_principal
  recap_o1lg
  TEST
  Oracle NoSQL Connection
  Database Schema Services
  Atasie_Oana_SQL.sql ->
  SQL Worksheet History
  Worksheet Query Builder
  RETURN colectie;

  EXCEPTION
    WHEN no_data_found
      THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista categoria data ca parametru.');
    WHEN exceptie_categ_gresita
      THEN RAISE_APPLICATION_ERROR(-20001, 'Pentru categoria data nu se pot parcurge tablele.' );
    WHEN OTHERS
      THEN RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');

  END exercitiu8;
/

-- APELARE:
SELECT exercitiu8('Animal omnivor salbatic') FROM DUAL; -- PROIECT.EXERCITIU8_COLECTIE('37500000', '40500000')
SELECT exercitiu8('Reptile') FROM DUAL; -- PROIECT.EXERCITIU8_COLECTIE('4500000X2', '9000000X3', '22500000')
SELECT exercitiu8('Mamifer') FROM DUAL; -- ORA-20000: Nu exista categoria data ca parametru.
SELECT exercitiu8('Insectivore') FROM DUAL; -- ORA-20001: Pentru categoria data nu se pot parcurge tablele.

-- EXERCITIU 9
-- 9. Definiti un subprogram stocat de tip procedura care sa utilizeze 5 dintre tablele definite. Tratati toate
-- exceptiile care pot aparea. Apelati subprogramul astfel incat sa evidențiatii toate cazurile tratate.

```

Script Output X | Query Result X | All Rows Fetched: 1 in 0.002 seconds

EXERCITIU8(ANIMALOMNIVORSALBATIC)
1 PROIECT.EXERCITIU8_COLECTIE('37500000', '40500000')

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

File Edit View Navigate Run Source Team Tools Window Help
+ Oracle Connections
  exam57
  gr241
  grupa41_o1lg
  grupa44_o1lg
  Oana
  project
  project_principal
  recap_o1lg
  TEST
  Oracle NoSQL Connection
  Database Schema Services
  Atasie_Oana_SQL.sql ->
  SQL Worksheet History
  Worksheet Query Builder
  RETURN colectie;

  EXCEPTION
    WHEN no_data_found
      THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista categoria data ca parametru.');
    WHEN exceptie_categ_gresita
      THEN RAISE_APPLICATION_ERROR(-20001, 'Pentru categoria data nu se pot parcurge tablele.' );
    WHEN OTHERS
      THEN RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');

  END exercitiu8;
/

-- APELARE:
SELECT exercitiu8('Animal omnivor salbatic') FROM DUAL; -- PROIECT.EXERCITIU8_COLECTIE('37500000', '40500000')
SELECT exercitiu8('Reptile') FROM DUAL; -- PROIECT.EXERCITIU8_COLECTIE('4500000X2', '9000000X3', '22500000')
SELECT exercitiu8('Mamifer') FROM DUAL; -- ORA-20000: Nu exista categoria data ca parametru.
SELECT exercitiu8('Insectivore') FROM DUAL; -- ORA-20001: Pentru categoria data nu se pot parcurge tablele.

-- EXERCITIU 9
-- 9. Definiti un subprogram stocat de tip procedura care sa utilizeze 5 dintre tablele definite. Tratati toate
-- exceptiile care pot aparea. Apelati subprogramul astfel incat sa evidențiatii toate cazurile tratate.

```

Script Output X | Query Result X | All Rows Fetched: 1 in 0.007 seconds

EXERCITIU8(REPTILE)
1 PROIECT.EXERCITIU8_COLECTIE('4500000X2', '9000000X3', '22500000')

```
END IF;
END LOOP;

RETURN colectie;

EXCEPTION
    WHEN no_data_found
        THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista categoria data ca parametru.');
    WHEN exceptie_categ_gresita
        THEN RAISE_APPLICATION_ERROR(-20001, 'Pentru categoria data nu se pot parcurge tablele.' );
    WHEN OTHERS
        THEN RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');

END exercitiu8;
/

-- APELARE:
SELECT exercitiu8('Animal omnivor salbatic') FROM DUAL; -- PROIECT.EXERCITIU8_COLECTIE('37500000', '40500000')
SELECT exercitiu8('Reptile') FROM DUAL; -- PROIECT.EXERCITIU8_COLECTIE('4500000X2', '9000000X3', '22500000')
SELECT exercitiu8('Mamifer') FROM DUAL; -- ORA-20000: Nu exista categoria data ca parametru.
SELECT exercitiu8('Insectivore') FROM DUAL; -- ORA-20001: Pentru categoria data nu se pot parcurge table.
```

Script Output X Query Result X

Executing:SELECT exercitiu8('Mamifer') FROM DUAL in 0 seconds

ORA-20000: Nu exista categoria data ca parametru.
ORA-06512: at 'PROIECT.EXERCITIU8', line 56
20000. 00000 - "%"
*Cause: The stored procedure 'raise_application_error'
was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
the application administrator or DBA for more information.

```
END IF;
END LOOP;

RETURN colectie;

EXCEPTION
    WHEN no_data_found
        THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista categoria data ca parametru.');
    WHEN exceptie_categ_gresita
        THEN RAISE_APPLICATION_ERROR(-20001, 'Pentru categoria data nu se pot parcurge tablele.' );
    WHEN OTHERS
        THEN RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');

END exercitiu8;
/

-- APELARE:
SELECT exercitiu8('Animal omnivor salbatic') FROM DUAL; -- PROIECT.EXERCITIU8_COLECTIE('37500000', '40500000')
SELECT exercitiu8('Reptile') FROM DUAL; -- PROIECT.EXERCITIU8_COLECTIE('4500000X2', '9000000X3', '22500000')
SELECT exercitiu8('Mamifer') FROM DUAL; -- ORA-20000: Nu exista categoria data ca parametru.
SELECT exercitiu8('Insectivore') FROM DUAL; -- ORA-20001: Pentru categoria data nu se pot parcurge table.
```

Script Output X Query Result X

Executing:SELECT exercitiu8('Insectivore') FROM DUAL in 0 seconds

ORA-20001: Pentru categoria data nu se pot parcurge table.
ORA-06512: at 'PROIECT.EXERCITIU8', line 58

9. Definiți un subprogram stocat de tip procedură care să utilizeze 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Peste câteva luni, lanțul de grădini zoologice trebuie să reînnoiască anumite contracte cu anumiți furnizori, de aceea, este realizat un raport despre furnizori și mâncarea pe care ei o asigură. Astfel, în cadrul acestui raport, pentru un furnizor (denumire) se dorește cunoașterea mâncărurilor pe care acesta le asigură (id), cât și speciile (denumire) care se hrănesc cu aceste tipuri de mâncare, alături de procentul pe care animalele din speciile respective îl mănâncă din cantitatea totală adusă de furnizor.

Se dorește afișarea astfel:

id_mancare1: (denumire_specieA, procentA), (denumire_specieB, procentB),

id_mancare2: (denumire_specieX, procentX), (denumire_specieY, procentY),

....

```
CREATE OR REPLACE TYPE exercitiu9_colectie IS TABLE OF VARCHAR2(100); --  
colectia finala  
/
```

```
CREATE OR REPLACE PROCEDURE exercitiu9  
( denum_furnizor IN furnizor.denumire%TYPE, colectie OUT exercitiu9_colectie)  
IS  
    CURSOR cursor_ex9(parametru furnizor.denumire%TYPE) IS  
        SELECT m.id_mancare, s.denumire,  
        ROUND(shc.cantitate_medie_zilnica*100/ead.cantitate,5)  
        FROM furnizor f JOIN este_asigurata_de ead ON (f.id_furnizor = ead.id_furnizor)  
        JOIN mancare m ON (ead.id_mancare = m.id_mancare)  
        JOIN se_hraneste_cu shc ON (m.id_mancare = shc.id_mancare)  
        JOIN specie s ON (shc.id_specie = s.id_specie)  
        WHERE LOWER(f.denumire) = LOWER(parametru)  
        GROUP BY m.id_mancare, s.denumire,  
        ROUND(shc.cantitate_medie_zilnica*100/ead.cantitate,5)  
        ORDER BY 1;  
  
    idul_mancare mancare.id_mancare%TYPE;  
    denum_specie specie.denumire%TYPE;  
    procent NUMBER(10,5);
```

```

ok NUMBER;
i NUMBER;
procent2 VARCHAR2(10);
verificare VARCHAR2(30);
arrive_false EXCEPTION;

BEGIN

    SELECT 'Exista'
    INTO verificare
    FROM furnizor
    WHERE LOWER(denumire) = LOWER(denum_furnizor); -- pentru erorile
no_data_found si too_many_rows

    SELECT min(s.denumire)
    INTO verificare
    FROM furnizor f JOIN este_asigurata_de ead ON (f.id_furnizor = ead.id_furnizor)
        JOIN mancare m ON (ead.id_mancare = m.id_mancare)
        JOIN se_hraneste_cu shc ON (m.id_mancare = shc.id_mancare)
        JOIN specie s ON (shc.id_specie = s.id_specie)
    WHERE LOWER(f.denumire) = LOWER(denum_furnizor);

    IF verificare IS NULL
        THEN RAISE arrive_false; -- pentru a verifica daca se poate ajunge de la furnizor la
specie
    END IF;

    OPEN cursor_ex9(denum_furnizor);
    colectie := exercitiu9_colectie();
    ok := 0;
    LOOP
        FETCH cursor_ex9 INTO idul_mancare, denum_specie, procent;
        EXIT WHEN cursor_ex9%NOTFOUND;

        IF procent < 1 THEN
            procent2 := '0' || TO_CHAR(procent);
        ELSE
            procent2 := TO_CHAR(procent);
        END IF;

```

```

IF ok = 0 THEN
    colectie.EXTEND;
    i := colectie.FIRST;
    colectie(i) := TO_CHAR(idul_mancare) || '(' || denum_specie || ',' || procent2 || ')';
    ok := 1;
ELSE
    IF SUBSTR(colectie(i), 0, 1) = idul_mancare THEN
        colectie(i) := colectie(i) || ',(' || denum_specie || ',' || procent2 || ')';
    ELSE
        colectie.EXTEND;
        i := colectie.LAST;
        colectie(i) := TO_CHAR(idul_mancare) || ':(' || denum_specie || ',' || procent2 || ')';
    END IF;
END IF;
END LOOP;
CLOSE cursor_ex9;
EXCEPTION
    WHEN zero_divide
        THEN RAISE_APPLICATION_ERROR(-20000, 'S-a incercat impartirea prin 0.');
    WHEN no_data_found
        THEN RAISE_APPLICATION_ERROR(-20001, 'Nu exista furnizorul dat ca parametru.');
    WHEN arrive_false
        THEN RAISE_APPLICATION_ERROR(-20002, 'Nu se pot parurge tabelele.');
    WHEN too_many_rows
        THEN RAISE_APPLICATION_ERROR(-20003, 'Exista mai multi furnizori cu denumirea data.');
    WHEN others
        THEN RAISE_APPLICATION_ERROR(-20004, 'Alta eroare.');
END exercitiu9;
/

```

Apelarea subprogramului cu bloc PL/SQL:

```

DECLARE
    rezultat exercitiu9_colectie;
BEGIN
    exercitiu9('FERMA COROG',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;

```

```

END;
/
-- 3:(Armasar, 1.66667), (Ponei, 1.33333), (Porc mistret, 0.33333)
-- 8:(Killi, 0.00444), (Perus, 0.02963), (Sanitari, 0.00741), (Urs brun, 0.16296)
-----
DECLARE
    rezultat exercitiu9_colectie;
BEGIN
    exercitiu9('TEST',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- ORA-20001: Nu exista furnizorul dat ca parametru.
-----
DECLARE
    rezultat exercitiu9_colectie;
BEGIN
    exercitiu9('Global Food',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- ORA-20003: Exista mai multi furnizori cu denumirea data.
-----
DECLARE
    rezultat exercitiu9_colectie;
BEGIN
    exercitiu9('Food Food',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- ORA-20002: Nu se pot parcurge tabelele.
-----
DECLARE
    rezultat exercitiu9_colectie;

```

```

BEGIN
    exercitiu9('American Zoo',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- ORA-20000: S-a incercat impartirea prin 0.

```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Title Bar:** Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql
- Toolbars:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help.
- Connections:** Oracle Connections (exam57, gr241, grupa1_01g, grupa4_01g, Oana, project, project_principal, recap_01g, TEST, Oracle NoSQL Connection, Database Schema Service).
- SQL Worksheet:** The main workspace contains the PL/SQL code for EXERCITIU9_COLECTIE. The code handles various error cases (zero_divide, no_data_found, arrive_false, too_many_rows, others) and performs calculations on the colectie table.
- Script Output:** Shows the message "Task completed in 0.144 seconds".
- Output:** Shows the results of the compilation: "Type EXERCITIU9_COLECTIE compiled" and "Procedure EXERCITIU9 compiled".

The screenshot shows the Oracle SQL Developer interface with the following details:

- Title Bar:** Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql
- Toolbars:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help.
- Connections:** Oracle Connections (exam57, gr241, grupa1_01g, grupa4_01g, Oana, project, project_principal, recap_01g, TEST, Oracle NoSQL Connection, Database Schema Service).
- SQL Worksheet:** The main workspace contains the PL/SQL code for EXERCITIU9. It includes a comment about the subprogram and defines a result set for the exercitiu9 procedure.
- Output:** Shows the results of the execution: "3: (Armasar, 1.66667), (Ponei, 1.33333), (Porc mistret, 0.33333)" and "8: (Kili, 0.00444), (Perus, 0.02963), (Sanitari, 0.00741), (Urs brun, 0.16296)".

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```
File Edit View Navigate Run Source Team Tools Window Help
Connections Atasie_Oana_SQL.sql SQL Worksheet History project
Worksheet Query Builder
resultat exercitiu9_colectie;
BEGIN
    exercitiu9('FERMA COROG',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- 3: (Armasar, 1.66667), (Ponei, 1.33333), (Porc mistret, 0.33333)
-- 8: (Killi, 0.0044), (Perus, 0.02963), (Sanitari, 0.00741), (Urs brun, 0.16296)
-----
DECLARE
    resultat exercitiu9_colectie;
BEGIN
    exercitiu9('TEST',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- ORA-20001: Nu exista furnizorul dat ca parametru.
-----
DECLARE
    resultat exercitiu9_colectie;
BEGIN
    exercitiu9('TEST',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
Error report -
ORA-20001: Nu exista furnizorul dat ca parametru.
ORA-06512: at "PROJECT.EXERCITIU9", line 75
ORA-06512: at line 4
```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```
File Edit View Navigate Run Source Team Tools Window Help
Connections Atasie_Oana_SQL.sql SQL Worksheet History project
Worksheet Query Builder
FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
    DBMS_OUTPUT.PUT_LINE(rezultat(i));
END LOOP;
END;
/
-- ORA-20001: Nu exista furnizorul dat ca parametru.
-----
DECLARE
    resultat exercitiu9_colectie;
BEGIN
    exercitiu9('Global Food',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- ORA-20003: Există mai multi furnizori cu denumirea data.
-----
DECLARE
    resultat exercitiu9_colectie;
BEGIN
    exercitiu9('Food Food',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
Error report -
ORA-20003: Există mai multi furnizori cu denumirea data.
ORA-06512: at "PROJECT.EXERCITIU9", line 79
ORA-06512: at line 4
```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

File Edit View Navigate Run Source Team Tools Window Help
Connections Atasie_Oana_SQL.sql SQL Worksheet History project
Worksheet Query Builder
BEGIN
    exercitiu9('Global Food',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- ORA-20003: Există mai multi furnizori cu denumirea data.

=DECLARE
    rezultat exercitiu9_colectie;
BEGIN
    exercitiu9('Food Food',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- ORA-20002: Nu se pot parcurge tabelele.

=DECLARE
    rezultat exercitiu9_colectie;
BEGIN
    exercitiu9('Food Food',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
Error report -
ORA-20002: Nu se pot parcurge tabelele.
ORA-06512: at "PROJECT.EXERCITIU9", line 77
ORA-06512: at line 4

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

File Edit View Navigate Run Source Team Tools Window Help
Connections Atasie_Oana_SQL.sql SQL Worksheet History project
Worksheet Query Builder
BEGIN
    rezultat exercitiu9_colectie;
    exercitiu9('Food Food',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- ORA-20002: Nu se pot parcurge tabelele.

=DECLARE
    rezultat exercitiu9_colectie;
BEGIN
    exercitiu9('American Zoo',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- ORA-20000: S-a incercat impartirea prin 0.

EXERCITIU 10
Script Output x | Task completed in 0.089 seconds
END;
Error report -
ORA-20000: S-a incercat impartirea prin 0.
ORA-06512: at "PROJECT.EXERCITIU9", line 73
ORA-06512: at line 4
20000. 00000 - "%s"
*Cause:   The stored procedure 'raise_application_error'
          was called which causes this error to be generated.
*Action:  Correct the problem as described in the error message or contact
          the application administrator or DBA for more information.

```

10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

Pentru a avea siguranța că datele angajaților sunt protejate, doar persoanele cu permisiune pot efectua operații asupra tabelului ANGAJAT din baza de date, iar aceste operații pot fi făcute doar de user-ul "proiect" în timpul programului de muncă a zoo-urilor și anume:

*Luni, Marți, Miercuri, Joi, Vineri: 09 -> 18
Sâmbăta, Duminica: Închis*

```
CREATE OR REPLACE TRIGGER exercitiu10
  BEFORE INSERT OR DELETE OR UPDATE on angajat
BEGIN
  IF USER <> UPPER('proiect')
    THEN
      RAISE_APPLICATION_ERROR(-20000,'Nu aveti dreptul sa executati operatii
asupra tabelului!');

      ELSIF (LOWER(TO_CHAR(SYSDATE,'DY')) = LOWER('SAT')) OR
(LOWER(TO_CHAR(SYSDATE,'DY')) = LOWER('SUN'))
      THEN
        RAISE_APPLICATION_ERROR(-20001,' Nu puteti executa operatii asupra
tabelului deoarece gradinile zoologice'
          || ' sunt inchise toata ziua!');

      ELSIF ((LOWER(TO_CHAR(SYSDATE,'DY')) = LOWER('TUE') OR
LOWER(TO_CHAR(SYSDATE,'DY')) = LOWER('WED')
        OR LOWER(TO_CHAR(SYSDATE,'DY')) = LOWER('THU') OR
LOWER(TO_CHAR(SYSDATE,'DY')) = LOWER('MON')
        OR LOWER(TO_CHAR(SYSDATE,'DY')) = LOWER('FRI'))
        AND ((TO_CHAR(SYSDATE,'HH24')) < 9 OR (TO_CHAR(SYSDATE,'HH24')) >
18))
      THEN
        RAISE_APPLICATION_ERROR(-20002,'Operatiile asupra tabelului sunt permise
doar in timpul programul de lucru!');

      END IF;
END;
/
```

Declansare:

```
INSERT INTO ANGAJAT
VALUES(1000,1,3,'Mihai','Eugen',2810005074151,5641,to_date('16/11/2015','dd/mm/yyyy'))
);
-- Marti la ora 17:35 -> 1 row inserted.
```

```
-----  
INSERT INTO ANGAJAT
VALUES(1001,2,3,'Dumitrashe','Larisa',2810005084151,5641,to_date('20/01/2017','dd/mm/
yyyy'));
-- Daca se incerca randul de mai sus pe userul 'TEST':
-- ORA-20000: Nu aveti dreptul sa executati operatii asupra tabelului!
```

```
-----  
UPDATE ANGAJAT
SET prenume = 'Eugen'
WHERE id_angajat = 1000;
-- Marti la ora 20:
-- ORA-20002: Operatiile asupra tabelului sunt permise doar in programul de lucru!
```

```
-----  
DELETE FROM ANGAJAT
WHERE id_angajat = 1000;
-- Sambata:
-- ORA-20001: Nu puteti executa operatii asupra tabelului deoarece gradinile zoologice sunt
inchise toata ziua!
```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Title Bar:** Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql
- Connections:** Oracle Database 11g, Atasie_Oana_SQL
- SQL Worksheet:** The main pane displays the SQL code for creating a trigger:

```
CREATE OR REPLACE TRIGGER exercitiu10
  BEFORE INSERT OR DELETE OR UPDATE ON angajat
BEGIN
  IF USER <> UPPER('proiect')
    THEN
      RAISE_APPLICATION_ERROR(-20000, 'Nu aveti dreptul sa executati operatii asupra tabelului!');
    ELSIF (LOWER(TO_CHAR(SYSDATE, 'DY')) = LOWER('SAT')) OR (LOWER(TO_CHAR(SYSDATE, 'DY')) = LOWER('SUN'))
    THEN
      RAISE_APPLICATION_ERROR(-20001, ' Nu puteti executa operatii asupra tabelului deoarece gradinile zoologice'
                                || ' sunt inchise toata ziua');

    ELSIF ((LOWER(TO_CHAR(SYSDATE, 'DY')) = LOWER('TUE') OR LOWER(TO_CHAR(SYSDATE, 'DY')) = LOWER('WED'))
           OR LOWER(TO_CHAR(SYSDATE, 'DY')) = LOWER('THU') OR LOWER(TO_CHAR(SYSDATE, 'DY')) = LOWER('MON')
           OR LOWER(TO_CHAR(SYSDATE, 'DY')) = LOWER('FRI'))
           AND ((TO_CHAR(SYSDATE, 'HH24')) < 9 OR (TO_CHAR(SYSDATE, 'HH24')) > 18)
    THEN
      RAISE_APPLICATION_ERROR(-20002, 'Operatiile asupra tabelului sunt permise doar in timpul programul de lucru');

    END IF;
END;
```
- Script Output:** Task completed in 0.069 seconds
- Status Bar:** Trigger EXERCITIU10 compiled

5:35:39 PM
Tuesday, January 5, 2021

January 2021						
Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Today
Add an event or reminder
No events

```

BEGIN
    -- DECLASARE
    INSERT INTO ANGAJAT VALUES(1000,1,3,'Mihai','Eugen',2810005074151,5641,to_date('16/11/2015','dd/mm/yyyy'));
    -- Marti la ora 17:35 -> 1 row inserted.

    INSERT INTO ANGAJAT VALUES(1001,2,3,'Dumitache','Larisa',2810005084151,5641,to_date('20/01/2017','dd/mm/yyyy'));

    UPDATE ANGAJAT
    SET prenume = 'Eugen'
    WHERE id_angajat = 1000;
    -- Marti la ora 20:00
    -- ORA-20000: Nu aveti dreptul sa executati operatii asupra tabelului!
    -- ORA-20002: Operatiile asupra tabelului sunt permise doar in programul de lucru

    DELETE FROM ANGAJAT
    WHERE id_angajat = 1000;
    -- Duminica
    -- ORA-20001: Nu puteti executa operatii asupra tabelului deoarece gradinile zoologice sunt inchise toata ziua

    ALTER TRIGGER exercitiul10 DISABLE;
END;
/

```

Script Output X | Query Result X
Task completed in 0.077 seconds
1 row inserted.

```

ELSIF ((LOWER(TO_CHAR(SYSDATE,'DY')) = LOWER('TUE') OR LOWER(TO_CHAR(SYSDATE,'DY')) = LOWER('WED')
        OR LOWER(TO_CHAR(SYSDATE,'DY')) = LOWER('THU') OR LOWER(TO_CHAR(SYSDATE,'DY')) = LOWER('MON')
        OR LOWER(TO_CHAR(SYSDATE,'DY')) = LOWER('FRI'))
        AND ((TO_CHAR(SYSDATE,'HH24')) < 9 OR (TO_CHAR(SYSDATE,'HH24')) > 19))
THEN
    RAISE_APPLICATION_ERROR(-20000,'Operatiile asupra tabelului sunt permise doar in timpul programului de lucru!');

END IF;
/

```

-- DECLASARE
INSERT INTO ANGAJAT VALUES(1000,1,3,'Mihai','Eugen',2810005074151,5641,to_date('16/11/2015','dd/mm/yyyy'));
-- Marti la ora 17:35 -> 1 row inserted.

-- Daca se incerca randul de mai sus pe userul 'TEST':
-- ORA-20000: Nu aveti dreptul sa executati operatii asupra tabelului!
-- ORA-20002: Operatiile asupra tabelului sunt permise doar in programul de lucru

UPDATE ANGAJAT
SET prenume = 'Eugen'
WHERE id_angajat = 1000;

Script Output X
Task completed in 0.038 seconds

```

Error starting at line : 966 in command -
INSERT INTO ANGAJAT VALUES(1001,2,3,'Dumitache','Larisa',2810005084151,5641,to_date('20/01/2017','dd/mm/yyyy'))
Error report -
ORA-20000: Nu aveti dreptul sa executati operatii asupra tabelului!
ORA-06512: at "TEST.EXERCITIU10", line 4
ORA-04088: error during execution of trigger 'TEST.EXERCITIU10'

```

The screenshot shows the Oracle SQL Developer interface. On the left, a script named 'Atasie_Oana_SQL.sql' is open in the 'Script Output' tab, displaying an error message:

```
Error starting at line : 976 in command -
DELETE FROM ANGAJAT
WHERE id_angajat = 1000
Error report -
ORA-20001: Nu puteti executa operatii asupra tabelului decarece gradinile zoologice sunt inchise toata ziua
ORA-06512: at "PROIECT.EXERCITIU10", line 8
ORA-04088: error during execution of trigger 'PROIECT.EXERCITIU10'
```

On the right, a Windows taskbar calendar for January 2021 is visible, showing the date as Saturday, January 9, 2021, with the number 9 highlighted.

11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

Politica lanțului de grădini zoologice presupune următoarea regulă: salariul unui angajat nu poate depăși 20% din salariul mediu al funcției ocupate la zoo-ul la care lucrează.

```
CREATE OR REPLACE PACKAGE pachet_exercitiu11
IS
    TYPE record_ex11 IS RECORD
    (
        nr_angajati NUMBER,
        salariu_total angajat.salariu%TYPE,
        idul_functie angajat.id_functie%TYPE,
        idul_zoo angajat.id_zoo%TYPE
    );
    TYPE tabel_ex11 IS TABLE OF record_ex11;
    x tabel_ex11;
END;
/
```

```
CREATE OR REPLACE TRIGGER exercitiu11_trigger_comanda
BEFORE INSERT OR UPDATE ON angajat
BEGIN
```

```

SELECT count(id_angajat), sum(salariu), id_functie, id_zoo
BULK COLLECT INTO pachet_exercitiu11.x
FROM angajat
GROUP BY id_functie, id_zoo;

END;
/

CREATE OR REPLACE TRIGGER exercitiu11_trigger_linie
BEFORE INSERT OR UPDATE ON angajat
FOR EACH ROW
BEGIN

FOR i IN pachet_exercitiu11.x.FIRST .. pachet_exercitiu11.x.LAST LOOP
    IF pachet_exercitiu11.x(i).idul_functie = :NEW.id_functie AND
pachet_exercitiu11.x(i).idul_zoo = :NEW.id_zoo
        AND
0.2*(pachet_exercitiu11.x(i).salariu_total/pachet_exercitiu11.x(i).nr_angajati) < :NEW.salariu
    THEN
        IF INSERTING THEN
            RAISE_APPLICATION_ERROR(-20000,'Angajatul care se doreste a fi inserat
are salariul mai mare decat
                20% din salariul mediu pe functia ocupata la zoo-ul la care
lucreaza.');
        ELSIF UPDATING THEN
            RAISE_APPLICATION_ERROR(-20001,'Angajatul care se doreste a fi
modificat are salariul mai mare decat
                20% din salariul mediu pe functia ocupata la zoo-ul la care
lucreaza.');
        END IF;
    END IF;
END LOOP;
END;
/

```

Declansare:

```

INSERT INTO ANGAJAT VALUES
(300,3,2,'Murariu','Claudia',2865015084111,5000,to_date('13/07/2015','dd/mm/yyyy'));
-- Angajatul care se doreste a fi inserat are salariul mai mare decat

```

-- 20% din salariul mediu pe functia ocupata la zoo-ul la care lucreaza.

UPDATE ANGAJAT

SET salariu = 800

WHERE id_angajat = 10;

-- 1 row updated.

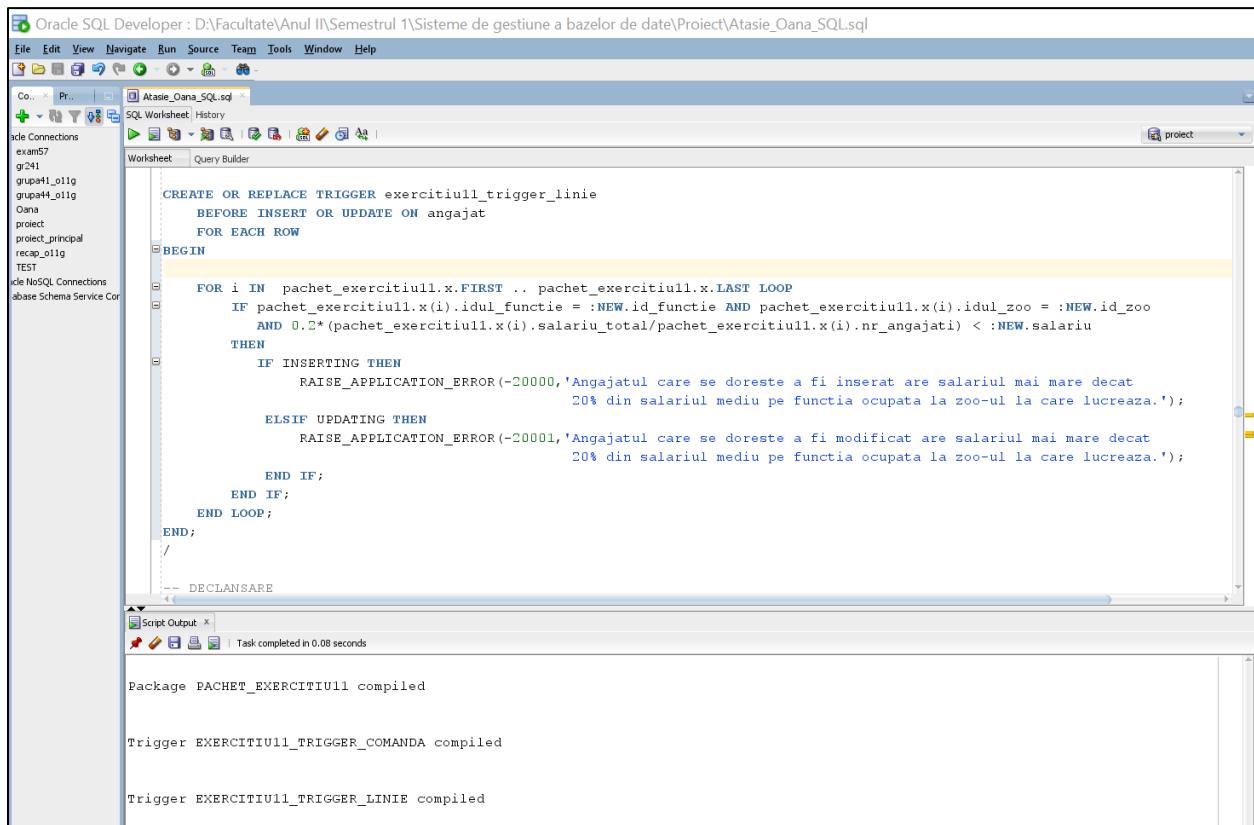
UPDATE ANGAJAT

SET salariu = 1020

WHERE id_angajat = 17;

-- Angajatul care se doreste a fi modificat are salariul mai mare decat

-- 20% din salariul mediu pe functia ocupata la zoo-ul la care lucreaza.



The screenshot shows the Oracle SQL Developer interface with the following details:

- Title Bar:** Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql
- Toolbar:** File Edit View Navigate Run Source Team Tools Window Help
- Connections:** exam57, gr241, grupa41_o11g, grupa44_o11g, Oana, project, project_principal, recap_o11g, TEST
- SQL Worksheet:** History, Worksheet, Query Builder
- Code Area:** The code is a CREATE OR REPLACE TRIGGER statement for the 'angajat' table. It uses a FOR EACH ROW loop to iterate over rows in the 'pachet_exercitiu11.x' package. Inside the loop, it checks if the department function ID matches the new function ID and if the average salary of employees in that department is less than 20% of the new salary. If so, it raises an application error. The trigger is named 'exercitiu11_trigger_linie'. The code also includes comments explaining the logic.
- Output Area:** Shows the results of the compilation:
 - Package PACHET_EXERCITIU11 compiled
 - Trigger EXERCITIU11_TRIGGER_COMANDA compiled
 - Trigger EXERCITIU11_TRIGGER_LINIE compiled

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

    THEN
        IF INSERTING THEN
            RAISE_APPLICATION_ERROR(-20000,'Angajatul care se doreste a fi inserat are salariul mai mare decat
                                         20% din salariul mediu pe functia ocupata la zoo-ul la care lucreaza.');
        ELSIF UPDATING THEN
            RAISE_APPLICATION_ERROR(-20001,'Angajatul care se doreste a fi modificat are salariul mai mare decat
                                         20% din salariul mediu pe functia ocupata la zoo-ul la care lucreaza.');
        END IF;
    END LOOP;
END;
/

-- DECLANSARE
INSERT INTO ANGAJAT VALUES (300,3,2,'Murariu','Claudia',2865015084111,5000,to_date('13/07/2015','dd/mm/yyyy'));
-- Angajatul care se doreste a fi inserat are salariul mai mare decat
-- 20% din salariul mediu pe functia ocupata la zoo-ul la care lucreaza.

UPDATE ANGAJAT
SET salariu = 800
WHERE id_angajat = 10;
-- 1 row updated.

Script Output: x | Task completed in 0.055 seconds

Error starting at line : 1,037 in command -
INSERT INTO ANGAJAT VALUES (300,3,2,'Murariu','Claudia',2865015084111,5000,to_date('13/07/2015','dd/mm/yyyy'))
Error report -
ORA-20000: Angajatul care se doreste a fi inserat are salariul mai mare decat
                                         20% din salariul mediu pe functia ocupata la zoo-ul la care lucreaza.
ORA-06512: at "PROIECT.EXERCITIU11_TRIGGER_LINIE", line 8
ORA-04088: error during execution of trigger 'PROIECT.EXERCITIU11_TRIGGER_LINIE'

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

    THEN
        ELSIF UPDATING THEN
            RAISE_APPLICATION_ERROR(-20001,'Angajatul care se doreste a fi modificat are salariul mai mare decat
                                         20% din salariul mediu pe functia ocupata la zoo-ul la care lucreaza.');
        END IF;
    END LOOP;
END;
/

-- DECLANSARE
INSERT INTO ANGAJAT VALUES (300,3,2,'Murariu','Claudia',2865015084111,5000,to_date('13/07/2015','dd/mm/yyyy'));
-- Angajatul care se doreste a fi inserat are salariul mai mare decat
-- 20% din salariul mediu pe functia ocupata la zoo-ul la care lucreaza.

UPDATE ANGAJAT
SET salariu = 800
WHERE id_angajat = 10;
-- 1 row updated.

UPDATE ANGAJAT
SET salariu = 1000
WHERE id_angajat = 25;
-- 1 row updated.

Script Output: x | Task completed in 0.053 seconds

1 row updated.

```

The screenshot shows the Oracle SQL Developer interface with a script named 'Atasie_Oana_SQL.sql' open in the Worksheet tab. The script contains SQL commands for updating employee salaries and disabling triggers. The 'Script Output' tab at the bottom shows the execution results, including errors related to trigger execution.

```
END;
/
-- DECLANSARE
INSERT INTO ANGAJAT VALUES (300,3,2,'Murariu','Claudia',2865015084111,5000,to_date('13/07/2015','dd/mm/yyyy'));
-- Angajatul care se doreste a fi inserat are salariul mai mare decat
-- 20% din salariul mediu pe functia ocupata la zoo-ul la care lucreaza.

-----
UPDATE ANGAJAT
SET salariu = 800
WHERE id_angajat = 10;
-- 1 row updated.

-----
UPDATE ANGAJAT
SET salariu = 1020
WHERE id_angajat = 17;
-- Angajatul care se doreste a fi modificat are salariul mai mare decat
-- 20% din salariul mediu pe functia ocupata la zoo-ul la care lucreaza.

-----
ALTER TRIGGER exercitiu11_trigger_comanda DISABLE;
ALTER TRIGGER exercitiu11_trigger_linie DISABLE;
```

Script Output x | Task completed in 0.072 seconds

```
Error starting at line : 1,046 in command -
UPDATE ANGAJAT
SET salariu = 1020
WHERE id_angajat = 17
Error report -
ORA-20001: Angajatul care se doreste a fi modificat are salariul mai mare decat
          20% din salariul mediu pe functia ocupata la zoo-ul la care lucreaza.
ORA-06512: at "PROIECT.EXERCITIU11_TRIGGER_LINIE", line 11
ORA-04088: error during execution of trigger 'PROIECT.EXERCITIU11_TRIGGER_LINIE'
```

12. Definiți un trigger de tip LDD. Declanșați trigger-ul.

Patronul dorește reținerea evidenței modificărilor bazei de date, de tip CREATE, ALTER și DROP, iar pentru aceasta, este esențială crearea unui tabel care să rețină informații despre aceste modificări în momentul în care se fac: de fiecare dată când se utilizează o comandă CREATE, ALTER sau DROP, informații despre această comandă vor fi adăugate în tabel.

CREATE TABLE info

```
(utilizator VARCHAR2(50),
baza_de_date VARCHAR2(30),
eveniment VARCHAR2(50),
nume_object VARCHAR2(30),
tip_object VARCHAR2(30),
data DATE
);
```

CREATE OR REPLACE TRIGGER exercitiu12

```
AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
    INSERT INTO info VALUES (SYS.LOGIN_USER, SYS.DATABASE_NAME,
    SYS.SYSEVENT,
                SYS.DICTIONARY_OBJ_NAME, SYS.DICTIONARY_OBJ_TYPE,
    SYSDATE);
END;
/

-- DECLANSARE
CREATE TABLE test AS SELECT * FROM ZOO;
-- Table TEST created.

-----
ALTER TABLE test
DROP COLUMN adresa;
-- Table TEST altered.

-----
DROP TABLE test;
-- Table TEST dropped.

select * from info;
```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```
CREATE TABLE info
(
    utilizator VARCHAR2(50),
    baza_de_date VARCHAR2(30),
    eveniment VARCHAR2(50),
    nume_object VARCHAR2(30),
    tip_object VARCHAR2(30),
    data DATE
);

CREATE OR REPLACE TRIGGER exercitiul2
    AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
    INSERT INTO info VALUES (SYS.LOGIN_USER, SYS.DATABASE_NAME, SYS.SYSEVENT,
        SYS.DICTIONARY_OBJ_NAME, SYS.DICTIONARY_OBJ_TYPE, SYSDATE);
END;
/
-- DECLANSARE
CREATE TABLE test AS SELECT * FROM ZOO;

Table INFO created.

Trigger EXERCITIU12 compiled
```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```
END;
/
-- DECLANSARE
CREATE TABLE test AS SELECT * FROM ZOO;
-- Table TEST created.

ALTER TABLE test
DROP COLUMN adresa;
-- Table TEST altered.

DROP TABLE test;
-- Table TEST dropped.

select * from info;

ALTER TRIGGER exercitiul2 DISABLE;

-- EXERCITIU 13
-- 13. Definiti un pacchet care sa contina toate obiectele definite in cadrul proiectului.
```

Table TEST created.

Table TEST altered.

Table TEST dropped.

```

END;
/
-- DECLANSARE
CREATE TABLE test AS SELECT * FROM ZOO;
-- Table TEST created.

ALTER TABLE test
DROP COLUMN adresa;
-- Table TEST altered.

DROP TABLE test;
-- Table TEST dropped.

select * from info;

ALTER TRIGGER exercitiu12 DISABLE;

-- EXERCITIU 13
-- 13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului

```

Script Output:

UTILIZATOR	DATA_CARE	EVENIMENT	NUME_OBJECT	TIP_OBJECT	DATA
PROIECT OR11G		CREATE	TEST	TABLE	09-JAN-21
PROIECT OR11G		ALTER	TEST	TABLE	09-JAN-21
PROIECT OR11G		DROP	TEST	TABLE	09-JAN-21

13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

CREATE OR REPLACE PACKAGE exercitiu13

IS

-- exercitiu 6

PROCEDURE exercitiu6;

-- exercitiu 7

PROCEDURE exercitiu7;

-- exercitiu 8

TYPE exercitiu8_colectie IS TABLE OF VARCHAR2(50);

FUNCTION exercitiu8 (denum_categ IN categorie.denumire%TYPE)

RETURN exercitiu8_colectie;

-- exercitiu 9

TYPE exercitiu9_colectie IS TABLE OF VARCHAR2(100);

PROCEDURE exercitiu9 (denum_furnizor IN furnizor.denumire%TYPE,

colectie OUT exercitiu9_colectie

);

```

END exercitiu13;
/

CREATE OR REPLACE PACKAGE BODY exercitiu13
IS
    -- exercitiu 6
    PROCEDURE exercitiu6
    IS
        TYPE record_exercitiu6 IS RECORD(
            idul_zoo zoo.id_zoo%TYPE,
            idul_angajat angajat.id_angajat%TYPE
        );
        TYPE tablou_indexat_ex6 IS TABLE OF record_exercitiu6 INDEX BY
        PLS_INTEGER;
        colectie_ex6 tablou_indexat_ex6;
        salariuV_ang_curent angajat.salariu%TYPE;
        salariuN_ang_curent angajat.salariu%TYPE;

        BEGIN
            SELECT id_zoo, id_angajat
            BULK COLLECT INTO colectie_ex6
            FROM angajat a
            WHERE data_angajarii = (SELECT MIN(data_angajarii)
                FROM angajat
                WHERE id_zoo = a.id_zoo
            )
            GROUP BY id_zoo, id_angajat
            ORDER BY 1;

            FOR i IN colectie_ex6.FIRST .. colectie_ex6.LAST LOOP
                IF colectie_ex6.EXISTS(colectie_ex6.PRIOR(i)) AND
                    colectie_ex6(i).idul_zoo = colectie_ex6(colectie_ex6.PRIOR(i)).idul_zoo
                THEN
                    DBMS_OUTPUT.PUT_LINE('Angajatul ' || colectie_ex6(i).idul_angajat || '
este unul dintre angajatii cu'
                                || ' cea mai mare vechime de la zoo-ul cu id-ul ' ||
colectie_ex6(i).idul_zoo || '.');
                ELSIF colectie_ex6.EXISTS(colectie_ex6.NEXT(i)) AND
                    colectie_ex6(i).idul_zoo = colectie_ex6(colectie_ex6.NEXT(i)).idul_zoo
                THEN
                    DBMS_OUTPUT.PUT_LINE('Angajatul ' || colectie_ex6(i).idul_angajat || '
este cel mai nou angajat din zoo-ul cu id-ul ' ||
colectie_ex6(i).idul_zoo || '.');
                END IF;
            END LOOP;
        END;
    
```

```

        THEN
            DBMS_OUTPUT.PUT_LINE('Angajatul ' || colectie_ex6(i).idul_angajat || '
este unul dintre angajatii cu'
                || ' cea mai mare vechime de la zoo-ul cu id-ul ' || '
colectie_ex6(i).idul_zoo || '.');
        ELSE
            DBMS_OUTPUT.PUT_LINE('Angajatul ' || colectie_ex6(i).idul_angajat || ' are
cea mai mare vechime de la'
                || ' zoo-ul cu id-ul ' || colectie_ex6(i).idul_zoo || '.');
        END IF;

        SELECT salariu
        INTO salariuV_ang_curent
        FROM angajat
        WHERE id_angajat = colectie_ex6(i).idul_angajat;

        UPDATE angajat
        SET salariu = salariu + ROUND(salariu * 0.1)
        WHERE id_angajat = colectie_ex6(i).idul_angajat
        RETURNING salariu INTO salariuN_ang_curent;

        DBMS_OUTPUT.PUT_LINE('Angajatul avea salariul ' || salariuV_ang_curent || ',
iar acum are salariul '
                || salariuN_ang_curent || '.');
        DBMS_OUTPUT.NEW_LINE;
    END LOOP;
END exercitiu6;

```

```

-- exercitiu 7
PROCEDURE exercitiu7 IS
    idul_zoo zoo.id_zoo%TYPE;
    nume_zoo zoo.denumire%TYPE;
    oras_zoo zoo.oras%TYPE;
    idul_anim animal.id_animal%TYPE;
    anim_data_nasterii animal.data_nasterii%TYPE;
    denumire_specie specie.denumire%TYPE;
    numar_top number :=0;
    data_veche animal.data_nasterii%TYPE;

```

```

CURSOR cursor_zoo IS
    SELECT id_zoo, denumire, oras
    FROM zoo;

CURSOR cursor_animal(parametru_zoo_id zoo.id_zoo%TYPE) IS
    SELECT a.id_animal, data_nasterii, s.denumire
    FROM animal a JOIN specie s ON (a.id_specie = s.id_specie)
    WHERE id_zoo = parametru_zoo_id AND MONTHS_BETWEEN('01-FEB-
2021',data_nasterii) >= 6*12
    ORDER BY data_nasterii;

BEGIN
    OPEN cursor_zoo;
    LOOP
        FETCH cursor_zoo INTO idul_zoo, nume_zoo, oras_zoo;
        EXIT WHEN cursor_zoo%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Gradina zoologica "' || nume_zoo || " din orasul '" ||
oras_zoo
                || ' are urmatorul top trei: ');
        DBMS_OUTPUT.PUT_LINE('-----');
        numar_top :=0;
        SELECT SYSDATE
        INTO data_veche
        FROM dual;
        OPEN cursor_animal(idul_zoo);
        LOOP
            FETCH cursor_animal INTO idul_anim, anim_data_nasterii, denumire_specie;
            IF anim_data_nasterii <> data_veche THEN
                numar_top := numar_top + 1;
            END IF;
            data_veche := anim_data_nasterii;
            EXIT WHEN numar_top > 3 OR cursor_animal%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE(numar_top || '. Animalul cu id-ul ' || idul_anim ||
'si denumirea speciei ' || denumire_specie
                || ' are data de nastere ' || anim_data_nasterii || '.');
        END LOOP;
        IF numar_top = 0 THEN
            DBMS_OUTPUT.PUT_LINE( 'Nu exista animale cu varsta mai mare de 6 ani
la zoo-ul "' || nume_zoo || '".');
        ELSIF numar_top < 3 THEN

```

```

        DBMS_OUTPUT.PUT_LINE( 'Nu se poate ajunge la top 3 la zoo-ul "' || 
nume_zoo || '".');
        END IF;
        CLOSE cursor_animal;
        DBMS_OUTPUT.NEW_LINE;
    END LOOP;
    CLOSE cursor_zoo;
END exercitiu7;

-- exercitiu 8
FUNCTION exercitiu8
( denum_categ IN categorie.denumire%TYPE)
RETURN exercitiu8_colectie IS
colectie exercitiu8_colectie;
i INTEGER;
j INTEGER;
aux VARCHAR2(50);
exceptie_categ_gresita EXCEPTION;
verificare VARCHAR2(10);

BEGIN

    SELECT 'Exista'
    INTO verificare
    FROM categorie
    WHERE LOWER(denumire) = LOWER(denum_categ); -- pentru a verifica daca
exista categoria data

    SELECT latime*inaltime*lungime
    BULK COLLECT INTO colectie
    FROM categorie c JOIN specie s ON (c.id_categorie = s.id_categorie)
        JOIN tarc t ON(s.id_tarc = t.id_tarc)
    WHERE LOWER(c.denumire) = LOWER(denum_categ)
    GROUP BY latime*inaltime*lungime, t.id_tarc
    ORDER BY 1;

    IF colectie.COUNT = 0 -- pentru a verifica daca se poate ajunge de la categorii la tarc
    THEN RAISE exceptie_categ_gresita;
    END IF;

```

```

i := colectie.FIRST;
WHILE i <= colectie.LAST LOOP
    IF i > colectie.FIRST THEN
        j := colectie.PRIOR(i);
        aux := colectie(j);
        IF colectie(i) = aux OR (INSTR(aux,'X') <> 0 AND SUBSTR(aux, 0,
LENGTH(aux)-2) = colectie(i)) THEN
            IF INSTR(aux,'X') <> 0 THEN
                aux := CONCAT(SUBSTR(aux, 0, LENGTH(aux)-1),
TO_CHAR(TO_NUMBER(SUBSTR(aux, LENGTH(aux)))+1));
            ELSE
                aux := CONCAT(aux,'X2');
            END IF;
            colectie(j) := aux;
            colectie.DELETE(i);
            i := colectie.NEXT(i);
        ELSE
            i := colectie.NEXT(i);
        END IF;
    ELSE
        i := colectie.NEXT(i);
    END IF;
END LOOP;

RETURN colectie;

EXCEPTION
    WHEN no_data_found
        THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista categoria data ca
parametru.');
    WHEN exceptie_categ_gresita
        THEN RAISE_APPLICATION_ERROR(-20001, 'Pentru categoria data nu se pot
parcurge tabele.' );
    WHEN OTHERS
        THEN RAISE_APPLICATION_ERROR(-20002, 'Alta eroare!');
END exercitiu8;

-- exercitiu 9
PROCEDURE exercitiu9
    (denum_furnizor IN furnizor.denumire%TYPE, colectie OUT exercitiu9_colectie)

```

```

IS
CURSOR cursor_ex9(parametru furnizor.denumire%TYPE) IS
    SELECT m.id_mancare, s.denumire,
ROUND(shc.cantitate_medie_zilnica*100/ead.cantitate,5)
    FROM furnizor f JOIN este_asigurata_de ead ON (f.id_furnizor =
ead.id_furnizor)
        JOIN mancare m ON (ead.id_mancare = m.id_mancare)
        JOIN se_hraneste_cu shc ON (m.id_mancare = shc.id_mancare)
        JOIN specie s ON (shc.id_specie = s.id_specie)
    WHERE LOWER(f.denumire) = LOWER(parametru)
    GROUP BY m.id_mancare, s.denumire,
ROUND(shc.cantitate_medie_zilnica*100/ead.cantitate,5)
    ORDER BY 1;

```

```

idul_mancare mancare.id_mancare%TYPE;
denum_specie specie.denumire%TYPE;
procent NUMBER(10,5);
ok NUMBER;
i NUMBER;
procent2 VARCHAR2(10);
verificare VARCHAR2(30);
arrive_false EXCEPTION;

```

```
BEGIN
```

```

SELECT 'Exista'
INTO verificare
FROM furnizor
WHERE LOWER(denumire) = LOWER(denum_furnizor);

```

```

SELECT min(s.denumire)
INTO verificare
FROM furnizor f JOIN este_asigurata_de ead ON (f.id_furnizor = ead.id_furnizor)
    JOIN mancare m ON (ead.id_mancare = m.id_mancare)
    JOIN se_hraneste_cu shc ON (m.id_mancare = shc.id_mancare)
    JOIN specie s ON (shc.id_specie = s.id_specie)
WHERE LOWER(f.denumire) = LOWER(denum_furnizor);

```

```

IF verificare IS NULL
    THEN RAISE arrive_false;

```

```

END IF;

OPEN cursor_ex9(denum_furnizor);
    colectie := exercitiu9_colectie();
    ok := 0;
LOOP
    FETCH cursor_ex9 INTO idul_mancare, denum_specie, procent;
    EXIT WHEN cursor_ex9%NOTFOUND;

    IF procent < 1 THEN
        procent2 := '0' || TO_CHAR(procent);
    ELSE
        procent2 := TO_CHAR(procent);
    END IF;

    IF ok = 0 THEN
        colectie.EXTEND;
        i := colectie.FIRST;
        colectie(i) := TO_CHAR(idul_mancare) || ':' || denum_specie || ',' || procent2 ||
    );
        ok := 1;
    ELSE
        IF SUBSTR(colectie(i), 0, 1) = idul_mancare THEN
            colectie(i) := colectie(i) || ',(' || denum_specie || ',' || procent2 || ')';
        ELSE
            colectie.EXTEND;
            i := colectie.LAST;
            colectie(i) := TO_CHAR(idul_mancare) || ':' || denum_specie || ',' || procent2
    || ')';
        END IF;
    END IF;
END LOOP;
CLOSE cursor_ex9;
EXCEPTION
    WHEN zero_divide
        THEN RAISE_APPLICATION_ERROR(-20000, 'S-a incercat impartirea prin
0.');
    WHEN no_data_found
        THEN RAISE_APPLICATION_ERROR(-20001, 'Nu exista furnizorul dat ca
parametru.');

```

```

WHEN arrive_false
    THEN RAISE_APPLICATION_ERROR(-20002, 'Nu se pot parurge
tabelele.');
WHEN too_many_rows
    THEN RAISE_APPLICATION_ERROR(-20003, 'Exista mai multi furnizori
cu denumirea data.');
WHEN others
    THEN RAISE_APPLICATION_ERROR(-20004, 'Alta eroare.');
END exercitiu9;

END exercitiu13;
/

-- Apelare exercitiu 6
EXECUTE exercitiu13.exercitiu6();
-- Apelare exercitiu 7
EXECUTE exercitiu13.exercitiu7();
-- Apelare exercitiu 8
DECLARE
    rezultat exercitiu13.exercitiu8_colecie;
    i NUMBER;
BEGIN
    rezultat := exercitiu13.exercitiu8('REPTiLE');
    i := rezultat.FIRST;
    WHILE i <= rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
        i := rezultat.NEXT(i);
    END LOOP;
END;
/
-- 4500000X2
-- 9000000X3
-- 2250000
DECLARE
    rezultat exercitiu13.exercitiu8_colecie;
    i NUMBER;
BEGIN
    rezultat := exercitiu13.exercitiu8('Mamifer');
    i := rezultat.FIRST;
    WHILE i <= rezultat.LAST LOOP

```

```

        DBMS_OUTPUT.PUT_LINE(resultat(i));
        i := resultat.NEXT(i);
    END LOOP;
END;
/
-- ORA-20000: Nu exista categoria data ca parametru.

DECLARE
    resultat exercitiu13.exercitiu8_colecie;
    i NUMBER;
BEGIN
    resultat := exercitiu13.exercitiu8('Insectivore');
    i := resultat.FIRST;
    WHILE i <= resultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(resultat(i));
        i := resultat.NEXT(i);
    END LOOP;
END;
/
-- ORA-20001: Pentru categoria data nu se pot parurge tabele.
-- Apelare exercitiu 9

DECLARE
    resultat exercitiu13.exercitiu9_colecie;
BEGIN
    exercitiu13.exercitiu9('FERMA COROG',resultat);
    FOR i IN resultat.FIRST .. resultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(resultat(i));
    END LOOP;
END;
/
-- 3:(Armasar, 1.66667), (Ponei, 1.33333), (Porc mistret, 0.33333)
-- 8:(Killi, 0.00444), (Perus, 0.02963), (Sanitari, 0.00741), (Urs brun, 0.16296)

DECLARE
    resultat exercitiu13.exercitiu9_colecie;
BEGIN
    exercitiu13.exercitiu9('TEST',resultat);
    FOR i IN resultat.FIRST .. resultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(resultat(i));
    END LOOP;
END;
/

```

```
-- ORA-20001: Nu exista furnizorul dat ca parametru.  
DECLARE  
    rezultat exercitiu13.exercitiu9_colecie;  
BEGIN  
    exercitiu13.exercitiu9('Global Food',rezultat);  
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP  
        DBMS_OUTPUT.PUT_LINE(rezultat(i));  
    END LOOP;  
END;  
/  
-- ORA-20003: Exista mai multi furnizori cu denumirea data.  
DECLARE  
    rezultat exercitiu13.exercitiu9_colecie;  
BEGIN  
    exercitiu13.exercitiu9('Food Food',rezultat);  
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP  
        DBMS_OUTPUT.PUT_LINE(rezultat(i));  
    END LOOP;  
END;  
/  
-- ORA-20002: Nu se pot parcurge tabelele.  
DECLARE  
    rezultat exercitiu13.exercitiu9_colecie;  
BEGIN  
    exercitiu13.exercitiu9('American Zoo',rezultat);  
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP  
        DBMS_OUTPUT.PUT_LINE(rezultat(i));  
    END LOOP;  
END;  
/  
-- ORA-20000: S-a incercat impartirea prin 0.
```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

Connections | Atasie_Oana_SQL.sql
File Edit View Navigate Run Source Team Tools Window Help
+ Oracle Connections
  + exam57
  + g241
  + grupa41_01ig
  + grupa44_01ig
  + Oana
  + project
  + project_principa
  + recap_01ig
  + TEST
+ Oracle NoSQL Conn
Database Schema Se

Worksheet | Query Builder
IF SUBSTR(colectie(i), 0, 1) = idul_mancare THEN
    colectie(i) := colectie(i) || ', (' || denum_specie || ',' || procent2 || ')';
ELSE
    colectie.EXTEND;
    i := colectie.LAST;
    colectie(i) := TO_CHAR(idul_mancare) || ':(' || denum_specie || ',' || procent2 || ')';
END IF;
END LOOP;
CLOSE cursor_ex9;
EXCEPTION
    WHEN zero_divide
        THEN RAISE_APPLICATION_ERROR(-20000, 'S-a incercat impartirea prin 0.');
    WHEN no_data_found
        THEN RAISE_APPLICATION_ERROR(-20001, 'Nu exista furnizorul dat ca parametru.');
    WHEN arrive_false
        THEN RAISE_APPLICATION_ERROR(-20002, 'Nu se pot parcurge tabelele.');
    WHEN too_many_rows
        THEN RAISE_APPLICATION_ERROR(-20003, 'Exista mai multi furnizori cu denumirea data.');
    WHEN others
        THEN RAISE_APPLICATION_ERROR(-20004, 'Alta eroare.');
END exercitiu9;
END exercitiu13;
/

```

Script Output | Task completed in 0.15 seconds

```

Package EXERCITIU13 compiled

Package Body EXERCITIU13 compiled

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

Connections | Atasie_Oana_SQL.sql
File Edit View Navigate Run Source Team Tools Window Help
+ Oracle Connections
  + exam57
  + g241
  + grupa41_01ig
  + grupa44_01ig
  + Oana
  + project
  + project_principa
  + recap_01ig
  + TEST
+ Oracle NoSQL Conn
Database Schema Se

Worksheet | Query Builder
    THEN RAISE_APPLICATION_ERROR(-20003, 'EXISTA MAI MULTI FURNIZORI CU ID-UL 1.');
    WHEN others
        THEN RAISE_APPLICATION_ERROR(-20004, 'Alta eroare.');
END exercitiu9;
END exercitiu13;
/
-- Apelare exercitiu 6
EXECUTE exercitiu13.exercitiu6();
-- Apelare exercitiu 7
EXECUTE exercitiu13.exercitiu7();
-- Apelare exercitiu 8
DECLARE
    rezultat exercitiu13.exercitiu8_colectie;
    i NUMBER;
BEGIN
    rezultat := exercitiu13.exercitiu8('REPTILE');
    i := rezultat.FIRST;
    WHILE i <= rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
        i := rezultat.NEXT(i);
    END LOOP;
END;

```

Items Output | Buffer Size:20000

```

project x

rezultat(1) Angajatul 49 este unul dintre angajatii cu cea mai mare vechime de la zoo-ul cu id-ul 1. Angajatul avea salariul 3626, iar acum are salariul 3589.

rezultat(2) Angajatul 78 este unul dintre angajatii cu cea mai mare vechime de la zoo-ul cu id-ul 1. Angajatul avea salariul 3050, iar acum are salariul 3355.

rezultat(3) Angajatul 56 are cea mai mare vechime de la zoo-ul cu id-ul 2. Angajatul avea salariul 4563, iar acum are salariul 5019.

rezultat(4) Angajatul 30 are cea mai mare vechime de la zoo-ul cu id-ul 3. Angajatul avea salariul 3677, iar acum are salariul 4045.

rezultat(5) Angajatul 65 este unul dintre angajatii cu cea mai mare vechime de la zoo-ul cu id-ul 4. Angajatul avea salariul 6320, iar acum are salariul 6952.

rezultat(6) Angajatul 80 este unul dintre angajatii cu cea mai mare vechime de la zoo-ul cu id-ul 4. Angajatul avea salariul 7132, iar acum are salariul 7845.

rezultat(7) Angajatul 81 este unul dintre angajatii cu cea mai mare vechime de la zoo-ul cu id-ul 4. Angajatul avea salariul 5721, iar acum are salariul 6293.

rezultat(8) Angajatul 74 este unul dintre angajatii cu cea mai mare vechime de la zoo-ul cu id-ul 5. Angajatul avea salariul 5030, iar acum are salariul 5533.

rezultat(9) Angajatul 75 este unul dintre angajatii cu cea mai mare vechime de la zoo-ul cu id-ul 5. Angajatul avea salariul 4000, iar acum are salariul 4400.

```

Script Output | Task completed in 0.091 seconds

```

PL/SQL procedure successfully completed.

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Project\Atasie_Oana_SQL.sql

```

-- Apelare exercitiu 6
EXECUTE exercitiu3.exercitiu6();
-- Apelare exercitiu 7
EXECUTE exercitiu3.exercitiu7();
-- Apelare exercitiu 8
DECLARE
    rezultat exercitiu3.exercitiu8_colectie;
    i NUMBER;
BEGIN
    rezultat := exercitiu3.exercitiu8('REPTILE');
    i := rezultat.FIRST;
    WHILE i <= rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
        i := rezultat.NEXT(i);
    END LOOP;
END;
/

```

PL/SQL procedure successfully completed.

Dbsm Output

```

Gradina zoologica "Gradina Zoologica din Bucuresti" din orasul Bucuresti are urmatorul top tre
-----
1. Animalul cu id-ul 89 si denumirea speciei Discus are data de nastere 09-MAR-14.
2. Animalul cu id-ul 55 si denumirea speciei Gorila are data de nastere 11-MAR-14.
3. Animalul cu id-ul 56 si denumirea speciei Perus are data de nastere 11-MAR-14.
4. Animalul cu id-ul 72 si denumirea speciei Gorila are data de nastere 02-MAY-14.

Gradina zoologica "Gradina Zoologica din Timisoara" din orasul Timisoara are urmatorul top tre
-----
1. Animalul cu id-ul 89 si denumirea speciei Discus are data de nastere 09-MAR-14.
2. Animalul cu id-ul 55 si denumirea speciei Gorila are data de nastere 11-MAR-14.
3. Animalul cu id-ul 56 si denumirea speciei Perus are data de nastere 11-MAR-14.
4. Animalul cu id-ul 72 si denumirea speciei Gorila are data de nastere 02-MAY-14.

Gradina zoologica "Gradina Zoologica din Oradea" din orasul Oradea are urmatorul top tre
-----
1. Animalul cu id-ul 57 si denumirea speciei Camila are data de nastere 29-DEC-13.
2. Animalul cu id-ul 23 si denumirea speciei Zebra are data de nastere 23-MAR-14.
3. Animalul cu id-ul 42 si denumirea speciei Ponei are data de nastere 10-MAY-14.
4. Animalul cu id-ul 86 si denumirea speciei Armasar are data de nastere 10-MAY-14.

Gradina zoologica "Gradina Zoologica din Resita" din orasul Resita are urmatorul top tre
-----
1. Animalul cu id-ul 64 si denumirea speciei Crocodil are data de nastere 30-MAY-14.
2. Animalul cu id-ul 79 si denumirea speciei Aligatosauridae are data de nastere 02-OCT-14.
3. Animalul cu id-ul 78 si denumirea speciei Crocodil are data de nastere 16-NOV-14.

Gradina zoologica "Gradina Zoologica din Targu Mures" din orasul Targu Mures are urmatorul top tre
-----
1. Animalul cu id-ul 40 si denumirea speciei Crocodil are data de nastere 05-JUN-14.
2. Animalul cu id-ul 8 si denumirea speciei Leu are data de nastere 16-JUN-14.
Nu se poate ajunge la top 3 la zoo-ul "Gradina Zoologica din Targu Mures".

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Project\Atasie_Oana_SQL.sql

```

-- Apelare exercitiu 6
EXECUTE exercitiu3.exercitiu6();
-- Apelare exercitiu 7
EXECUTE exercitiu3.exercitiu7();
-- Apelare exercitiu 8
DECLARE
    rezultat exercitiu3.exercitiu8_colectie;
    i NUMBER;
BEGIN
    rezultat := exercitiu3.exercitiu8('REPTILE');
    i := rezultat.FIRST;
    WHILE i <= rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
        i := rezultat.NEXT(i);
    END LOOP;
END;
/

```

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

Dbsm Output

```

45000000X2
90000000X3
22500000

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

Connections | Atasie_Oana_SQL.sql | SQL Worksheet | History | project
+ exam57
+ gr241
+ grupa41_olig
+ grupa44_olig
+ Oana
+ project
+ project_principa
+ recap_olig
+ TEST
Oracle NoSQL Conn
Database Schema Se

Worksheet | Query Builder
END LOOP;
/
-- 4500000X2
-- 9000000X3
-- 22500000
DECLARE
    rezultat exercitiu3.exercitiu8_colectie;
    i NUMBER;
BEGIN
    rezultat := exercitiu3.exercitiu8('Mamifer');
    i := rezultat.FIRST;
    WHILE i <= rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
        i := rezultat.NEXT(i);
    END LOOP;
/
-- ORA-20000: Nu exista categoria data ca parametru.
DECLARE
    rezultat exercitiu3.exercitiu8_colectie;
    i NUMBER;
BEGIN
    rezultat := exercitiu3.exercitiu8('Insectivore');
    i := rezultat.FIRST;
    WHILE i <= rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
        i := rezultat.NEXT(i);
    END LOOP;
/
-- ORA-20000: Nu exista categoria data ca parametru.
ORA-06512: at "PROIECT.EXERCITIU3", line 170
ORA-06512: at line 5
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
        was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
        the application administrator or DBA for more information.

```

Script Output x

Task completed in 0.082 seconds

Error report -

ORA-20000: Nu exista categoria data ca parametru.

ORA-06512: at "PROIECT.EXERCITIU3", line 170

ORA-06512: at line 5

20000. 00000 - "%s"

*Cause: The stored procedure 'raise_application_error' was called which causes this error to be generated.

*Action: Correct the problem as described in the error message or contact the application administrator or DBA for more information.

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

Connections | Atasie_Oana_SQL.sql | SQL Worksheet | History | project
+ exam57
+ gr241
+ grupa41_olig
+ grupa44_olig
+ Oana
+ project
+ project_principa
+ recap_olig
+ TEST
Oracle NoSQL Conn
Database Schema Se

Worksheet | Query Builder
DBMS_OUTPUT.PUT_LINE(rezultat(i));
    i := rezultat.NEXT(i);
END LOOP;
/
-- ORA-20000: Nu exista categoria data ca parametru.
DECLARE
    rezultat exercitiu3.exercitiu8_colectie;
    i NUMBER;
BEGIN
    rezultat := exercitiu3.exercitiu8('Insectivore');
    i := rezultat.FIRST;
    WHILE i <= rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
        i := rezultat.NEXT(i);
    END LOOP;
/
-- ORA-20001: Pentru categoria data nu se pot parcurge table.
-- Apelare exercitiu 9
DECLARE
    rezultat exercitiu3.exercitiu9_colectie;
BEGIN
    exercitiu3.exercitiu9('FERMA COROG',rezultat);
/
DBMS_OUTPUT.PUT_LINE(rezultat(i));
    i := rezultat.NEXT(i);
END LOOP;
END;
Error report -
ORA-20001: Pentru categoria data nu se pot parcurge table.
ORA-06512: at "PROIECT.EXERCITIU3", line 172
ORA-06512: at line 5

```

Script Output x

Task completed in 0.086 seconds

The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Connections' and 'Reports' sections, with 'Atasie_Oana_SQL.sql' selected in the 'Connections' tree. The main workspace contains a 'Query Builder' window with the following PL/SQL code:

```
i := rezultat.NEXT(i);
END LOOP;
/
-- ORA-20001: Pentru categoria data nu se pot parcurge table.
-- Apelare exercitiu 5
DECLARE
    rezultat exercitiul3.exercitiu9_colectie;
BEGIN
    exercitiul3.exercitiu9('FERMA COROG',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;

-- 3: (Armasar, 1.66667), (Ponei, 1.33333), (Porc mistret, 0.33333)
-- 8: (Kili, 0.00444), (Perus, 0.02963), (Sanitari, 0.00741), (Urs brun, 0.16296)

DECLARE
    rezultat exercitiul3.exercitiu9_colectie;
BEGIN
    exercitiul3.exercitiu9('TEST',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
```

The 'Script Output' window at the bottom shows the results of the execution:

```
PL/SQL procedure successfully completed.
```

The 'Doms Output' window on the right shows the output of the DBMS_OUTPUT.PUT_LINE statements:

```
3: (Armasar, 1.66667), (Ponei, 1.33333), (Porc mistret, 0.33333)
8: (Kili, 0.00444), (Perus, 0.02963), (Sanitari, 0.00741), (Urs brun, 0.16296)
```

The screenshot shows the Oracle SQL Developer interface with the following details:

- File Edit View Navigate Run Source Team Tools Window Help**
- Connections**: Atasie_Oana_SQL_sd
- Oracle Connections**: Atasie_Oana_SQL_sd, p2m7, p241, grusel_olig, grusel44_olig, Oana, project, project_principala, recap_olig, TEST, Oracle NoSQL Conn.
- Database Schema Set**
- Worksheet**: Query Builder, rezultat exercitiul3.exercitiu9_colectie;
- Script Output**: Task completed in 0.063 seconds
- Logs**: Error report - ORA-20001: Nu exista furnizorul dat ca parametru. ORA-06512: at "PROJECT.EXERCITIUL3", line 253 ORA-06512: at line 4

```
begin
    exercitiul3.exercitiu9('PERMA COROG',rezultat);
    for i in rezultat.FIRST .. rezultat.LAST loop
        dbms_output.put_line(rezultat(i));
    end loop;
end;
/
-- 3:(Armasar, 1.66667), (Ponei, 1.33333), (Porc mistret, 0.33333)
-- 8:(Kili, 0.00444), (Perus, 0.02963), (Sanitari, 0.00741), (Urs brun, 0.16296)
declare
    rezultat exercitiul3.exercitiu9_colectie;
begin
    exercitiul3.exercitiu9('TEST',rezultat);
    for i in rezultat.FIRST .. rezultat.LAST loop
        dbms_output.put_line(rezultat(i));
    end loop;
end;
/
-- ORA-20001: Nu exista furnizorul dat ca parametru.
declare
    rezultat exercitiul3.exercitiu9_colectie;
begin
    exercitiul3.exercitiu9('Global Food',rezultat);
    for i in rezultat.FIRST .. rezultat.LAST loop
        dbms_output.put_line(rezultat(i));
    end loop;
end;
/
Error report -
ORA-20001: Nu exista furnizorul dat ca parametru.
ORA-06512: at "PROJECT.EXERCITIUL3", line 253
ORA-06512: at line 4
```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

Connections     SQL Worksheet: History     project
+ Oracle Connections     Atase_Oana_SQL.sql
  + exam57     SQL Worksheet     Query Builder
  + g241     +
  + grupa41_olig     +
  + grupa44_olig     +
  + Oana     +
  + project     +
  + project_principa
  + recap_olig     +
  + TEST     +
  + Oracle NoSQL Conn
  + Database Schema Se

Reports     Script Output: x
+ All Reports     | Task completed in 0.105 seconds
+ Analytic View Re
+ Data Dictionary I
+ Data Modeler Re
+ OLAP Reports
+ TimesTen Report
+ User Defined Re

Worksheet     Query Builder
  +
  +-- 3: (Armasar, 1.66667), (Ponei, 1.33333), (Porc mistret, 0.33333)
  +-- 8: (Killi, 0.00444), (Perus, 0.02963), (Sanitari, 0.00741), (Urs brun, 0.16296)

  DECLARE
    resultat exercitiu3.exercitiu9_colectie;
  BEGIN
    exercitiu3.exercitiu9('TEST',resultat);
    FOR i IN resultat.FIRST .. resultat.LAST LOOP
      DBMS_OUTPUT.PUT_LINE(resultat(i));
    END LOOP;
  END;
  /
  +-- ORA-20001: Nu exista furnizorul dat ca parametru.

  DECLARE
    resultat exercitiu3.exercitiu9_colectie;
  BEGIN
    exercitiu3.exercitiu9('Global Food',resultat);
    FOR i IN resultat.FIRST .. resultat.LAST LOOP
      DBMS_OUTPUT.PUT_LINE(resultat(i));
    END LOOP;
  END;
  /
  +-- ORA-20003: Există mai multi furnizori cu denumirea data.

next ADO

  Script Output: x
  + Task completed in 0.105 seconds
    FOR i IN resultat.FIRST .. resultat.LAST LOOP
      DBMS_OUTPUT.PUT_LINE(resultat(i));
    END LOOP;
  END;
  Error report -
ORA-20003: Există mai multi furnizori cu denumirea data.
ORA-06512: at "PROIECT.EXERCITIU13", line 257
ORA-06512: at line 4

```

Oracle SQL Developer : D:\Facultate\Anul II\Semestrul 1\Sisteme de gestiune a bazelor de date\Proiect\Atasie_Oana_SQL.sql

```

Connections     SQL Worksheet: History     project
+ Oracle Connections     Atase_Oana_SQL.sql
  + exam57     SQL Worksheet     Query Builder
  + g241     +
  + grupa41_olig     +
  + grupa44_olig     +
  + Oana     +
  + project     +
  + project_principa
  + recap_olig     +
  + TEST     +
  + Oracle NoSQL Conn
  + Database Schema Se

Reports     Script Output: x
+ All Reports     | Task completed in 0.106 seconds
+ Analytic View Re
+ Data Dictionary I
+ Data Modeler Re
+ OLAP Reports
+ TimesTen Report
+ User Defined Re

Worksheet     Query Builder
  +
  +-- ORA-20001: Nu există furnizorul dat ca parametru.

  DECLARE
    resultat exercitiu3.exercitiu9_colectie;
  BEGIN
    exercitiu3.exercitiu9('Global Food',resultat);
    FOR i IN resultat.FIRST .. resultat.LAST LOOP
      DBMS_OUTPUT.PUT_LINE(resultat(i));
    END LOOP;
  END;
  /
  +-- ORA-20003: Există mai multi furnizori cu denumirea data.

  DECLARE
    resultat exercitiu3.exercitiu9_colectie;
  BEGIN
    exercitiu3.exercitiu9('Food Food',resultat);
    FOR i IN resultat.FIRST .. resultat.LAST LOOP
      DBMS_OUTPUT.PUT_LINE(resultat(i));
    END LOOP;
  END;
  /
  +-- ORA-20002: Nu se pot parcurge tablele.

  DECLARE
    resultat exercitiu3.exercitiu9_colectie;
  DETERM

  Script Output: x
  + Task completed in 0.106 seconds
    exercitiu3.exercitiu9('Food Food',resultat);
    FOR i IN resultat.FIRST .. resultat.LAST LOOP
      DBMS_OUTPUT.PUT_LINE(resultat(i));
    END LOOP;
  END;
  Error report -
ORA-20002: Nu se pot parcurge tablele.
ORA-06512: at "PROIECT.EXERCITIU13", line 255
ORA-06512: at line 4

```

```

DECLARE
    rezultat exercitiu13.exercitiu9_colectie;
BEGIN
    exercitiu13.exercitiu9('Food Food',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- ORA-20003: Există mai mulți furnizori cu denumirea data.
DECLARE
    rezultat exercitiu13.exercitiu9_colectie;
BEGIN
    exercitiu13.exercitiu9('American Zoo',rezultat);
    FOR i IN rezultat.FIRST .. rezultat.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(rezultat(i));
    END LOOP;
END;
/
-- ORA-20000: Nu se pot parcurge tabelele.

```

Script Output x | Task completed in 0.051 seconds

```

END;
Error report -
ORA-00000: S-a incercat impartirea prin 0.
ORA-06512: at "PROJECT.EXERCITIU13", line 251
ORA-06512: at line 4
20000. 00000 - "%s"
*Cause: The stored procedure 'raise_application_error'
        was called which causes this error to be generated.
*Action: Correct the problem as described in the error message or contact
        the application administrator or DBA for more information.

```

14. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare pentru acțiuni integrate.

Patronul zoo-ului dorește să crească anumite salarii ale angajaților cu salariile cele mai mici, însă pentru a decide cui să i se modifice salariul pentru început, este nevoie de următoarele informații:

- *în cadrul fiecărei funcții, pentru fiecare zoo, se dorește cunoașterea salariului total*
- *în cadrul fiecărei funcții, pentru fiecare zoo, se dorește cunoașterea procentului pe care salariul minim îl reprezintă din salariul total.*

CREATE OR REPLACE PACKAGE exercitiu14
IS

CURSOR cursor_functie IS

```

SELECT id_functie, denumire
FROM functie;

```

PROCEDURE informatii;

```

FUNCTION functie_salariu_minim (idul_zoo zoo.id_zoo%TYPE, denum_functie
functie.denumire%TYPE)

```

```

RETURN angajat.salariu%TYPE;

```

```

END exercitiu14;
/

CREATE OR REPLACE PACKAGE BODY exercitiu14
IS

    FUNCTION functie_salariu_minim (idul_zoo zoo.id_zoo%TYPE, denum_functie
functie.denumire%TYPE)
        RETURN angajat.salariu%TYPE
    IS
        salariu_minim_zoo_functie angajat.salariu%TYPE;
    BEGIN

        SELECT MIN(salariu)
        INTO salariu_minim_zoo_functie
        FROM functie f JOIN angajat a ON (f.id_functie = a.id_functie)
        WHERE LOWER(denumire) = LOWER(denum_functie) AND id_zoo = idul_zoo;

        RETURN salariu_minim_zoo_functie;

    END functie_salariu_minim;

PROCEDURE informatii
IS
    TYPE record_sal_total IS RECORD(
        denumire_functie.denumire%TYPE,
        id_zoo zoo.id_zoo%TYPE,
        salariu_total angajat.salariu%TYPE
    );
    TYPE tab1 IS TABLE OF record_sal_total;
    TYPE tab2 IS TABLE OF tab1;
    t1 tab1;
    t2 tab2;
    i INTEGER;
    j INTEGER;
    salariu_minim angajat.salariu%TYPE;
    TYPE record_procent IS RECORD(
        id_zoo zoo.id_zoo%TYPE,
        procent_salariu_minim NUMBER(10),
        denumire_functie.denumire%TYPE
    );
    TYPE tab3 IS TABLE OF record_procent;
    TYPE tab4 IS TABLE OF tab3;
    t3 tab4;
    t4 tab4;
    v_record record_procent;

```

```

BEGIN

    t2 := tab2();
    t4 := tab4();

    FOR i IN cursor_functie LOOP

        SELECT MAX(denumire), id_zoo, SUM(salariu)
        BULK COLLECT INTO t1
        FROM functie f JOIN angajat a ON (f.id_functie = a.id_functie)
        WHERE f.id_functie = i.id_functie
        GROUP BY id_zoo;

        t2.EXTEND;
        t2(t2.LAST) := t1;

    END LOOP;

    FOR i IN t2.FIRST..t2.LAST LOOP
        t4.EXTEND;
        t4(t4.LAST) := tab3();
        FOR j IN t2(i).FIRST..t2(i).LAST LOOP
            DBMS_OUTPUT.PUT_LINE('La zoo-ul ' || t2(i)(j).id_zoo || ' functia ' ||
t2(i)(j).denumire || ' are salariul
total ' || t2(i)(j).salariu_total || ':');

            salariu_minim := functie_salariu_minim(t2(i)(j).id_zoo, t2(i)(j).denumire);

            v_record.id_zoo := t2(i)(j).id_zoo;
            v_record.procent_salariu_minim := ROUND(salariu_minim*100/
t2(i)(j).salariu_total);
            v_record.denumire := t2(i)(j).denumire;

            t4(t4.LAST).EXTEND;
            t4(t4.LAST)(t4(t4.LAST).LAST):= v_record;

        END LOOP;
        DBMS_OUTPUT.NEW_LINE;
    END LOOP;

    DBMS_OUTPUT.NEW_LINE;

    FOR i IN t4.FIRST..t4.LAST LOOP
        FOR j IN t4(i).FIRST..t4(i).LAST LOOP

```

```
DBMS_OUTPUT.PUT_LINE('La zoo-ul '|| t4(i).id_zoo || ' procentul pe care  
salariul minim
```

```
de pe functia '|| t4(i).denumire || ' il reprezinta din salariul total de la  
zoo-ul si functia respectiva este egal cu '|| t4(i).procent_salariu_minim || ');  
END LOOP;  
DBMS_OUTPUT.NEW_LINE;  
END LOOP;  
END informatii;  
END exercitiu14;  
/  
/
```

```
EXECUTE exercitiu14.informatii();
```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Connections:** Atane_Data_SQL* (selected)
- Worksheet:** Query Builder
- Code:**

```
t4(t4.LAST).EXTEND;
t4(t4.LAST)(t4(t4.LAST).LAST):= v_record;

END LOOP;
DBMS_OUTPUT.NEW_LINE;
END LOOP;

DBMS_OUTPUT.NEW_LINE;

FOR i IN t4.FIRST..t4.LAST LOOP
  FOR j IN t4(i).FIRST..t4(i).LAST LOOP
    DBMS_OUTPUT.PUT_LINE('La zoo-ul '|| t4(i).id_zoo || ' procentul pe care
de pe functia '|| t4(i).denumire || ' il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu '|| t4(i).procent_salariu_minim || '.');
  END LOOP;
  DBMS_OUTPUT.NEW_LINE;
END LOOP;
END informatii;
END exercitiu14;
/
EXECUTE exercitiu14.informatii();
```
- Script Output:**

```
Task completed in 0.065 seconds

Package EXERCITIU14 compiled

Package Body EXERCITIU14 compiled

PL/SQL procedure successfully completed.
```
- Output Window:** Displays the results of the DBMS_OUTPUT.PUT_LINE statements, listing the percentage of minimum salary for each function across all departments.

Oracle SQL Developer

```

t4(t4.LAST).EXTEND;
t4(t4.LAST).(t4(t4.LAST).LAST):= v_record;

END LOOP;
DBMS_OUTPUT.NEW_LINE;
END LOOP;

FOR i IN t4.FIRST..t4.LAST LOOP
  FOR j IN t4(i).FIRST..t4(i).LAST LOOP
    DBMS_OUTPUT.PUT_LINE('La zoo-ul ' || t4(i).j.id_zoo || ' procentul pe care
de pe functia ' || t4(i).j.denumire || ' il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu ' || t4(i).j.procent_salariu_minim || '.');
  END LOOP;
  DBMS_OUTPUT.NEW_LINE;
END LOOP;
END informatii;
/
EXECUTE exercitiu14.informatii();

```

Script Output

Task completed in 0.065 seconds

Package EXERCITIU14 compiled

Package Body EXERCITIU14 compiled

PL/SQL procedure successfully completed.

Items Output

La zoo-ul 5 functia Director General are salariul total 11227.
La zoo-ul 4 functia Director General are salariul total 11380.
La zoo-ul 3 functia Director General are salariul total 14360.

La zoo-ul 1 functia Veterinar are salariul total 17022.
La zoo-ul 2 functia Veterinar are salariul total 15419.
La zoo-ul 4 functia Veterinar are salariul total 11855.
La zoo-ul 5 functia Veterinar are salariul total 13955.
La zoo-ul 3 functia Veterinar are salariul total 12755.

La zoo-ul 1 procentul pe care salariul minim de pe functia Vanzator il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 100.
La zoo-ul 2 procentul pe care salariul minim de pe functia Vanzator il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 43.
La zoo-ul 5 procentul pe care salariul minim de pe functia Vanzator il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 48.
La zoo-ul 4 procentul pe care salariul minim de pe functia Vanzator il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 17.
La zoo-ul 3 procentul pe care salariul minim de pe functia Vanzator il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 100.

La zoo-ul 1 procentul pe care salariul minim de pe functia Ingrijitor il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 100.

de pe functia Ingrijitor il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 15.
La zoo-ul 2 procentul pe care salariul minim de pe functia Ingrijitor il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 9.
La zoo-ul 5 procentul pe care salariul minim de pe functia Ingrijitor il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 10.
La zoo-ul 4 procentul pe care salariul minim de pe functia Ingrijitor il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 11.
La zoo-ul 3 procentul pe care salariul minim de pe functia Ingrijitor il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 12.

La zoo-ul 1 procentul pe care salariul minim de pe functia Securitate il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 47.
La zoo-ul 2 procentul pe care salariul minim de pe functia Securitate il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 35.
La zoo-ul 5 procentul pe care salariul minim de pe functia Securitate il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 23.
La zoo-ul 4 procentul pe care salariul minim de pe functia Securitate il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 26.
La zoo-ul 3 procentul pe care salariul minim de pe functia Securitate il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 42.

La zoo-ul 1 procentul pe care salariul minim de pe functia Director General il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 100.
La zoo-ul 2 procentul pe care salariul minim de pe functia Director General il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 30.
La zoo-ul 5 procentul pe care salariul minim

Oracle SQL Developer

```

t4(t4.LAST).EXTEND;
t4(t4.LAST).(t4(t4.LAST).LAST):= v_record;

END LOOP;
DBMS_OUTPUT.NEW_LINE;
END LOOP;

FOR i IN t4.FIRST..t4.LAST LOOP
  FOR j IN t4(i).FIRST..t4(i).LAST LOOP
    DBMS_OUTPUT.PUT_LINE('La zoo-ul ' || t4(i).j.id_zoo || ' procentul pe care
de pe functia ' || t4(i).j.denumire || ' il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu ' || t4(i).j.procent_salariu_minim || '.');
  END LOOP;
  DBMS_OUTPUT.NEW_LINE;
END LOOP;
END informatii;
/
EXECUTE exercitiu14.informatii();

```

Script Output

Task completed in 0.065 seconds

Package EXERCITIU14 compiled

Package Body EXERCITIU14 compiled

PL/SQL procedure successfully completed.

Items Output

de pe functia Ingrijitor il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 15.
La zoo-ul 2 procentul pe care salariul minim de pe functia Ingrijitor il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 9.
La zoo-ul 5 procentul pe care salariul minim de pe functia Ingrijitor il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 10.
La zoo-ul 4 procentul pe care salariul minim de pe functia Ingrijitor il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 11.
La zoo-ul 3 procentul pe care salariul minim de pe functia Ingrijitor il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 12.

La zoo-ul 1 procentul pe care salariul minim de pe functia Securitate il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 47.
La zoo-ul 2 procentul pe care salariul minim de pe functia Securitate il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 35.
La zoo-ul 5 procentul pe care salariul minim de pe functia Securitate il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 23.
La zoo-ul 4 procentul pe care salariul minim de pe functia Securitate il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 26.
La zoo-ul 3 procentul pe care salariul minim de pe functia Securitate il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 42.

La zoo-ul 1 procentul pe care salariul minim de pe functia Director General il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 100.
La zoo-ul 2 procentul pe care salariul minim de pe functia Director General il reprezinta din salariul total de la zoo-ul si functia respectiva este egal cu 30.
La zoo-ul 5 procentul pe care salariul minim

Oracle SQL Developer

File Edit View Navigate Run Test Tools Window Help

Connections Oracle Connections Oracle NetSQL Conn Database Schema Set

Alarie_Dana_SQL.sql

SQL Worksheet History

Worksheet Query Builder

```
t4(t4.LAST).EXTEND;
t4(t4.LAST)(t4(t4.LAST).LAST):= v_record;

END LOOP;
DBMS_OUTPUT.NEW_LINE;
END LOOP;

DBMS_OUTPUT.NEW_LINE;

FOR i IN t4.FIRST..t4.LAST LOOP
  FOR j IN t4(i).FIRST..t4(i).LAST LOOP
    DBMS_OUTPUT.PUT_LINE('La zoo-ul ' || t4(i).j.id_zoo || ' procentul pe care
de pe functia ' || t4(i).j.denumire || ' il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu ' || t4(i).j.procent_salariu_minim || '.');
  END LOOP;
  DBMS_OUTPUT.NEW_LINE;
END LOOP;
END informatii;
/
EXECUTE exercitiu4.informatii();
```

Script Output x Task completed in 0.065 seconds

Package EXERCITIU14 compiled

Package Body EXERCITIU14 compiled

PL/SQL procedure successfully completed.

Output:

La zoo-ul 3 procentul pe care salariul minim
de pe functia Securitate il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu 42.

La zoo-ul 1 procentul pe care salariul minim
de pe functia Director General il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu 100.

La zoo-ul 2 procentul pe care salariul minim
de pe functia Director General il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu 30.

La zoo-ul 5 procentul pe care salariul minim
de pe functia Director General il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu 47.

La zoo-ul 4 procentul pe care salariul minim
de pe functia Director General il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu 44.

La zoo-ul 3 procentul pe care salariul minim
de pe functia Director General il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu 31.

La zoo-ul 1 procentul pe care salariul minim
de pe functia Veterinar il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu 18.

La zoo-ul 2 procentul pe care salariul minim
de pe functia Veterinar il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu 37.

La zoo-ul 4 procentul pe care salariul minim
de pe functia Veterinar il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu 45.

La zoo-ul 5 procentul pe care salariul minim
de pe functia Veterinar il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu 22.

La zoo-ul 3 procentul pe care salariul minim
de pe functia Veterinar il reprezinta din salariul total de la
zoo-ul si functia respectiva este egal cu 33.