1. Consider the following relational schema:
   CUSTOMER (SSN, Name, Surname, PhoneNum, Plan)
   PRICINGPLAN (Code, ConnectionFee, PricePerSecond)
   PHONECALL (SSN, Date, Time, CalledNum, Seconds)
   BILL (SSN, Month, Year, amount)
   Build the following triggers

   T1. Write a trigger that after each phone call updates the customer's bill.
   T2. We make the assumption that the bills to be updated are always already present in the database. In order to do this, we can create another trigger that creates a bill with an amount of 0 for each registered customer at the beginning of each month (suppose we have the event END_MONTH)
   T3. Write a trigger that at the end of each month discount the bills by 5 cents per call to direct users of the company (that is, to numbers of registered users in the table CUSTOMER) if the total monthly amount of the bill exceeds 100 €.

2. Consider the following relational schema:
   TITLE (TitleCode, Name, Type)
   TRANSACTION (TransCode, SellerCode, BuyerCode, TitleCode, Quantity, Value, Date, Instant) OPERATOR (Code, Name, Address, Availability)

   Build a trigger system that keeps the value of Availability of Operator updated after insertion of tuples in Transaction, taking into account that for each transaction in which the operator sells, the amount of the transaction must be added to its availability and subtracted for purchases. Also, enter the operators whose availability falls below zero in a table that lists the operators "uncovered". Assuming that there is: UNCOVERED (Code, Name, Address)

3. Consider the following relational schema: MATCH (Day, HomeTeam, AwayTeam, HomeGoal, AwayGoal) STANDING (Day, Team, Score) Assuming that the first table is fed through entries and that the second is properly derived from the first, write the active rules that construct the ranking, giving 3 points to the teams that win, 1 point to those that tie and 0 points to those that lose.

4. Consider the following relational schema for the European volleyball tournament:
   PLAYER (PlayerId, Name, Team, Height, Birthday, PlayedMatches)
   TEAM (Team, Coach, WonGames)
   MATCH (MatchId, Date, Team1, Team2, WonSetsTeam1, WonSetsTeam2, Referee)
   PLAYED (MatchId, PlayerId, Role, ScoredPoints)

T1. Build a trigger that keeps the value of WonGames after insertions in GAME taking into account that WonGames is relative to the entire history of the team, not only to the current tournament and that a team wins a game when he wins 3 sets.
T2. Building also a trigger that keeps PlayedMatches of PLAYER updated after insertions in PLAYED.