**Gestionarea unei pizzerii**

**Gestionarea unei baze de date**

**Radulescu Alexia-Bianca**

**Grupa 233**

Gestionarea unei pizzerii

Ex1. Prezentați pe scurt baza de date (utilitatea ei).

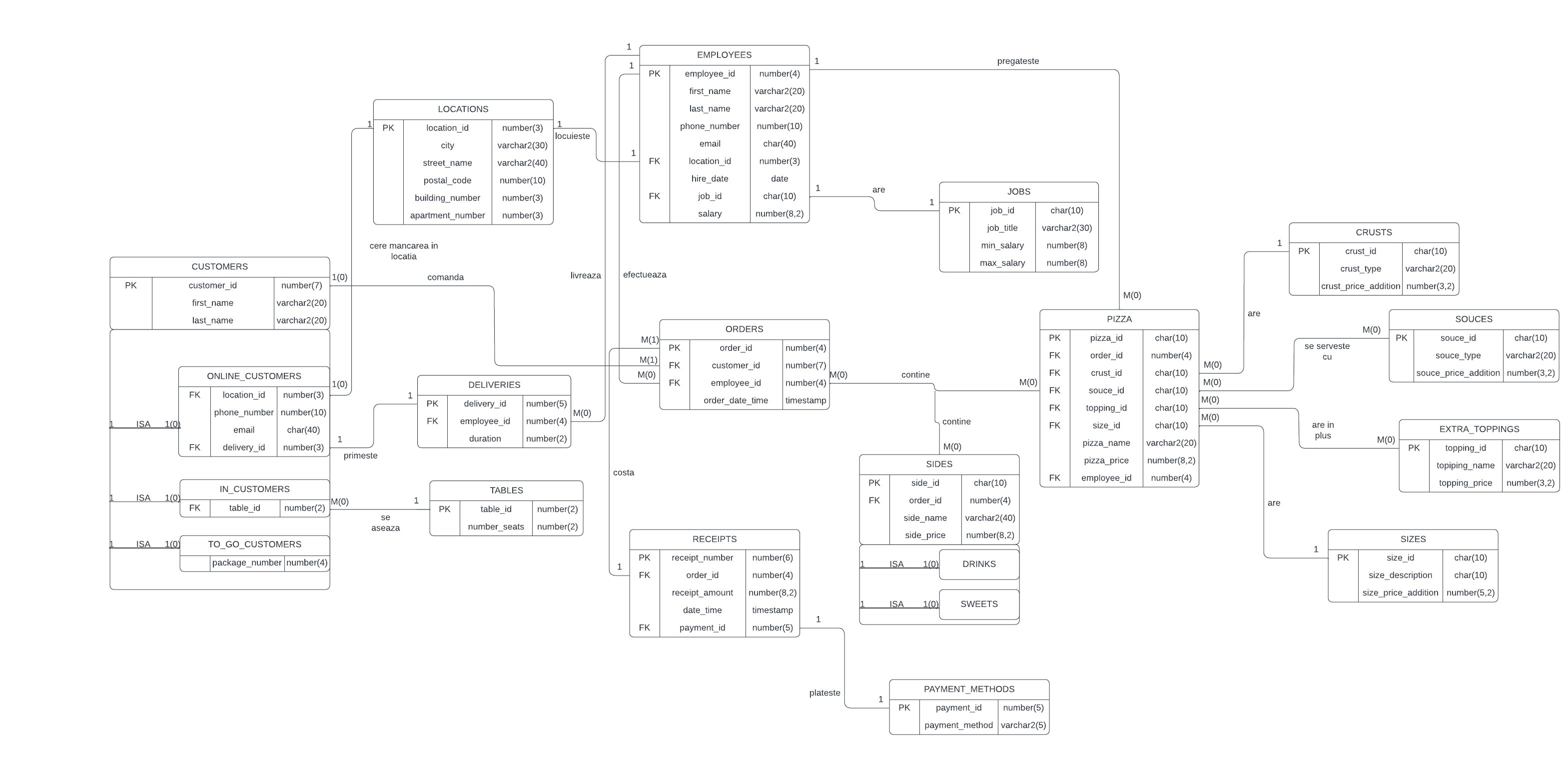
Baza de date contine informatii atat despre angajatii pizzeriei cat si despre clientii sai. Clientii au optiunea de a comanda online sau direct in restaurant unde pot sa ocupe o masa, pentru a se bucura de mancarea comandata, sau pot lua mancarea la pachet. Clientii pot alege sa comande pizza si/sau bauturi si/sau desert.

La orice pizza comandata clientul are de ales dintre mai multe dimensiuni, tipuri de blat si sosuri si are optiunea de a adauga topinguri. Pentru fiecare comanda este emisa o nota de plata care poate fi platita cu cash sau cu card. Comenzile comandate online sunt livrate de un angajat al pizzeriei care are jobul de curier.

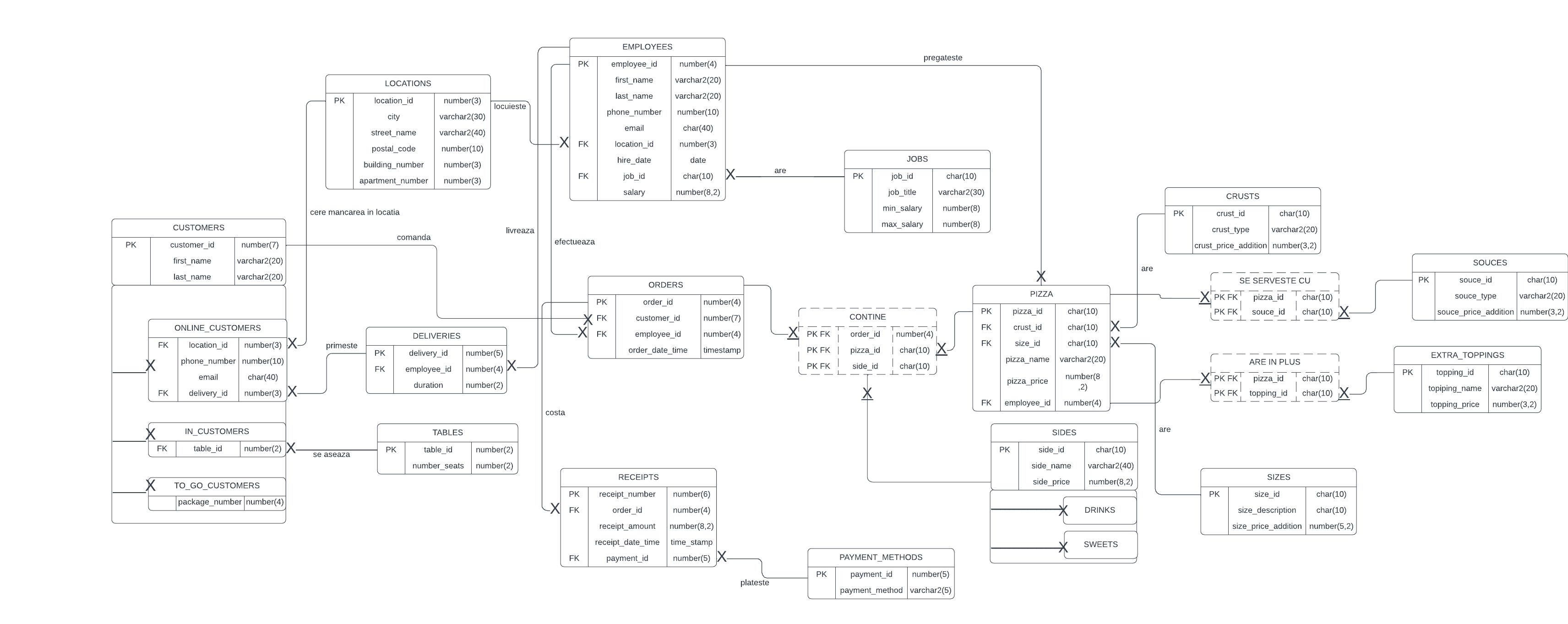
La fiecare comanda exista un angajat care are rolul de a prelua cererile clientilor si a le transmite bucatariei. Iar fiecare pizza preparata are asociat un bucatar.

Aceasta baza de date are rolul de a tine evidenta clientilor, comenzilor si banilor incasati prin fiecare comanda, dar si angajatilor si participarii lor in fiecare comanda.

Ex2. Realizați diagrama entitate-relație (ERD).



Ex3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.



Ex4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

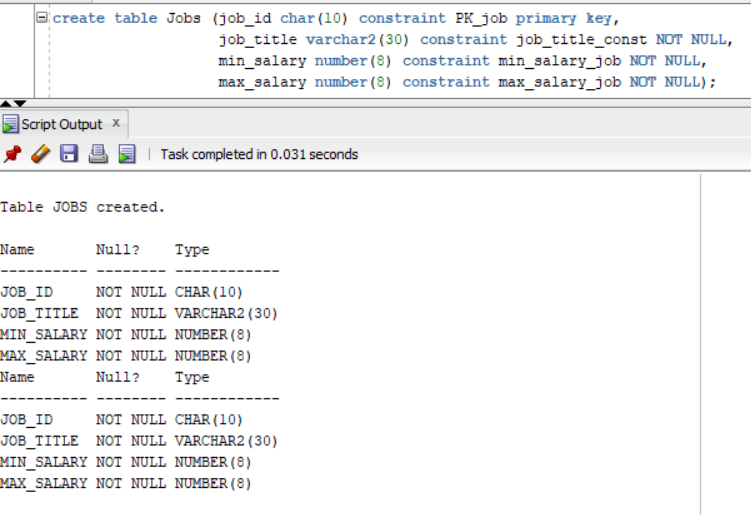
**JOBS**

create table Jobs (job\_id char(10) constraint PK\_job primary key,

job\_title varchar2(30) constraint job\_title\_const NOT NULL,

min\_salary number(8) constraint min\_salary\_job NOT NULL,

max\_salary number(8) constraint max\_salary\_job NOT NULL);



**LOCATIONS**

create table Locations (location\_id number(3) constraint PK\_location primary key,

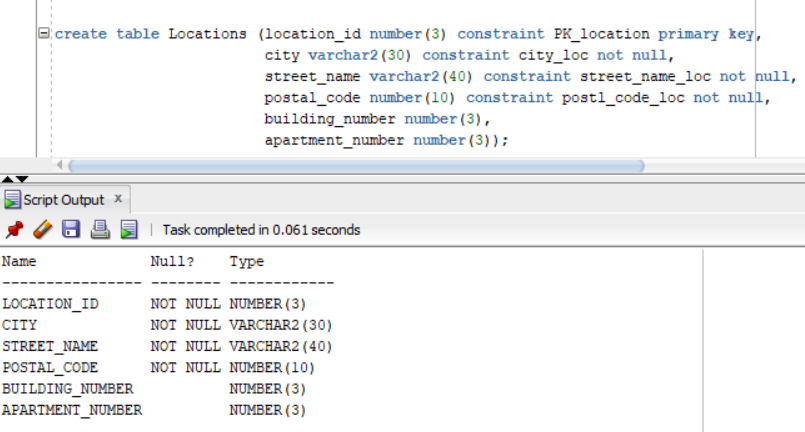
city varchar2(30) constraint city\_loc not null,

street\_name varchar2(40) constraint street\_name\_loc not null,

postal\_code number(10) constraint postl\_code\_loc not null,

building\_number number(3),

apartment\_number number(3));



**EMPLOYEES**

create table Employees (employee\_id number(4) constraint PK\_employee primary key,

first\_name varchar2(20) constraint last\_name\_emp not null,

last\_name varchar2(20) constraint first\_name\_emp not null,

phone\_number number(10) unique,

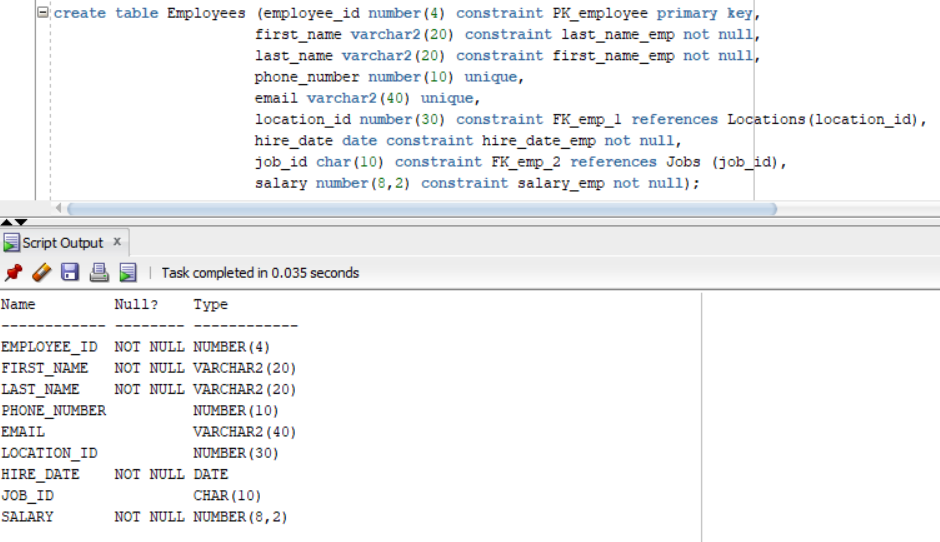
email varchar2(40) unique,

location\_id number(30) constraint FK\_emp\_1 references Locations(location\_id),

hire\_date date constraint hire\_date\_emp not null,

job\_id char(10) constraint FK\_emp\_2 references Jobs (job\_id),

salary number(8,2) constraint salary\_emp not null);

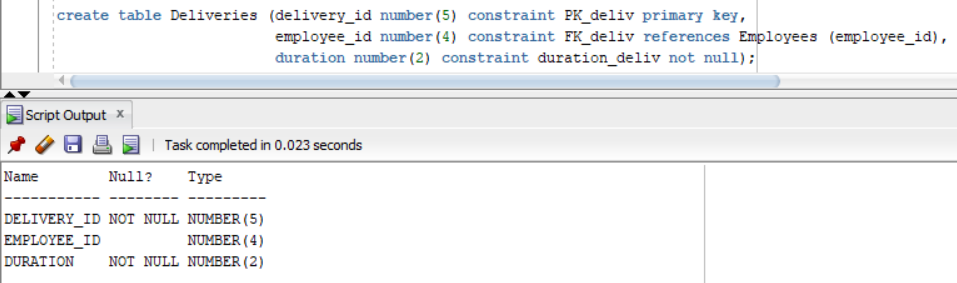


**DELIVERIES**

create table Deliveries (delivery\_id number(5) constraint PK\_deliv primary key,

employee\_id number(4) constraint FK\_deliv references Employees (employee\_id),

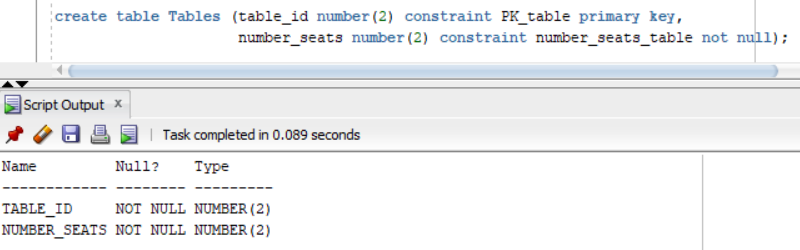
duration number(2) constraint duration\_deliv not null);



**TABLES**

create table Tables (table\_id number(2) constraint PK\_table primary key,

number\_seats number(2) constraint number\_seats\_table not null);

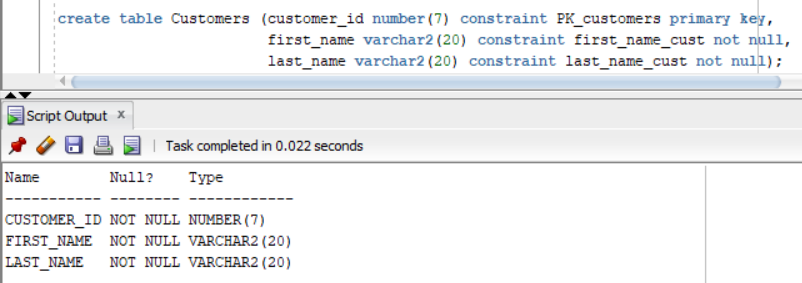


**CUSTOMERS**

create table Customers (customer\_id number(7) constraint PK\_customers primary key,

first\_name varchar2(20) constraint first\_name\_cust not null,

last\_name varchar2(20) constraint last\_name\_cust not null);



**ONLINE\_CUSTOMERS**

create table Online\_Customers

as select \* from Customers;

delete online\_customers;

alter table Online\_Customers

add location\_id number(3)constraint FK\_ocust references Locations(location\_id);

alter table Online\_Customers

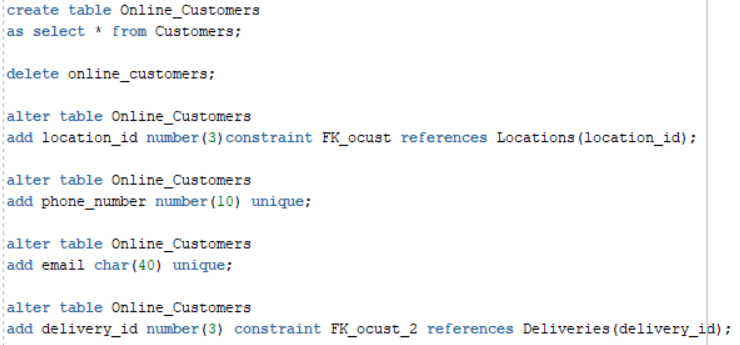
add phone\_number number(10) unique;

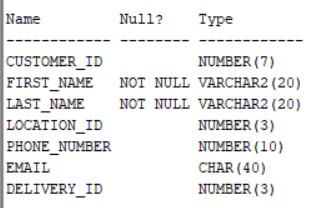
alter table Online\_Customers

add email char(40) unique;

alter table Online\_Customers

add delivery\_id number(3) constraint FK\_ocust\_2 references Deliveries(delivery\_id);





**IN\_CUSTOMERS**

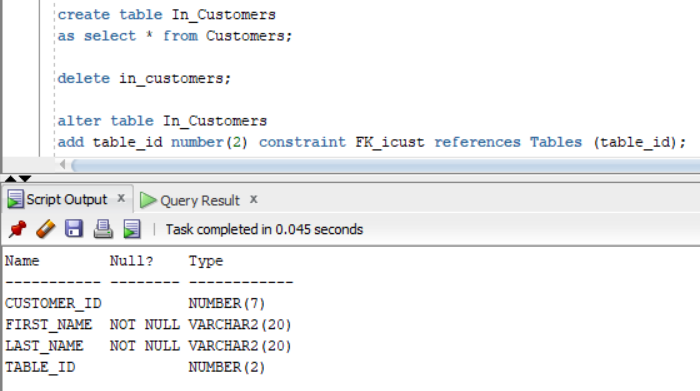
create table In\_Customers

as select \* from Customers;

delete in\_customers;

alter table In\_Customers

add table\_id number(2) constraint FK\_icust references Tables (table\_id);



**TO\_GO\_CUSTOMERS**

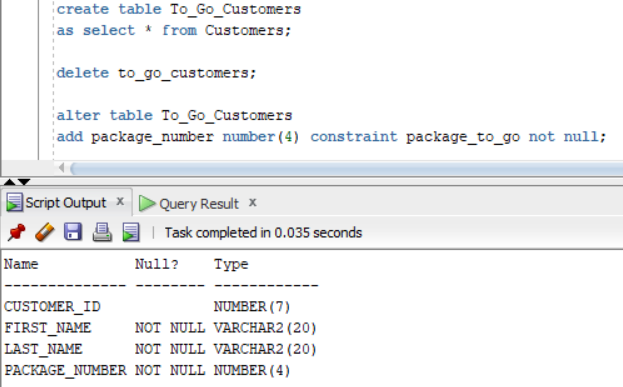
create table To\_Go\_Customers

as select \* from Customers;

delete to\_go\_customers;

alter table To\_Go\_Customers

add package\_number number(4) constraint package\_to\_go not null;



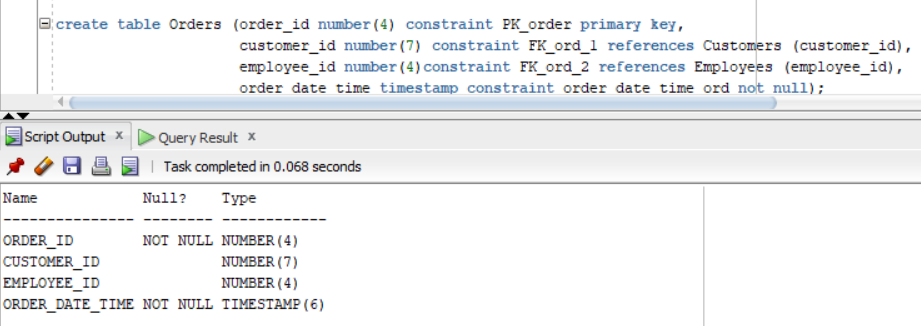
**ORDERS**

create table Orders (order\_id number(4) constraint PK\_order primary key,

customer\_id number(7) constraint FK\_ord\_1 references Customers (customer\_id),

employee\_id number(4)constraint FK\_ord\_2 references Employees (employee\_id),

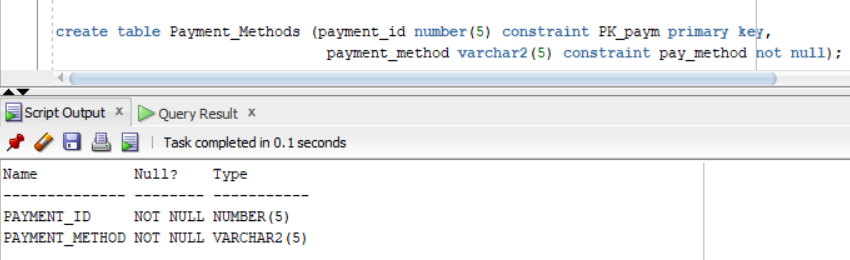
order\_date\_time timestamp constraint order\_date\_time\_ord not null);



**PAYMENT\_METHODS**

create table Payment\_Methods (payment\_id number(5) constraint PK\_paym primary key,

payment\_method varchar2(5) constraint pay\_method not null);



**RECEIPTS**

create sequence id\_generator

start with 4111

increment by 1

maxvalue 100000

nocycle

nocache;

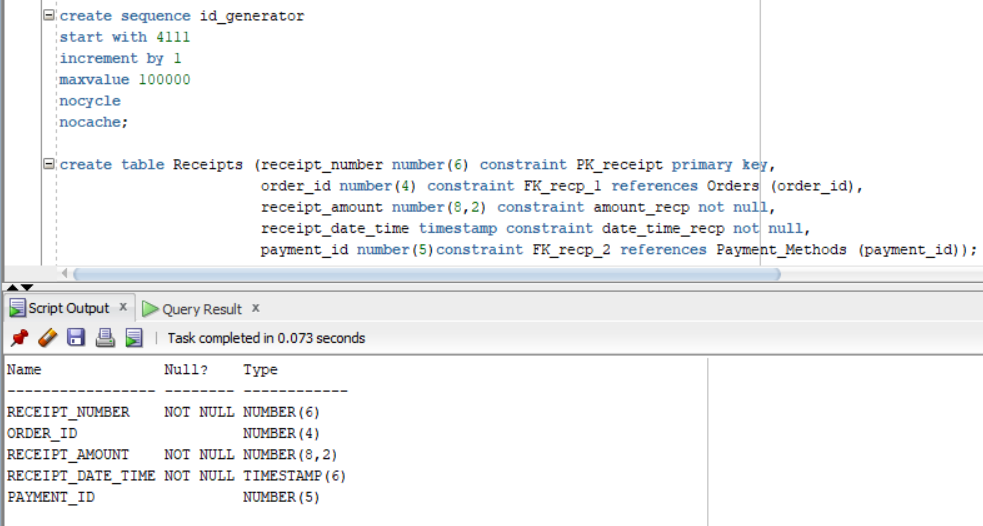
create table Receipts (receipt\_number number(6) constraint PK\_receipt primary key,

order\_id number(4) constraint FK\_recp\_1 references Orders (order\_id),

receipt\_amount number(8,2) constraint amount\_recp not null,

receipt\_date\_time timestamp constraint date\_time\_recp not null,

payment\_id number(5)constraint FK\_recp\_2 references Payment\_Methods (payment\_id));

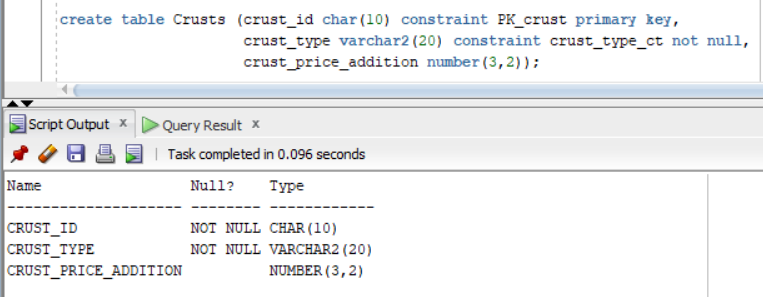


**CRUSTS**

create table Crusts (crust\_id char(10) constraint PK\_crust primary key,

crust\_type varchar2(20) constraint crust\_type\_ct not null,

crust\_price\_addition number(3,2));

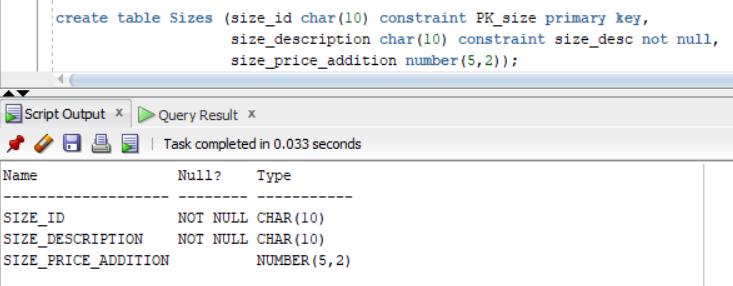


**SIZES**

create table Sizes (size\_id char(10) constraint PK\_size primary key,

size\_description char(10) constraint size\_desc not null,

size\_price\_addition number(5,2));

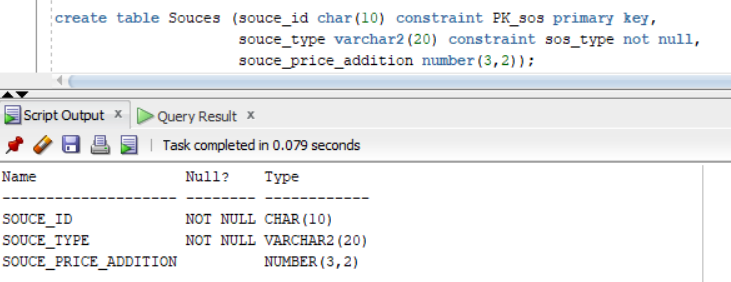


**SOUCES**

create table Souces (souce\_id char(10) constraint PK\_sos primary key,

souce\_type varchar2(20) constraint sos\_type not null,

souce\_price\_addition number(3,2));



**PIZZA**

create table Pizza (pizza\_id char(10) constraint PK\_pizza primary key,

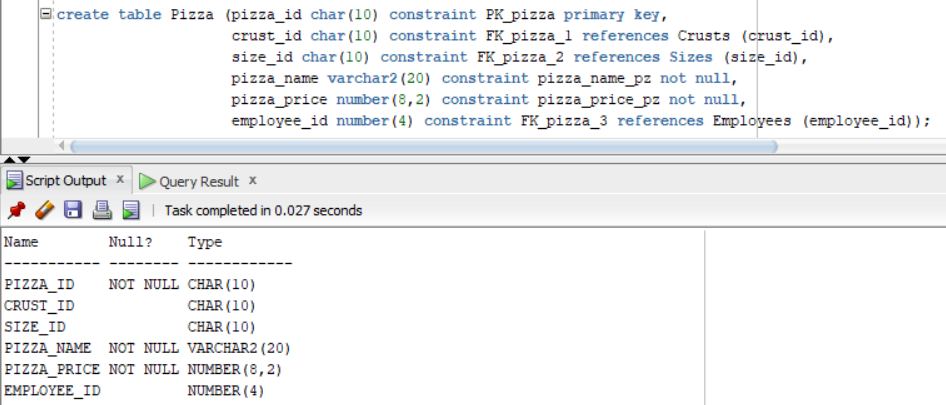
crust\_id char(10) constraint FK\_pizza\_1 references Crusts (crust\_id),

size\_id char(10) constraint FK\_pizza\_2 references Sizes (size\_id),

pizza\_name varchar2(20) constraint pizza\_name\_pz not null,

pizza\_price number(8,2) constraint pizza\_price\_pz not null,

employee\_id number(4) constraint FK\_pizza\_3 references Employees (employee\_id));



**SE\_SERVESTE\_CU**

create table Se\_serveste\_cu (pizza\_id char(10) constraint PK\_FK\_serv\_pizza references Pizza (pizza\_id),

souce\_id char(10) constraint PK\_FK\_serv\_sos references Souces (souce\_id));

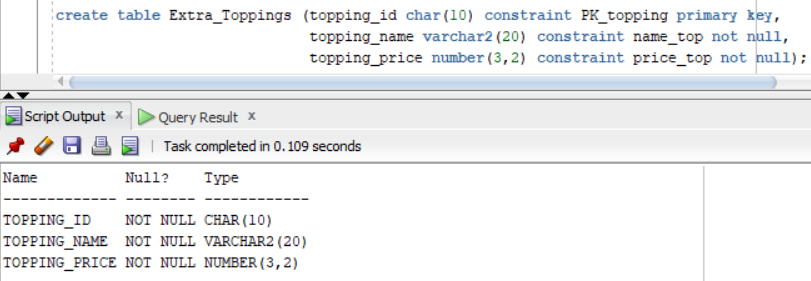


**EXTRA\_TOPPINGS**

create table Extra\_Toppings (topping\_id char(10) constraint PK\_topping primary key,

topping\_name varchar2(20) constraint name\_top not null,

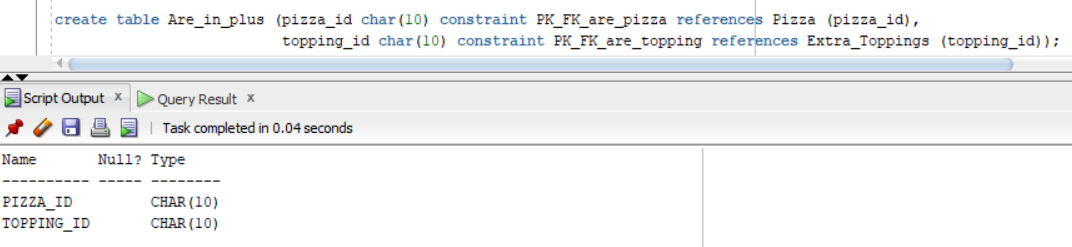
topping\_price number(3,2) constraint price\_top not null);



**ARE\_IN\_PLUS**

create table Are\_in\_plus (pizza\_id char(10) constraint PK\_FK\_are\_pizza references Pizza (pizza\_id),

topping\_id char(10) constraint PK\_FK\_are\_topping references Extra\_Toppings (topping\_id));

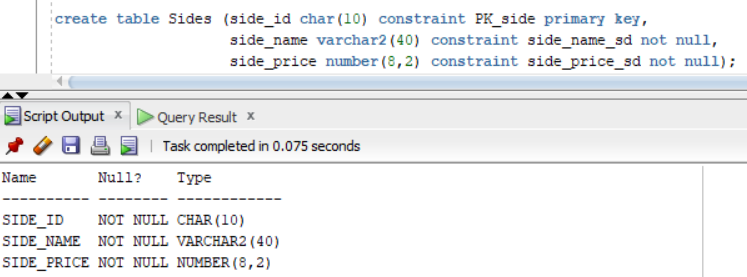


**SIDES**

create table Sides (side\_id char(10) constraint PK\_side primary key,

side\_name varchar2(40) constraint side\_name\_sd not null,

side\_price number(8,2) constraint side\_price\_sd not null);

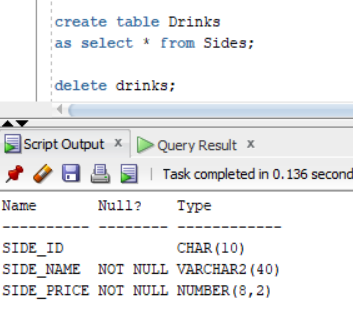


**DRINKS**

create table Drinks

as select \* from Sides;

delete drinks;

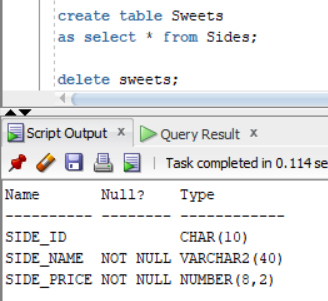


**SWEETS**

create table Sweets

as select \* from Sides;

delete sweets;



**CONTINE**

drop sequence id\_generator;

create sequence id\_generator

start with 301

increment by 1

maxvalue 1000

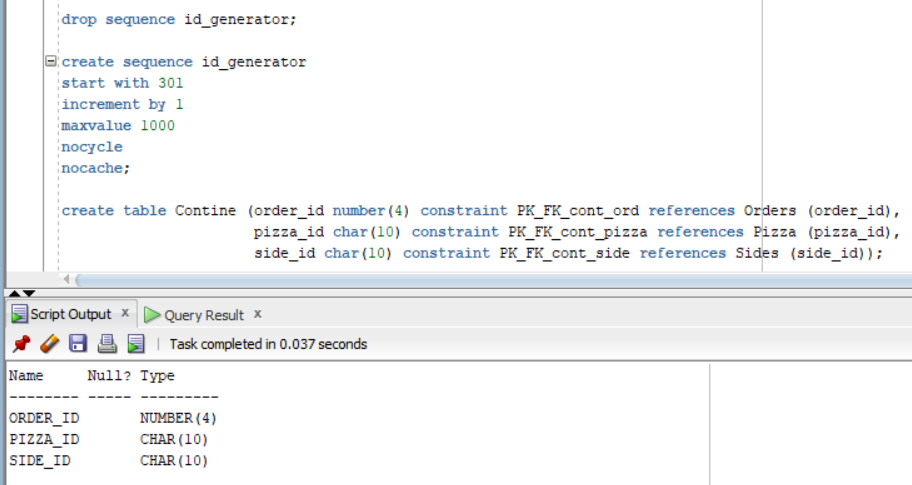
nocycle

nocache;

create table Contine (order\_id number(4) constraint PK\_FK\_cont\_ord references Orders (order\_id),

pizza\_id char(10) constraint PK\_FK\_cont\_pizza references Pizza (pizza\_id),

side\_id char(10) constraint PK\_FK\_cont\_side references Sides (side\_id));



Ex5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

**JOBS**

insert into jobs values('MG', 'Manager', 70000, 85000);

insert into jobs values('CT', 'Contabil', 20000, 30000);

insert into jobs values('CH', 'Bucatar', 65000, 70000);

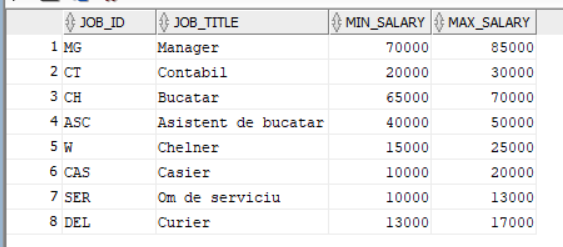
insert into jobs values('ASC', 'Asistent de bucatar', 40000, 50000);

insert into jobs values('W', 'Chelner', 15000, 25000);

insert into jobs values('CAS', 'Casier', 10000, 20000);

insert into jobs values('SER', 'Om de serviciu', 10000, 13000);

insert into jobs values('DEL', 'Curier', 13000, 17000);



**LOCATIONS**

insert into locations values(1, 'Ploiesti', 'Gornistilor', 10455, 130, 6);

insert into locations values(2, 'Ploiesti', 'Zamora', 10305, 43, 2);

insert into locations values(3, 'Ploiesti', 'Beius', 10080, 13,null);

insert into locations values(4, 'Ploiesti', 'Grindului', 10045, 56, 9);

insert into locations values(5, 'Ploiesti', 'Porumbitei', 10235, 160, 4);

insert into locations values(6, 'Ploiesti', 'Vi?inilor', 10897, 29, 14);

insert into locations values(7, 'Ploiesti', 'Regimentului', 10009, 49, null);

insert into locations values(8, 'Ploiesti', '8 Martie', 10570, 68, null);

insert into locations values(9, 'Ploiesti', 'Albinei', 10056, 98, 10);

insert into locations values(10, 'Ploiesti', 'Cameliei', 10017, 142, null);

insert into locations values(11, 'Ploiesti', 'Dumbravei', 10115, 124, 5);

insert into locations values(12, 'Ploiesti', 'Fluturilor', 10408, 119, 107);

insert into locations values(13, 'Ploiesti', 'Gheorghe Doja', 10023, 52, 1);

insert into locations values(14, 'Ploiesti', 'Furnica', 10222, 407, 3);

insert into locations values(15, 'Ploiesti', 'Ion Heliade', 10670, 35, null);

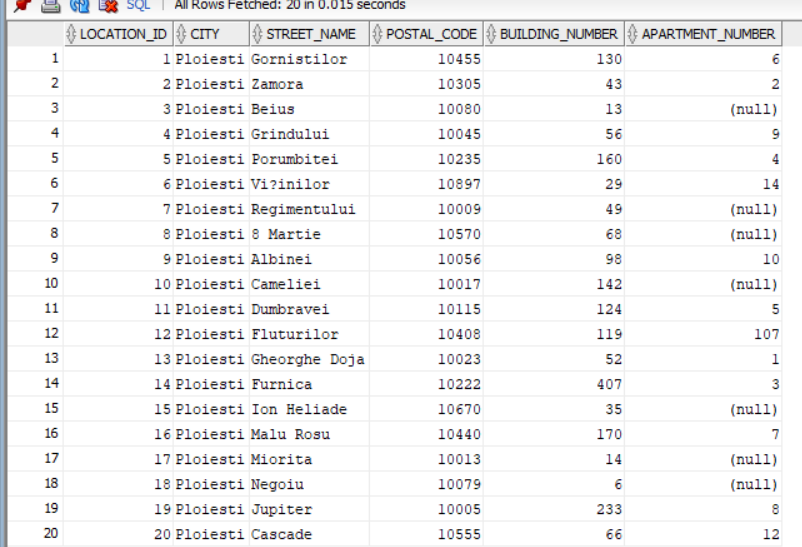
insert into locations values(16, 'Ploiesti', 'Malu Rosu', 10440, 170, 7);

insert into locations values(17, 'Ploiesti', 'Miorita', 10013, 14, null);

insert into locations values(18, 'Ploiesti', 'Negoiu', 10079, 6, null);

insert into locations values(19, 'Ploiesti', 'Jupiter', 10005, 233, 8);

insert into locations values(20, 'Ploiesti', 'Cascade', 10555, 66, 12);



**EMPLOYEES**

insert into employees values(101, 'Evelyn', 'Harper', 0785908519,'evelynH@gmail.com', 8, '06-NOV-00', 'MG', 75000);

insert into employees values(102, 'Benjamin', 'Garcia', 0770874084,'benjG@gmail.com', 12, '16-SEP-17', 'W', 25000);

insert into employees values(103, 'Ava', 'Brown', 0709472439,'avaB@gmail.com', 4, '04-MAR\_20', 'W', 15000);

insert into employees values(104, 'Isabella', 'Smith', 0774080650,'isabellaS@gmail.com', 14, '25-NOV-96', 'CH', 70000);

insert into employees values(105, 'Amelia', 'Hill', 0727314772,'ameliaH@gmail.com', 18, '27-DEC-10', 'DEL', 16000);

insert into employees values(106, 'Sophia', 'Wlliams', 0760803376,'sophiaW@gmail.com', 9, '24-JUL-15', 'W', 18000);

insert into employees values(107, 'Charlotte', 'Moore', 0767231470,'chrM@gmail.com', 15, '28-OCT\_04', 'SER', 11000);

insert into employees values(108, 'Emma', 'Donovan', 0746107617,'emmaD@gmail.com', 17, '13-NOV-15', 'ASC', 45000);

insert into employees values(109, 'Oliver', 'Carter', 0781361168,'oliverC@gmail.com', 2, '11\_MAR-11', 'CT', 30000);

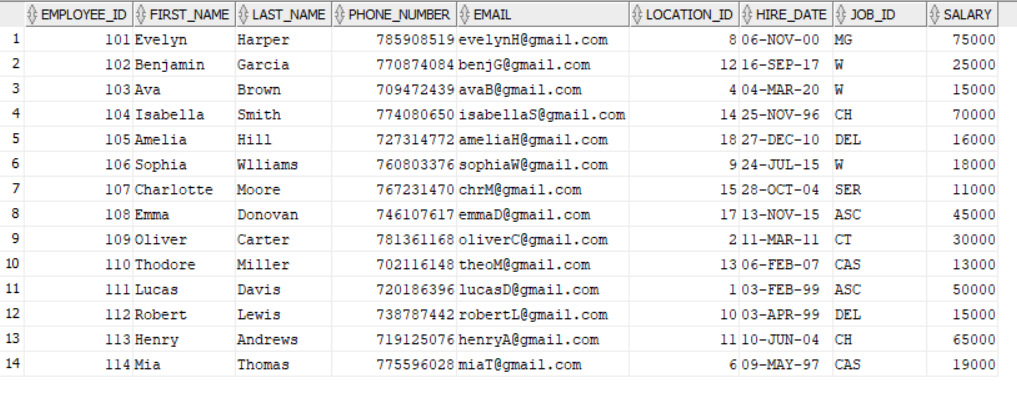
insert into employees values(110, 'Thodore', 'Miller', 0702116148,'theoM@gmail.com', 13, '06-FEB-07', 'CAS', 13000);

insert into employees values(111, 'Lucas', 'Davis', 0720186396,'lucasD@gmail.com', 1, '03-FEB-99', 'ASC', 50000);

insert into employees values(112, 'Robert', 'Lewis', 0738787442,'robertL@gmail.com', 10, '03-APR-99', 'DEL', 15000);

insert into employees values(113, 'Henry', 'Andrews', 0719125076,'henryA@gmail.com', 11, '10-JUN-04', 'CH', 65000);

insert into employees values(114, 'Mia', 'Thomas', 0775596028,'miaT@gmail.com', 6, '09-MAY-97', 'CAS', 19000);



**DELIVERIES**

insert into deliveries values(21, 112, 31);

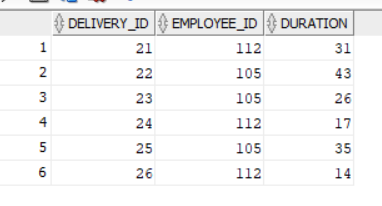
insert into deliveries values(22, 105, 43);

insert into deliveries values(23, 105, 26);

insert into deliveries values(24, 112, 17);

insert into deliveries values(25, 105, 35);

insert into deliveries values(26, 112, 14);



**TABLES**

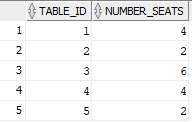
insert into tables values(1, 4);

insert into tables values(2, 2);

insert into tables values(3, 6);

insert into tables values(4, 4);

insert into tables values(5, 2);



**CUSTOMERS**

insert into Customers values(1121, 'Ronald', 'Macias');

insert into Customers values(1122, 'Caoimhe', 'Rosario');

insert into Customers values(1123, 'Harry', 'Dunne');

insert into Customers values(1124, 'Chanice', 'Rollins');

insert into Customers values(1125, 'Jayden-Lee', 'Davidson');

insert into Customers values(1126, 'Kinga', 'Betts');

insert into Customers values(1127, 'Catriona', 'Mclean');

insert into Customers values(1128, 'Bradleig', 'Parrish');

insert into Customers values(1129, 'Callum', 'Clarkson');

insert into Customers values(1130, 'Jaxson', 'Leach');

insert into Customers values(1131, 'Jordanne', 'Corona');

insert into Customers values(1132, 'Fintan', 'Adkins');

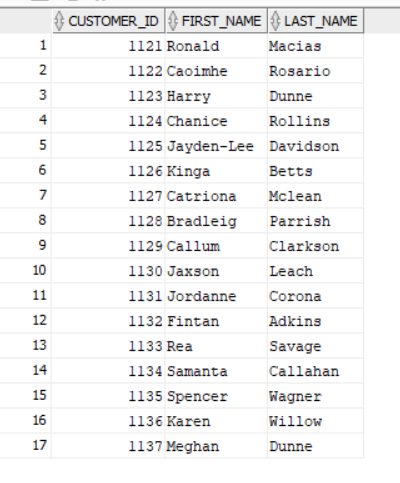
insert into Customers values(1133, 'Rea', 'Savage');

insert into Customers values(1134, 'Samanta', 'Callahan');

insert into Customers values(1135, 'Spencer', 'Wagner');

insert into Customers values(1136, 'Karen', 'Willow');

insert into Customers values(1137, 'Meghan', 'Dunne');



**ONLINE\_CUSTOMERS**

insert into Online\_Customers values(1127, 'Catriona', 'Mclean', 16, 0795051829, 'catrionaM@gmail.com', 24);

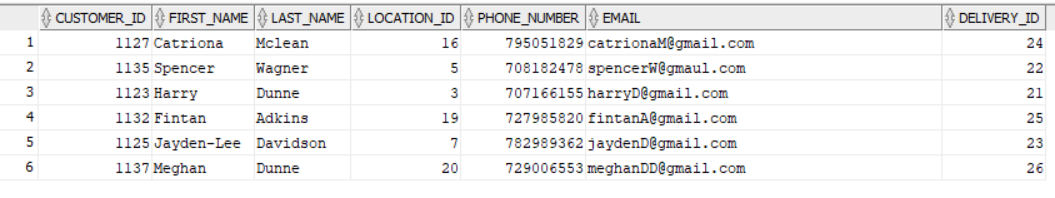
insert into Online\_Customers values(1135, 'Spencer', 'Wagner', 5, 0708182478, 'spencerW@gmaul.com', 22);

insert into Online\_Customers values(1123, 'Harry', 'Dunne', 3, 0707166155, 'harryD@gmail.com', 21);

insert into Online\_Customers values(1132, 'Fintan', 'Adkins', 19, 0727985820, 'fintanA@gmail.com', 25);

insert into Online\_Customers values(1125, 'Jayden-Lee', 'Davidson', 7, 0782989362, 'jaydenD@gmail.com', 23);

insert into Online\_Customers values(1137, 'Meghan', 'Dunne', 20, 0729006553, 'meghanDD@gmail.com', 26)



**IN\_CUSTOMERS**

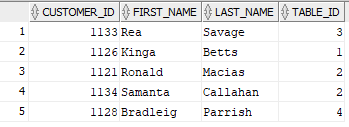
insert into In\_Customers values(1133, 'Rea', 'Savage', 3);

insert into In\_Customers values(1126, 'Kinga', 'Betts', 1);

insert into In\_Customers values(1121, 'Ronald', 'Macias', 2);

insert into In\_Customers values(1134, 'Samanta', 'Callahan', 2);

insert into In\_Customers values(1128, 'Bradleig', 'Parrish', 4);



**TO\_GO\_CUSTOMERS**

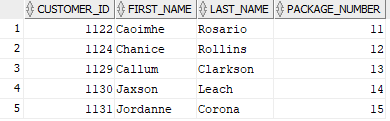
insert into To\_Go\_Customers values(1122, 'Caoimhe', 'Rosario', 11);

insert into To\_Go\_Customers values(1124, 'Chanice', 'Rollins', 12);

insert into To\_Go\_Customers values(1129, 'Callum', 'Clarkson', 13);

insert into To\_Go\_Customers values(1130, 'Jaxson', 'Leach', 14);

insert into To\_Go\_Customers values(1131, 'Jordanne', 'Corona', 15);



**ORDERS**

insert into orders values(301, 1121, 102, '3-FEB-20 10:30:15');

insert into orders values(302, 1122, 103, '18\_FEB-20 10:45:07');

insert into orders values(303, 1123, 106, '12\_APR\_20 11:30:29');

insert into orders values(304, 1124, 102, '2-MAY-20 12:30:36');

insert into orders values(305, 1125, 103, '8-MAY-20 12:45:18');

insert into orders values(306, 1126, 106, '25-JUN-20 13:15:20');

insert into orders values(307, 1127, 102, '23-AUG-20 14:30:42');

insert into orders values(308, 1128, 103, '2-SEP-20 15:32:09');

insert into orders values(309, 1129, 106, '16-SEP-20 16:45:57');

insert into orders values(310, 1130, 102, '1-FEB-21 17:00:18');

insert into orders values(311, 1131, 103, '11-FEB-21 17:45:02')

insert into orders values(312, 1132, 106, '13-APR-21 18:00:11');

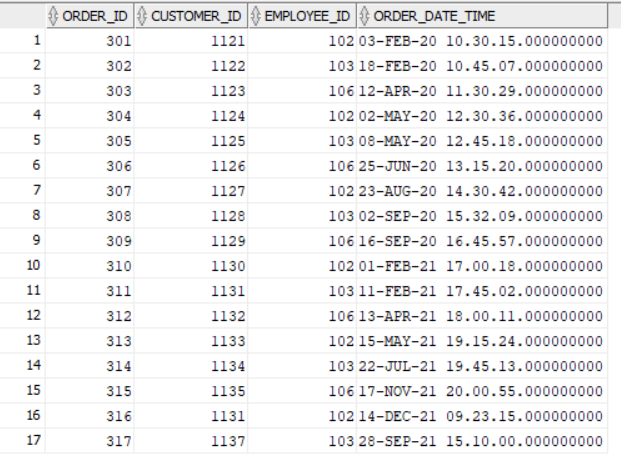
insert into orders values(313, 1133, 102, '15-MAY-21 19:15:24');

insert into orders values(314, 1134, 103, '22-JUL-21 19:45:13');

insert into orders values(315, 1135, 106, '17-NOV-21 20:00:55');

insert into orders values(316, 1131, 102, '14-DEC-21 9:23:15');

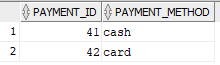
insert into orders values(317, 1137, 103, '28-SEP-21 15:10:00');



**PAYMENT\_METHODS**

insert into payment\_methods values(41, 'cash');

insert into payment\_methods values(42, 'card');



**RECEIPTS**

insert into receipts values(id\_generator.nextval, 301, 28.55, '3-FEB-20 12:05:23', 41);

insert into receipts values(id\_generator.nextval, 302, 93.4, '18\_FEB-20 11:10:42', 42);

insert into receipts values(id\_generator.nextval, 303, 52.05, '12\_APR\_20 12:08:12', 42);

insert into receipts values(id\_generator.nextval, 304, 47.45, '2-MAY-20 12:55:04', 41);

insert into receipts values(id\_generator.nextval, 305, 42.95, '8-MAY-20 13:32:36', 41);

insert into receipts values(id\_generator.nextval, 306, 58.45, '25-JUN-20 15:37:15', 42);

insert into receipts values(id\_generator.nextval, 307, 46, '23-AUG-20 15:14:08', 42);

insert into receipts values(id\_generator.nextval, 308, 12.95, '2-SEP-20 16:51:23', 41);

insert into receipts values(id\_generator.nextval, 309, 24.75, '16-SEP-20 17:12:38', 41);

insert into receipts values(id\_generator.nextval, 310, 6.5, '1-FEB-21 17:06:32', 41);

insert into receipts values(id\_generator.nextval, 311, 26.65, '11-FEB-21 16:01:14 ', 42);

insert into receipts values(id\_generator.nextval, 312, 70.2, '13-APR-21 18:42:56', 42);

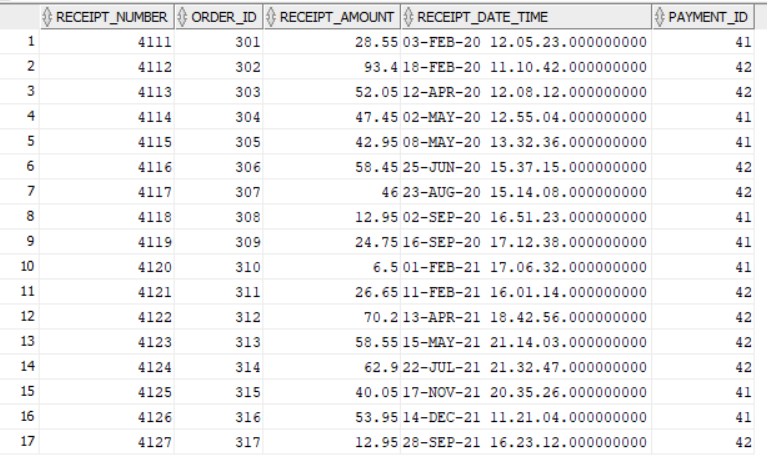
insert into receipts values(id\_generator.nextval, 313, 58.55, '15-MAY-21 21:14:03', 42);

insert into receipts values(id\_generator.nextval, 314, 62.9, '22-JUL-21 21:32:47', 42);

insert into receipts values(id\_generator.nextval, 315, 40.05, '17-NOV-21 20:35:26', 41);

insert into receipts values(id\_generator.nextval, 316, 53.95, '14-DEC-21 11:21:04 ', 41);

insert into receipts values(id\_generator.nextval, 317, 12.95, '28-SEP-21 16:23:12', 42);



**CRUSTS**

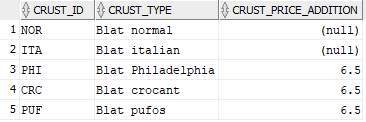
insert into crusts values('NOR', 'Blat normal', null);

insert into crusts values('ITA', 'Blat italian', null);

insert into crusts values('PHI', 'Blat Philadelphia', 6.5);

insert into crusts values('CRC', 'Blat crocant', 6.5);

insert into crusts values('PUF', 'Blat pufos', 6.5);

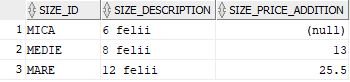


**SIZES**

insert into sizes values('MICA', '6 felii', null);

insert into sizes values('MEDIE', '8 felii', 13);

insert into sizes values('MARE', '12 felii', 25.5);



**SOUCES**

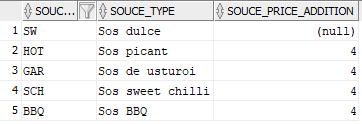
insert into souces values('SW', 'Sos dulce', null);

insert into souces values('HOT', 'Sos picant', 4);

insert into souces values('GAR', 'Sos de usturoi', 4);

insert into souces values('SCH', 'Sos sweet chilli', 4);

insert into souces values('BBQ', 'Sos BBQ', 4);



**PIZZA**

insert into pizza values('QFNMi', 'NOR', 'MICA', 'Quattro Formaggi', 28.55, 104);

insert into pizza values('PIMe', 'ITA', 'MEDIE', 'Pepperoni', 42.95, 108);

insert into pizza values('CaPMe', 'PUF', 'MEDIE', 'Carnivora', 52.05, 111);

insert into pizza values('DNMa', 'NOR', 'MARE', 'Diavola', 47.45, 113);

insert into pizza values('CPMe', 'PHI', 'MEDIE', 'Capriciosa', 45.45, 104);

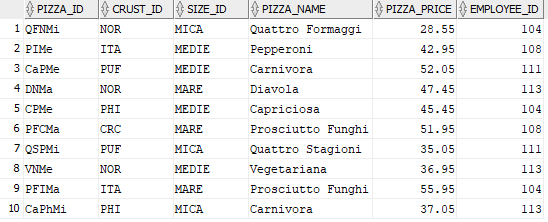
insert into pizza values('PFCMa', 'CRC', 'MARE', 'Prosciutto Funghi', 51.95, 108);

insert into pizza values('QSPMi', 'PUF', 'MICA', 'Quattro Stagioni', 35.05, 111);

insert into pizza values('VNMe', 'NOR', 'MEDIE', 'Vegetariana', 36.95, 113);

insert into pizza values('PFIMa', 'ITA', 'MARE', 'Prosciutto Funghi', 55.95, 104);

insert into pizza values('CaPhMi', 'PHI', 'MICA', 'Carnivora', 37.05, 113);



**SE\_SERVESTE\_CU**

insert into se\_serveste\_cu values('QSPMi','GAR');

insert into se\_serveste\_cu values('QFNMi','SW');

insert into se\_serveste\_cu values('CPMe','HOT');

insert into se\_serveste\_cu values('CaPMe','GAR');

insert into se\_serveste\_cu values('PFCMa','SCH');

insert into se\_serveste\_cu values('QSPMi','SW');

insert into se\_serveste\_cu values('PFCMa','HOT');

insert into se\_serveste\_cu values('CPMe','BBQ');

insert into se\_serveste\_cu values('CaPMe','SW');



**EXTRA\_TOPPINGS**

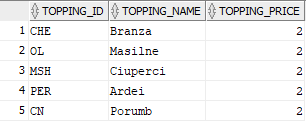
insert into extra\_toppings values('CHE', 'Branza', 2);

insert into extra\_toppings values('OL', 'Masilne', 2);

insert into extra\_toppings values('MSH', 'Ciuperci', 2);

insert into extra\_toppings values('PER', 'Ardei', 2);

insert into extra\_toppings values('CN', 'Porumb', 2);



**ARE\_IN\_PLUS**

insert into are\_in\_plus values('PIMe','CHE');

insert into are\_in\_plus values('DNMa','OL');

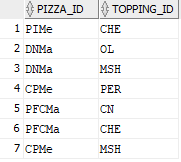
insert into are\_in\_plus values('DNMa','MSH');

insert into are\_in\_plus values('CPMe','PER');

insert into are\_in\_plus values('PFCMa','CN');

insert into are\_in\_plus values('PFCMa','CHE');

insert into are\_in\_plus values('CPMe','MSH');



**SIDES**

insert into sides values('AP', 'Apa Plata', 5);

insert into sides values('CP', 'Choco Pizza', 26.65);

insert into sides values('FN', 'Fanta', 6.5);

insert into sides values('SC', 'Soufle de ciocolata', 10.95);

insert into sides values('CC', 'Coca Cola', 6.5);

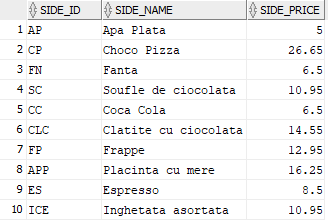
insert into sides values('CLC', 'Clatite cu ciocolata', 14.55);

insert into sides values('FP', 'Frappe', 12.95);

insert into sides values('APP', 'Placinta cu mere', 16.25);

insert into sides values('ES', 'Espresso', 8.5);

insert into sides values('ICE', 'Inghetata asortata', 10.95);



**DRINKS**

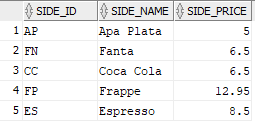
insert into drinks values('AP', 'Apa Plata', 5);

insert into drinks values('FN', 'Fanta', 6.5);

insert into drinks values('CC', 'Coca Cola', 6.5);

insert into drinks values('FP', 'Frappe', 12.95);

insert into drinks values('ES', 'Espresso', 8.5);



**SWEETS**

insert into sweets values('CP', 'Choco Pizza', 26.65);

insert into sweets values('SC', 'Soufle de ciocolata', 10.95);

insert into sweets values('CLC', 'Clatite cu ciocolata', 14.55);

insert into sweets values('APP', 'Placinta cu mere', 16.25);

insert into sweets values('ICE', 'Inghetata asortata', 10.95);



**CONTINE**

insert into contine values(id\_generator.nextval , 'QFNMi', null);

insert into contine values(id\_generator.nextval , 'PIMe', null);

insert into contine values(id\_generator.nextval , 'CaPMe', null);

insert into contine values(id\_generator.nextval , 'DNMa', null);

insert into contine values(id\_generator.nextval , 'PIMe', null);

insert into contine values(id\_generator.nextval , 'PFCMa','CC');

insert into contine values(id\_generator.nextval , 'QSPMi','SC');

insert into contine values(id\_generator.nextval , null ,'FP');

insert into contine values(id\_generator.nextval , null ,'ES');

insert into contine values(id\_generator.nextval , null ,'FN');

insert into contine values(id\_generator.nextval , null ,'CP');

insert into contine values(id\_generator.nextval , 'DNMa' ,'APP');

insert into contine values(id\_generator.nextval , 'CaPMe' ,'FN');

insert into contine values(id\_generator.nextval , 'PFCMa','ICE');

insert into contine values(id\_generator.nextval , 'QSPMi' ,'AP');

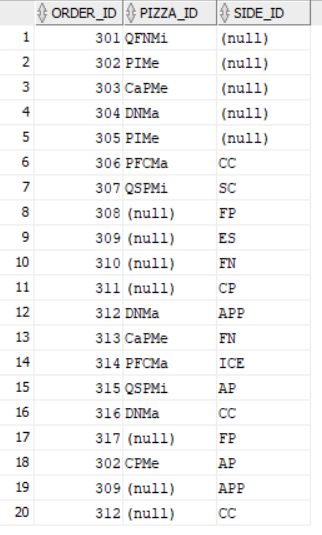
insert into contine values(id\_generator.nextval , 'DNMa' ,'CC');

insert into contine values(id\_generator.nextval , null ,'FP');

insert into contine values(302 , 'CPMe', 'AP');

insert into contine values(309 , null ,'APP');

insert into contine values(312 , null ,'CC');



Ex6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze două tipuri diferite de colecții studiate. Apelați subprogramul.

Pentru fiecare comanda sa se afiseze numele si tipul de blat ai fiecarei pizze sau sa se afiseze "Nu contine pizza" cand comanda nu contine pizza.

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE ex6

IS

--in t1 retin toate comenzile

TYPE t\_imbricat IS TABLE OF orders.order\_id%type;

t1 t\_imbricat;

--in t2 retin toate pizzele din fiecare comanda din t1

TYPE vector IS VARRAY(10) OF pizza.pizza\_id%type;

t2 vector:= vector();

i NUMBER(2):=0;

j NUMBER(2):=0;

nr NUMBER(2):=0;

blat crusts.crust\_type%type;

nume pizza.pizza\_name%type;

BEGIN

SELECT order\_id

BULK COLLECT INTO t1

FROM orders;

FOR i in t1.FIRST..t1.LAST LOOP

DBMS\_OUTPUT.PUT\_LINE('Comanda '|| t1(i)||': ');

SELECT pizza\_id

BULK COLLECT INTO t2

FROM contine

WHERE order\_id = t1(i);

nr:=0;

FOR j in t2.FIRST..t2.LAST LOOP

IF t2(j) IS NOT NULL THEN

SELECT p.pizza\_name, c.crust\_type

INTO nume, blat

FROM crusts c, pizza p

WHERE c.crust\_id = p.crust\_id

AND p.pizza\_id = t2(j);

DBMS\_OUTPUT.PUT\_LINE('Pizza '|| nume||' cu '|| blat);

nr:= nr+1;

END IF;

END LOOP;

IF nr = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Nu contine pizza');

END IF;

t2.delete;

DBMS\_OUTPUT.PUT\_LINE(' ');

END LOOP;

END ex6;

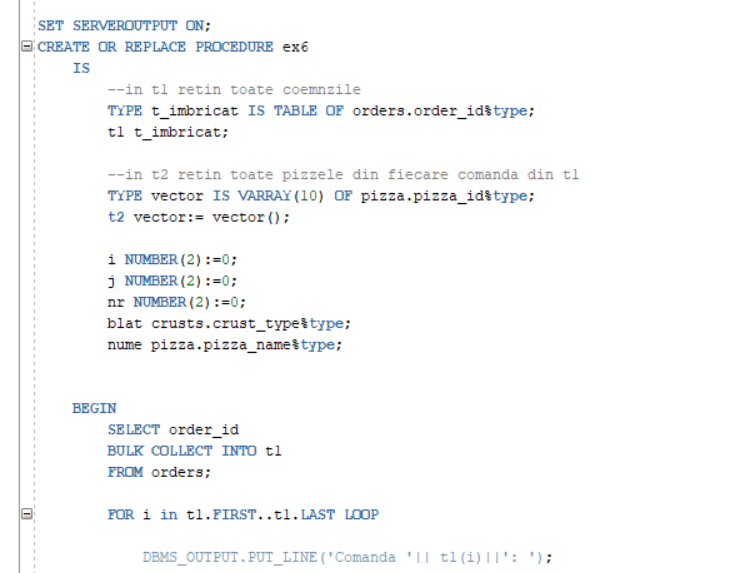
/

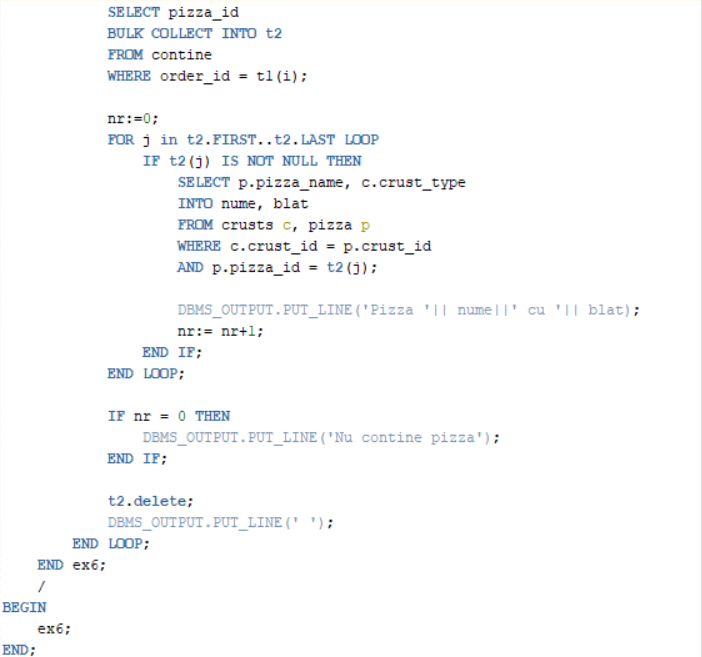
BEGIN

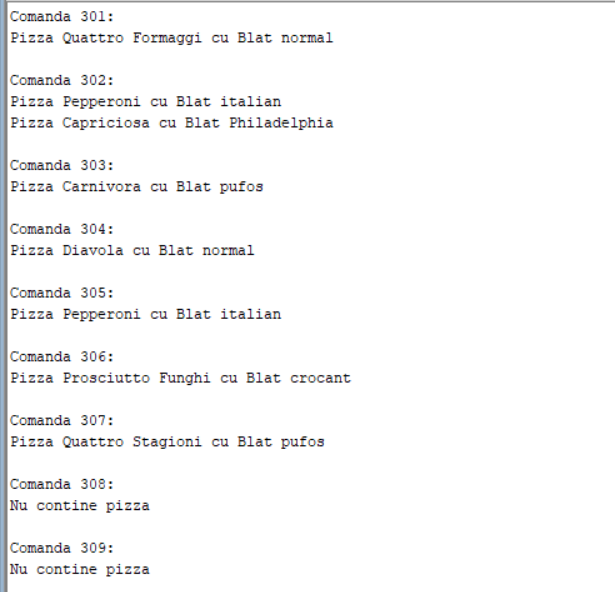
ex6;

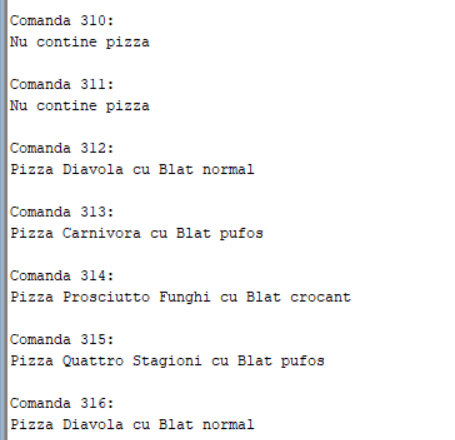
END;

/









Ex7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat. Apelați subprogramul.

Sa se afiseze numele, prenumele si numele jobului angajatilor care au salariul minim mai mare decat o suma data ca parametru si care au fost angajati dupa anul 2000.

CREATE OR REPLACE PROCEDURE ex7 (salariu jobs.min\_salary%type)

IS

--in c1 retin id-ul si numele joburilor cu salariul minim mai mare decat salariul dat ca parametru

CURSOR c1 IS

SELECT job\_id, job\_title

FROM jobs

WHERE min\_salary > salariu;

--in c2 retin numele, prenumele si data angajarii angajatilor pentru fiecare job din c1

CURSOR c2 (job jobs.job\_id%type) IS

SELECT last\_name, first\_name, hire\_date

FROM employees

WHERE job\_id = job;

v\_nume employees.last\_name%type;

v\_prenume employees.first\_name%type;

v\_data employees.hire\_date%type;

BEGIN

FOR i in c1 LOOP

open c2(i.job\_id);

LOOP

FETCH c2 INTO v\_nume, v\_prenume, v\_data;

EXIT WHEN c2%NOTFOUND;

IF to\_char(v\_data, 'YYYY')>=2000 THEN

DBMS\_OUTPUT.PUT\_LINE(i.job\_title || ' ' || v\_nume || ' ' || v\_prenume);

END IF;

END LOOP;

CLOSE c2;

END LOOP;

END ex7;

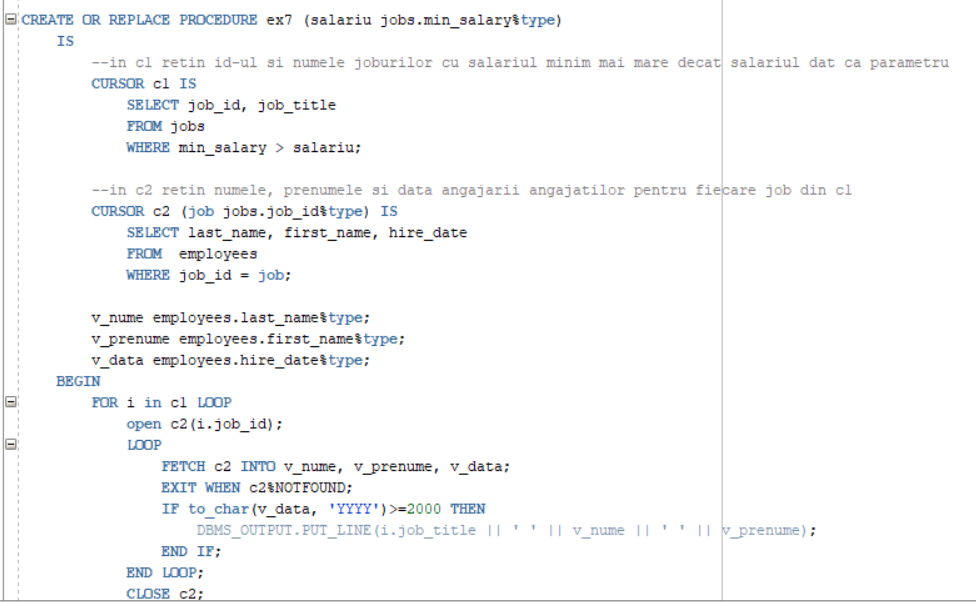
/

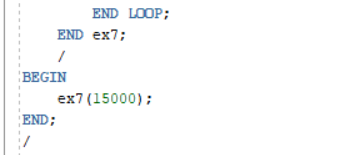
BEGIN

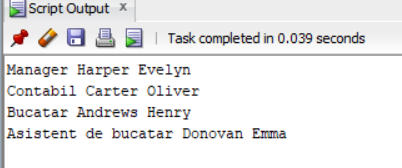
ex7(15000);

END;

/







Ex8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Fiind dat ca parametru numele unui client, sa se afiseze suma totala a comenzilor plasate de acesta. Sa se trateze cazurile in care exista mai multi clienti cu acelasi nume si daca nu exista clienti cu numele dat.

CREATE OR REPLACE FUNCTION ex8(v\_nume customers.last\_name%type)

RETURN NUMBER IS rezultat NUMBER(10,2);

contor NUMBER;

BEGIN

--tratare exceptii

SELECT COUNT(\*)

INTO contor

FROM customers

WHERE last\_name = v\_nume;

IF contor = 0 THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista clienti cu numele dat');

ELSIF contor>1 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Exista mai multi clienti cu numele dat');

END IF;

SELECT SUM(r.receipt\_amount)

INTO rezultat

FROM customers c

JOIN orders o ON c.customer\_id = o.customer\_id

JOIN receipts r ON o.order\_id = r.order\_id

AND c.last\_name = v\_nume;

RETURN rezultat;

END ex8;

/

BEGIN

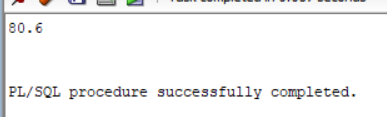
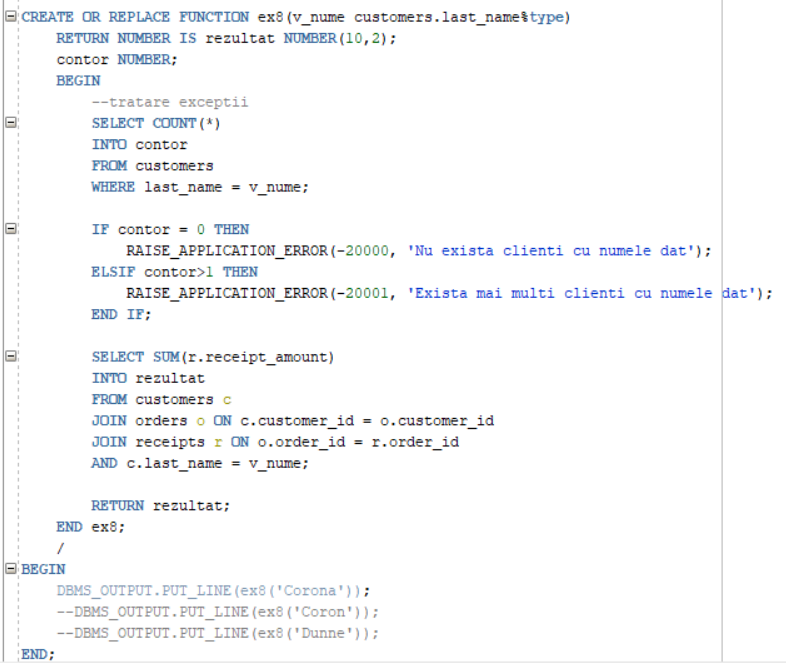
DBMS\_OUTPUT.PUT\_LINE(ex8('Corona'));

--DBMS\_OUTPUT.PUT\_LINE(ex8('Coron'));

--DBMS\_OUTPUT.PUT\_LINE(ex8('Dunne'));

END;

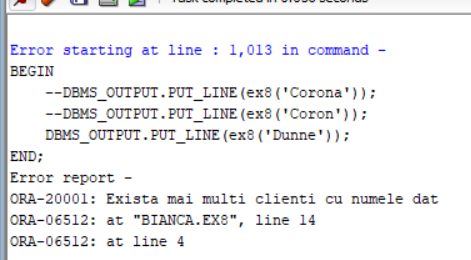
/



Prima exceptie:



A doua exceptie:



Ex9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Fiind dat ca parametru numele unui client, in cazul in care a plasat comanda online, sa se afiseze numele clientului, orasul, strada si metoda de plata. Sa se trateze cazurile in care nu exista un client cu acest nume, clientul respectiv nu a plasat o comanda online sau exista mai multi clienti cu acest nume care au plasat o comanda online.

CREATE OR REPLACE PROCEDURE ex9(nume IN OUT online\_customers.last\_name%type,

oras OUT locations.city%type,

strada OUT locations.street\_name%type,

plata OUT payment\_methods.payment\_method%type)

IS

contor NUMBER;

exceptie EXCEPTION;

BEGIN

SELECT COUNT(\*)

INTO contor

FROM customers

WHERE last\_name = nume;

IF contor = 0 THEN

RAISE exceptie;

END IF;

SELECT l.city, l.street\_name, p.payment\_method

INTO oras, strada, plata

FROM online\_customers c, locations l, orders o, receipts r, payment\_methods p

WHERE c.location\_id = l.location\_id

AND c.customer\_id = o.customer\_id

AND r.order\_id = o.order\_id

AND r.payment\_id = p.payment\_id

AND c.last\_name = nume;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista clienti care au plasat comanda online cu numele dat');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Exista mai multi clienti care au plasat comanda online cu numele dat');

WHEN exceptie THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Nu exista clienti cu numele dat');

END ex9;

/

DECLARE

v\_nume online\_customers.last\_name%type;

v\_oras locations.city%type;

v\_strada locations.street\_name%type;

v\_plata payment\_methods.payment\_method%type;

BEGIN

v\_nume:='Adkins';

--v\_nume:='Dunne';

--v\_nume:='Rad';

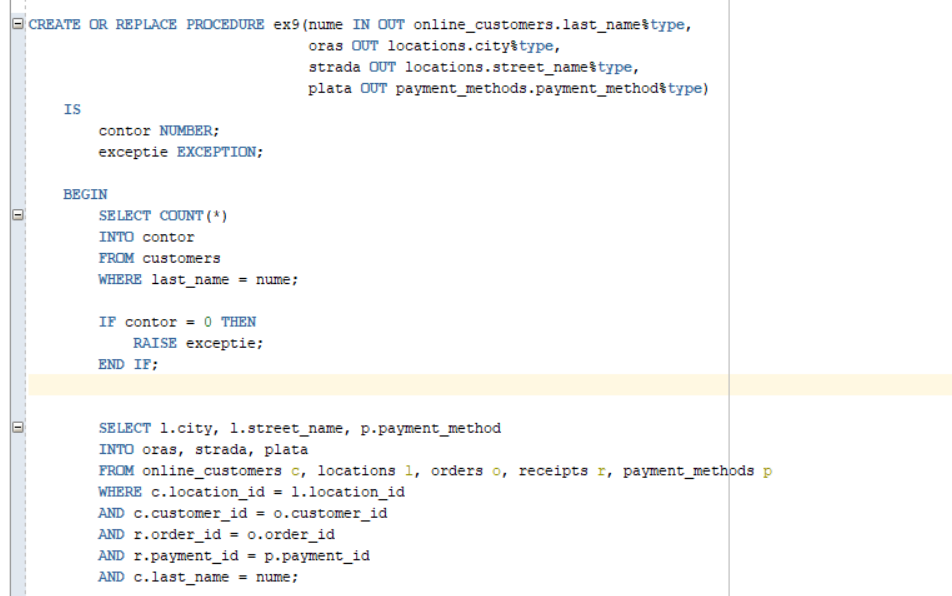
--v\_nume:='Macias';

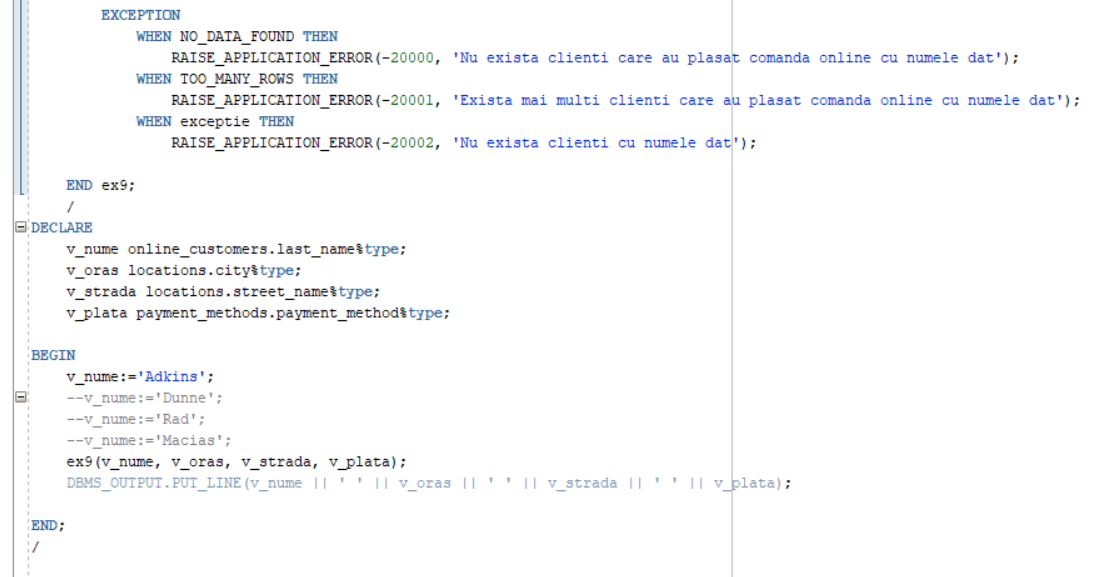
ex9(v\_nume, v\_oras, v\_strada, v\_plata);

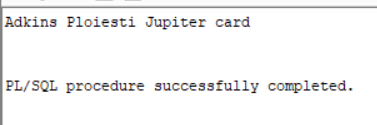
DBMS\_OUTPUT.PUT\_LINE(v\_nume || ' ' || v\_oras || ' ' || v\_strada || ' ' || v\_plata);

END;

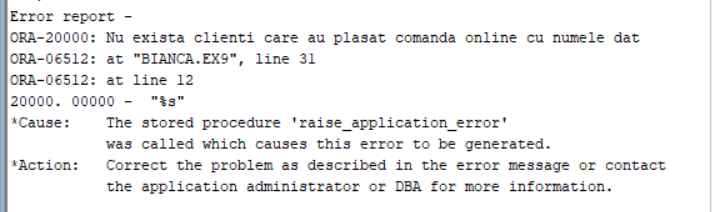
/



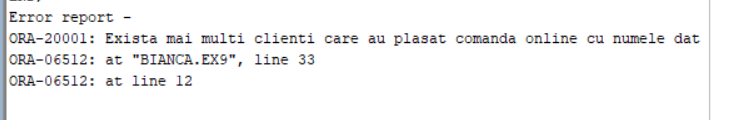




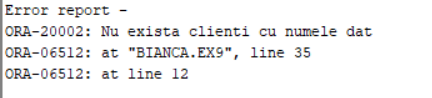
Prima exceptie:



A doua exceptie:



A treia exceptie:



Ex10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

Trigger care interzice stergerea datelor din tabelul JOBS.

CREATE OR REPLACE TRIGGER ex10

BEFORE DELETE ON jobs

BEGIN

RAISE\_APPLICATION\_ERROR(-20001, 'Este interzisa stergerea datelor din tabelul JOBS');

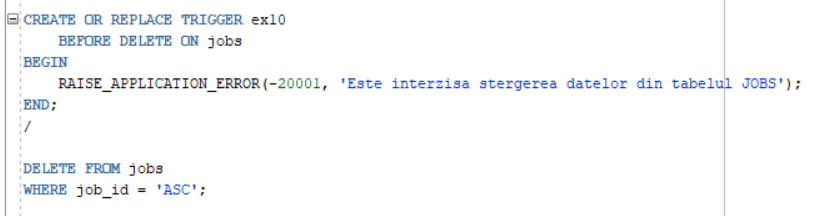
END;

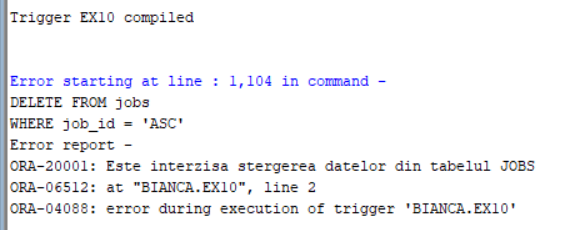
/

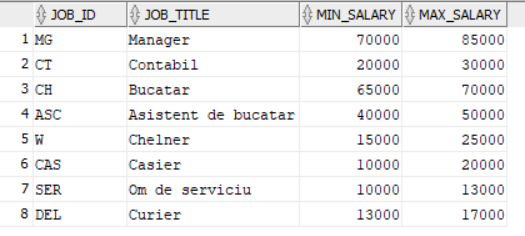
DELETE FROM jobs

WHERE job\_id = 'ASC';

select \* from jobs;







Ex11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

Trigger care interzice cresterea salariului unui angajat cu mai mult de 10%.

CREATE OR REPLACE TRIGGER ex11

BEFORE UPDATE ON employees

FOR EACH ROW

BEGIN

IF (:NEW.salary - :OLD.salary > :OLD.salary\*(10/100)) THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Salariul nu poate creste cu mai mult de 10%');

END IF;

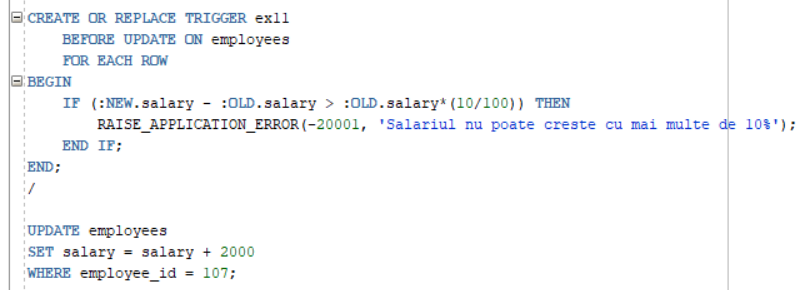
END;

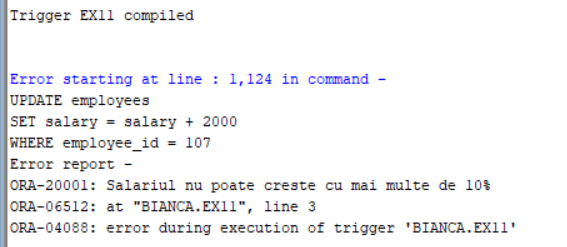
/

UPDATE employees

SET salary = salary + 2000

WHERE employee\_id = 107;





Ex12. Definiți un trigger de tip LDD. Declanșați trigger-ul.

Trigger care se declanseaza la fiecare executie de operatii LDD si salveaza modificarile in tabela userB.

CREATE TABLE userB(nume\_bd VARCHAR2(50),

user\_logat VARCHAR2(30),

eveniment VARCHAR2(20),

tip\_obiect\_referit VARCHAR2(30),

nume\_obiect\_referit VARCHAR2(30),

data TIMESTAMP(3));

CREATE OR REPLACE TRIGGER ex12

AFTER CREATE OR DROP OR ALTER ON SCHEMA

BEGIN

INSERT INTO userB

VALUES(SYS.DATABASE\_NAME, SYS.LOGIN\_USER,

SYS.SYSEVENT, SYS.DICTIONARY\_OBJ\_TYPE,

SYS.DICTIONARY\_OBJ\_NAME, SYSTIMESTAMP(3));

END;

/

ALTER TABLE locations

ADD country VARCHAR(40);

select \* from locations;

select \* from userB;

ALTER TABLE locations

DROP COLUMN country;

