Baza de date a unui lant de restaurante

Realizat de: Petchi Andrei Luca Grupa 234

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

Pentru acest proiect , am ales realizarea bazei de date a unui lant de restaurante cu specific japonez , format din „Japanos Fusion’’, „Japanos By The Lake ”, „ Japanos Tineretului ”, „ Japanos Takeaway ” si „Japanos Dorobanti”. Restaurantele functioneaza pe un sistem clasic, clientului ii este preluata comanda de catre un ospatar, urmand ca aceasta sa fie preparata de un bucatar. Acesta plateste la final nota, avand posibilitatea platii cash sau card. Clientii au posibilitatea de a face rezervari sau comenzi online. In cazul comenziilor online , comanda este livrata de catre un curier din cadrul restaurantului solicitat . Baza de date propusa faciliteaza managementul restaurantelor din retea, angajatii , clientii si comenziile fiind reprezentati de entitatile “ANGAJATI” , “CLIENTI” respectiv “COMANDA” tinand evidenta rolului fiecarui angajat, performanta lor si datele fiecarei comenzi. De asemenea , baza de date ofera informatii despre stocuri, vanzari cat si preferintele clientilor, ajuta la luarea deciziilor informate, optimizand costurile, cresterea eficientei operationale si imbunatatirea experientei clientilor.

Reguli

-Fiecare angajat lucreaza intr-o singura locatie, iar fiecare locatie la randul ei are cel putin un angajat pentru fiecare departament.

-Rezervarea unei mese este posibila cat si comandarea prin livrare.

-Angajatii sunt impartiti in :chefi,staff si curieri, fiecare angajat avand id-ul unic de angajat dar si id-ul pentru departamentul din care fac parte.

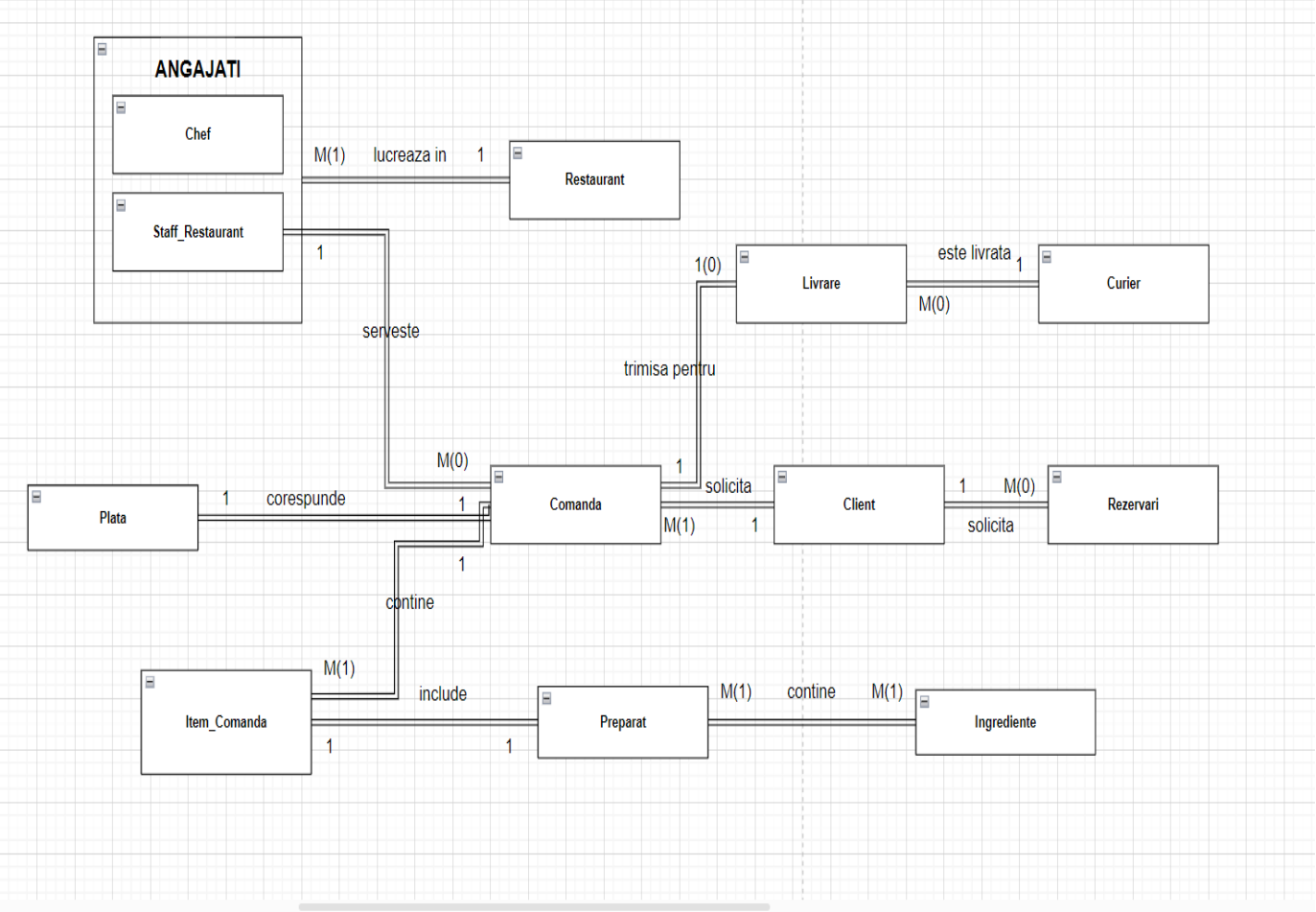
-Fiecare comanda este unica,identifiabila prin id. -Orice comanda are nota de plata.

-O comanda trimisa prin livrare este preluata de un curier. -Meniul contine diferite preparate,pretul lor fiind flexibil in functie de perioada.

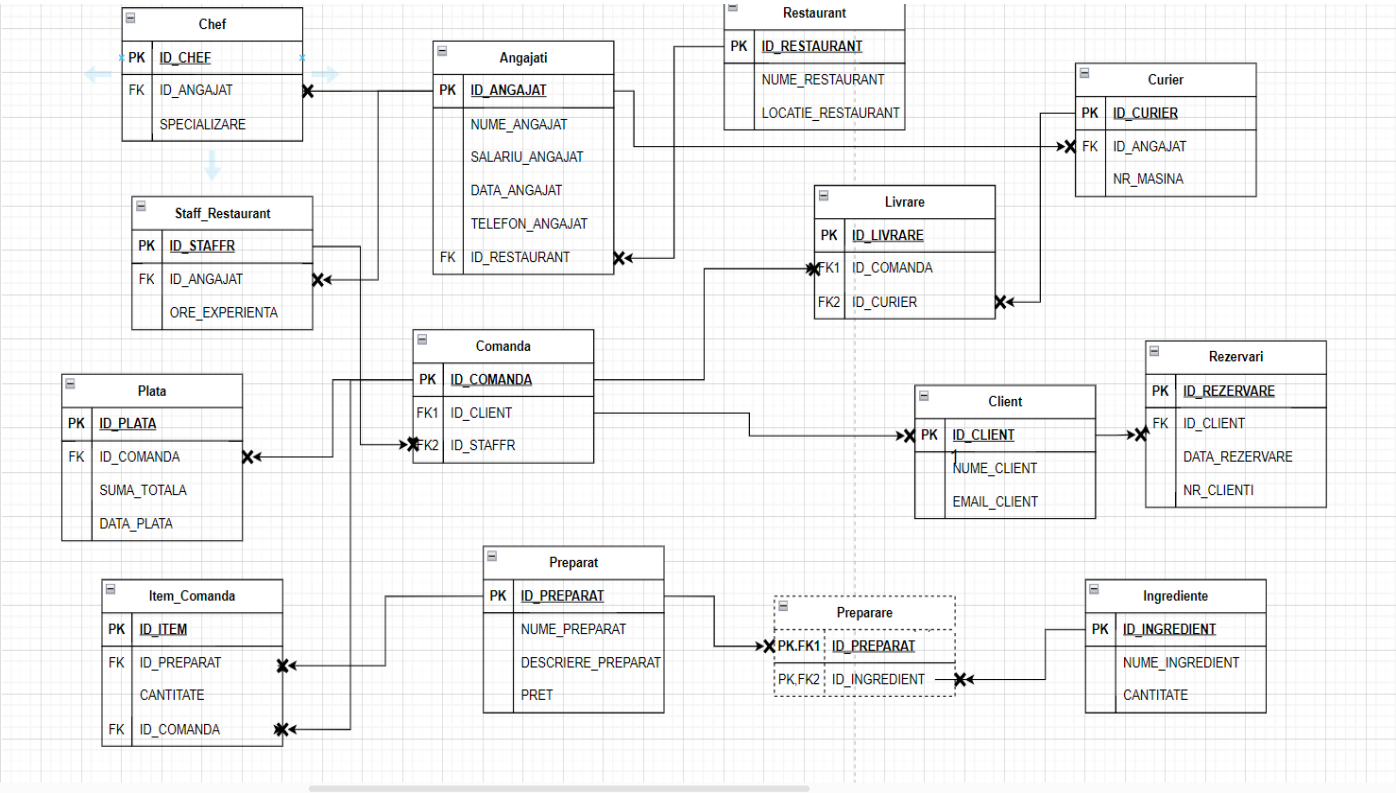
-Angajatii au salariul in functie de vechimea lor.

-Fiecare curier are o masina pentru livrare, costurile ei fiind decontate de catre restaurant.

2. Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele trebuie definite în limba română.



3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română.



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

```
create TABLE  RESTAURANT(
ID_RESTAURANT int not null primary key,
NUME_RESTAURANT varchar2(50),
LOCATIE_RESTAURANT varchar2(50)
);


create TABLE ANGAJATI(
ID_ANGAJAT int not null primary key,
NUME_ANGAJAT varchar2(50),
SALARIU_ANGAJAT int ,
DATA_ANGAJAT date,
TELEFON_ANGAJAT varchar2(10),
```

```
ID_RESTAURANT not null,
foreign key(ID_RESTAURANT) references RESTAURANT(ID_RESTAURANT)
);
create TABLE STAFF_RESTAURANT(
ID_STAFFR int not null primary key,
ID_ANGAJAT not null,
LUNI_EXPERIENTA int,
foreign key(ID_ANGAJAT) references ANGAJATI(ID_ANGAJAT)
);
create TABLE CHEF(
ID_CHEF int not null primary key,
ID_ANGAJAT not null,
SPECIALIZARE varchar2(20),
foreign key(ID_ANGAJAT) references ANGAJATI(ID_ANGAJAT)
);

create TABLE CLIENTI(
ID_CLIENT int not null primary key,
NUME_CLIENT varchar2(50),
EMAIL_CLIENT varchar2(50)
);

create TABLE REZERVARI(
ID_REZERVARE int not null primary key,
ID_CLIENT not null,
DATA_REZERVARE date,
NR_CLIENTI int,
foreign key(ID_CLIENT) references CLIENT(ID_CLIENT)
);
```

```
create TABLE COMANDA(
ID_COMANDA int not null primary key,
ID_CLIENT not null,
ID_STAFFR not null,
foreign key(ID_CLIENT) references CLIENT(ID_CLIENT),
foreign key(ID_STAFFR) references STAFF_RESTAURANT(ID_STAFFR)
);

create TABLE PREPARAT(
ID_PREPARAT int not null primary key,
NUME_PREPARAT varchar2(20),
DESCRIERE_PREPARAT varchar2(200)
PRET int,


);


create TABLE INGREDIENTE(
ID_INGREDIENT int not null primary key,
NUME_INGREDIENT varchar2(50),
Cantitate int,
foreign key(ID_COMANDA) references COMANDA(ID_COMANDA)
);


create TABLE PLATA(
ID_PLATA int not null primary key,
```

```
ID_COMANDA not null,
SUMA_TOTALA decimal,
DATA_PLATA date,
foreign key(ID_COMANDA)references COMANDA(ID_COMANDA)
);

create TABLE CURIER(
ID_CURIER int not  null primary key,
ID_ANGAJAT not null,
NR_MASINA int,
foreign key(ID_ANGAJAT)references ANGAJATI(ID_ANGAJAT)
);

create TABLE LIVRARE(
ID_LIVRARE int not null primary key,
ID_COMANDA not null,
ID_CURIER not null,
foreign key(ID_COMANDA) references COMANDA(ID_COMANDA),
foreign key(ID_CURIER) references CURIER(ID_CURIER)
);

create TABLE  ITEM_COMANDA(
ID_ITEM int not null primary key,
CANTITATE int,
ID_PREPARAT not null,
ID_COMANDA not null,
foreign key(ID_PREPARAT) references PREPARAT(ID_PREPARAT),
foreign key(ID_COMANDA) references COMANDA(ID_COMANDA)
);
```

Create TABLE PREPARARE(

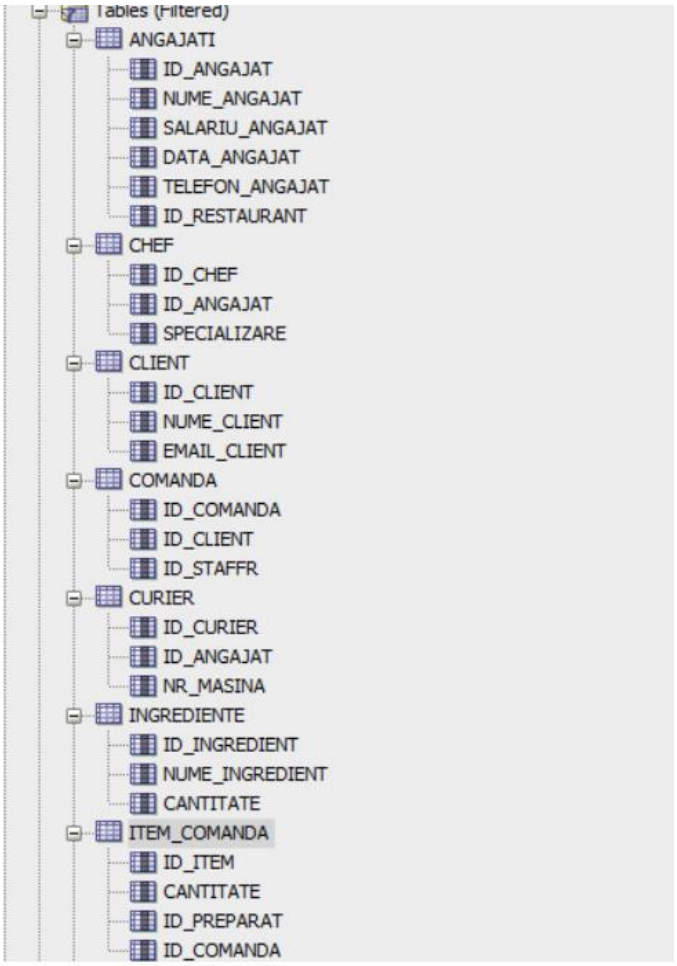
ID_PREPARAT not null,

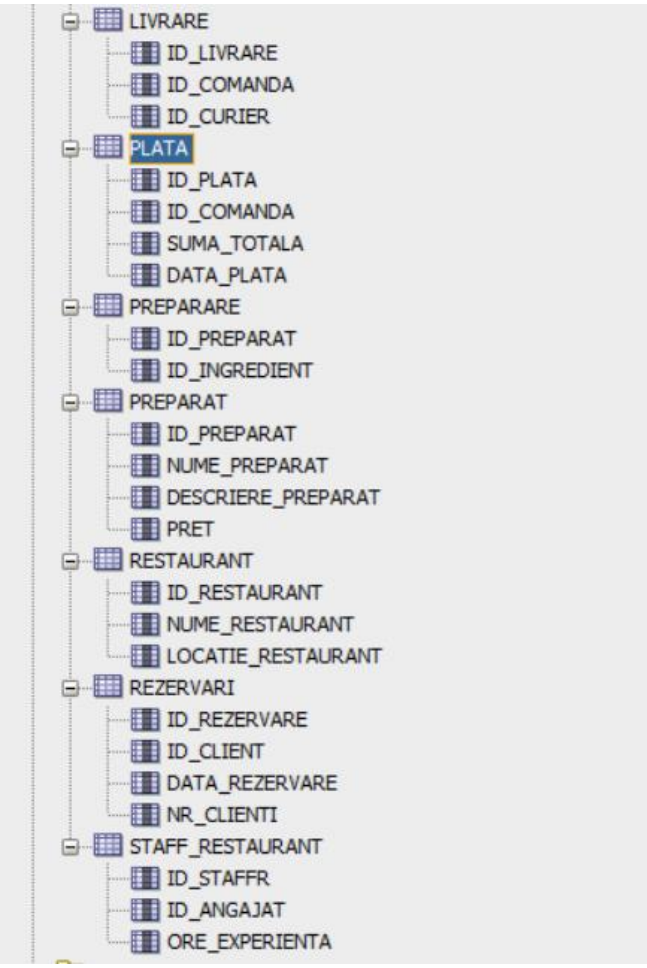
ID_INGREDIENT not null,

Foreign key(ID_PREPARAT) references PREPARAT(ID_PREPARAT),

Foreign key(ID_INGREDIENT) references INGREDIENT(ID_INGREDIENT)

);





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

```
insert into RESTAURANT VALUES(1,'Japanos Fusion','Soseaua Nordului');

insert into RESTAURANT VALUES(2,'Japanos by the Lake','Parcul Herastrau');

insert into RESTAURANT VALUES(3,'Japanos Tineretului','Bvd. Dimitrie Cantemir');

insert into RESTAURANT VALUES(4,'Japanos Takeaway','Bulevardul Lacul Tei');

insert into RESTAURANT VALUES(5,'Japanos Dorobanti','Strada George Calinescu');
```

	ID_RESTAURANT	NUME_RESTAURANT	LOCATIE_RESTAURANT
1	1	Japanos Fusion	Soseaua Nordului
2	2	Japanos by the Lake	Parcul Herastrau
3	3	Japanos Tineretului	Bvd. Dimitrie Cantemir
4	4	Japanos Takeaway	Bulevardul Lacul Tei
5	5	Japanos Dorobanti	Strada George Calinescu

SELECT * FROM ANGAJATI;

insert into ANGAJATI VALUES(111,'Popescu Radu',4400,to_date('01/03/2022','dd/mm/yyyy'),'0721020282',1);

insert into ANGAJATI VALUES(112,'Antonescu Mihai',5000,to_date('02/05/2021','dd/mm/yyyy'),'0724040502',2);

insert into ANGAJATI VALUES(113,'Andrei Stefan Catalin',5500,to_date('10/07/2020','dd/mm/yyyy'),'0732028213',3);

insert into ANGAJATI VALUES(114,'Nastea Calin',4400,to_date('01/10/2022','dd/mm/yyyy'),'0743209876',4);

insert into ANGAJATI VALUES(115,'Georgescu Maria',4400,to_date('25/05/2022','dd/mm/yyyy'),'0721384821',5);

insert into ANGAJATI VALUES(116,'Ionescu Ana',6000,to_date('14/03/2017','dd/mm/yyyy'),'0755338212',1);

insert into ANGAJATI VALUES(117,'Nedelcu Ana',5000,to_date('20/04/2020','dd/mm/yyyy'),'0722308202',2);

insert into ANGAJATI VALUES(118,'Ionita Alexandru',5000,to_date('14/12/2021','dd/mm/yyyy'),'0723458922',3);

insert into ANGAJATI VALUES(119,'Oprea Laurentiu',5000,to_date('01/06/2021','dd/mm/yyyy'),'0744994421',4);

insert into ANGAJATI VALUES(120,'Voinea Alexandru',6000,to_date('12/03/2019','dd/mm/yyyy'),'0721982394',5);

insert into ANGAJATI VALUES(121,'Ceausu Florina',7500,to_date('04/06/2016','dd/mm/yyyy'),'0721992182',1);

insert into ANGAJATI VALUES(122,'Velcea Maria',5500,to_date('05/02/2020','dd/mm/yyyy'),'0723402345',1);

insert into ANGAJATI VALUES(123,'Ammari Abdul',7500,to_date('06/10/2015','dd/mm/yyyy'),'0723449942',2);

```
insert into ANGAJATI VALUES(124,'Dumitru
Tudor',4400,to_date('20/09/2022','dd/mm/yyyy'),'0728992314',3);

insert into ANGAJATI VALUES(125,'Spataru
Vlad',5000,to_date('10/07/2021','dd/mm/yyyy'),'0788991234',4);

insert into ANGAJATI VALUES(126,'Dumitru
Vlad',5000,to_date('15/05/2021','dd/mm/yyyy'),'0799124422',5);

insert into ANGAJATI VALUES(127,'Draghici
Rares',5500,to_date('23/09/2021','dd/mm/yyyy'),'0723934412',1);

insert into ANGAJATI VALUES(128,'West
Kanye',4400,to_date('21/07/2021','dd/mm/yyyy'),'0721349821',2);

insert into ANGAJATI VALUES(129,'Dragomir
Stefania',6000,to_date('13/05/2020','dd/mm/yyyy'),'0788239124',3);

insert into ANGAJATI VALUES(130,'Iliescu
Ionut',5000,to_date('04/07/2021','dd/mm/yyyy'),'0728445503',4);

insert into ANGAJATI VALUES(131,'Turcu
Denisa',5500,to_date('01/04/2021','dd/mm/yyyy'),'0723557705',5);
```

	ID_ANGAJAT	NUME_ANGAJAT	SALARIU_ANGAJAT	DATA_ANGAJAT	TELEFON_ANGAJAT	ID_RESTAURANT
1	111	Popescu Radu	4400	01-MAR-22	0721020282	1
2	112	Antonescu Mihai	5000	02-MAY-21	0724040502	2
3	113	Andrei Stefan Catalin	5500	10-JUL-20	0732028213	3
4	114	Nastea Calin	4400	01-OCT-22	0743209876	4
5	115	Georgescu Maria	4400	25-MAY-22	0721384821	5
6	116	Ionescu Ana	6000	14-MAR-17	0755338212	1
7	117	Nedelcu Ana	5000	20-APR-20	0722308202	2
8	118	Ionita Alexandru	5000	14-DEC-21	0723458922	3
9	119	Oprea Laurentiu	5000	01-JUN-21	0744994421	4
10	120	Voinea Alexandru	6000	12-MAR-19	0721982394	5
11	121	Ceausu Florina	7500	04-JUN-16	0721992182	1
12	123	Ammari Abdul	7500	06-OCT-15	0723449942	2
13	124	Dumitru Tudor	4400	20-SEP-22	0728992314	3
14	125	Spataru Vlad	5000	10-JUL-21	0788991234	4
15	126	Dumitru Vlad	5000	15-MAY-21	0799124422	5
16	127	Draghici Rares	5500	23-SEP-21	0723934412	1
17	128	West Kanye	4400	21-JUL-21	0721349821	2
18	129	Dragomir Stefania	6000	13-MAY-20	0788239124	3
19	130	Iliescu Ionut	5000	04-JUL-21	0728445503	4
20	131	Turcu Denisa	5500	01-APR-21	0723557705	5
21	122	Velcea Maria	5500	05-FEB-20	0723402345	1

```
SELECT * FROM CLIENT;


insert into CLIENT(ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)

VALUES(seq_id_client.NEXTVAL,'Ion Dan','iondan@gmail.com');

insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)

VALUES(seq_id_client.NEXTVAL,'Oprea Ion','oprion@gmail.com');
```

```
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Letea Radu','letear@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Dobrin Letitia','dobr@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Camelia Carmen','carmeliac@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Rosoiu Luca','lucarosoiu@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Florea Andrei','floreandr@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Pintilie Ana','anapintilie@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Mihail Alexandra','alexandramihail@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Omar Ismail ','ismailomul@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Bisoc Alina','alinab@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Pene Florin','peneflorin@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Nitu Miruna','mirunadent@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Popovici David','davidpopovici@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Anghel Luminita','luminitaanghel@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Traistariu Mihai','mihaitraistariu@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
```

```
VALUES(seq_id_client.NEXTVAL,'Dumitrescu Bogdan','bogdand@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Dumitrascu Maria','mariadum@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Pavel Alina','pavelalina@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Ionescu Roxana','roxanaione@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Ionescu Mihai','ionescumihai@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Panait Razvan','panaitrazvan@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Stoian Radu','stoianradu@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Mihail Roxana','miharox@gmail.com');
insert into CLIENT (ID_CLIENT,NUME_CLIENT,EMAIL_CLIENT)
VALUES(seq_id_client.NEXTVAL,'Ivan Mihai','csgod@gmail.com');
```

	ID_CLIENT	NUME_CLIENT	EMAIL_CLIENT
1	1001	Ion Dan	iondan@gmail.com
2	1002	Oprea Ion	oprion@gmail.com
3	1003	Letea Radu	letear@gmail.com
4	1004	Dobrin Letitia	dobr@gmail.com
5	1005	Camelia Carmen	carmeliac@gmail.com
6	1006	Rosoiu Luca	lucarosoiu@gmail.com
7	1007	Florea Andrei	floreandr@gmail.com
8	1008	Pintilie Ana	anapintilie@gmail.com
9	1009	Mihail Alexandra	alexandramihail@gmail.com
10	1010	Omar Ismail	ismailomul@gmail.com
11	1011	Bisoc Alina	alinab@gmail.com
12	1012	Pene Florin	peneflorin@gmail.com
13	1013	Nitu Miruna	mirunadent@gmail.com
14	1014	Popovici David	davidpopovici@gmail.com
15	1015	Anghel Luminita	luminitaanghel@gmail.com
16	1016	Traistariu Mihai	mihaitraistariu@gmail.com
17	1017	Dumitrescu Bogdan	bogdand@gmail.com
18	1018	Dumitrascu Maria	mariadum@gmail.com
19	1019	Pavel Alina	pavelalina@gmail.com
20	1020	Ionescu Roxana	roxanaione@gmail.com
21	1021	Ionescu Mihai	ionescumihai@gmail.com
22	1022	Panait Razvan	panaitrazvan@gmail.com
23	1023	Stoian Radu	stoianradu@gmail.com
24	1024	Mihail Roxana	miharox@gmail.com
25	1025	Ivan Mihai	csgod@gmail.com

SELECT * FROM STAFF_RESTAURANT;

insert into STAFF_RESTAURANT VALUES(501,114,7);

insert into STAFF_RESTAURANT VALUES(502,111,14);

insert into STAFF_RESTAURANT VALUES(503,117,25);

insert into STAFF_RESTAURANT VALUES(504,128,22);

insert into STAFF_RESTAURANT VALUES(505,125,20);

insert into STAFF_RESTAURANT VALUES(506,131,25);

	ID_STAFFR	ID_ANGAJAT	ORE_EXPERIENTA
1	501	114	7
2	502	111	14
3	503	117	25
4	504	128	22
5	505	125	20
6	506	131	25

SELECT * FROM COMANDA;

insert into COMANDA VALUES(10,1004,501);

insert into COMANDA VALUES(11,1001,501);

insert into COMANDA VALUES(12,1010,502);

insert into COMANDA VALUES(13,1012,502);

insert into COMANDA VALUES(14,1014,503);

insert into COMANDA VALUES(15,1013,503);

insert into COMANDA VALUES(16,1001,503);

insert into COMANDA VALUES(17,1009,503);

insert into COMANDA VALUES(18,1008,503);

insert into COMANDA VALUES(19,1006,504);

insert into COMANDA VALUES(20,1013,504);

insert into COMANDA VALUES(21,1015,505);

insert into COMANDA VALUES(22,1020,506);

insert into COMANDA VALUES(23,1021,503);

insert into COMANDA VALUES(24,1022,505);

insert into COMANDA VALUES(25,1023,503);

insert into COMANDA VALUES(26,1024,501);

insert into COMANDA VALUES(27,1025,503);

insert into COMANDA VALUES(28,1001,501);

insert into COMANDA VALUES(29,1001,505);

insert into COMANDA VALUES(30,1001,505);

insert into COMANDA VALUES(31,1001,506);

	ID_COMANDA	ID_CLIENT	ID_STAFFR
1	10	1004	501
2	11	1001	501
3	12	1010	502
4	13	1012	502
5	14	1014	503
6	15	1013	503
7	16	1001	503
8	17	1009	503
9	18	1008	503
10	19	1006	504
11	20	1013	504
12	21	1015	505
13	22	1020	506
14	23	1021	503
15	24	1022	505
16	25	1023	503
17	26	1024	501
18	27	1025	503

SELECT * FROM CHEF;

insert into CHEF VALUES(701,120,'Sushi');

insert into CHEF VALUES(702,121,'Sushi');

insert into CHEF VALUES(703,123,'Sushi');

insert into CHEF VALUES(704,116,'Ramen');

insert into CHEF VALUES(705,112,'Ramen');

insert into CHEF VALUES(706,119,'Ramen');

insert into CHEF VALUES(707,113,'Dessert');

insert into CHEF VALUES(708,118,'Dessert');

insert into CHEF VALUES(709,127,'Sushi');

insert into CHEF VALUES(710,130,'Ramen');

	ID_CHEF	ID_ANGAJAT	SPECIALIZARE
1	701	120	Sushi
2	702	121	Sushi
3	703	123	Sushi
4	704	116	Ramen
5	705	112	Ramen
6	706	119	Ramen
7	707	113	Dessert
8	708	118	Dessert
9	709	127	Sushi
10	710	130	Ramen

SELECT * FROM INGREDIENTE;

insert into INGREDIENTE VALUES(801,'Ton',90);

insert into INGREDIENTE VALUES(802,'Somon',55);

insert into INGREDIENTE VALUES(803,'Orez',231);

insert into INGREDIENTE VALUES(804,'Sos Soia',30);

insert into INGREDIENTE VALUES(805,'Creveti',40);

insert into INGREDIENTE VALUES(806,'Wasabi',70);

insert into INGREDIENTE VALUES(807,'Crab',220);

insert into INGREDIENTE VALUES(808,'Ciuperci',300);

insert into INGREDIENTE VALUES(809,'Pasta Miso',120);

insert into INGREDIENTE VALUES(810,'Porc',50);

insert into INGREDIENTE VALUES(811,'Avocado',80);

insert into INGREDIENTE VALUES(812,'Castravete',90);

	ID_INGREDIENT	NUME_INGREDIENT	CANTITATE
1	801	Ton	90
2	802	Somon	55
3	803	Orez	231
4	804	Sos Soia	30
5	805	Creveti	40
6	806	Wasabi	70
7	807	Crab	220
8	808	Ciuperci	300
9	809	Pasta Miso	120
10	810	Porc	50
11	811	Avocado	80
12	812	Castravete	90

SELECT * FROM PREPARAT;

insert into PREPARAT VALUES(901,'Nigiri Ton','File de ton pe un pat de orez presat ',801);

insert into PREPARAT VALUES(902,'Nigiri Somon','File de somon pe un pat de orez presat ',802);

insert into PREPARAT VALUES(903,'Maki Ton','Rulou Maki clasic cu ton',801);

insert into PREPARAT VALUES(904,'Maki Somon','Rulou Maki clasic cu somon',802);

insert into PREPARAT VALUES(905,'Maki Avocado','Rulou Maki clasic cu avocado',811);

insert into PREPARAT VALUES(906,'Maki Spicy Somon','Rulou Maki clasic somon cu sosul spicy Japanos',802);

insert into PREPARAT VALUES(907,'Maki California','Rulou Maki care contine crab,avocado , castravete si seminte de susan',807);

insert into PREPARAT VALUES(908,'Maki Philadelphia','Rulou Maki care contine somon, cream cheese si castravete',802);

insert into PREPARAT VALUES(909,'Ramen Veggie','Ramen vegetarian pe baza de ciuperci shiitake',808);

insert into PREPARAT VALUES(910,'Ramen Porc','Ramenul clasic, construit in jurul porcului in stil chashu ',810);

insert into PREPARAT VALUES(911,'Supa Miso','Supa traditionala japoneza, ingredientul principal fiind pasta miso',809);

insert into PREPARAT VALUES(912,'Mochi','Prajitura japoneza facuta din orez',803);

UPDATE PREPARAT SET PRET = 20 WHERE ID_PREPARAT=901;

UPDATE PREPARAT SET PRET = 21 WHERE ID_PREPARAT=902;

UPDATE PREPARAT SET PRET = 25 WHERE ID_PREPARAT=903;

UPDATE PREPARAT SET PRET = 30 WHERE ID_PREPARAT=904;

UPDATE PREPARAT SET PRET = 20 WHERE ID_PREPARAT=905;

UPDATE PREPARAT SET PRET = 30 WHERE ID_PREPARAT=906;

UPDATE PREPARAT SET PRET = 30 WHERE ID_PREPARAT=907;

UPDATE PREPARAT SET PRET = 30 WHERE ID_PREPARAT=908;

UPDATE PREPARAT SET PRET = 25 WHERE ID_PREPARAT=909;

UPDATE PREPARAT SET PRET = 25 WHERE ID_PREPARAT=910;

UPDATE PREPARAT SET PRET = 20 WHERE ID_PREPARAT=911;

UPDATE PREPARAT SET PRET = 12 WHERE ID_PREPARAT=912;

Script Output | Query Result

SQL | All Rows Fetched: 12 in 0.004 seconds

	ID_PREPARAT	NUME_PREPARAT	DESCRIERE_PREPARAT	PRET
1	901	Nigiri Ton	File de ton pe un pat de orez presat	20
2	902	Nigiri Somon	File de somon pe un pat de orez presat	21
3	903	Maki Ton	Rulou Maki clasic cu ton	25
4	904	Maki Somon	Rulou Maki clasic cu somon	30
5	905	Maki Avocado	Rulou Maki clasic cu avocado	20
6	906	Maki Spicy Somon	Rulou Maki clasic somon cu sosul spicy Japanos	30
7	907	Maki California	Rulou Maki care contine crab,avocado , castravete si seminte de susan	30
8	908	Maki Philadelphia	Rulou Maki care contine somon, cream cheese si castravete	30
9	909	Ramen Veggie	Ramen vegetarian pe baza de ciuperci shiitake	25
10	910	Ramen Porc	Ramenul clasic, construit in jurul porcului in stil chashu	25
11	911	Supa Miso	Supa traditionala japoneza, ingredientul principal fiind pasta miso	20
12	912	Mochi	Prajitura japoneza facuta din orez	12

SELECT * FROM CURIER;

insert into CURIER VALUES(1501,115,111);

insert into CURIER VALUES(1502,122,222);

insert into CURIER VALUES(1503,124,333);

insert into CURIER VALUES(1504,126,444);

insert into CURIER VALUES(1505,129,555);

	ID_CURIER	ID_ANGAJAT	NR_MASINA
1	1501	115	111
2	1503	124	333
3	1504	126	444
4	1505	129	555
5	1502	122	222

SELECT * FROM ITEM_COMANDA;

insert into ITEM_COMANDA VALUES(651,1,901,10);

insert into ITEM_COMANDA VALUES(652,2,905,11);

insert into ITEM_COMANDA VALUES(653,4,903,12);

insert into ITEM_COMANDA VALUES(654,3,906,13);

insert into ITEM_COMANDA VALUES(655,1,907,14);

insert into ITEM_COMANDA VALUES(656,1,910,15);

insert into ITEM_COMANDA VALUES(657,2,910,16);

insert into ITEM_COMANDA VALUES(658,2,903,17);

insert into ITEM_COMANDA VALUES(659,1,901,18);

insert into ITEM_COMANDA VALUES(660,3,902,19);

insert into ITEM_COMANDA VALUES(661,4,904,20);

insert into ITEM_COMANDA VALUES(662,5,905,21);

insert into ITEM_COMANDA VALUES(663,6,912,22);

insert into ITEM_COMANDA VALUES(664,2,911,13);

insert into ITEM_COMANDA VALUES(665,3,906,14);

insert into ITEM_COMANDA VALUES(666,4,908,13);

insert into ITEM_COMANDA VALUES(667,1,909,16);

insert into ITEM_COMANDA VALUES(668,2,903,17);

insert into ITEM_COMANDA VALUES(669,1,902,18);

insert into ITEM_COMANDA VALUES(670,1,906,13);

insert into ITEM_COMANDA VALUES(671,3,909,14);

Script Output | Query Result

SQL | All Rows Fetched: 21 in 0.006 seconds

	ID_ITEM	CANTITATE	ID_PREPARAT	ID_COMANDA
1	651	1	901	10
2	652	2	905	11
3	653	4	903	12
4	654	3	906	13
5	655	1	907	14
6	656	1	910	15
7	657	2	910	16
8	658	2	903	17
9	659	1	901	18
10	660	3	902	19
11	661	4	904	20
12	662	5	905	21
13	663	6	912	22
14	665	3	906	14
15	664	2	911	13
16	666	4	908	13
17	667	1	909	16
18	668	2	903	17
19	669	1	902	18
20	670	1	906	13
21	671	3	909	14

SELECT * FROM PLATA;

insert into PLATA VALUES(350,10,48,to_date('10/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(351,11,50,to_date('08/05/2023','dd/mm/yyyy'));

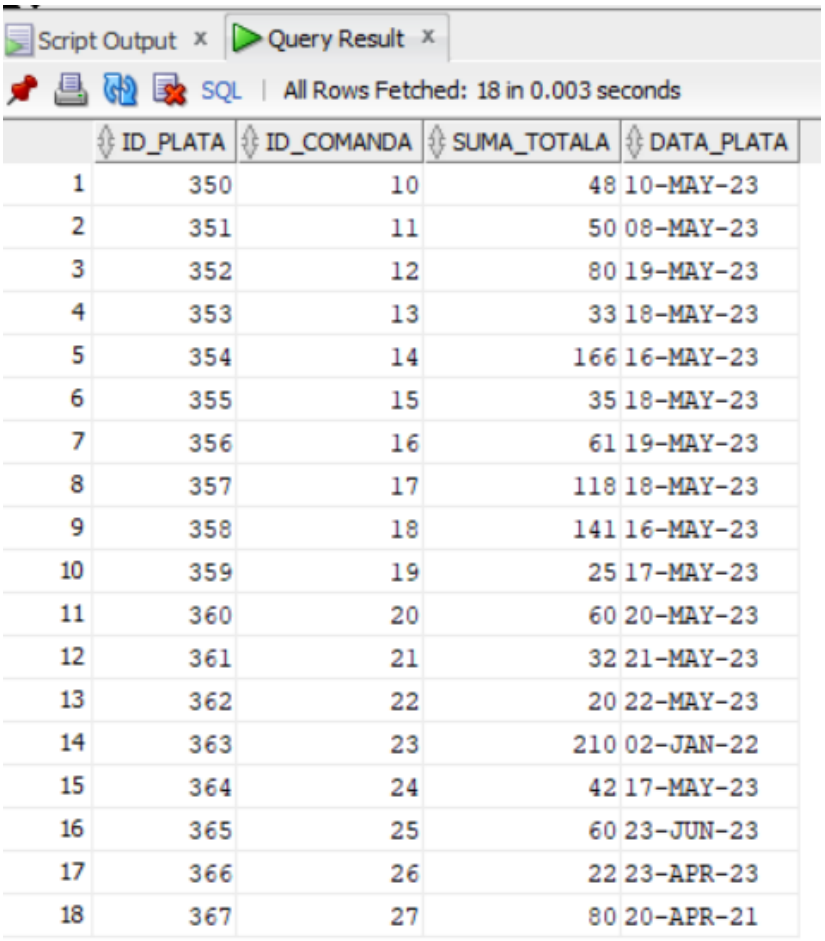
insert into PLATA VALUES(352,12,80,to_date('19/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(353,13,33,to_date('18/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(354,14,166,to_date('16/05/2023','dd/mm/yyyy'));

insert into PLATA VALUES(355,15,35,to_date('18/05/2023','dd/mm/yyyy'));

```
insert into PLATA VALUES(356,16,61,to_date('19/05/2023','dd/mm/yyyy'));
insert into PLATA VALUES(357,17,118,to_date('18/05/2023','dd/mm/yyyy'));
insert into PLATA VALUES(358,18,141,to_date('16/05/2023','dd/mm/yyyy'));
insert into PLATA VALUES(359,19,25,to_date('17/05/2023','dd/mm/yyyy'));
insert into PLATA VALUES(360,20,60,to_date('20/05/2023','dd/mm/yyyy'));
insert into PLATA VALUES(361,21,32,to_date('21/05/2023','dd/mm/yyyy'));
insert into PLATA VALUES(362,22,20,to_date('22/05/2023','dd/mm/yyyy'));
insert into PLATA VALUES(363,23,210,to_date('02/01/2022','dd/mm/yyyy'));
insert into PLATA VALUES(364,24,42,to_date('17/05/2023','dd/mm/yyyy'));
insert into PLATA VALUES(365,25,60,to_date('23/06/2023','dd/mm/yyyy'));
insert into PLATA VALUES(366,26,22,to_date('23/04/2023','dd/mm/yyyy'));
insert into PLATA VALUES(367,27,80,to_date('20/04/2021','dd/mm/yyyy'));
```


Script Output | Query Result

SQL | All Rows Fetched: 18 in 0.003 seconds

	ID_PLATA	ID_COMANDA	SUMA_TOTALA	DATA_PLATA
1	350	10	48	10-MAY-23
2	351	11	50	08-MAY-23
3	352	12	80	19-MAY-23
4	353	13	33	18-MAY-23
5	354	14	166	16-MAY-23
6	355	15	35	18-MAY-23
7	356	16	61	19-MAY-23
8	357	17	118	18-MAY-23
9	358	18	141	16-MAY-23
10	359	19	25	17-MAY-23
11	360	20	60	20-MAY-23
12	361	21	32	21-MAY-23
13	362	22	20	22-MAY-23
14	363	23	210	02-JAN-22
15	364	24	42	17-MAY-23
16	365	25	60	23-JUN-23
17	366	26	22	23-APR-23
18	367	27	80	20-APR-21

SELECT * FROM REZERVARI;

insert into REZERVARI VALUES(6001,1001,to_date('10/05/2023','dd/mm/yyyy'),3);

insert into REZERVARI VALUES(6002,1002,to_date('08/05/2023','dd/mm/yyyy'),2);

insert into REZERVARI VALUES(6003,1003,to_date('19/05/2023','dd/mm/yyyy'),4);

insert into REZERVARI VALUES(6004,1004,to_date('18/05/2023','dd/mm/yyyy'),2);

insert into REZERVARI VALUES(6005,1005,to_date('16/05/2023','dd/mm/yyyy'),2);

insert into REZERVARI VALUES(6006,1006,to_date('18/05/2023','dd/mm/yyyy'),2);

insert into REZERVARI VALUES(6007,1007,to_date('19/05/2023','dd/mm/yyyy'),3);

insert into REZERVARI VALUES(6008,1008,to_date('18/05/2023','dd/mm/yyyy'),5);

insert into REZERVARI VALUES(6009,1009,to_date('16/05/2023','dd/mm/yyyy'),2);

insert into REZERVARI VALUES(6010,1010,to_date('17/05/2023','dd/mm/yyyy'),3);

insert into REZERVARI VALUES(6011,1020,to_date('17/05/2023','dd/mm/yyyy'),3);

	ID_REZERVARE	ID_CLIENT	DATA_REZERVARE	NR_CLIENTI
1	6001	1001	10-MAY-23	3
2	6002	1002	08-MAY-23	2
3	6003	1003	19-MAY-23	4
4	6004	1004	18-MAY-23	2
5	6005	1005	16-MAY-23	2
6	6006	1006	18-MAY-23	2
7	6007	1007	19-MAY-23	3
8	6008	1008	18-MAY-23	5
9	6009	1009	16-MAY-23	2
10	6010	1010	17-MAY-23	3

SELECT * FROM LIVRARE;

insert into LIVRARE VALUES(2301,23,1501);

insert into LIVRARE VALUES(2302,24,1502);

insert into LIVRARE VALUES(2303,25,1503);

insert into LIVRARE VALUES(2304,26,1504);

insert into LIVRARE VALUES(2305,27,1505);

	ID_LIVRARE	ID_COMANDA	ID_CURIER
1	2301	23	1501
2	2302	24	1502
3	2303	25	1503
4	2304	26	1504
5	2305	27	1505

SELECT * FROM PREPARARE;

Insert into PREPARARE VALUES(901,802);

Insert into PREPARARE VALUES(902,809);

Insert into PREPARARE VALUES(903,808);

Insert into PREPARARE VALUES(904,804);

Insert into PREPARARE VALUES(905,801);

Insert into PREPARARE VALUES(906,801);

Insert into PREPARARE VALUES(907,803);

Insert into PREPARARE VALUES(908,803);

Insert into PREPARARE VALUES(909,806);

Insert into PREPARARE VALUES(910,807);

Insert into PREPARARE VALUES(911,802);

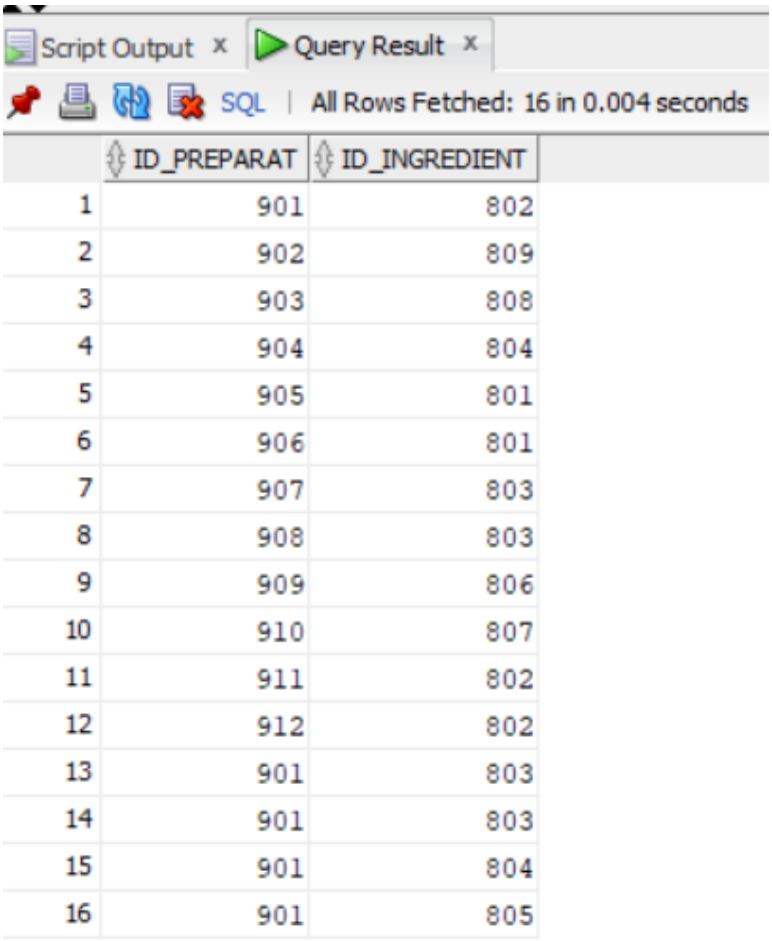
Insert into PREPARARE VALUES(912,802);

Insert into PREPARARE VALUES(901,803);

Insert into PREPARARE VALUES(901,803);

Insert into PREPARARE VALUES(901,804);

Insert into PREPARARE VALUES(901,805);



	ID_PREPARAT	ID_INGREDIENT
1	901	802
2	902	809
3	903	808
4	904	804
5	905	801
6	906	801
7	907	803
8	908	803
9	909	806
10	910	807
11	911	802
12	912	802
13	901	803
14	901	803
15	901	804
16	901	805

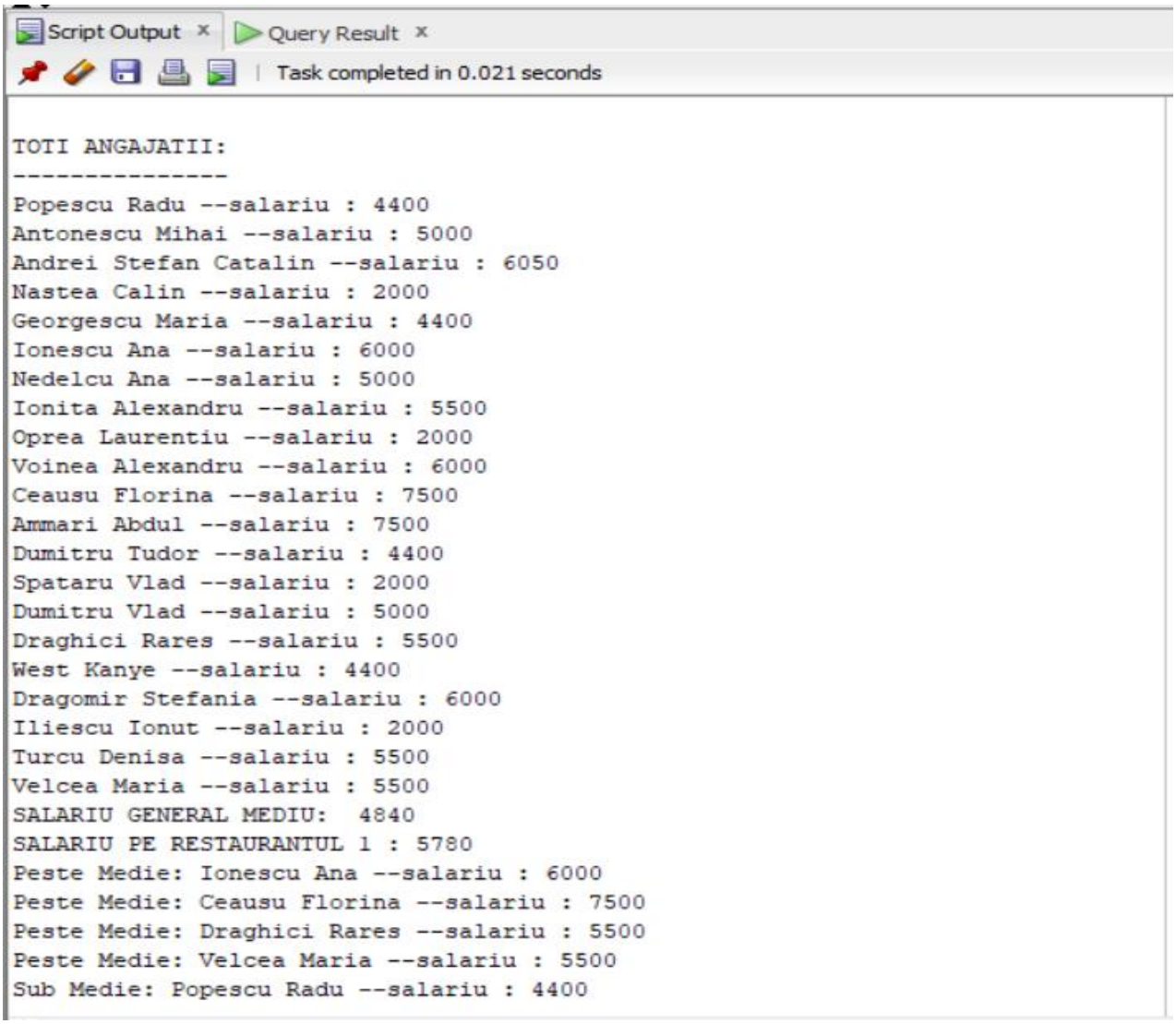
6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.

Sa se afiseze angajatii unui restaurant al carui id este dat, salariul mediu pe toate restaurantele, salariul mediu pe restaurantul specificat, si daca angajatii din restaurantul specificat au salariu sub/peste medie.

```
CREATE OR REPLACE PROCEDURE EX_6(id RESTAURANT.ID_RESTAURANT%type)
IS
TYPE Tablou_Angajati is TABLE OF ANGAJATI%ROWTYPE index by pls_integer;
TYPE tablou_imbricat is TABLE OF ANGAJATI%ROWTYPE;
TYPE vector is varray(100) of ANGAJATI%rowtype;
```

```
toti_ang tablou_angajati;
tt vector;
ts tablou_imbricat;
medie_total int;
medie_t int;

BEGIN
SELECT * BULK COLLECT INTO toti_ang from ANGAJATI;

DBMS_OUTPUT.PUT_LINE('TOTI ANGAJATII:');
DBMS_OUTPUT.PUT_LINE('---------------');
FOR i in 1..toti_ang.COUNT loop
   DBMS_OUTPUT.PUT_LINE(toti_ang(i).nume_angajat || ' --salariu : ' ||
toti_ang(i).salariu_angajat );
end loop;
select round(avg(salariu_angajat))
into medie_total
from angajati;

select round(avg(salariu_angajat))
into medie_t
from angajati where id_restaurant=id;
DBMS_OUTPUT.PUT_LINE('SALARIU GENERAL MEDIU:  ' || medie_total);
DBMS_OUTPUT.PUT_LINE('SALARIU PE RESTAURANTUL ' || id || ' : ' || medie_t);
SELECT * BULK COLLECT INTO tt FROM ANGAJATI WHERE ID_RESTAURANT= id
AND SALARIU_ANGAJAT > medie_total;
SELECT * BULK COLLECT INTO ts FROM ANGAJATI WHERE ID_RESTAURANT= id
AND SALARIU_ANGAJAT < medie_total;

FOR i in 1..tt.COUNT loop
```

```
   DBMS_OUTPUT.PUT_LINE('Peste Medie: ' || tt(i).nume_angajat || ' --salariu : ' ||
tt(i).salariu_angajat );

end loop;

FOR i in 1..ts.COUNT loop

   DBMS_OUTPUT.PUT_LINE('Sub Medie: ' || ts(i).nume_angajat || ' --salariu : ' ||
ts(i).salariu_angajat );

end loop;

END;

/

BEGIN

   EX_6(1);

END;
```



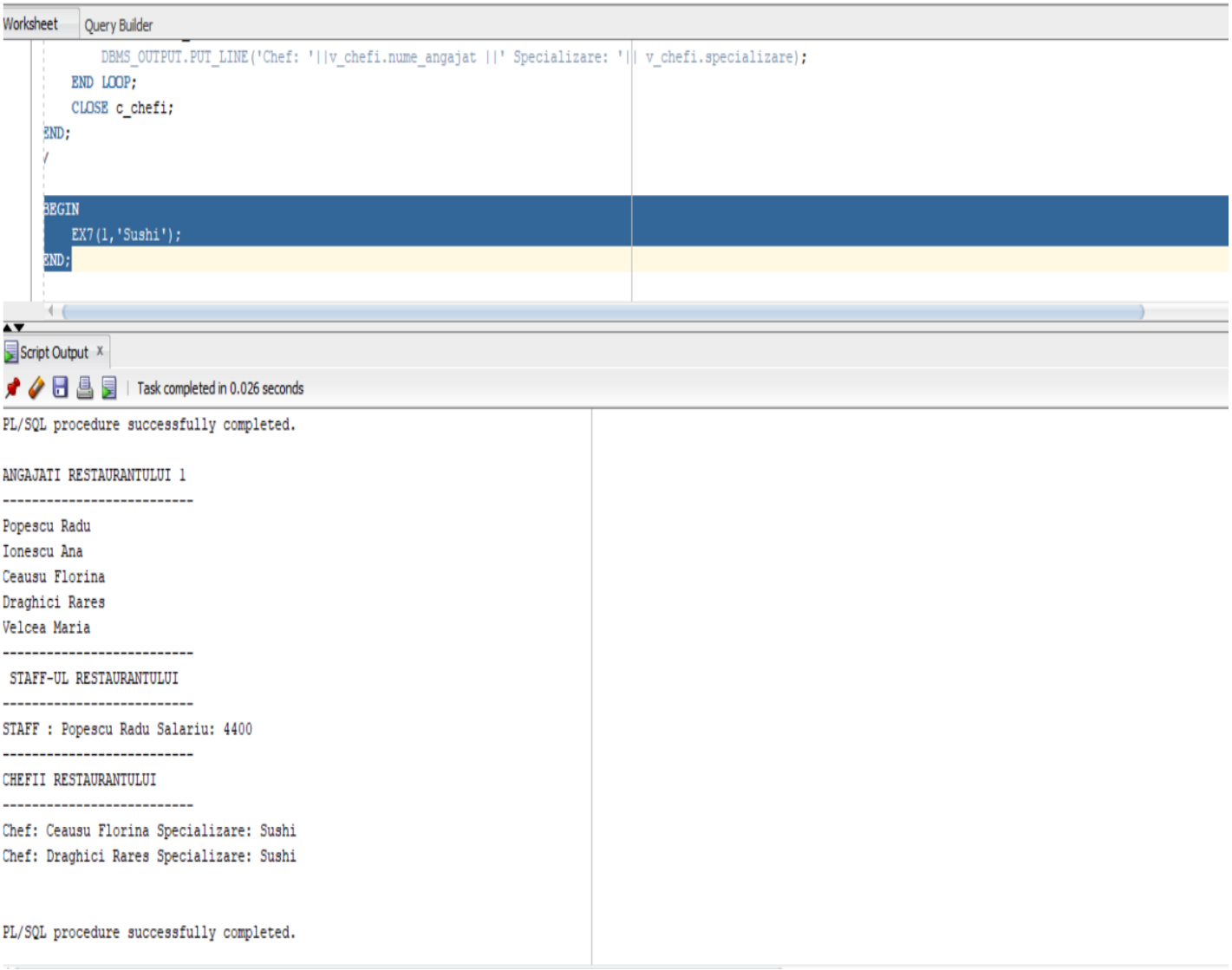
7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul.

Sa se afiseze angajatii unui restaurant al carui id este dat , apoi sa se afiseze staff-ul restaurantului.Se cere si afisarea chefilor din restaurantul respectiv a caror specialitate este data .

```
CREATE OR REPLACE PROCEDURE ex7 (id_rest IN
RESTAURANT.ID_RESTAURANT%type,p_spec IN VARCHAR2)

AS
```

```
    cursor c_ang(id_rest IN RESTAURANT.ID_RESTAURANT%type) is
        select * from ANGAJATI a
        where a.id_restaurant=id_rest;
       cursor c_chefi is
        select * from chef ch
        join angajati a on ch.id_angajat=a.id_angajat
        where a.id_restaurant=id_rest AND ch.specializare=p_spec;

        v_angajati c_ang%ROWTYPE;
        v_chefi c_chefi%ROWTYPE;
    cursor c_staffr(id_rest IN RESTAURANT.ID_RESTAURANT%TYPE) IS
        select * from STAFF_RESTAURANT s JOIN ANGAJATI a ON
s.ID_ANGAJAT=a.id_angajat
        where a.id_restaurant=id_rest;
        v_staffr c_staffr%ROWTYPE;

BEGIN
    DBMS_OUTPUT.PUT_LINE(' ANGAJATI RESTAURANTULUI ' || id_rest);
    DBMS_OUTPUT.PUT_LINE('-------------------------');
    OPEN c_ang(id_rest);
    LOOP
        FETCH c_ang INTO v_angajati;
        EXIT WHEN c_ang%NOTFOUND;

        DBMS_OUTPUT.PUT_LINE(v_angajati.NUME_ANGAJAT);
    END LOOP;
    CLOSE c_ang;
    DBMS_OUTPUT.PUT_LINE('-------------------------');
    DBMS_OUTPUT.PUT_LINE(' STAFF-UL RESTAURANTULUI ');
```

```
   DBMS_OUTPUT.PUT_LINE('--------------------------');
   OPEN c_staffr(id_rest);
   LOOP
      FETCH c_staffr INTO v_staffr;
      EXIT WHEN c_staffr%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('STAFF : '||v_staffr.NUME_ANGAJAT ||' Salariu: '
||v_staffr.SALARIU_ANGAJAT );
   END LOOP;
   DBMS_OUTPUT.PUT_LINE('--------------------------');
   DBMS_OUTPUT.PUT_LINE('CHEFII RESTAURANTULUI ');
   DBMS_OUTPUT.PUT_LINE('--------------------------');
   OPEN c_chefi;
   LOOP
      FETCH c_chefi INTO v_chefi;
      EXIT WHEN c_chefi%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('Chef: '||v_chefi.nume_angajat ||' Specializare: '||
v_chefi.specializare);
   END LOOP;
   CLOSE c_chefi;
END;
/
```



8. 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.

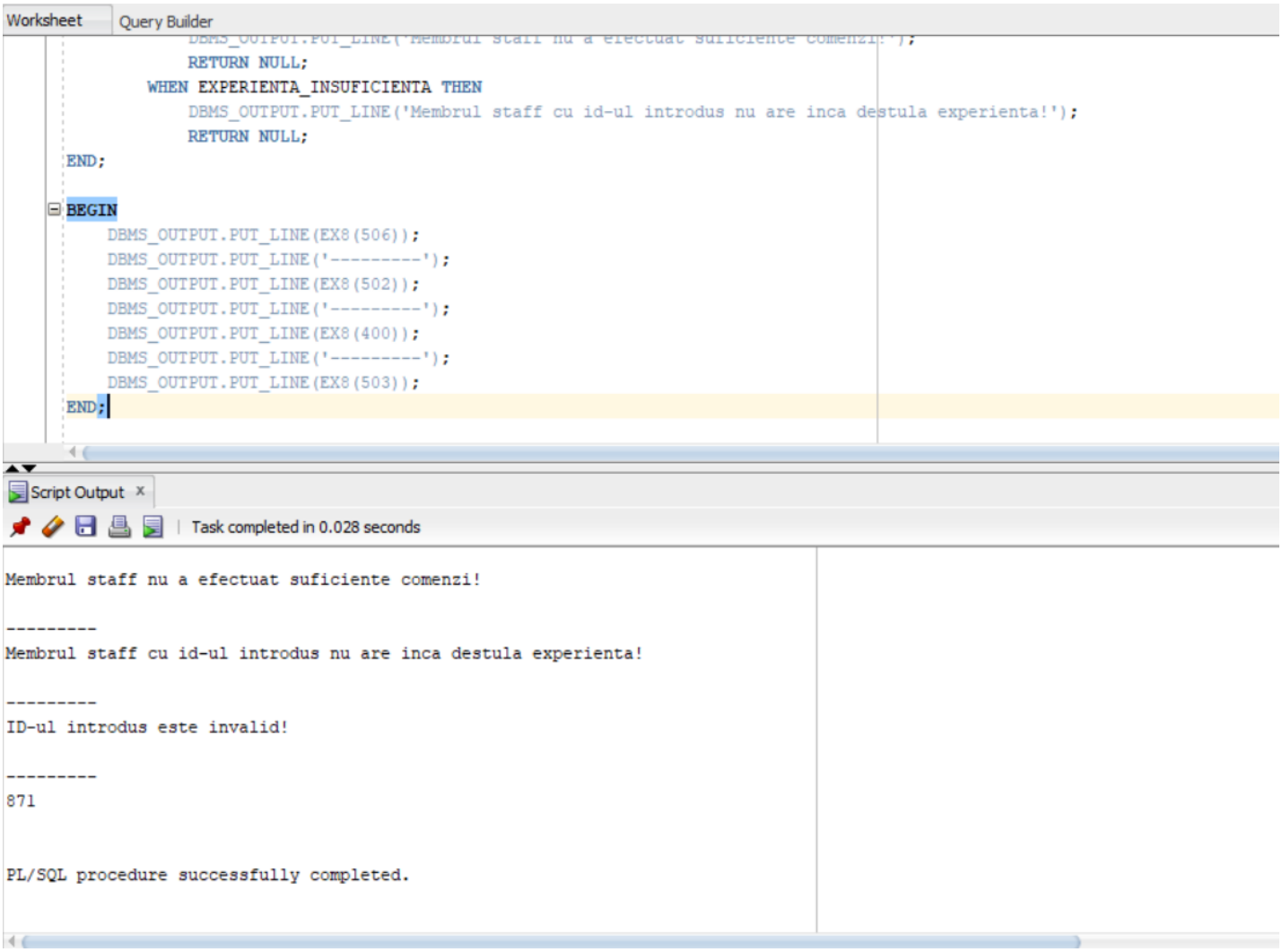
Definiți minim 2 excepții proprii. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate.

Pentru un membru din staff al carui id este dat, sa se afiseze suma totala din comenzile efectuate de acesta. Membrii din staff care nu au cel putin 20 de ore de experinta sau cel putin 2 comenzi efectuate nu sunt luati in considerare.

```
CREATE OR REPLACE FUNCTION EX8(p_id_staffr IN NUMBER) RETURN VARCHAR2
AS
    v_suma NUMBER;
```

```
    v_total NUMBER;
    v_cnt NUMBER;
    v_cursor SYS_REFCURSOR;
    v_ore_experienta NUMBER;
    ID_INVALID EXCEPTION;
    PREA_PUTINE_COMENZI EXCEPTION;
    EXPERIENTA_INSUFICIENTA EXCEPTION;

BEGIN
    IF p_id_staffr < 501 OR p_id_staffr > 506 THEN
        RAISE ID_INVALID;
    END IF;
    SELECT ORE_EXPERIENTA INTO v_ore_experienta FROM STAFF_RESTAURANT
    WHERE ID_STAFFR=p_id_staffr;
    IF v_ore_experienta < 20 THEN
        RAISE EXPERIENTA_INSUFICIENTA;
    END IF;

    OPEN v_cursor FOR
        SELECT p.SUMA_TOTALA FROM PLATA p
        JOIN COMANDA c ON p.ID_COMANDA = c.ID_COMANDA
        JOIN STAFF_RESTAURANT s ON c.ID_STAFFR = s.ID_STAFFR
        WHERE s.ID_STAFFR=p_id_staffr;
    v_suma :=0;
    v_total:=0;
    v_cnt:=0;
    LOOP
        FETCH v_cursor INTO v_suma;
        EXIT WHEN v_cursor%NOTFOUND;
```

```
    v_total:= v_total + v_suma;
    v_cnt:=v_cnt+1;
  END LOOP;
  CLOSE v_cursor;
  IF v_cnt < 3 THEN
    RAISE PREA_PUTINE_COMENZI;
  END IF;
  RETURN v_total;
  EXCEPTION
    WHEN ID_INVALID THEN
      DBMS_OUTPUT.PUT_LINE('ID-ul introdus este invalid!');
      RETURN NULL;
    WHEN PREA_PUTINE_COMENZI THEN
      DBMS_OUTPUT.PUT_LINE('Membrul staff nu a efectuat suficiente comenzi!');
      RETURN NULL;
    WHEN EXPERIENTA_INSUFICIENTA THEN
      DBMS_OUTPUT.PUT_LINE('Membrul staff cu id-ul introdus nu are inca destula
experienta!');
      RETURN NULL;
END;
```



9. 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.

Sa se afiseze pentru un client al carui nume ii este dat, cel mai scump preparat comandat(nume+pret) impreuna cu id-ul platii respective .

CREATE OR REPLACE PROCEDURE ex9(p_nume_client IN VARCHAR2) AS

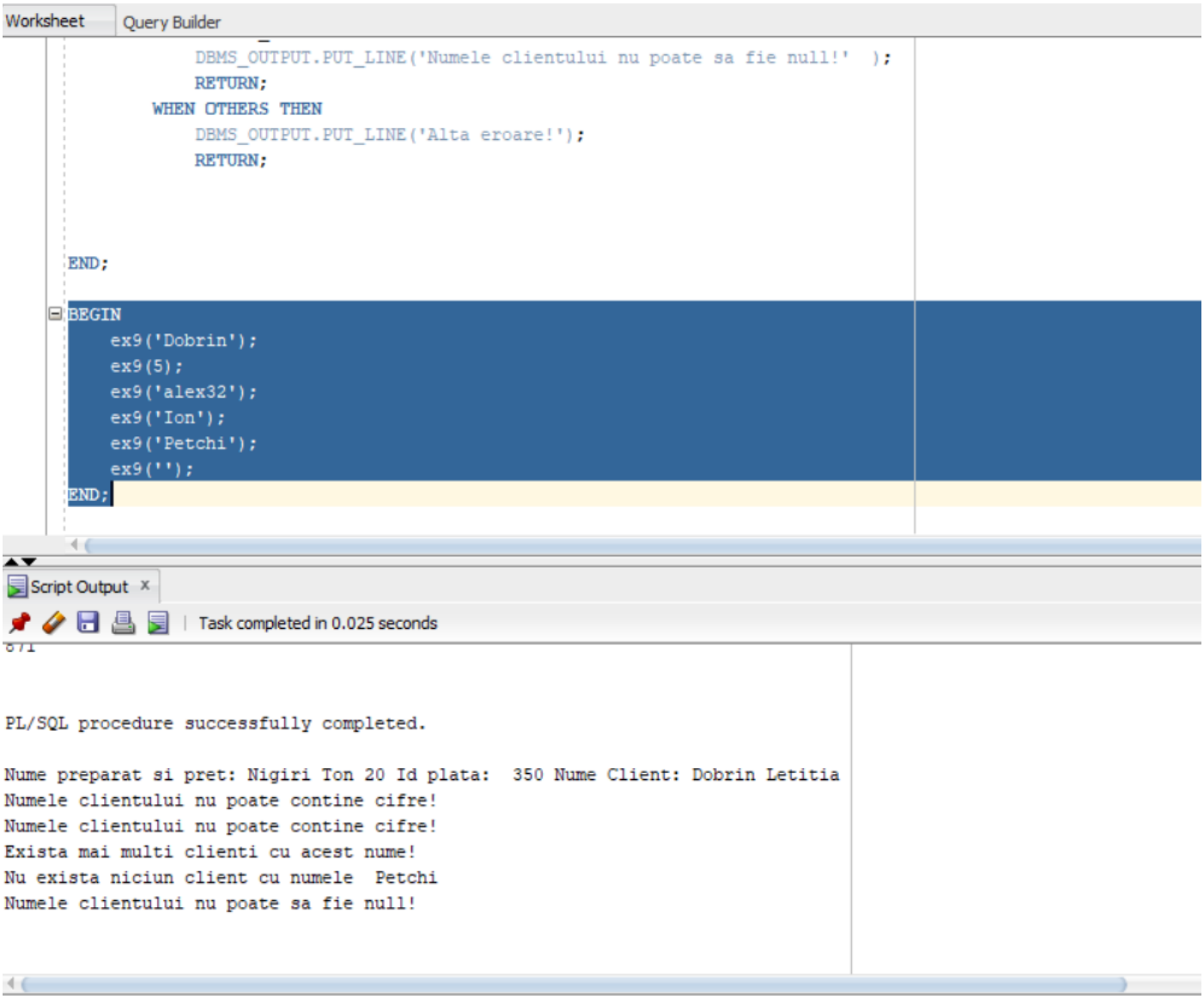
v_nume_preparat PREPARAT.NUME_PREPARAT%TYPE;

v_pret_preparat PREPARAT.PRET%TYPE;

```
    v_id_plata PLATA.ID_PLATA%TYPE;
    v_nume_client CLIENT.NUME_CLIENT%TYPE;
    v_init CLIENT.NUME_CLIENT%TYPE := 'gol';
    final_nume_preparat PREPARAT.NUME_PREPARAT%TYPE;
    final_pret_preparat PREPARAT.PRET%TYPE := 0;
    final_id_plata PLATA.ID_PLATA%TYPE;
    final_nume_client CLIENT.NUME_CLIENT%TYPE;

    CURSOR c_items IS
       SELECT pr.NUME_PREPARAT,pr.PRET,p.ID_PLATA,cl.NUME_CLIENT
       FROM CLIENT cl
       JOIN COMANDA co ON co.id_client = cl.id_client
       JOIN PLATA p ON p.id_comanda=co.id_comanda
       JOIN ITEM_COMANDA i ON i.id_comanda=co.id_comanda
       JOIN PREPARAT pr ON pr.ID_PREPARAT=i.id_preparat
       WHERE UPPER(cl.NUME_CLIENT) LIKE '%' || UPPER(p_nume_client) || '%';

    INPUT_INVALID EXCEPTION;
    WRONG_TYPE EXCEPTION;
BEGIN
    IF regexp_like (p_nume_client,'[0-9]') THEN
       RAISE WRONG_TYPE;
    END IF;
    IF p_nume_client IS NULL THEN
       RAISE INPUT_INVALID;

    END IF;
    OPEN c_items;
    LOOP
```

```
      FETCH c_items INTO v_nume_preparat,v_pret_preparat,v_id_plata,v_nume_client;
      EXIT WHEN c_items%NOTFOUND;
      if v_init !=v_nume_client and v_init!='gol' then
         raise TOO_MANY_ROWS;
      END IF;

      v_init:= v_nume_client;
      IF(v_pret_preparat > final_pret_preparat) THEN
         final_nume_preparat := v_nume_preparat;
         final_pret_preparat :=v_pret_preparat;
         final_id_plata :=v_id_plata;
         final_nume_client :=v_nume_client;
      END IF;
   END LOOP;
   IF v_init = 'gol' then
      RAISE NO_DATA_FOUND;
   END IF;
   DBMS_OUTPUT.PUT_LINE( ' Nume preparat si pret: '||final_nume_preparat || ' ' ||
final_pret_preparat|| ' Id plata:  ' || final_id_plata || ' Nume Client: ' ||final_nume_client);
   EXCEPTION
      WHEN TOO_MANY_ROWS THEN
         DBMS_OUTPUT.PUT_LINE('Exista mai multi clienti cu acest nume!');
         RETURN;
      WHEN WRONG_TYPE THEN
         DBMS_OUTPUT.PUT_LINE('Numele clientului nu poate contine cifre!');
         RETURN;
      WHEN NO_DATA_FOUND THEN
         DBMS_OUTPUT.PUT_LINE('Nu exista niciun client cu numele  '|| p_nume_client );
         RETURN;
```

```
    WHEN INPUT_INVALID THEN
      DBMS_OUTPUT.PUT_LINE('Numele clientului nu poate sa fie null!'  );
      RETURN;
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Alta eroare!');
      RETURN;



END;
```

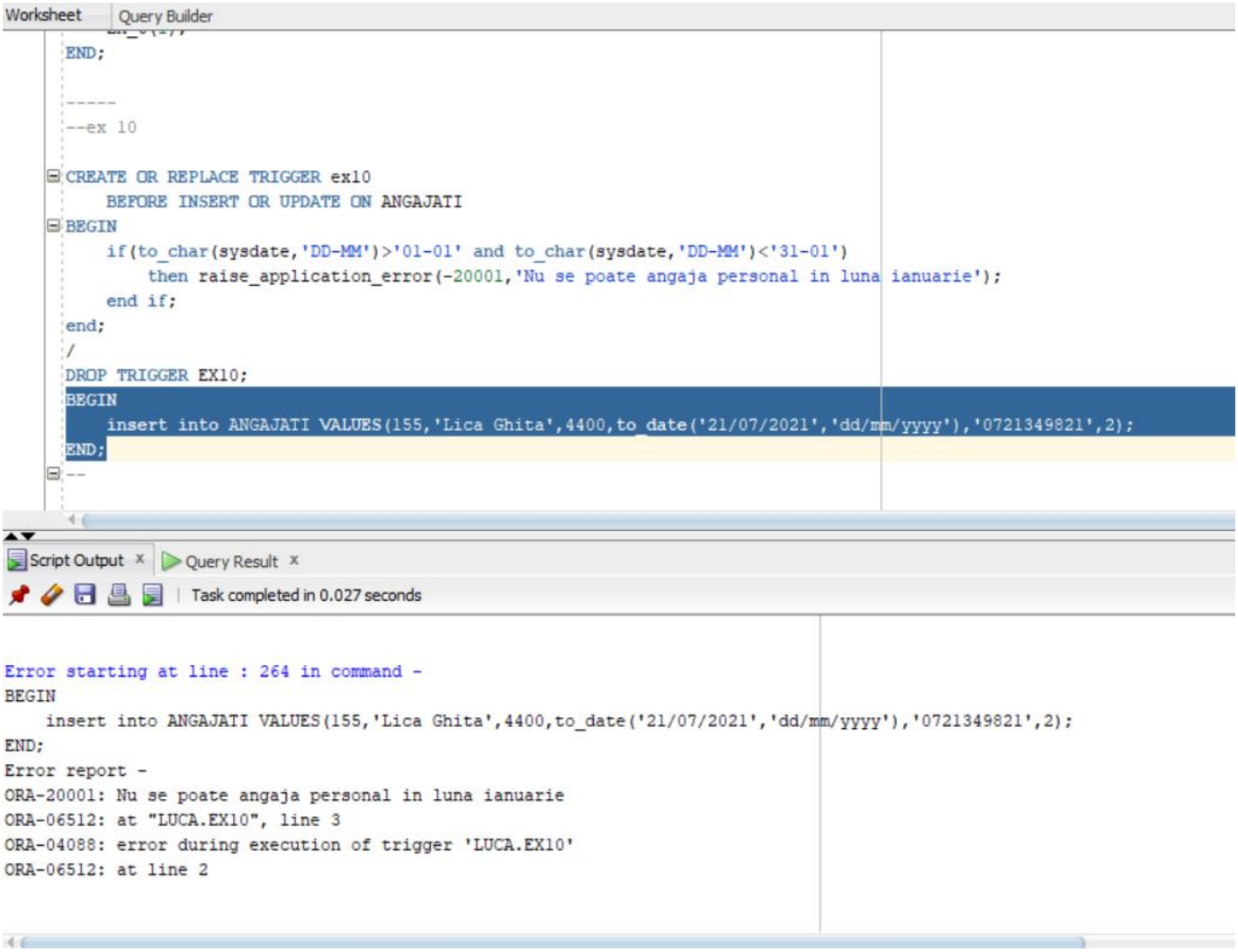


10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

Creati un trigger care interzice angajarea personalului in luna ianuarie.

```
CREATE OR REPLACE TRIGGER ex10
   BEFORE INSERT OR UPDATE ON ANGAJATI
BEGIN
   if(to_char(sysdate,'DD-MM')>'01-01' and to_char(sysdate,'DD-MM')<'31-01')
      then raise_application_error(-20001,'Nu se poate angaja personal in luna ianuarie');
```

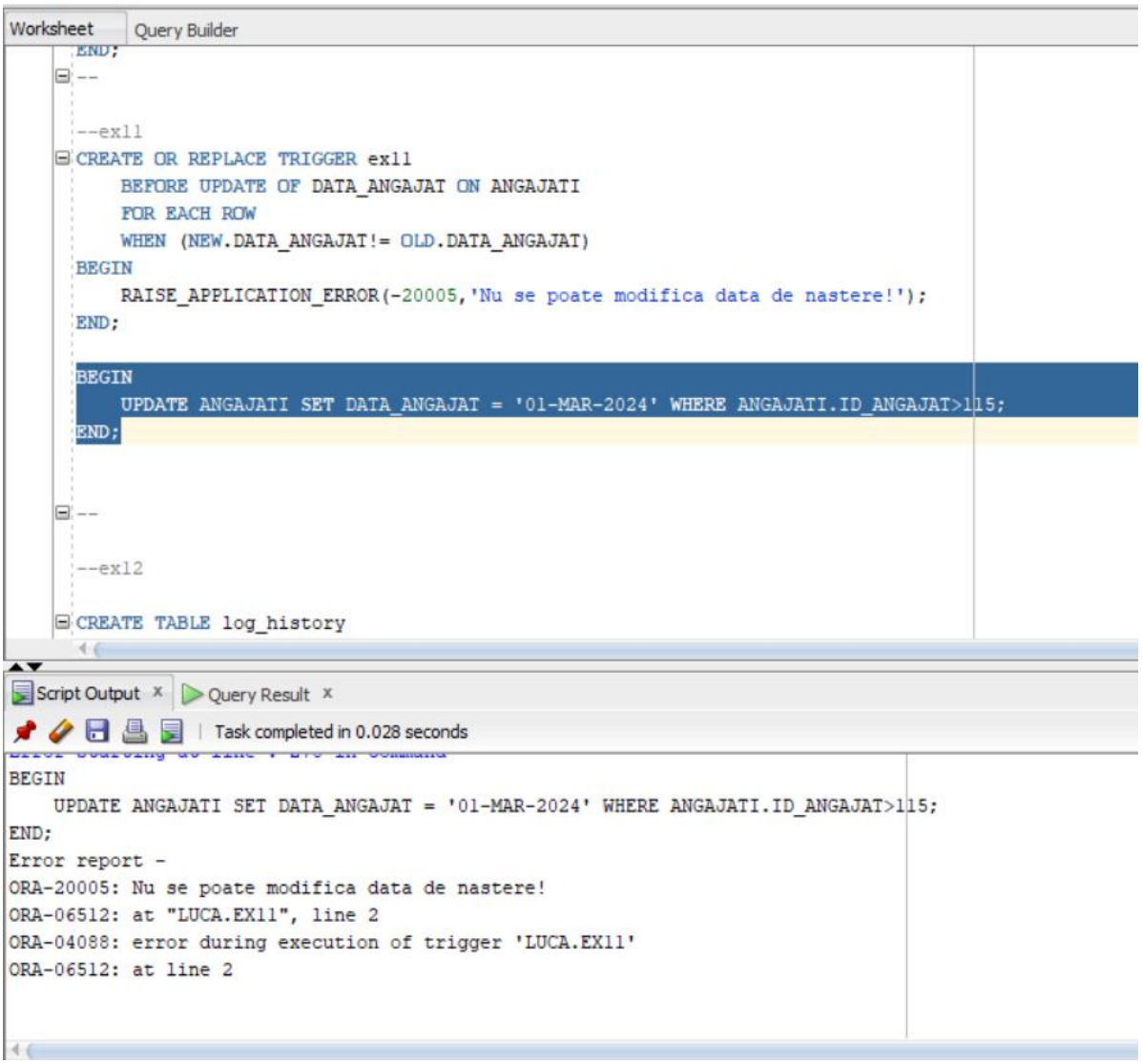
```
    end if;
end;
/
DROP TRIGGER EX10;
BEGIN
    insert into ANGAJATI VALUES(155,'Lica
Ghita',4400,to_date('21/07/2021','dd/mm/yyyy'),'0721349821',2);
END;
```



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

Creati un trigger care interzice modificarea datelor de nastere a angajatilor

```
CREATE OR REPLACE TRIGGER ex11
   BEFORE UPDATE OF DATA_ANGAJAT ON ANGAJATI
   FOR EACH ROW
   WHEN (NEW.DATA_ANGAJAT!= OLD.DATA_ANGAJAT)
BEGIN
   RAISE_APPLICATION_ERROR(-20005,'Nu se poate modifica data de nastere!');
END;


BEGIN
   UPDATE ANGAJATI SET DATA_ANGAJAT = '01-MAR-2024' WHERE
ANGAJATI.ID_ANGAJAT>115;
END;
```



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

Creati un trigger care sa permita modificarea schemei doar de utilizatorul Luca. Sa se salveze modificarile facute asupra schemei.

```
CREATE TABLE log_history
(
   username VARCHAR2(20),
   log_date DATE,
   event varchar2(100)
```

```
);

CREATE OR REPLACE TRIGGER ex12
   AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
   IF USER != 'Paul' THEN
      RAISE_APPLICATION_ERROR(-20910,'Numai administratorul bazei de date poate
realiza operatia');
   END IF;
   INSERT INTO log_history VALUES(SYS.LOGIN_USER,SYSDATE,SYS.SYSEVENT);
END;
/
DROP TRIGGER EX12;

CREATE TABLE testt(id NUMBER);
SELECT * FROM log_history;
```

```
    log_date DATE,
    event varchar2(100)
);

CREATE OR REPLACE TRIGGER ex12
    AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
    IF USER != 'Luca' THEN
        RAISE_APPLICATION_ERROR(-20910,'Numai administratorul bazei de date poate realiza operatia');
    END IF;
    INSERT INTO log_history VALUES(SYS.LOGIN_USER,SYSDATE,SYS.SYSEVENT);
END;
/
DROP TRIGGER EX12;

CREATE TABLE testt(id NUMBER);
SELECT * FROM log_history;
```

Script Output | Query Result

SQL | All Rows Fetched: 25 in 0.02 seconds

	USERNAME	LOG_DATE	EVENT
15	LUCA	12-JAN-24	CREATE
16	LUCA	12-JAN-24	CREATE
17	LUCA	12-JAN-24	CREATE
18	LUCA	12-JAN-24	CREATE
19	LUCA	12-JAN-24	CREATE
20	LUCA	12-JAN-24	CREATE
21	LUCA	12-JAN-24	CREATE
22	LUCA	12-JAN-24	CREATE
23	LUCA	12-JAN-24	CREATE
24	LUCA	12-JAN-24	CREATE
25	LUCA	12-JAN-24	CREATE

```
CREATE OR REPLACE TRIGGER ex12
    AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
    IF USER != 'Paul' THEN
        RAISE_APPLICATION_ERROR(-20910,'Numai administratorul bazei de date poate realiza operatia');
    END IF;
    INSERT INTO log_history VALUES(SYS.LOGIN_USER,SYSDATE,SYS.SYSEVENT);
END;
/
DROP TRIGGER EX12;

CREATE TABLE testt(id NUMBER);
SELECT * FROM log_history;
```

Script Output | Query Result

Task completed in 0.026 seconds

```
Error starting at line : 302 in command -
CREATE TABLE testt(id NUMBER)
Error report -
ORA-00604: error occurred at recursive SQL level 1
ORA-20910: Numai administratorul bazei de date poate realiza operatia
ORA-06512: at line 3
00604. 00000 -  "error occurred at recursive SQL level %s"
*Cause:    An error occurred while processing a recursive SQL statement
           (a statement applying to internal dictionary tables).
*Action:   If the situation described in the next error on the stack
           can be corrected, do so; otherwise contact Oracle Support.
```