1. **Create the following tables**

```
CREATE TABLE Contracts(Ref  VARCHAR(10) PRIMARY KEY,
                       Organization      VARCHAR(100),
                       ContDate        DATE,
                       NumRoutes NUMBER(2,0));


CREATE TABLE Routes(Ref VARCHAR(10) REFERENCES Contracts ON
DELETE CASCADE,
                Origin         VARCHAR(50),
                Destination    VARCHAR(50),
                Vehicle        VARCHAR(20),
                PRIMARY KEY (Ref, Origin, Destination));
```

a) Write a stored procedure with a reference contract as input parameter. The procedure must update the information in NumRoutes, according to the number of routes associated with the contract. The procedure also must print the name of the organization and the total number of associated routes. You must declare an exception that is thrown to show a message if the reference does not have associated any route.

b) Create a trigger to keep updated the value of NumRoutes whenever a row is inserted or deleted.

2. **Create the following tables.**

```
CREATE TABLE Departments (CodDept   CHAR(5) PRIMARY KEY,
                          Name      VARCHAR(100));

CREATE TABLE Employees (SSN       CHAR(9) PRIMARY KEY,
                        Name      VARCHAR(100),
                       CodDept    CHAR(5) REFERENCES Departments
                       on    delete set NULL,
                        Salary     NUMBER(4,0));



CREATE TABLE Changes(IdChange      VARCHAR(10) PRIMARY KEY,
                     UserId        VARCHAR(8),
                     OldSalary     NUMBER(4,0),
                    NewSalary      NUMBER(4,0));
```

a) Write a trigger that records in the table Changes any update of the salary of the employees. The trigger must store the user, the date of the change, and both the salary before the change and the updated salary. The ID will be obtained from a sequence called `SEQChanges`.

b) Write a stored procedure that lists for each department the name and salary of each employee whose salary is lower than the average of the department. For each department the procedure must show the total amount of these salaries by department.

3. **Create the following tables.**

```
Create table Author (
SSN CHAR(9) PRIMARY KEY,
Name VARCHAR(50) NOT NULL,
LastName VARCHAR(50) NOT NULL);
Country VARCHAR(30) NOT NULL);
NumArticles CHAR(3) NOT NULL);

Create table Journal (
ISSN VARCHAR(9) PRIMARY KEY,
Name VARCHAR(100) NOT NULL);

Create table Article (
DOI CHAR(30) PRIMARY KEY,
Title VARCHAR(100) NOT NULL,
ISSNJournal VARCHAR(9) NOT NULL REFERENCES Journal(ISSN) ON DELETE
CASCADE,
NumAuthors NUMBER(1,0) NOT NULL,
CHECK NumAuthors BETWEEN 1 AND 4);

Create table Sign (
SSN CHAR(9) NOT NULL REFERENCES Author,
DOI CHAR(30) NOT NULL REFERENCES Article ON DELETE CASCADE,
PRIMARY KEY(SSN, DOI));
```

a. Write a stored procedure with a journal as input parameter. The procedure must show the data associated to the journal (ISSN and Name) and the names and surnames of the authors that have signed at least one article published in the journal. In the case that the journal has not associated any article, the procedure will show the message. “No Authors”.
b. Write a trigger that updates the column NumArticles in the table Author when a new article is inserted or deleted.

4. **Create the following tables.**

```
Create table Airport(
Code CHAR(6) PRIMARY KEY,
Name VARCHAR(30) NOT NULL,
Country VARCHAR(30)NOT NULL);

Create table Flight(
FNumber CHAR(6),
FDate DATE,
Departure CHAR(6) NOT NULL REFERENCES Airport on delete set NULL,
Arrival CHAR(6) NOT NULL REFERENCES Airport on delete set NULL,
Price NUMBER(6,2),
Seats NUMBER(3) DEFAULT 100,
primary key (FNumber,FDate),
unique (FDate, Departure, Arrival),
check(departure<>arrival));


Create table Tickets(
eticket CHAR(6),
fdate DATE NOT NULL,
Passport CHAR(10) NOT NULL,
PRIMARY KEY(eticket, fdate, passport),
FOREIGN KEY(eticket, fdate) REFERENCES Flight);
```

a) Write a stored procedure with a date, departure airport, arrival airport and a passport number as input parameters. The procedure must register a ticket for the passenger in the first flight between the provided airports in which there are available seats. In the case that we cannot find a flight that fulfils the requirements, the procedure will show a message.

b) Write a trigger that updates in the table Sales the total amount and the number of tickets of the flights, for each ticket that is sold or returned. In case of return, the airline only refunds 150€.

c) Execute following statements.

```
drop table ComissionAC;
drop table deposit;
drop table log;
create table CommissionAC(acNumber char(20), amount number(10,2));
create table deposit (acNumber char(20));
create table log( msg varchar(50));
```

Write a trigger that registers in the table log a message that includes the account number, the associated amount and the text “Associated Deposit” when a row from CommissionAC is deleted and the account is also stored in the table deposit. If the row is not stored in this table the message will include the text “Preferred Client”

Test the trigger executing these statements:

```
 insert into CommissionAC values ('12345678900987654321',13.9);
insert into CommissionAC values('12345123131333344321',13.0);
insert into CommissionAC values ('37423462487654321478',13.9);
insert into deposit values ('37423462487654321478');
delete from CommissionAC;
```

d) Execute following statements.

```
drop table Records;
drop table Marks;
create table Records(event number  primary key,  time number);
create table Marks(event number, dateEvent date, tiempo number,
primary key (event, dateEvent));
```

Write a trigger associated to the insertion of a new row in the table Marks. If the new time is less than the record associated with the event , then this will update in the table Records.

Test the trigger executing these statements:

```
      delete from Marks;
      delete from Records;
```

```
insert into Marks values (1, to_date(‘01/02/2015’),3.8);
insert into Marks values (1, to_date(‘02/02/2015’),4.2);
insert into Marks values (1, to_date(‘03/02/2015’),3.5);
```

e) **Execute following statements.**

```
drop table Books cascade constraints;
drop table Copies cascade constraints;
create table Books(isbn char(13) primary key,
                   copiesNumber integer);
create table Copies(catalogNumber char(5) primary key,
                    isbn char(13) not null,
                    FOREIGN KEY (isbn) REFERENCES Books);
```

Write a trigger associated to the insertion of a new row in the table Copies, such that, if the new isbn is not stored in the table Books, then a new row will be inserted in this table with the new isbn and copiesNumber equal to 1. If the isbn is stored in the table Books, the trigger will update the value of the column copiesNumber accordingly.

Test the trigger executing these statements.

```
insert into Copies values (‘A1’,‘1234567891011’);
insert into Copies values (‘A2’,‘1234567891011’);
```