

By Sergio Diez, Mario Celada, Sara Gómez

## D) SQL (Transact SQL and PL/SQL (i.e., Oracle) scripts)

### 8. SQL sentences in Microsoft SQL

**Sara Tables:** PRODUCTS, PACKAGINS, ARRIVALPOINTS, GUIDELANGUAGE, GAME, FILMS, CONSOLES, PERIPHERALS, NATGAM, NATFIL, INTCON, INTPER, INTERNATIONAL, NATIONAL, SUPPLIERS, SUPPHONENUM, SUPEMAIL.

**Sergio Tables:** CLIENTS, SUGGESTIONS, DISABILITIES, DISEASES, EVECLI, EVENTS, NONPARTNERS, PARTNERS, OFFERS, BUY, RENT.

**Mario Tables:** EMPLOYEES, SALARIES, HOLIDAYS, WORKOFFPERIODS, ALLERGIES, EMDISEASES, EME-MAIL, EMPHONE NUMBER, EMPHOL, EMPWOF, EMPPRO, SALESMANS, SUPPORTMANS, SECURITYMANS, SOFTWARE\_ENGINEERS, DELIVERYMANS, ACCOUNTMANAGERS, ACCEXP, SOFWEB, THIEVES, WEBPAGES, EXPENSES, VEHICLES.

```
CREATE TABLE SUPPLIERS(  
SupplierID int,  
Fax decimal(9),  
Company_Name varchar(20) not null,  
Type varchar(8) not null,  
constraint PK_SUPPLIERS primary key(SupplierID)  
);
```

```
CREATE TABLE SUPEMAIL(  
SupplierID int,  
EmailAdress varchar(30),  
constraint FK_EMAIL_SupplierID foreign key(SupplierID) references SUPPLIERS,  
constraint PK_EMAIL primary key(SupplierID,EmailAdress)  
);
```

```
CREATE TABLE SUPPHONENUMBER(  
SupplierID int,  
PhoneNumber int,
```

```
constraint FK_PHONENUMBER_SupplierID foreign key(SupplierID) references
SUPPLIERS,
constraint PK_PHONENUMBER primary key(SupplierID,PhoneNumber)
);
```

```
CREATE TABLE INTERNATIONAL(
SupplierID int, --nvarchar?
constraint FK_INTERNATIONAL_SupplierID foreign key(SupplierID) references
SUPPLIERS,
constraint PK_INTERNATIONAL primary key(SupplierID)
);
```

```
CREATE TABLE NATIONALs( -- la s es para evitar que se coloree en transact
SupplierID int,
constraint FK_NATIONAL_SupplierID foreign key(SupplierID) references SUPPLIERS,
constraint PK_NATIONAL primary key(SupplierID)
);
```

```
CREATE TABLE ARRIVALPOINTS(
APID int,
Shopping_Centre varchar(3) ,
Country varchar(15) not null,
Door# int,
Type varchar(15),
Zip int,
Street int not null,
City varchar(20) not null,
constraint Shopping_Centre_VALUE CHECK (Shopping_Centre in ('YES', 'NO')),
constraint PK_ARRIVALPOINTS primary key(APID)
);
```

```
CREATE TABLE EMPLOYEES(
SSN char(13),
```

```

DNI char(9) not null, --cadena de texto fija
Name varchar(20) not null, --cadena de texto variable
Surname varchar(20) not null,
City varchar(50),
Zip numeric(5,0),
Street varchar(50),
Age decimal(3,0) null,
Gender varchar(1),
Birthdate date not null,
HiringDate date not null,
Antiquity decimal(2,0) null,
Type varchar(17) not null, --el numero que definimos como maximo de la cadena seria el mas
largode los tipos que puede haber de empleados
constraint PK_EMPLOYEES_SSN primary key(SSN),
constraint U_EMPLOYEES_DNI UNIQUE(DNI),
constraint CHK_EMPLOYEES_Type CHECK (Type
IN('ACCOUNTMANAGERS','DELIVERYMANS','SOFTWAREENGINEERS','SECURITY
MANS','SUPPORTMANS','SALESMANS'))
);

```

```

CREATE TABLE SECURITYMANS(
SSN char(13),
Specialisation varchar(15) not null default 'NOSPECIALI.',
constraint PK_SECURITYMANS_SSN primary key(SSN),
constraint FK_SECURITYMANS_SSN foreign key(SSN) references EMPLOYEES
);

```

```

CREATE TABLE THIEVES(
ThieveID int,
--Picture BLOB(16M) formato para imagenes
Nationality VARCHAR(30),
HairColour VARCHAR(15),
Gender VARCHAR(1) default 'O',
Dangerous int default 0,
SSN CHAR(13) not null,

```

```
CaptureDateHour DATE not null,  
constraint Violent_Value CHECK (Dangerous in ('YES','NO')),  
constraint Gender_Values_T CHECK (Gender in ('M', 'F', 'O')),  
constraint PK_THIEVES primary key(ThieveID),  
constraint FK_SSN_THIEVES foreign key(SSN) references SECURITYMANS  
);
```

```
CREATE TABLE PRODUCTS(  
UPC int,  
Origing_Country varchar(15),  
Discount char(3),  
Warranty int default '40',  
GuideLanguage int not null,  
Price_PhysicalStore int,  
Price_Online int,  
Day int,  
Month int,  
Year int,  
Type varchar(8) not null,  
APID int,  
SendingTime date default getdate(),  
ThieveID int,  
constraint FK_PRODUCTS_APID foreign key(APID) references ARRIVALPOINTS,  
constraint FK_PRODUCTS_ThieveID foreign key(ThieveID) references THIEVES,  
constraint PK_PRODUCTS_UPC primary key(UPC),  
constraint CHK_PRODUCTS_Month CHECK(Month < 13),  
constraint CHK_PRODUCTS_PriceCheckPhysical CHECK(Price_PhysicalStore > 0),  
constraint CHK_PRODUCTS_PriceCheckOnline CHECK(Price_Online > 0),  
constraint CHK_PRODUCTS_Year CHECK(Year > 2000)  
);
```

```
CREATE TABLE PERIPHERALS(  
UPC int,  
Type varchar(15) not null,  
constraint FK_PERIPHERALS_UPC foreign key(UPC) references PRODUCTS,
```

```
constraint PK_PERIPHERALS primary key(UPC)
);
```

```
CREATE TABLE INTPER(
UPC int,
Price varchar(15) not null,
SupplierID int,
constraint FK_INTPER_UPC foreign key(UPC) references PERIPHERALS,
constraint FK_INTPER_SupplierID foreign key(SupplierID) references INTERNATIONAL,
constraint PK_INTPER primary key(UPC,SupplierID)
);
```

```
CREATE TABLE CONSOLES(
UPC int,
Name varchar(15) not null,
Brand varchar(15) not null,
constraint FK_CONSOLES_UPC foreign key(UPC) references PRODUCTS,
constraint PK_CONSOLES primary key(UPC)
);
```

```
CREATE TABLE INTCON(
UPC int,
Price varchar(15) not null,
SupplierID int,
constraint FK_INTCON_UPC foreign key(UPC) references CONSOLES,
constraint FK_INTCON_SupplierID foreign key(SupplierID) references INTERNATIONAL,
constraint PK_INTCON primary key(UPC,SupplierID)
);
```

```
CREATE TABLE FILMS(
UPC int,
Tittle varchar(15) not null,
Genre varchar(15) not null,
constraint FK_FILMS_UPC foreign key(UPC) references PRODUCTS,
constraint PK_FILMS primary key(UPC)
```

);

```
CREATE TABLE NATFIL(  
  UPC int,  
  Price varchar(15) not null,  
  SupplierID int,  
  constraint FK_NATFIL_UPC foreign key(UPC) references FILMS,  
  constraint FK_NATFIL_SupplierID foreign key(SupplierID) references NATIONALs,  
  constraint PK_NATFIL primary key(UPC,SupplierID)  
);
```

```
CREATE TABLE GAMES(  
  UPC int,  
  Type varchar(15) not null,  
  Platform varchar(15) not null,  
  constraint FK_GAMES_UPC foreign key(UPC) references PRODUCTS,  
  constraint PK_GAMES primary key(UPC)  
);
```

```
CREATE TABLE NATGAM(  
  UPC int,  
  Price varchar(15) not null,  
  SupplierID int,  
  constraint FK_NATGAM_UPC foreign key(UPC) references GAMES,  
  constraint FK_NATGAM_SupplierID foreign key(SupplierID) references NATIONALs,  
  constraint PK_NATGAM primary key(UPC,SupplierID)  
);
```

```
CREATE TABLE GUIDELANGUAGE(  
  UPC int ,  
  ProductManualLanguage varchar(20) not null,  
  constraint FK_GUIDELANGUAGE_UPC foreign key(UPC) references PRODUCTS,  
  constraint PK_GUIDELANGUAGE primary key(UPC,ProductManualLanguage)  
);
```

```
CREATE TABLE PACKAGINGS(  
UPC int,  
Type varchar(15) default 'NORMAL',  
Height int not null,  
Width int not null,  
Colour varchar(20) default 'BLACK',  
Material varchar(30) default 'PLASTIC',  
Security int,  
constraint FK_PACKAGINGS_UPC foreign key(UPC) references PRODUCTS,  
constraint PK_PACKAGINGS primary key(UPC,Type)  
);
```

```
CREATE TABLE WORKOFFPERIODS(  
WOPID char(15),  
BeginingDate date not null,  
EndingDate date not null,  
Reason varchar(20) not null default 'NO REASON',  
TotalDays numeric(3) not null default 0,  
constraint PK_WORKOFFPERIODS_WOPID primary key(WOPID)  
);
```

```
CREATE TABLE EMPWOP(  
SSN char(13) not null,  
WOPID char(15),  
constraint PK_EMPWOP_WOPID primary key(WOPID),  
constraint FK_EMPWOP_WOPID foreign key(WOPID) references WORKOFFPERIODS,  
constraint FK_EMPWOP_SSN foreign key(SSN) references EMPLOYEES  
);
```

```
CREATE TABLE HOLIDAYS(  
HolidayID char(15),  
StartDate date not null,  
EndDate date not null,  
Season varchar(10) not null default 'NOSEASON',  
TotalDays numeric(2) not null default 0,
```

```
constraint PK_HOLIDAYS_HolidayID primary key(HolidayID),
constraint CHK_HOLIDAYS_Season CHECK(Season
IN('WINTER','AUTUMN','SPRING','SUMMER'))
);
```

```
CREATE TABLE EMPHOL(
SSN char(13) not null,
HolidayID char(15),
constraint PK_EMPHOL_HolidayID primary key(HolidayID),
constraint FK_EMPHOL_HolidayID foreign key(HolidayID) references HOLIDAYS,
constraint FK_EMPHOL_SSN foreign key(SSN) references EMPLOYEES
);
```

```
CREATE TABLE SALARIES(
SSN char(13),
Month_Name varchar(10),
TypeNumber decimal(1,0) not null, --la tienda tiene empleados siempre
BankAccount char(20) not null,
Type varchar(20) null,--sera regular o extrasalary en funcion de lo que se introduzca en el
typeNumber
Amount decimal(4,0) null,
constraint CHK_SALARIES_Type CHECK(type IN('REGULAR','EXTRASALARY')),
constraint PK_SALARIES_SSN_MONTH primary key(SSN,Month_Name),
constraint FK_SALARIES_SSN foreign key(SSN) references EMPLOYEES
);
```

```
CREATE TABLE ALLERGIES(
SSN char(13),
AllergyType varchar(10) default 'NOALLG',
constraint PK_ALLERGIES_SSN_AllergyType primary key(SSN,AllergyType),
constraint FK_ALLERGIES_SSN foreign key(SSN) references EMPLOYEES
);
```

```
CREATE TABLE EMPDISEASES(
SSN char(13),
```



```
DiseaseName varchar(20) default 'THE EMPLOYEE HAS NO DISEASES',
constraint PK_EMPDISEASES_SSN_DiseaseName primary key(SSN, DiseaseName),
constraint FK_EMPDISEASES_SSN foreign key(SSN) references EMPLOYEES
);
```

```
CREATE TABLE EMPEMAILS(
SSN char(13),
EmailAdress varchar(50) default 'NO REGISTERED EMAILS',
constraint PK_EMPEMAIL_SSN_EmailAdress primary key(SSN, EmailAdress),
constraint FK_EMPEMAIL_SSN foreign key(SSN) references EMPLOYEES
);
```

```
CREATE TABLE EMPHONE#(
SSN char(13),
PhoneNumber numeric(20) default '0',
constraint PK_EMPHONE#_SSN_PhoneNumber primary key(SSN, PhoneNumber),
constraint FK_EMPHONE#_SSN foreign key(SSN) references EMPLOYEES
);
```

```
CREATE TABLE SUPPORTMANS(
SSN char(13),
LanguageLevel varchar(9) not null default 'NOTSPEC',
constraint PK_SUPPORTMANS_SSN primary key(SSN),
constraint FK_SUPPORTMANS_SSN foreign key(SSN) references EMPLOYEES
);
```

```
CREATE TABLE SOFTWAREENGINEERS(
SSN char(13),
UniversityTittle varchar(50) not null default '',
NumProgrammingLanguages numeric(3) not null default 0,
constraint PK_SOFTWAREENGINEERS_SSN primary key(SSN),
constraint FK_SOFTWAREENGINEERS_SSN foreign key(SSN) references EMPLOYEES,
constraint CHK_SOFTWAREENGINEERS_NumProgrammingLanguages
CHECK(NumProgrammingLanguages > 0 or NumProgrammingLanguages = 0)
);
```

```
CREATE TABLE ACCOUNTMANAGERS(  
SSN char(13),  
UniversityTitle varchar(50) not null default '',  
constraint PK_ACCOUNTMANAGERS_SSN primary key(SSN),  
constraint FK_ACCOUNTMANAGERS_SSN foreign key(SSN) references EMPLOYEES  
);
```

```
CREATE TABLE SALESMANS(  
SSN char(13),  
constraint PK_SALESMANS_SSN primary key(SSN),  
constraint FK_SALESMANS_SSN foreign key(SSN) references EMPLOYEES  
);
```

```
CREATE TABLE VEHICLES(  
VehicleID char(9),  
LicensePlate varchar(8) not null,  
FuelConsumption numeric(3), --litres per 100km  
TypeOfVehicle varchar(10) not null, --the longest type is motorcycle  
FuelType varchar(6) not null default 'DIESEL', --diesel or petrol  
constraint PK_VEHICLES_VehicleID primary key (VehicleID),  
constraint U_VEHICLES_VehicleID unique (LicensePlate)  
);
```

```
CREATE TABLE DELIVERYMANS(  
SSN char(13),  
DrivingLicense varchar(5) not null,  
VehicleID char(9) not null,  
constraint PK_DELIVERYMANS_SSN primary key(SSN),  
constraint FK_DELIVERYMANS_SSN foreign key(SSN) references EMPLOYEES,  
constraint FK1_DELIVERYMANS_SSN foreign key(VehicleID) references VEHICLES  
);
```

```
CREATE TABLE CLIENTS( --international  
Client_ID int IDENTITY(0,1),
```

```
Birthday_Date DATE,  
DNI NVARCHAR(9) not null,  
Gender NVARCHAR(1) default 'O',  
AGE int,  
Type VARCHAR(11) not null,  
constraint PK_Clients_ID primary key(Client_ID),  
constraint Gender_Values CHECK (Gender in ('M', 'F', 'O'))  
);
```

```
CREATE TABLE SUGGESTIONS(  
Suggestion# int not null,  
Description NVARCHAR(144),  
Date DATE not null,  
Product NVARCHAR(144),  
Rate int default 5, --the number of the stars  
Client_ID int,  
constraint PK_SUGGESTIONS# primary key(Suggestion#),  
constraint FK_CLIENTS_CLIENT_ID foreign key(Client_ID) references CLIENTS,  
constraint Rate_Values CHECK (Rate > 0 and Rate < 6)  
);
```

```
CREATE TABLE DISABILITIES(  
Client_ID int ,  
Disabilities VARCHAR(144),  
constraint PK_DISABILITIES primary key(Client_ID, Disabilities),  
constraint FK_CLIENT_ID foreign key(Client_ID) references CLIENTS  
);
```

```
CREATE TABLE DISEASES(  
Client_ID int,  
Diseases VARCHAR(144),  
constraint PK_DISEASES primary key(Client_ID, Diseases),  
constraint FK_DISEASES_CLIENT_ID foreign key(Client_ID) references CLIENTS  
);
```

```
CREATE TABLE EVENTS(  
Event_ID int,  
Price float not null default 0.0,  
HourE VARCHAR(5),  
Location VARCHAR(144) not null,  
DateDay DATE,  
DateMonth DATE,  
DateYear DATE,  
constraint PK_EVENT_ID_CLIENTS primary key(Event_ID)  
);
```

```
CREATE TABLE EVECLI(  
Client_ID int not null,  
Event_ID int not null,  
constraint PK_EVENT_ID_CLIENTS_EVE primary key(Client_ID, Event_ID),  
constraint FK_EVENT_ID_CLIENT_ID foreign key(Client_ID) references CLIENTS,  
constraint FK_EVENT_ID foreign key(Event_ID) references EVENTS  
);
```

```
CREATE TABLE NONPARTNERS(  
Client_ID int not null,  
FirstTimeInAGameStore nvarchar(3) not null,  
constraint PK_NPART_CLIENT_ID primary key(Client_ID),  
constraint FIRSTTIME_VALUE CHECK (FirstTimeInAGameStore in ('YES', 'NO')),  
constraint FK_TYPE_CLIENTS_NONPAR foreign key(Client_ID) references CLIENTS  
);
```

```
CREATE TABLE PARTNERS(  
Client_ID int,  
Points# int not null default 0,  
FullNameName NVARCHAR(20) not null,  
FullNameSurname NVARCHAR(40) not null,  
constraint PK_PART_CLIENT_ID primary key(Client_ID),  
constraint POINTS#_VALUE CHECK (Points# > 0),  
constraint FK_TYPE_CLIENTS_PAR foreign key(Client_ID) references CLIENTS
```

);

```
CREATE TABLE OFFERS(  
OfferCode int,  
Type VARCHAR(20),  
Description VARCHAR(144),  
RequiredPoints int not null,  
Percentage VARCHAR(3) not null default '00%',  
Client_ID int,  
constraint PK_OFFERS primary key(OfferCode),  
constraint REQUIREDPOINTS_VALUE CHECK (RequiredPoints > 0),  
constraint FK_PARTNER_ID foreign key(Client_ID) references PARTNERS  
);
```

```
CREATE TABLE BUY(  
BuyDate DATE,  
Client_ID int not null,  
UPC int not null,  
SSN CHAR(13) not null,  
constraint PK_BUY primary key(BuyDate, Client_ID, UPC),  
constraint FK_CLIENT_ID_BUY foreign key(Client_ID) references CLIENTS,  
constraint FK_UPC_BUY foreign key(UPC) references PRODUCTS,  
constraint FK_SSN_BUY foreign key(SSN) references SALESMANS  
);
```

```
CREATE TABLE RENT(  
RentDate DATE,  
Client_ID int not null,  
UPC int not null,  
SSN CHAR(13) not null,  
constraint PK_RENT primary key(RentDate, Client_ID, UPC),  
constraint FK_CLIENT_ID_RENT foreign key(Client_ID) references CLIENTS,  
constraint FK_UPC_RENT foreign key(UPC) references PRODUCTS,  
constraint FK_SSN_RENT foreign key(SSN) references SALESMANS  
);
```

```
CREATE TABLE EMPPRO(  
SSN CHAR(13) not null,  
UPC int not null,  
constraint PK_EMPPRO primary key(SSN, UPC),  
constraint FK_SSN_EMPPRO foreign key(SSN) references EMPLOYEES,  
constraint FK_UPC_EMPPRO foreign key(UPC) references PRODUCTS  
);
```

```
CREATE TABLE WEBPAGES(  
URL VARCHAR(144),  
Domain VARCHAR(144) not null,  
Language VARCHAR(20) not null default 'SPANISH',  
Country VARCHAR(20) not null,  
Visits int not null,  
SecurityProtocol VARCHAR(144),  
constraint PK_URL primary key(URL)  
);
```

```
CREATE TABLE SOFTWEB(  
URL VARCHAR(144) not null,  
SSN CHAR(13) not null,  
constraint PK_SOFTWEB primary key(URL, SSN),  
constraint FK_URL_SOFTWEB foreign key(URL) references WEBPAGES,  
constraint FK_SSN_SOFTWEB foreign key(SSN) references SOFTWAREENGINEERS  
);
```

```
CREATE TABLE EXPENSES(  
ExpenseNumber int,  
Amount int not null,  
ExpensesDate DATE not null,  
Description VARCHAR(144) default 'NO DESCRIPTION',  
Paid varchar(3) not null,  
constraint PK_EXPENSES primary key(ExpenseNumber),  
constraint PAID_VALUE CHECK (Paid in ('YES','NO'))
```

);

```
CREATE TABLE ACCEXP(  
  SSN CHAR(13),  
  ExpenseNumber int,  
  constraint PK_ACCEXP primary key(SSN, ExpenseNumber),  
  constraint FK_SSN_ACCEXP foreign key(SSN) references ACCOUNTMANAGERS,  
  constraint FK_EXPENSENUMBER foreign key(ExpenseNumber) references EXPENSES  
);
```

```
ALTER TABLE SALARIES DROP CONSTRAINT FK_SALARIES_SSN
```

```
ALTER TABLE SALARIES ADD CONSTRAINT FK_SALARIES_SSN FOREIGN KEY  
(SSN) REFERENCES EMPLOYEES ON DELETE CASCADE  
GO
```

```
CREATE TRIGGER tr_deletions  
on employees  
for delete  
as  
delete SALARIES  
where SSN = (select SSN  
              from deleted  
              where ssn = deleted.SSN)
```

```
return
```

```
GO
```

```
CREATE TRIGGER tr_mensaje  
on employees  
for insert  
as  
print('A new employee has been added to the database')  
return
```

```
go
```

```

CREATE TRIGGER tr_age
  ON EMPLOYEES
  for insert
as
  update EMPLOYEES
  set EMPLOYEES.Age=DATEDIFF(year,EMPLOYEES.BirthdayDate, getdate())
  from inserted
  where EMPLOYEES.SSN = inserted.SSN

return
go
CREATE TRIGGER tr_Antiquity ---holidays and wop pero con dias
on EMPLOYEES
FOR INSERT
AS
  update EMPLOYEES
  set EMPLOYEES.Antiquity=DATEDIFF(year,EMPLOYEES.HiringDate, getdate())
  from inserted
  where EMPLOYEES.SSN = inserted.SSN

return
go
CREATE TRIGGER tr_Amount
on salaries
for insert
as
  IF((select SALARIES.TypeNumber
  from EMPLOYEES inner join SALARIES on EMPLOYEES.SSN = SALARIES.SSN
  where EMPLOYEES.SSN = (select SSN from inserted)) = '1')
    update SALARIES
      set SALARIES.Amount = 2000
      from inserted
      where SALARIES.SSN = inserted.SSN

ELSE

```



```

        update SALARIES
            set SALARIES.Amount = 2000*2
            from inserted
            where SALARIES.SSN = inserted.SSN

return
go
CREATE TRIGGER tr_Type
on salaries
for insert
as
IF ((select SALARIES.TypeNumber
      from SALARIES
      where SALARIES.SSN =(select inserted.SSN
                           from inserted)) = '1') -- si para el
empleado introducido el salario es regular se le pone el salario base y si no es paga extra y se
multiplica por 2
        update SALARIES
            set SALARIES.Type = 'REGULAR'
            from inserted
            where SALARIES.SSN = inserted.SSN

ELSE

        update SALARIES
            set SALARIES.Type = 'EXTRASALARY'
            from inserted
            where SALARIES.SSN = inserted.SSN

return

```

/\*\*SENTENCIAS PROBAR TRIGGER DELETE

insert into

EMPLOYEES(Name,Surname,BirthdayDate,DNI,Gender,SSN,Street,Zip,City,HiringDate,Type)

values('Perico','Delgado','10/20/1963','01852033J','M',3897046317222,'La puerta la villa','21003','Gijon','09/22/1999','SALESMANS')

insert into SALARIES(BankAccount,Month\_Name,SSN,TypeNumber)

values('524663','February',3897046317222,2)

select \*

from EMPLOYEES

select \*

from SALARIES

delete EMPLOYEES

where SSN = '3897046317222'

\*/

/\*\*SENTENCIAS PARA PROBAR EL RESTO DE TRIGGERS

INSERT INTO EMPLOYEES

(SSN,DNI,Name,Surname,City,Zip,Street,Age,Gender,BirthdayDate,HiringDate,Type)

VALUES(1321621,'01010066K','Luisa','Gonzalez','Leon',24009,'C/Ordoño II',19,'M','02/21/2000','10/01/2005','ACCOUNTMANAGERS')

insert into SALARIES(BankAccount,Month\_Name,SSN,TypeNumber)

```
values('524663','feb',1321621,1)
```

```
INSERT INTO EMPLOYEES
```

```
(SSN,DNI,Name,Surname,City,Zip,Street,Age,Gender,BirthdayDate,HiringDate,Type)
```

```
VALUES(1321622,'01010066H','Ramon','Garcia','Gijon',33009,'Av/Llano',19,'M','04/11/1990','11/05/2000','SOFTWAREENGINEERS')
```

```
insert into SALARIES(BankAccount,Month_Name,SSN,TypeNumber)
```

```
values('524663','feb',1321622,2)
```

```
SELECT *
```

```
FROM SALARIES
```

```
*/
```