By Sergio Diez, Mario Celada, Sara Gómez

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

10. SQL sentences in Oracle

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 number(9),
 Fax number(9),
 Company_Name varchar2(20),
 Type varchar2(8) not null,
constraint PK_SUPPLIERS primary key(SupplierID)
);
```

```
CREATE TABLE SUPEMAIL(
 SupplierID number(9),
 EmailAdress varchar2(30),
constraint FK_SUPEMAIL_SupplierID foreign key(SupplierID) references
SUPPLIERS,
constraint PK_SUPEMAIL primary key(SupplierID,EmailAdress)
);

CREATE TABLE SUPPHONENUM(
 SupplierID number(9),
 PhoneNumber number(9),
constraint FK_SUPPHONENUM_SupplierID foreign key(SupplierID) references
SUPPLIERS,
constraint PK_SUPPHONENUM primary key(SupplierID,PhoneNumber)
);

CREATE TABLE INTERNATIONAL(
 SupplierID number(9),
constraint FK_INTERNATIONAL_SupplierID foreign key(SupplierID) references
SUPPLIERS,
constraint PK_INTERNATIONAL primary key(SupplierID)
);

CREATE TABLE NATIONAL(
 SupplierID number(9),
constraint FK_NATIONAL_SupplierID foreign key(SupplierID) references
SUPPLIERS,
constraint PK_NATIONAL primary key(SupplierID)
);

CREATE TABLE ARRIVALPOINTS(
  APID number(9),
  Shopping_Centre varchar2(20),
  Country varchar2(15),
  Door# number(9),
  Type varchar2(15),
  Zip number(9),
  Street number(9),
  City varchar2(20),
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 varchar2(20) not null, --cadena de texto variable
Surname varchar2(20) not null,
City varchar2(50),
Zip number(20),
Street varchar2(50),
Age decimal(3,0) null,
Gender varchar2(1),
BirthdayDate date not null,
HiringDate date not null,
Antiquity number(2) null,
Type varchar2(17) not null, --el numero que definimos como maximo de la cadena seria el mas largo de 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','SECURITYMANS','SUPPORTMANS','SALESMANS'))
);


CREATE TABLE SECURITYMANS(
SSN char(13),
Specialisation varchar2(15) default 'NOSPECIALYSED' not null,
constraint PK_SECURITYMANS_SSN primary key(SSN),
constraint FK_SECURITYMANS_SSN foreign key(SSN) references EMPLOYEES
);
CREATE TABLE THIEVES(
ThieveID number(9),
--Picture BLOB(16M) formato para imagenes
Nationality VARCHAR2(30),
HairColour VARCHAR2(15),
Gender VARCHAR2(1) default 'O',
Dangerous number default 0,
SSN CHAR(13) not null,
CaptureDateHour DATE DEFAULT SYSDATE,
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 number(9),
Origing_Country varchar2(15),
Discount char(3),
Warranty NUMBER(3) DEFAULT 2,
GuideLanguage NUMBER(9),
Price_PhysicalStore NUMBER(9),
Price_Online NUMBER(9),
Day number(2),
Month number(2),
Year number(4),
Type varchar2(8) not null,
APID NUMBER(9),
SendingTime DATE DEFAULT SYSDATE,
ThieveID NUMBER(9),
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_PriceChPh CHECK(Price_PhysicalStore > 0),
constraint CHK_PRODUCTS_PriceChOn CHECK(Price_Online > 0),
constraint CHK_PRODUCTS_Year CHECK(Year > 2000)
);

CREATE TABLE PERIPHERALS(
 UPC number(9),
 Type varchar2(15),
constraint FK_PERIPHERALS_UPC foreign key(UPC) references PRODUCTS,
constraint PK_PERIPHERALS primary key(UPC)
);

CREATE TABLE INTPER(
 UPC number(9),
 Price varchar2(15),
 SupplierID number(9),
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 number(9),
 Name varchar2(15),
 Brand varchar(15),
constraint FK_CONSOLES_UPC foreign key(UPC) references PRODUCTS,
constraint PK_CONSOLES primary key(UPC)
);

CREATE TABLE INTCON(
 UPC number(9),
 Price varchar2(15),
 SupplierID number(9),
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 number(9),
 Tittle varchar2(15),
 Genre varchar(15),
constraint FK_FILMS_UPC foreign key(UPC) references PRODUCTS,
constraint PK_FILMS primary key(UPC)
);

CREATE TABLE NATFIL(
 UPC number(9),
 Price varchar2(15),
 SupplierID number(9),
constraint FK_NATFIL_UPC foreign key(UPC) references FILMS,
constraint FK_NATFIL_SupplierID foreign key(SupplierID) references NATIONAL,
constraint PK_NATFIL primary key(UPC,SupplierID)
);

CREATE TABLE GAMES(
 UPC number(9),
 Type varchar2(15),
 Platform varchar(15),
constraint FK_GAMES_UPC foreign key(UPC) references PRODUCTS,
constraint PK_GAMES primary key(UPC)
);

CREATE TABLE NATGAM(
 UPC number(9),
 Price varchar2(15),
 SupplierID number(9),
constraint FK_NATGAM_UPC foreign key(UPC) references GAMES,
constraint FK_NATGAM_SupplierID foreign key(SupplierID) references
NATIONAL,
constraint PK_NATGAM primary key(UPC,SupplierID)
);
```

```
CREATE TABLE GUIDELANGUAGE(
  UPC number(9),
  ProductManualLanguage varchar2(20),
constraint FK_GUIDELANGUAGE_UPC foreign key(UPC) references PRODUCTS,
constraint PK_GUIDELANGUAGE primary key(UPC,ProductManualLanguage)
);
```

```
CREATE TABLE PACKAGINGS(
  UPC number(9),
  Type varchar2(15) DEFAULT 'NORMAL',
  Height number(10),
  Width number(10),
  Colour varchar2(20) DEFAULT 'BLACK',
  Material varchar(30) DEFAULT 'PLASTIC',
  Security number(9),
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 varchar2(20) default 'NOT SPECIFYED' not null,
TotalDays number(3)  default 0 not null,
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 varchar2(10) default 'NOSEASON' not null,
TotalDays number(2) default 0 not null,
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 SUPPORTMANS(
SSN char(9),
LanguageLevel varchar2(9),
constraint PK_SUPPORTMANS_SSN primary key(SSN),
constraint FK_SUPPORTMANS_SSN foreign key(SSN) references EMPLOYEES
);
```

```
CREATE TABLE SALARIES(
SSN char(13),
Month_Name varchar2(10),
TypeNumber number(1) not null,
BankAccount char(20) not null,
Type varchar2(20) null,
Amount number(4) null,
constraint CK_SALARIES_Type CHECK(type
IN('REGULAR','EXTRASALARY')),
constraint PK_SALARIES_SSNMONTH  primary key(SSN,Month_Name),
constraint FK_SALARIES_SSN foreign key(SSN) references EMPLOYEES
);

CREATE TABLE ALLERGIES(
SSN char(13),
AllergyType varchar2(10) default 'NOALLERG',
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 varchar2(20) default '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 varchar2(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 NUMBER(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 varchar2(9) default 'NOTSPEC.' not null,
constraint PK_SUPPORTMANS_SSN primary key(SSN),
constraint FK_SUPPORTMANS_SSN foreign key(SSN) references EMPLOYEES
);

CREATE TABLE SOFTWAREEG(
SSN char(13),
UnivTittle varchar2(50) default '' not null,
NumPrgLang number(3) default 0 not null,
constraint PK_SOFTWAREENG_SSN primary key(SSN),
constraint FK_SOFTWAREENG_SSN foreign key(SSN) references EMPLOYEES,
constraint CHK_SOFTWAREEG_NumPrgLang CHECK(NumPrgLang > 0 or
NumPrgLang= 0)
);

CREATE TABLE ACCOUNTMANAGERS(
SSN char(13),
UniversityTittle varchar2(50) default '' not null,
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 varchar2(8) not null,
FuelConsumption number(3), --litres per 100km
TypeOfVehicle varchar2(10) not null, --the longest type is motorcycle
FuelType varchar2(6) default 'DIESEL' not null, --diesel or petrol
constraint PK_VEHICLES_VehicleID primary key (VehicleID),
constraint U_VEHICLES_VehicleID unique (LicensePlate)
);
CREATE TABLE CLIENTS(
  Client_ID number(9),
  Bithday_Date DATE DEFAULT SYSDATE,
  DNI NVARCHAR2(9),
  Gender NVARCHAR2(1),
  AGE NUMBER(3),
  Type VARCHAR2(8) not null,
constraint PK_Clients_ID primary key(Client_ID),
constraint Gender_Values CHECK (Gender in ('M', 'F', 'O'))
);

CREATE TABLE SUGGESTIONS(
  Suggestion# NUMBER(9),
  Description NVARCHAR2(144),
  Daate date,
  Product NVARCHAR2(144),
  Rate NUMBER(1),
  Client_ID NUMBER(9),
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 NUMBER(9),
  Disabilities VARCHAR2(144),
constraint PK_DISABILITIES primary key(Client_ID, Disabilities),
constraint FK_CLIENT_ID foreign key(Client_ID) references CLIENTS
);
```

```
CREATE TABLE DISEASES(
  Client_ID NUMBER(9),
  Diseases VARCHAR2(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 NUMBER(9),
  Price NUMBER(9),
  Hour VARCHAR2(5),
  Location VARCHAR2(144),
  DateDay DATE DEFAULT SYSDATE,
  DateMonth DATE DEFAULT SYSDATE,
  DateYear DATE DEFAULT SYSDATE,
constraint PK_EVENT_ID_CLIENTS primary key(Event_ID)
);

CREATE TABLE EVECLI(
  Client_ID NUMBER(9),
  Event_ID NUMBER(9),
constraint PK_EVENT_ID_CLI 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 NUMBER(9),
  FirstTimeInAGameStore NVARCHAR2(3),
constraint PK_NPART_CLIENT_ID primary key(Client_ID),
constraint FIRSTTIME_VALUE CHECK (FirstTimeInAGameStore in (0, 1))
```

```
);



CREATE TABLE PARTNERS(
  Client_ID NUMBER(9),
  Points# NUMBER(9),
  FullNameName NVARCHAR2(20),
  FullNameSurname NVARCHAR2(40),
constraint PK_PART_CLIENT_ID primary key(Client_ID),
constraint POINTS#_VALUE CHECK (Points# > 0)
);



CREATE TABLE OFFERS(
  OfferCode NUMBER(9),
  Type VARCHAR2(20),
  Description VARCHAR2(144),
  RequiredPoints NUMBER(9),
  Percentage VARCHAR2(4),
  Client_ID NUMBER(9),
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 DEFAULT SYSDATE,
  Client_ID NUMBER(9),
  UPC NUMBER(9),
  SSN CHAR(9),
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 DEFAULT SYSDATE,
  Client_ID NUMBER(9),
  UPC NUMBER(9),
  SSN CHAR(9),
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(9),
  UPC NUMBER(9),
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 VARCHAR2(144),
  Domain VARCHAR2(144),
  Language VARCHAR2(20),
  Country VARCHAR2(20),
  Visits NUMBER(9),
  SecurityProtocol VARCHAR2(144),
constraint PK_URL primary key(URL)
);

CREATE TABLE SOFTWEB(
  URL VARCHAR2(144),
  SSN CHAR(9),
constraint PK_SOFTWEB primary key(URL, SSN),
constraint FK_URL_SOFTWEB foreign key(URL) references WEBPAGES,
constraint FK_SSN_SOFTWEB foreign key(SSN) references SOFTWAREEG
);


CREATE TABLE EXPENSES(
ExpenseNumber number(9),
Amount number(9) not null,
ExpensesDate DATE not null,
Description VARCHAR2(144) default 'NO DESCRIPTION',
Paid varchar2(3) not null,
constraint PK_EXPENSES primary key(ExpenseNumber),
constraint PAID_VALUE CHECK (Paid in ('YES','NO'))
);

CREATE TABLE ACCEXP(
  SSN CHAR(9),
  ExpenseNumber NUMBER(9),
constraint PK_ACCEXP primary key(SSN, ExpenseNumber),
constraint FK_SSN_ACCEXP foreign key(SSN) references ACCOUNTMANAGERS,
constraint FK_EXPENSENUMBER foreign key(ExpenseNumber) references
EXPENSES
);



/*TRIGGERS ORACLE


CREATE TRIGGER tr_Antiquity ---holidays and wop pero con dias
 AFTER INSERT on EMPLOYEES
```

```
BEGIN
   update EMPLOYEES
   set EMPLOYEES.Antiquity=DATEDIFF(year,EMPLOYEES.HiringDate, getdate())
   from inserted
   where  EMPLOYEES.SSN = inserted.SSN

END;
```

We tried to do it. But we were not able since in oracle we can declare statement level or row level triggers. Sql Server only has statement level. In Oracle, you can declare before triggers or after triggers. Sql Server only has after triggers.
The main reason is that in oracle inserted or deleted tables do not exist per se, the equivalent is to use a row level trigger and write :OLD and :NEW instead of deleted and inserted.

*/