Descrierea modelului real, utilitatea acestuia si regulile de functionare

Acest proiect consta in implementarea unei baze de date pentru a fi folosita de o multinationala in domeniul retail-ului. Astfel, baza de date permite eficientizarea proceselor zilnice din cadrul companiei si gestiunea eficienta a magazinelor, angajatilor, vanzarilor, produselor si furnizorilor.

Nucleul bazei de date se afla in legaturile dintre magazine, produse si furnizori, oferind o mai buna evidenta a acestora si posibilitati de interogari rapide in legatura cu domenii relevante: venitul intr-o perioada de timp, numarul angajatilor per magazin/locatie/departament/oras/tara, evidenta vechimii angajatilor, analiza rentabilitatii unui magazin individual si calculul profitului companiei intr-un interval de timp, etc.

Regulile de funcționare:

-furnizorii ofera multiple produse mai multor magazine

-fiecare produs se incadreaza intr-o singura categorie de produse si se poate comercializa en-gross sau en-detail

-intr-o tara se afla mai multe orase cu mai multe adrese la care se afla magazine

-un angajat apartine de un singur departament si un singur magazin

-companiile de mentenanta ofera servicii de securitate si curatenie magazinelor pentru care lucreaza

-toti angajatii unui departament au acelasi salariu de baza, peste care se adauga un eventual comision individual

Constrangeri

Pentru a putea fi operational, modelul respecta urmatoarele:

-fiecarei adrese ii corespunde exact un magazin si viceversa

-un angajat lucreaza la un singur departament si la un singur magazin

-un bon se realizeaza la un singur magazin

-un produs are un unic furnizor si o singura categorie din care face parte

-un produs nu poate costa mai mult de 999.9 euro

-se pot cumpara cel mult 99 de produse de acelasi fel pe un singur bon sau 99 de kilograme en-gros

-un furnizor apartine unui singur stat deorece se presupune ca nu aducem produse prin intermediari, ci se procura de la producatori locali)

-un magazin trebuie sa aiba exact o firma de curatenie si o firma de securitate

-o adresa se afla intr-un singur oras si un oras se afla intr-o singura tara

Entitati

Toate entitile sunt independente, mai putin cele doua subentititati ale companiilor de mentenanta, ci anume Security si Cleaning Companies

COUNTRIES: tarile, descrise prin tag/abreviere si numele lor reprezinta atat originea produselor cat si componenta indirecta a locatiei magazinelor. Cheia primara este state_tag

CITIES: orasele ce fac parte din tari, descrise prin tag/abreviere si reprezinta o componenta indirecta a locatiei complete a magazinelor. Cheia primara este city_tag

ADDRESSES: adresele magazinelor, identificate unic prin cheia primara address_code. Ele reprezinta locatia intr-un oras a unui magazin, retinand orasul, codul postal flexibil la variatiuni internationale, strada si numarul.

MARKETS: magazinele propriu-zise, identificate prin cheia primara id_market. Despre un magazin in particular se retin suprafata sa, codul adresei sale si codurile companiilor responsabile de curatenie si paza.

MAINTAINANCE_COMPANIES: companiile responsabile pentru paza si curatenia magazinelor. Cheia primara este id_comp si se retine data de inceput a contractului multinationalei cu o firma in particular pentru a putea identifica mai bine partenerii de incredere si loiali. Se retine si suma pe care acestea o revendica lunar, exprimata in euro si tipul companiei(curatenie/paza).

SECURITY: subentitate a MAINTAINANCE_COMPANIES caracterizata de tipul companiei ca fiind de paza si retine pentru o companie de paza specializarea acesteia

SECURITY: subentitate a MAINTAINANCE_COMPANIES caracterizata de tipul companiei ca fiind de curatenie si retine pentru o companie de curatenie frecventa de curatenie pe care aceasta se angajeaza sa o respecte, exprimata in zile.

EMPLOYEES: toti angajatii ce lucreaza pentru multinationala data, identificati prin cheia primara id_emp. Despre acestia se retine numele, prenumele, data angajarii, numarul de telefon si codurile pentru departamentul si magazinul din care fac parte.

DEPARTMENTS: departamentele multinationalei, ramurile pe care se proiecteaza activitatea acesteia. Cheia primara este id_depart si se retin numele departamentului si salariul de baza aferent(model)

ABILITIES: abilitatile pe care le poseda si de care au nevoie angajatii pentru a face fata situatiilor postului lor. Cheia primara este ability_tag si se retin numele abilitatii si durata in saptamani a training-urilor oferite de companie angajatilor pentru a le dezvolta abilitatea in cauza

LANGUAGES: Limbile cunoscute de angajati, identificate unic prin cheia primara lang_tag si se retine numele limbii.

RECEIPTS: bonurile de la casa de marcat, cheie primara receipt_id si retin data tranzactiei si magazinul

PRODUCT_TYPES: categoriile de produse comercializate si distribuite de furnizori, cheie primara prod_type_tag si se retine si numele produsului.

SUPPLIERS: furnizorii magazinelor, cheie primara id_supplier. Se retine numele firmei sau individului si tara de origine a marfurilor lor.

PRODUCTS: produsele ce se gasesc pe rafturile magazinelor, identificate prin cheia primara id_prod. Se retine denumirea uzuala a produsului, producatorul acestuia, categoria de produse din care face parte, tipul de vanzare(eng sau end) si pretul corespunzator unui kilogram de produs in cazul vanzarii en-gross sau a unei instante individuale in cazul en-detail

Relatii

COUNTRIES_have_CITIES: cardinalitate minima 1:0, maxima 1:n

CITIES_have_ADDRESSS: cardinalitate minima 1:1, maxima 1:n – un oras poate include mai multe adrese dar ar trebui sa includa cel putin una pentru a avea rost memorarea lui in baza de date, dar o adresa apartine de exact un oras

MARKETS_are_located_at_ADDRESSS: cardinalitate 1:1 constanta, deoarece la orice adresa se afla un singur magazin si viceversa

MARKETS_have_EMPLOYEES: cardinalitate minima 1:1, maxima 1:n – un magazin are mai multi angajati si pe termen lung ar trebui sa fie macar un angajat per magazin

DEPARTMENTS_have_EMPLOYEES: cardinalitate minima 1:1, maxima 1:n – un departament are mai multi angajati si pe termen lung ar trebui sa fie macar un angajat per departament

EMPLOYEES_know_LANGAUAGES_and_ABILITIES: cardinalitatea minima 0:0:0 si maxima n:m:p

- angajatii detin cunostinte multiple cunostinte in materie de limbi straine si abilitati necesare la locul de munca si eventual 0

MAINTAINANCE_COMPANIES(SECURITY)_protect_MARKETS: cardinalitate minima 1:1, maxima 1:n – o firma de paza poate asigura paza mai multor magazine, dar un magazin are doar o firma de paza cu care colaboreaza

MAINTAINANCE_COMPANIES(CLEANING)_clean_MARKETS: cardinalitate minima 1:1, maxima 1:n – situatie identica(cazul companiei de securitate)

SUPPLIERS_supply_PRODUCTS_to_MARKETS: relatie ternara, cardinalitatea minima 0:0:0 si maxima n:m:p – un singur furnizor furnizeaza un anumit produs la mai multe magazine, eventual 0, un singur produs este furnizat la un anumit magazin de mai multi furnizori, eventual 0 si un singur furnizor furnizeaza unui magazin specific mai multe produse, eventual 0

MARKETS_produce_RECEIPTS : cardinalitate minima 1:0, maxima 1:n – un bon este emis de un singur magazin(constrangeri) dar un magazin emite mai multe bonuri si eventual 0 daca nu are clienti

SUPPLIERS_is_from_COUNTRIES: cardinalitate minima 1:0, maxima 1:n - mai multi furnizori pot proveni din acelasi stat, insa un furnizor aprovizioneaza magazinul cu produse ce provin dintr-un singur stat, deoarece sunt ptoduse local conform modelului si constrangerilor)

RECEIPTS_contain(quantity)_PRODUCTS: cardinalitate n:m – lasam many to many deorece vrem sa stim pentru fiecare bon ce produse contine, insa fara a incalca FN1, asa ca recurgem la un tabel asociativ descris la categoria FN3)

SUPPLIERS_provide_PRODUCT_TYPES: minim 1:1, maxim n:m – lasam many to many pentru a permite ca o categorie de produse sa aiba mai multi furnizori(principiu esential de economie pentru a preveni un monopol) dar pentru a permite si unui furnizor sa ofere mai multe categorii de produse(o singura ferma poate asigura si fructe si legume)

Atribute

Pentru entitati:

PRODUCT_TYPES

Prod_type_tag# varchar(3)   codul categoriei de produs(‘VEG’ ,’MEA’,’CSM’,...)

Prod_name varchar(20) numele categoriei de produse(vegetables,meat,cosmetics,...)

SUPPLIERS

Id_supplier# number(3) cheia primara

Name varchar(20) numele furnziorului sau companiei, in sql se va nota name_s din cauza conflictului semantic cu ‘name’ stabilit de sistem

Org_state varchar(3)   codul tarii de origine pentru toate produsele furnizorului, foreign key din COUNTRIES

MARKETS

Id_market# number(4) cheia primara

Address_code number(4) codul adresei magazinului, foreign key din ADDRESSES

Surface number(4) suprafata totala a magazinului

Id_comp_sec number(3) codul firmei de paza, foreign key din MAINTAINANCE_COMPANIES

Id_comp_cle number(3) analog, codul firmei de curatenie

COUNTRIES

State_tag #varchar(3) codul tarii(‘ROM’,’TUR’,’USA’,...), cheia primara

State_name varchar(30) numele uzual al tarii (Romania,Turcia, Statele Unite ale Americii,...)

CITIES

City_tag# varchar(3) codul orasului(‘BUC’,’MAD’,’PAR’,...) cheia primara

City_name varchar(20) denumirea orasului(Bucuresti,madrid,paris,...)

State_tag varchar(30) codul tarii in care se afla orasul, foreign key din COUNTRIES

ADDRESSES

Address_code# number(4) cheia primara

City_tag varchar(3) codul orasului in care se gaseste adresa, foreign key din CITIES

Postal_code varchar(15) codul postal al adresei, rol in acuratetea localizarii

Street_name varchar(30) strada pe care se afla o adresa

Number number(3) numarul la care se afla o adresa

DEPARTMENTS

Id_depart# number(3) cheie primara

Departmentg_name varchar(20) denumirea departamentului

Base_salary number(5) salariul de baza pentru fiecare departament(model) exprimat in euro

MAINTAINANCE_COMPANIES

Id_comp# number (3) cheie primara

Company_name varchar(5) denumirea companiei

Collab_start_date date data inceperii contractului cu o firma partenera

Monthly_payment number(5) plata lunara pe care o companie partenera o primeste pentru prestarea serviciilor

Company_type varchar(12) ia valorile 'Cleaning' sau 'Security' pentru a determina subentitatile

Main_specialisation varchar(20) se refera doar la firmele de securitate, pentru cele de curatenie ia valoarea null si reprezinta specializarea firmei de securitate(antifurt,antitero,siguranta clientilor, a angajatilor, prevenirea evaziunii fiscale prin supravegherea nagjatilor etc)

Cleaning_frquency number(2) se refera doar la firmele de curatenie, pentru cele de securitate ia valoarea null si reprezinta o data la cate zile o companie de curatenie este obligata conform contractului sa curete un magazin

EMPLOYEES

Id_emp# number(4) cheia primara

Last_name varchar(20) numele de familie al unui angajat

First_name varchar(20) prenumele unui angajat

Hire_date date data angajarii unui angajat

Phone varchar(10) telefonul unui angajat, camp unic

Commission_quoef number (3,2) reprezinta ponderea cresterii salariului de baza (ex: un angajat care lucreaza la departamentul Bakery are un salariu de baza de 2200 euro insa daca are si un commission_quoef de 0.1 atunci va castiga cu 10% in plus fata de salariul de baza, ajungand cu un salariu efectiv de 2420 de euro)

Id_depart number(3) departamentul la care lucreaza un angajat, foreign key din DEPARTMENTS

Id_market number(4) magazinul la care lucreaza un angajat, foreign key din MARKETS

ABILITIES

Ability_tag varchar# (3) tag/abreviere pentru abilitate (‘TMW’,’TMM’,’LEA’,...) , cheia primara

Ability_name varchar(20) denumirea abilitatii (teamwork,time management, leadership,...)

Training_duration number(1) durata in saptamani a cursurilor si activitatilor dedicate dezvoltarii unei abilitati specifice

LANGUAGES

Lang_tag# varchar(3) tag/abreviere pentru limba(‘ROM’,’BLG’,’FRE’,...), cheie primara

Lang_name varchar(20) denumirea limbii(Romanian,Bulgarian,French,...)

PRODUCTS

Id_prod# number(3) cheie primara,

Prod_name varchar(20) denumirea generica a produsului(doar ‘sampon’, nu ‘sampon Elseve’)

Prod_type varchar(3) codul categoriei produsului, foreign key din PRODUCT_TYPES

Sell_type varchar(3) tipul de comercializare pentru un produs: la kilogram(‘eng’) sau la bucata(‘end’)

Price number(4,1) pretul produsului, un numar de cel mult 3 cifre si o zecimala(maximul de 999.9 din constrangeri), acesta reprezinta pretul unui kilogram de produs pentru cele cu vanzare en-gros si pretul unui obiect individual pentru produsele cu vanzare en-detail

RECEIPTS

Receipt_id# number(8) cheie primara

Transaction_date date data tranzactiei

Id_market number(4) magazinul la care s-a facut tranzactia, foreign key din MARKETS

Pentru relatii:

PROVIDE

prod_type_tag# varchar(3) componenta a cheii primare compuse, foreign key din PRODUCT_TYPES

id_supplier# number(3) componenta a cheii primare compuse,foreign key din SUPPLIERS

CONTAIN

Id_prod# number(3) componenta a cheii primare compuse, foreign key din PRODUCTS

Receipt_id# number(8) componenta a cheii primare compuse, foreign key din RECEIPTS

Quantity number(4,2) cantitatea cumparata poate fi numar intreg pentru en-detail sau numar real cu doua zecimale pentru en-gros

SUPPLY

Id_supplier# number(3) componenta a cheii primare compuse, foreign key din SUPPLIERS

Id_prod# number(3) componenta a cheii primare compuse, foreign key din PRODUCTS

Id_market# number(4) componenta a cheii primare compuse, foreign key din MARKETS

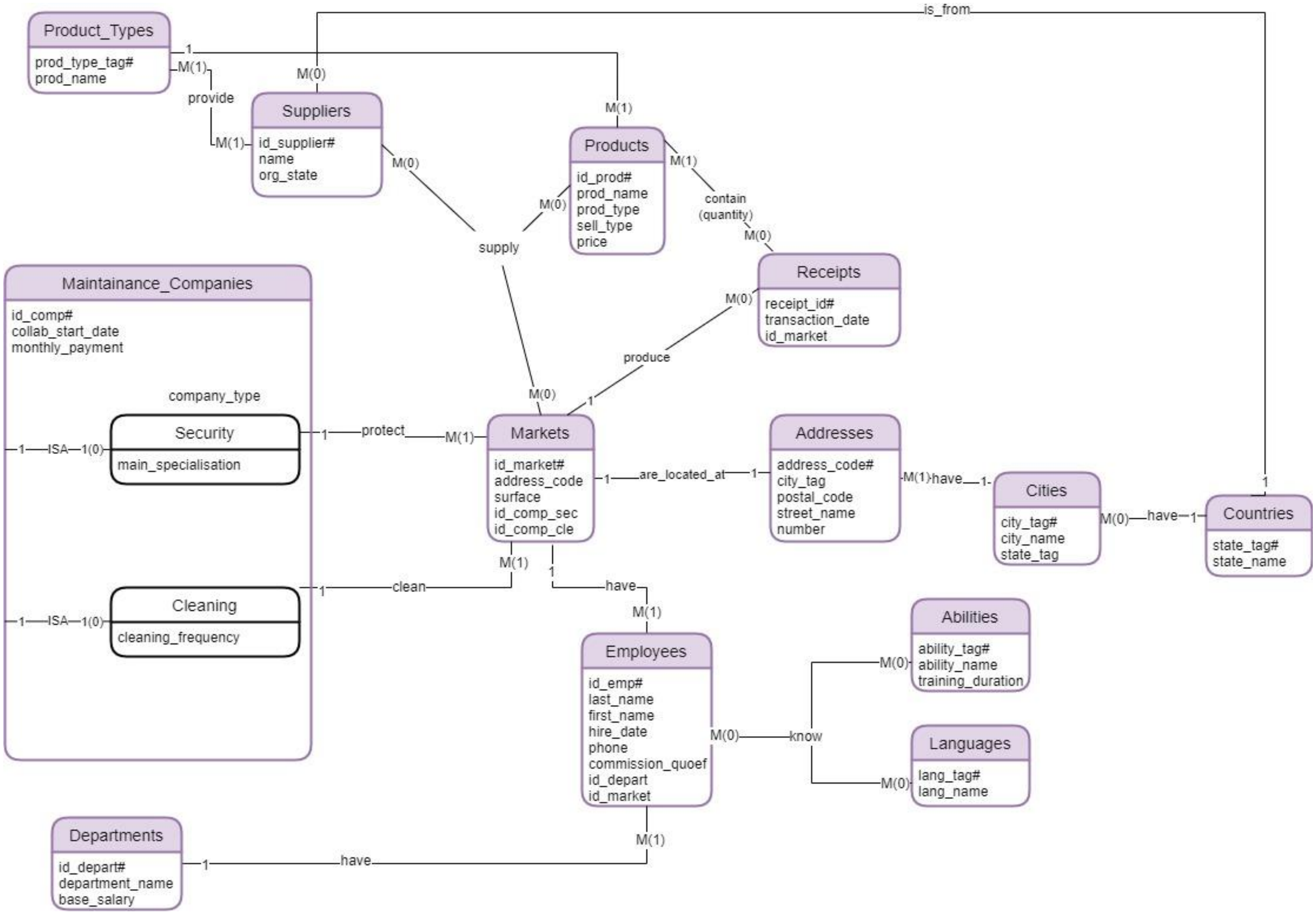
KNOW

id_emp# number(4) componenta a cheii primare compuse, foreign key din EMPLOYEES

ability_tag# varchar(3) componenta a cheii primare compuse, foreign key din ABILITIES

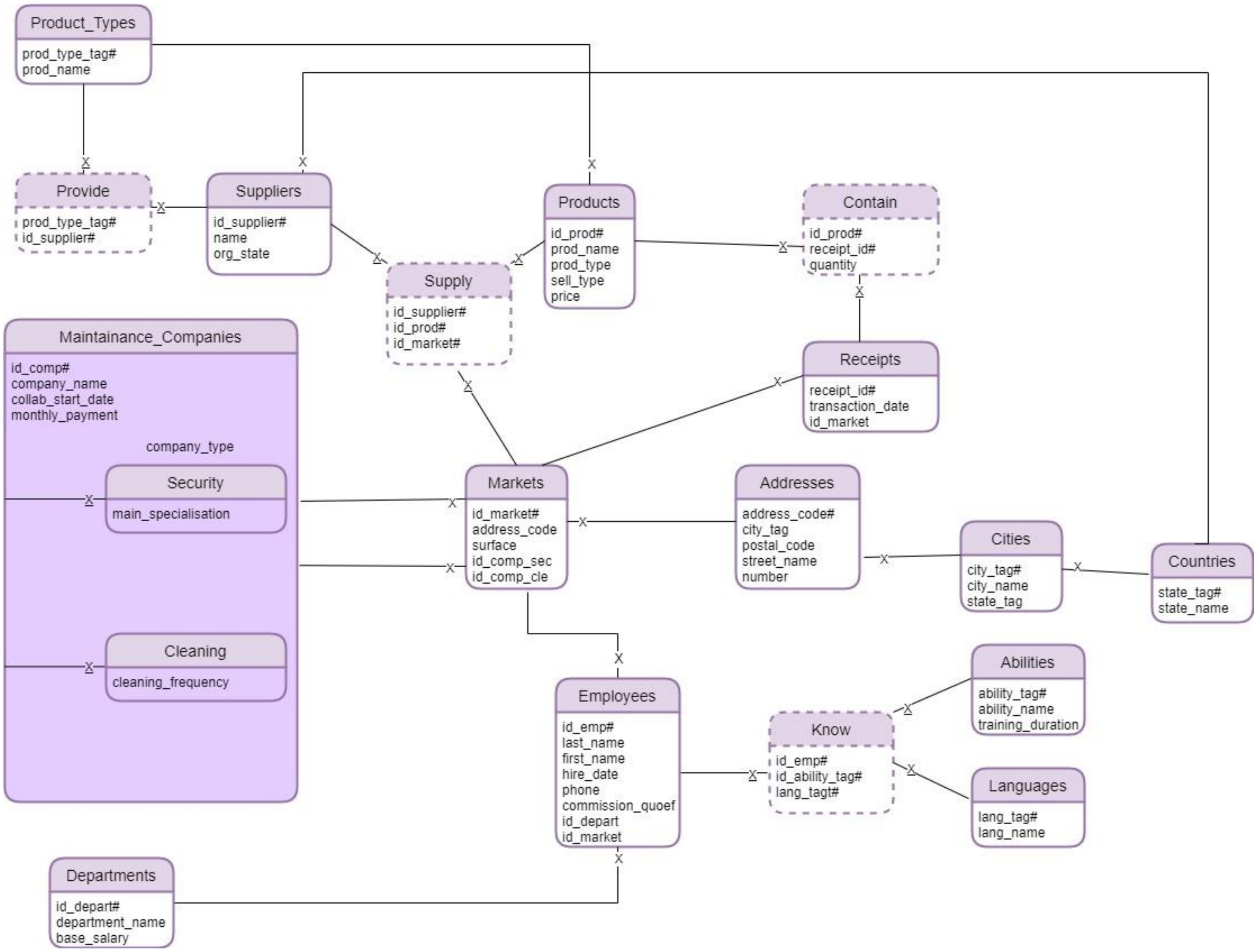
lang_tag# varchar(3) componenta a cheii primare compuse, foreign key din LANGUAGES

ERD



Product_Types
prod_type_tag#
prod_name
1
M(1)
provide
M(1)
is_from
M(0)
Suppliers
id_supplier#
name
org_state
M(0)
M(1)
Products
id_prod#
prod_name
prod_type
sell_type
price
M(0)
M(1)
contain
(quantity)
M(0)
supply
Receipts
receipt_id#
transaction_date
id_market
M(0)
produce
1
M(0)
Maintainance_Companies
id_comp#
collab_start_date
monthly_payment
company_type
Security
main_specialisation
1
ISA
1(0)
1
protect
M(1)
Markets
id_market#
address_code
surface
id_comp_sec
id_comp_cle
1
are_located_at
1
Addresses
address_code#
city_tag
postal_code
street_name
number
M(1)
have
1
Cities
city_tag#
city_name
state_tag
M(0)
have
1
Countries
state_tag#
state_name
1
Cleaning
cleaning_frequency
1
ISA
1(0)
1
clean
M(1)
1
have
M(1)
Employees
id_emp#
last_name
first_name
hire_date
phone
commission_quoef
id_depart
id_market
M(0)
know
M(0)
Abilities
ability_tag#
ability_name
training_duration
M(0)
Languages
lang_tag#
lang_name
M(1)
Departments
id_depart#
department_name
base_salary
1
have

Diagrama Conceptuala



Schemele Relationale

Din diagrama conceptuala se obtin urmatoarele scheme relationale:

COUNTRIES(state_tag#,state_name)

CITIES(city_tag#,city_name,state_tag)

ADDRESSES(address_code#,city_tag,postal_code,street_name,number)

MARKETS(id_market#,address_code,surface,id_comp_sec,id_comp_cle)

EMPLOYEES(id_emp#,last_name,first_name,hire_date,phone,commission_quoef,id_depart,id_market)

DEPARTMENTS(id_depart#,departmentg_name,base_salary)

ABILITIES(ability_tag#,ability_name,training_duration)

LANGUAGES(lang_tag#,lang_name)

MAINTAINANCE_COMPANIES(id_comp#,company_name,collab_start_date,monthly_payment, company_type,main_specialisation,cleaning_frequency)

PRODUCT_TYPES(prod_type_tag#,prod_name)

SUPPLIERS(id_supplier#,name,org_state)

PRODUCTS(id_prod#,prod_name,prod_type,sell_type,price)

RECEIPTS(receipt_id#,transaction_date,id_market)

SUPPLY(id_supplier#,id_prod#,id_market#)

PROVIDE(prod_type_tag#,id_supplier#)

CONTAIN(id_prod#,receipt_id#,quantity)

KNOW(id_emp#,ability_tag#,lang_tag#)

FN3

Diagrama se afla in forma normala 1 deoarece nu are atribute cu valori multiple

COD_FURNIZOR#	CODURI_PRODUSE#
100	101,102
101	100,101,102

In exemplul de mai jos avem transformarea corecta a tabelului de mai sus in FN1, tabel regasit in diagrama conceptuala si implementarea in SQL:

COD_FURNZIOR#	COD_PRODUS#
100	101
100	102
101	100
101	101
101	102

Diagrama se afla in FN2 deoarece nu exista atribute care depind de doar un atribut din cheia primara multipla. O singura tabela are minim doua chei primare si un atribut non-cheie, ci anume contine(din diagrama conceptuala). Cantitatea cumparata depinde integral de tranzactia data, adica id-ul bonului, si produsul de pe el, e imposibil sa deducem cantitatea doar din una dintre ele, deci se respecta fn2.

NR_BON#	COD_PRODUS#	CANTITATE
100	250	3
100	370	2
100	200	4
100	240	1

Un exemplu de non-FN2 ar fi fost urmatorul, deoarece denumirea produsului depinde doar de componenta cod_produs din cheia primara

NR_BON#	COD_PRODUS#	DENUMIRE_PRODUS
100	250	sapun
100	370	cartofi
100	200	carne de vita
100	240	sampon

Diagrama se afla in FN3 deoarece nu exista dependente tranzitive intre atributele care nu fac parte din cheia primara. De exemplu, daca tabela de adrese ar fi aratat in felul urmator aceasta nu ar mai fi fost in FN3:

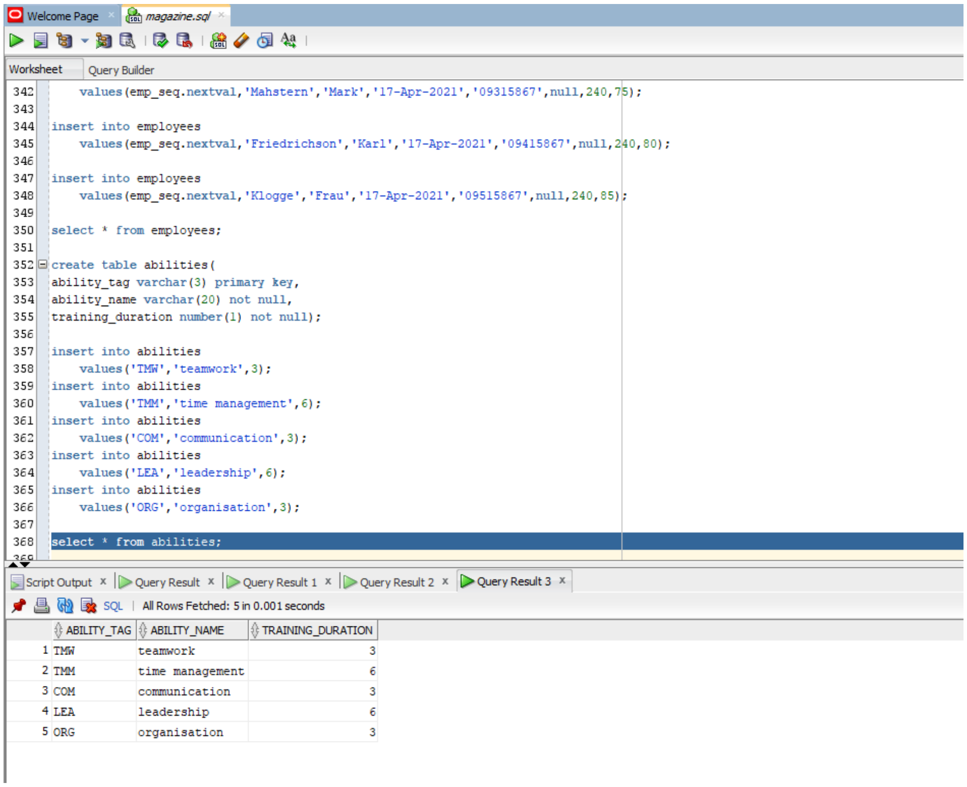
COD_ADRESA#	COD_ORAS	NUME_ORAS	COD_POSTAL	STRADA	NR	COD_STAT
100	200	Madrid	ZA56B3	Pedro de Valdivia	47	SPN
101	200	Madrid	ZA67C9	Carrer de Borrian	5	SPN

Se observa ca atributul cod_oras care nu este cheie primara determina atributele nume_oras si cod_stat. Aplicand regula Cassey-Delobel, tabela se imparte in doua, obtinand o tabela noua in care cod_oras este cheie primara pentru atributele pe care le deteremina si cea originala din care eliminam nume_oras si cod_stat, atributele dependente de cod_oras si pastram cod_oras drept foreign key din noua lui tabela si se obtine impartirea din diagrama:

COD_ADRESA#	COD_ORAS	COD_POSTAL	STRADA	NR
100	200	ZA56B3	Pedro de Valdivia	47
101	200	ZA67C9	Carrer de Borrian	5

COD_ORAS	NUME_ORAS	COD_STAT
200	Madrid	SPN

Crearea tabelelor in sql + secvente



```
    values(emp_seq.nextval,'Mahstern','Mark','17-Apr-2021','09315867',null,240,75);

insert into employees
    values(emp_seq.nextval,'Friedrichson','Karl','17-Apr-2021','09415867',null,240,80);

insert into employees
    values(emp_seq.nextval,'Klogge','Frau','17-Apr-2021','09515867',null,240,85);

select * from employees;

create table abilities(
ability_tag varchar(3) primary key,
ability_name varchar(20) not null,
training_duration number(1) not null);

insert into abilities
    values('TMW','teamwork',3);
insert into abilities
    values('TMM','time management',6);
insert into abilities
    values('COM','communication',3);
insert into abilities
    values('LEA','leadership',6);
insert into abilities
    values('ORG','organisation',3);

select * from abilities;
```

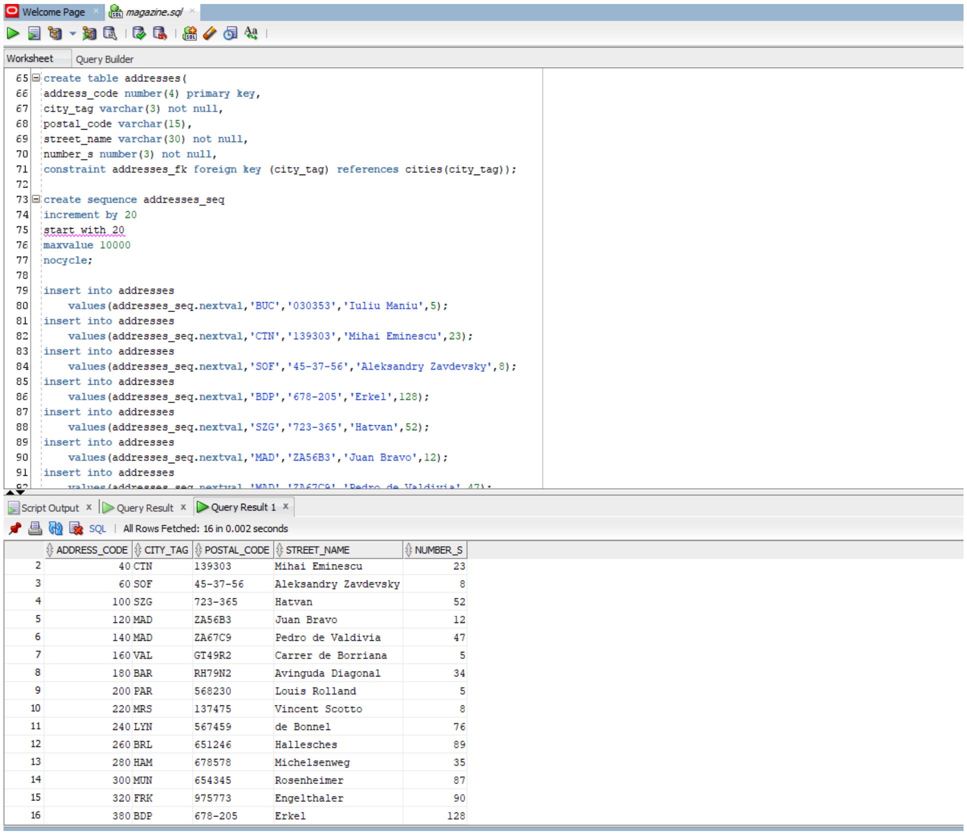
All Rows Fetched: 5 in 0.001 seconds

	ABILITY_TAG	ABILITY_NAME	TRAINING_DURATION
1	TMW	teamwork	3
2	TMM	time management	6
3	COM	communication	3
4	LEA	leadership	6
5	ORG	organisation	3

create table abilities(

ability_tag varchar(3) primary key,

```
ability_name varchar(20) not null,

training_duration number(1) not null);


insert into abilities

   values('TMW','teamwork',3);

insert into abilities

   values('TMM','time management',6);

insert into abilities

   values('COM','communication',3);

insert into abilities

   values('LEA','leadership',6);

insert into abilities

   values('ORG','organisation',3);


select * from abilities;
```



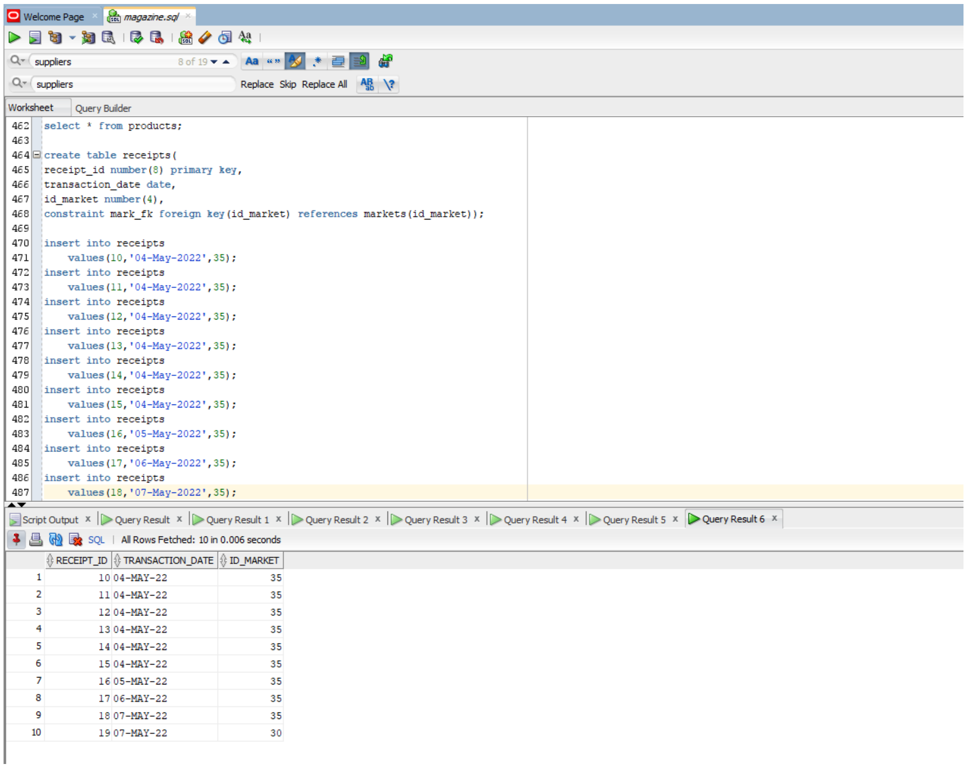
```
create table addresses(
address_code number(4) primary key,
city_tag varchar(3) not null,
postal_code varchar(15),
street_name varchar(30) not null,
number_s number(3) not null,
constraint addresses_fk foreign key (city_tag) references cities(city_tag));

create sequence addresses_seq
increment by 20
start with 20
maxvalue 10000
nocycle;

insert into addresses
    values(addresses_seq.nextval,'BUC','030353','Iuliu Maniu',5);
insert into addresses
    values(addresses_seq.nextval,'CTN','139303','Mihai Eminescu',23);
insert into addresses
    values(addresses_seq.nextval,'SOF','45-37-56','Aleksandry Zavdevsky',8);
insert into addresses
    values(addresses_seq.nextval,'BDP','678-205','Erkel',128);
insert into addresses
    values(addresses_seq.nextval,'SZG','723-365','Hatvan',52);
insert into addresses
    values(addresses_seq.nextval,'MAD','ZA56B3','Juan Bravo',12);
insert into addresses
    values(addresses_seq.nextval,'MAD','ZA67C9','Pedro de Valdivia',47);
```

All Rows Fetched: 16 in 0.002 seconds

	ADDRESS_CODE	CITY_TAG	POSTAL_CODE	STREET_NAME	NUMBER_S
2	40	CTN	139303	Mihai Eminescu	23
3	60	SOF	45-37-56	Aleksandry Zavdevsky	8
4	100	SZG	723-365	Hatvan	52
5	120	MAD	ZA56B3	Juan Bravo	12
6	140	MAD	ZA67C9	Pedro de Valdivia	47
7	160	VAL	GT49R2	Carrer de Borriana	5
8	180	BAR	RH79N2	Avinguda Diagonal	34
9	200	PAR	568230	Louis Rolland	5
10	220	MRS	137475	Vincent Scotto	8
11	240	LYN	567459	de Bonnel	76
12	260	BRL	651246	Hallesches	89
13	280	HAM	678578	Michelsenweg	35
14	300	MUN	654345	Rosenheimer	87
15	320	FRK	975773	Engelthaler	90
16	380	BDP	678-205	Erkel	128

```
create table addresses(
address_code number(4) primary key,
city_tag varchar(3) not null,
postal_code varchar(15),
street_name varchar(30) not null,
number_s number(3) not null,
constraint addresses_fk foreign key (city_tag) references cities(city_tag));

create sequence addresses_seq
increment by 20
start with 20
maxvalue 10000
nocycle;

insert into addresses
   values(addresses_seq.nextval,'BUC','030353','Iuliu Maniu',5);
insert into addresses
   values(addresses_seq.nextval,'CTN','139303','Mihai Eminescu',23);
insert into addresses
   values(addresses_seq.nextval,'SOF','45-37-56','Aleksandry Zavdevsky',8);
insert into addresses
   values(addresses_seq.nextval,'BDP','678-205','Erkel',128);
insert into addresses
   values(addresses_seq.nextval,'SZG','723-365','Hatvan',52);
insert into addresses
   values(addresses_seq.nextval,'MAD','ZA56B3','Juan Bravo',12);
insert into addresses
   values(addresses_seq.nextval,'MAD','ZA67C9','Pedro de Valdivia',47);
insert into addresses
```

```
   values(addresses_seq.nextval,'VAL','GT49R2','Carrer de Borriana',5);
insert into addresses
   values(addresses_seq.nextval,'BAR','RH79N2','Avinguda Diagonal',34);
insert into addresses
   values(addresses_seq.nextval,'PAR','568230','Louis Rolland',5);
insert into addresses
   values(addresses_seq.nextval,'MRS','137475','Vincent Scotto',8);
insert into addresses
   values(addresses_seq.nextval,'LYN','567459','de Bonnel',76);
insert into addresses
   values(addresses_seq.nextval,'BRL','651246','Hallesches',89);
insert into addresses
   values(addresses_seq.nextval,'HAM','678578','Michelsenweg',35);
insert into addresses
   values(addresses_seq.nextval,'MUN','654345','Rosenheimer',87);
insert into addresses
   values(addresses_seq.nextval,'FRK','975773','Engelthaler',90);
insert into addresses
   values(addresses_seq.nextval,'BUD','498-275','Szytemlen',83);

select * from addresses;
```



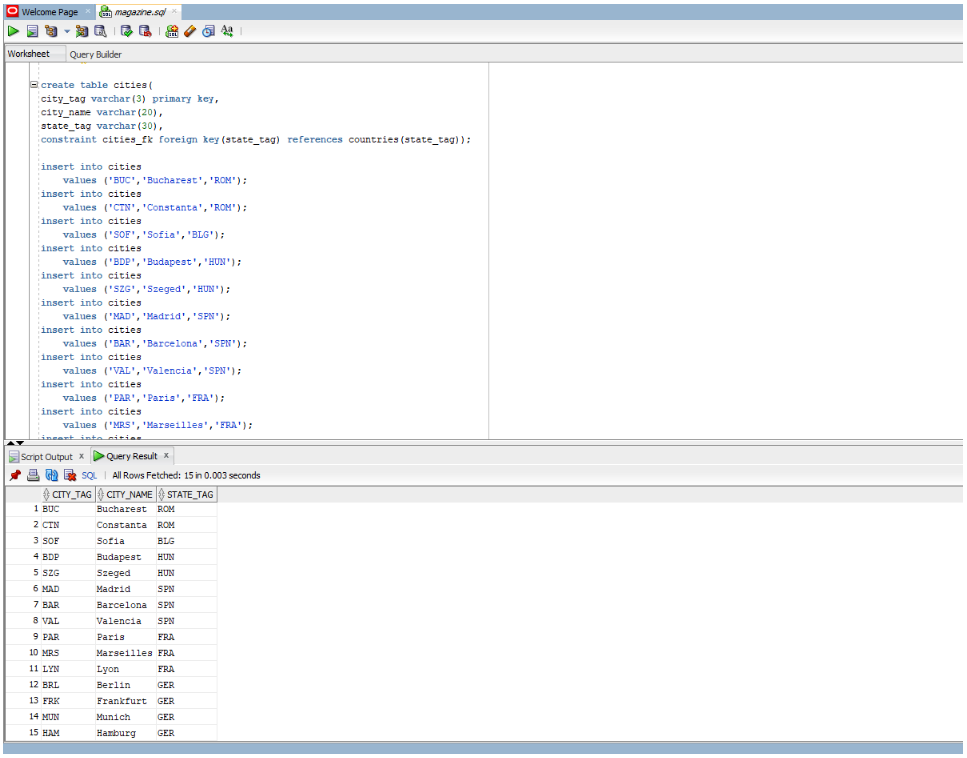
```
select * from products;

create table receipts(
receipt_id number(8) primary key,
transaction_date date,
id_market number(4),
constraint mark_fk foreign key(id_market) references markets(id_market));

insert into receipts
    values(10,'04-May-2022',35);
insert into receipts
    values(11,'04-May-2022',35);
insert into receipts
    values(12,'04-May-2022',35);
insert into receipts
    values(13,'04-May-2022',35);
insert into receipts
    values(14,'04-May-2022',35);
insert into receipts
    values(15,'04-May-2022',35);
insert into receipts
    values(16,'05-May-2022',35);
insert into receipts
    values(17,'06-May-2022',35);
insert into receipts
    values(18,'07-May-2022',35);
```

All Rows Fetched: 10 in 0.006 seconds

	RECEIPT_ID	TRANSACTION_DATE	ID_MARKET
1	10	04-MAY-22	35
2	11	04-MAY-22	35
3	12	04-MAY-22	35
4	13	04-MAY-22	35
5	14	04-MAY-22	35
6	15	04-MAY-22	35
7	16	05-MAY-22	35
8	17	06-MAY-22	35
9	18	07-MAY-22	35
10	19	07-MAY-22	30

```
create table receipts(

receipt_id number(8) primary key,

transaction_date date,

id_market number(4),

constraint mark_fk foreign key(id_market) references markets(id_market));


insert into receipts

   values(10,'04-May-2022',35);

insert into receipts

   values(11,'04-May-2022',35);

insert into receipts

   values(12,'04-May-2022',35);

insert into receipts

   values(13,'04-May-2022',35);

insert into receipts

   values(14,'04-May-2022',35);
```

```
insert into receipts
   values(15,'04-May-2022',35);
insert into receipts
   values(16,'05-May-2022',35);
insert into receipts
   values(17,'06-May-2022',35);
insert into receipts
   values(18,'07-May-2022',35);
insert into receipts
   values(19,'07-May-2022',30);


select * from receipts;
```



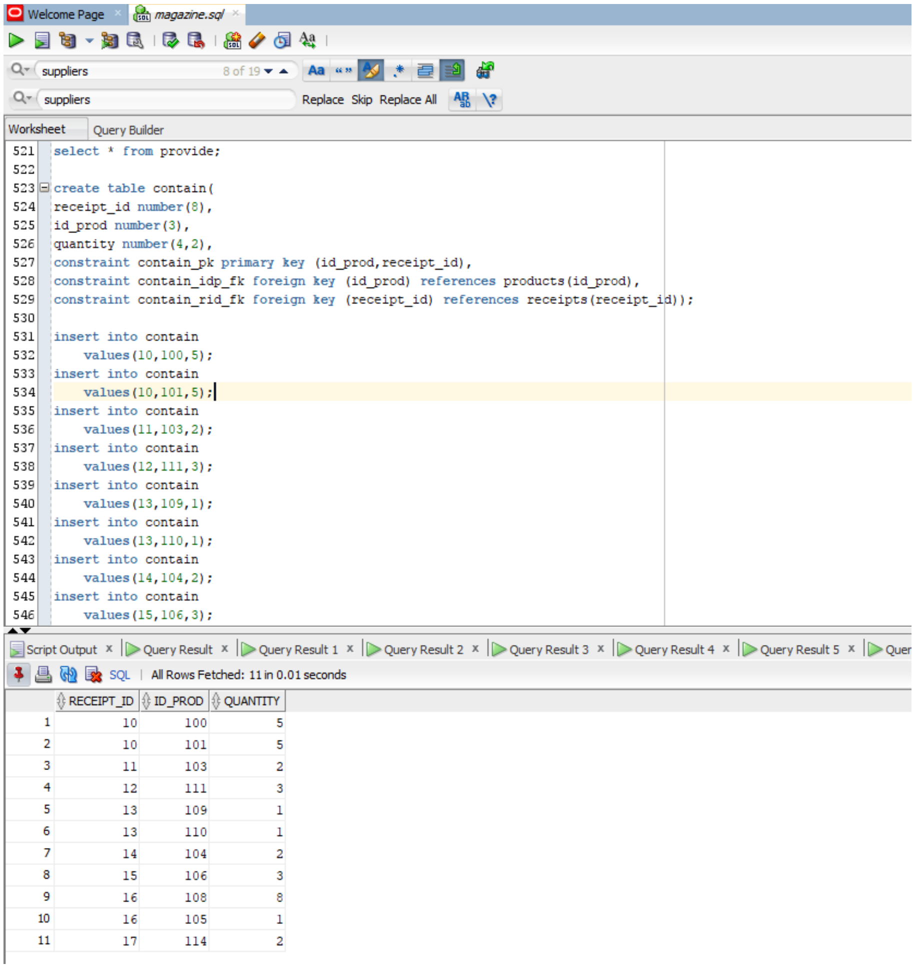
```
create table cities(
city_tag varchar(3) primary key,
city_name varchar(20),
state_tag varchar(30),
constraint cities_fk foreign key(state_tag) references countries(state_tag));

insert into cities
    values ('BUC','Bucharest','ROM');
insert into cities
    values ('CTN','Constanta','ROM');
insert into cities
    values ('SOF','Sofia','BLG');
insert into cities
    values ('BDP','Budapest','HUN');
insert into cities
    values ('SZG','Szeged','HUN');
insert into cities
    values ('MAD','Madrid','SPN');
insert into cities
    values ('BAR','Barcelona','SPN');
insert into cities
    values ('VAL','Valencia','SPN');
insert into cities
    values ('PAR','Paris','FRA');
insert into cities
    values ('MRS','Marseilles','FRA');
insert into cities
```

All Rows Fetched: 15 in 0.003 seconds

```
   CITY_TAG CITY_NAME  STATE_TAG
 1 BUC      Bucharest  ROM
 2 CTN      Constanta  ROM
 3 SOF      Sofia      BLG
 4 BDP      Budapest   HUN
 5 SZG      Szeged     HUN
 6 MAD      Madrid     SPN
 7 BAR      Barcelona  SPN
 8 VAL      Valencia   SPN
 9 PAR      Paris      FRA
10 MRS      Marseilles FRA
11 LYN      Lyon       FRA
12 BRL      Berlin     GER
13 FRK      Frankfurt  GER
14 MUN      Munich     GER
15 HAM      Hamburg    GER
```

```
create table cities(
city_tag varchar(3) primary key,
city_name varchar(20),
state_tag varchar(30),
constraint cities_fk foreign key(state_tag) references countries(state_tag));

insert into cities
   values ('BUC','Bucharest','ROM');
insert into cities
   values ('CTN','Constanta','ROM');
insert into cities
   values ('SOF','Sofia','BLG');
insert into cities
   values ('BDP','Budapest','HUN');
insert into cities
   values ('SZG','Szeged','HUN');
insert into cities
   values ('MAD','Madrid','SPN');
insert into cities
   values ('BAR','Barcelona','SPN');
insert into cities
   values ('VAL','Valencia','SPN');
insert into cities
   values ('PAR','Paris','FRA');
insert into cities
   values ('MRS','Marseilles','FRA');
insert into cities
   values ('LYN','Lyon','FRA');
insert into cities
   values ('BRL','Berlin','GER');
insert into cities
   values ('FRK','Frankfurt','GER');
insert into cities
```

```
   values ('MUN','Munich','GER');
insert into cities
   values ('HAM','Hamburg','GER');


select * from cities;
```



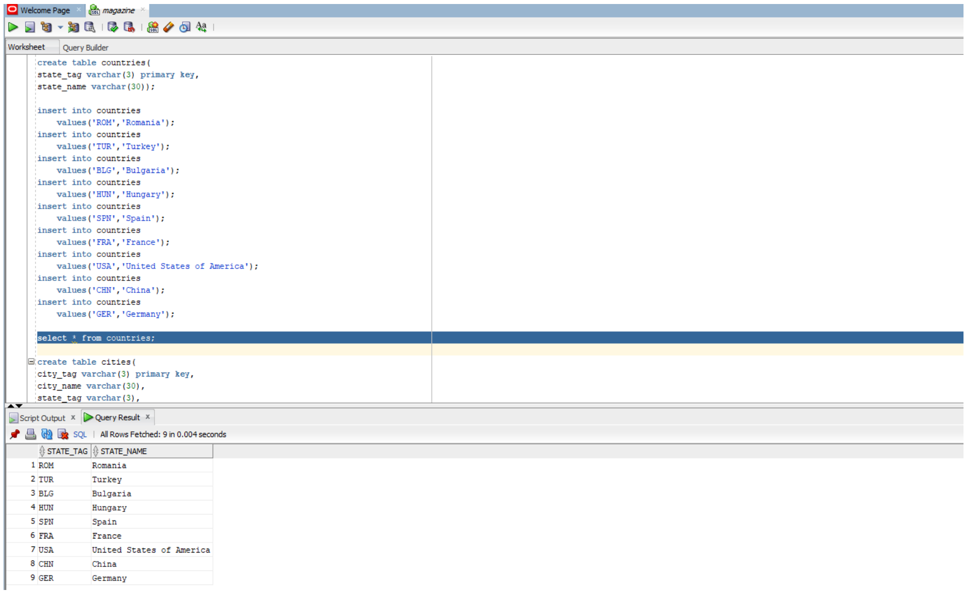
```
select * from provide;

create table contain(
receipt_id number(8),
id_prod number(3),
quantity number(4,2),
constraint contain_pk primary key (id_prod,receipt_id),
constraint contain_idp_fk foreign key (id_prod) references products(id_prod),
constraint contain_rid_fk foreign key (receipt_id) references receipts(receipt_id));

insert into contain
    values(10,100,5);
insert into contain
    values(10,101,5);
insert into contain
    values(11,103,2);
insert into contain
    values(12,111,3);
insert into contain
    values(13,109,1);
insert into contain
    values(13,110,1);
insert into contain
    values(14,104,2);
insert into contain
    values(15,106,3);
```

All Rows Fetched: 11 in 0.01 seconds

	RECEIPT_ID	ID_PROD	QUANTITY
1	10	100	5
2	10	101	5
3	11	103	2
4	12	111	3
5	13	109	1
6	13	110	1
7	14	104	2
8	15	106	3
9	16	108	8
10	16	105	1
11	17	114	2

```
create table contain(
receipt_id number(8),
id_prod number(3),
quantity number(4,2),
constraint contain_pk primary key (id_prod,receipt_id),
constraint contain_idp_fk foreign key (id_prod) references products(id_prod),
constraint contain_rid_fk foreign key (receipt_id) references receipts(receipt_id));
```

```
insert into contain
    values(10,100,5);
insert into contain
    values(10,101,5);
insert into contain
    values(11,103,2);
insert into contain
    values(12,111,3);
insert into contain
    values(13,109,1);
insert into contain
    values(13,110,1);
insert into contain
    values(14,104,2);
insert into contain
    values(15,106,3);
insert into contain
    values(16,108,8);
insert into contain
    values(16,105,1);
insert into contain
    values(17,114,2);

select * from contain;
```



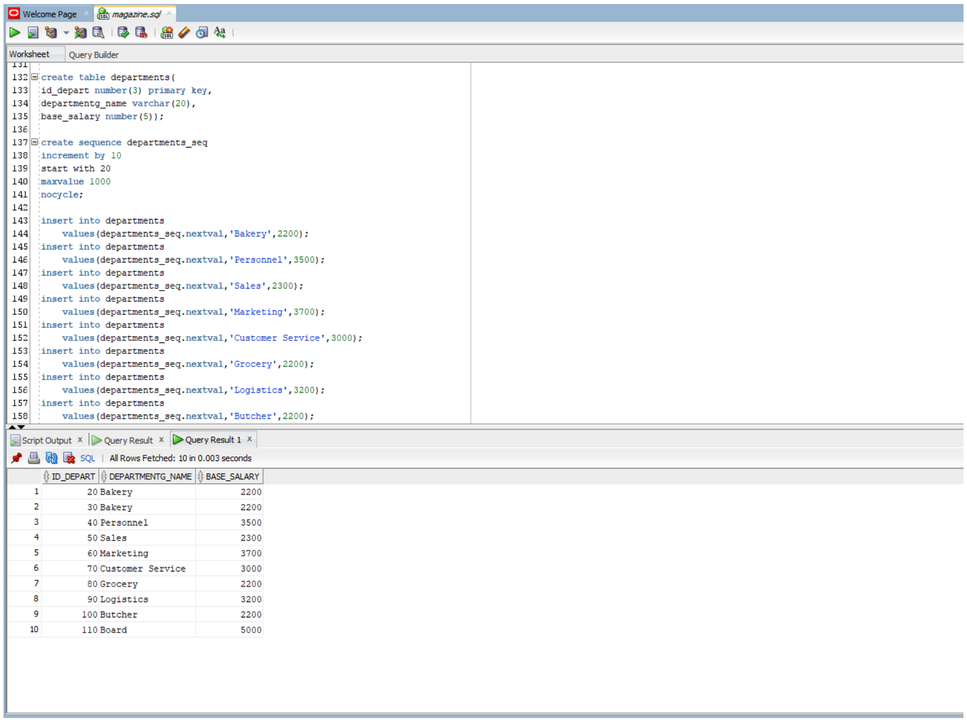
```
create table countries(
state_tag varchar(3) primary key,
state_name varchar(30));

insert into countries
    values('ROM','Romania');
insert into countries
    values('TUR','Turkey');
insert into countries
    values('BLG','Bulgaria');
insert into countries
    values('HUN','Hungary');
insert into countries
    values('SPN','Spain');
insert into countries
    values('FRA','France');
insert into countries
    values('USA','United States of America');
insert into countries
    values('CHN','China');
insert into countries
    values('GER','Germany');

select * from countries;

create table cities(
city_tag varchar(3) primary key,
city_name varchar(30),
state_tag varchar(3),
```

All Rows Fetched: 9 in 0.004 seconds

	STATE_TAG	STATE_NAME
1	ROM	Romania
2	TUR	Turkey
3	BLG	Bulgaria
4	HUN	Hungary
5	SPN	Spain
6	FRA	France
7	USA	United States of America
8	CHN	China
9	GER	Germany

```
create table countries(

state_tag varchar(3) primary key,

state_name varchar(30));


insert into countries

   values('ROM','Romania');

insert into countries

   values('TUR','Turkey');

insert into countries

   values('BLG','Bulgaria');

insert into countries

   values('HUN','Hungary');

insert into countries

   values('SPN','Spain');

insert into countries

   values('FRA','France');

insert into countries

   values('USA','United States of America');

insert into countries
```

```
   values('CHN','China');
insert into countries
   values('GER','Germany');


select * from countries;
```



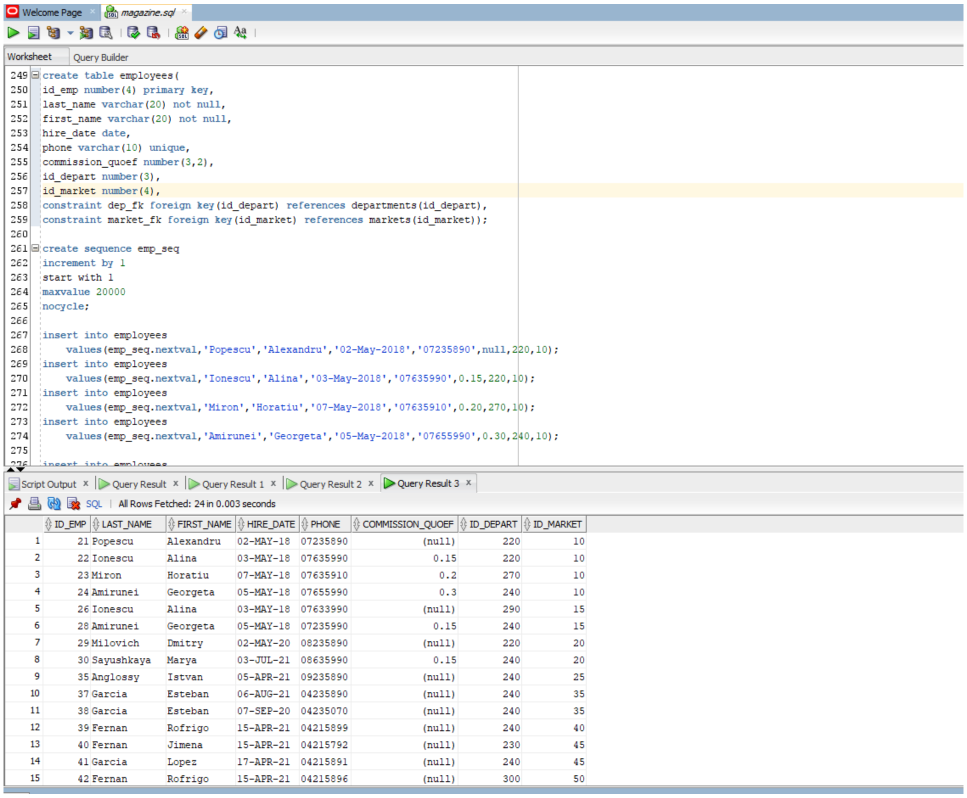
```
create table departments(
id_depart number(3) primary key,
departmentg_name varchar(20),
base_salary number(5));

create sequence departments_seq
increment by 10
start with 20
maxvalue 1000
nocycle;

insert into departments
    values(departments_seq.nextval,'Bakery',2200);
insert into departments
    values(departments_seq.nextval,'Personnel',3500);
insert into departments
    values(departments_seq.nextval,'Sales',2300);
insert into departments
    values(departments_seq.nextval,'Marketing',3700);
insert into departments
    values(departments_seq.nextval,'Customer Service',3000);
insert into departments
    values(departments_seq.nextval,'Grocery',2200);
insert into departments
    values(departments_seq.nextval,'Logistics',3200);
insert into departments
    values(departments_seq.nextval,'Butcher',2200);
```

All Rows Fetched: 10 in 0.003 seconds

	ID_DEPART	DEPARTMENTG_NAME	BASE_SALARY
1	20	Bakery	2200
2	30	Bakery	2200
3	40	Personnel	3500
4	50	Sales	2300
5	60	Marketing	3700
6	70	Customer Service	3000
7	80	Grocery	2200
8	90	Logistics	3200
9	100	Butcher	2200
10	110	Board	5000

```
create table departments(
id_depart number(3) primary key,
departmentg_name varchar(20) not null,
base_salary number(5) not null);


create sequence departments_seq
increment by 10
start with 20
maxvalue 1000
nocycle;
```

```
insert into departments
   values(departments_seq.nextval,'Bakery',2200);
insert into departments
   values(departments_seq.nextval,'Personnel',3500);
insert into departments
   values(departments_seq.nextval,'Sales',2300);
insert into departments
   values(departments_seq.nextval,'Marketing',3700);
insert into departments
   values(departments_seq.nextval,'Customer Service',3000);
insert into departments
   values(departments_seq.nextval,'Grocery',2200);
insert into departments
   values(departments_seq.nextval,'Logistics',3200);
insert into departments
   values(departments_seq.nextval,'Butcher',2200);
insert into departments
   values(departments_seq.nextval,'Board',5000);

select * from departments;
```



```
create table employees(
id_emp number(4) primary key,
last_name varchar(20) not null,
first_name varchar(20) not null,
hire_date date,
phone varchar(10) unique,
commission_quoef number(3,2),
id_depart number(3),
id_market number(4),
constraint dep_fk foreign key(id_depart) references departments(id_depart),
constraint market_fk foreign key(id_market) references markets(id_market));

create sequence emp_seq
increment by 1
start with 1
maxvalue 20000
nocycle;

insert into employees
    values(emp_seq.nextval,'Popescu','Alexandru','02-May-2018','07235890',null,220,10);
insert into employees
    values(emp_seq.nextval,'Ionescu','Alina','03-May-2018','07635990',0.15,220,10);
insert into employees
    values(emp_seq.nextval,'Miron','Horatiu','07-May-2018','07635910',0.20,270,10);
insert into employees
    values(emp_seq.nextval,'Amirunei','Georgeta','05-May-2018','07655990',0.30,240,10);

insert into employees
```

All Rows Fetched: 24 in 0.003 seconds

	ID_EMP	LAST_NAME	FIRST_NAME	HIRE_DATE	PHONE	COMMISSION_QUOEF	ID_DEPART	ID_MARKET
1	21	Popescu	Alexandru	02-MAY-18	07235890	(null)	220	10
2	22	Ionescu	Alina	03-MAY-18	07635990	0.15	220	10
3	23	Miron	Horatiu	07-MAY-18	07635910	0.2	270	10
4	24	Amirunei	Georgeta	05-MAY-18	07655990	0.3	240	10
5	26	Ionescu	Alina	03-MAY-18	07633990	(null)	290	15
6	28	Amirunei	Georgeta	05-MAY-18	07235990	0.15	240	15
7	29	Milovich	Dmitry	02-MAY-20	08235890	(null)	220	20
8	30	Sayushkaya	Marya	03-JUL-21	08635990	0.15	240	20
9	35	Anglossy	Istvan	05-APR-21	09235890	(null)	240	25
10	37	Garcia	Esteban	06-AUG-21	04235890	(null)	240	35
11	38	Garcia	Esteban	07-SEP-20	04235070	(null)	240	35
12	39	Fernan	Rofrigo	15-APR-21	04215899	(null)	240	40
13	40	Fernan	Jimena	15-APR-21	04215792	(null)	230	45
14	41	Garcia	Lopez	17-APR-21	04215891	(null)	240	45
15	42	Fernan	Rofrigo	15-APR-21	04215896	(null)	300	50

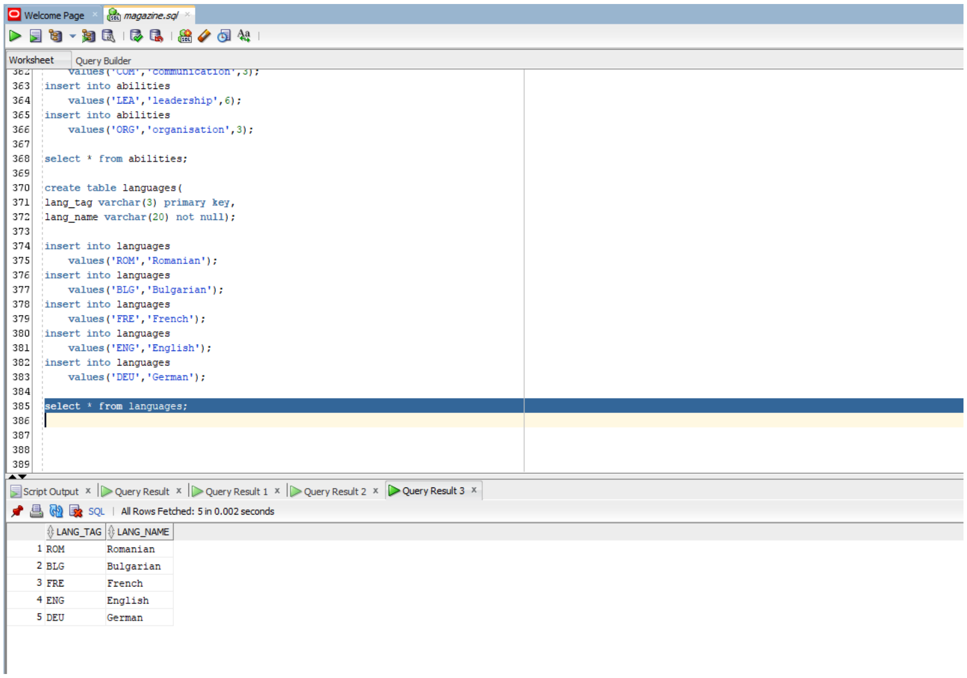
```
create table employees(

id_emp number(4) primary key,

last_name varchar(20) not null,

first_name varchar(20) not null,

hire_date date,

phone varchar(10) unique,

commission_quoef number(3,2),

id_depart number(3),

id_market number(4),

constraint dep_fk foreign key(id_depart) references departments(id_depart),

constraint market_fk foreign key(id_market) references markets(id_market));


create sequence emp_seq

increment by 1

start with 1
```

```
maxvalue 20000
nocycle;

insert into employees
   values(emp_seq.nextval,'Popescu','Alexandru','02-May-2018','07235890',null,220,10);
insert into employees
   values(emp_seq.nextval,'Ionescu','Alina','03-May-2018','07635990',0.15,220,10);
insert into employees
   values(emp_seq.nextval,'Miron','Horatiu','07-May-2018','07635910',0.20,270,10);
insert into employees
   values(emp_seq.nextval,'Amirunei','Georgeta','05-May-2018','07655990',0.30,240,10);

insert into employees
   values(emp_seq.nextval,'Mihailescu','Adina','07-Jun-2019','07235890',0.25,220,15);
insert into employees
   values(emp_seq.nextval,'Ionescu','Alina','03-May-2018','07633990',null,290,15);
insert into employees
   values(emp_seq.nextval,'Miron','Horatiu','07-May-2018','07635990',0.05,270,15);
insert into employees
   values(emp_seq.nextval,'Amirunei','Georgeta','05-May-2018','07235990',0.15,240,15);

insert into employees
   values(emp_seq.nextval,'Milovich','Dmitry','02-May-2020','08235890',null,220,20);
insert into employees
   values(emp_seq.nextval,'Sayushkaya','Marya','03-Jul-2021','08635990',0.15,240,20);
insert into employees
   values(emp_seq.nextval,'Dobrovich','Aleksandr','07-Aug-2020','08635990',0.20,240,20);

insert into employees
```

```
   values(emp_seq.nextval,'Milovich','Dmitry','02-May-2020','08235890',null,220,20);
insert into employees
   values(emp_seq.nextval,'Sayushkaya','Marya','03-Jul-2021','08635990',0.15,240,20);
insert into employees
   values(emp_seq.nextval,'Dobrovich','Aleksandr','07-Aug-2020','08635990',0.20,240,20);

insert into employees
   values(emp_seq.nextval,'Anglossy','Istvan','05-Apr-2021','09235890',null,240,25);

insert into employees
   values(emp_seq.nextval,'Gyorgethery','Marika','05-Apr-2021','09235890',null,240,30);

insert into employees
   values(emp_seq.nextval,'Garcia','Esteban','06-Aug-2021','04235890',null,240,35);
insert into employees
   values(emp_seq.nextval,'LLuro','Javier','07-Sep-2020','04235070',null,240,35);

insert into employees
   values(emp_seq.nextval,'Fernan','Rofrigo','15-Apr-2021','04215899',null,240,40);

insert into employees
   values(emp_seq.nextval,'Fernan','Jimena','15-Apr-2021','04215792',null,230,45);
insert into employees
   values(emp_seq.nextval,'Garcia','Lopez','17-Apr-2021','04215891',null,240,45);
insert into employees
   values(emp_seq.nextval,'Malfrida','Infanta','18-Apr-2021','04215991',0.15,240,45);

insert into employees
   values(emp_seq.nextval,'Fernan','Rofrigo','15-Apr-2021','04215896',null,300,50);
```

```
insert into employees
    values(emp_seq.nextval,'Garcia','Lopez','17-Apr-2021','04215871',null,240,50


insert into employees
    values(emp_seq.nextval,'Boivelle','Marie','15-Apr-2021','04215895',null,250,55);
insert into employees
    values(emp_seq.nextval,'Blois','isabelle','17-Apr-2021','04285891',null,240,55);


insert into employees
    values(emp_seq.nextval,'Marquie','Alessandre','15-Apr-2021','04215893',null,260,60);
insert into employees
    values(emp_seq.nextval,'Broget','Almec','17-Apr-2021','04225892',null,240,60);


insert into employees
    values(emp_seq.nextval,'Charlee','Antoine','15-Apr-2021','04215866',null,280,65);
insert into employees
    values(emp_seq.nextval,'Vivizon','Louis','17-Apr-2021','04215867',null,240,65);


insert into employees
    values(emp_seq.nextval,'Gustav','Klauss','17-Apr-2021','09215867',null,240,70);


insert into employees
    values(emp_seq.nextval,'Mahstern','Mark','17-Apr-2021','09315867',null,240,75);


insert into employees
    values(emp_seq.nextval,'Friedrichson','Karl','17-Apr-2021','09415867',null,240,80);


insert into employees
    values(emp_seq.nextval,'Klogge','Frau','17-Apr-2021','09515867',null,240,85);
```

```
select * from employees;
```



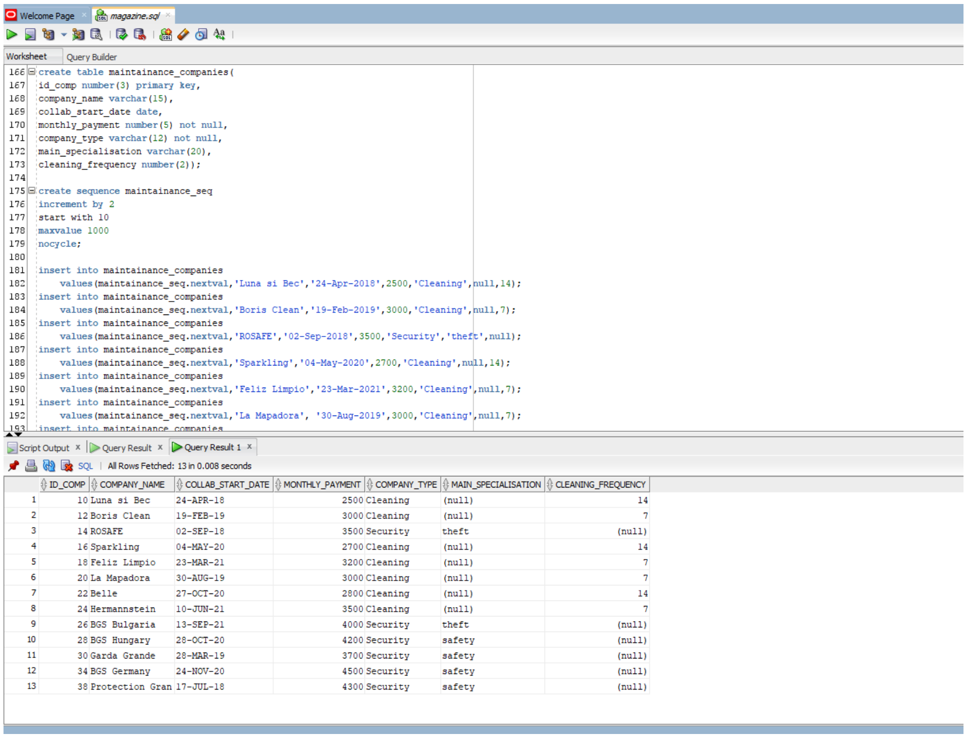
```
    values('COM','communication',3);
insert into abilities
    values('LEA','leadership',6);
insert into abilities
    values('ORG','organisation',3);

select * from abilities;

create table languages(
lang_tag varchar(3) primary key,
lang_name varchar(20) not null);

insert into languages
    values('ROM','Romanian');
insert into languages
    values('BLG','Bulgarian');
insert into languages
    values('FRE','French');
insert into languages
    values('ENG','English');
insert into languages
    values('DEU','German');

select * from languages;
```

All Rows Fetched: 5 in 0.002 seconds

	LANG_TAG	LANG_NAME
1	ROM	Romanian
2	BLG	Bulgarian
3	FRE	French
4	ENG	English
5	DEU	German

```
create table languages(
lang_tag varchar(3) primary key,
lang_name varchar(20) not null);


insert into languages
   values('ROM','Romanian');
insert into languages
   values('BLG','Bulgarian');
insert into languages
   values('FRE','French');
insert into languages
   values('ENG','English');
insert into languages
   values('DEU','German');
```

```
select * from languages;
```



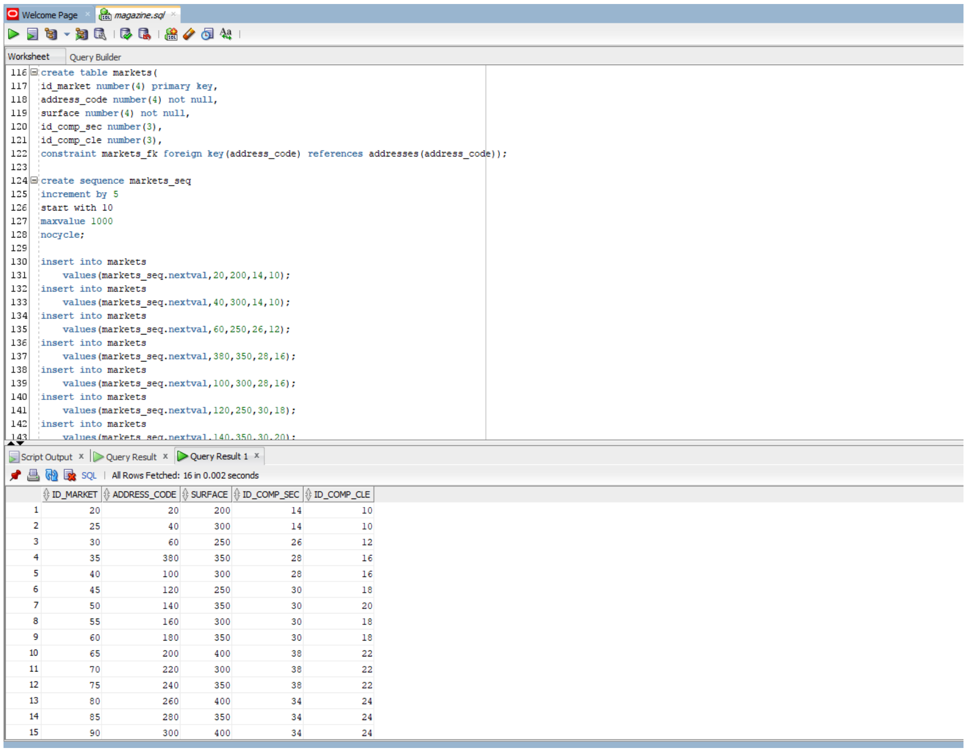
```
create table maintainance_companies(
id_comp number(3) primary key,
company_name varchar(15),
collab_start_date date,
monthly_payment number(5) not null,
company_type varchar(12) not null,
main_specialisation varchar(20),
cleaning_frequency number(2));

create sequence maintainance_seq
increment by 2
start with 10
maxvalue 1000
nocycle;

insert into maintainance_companies
    values(maintainance_seq.nextval,'Luna si Bec','24-Apr-2018',2500,'Cleaning',null,14);
insert into maintainance_companies
    values(maintainance_seq.nextval,'Boris Clean','19-Feb-2019',3000,'Cleaning',null,7);
insert into maintainance_companies
    values(maintainance_seq.nextval,'ROSAFE','02-Sep-2018',3500,'Security','theft',null);
insert into maintainance_companies
    values(maintainance_seq.nextval,'Sparkling','04-May-2020',2700,'Cleaning',null,14);
insert into maintainance_companies
    values(maintainance_seq.nextval,'Feliz Limpio','23-Mar-2021',3200,'Cleaning',null,7);
insert into maintainance_companies
    values(maintainance_seq.nextval,'La Mapadora', '30-Aug-2019',3000,'Cleaning',null,7);
insert into maintainance_companies
```

All Rows Fetched: 13 in 0.008 seconds

	ID_COMP	COMPANY_NAME	COLLAB_START_DATE	MONTHLY_PAYMENT	COMPANY_TYPE	MAIN_SPECIALISATION	CLEANING_FREQUENCY
1	10	Luna si Bec	24-APR-18	2500	Cleaning	(null)	14
2	12	Boris Clean	19-FEB-19	3000	Cleaning	(null)	7
3	14	ROSAFE	02-SEP-18	3500	Security	theft	(null)
4	16	Sparkling	04-MAY-20	2700	Cleaning	(null)	14
5	18	Feliz Limpio	23-MAR-21	3200	Cleaning	(null)	7
6	20	La Mapadora	30-AUG-19	3000	Cleaning	(null)	7
7	22	Belle	27-OCT-20	2800	Cleaning	(null)	14
8	24	Hermannstein	10-JUN-21	3500	Cleaning	(null)	7
9	26	BGS Bulgaria	13-SEP-21	4000	Security	theft	(null)
10	28	BGS Hungary	28-OCT-20	4200	Security	safety	(null)
11	30	Garda Grande	28-MAR-19	3700	Security	safety	(null)
12	34	BGS Germany	24-NOV-20	4500	Security	safety	(null)
13	38	Protection Gran	17-JUL-18	4300	Security	safety	(null)

```
create table maintainance_companies(
id_comp number(3) primary key,
company_name varchar(15),
collab_start_date date,
monthly_payment number(5) not null,
company_type varchar(12) not null,
main_specialisation varchar(20),
cleaning_frequency number(2));

create sequence maintainance_seq
increment by 2
start with 10
maxvalue 1000
nocycle;
```

```
insert into maintainance_companies
   values(maintainance_seq.nextval,'Luna si Bec','24-Apr-2018',2500,'Cleaning',null,14);
insert into maintainance_companies
   values(maintainance_seq.nextval,'Boris Clean','19-Feb-2019',3000,'Cleaning',null,7);
insert into maintainance_companies
   values(maintainance_seq.nextval,'ROSAFE','02-Sep-2018',3500,'Security','theft',null);
insert into maintainance_companies
   values(maintainance_seq.nextval,'Sparkling','04-May-2020',2700,'Cleaning',null,14);
insert into maintainance_companies
   values(maintainance_seq.nextval,'Feliz Limpio','23-Mar-2021',3200,'Cleaning',null,7);
insert into maintainance_companies
   values(maintainance_seq.nextval,'La Mapadora', '30-Aug-2019',3000,'Cleaning',null,7);
insert into maintainance_companies
   values(maintainance_seq.nextval,'Belle', '27-Oct-2020',2800,'Cleaning',null,14);
insert into maintainance_companies
   values(maintainance_seq.nextval,'Hermannstein', '10-Jun-2021',3500,'Cleaning',null,7);
insert into maintainance_companies
   values(maintainance_seq.nextval,'BGS Bulgaria', '13-Sep-2021',4000,'Security','theft',null);
insert into maintainance_companies
   values(maintainance_seq.nextval,'BGS Hungary', '28-Oct-2020',4200,'Security','safety',null);
insert into maintainance_companies
   values(maintainance_seq.nextval,'Garda Grande', '28-Mar-2019',3700,'Security','safety',null);
insert into maintainance_companies
   values(maintainance_seq.nextval,'Protection Gran', '17-Jul-2018',4300,'Security','safety',null);
insert into maintainance_companies
   values(maintainance_seq.nextval,'BGS Germany', '24-Nov-2020',4500,'Security','safety',null);

select * from maintainance_companies;
```



```
create table markets(
id_market number(4) primary key,
address_code number(4) not null,
surface number(4) not null,
id_comp_sec number(3),
id_comp_cle number(3),
constraint markets_fk foreign key(address_code) references addresses(address_code));

create sequence markets_seq
increment by 5
start with 10
maxvalue 1000
nocycle;

insert into markets
    values(markets_seq.nextval,20,200,14,10);
insert into markets
    values(markets_seq.nextval,40,300,14,10);
insert into markets
    values(markets_seq.nextval,60,250,26,12);
insert into markets
    values(markets_seq.nextval,380,350,28,16);
insert into markets
    values(markets_seq.nextval,100,300,28,16);
insert into markets
    values(markets_seq.nextval,120,250,30,18);
insert into markets
    values(markets_seq.nextval,140,350,30,20);
```

All Rows Fetched: 16 in 0.002 seconds

	ID_MARKET	ADDRESS_CODE	SURFACE	ID_COMP_SEC	ID_COMP_CLE
1	20	20	200	14	10
2	25	40	300	14	10
3	30	60	250	26	12
4	35	380	350	28	16
5	40	100	300	28	16
6	45	120	250	30	18
7	50	140	350	30	20
8	55	160	300	30	18
9	60	180	350	30	18
10	65	200	400	38	22
11	70	220	300	38	22
12	75	240	350	38	22
13	80	260	400	34	24
14	85	280	350	34	24
15	90	300	400	34	24

```
create table markets(

id_market number(4) primary key,

address_code number(4) not null,

surface number(4) not null,

id_comp_sec number(3),

id_comp_cle number(3),

constraint markets_fk foreign key(address_code) references addresses(address_code),

constraint sec_fk foreign key(id_comp_sec) references maintainance_companies(id_comp),

constraint cle_fk foreign key(id_comp_cle) references maintainance_companies(id_comp));


create sequence markets_seq

increment by 5

start with 10

maxvalue 1000

nocycle;
```

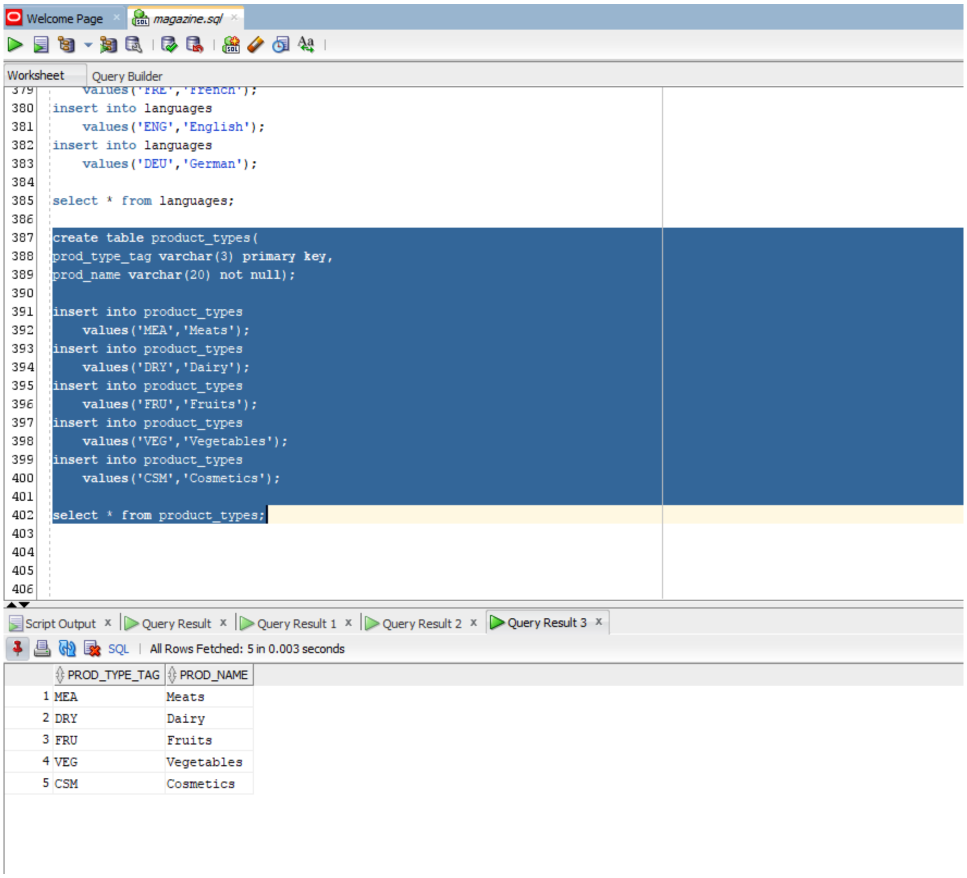
```
insert into markets
   values(markets_seq.nextval,20,200,14,10);
insert into markets
   values(markets_seq.nextval,40,300,14,10);
insert into markets
   values(markets_seq.nextval,60,250,26,12);
insert into markets
   values(markets_seq.nextval,380,350,28,16);
insert into markets
   values(markets_seq.nextval,100,300,28,16);
insert into markets
   values(markets_seq.nextval,120,250,30,18);
insert into markets
   values(markets_seq.nextval,140,350,30,20);
insert into markets
   values(markets_seq.nextval,160,300,30,18);
insert into markets
   values(markets_seq.nextval,180,350,30,18);
insert into markets
   values(markets_seq.nextval,200,400,38,22);
insert into markets
   values(markets_seq.nextval,220,300,38,22);
insert into markets
   values(markets_seq.nextval,240,350,38,22);
insert into markets
   values(markets_seq.nextval,260,400,34,24);
insert into markets
   values(markets_seq.nextval,280,350,34,24);
insert into markets
```

values(markets_seq.nextval,300,400,34,24);

insert into markets

values(markets_seq.nextval,320,300,34,24);

select * from markets;



```
    values('FRE','French');
insert into languages
    values('ENG','English');
insert into languages
    values('DEU','German');

select * from languages;

create table product_types(
prod_type_tag varchar(3) primary key,
prod_name varchar(20) not null);

insert into product_types
    values('MEA','Meats');
insert into product_types
    values('DRY','Dairy');
insert into product_types
    values('FRU','Fruits');
insert into product_types
    values('VEG','Vegetables');
insert into product_types
    values('CSM','Cosmetics');

select * from product_types;
```

All Rows Fetched: 5 in 0.003 seconds

	PROD_TYPE_TAG	PROD_NAME
1	MEA	Meats
2	DRY	Dairy
3	FRU	Fruits
4	VEG	Vegetables
5	CSM	Cosmetics

create table product_types(

prod_type_tag varchar(3) primary key,

prod_name varchar(20) not null);

insert into product_types

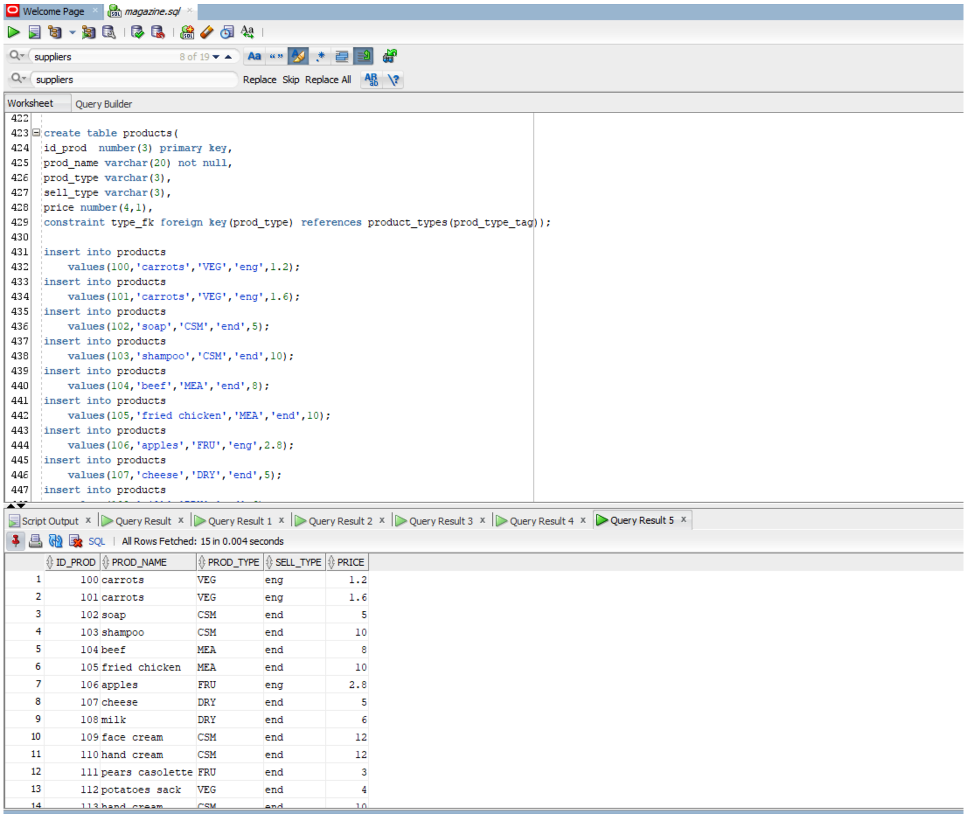
values('MEA','Meats');

insert into product_types

values('DRY','Dairy');

insert into product_types

```
   values('FRU','Fruits');

insert into product_types

   values('VEG','Vegetables');

insert into product_types

   values('CSM','Cosmetics');


select * from product_types;
```



```

create table products(
id_prod  number(3) primary key,
prod_name varchar(20) not null,
prod_type varchar(3),
sell_type varchar(3),
price number(4,1),
constraint type_fk foreign key(prod_type) references product_types(prod_type_tag));

insert into products
    values(100,'carrots','VEG','eng',1.2);
insert into products
    values(101,'carrots','VEG','eng',1.6);
insert into products
    values(102,'soap','CSM','end',5);
insert into products
    values(103,'shampoo','CSM','end',10);
insert into products
    values(104,'beef','MEA','end',8);
insert into products
    values(105,'fried chicken','MEA','end',10);
insert into products
    values(106,'apples','FRU','eng',2.8);
insert into products
    values(107,'cheese','DRY','end',5);
insert into products
```

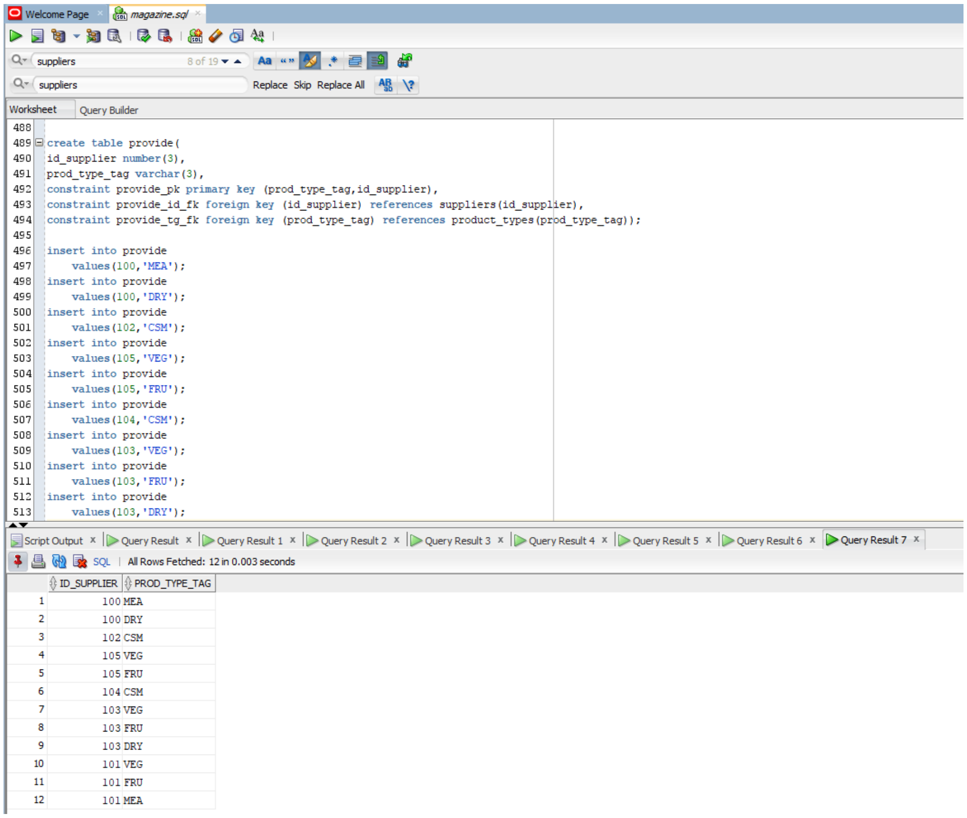
All Rows Fetched: 15 in 0.004 seconds

	ID_PROD	PROD_NAME	PROD_TYPE	SELL_TYPE	PRICE
1	100	carrots	VEG	eng	1.2
2	101	carrots	VEG	eng	1.6
3	102	soap	CSM	end	5
4	103	shampoo	CSM	end	10
5	104	beef	MEA	end	8
6	105	fried chicken	MEA	end	10
7	106	apples	FRU	eng	2.8
8	107	cheese	DRY	end	5
9	108	milk	DRY	end	6
10	109	face cream	CSM	end	12
11	110	hand cream	CSM	end	12
12	111	pears casolette	FRU	end	3
13	112	potatoes sack	VEG	end	4
14	113	hand cream	CSM	end	10

```
create table products(

id_prod  number(3) primary key,

prod_name varchar(20) not null,

prod_type varchar(3),

sell_type varchar(3),

price number(4,1),

constraint type_fk foreign key(prod_type) references product_types(prod_type_tag));
```

```
insert into products
   values(100,'carrots','VEG','eng',1.2);
insert into products
   values(101,'carrots','VEG','eng',1.6);
insert into products
   values(102,'soap','CSM','end',5);
insert into products
   values(103,'shampoo','CSM','end',10);
insert into products
   values(104,'beef','MEA','end',8);
insert into products
   values(105,'fried chicken','MEA','end',10);
insert into products
   values(106,'apples','FRU','eng',2.8);
insert into products
   values(107,'cheese','DRY','end',5);
insert into products
   values(108,'milk','DRY','end',6);
insert into products
   values(109,'face cream','CSM','end',12);
insert into products
   values(110,'hand cream','CSM','end',12);
insert into products
   values(111,'pears casolette','FRU','end',3);
insert into products
   values(112,'potatoes sack','VEG','end',4);
insert into products
   values(113,'hand cream','CSM','end',10);
insert into products
```

```
    values(114,'bananas','FRU','eng',2.5);


select * from products;
```



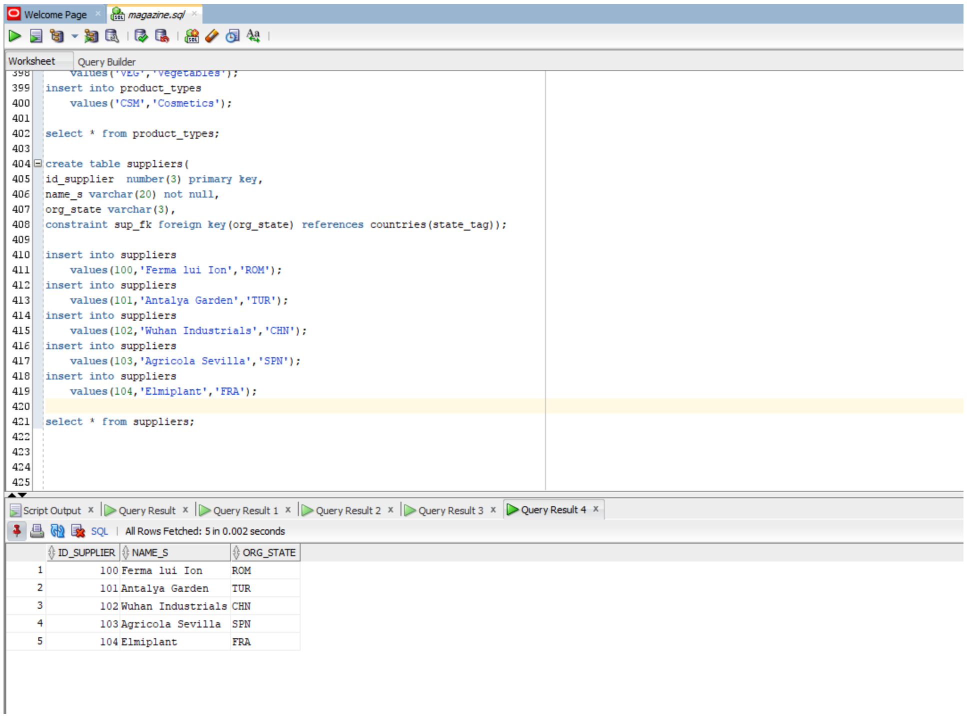
```
create table provide(
id_supplier number(3),
prod_type_tag varchar(3),
constraint provide_pk primary key (prod_type_tag,id_supplier),
constraint provide_id_fk foreign key (id_supplier) references suppliers(id_supplier),
constraint provide_tg_fk foreign key (prod_type_tag) references product_types(prod_type_tag));

insert into provide
    values(100,'MEA');
insert into provide
    values(100,'DRY');
insert into provide
    values(102,'CSM');
insert into provide
    values(105,'VEG');
insert into provide
    values(105,'FRU');
insert into provide
    values(104,'CSM');
insert into provide
    values(103,'VEG');
insert into provide
    values(103,'FRU');
insert into provide
    values(103,'DRY');
```

All Rows Fetched: 12 in 0.003 seconds

	ID_SUPPLIER	PROD_TYPE_TAG
1	100	MEA
2	100	DRY
3	102	CSM
4	105	VEG
5	105	FRU
6	104	CSM
7	103	VEG
8	103	FRU
9	103	DRY
10	101	VEG
11	101	FRU
12	101	MEA

```
create table provide(

id_supplier number(3),

prod_type_tag varchar(3),

constraint provide_pk primary key (prod_type_tag,id_supplier),

constraint provide_id_fk foreign key (id_supplier) references suppliers(id_supplier),

constraint provide_tg_fk foreign key (prod_type_tag) references product_types(prod_type_tag));


insert into provide

    values(100,'MEA');

insert into provide

    values(100,'DRY');

insert into provide
```

```
    values(102,'CSM');
insert into provide
    values(105,'VEG');
insert into provide
    values(105,'FRU');
insert into provide
    values(104,'CSM');
insert into provide
    values(103,'VEG');
insert into provide
    values(103,'FRU');
insert into provide
    values(103,'DRY');
insert into provide
    values(101,'VEG');
insert into provide
    values(101,'FRU');
insert into provide
    values(101,'MEA');

select * from provide;
```



```
    values('VEG','Vegetables');
insert into product_types
    values('CSM','Cosmetics');

select * from product_types;

create table suppliers(
id_supplier  number(3) primary key,
name_s varchar(20) not null,
org_state varchar(3),
constraint sup_fk foreign key(org_state) references countries(state_tag));

insert into suppliers
    values(100,'Ferma lui Ion','ROM');
insert into suppliers
    values(101,'Antalya Garden','TUR');
insert into suppliers
    values(102,'Wuhan Industrials','CHN');
insert into suppliers
    values(103,'Agricola Sevilla','SPN');
insert into suppliers
    values(104,'Elmiplant','FRA');

select * from suppliers;
```

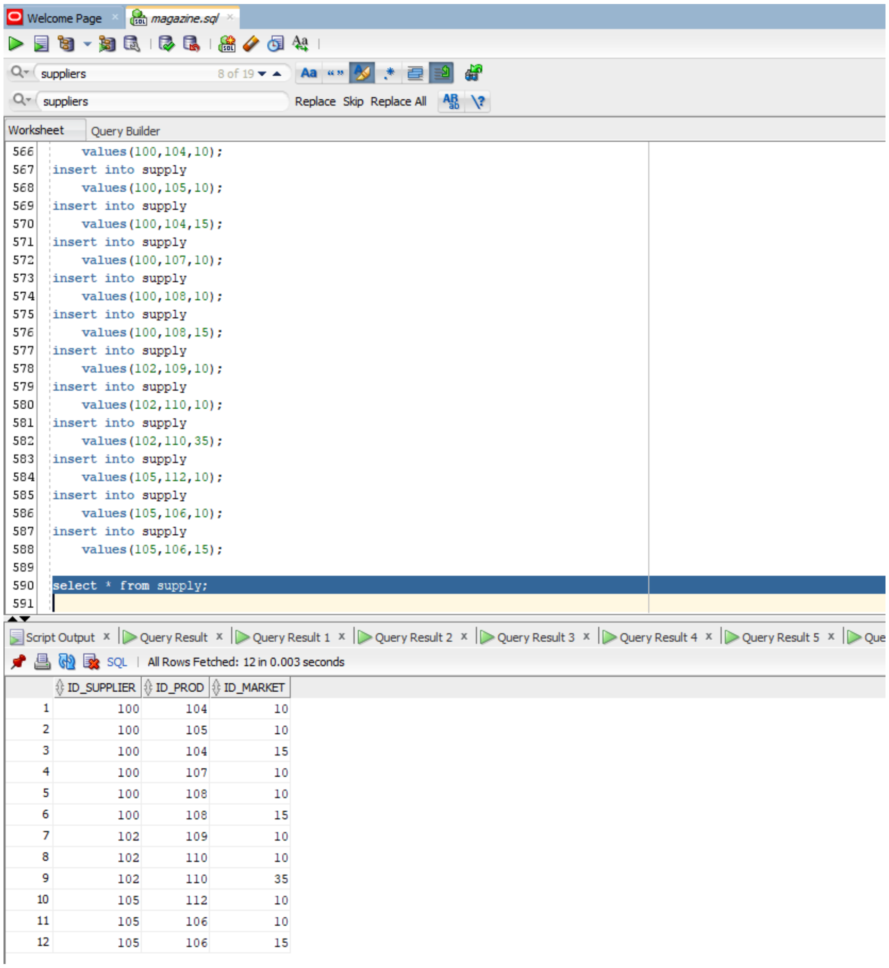
```
All Rows Fetched: 5 in 0.002 seconds

     ID_SUPPLIER NAME_S            ORG_STATE
  1          100 Ferma lui Ion     ROM
  2          101 Antalya Garden    TUR
  3          102 Wuhan Industrials CHN
  4          103 Agricola Sevilla  SPN
  5          104 Elmiplant         FRA
```

```
create table suppliers(

id_supplier  number(3) primary key,

name_s varchar(20) not null,

org_state varchar(3),

constraint sup_fk foreign key(org_state) references countries(state_tag));


insert into suppliers

    values(100,'Ferma lui Ion','ROM');

insert into suppliers

    values(101,'Antalya Garden','TUR');

insert into suppliers

    values(102,'Wuhan Industrials','CHN');

insert into suppliers

    values(103,'Agricola Sevilla','SPN');

insert into suppliers

    values(104,'Elmiplant','FRA');
```

insert into suppliers

values(105,'Kasimiva Fermata','BLG');

select * from suppliers;



```
    values(100,104,10);
insert into supply
    values(100,105,10);
insert into supply
    values(100,104,15);
insert into supply
    values(100,107,10);
insert into supply
    values(100,108,10);
insert into supply
    values(100,108,15);
insert into supply
    values(102,109,10);
insert into supply
    values(102,110,10);
insert into supply
    values(102,110,35);
insert into supply
    values(105,112,10);
insert into supply
    values(105,106,10);
insert into supply
    values(105,106,15);

select * from supply;
```

All Rows Fetched: 12 in 0.003 seconds

	ID_SUPPLIER	ID_PROD	ID_MARKET
1	100	104	10
2	100	105	10
3	100	104	15
4	100	107	10
5	100	108	10
6	100	108	15
7	102	109	10
8	102	110	10
9	102	110	35
10	105	112	10
11	105	106	10
12	105	106	15

create table supply(

id_supplier number(3),

id_prod number(3),

id_market number (4),

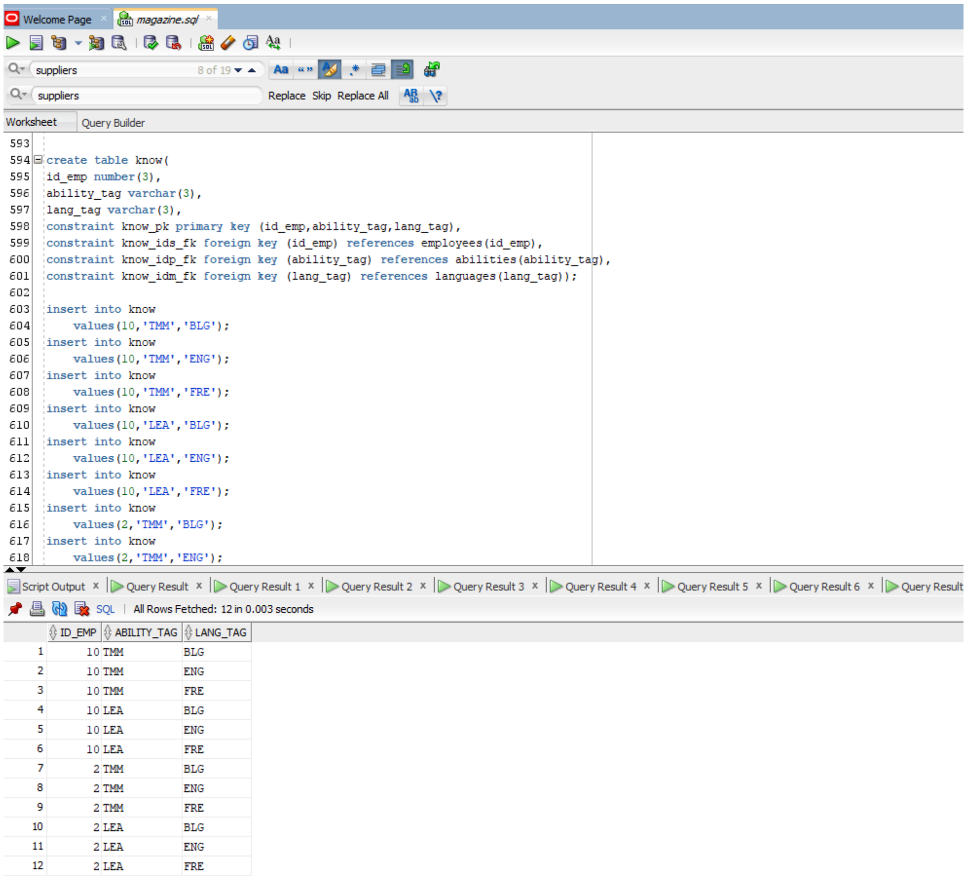
constraint supply_pk primary key (id_supplier,id_prod,id_market),

constraint supply_ids_fk foreign key (id_supplier) references suppliers(id_supplier),

constraint supply_idp_fk foreign key (id_prod) references products(id_prod),

constraint supply_idm_fk foreign key (id_market) references markets(id_market));

```
insert into supply

   values(100,104,10);

insert into supply

   values(100,105,10);

insert into supply

   values(100,104,15);

insert into supply

   values(100,107,10);

insert into supply

   values(100,108,10);

insert into supply

   values(100,108,15);

insert into supply

   values(102,109,10);

insert into supply

   values(102,110,10);

insert into supply

   values(102,110,35);

insert into supply

   values(105,112,10);

insert into supply

   values(105,106,10);

insert into supply

   values(105,106,15);


select * from supply;
```



```
create table know(
id_emp number(3),
ability_tag varchar(3),
lang_tag varchar(3),
constraint know_pk primary key (id_emp,ability_tag,lang_tag),
constraint know_ids_fk foreign key (id_emp) references employees(id_emp),
constraint know_idp_fk foreign key (ability_tag) references abilities(ability_tag),
constraint know_idm_fk foreign key (lang_tag) references languages(lang_tag));

insert into know
    values(10,'TMM','BLG');
insert into know
    values(10,'TMM','ENG');
insert into know
    values(10,'TMM','FRE');
insert into know
    values(10,'LEA','BLG');
insert into know
    values(10,'LEA','ENG');
insert into know
    values(10,'LEA','FRE');
insert into know
    values(2,'TMM','BLG');
insert into know
    values(2,'TMM','ENG');
```

All Rows Fetched: 12 in 0.003 seconds

	ID_EMP	ABILITY_TAG	LANG_TAG
1	10	TMM	BLG
2	10	TMM	ENG
3	10	TMM	FRE
4	10	LEA	BLG
5	10	LEA	ENG
6	10	LEA	FRE
7	2	TMM	BLG
8	2	TMM	ENG
9	2	TMM	FRE
10	2	LEA	BLG
11	2	LEA	ENG
12	2	LEA	FRE

```
create table know(

id_emp number(3),

ability_tag varchar(3),

lang_tag varchar(3),

constraint know_pk primary key (id_emp,ability_tag,lang_tag),

constraint know_ids_fk foreign key (id_emp) references employees(id_emp),

constraint know_idp_fk foreign key (ability_tag) references abilities(ability_tag),

constraint know_idm_fk foreign key (lang_tag) references languages(lang_tag));


insert into know

   values(10,'TMM','BLG');

insert into know

   values(10,'TMM','ENG');
```

```
insert into know
   values(10,'TMM','FRE');
insert into know
   values(10,'LEA','BLG');
insert into know
   values(10,'LEA','ENG');
insert into know
   values(10,'LEA','FRE');
insert into know
   values(2,'TMM','BLG');
insert into know
   values(2,'TMM','ENG');
insert into know
   values(2,'TMM','FRE');
insert into know
   values(2,'LEA','BLG');
insert into know
   values(2,'LEA','ENG');
insert into know
   values(2,'LEA','FRE');
```

Cele 5 cereri in sql(rezolvari si in fisierul cu sql):

1.Pentru fiecare magazin, sa se afiseze suprafata, strada, numele orasului si numele tarii si o coloana cu alias-ul dimenisune care sa ia valoarea opulent pentru suprafete de 400 m^2 si mare altfel si rezultatele se ordoneaza descrescator dupa suprafata

```
select m.surface, a.street_name, ci.city_name, co.state_name,
case m.surface
```

```
when 400  then 'opulent'
else 'mare'
end
as "dimensiune"
from markets m, addresses a, cities ci, countries co
where m.address_code = a.address_code
and a.city_tag = ci.city_tag
and ci.state_tag = co.state_tag
order by 1 desc;
```

2.Pentru angajatii care lucreaza in spania sa se afiseze de cate luni s-au angajat si ce comision au

```
with ang_spn as
(select *
from employees e, markets m, addresses a, cities c, countries s
where e.id_market = m.id_market and m.address_code = a.address_code
and a.city_tag = c.city_tag and c.state_tag = s.state_tag and s.state_name = initcap('spain'))
select trunc((sysdate-e.hire_date)/12) as "vechime",nvl(e.commission_quoef,0) as "comision"
from ang_spn e;

select * from employees;
```

3.Sa se afiseze id-ul magazinului si data angajarii celui mai vechi angajat de la magazinul respectiv pentru magazinele la care lucreaza minim 2 angajati, dar luna sa fie scrisa cu litere mici.

```
select m.id_market, lower(to_char(min(e.hire_date))) as "vechime maxima"

from markets m, employees e

where m.id_market = e.id_market

group by e.id_market

having count(*)>=2;
```

4.Numele distincte ale produselor si daca sunt sau nu lacto-ovo-vegetariene pentru cele furnizate la magazine in care lucreaza angajati din departamente cu id par

```
select distinct p.prod_name,

decode(p.prod_type, 'CSM', 'NU', 'MEA', 'NU', 'DA') as "lacto-ovo-vegan?"

from products p,supply s

where p.id_prod = s.id_prod

and s.id_market in (select e.id_market

                from employees e

                where e.id_depart in (select d.id_depart

                                from departments d

                                where mod(d.id_depart,2) = 0));
```

5. Ultima zi din din luna angajarii angajatilor al caror magazin arondat se afla in Bucuresti

```
select last_day(e.hire_date)

from employees e

where e.id_market in(select m.id_market

                from markets m

                where e.id_market = m.id_market

                and m.address_code in (select a.address_code

                                from addresses a
```

```
                              where a.address_code = m.address_code and a.city_tag = 'BUC'));
```

3 modificari de date prin subcereri(rezolvari si in fisierul cu sql)

6. Cresteti cu 10% pretul tuturor produselor lactate.

```
update products

set price = price + price * 10

where prod_type = 'DRY';


rollback;


--alternativ cu subcerere: pentru produsele ce au furnizori din Bulgaria

update products

set price = price + price * 10

where prod_type = 'DRY'

or id_prod in (select p.id_prod

        from suppliers s,products p, supply a

        where s.id_supplier = a.id_supplier and a.id_prod = p.id_prod and s.org_state = 'BLG');


rollback;
```

7. Stegreti din tabela de contain intrarile ce contin id-uri de bonuri in valoare totala de mai putin de 10 euro

```
delete

from contain

where receipt_id in (select r.receipt_id
            from contain r
            group by receipt_id
            having sum(quantity*id_prod)<10);

rollback;
```

8. Reduceti cu 100 de euro plata lunara catre fiecare companie de curatenie care curata mai rar de o data la 7 zile inclusiv

```
update maintainance_companies
set monthly_payment = monthly_payment - 100
where lower(company_type) = 'cleaning'
and id_comp in (select mc.id_comp
          from maintainance_companies mc
          where mc.cleaning_frequency <= 7);
rollback;
```

Division si Outer Join

9.Numele,magazinul,strada si orasul pentru angajatii care lucreaza la departamentul ‘Sales’

```
with
a1 as (select * from employees e full outer join departments d on e.id_depart =d.id_depart),
a2 as (select * from markets m full outer join addresses a on m.address_code = a.address_code),
a12 as (select * from a1 full outer join a2 on a1.id_market = a2.id_market)
select distinct last_name, surface, street_name, c.city_name
from a12,cities c
where departmentg_name like 'Sales' and c.city_tag = a12.city_tag;
```

10. Magazinele din Romania aprovizionate cu toate produse existente

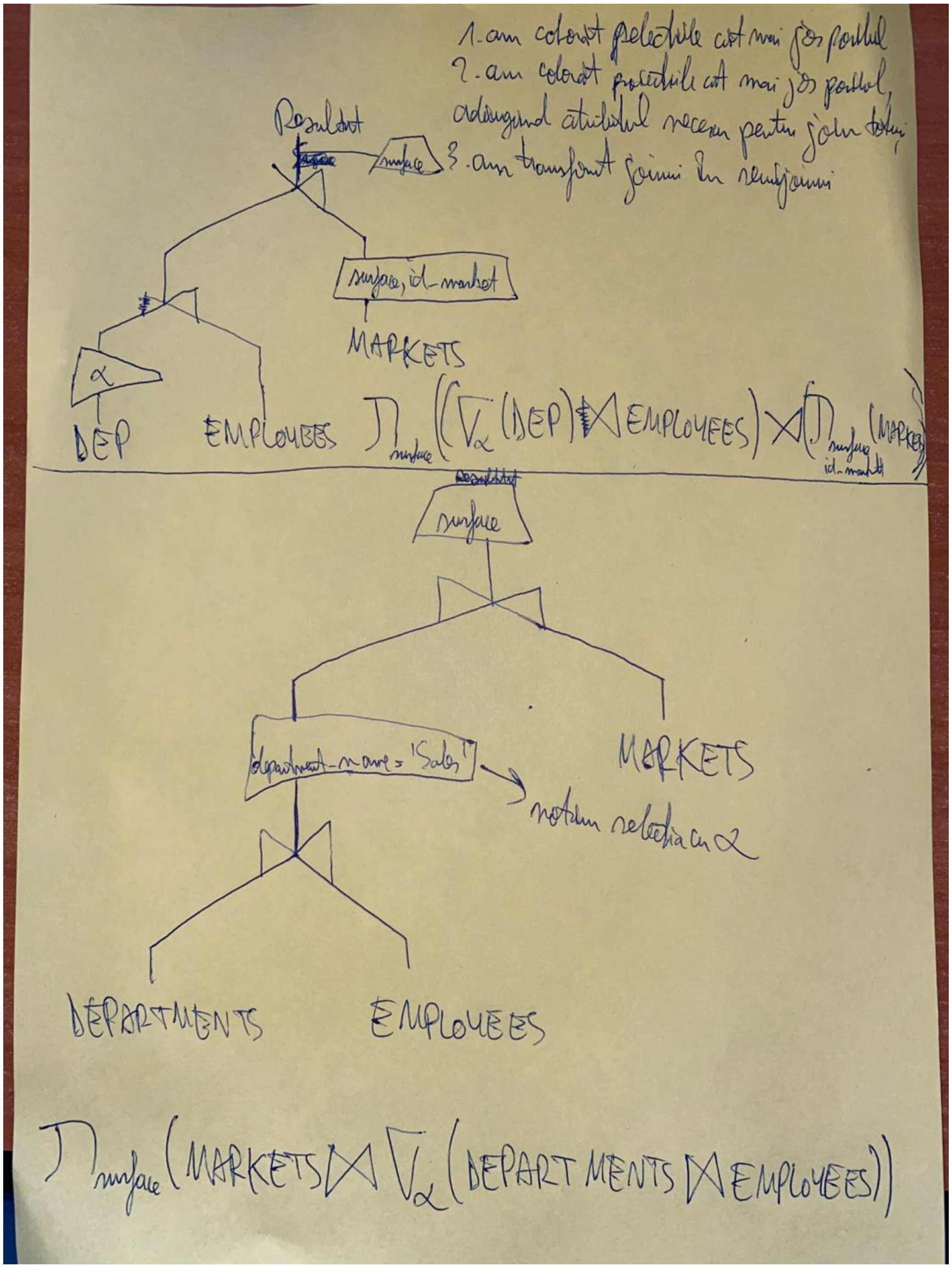
```
SELECT id_market
FROM   supply
MINUS
SELECT id_market
from(SELECT id_market, id_prod
    FROM (SELECT DISTINCT id_market
         FROM supply) t1, (SELECT id_prod
                        FROM products) t2
     MINUS
     SELECT id_market, id_prod
     FROM supply) t3;
```

11. Furnizorii care aprovizioneaza toate magazinele cu minim o categorie de produse

```
SELECT id_supplier
FROM supply
WHERE id_market IN (SELECT id_market
                FROM markets)
GROUP BY id_supplier
HAVING COUNT(id_market)=(SELECT COUNT(*)
                   FROM markets);
```

Optimizare

Se cere sa se afiseze suprafara fiecarui magazin care are angajati care fac parte din departamentul ‘Sales’. In partea de sus a pozei este varianta optimizata iar ambele coduri in sql sunt si in documentul special



1. am coborat selectiile cat mai jos posibil
2. am coborat proiectiile cat mai jos posibil, adaugand atributul necesar pentru join-uri;
3. am transformat joinii in semijoinii

$\Pi_{surface}((\sigma_{\alpha}(DEP) \bowtie EMPLOYEES) \ltimes (\Pi_{surface, id\_market}(MARKETS)))$

$\Pi_{surface}(MARKETS \bowtie \sigma_{\alpha}(DEPARTMENTS \bowtie EMPLOYEES))$

```
--neoptimizat
select surface
from markets, (select *
        from departments d, employees e
        where d.id_depart = e.id_depart

        and departmentg_name = 'Sales') aux

where markets.id_market = aux.id_market;


--optimizat

select surface

from (select surface,id_market from markets) a, (select *

                                    from employees

                                    where id_depart = (select id_depart

                                                from departments

                                                where departmentg_name = 'Sales')) b

where a.id_market = b.id_market;
```

FN superioare

FNBC este satisfacuta daca si numai daca fiecare determinant dintr-o relatie este cheie candidat. De exemplu, FNBC nu ar fi satisfacuta daca presupunem ca, in regulile modelului conceptual, am avea ca un magazin poate fi aprovizionat de un singur furnizor. Atunci, tabelul SUPPLY ar arata in felul urmator:

Id_prod#	Id_supplier#	Id_market
37	55	120
24	60	130
42	55	120

FNBC nu este satisfacuta deorece exista dependenta id_market => id_supplier#, asa ca se aplica regula Casey-Delobel si se partitioneaza tabelul in felul urmator:

Id_market#	Id_supplier
120	55
130	60
Id_market#	Id_prod#
120	37
120	42

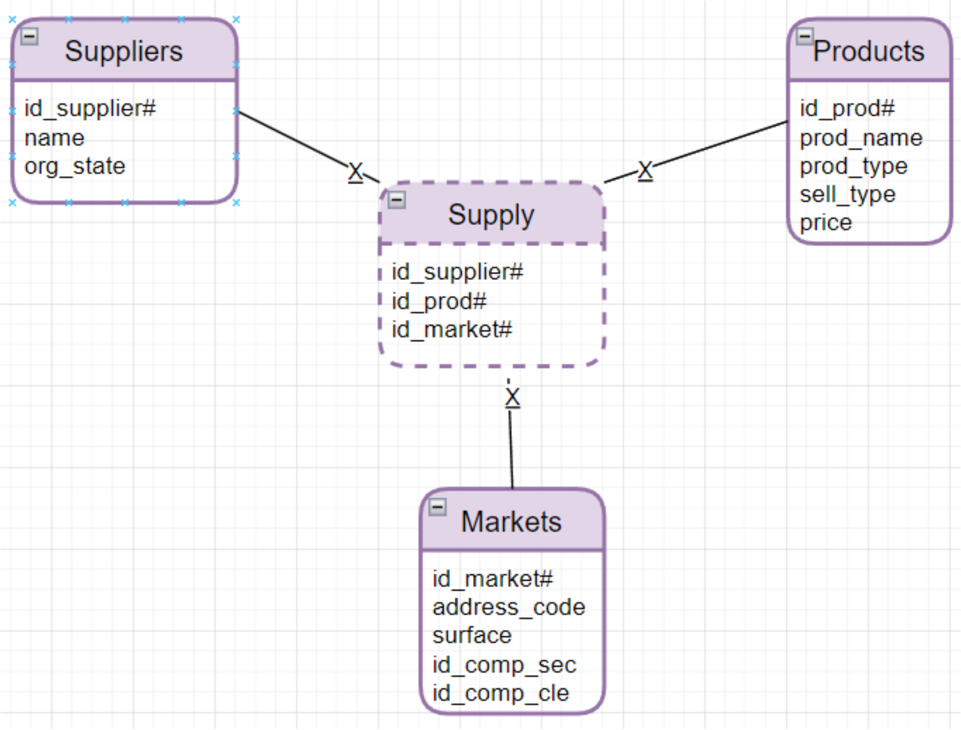
Informatia initiala se poate reconstrui prin aplicarea unui join intre cele doua tabele, dupa id_market.

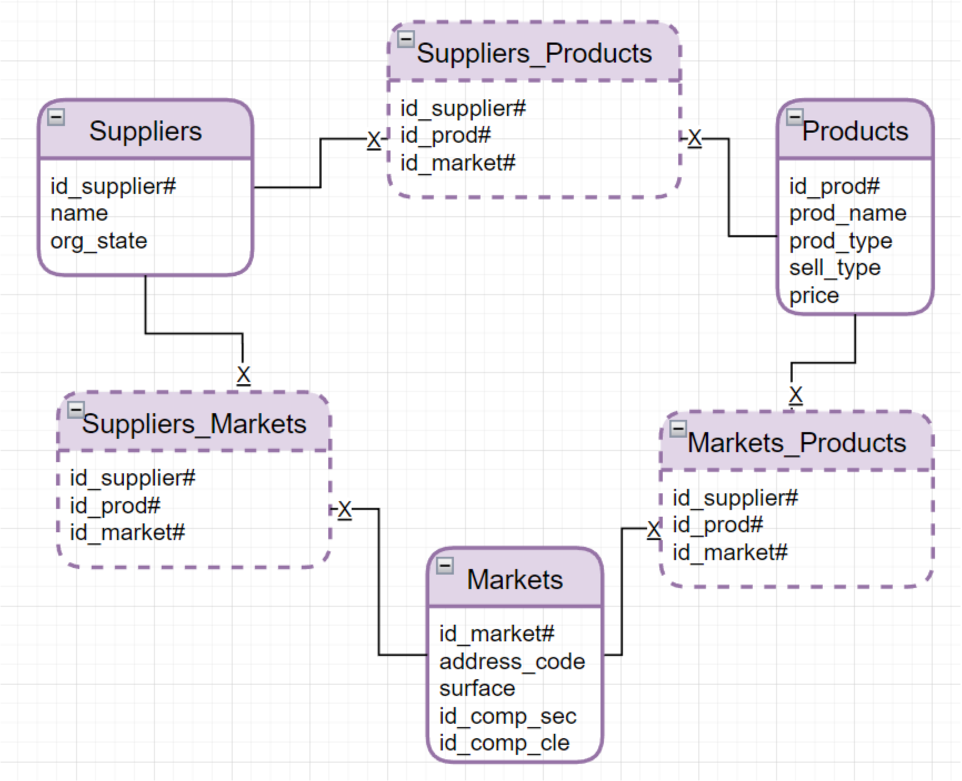
FN4 nu este satisfacuta in diagrama (problema corespondentei la tabelul KNOW) si aplicam Cassey-Delobel si spargem tabelul. Astfel, ar trebui ca in loc sa fie cum este acum in diagrama, sa fie asa:

ID_ANGAJAT	COD_LIMBA
100	ENG
100	DEU
101	ENG

ID_ANGAJAT	COD_ABILITATE
100	TMM
101	TMM
101	LEA

FN5 nu este respectata, deorece relatia ternara SUPPLY prezinta redundante in cadrul relatiilor many to many(m:n) ce o definesc. Astfel, relatia ternara se poate descompune in 3 relatii binare ciclice fara pierdere de informatie in felul urmator:





Deoarece relatia ternara a fost inlocuita cu 3 relatii ciclice fara pierdere de informatie , concluzionam ca aceasta era redundanta, deci incalca FN5.

Denormalizarea

Denormalizarea reprezinta includerea unor informatii redundante in relatii cu scopul eficientizarii timpului de executie. Chiar daca aceste informatii complete se pot obtine aplicand join-uri multiple, ele pot fi incluse in tabele, chiar daca rezulta in memorie suplimentara consumata, deoarece pot reprezenta informatii folosite des in cereri. De exemplu, relatia Products se poate transforma prin denormalizare in Products_denormalizat(id_prod#, prod_name, prod_type, sell_type, price, origin_state), unde origin_state este foreign key din Countries. Astfel, tara de origine a oricarui produs este mult mai usor de obtinut decat in cazul initial, in care ar fi fost necesare join-uri intre tabelele Products, Product_Types, Suppliers si Countries, economisind timp de executie.

Id_prod#	Prod_name	Prod_type	Sell_type	Price	Origin_state
100	cartofi	legume	ENG	2.5	TUR
101	balsam	cosmetice	END	10	FRA

102	bibelou	decoratiuni	END	35	CHN