

```

create database ethazi4
collate utf8mb4_spanish_ci;
use ethazi4; create table
Local
(NIF char(9) primary key,
Nombre varchar(20) not null,
Tipo enum('Bar', 'Restaurante', 'Cafetería'),
Propietario varchar(20) not null,
Direccion varchar(30)
);
create table Usuario
(DNI char(9) primary key,
Nombre varchar(20) not null,
Apellido varchar(40) not null,
Contrasenya blob not null, NIF char(9) not null, constraint fk_NIF_Usuario foreign key
(NIF) references Local (NIF) on update cascade on delete cascade
);
create table Fabricante
(Nombre varchar(20) primary key,
Tiempo int not null
);
create table Producto
(Nombre varchar(50) primary key ,
Fec_Cad date not null,
Tipo enum('Bebida', 'Comida', 'Plato'),
Precio_Venta float not null,
Precio_Compra float not null, N_Fabricante varchar(20) not null, constraint
fk_N_Fabricante_Producto foreign key (N_Fabricante) references
Fabricante(Nombre) on update cascade on delete cascade
);

```

```

create table Operaciones
(NumTrans int primary key,
Total_Operaciones float,
NIF char(9),
TipoOperacion set ('F','P','T','A','C'),
constraint fk_NIF_Operaciones foreign key (NIF) references Local (NIF)
);create table Pedidos
(NumTrans int primary key, Domicilio varchar(20), constraint
fk_NumTrans_Pedidos foreign key (NumTrans) references Operaciones
(NumTrans) on update cascade on delete cascade
);
create table NIFFactura
(NIF char(9) primary key,
Nom_Des varchar(20) not null,
Apellidos_Des varchar(30) not null
);
create table Factura
(NumTrans int primary key, NIF char(9) not null, constraint
fk_NumTrans_Factura foreign key (NumTrans) references Operaciones
(NumTrans) on update cascade on delete cascade, constraint fk_NIF_Factura foreign key
(NIF) references NIFFactura (NIF) on update cascade on delete cascade
);
create table Tiene
(NomProducto varchar(50),
NumTrans int not null,
N_Unidades int not null,
Precio float not null,
Fecha date,
TipoOperacion set ('F','P','T','A','C'),
constraint fk_NomProducto_Tiene foreign key(NomProducto) references Producto (Nombre)
on update cascade on delete cascade, constraint fk_NumTrans_Tiene foreign key(NumTrans)

```

```

references Operaciones(NumTrans) on update cascade on delete cascade, constraint
pk_Tiene primary key(NomProducto,NumTrans)
);
create table Vende
(NomProducto varchar(50),
Stock float not null, NIFLocal char(9), constraint fk_NomProducto_Vende foreign
key (NomProducto) references Producto
(Nombre) on update cascade on delete cascade, constraint fk_NIFLocal_Vende
foreign key (NIFLocal) references Local (NIF) on update cascade on delete cascade,
constraint pk_Vende primary key(NomProducto,NIFLocal)
);
create table Aprovisionamiento( NumTrans int primary key, Nom_Fabricante char(30) not
null, constraint fk_NumTrans_Aprovisionamiento foreign key (NumTrans) references
Operaciones(NumTrans) on update cascade on delete cascade, constraint
fk_Nom_Fabricante_Aprovisionamiento foreign key (Nom_Fabricante) references
Fabricante (Nombre) on delete cascade on update cascade
);
create table Comanda
(
NumTrans int primary key, constraint fk_NumTrans_Comanda foreign key
(NumTrans) references Operaciones
(NumTrans) on update cascade on delete cascade
);
create table Plato
(
Cod_Plato char(10) primary key,
TipoDePlato enum ('Normal','Vegetariano','Vegano'),
Nombre varchar(50) not null,
TipoPosicion enum ('Primero', 'Segundo', 'Postre'),
Precio float not null
);

```

```

create table Incluye
(
    Num_Trans int not null,
    Cod_Plato char(10) not null,
    Num_Platos int not null,
    PrecioActual float not null, constraint fk_NumTrans_Incluye foreign key (Num_Trans)
references Comanda (NumTrans) on update cascade on delete cascade, constraint
fk_Cod_Plato_Incluye foreign key (Cod_Plato) references Plato (Cod_Plato) on update
cascade on delete cascade, constraint pk_Incluye primary key (Num_Trans,Cod_Plato)
);

create table Ofrece (NIFLocal char(9), Cod_Plato char(10), constraint fk_NIF_Ofrece
foreign key (NIFLocal) references Local (NIF) on update cascade on delete cascade,
constraint fk_Cod_Plato_Ofrece foreign key (Cod_Plato) references Plato (Cod_Plato) on
update cascade on delete cascade, constraint pk_Ofrece primary key (NIFLocal,
Cod_Plato)
);create table Ingrediente
(
    Nombre varchar(50) primary key,
    HechoCon set ('Gluten', 'Mariscos', 'Frutos secos')
);

create table Contiene
(
    Nom_Ingrediente varchar(50), Cod_Plato char(10), constraint
fk_Nom_Ingrediente_Contiene foreign key (Nom_Ingrediente) references Ingrediente
(Nombre) on update cascade on delete cascade, constraint fk_Cod_Plato_Contiene foreign
key (Cod_Plato) references Plato (Cod_Plato) on update cascade on delete cascade,
constraint pk_Ofrece primary key (Nom_Ingrediente, Cod_Plato)
);

create table Fecha
(
    Fecha datetime primary key

```

```

);

create table HistoricoEstablecimientoSemanal
(
    Prob float default 0,
    Fecha datetime,
    Nom1 varchar(50), Nom2 varchar(50), constraint fk_Fech foreign key (Fecha) references
    Fecha (Fecha) on update cascade on delete cascade, constraint fk_Nom1 foreign key (Nom1)
    references Producto (Nombre) on update cascade on delete cascade, constraint fk_Nom2
    foreign key (Nom2) references Producto (Nombre) on update cascade on delete cascade,
    constraint pk_HisEstSemTd primary key (Fecha, Nom1, Nom2)
);

create table HistoricoEstablecimientoSemanalTd
(
    Prob float default 0,
    Fecha datetime,
    Nom1 varchar(50),
    Nom2 varchar(50),
    NifLocal char(9),
    constraint fk_Fec foreign key (Fecha) references Fecha (Fecha) on update cascade on
    delete cascade,
    constraint fk_Nombre1 foreign key (Nom1) references Vende (NomProducto) on update
    cascade on delete cascade,
    constraint fk_Nombre2 foreign key (Nom2) references Vende (NomProducto) on update
    cascade on delete cascade, constraint fk_Niflocal foreign key (NifLocal) references Vende
    (NIFLocal) on update cascade on delete cascade, constraint pk_HisEstSemTd primary key
    (Fecha, Nom1, Nom2, NifLocal)
);

insert into Local values('12345678B', 'Uria', 'Bar', 'Pajarito', 'Arturo Campión'); insert into Local
values('12345678R', 'Chino Ron City', 'Restaurante', 'Xiansheng', 'Pintores Arrúe'); insert into
Local values('12345678C', 'Gazteleku Berria', 'Cafetería', 'Roberto', 'Oñatiko Unibertsitatea');
insert into Fabricante values ('Amazon', 1); insert into Fabricante values ('Nescafé', 1); insert

```

into Fabricante values ('Bezoya',1); insert into Fabricante values ('Don Simon',1); insert into Fabricante values ('1906' ,1); insert into Fabricante values ('Bizkaiko Txakolina',1); insert into Producto values('Sidra','2022-02-02' , 'Bebida', 2, 1.50, 'Amazon'); insert into Producto values('Kafea','2022-02-02' , 'Bebida', 1.20, 0.70, 'Nescafé');

insert into Producto values('Ura','2022-02-02' , 'Bebida', 1, 0.50, 'Bezoya'); insert into Producto values('Zukua','2022-02-02' , 'Bebida', 1.50, 1, 'Don Simon'); insert into Producto values('Patata tortilla','2022-02-02' , 'Comida', 1.50, 1, 'Amazon'); insert into Producto values('Ardoa','2022-02-02' , 'Bebida', 1.20, 0.70, 'Amazon'); insert into Producto values('Txakoli','2022-02-02' , 'Bebida', 1.50, 1, 'Amazon'); insert into Producto values('Gilda','2022-02-02' , 'Comida', 1, 0.50, 'Amazon'); insert into Producto values('Garagardoa','2022-02-02' , 'Bebida', 1.25, 0.70, 'Amazon'); insert into Producto values('Colacao','2022-02-02' , 'Bebida', 1, 0.50, 'Amazon'); insert into Producto values('Falafel','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Producto values('Nuggets con patata','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Producto values('Lentejas','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Producto values('Garbanzos','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Producto values('Pure de verduras','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Producto values('Croquetas de espinaca','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Producto values('Carbonara Vegana','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Producto values('Sopa de cebolla Francesa','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Producto values('Curry de anacardos','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Producto values('Bollo suizo vegano','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Producto values('Flan de caqui','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Producto values('Arroz con leche','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Producto values('Entrecot','2022-02-02' , 'Plato', 3, 3, 'Amazon'); insert into Vende values ('Sidra', 1000, '12345678C'); insert into Vende values ('Sidra', 1000, '12345678B'); insert into Vende values ('Sidra', 10, '12345678R'); insert into Vende values ('Kafea', 10, '12345678C'); insert into Vende values ('Kafea', 1000, '12345678B'); insert into Vende values ('Kafea', 10, '12345678R'); insert into Vende values ('Ura', 10, '12345678C'); insert into Vende values ('Ura', 1000, '12345678B'); insert into Vende values ('Ura', 10, '12345678R'); insert into Vende values ('Zukua', 10, '12345678C');

```

insert into Vende values ('Zukua', 1000, '12345678B'); insert
into Vende values ('Zukua', 10, '12345678R'); insert into
Vende values ('Patata Tortilla', 10, '12345678C'); insert into
Vende values ('Patata Tortilla', 1000, '12345678B'); insert into
Vende values ('Patata Tortilla', 10, '12345678R'); insert into
Vende values ('Ardoa', 1000, '12345678R'); insert into Vende
values ('Ardoa', 1000, '12345678C'); insert into Vende values
('Ardoa', 10, '12345678B'); insert into Vende values ('Txakoli',
10, '12345678R'); insert into Vende values ('Txakoli', 1000,
'12345678C'); insert into Vende values ('Txakoli', 1000,
'12345678B'); insert into Vende values ('Gilda', 10,
'12345678R'); insert into Vende values ('Gilda', 1000,
'12345678C'); insert into Vende values ('Gilda', 1000,
'12345678B'); insert into Vende values ('Garagardoa', 10,
'12345678R'); insert into Vende values ('Garagardoa', 1000,
'12345678C'); insert into Vende values ('Garagardoa', 1000,
'12345678B'); insert into Vende values ('Colacao', 10,
'12345678R'); insert into Vende values ('Colacao', 1000,
'12345678C'); insert into Vende values ('Colacao', 1000,
'12345678B'); insert into ingrediente values ('Pan','Gluten');
insert into ingrediente values ('Tomate', ''); insert into
ingrediente values ('Garbanzo',''); insert into ingrediente
values ('Ajo',''); insert into ingrediente values ('Cebolla','');
insert into ingrediente values ('Cilantro',''); insert into
ingrediente values ('Lentejas', ''); insert into ingrediente
values ('Chorizo',''); insert into ingrediente values ('Patata','');
insert into ingrediente values ('Zanahoria',''); insert into
ingrediente values ('Guisantes','');

insert into ingrediente values ('Pan rallado','Gluten'); insert into ingrediente values
('Agua', ''); insert into ingrediente values ('Sal',''); insert into ingrediente values

```

('Calabacin',''); insert into ingrediente values ('Harina',''); insert into ingrediente values ('Espinacas',''); insert into ingrediente values ('Pasta','Gluten'); insert into ingrediente values ('Tofu','');insert into ingrediente values ('Leche vegetal',''); insert into ingrediente values ('Anacardos',''); insert into ingrediente values ('Arroz',''); insert into ingrediente values ('Curry',''); insert into ingrediente values ('Margarina',''); insert into ingrediente values ('Leche de soja',''); insert into ingrediente values ('Azucar',''); insert into ingrediente values ('Levadura',''); insert into ingrediente values ('Harina de trigo',''); insert into ingrediente values ('Leche',''); insert into ingrediente values ('Canela',''); insert into ingrediente values ('Caqui',''); insert into ingrediente values ('Huevo',''); insert into ingrediente values ('Caramelo liquido',''); insert into ingrediente values ('Carne',''); insert into ingrediente values ('Patatas fritas',''); insert into plato values (1,'Vegetariano','Falafel','Segundo',3); insert into contiene values ('Pan',1); insert into contiene values ('Tomate',1); insert into contiene values ('Garbanzo',1); insert into contiene values ('Ajo',1); insert into contiene values ('Cebolla',1); insert into contiene values ('Cilantro',1); insert into plato values (2,'Vegano','Nuggets con patata','Segundo',3);

insert into contiene values ('Cebolla',2); insert into contiene values ('Patata',2); insert into contiene values ('Zanahoria',2); insert into contiene values ('Guisantes',2); insert into contiene values ('Pan rallado',2); insert into plato values (3,'Normal','Lentejas','Primero',3); insert into contiene values ('Pan',3); insert into contiene values ('Lentejas',3); insert into contiene values ('Chorizo',3); insert into contiene values ('Patata',3); insert into plato values (4,'Normal','Garbanzos','Primero',3); insert into contiene values ('Pan',4); insert into contiene values ('Garbanzo',4); insert into contiene values ('Patata',4); insert into contiene values ('Chorizo',4); insert into plato values (5,'Normal','Pure de verduras','Primero',3); insert into contiene values ('Pan',5); insert into contiene values ('Zanahoria',5);insert into contiene values ('Calabacin',5); insert into contiene values ('cebolla',5); insert into plato values (6,'Vegano','Croquetas de espinaca','Segundo',3); insert into contiene values ('Pan',6); insert into contiene values ('Harina',6); insert into contiene values ('Pan rallado',6); insert into contiene



```

values ('Espinacas',6); insert into plato values (7, 'Vegano', 'Carbonara
Vegana','Primero',3); insert into contiene values ('Pan',7); insert into contiene
values ('Pasta',7); insert into contiene values ('Tofu',7); insert into contiene values
('Leche vegetal',7); insert into plato values (8, 'Vegetariano', 'Sopa de cebolla
Francesa','Primero',3); insert into contiene values ('Pan',8);

insert into contiene values ('Agua',8); insert into contiene values
('Cebolla',8); insert into contiene values ('Sal',8); insert into plato values
(9, 'Vegetariano', 'Curry de anacardos','Primero',3); insert into contiene
values ('Pan',9); insert into contiene values ('Anacardos',9); insert into
contiene values ('Arroz',9); insert into contiene values ('Curry',9); insert
into plato values (10, 'Vegano', 'Bollo suizo vegano', 'Postre',3); insert into
contiene values ('Margarina',10); insert into contiene values ('Leche de
soja',10); insert into contiene values ('Levadura',10); insert into contiene
values ('Azucar',10); insert into contiene values ('Harina de trigo',10);
insert into contiene values ('Agua',10); insert into contiene values
('Sal',10); insert plato values(11 , 'Vegetariano', 'Flan de caqui', 'Postre',3);
insert into contiene values ('Caqui',11); insert into contiene values
('Huevo',11); insert into contiene values ('Leche',11); insert into contiene
values ('Caramelo liquido',11); insert into contiene values ('Azucar',11);
insert into plato values (12, 'Normal', 'Arroz con leche','Postre',3); insert
into contiene values ('Arroz',12); insert into contiene values ('Leche',12);
insert into contiene values ('Azucar',12);

insert into contiene values ('Canela',12);insert into plato values (13, 'Normal',
'Entrecot','Segundo',3); insert into
contiene values ('Carne',13); insert into
contiene values ('Sal',13); insert into
contiene values ('Ajo',13); insert into
contiene values ('Patatas fritas',13);

delimiter // create trigger
modificar_stock after insert on tiene

```

```

for each row begin if (select
TipoOperacion from operaciones
where NumTrans = NEW.NumTrans) !=
'A' then update vende set Stock =
Stock - NEW.N_Unidades where
NIFLocal = (select NIF from
operaciones where NumTrans =
NEW.NumTrans) and NomProducto =
NEW.NomProducto;
end if;

end;// delimiter // create trigger modificar_stock2 after update on tiene for each row
begin update vende set Stock = Stock - (NEW.N_Unidades-OLD.N_Unidades) where
NIFLocal = (select NIF from operaciones where NumTrans = NEW.NumTrans) and
NomProducto = NEW.NomProducto; end;// delimiter // create trigger
encriptar_contrasenya before insert on usuario for each row begin declare
contrasenya varchar(20) default (select Contrasenya from Usuario where DNI =
NEW.DNI); set Contrasenya = aes_encrypt(contrasenya,'elorrieta'); end;//
DELIMITER //

create event llegada_apro on schedule every
1 day starts current_date() on completion
preserve do begin
DECLARE fec date;
declare fin bool default 0;

declare zbk integer default 0; DECLARE C1 CURSOR FOR select Fecha,NumTrans from
operaciones where
TipoOperacion = 'A' and Fecha = DATE_ADD(current_date(),INTERVAL -3 DAY);

declare continue handler for not found set fin = 1; open C1; fetch c1 into
fec,zbk; while fin = 0 do update vende set Stock = Stock + (select N_Unidades
from tiene where NumTrans = zbk) where NIFLocal = (select NIF from
operaciones where
NumTrans = zbk) and NomProducto = (select NomProducto from tiene where NumTrans =

```

```

zbc); fetch c1 into
fec,zbc; end while;
close C1; end//

DELIMITER ; DELIMITER // create function importeTotal() returns float reads sql data
begin declare totala float; select sum(Precio) into totala from tiene where NumTrans =
(select max(NumTrans) from
tiene); return
totala; end//

DELIMITER ;
DELIMITER //

CREATE FUNCTION funtzioProbabilitateLokala(prodA varchar(50), prodB varchar(50),nif
char(9))
RETURNS float READS
SQL DATA
begin declare emaitza float; declare Pab float; declare tot1 int; declare tot2 int; declare Pa
float; declare Pb float; select count(*) into Pab from tiene a join operaciones b on
a.numtrans=b.numtrans where
a.NumTrans in (select b.numtrans from tiene a join
operaciones b on a.numtrans=b.numtrans
where NomProducto=prodA and b.NIF=nif) and NomProducto=prodB and b.NIF=nif;select
count(a.NumTrans) into tot1 from tiene a join operaciones b on
a.numtrans=b.numtrans where NomProducto=prodA and b.NIF=nif; select count(*) into
Pa from tiene a join operaciones b on a.numtrans=b.numtrans where
a.NumTrans in (select distinct a.NumTrans from tiene a join operaciones b on
a.numtrans=b.numtrans where b.NIF=nif) and NomProducto=prodA and b.NIF=nif; select
count(*) into Pb from tiene a join operaciones b on a.numtrans=b.numtrans where
a.NumTrans in (select distinct a.NumTrans from tiene a join operaciones b on
a.numtrans=b.numtrans where b.NIF=nif) and NomProducto=prodB and b.NIF=nif; select
max(a.NumTrans) into tot2 from tiene a join operaciones b on a.numtrans=b.numtrans
where b.NIF=nif; set emaitza=((Pab/tot1)*(Pa/tot2))/(Pb/tot2); return emaitza; end//

```

DELIMITER;

DELIMITER //

```
CREATE FUNCTION funtzioprobabilitateOrokorra(prodA varchar(50), prodB varchar(50))
RETURNS float READS SQL DATA begin declare emaitza float; declare Pab float; declare tot1
int; declare tot2 int; declare Pa float; declare Pb float; select count(*) into Pab from tiene
where NumTrans in (select NumTrans from tiene where NomProducto=ProdA) and
NomProducto=prodB; select count(NumTrans) into tot1 from tiene where
NomProducto=prodA; select count(*) into Pa from tiene where NumTrans in (select
distinct NumTrans from tiene)
and NomProducto=prodA; select count(*) into Pb from tiene where NumTrans in (select
distinct NumTrans from tiene)
and NomProducto=prodB;
select max(NumTrans) into tot2 from tiene;
set emaitza=((Pab/tot1)*(Pa/tot2))/(Pb/tot2);
return emaitza; end//
```