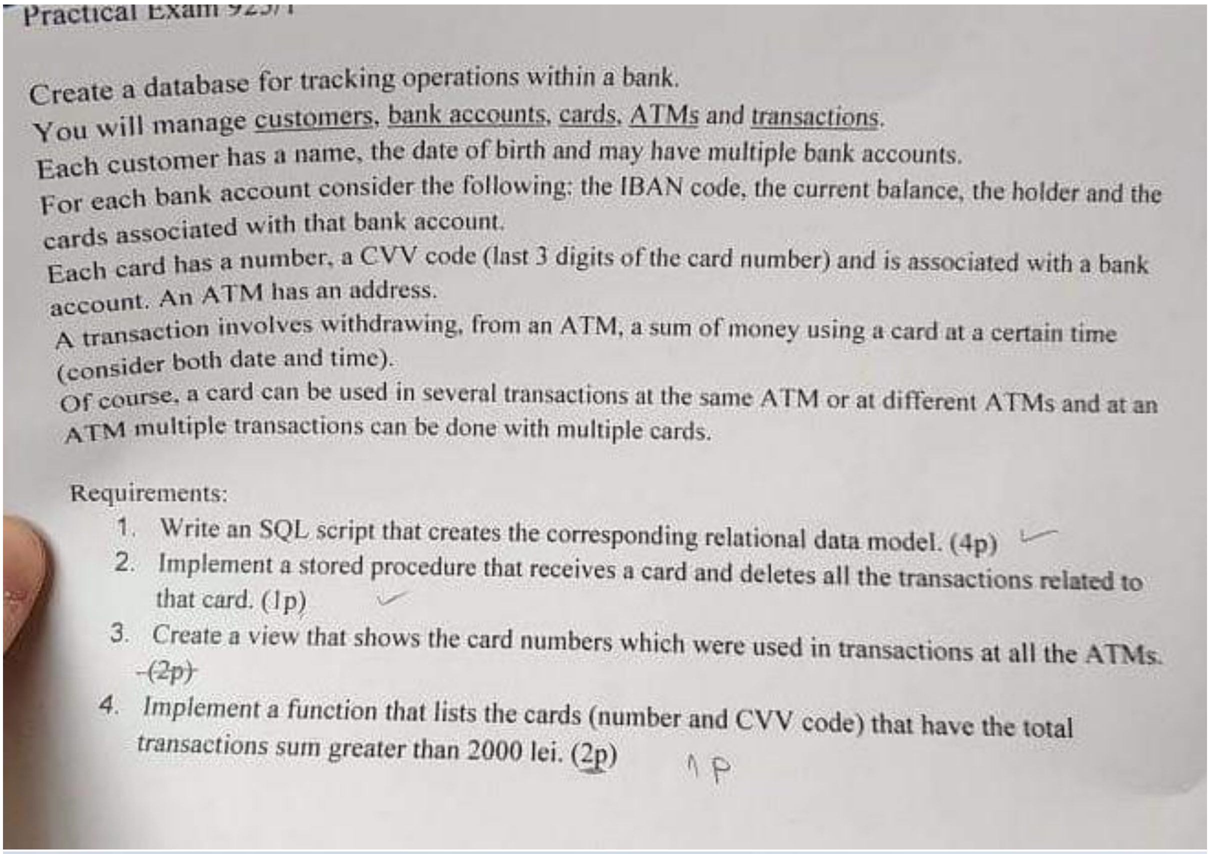


Practical Exam 923/1

Create a database for tracking operations within a bank.

You will manage customers, bank accounts, cards, ATMs and transactions.

Each customer has a name, the date of birth and may have multiple bank accounts.

For each bank account consider the following: the IBAN code, the current balance, the holder and the cards associated with that bank account.

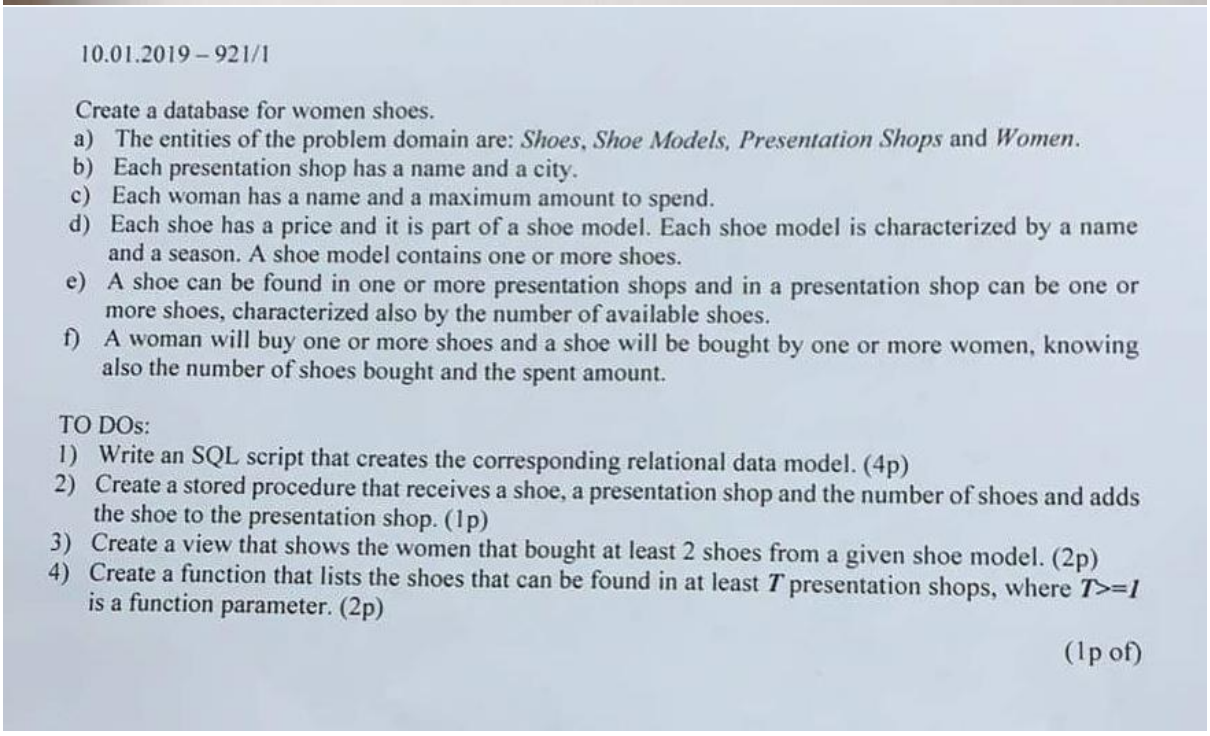
Each card has a number, a CVV code (last 3 digits of the card number) and is associated with a bank account. An ATM has an address.

A transaction involves withdrawing, from an ATM, a sum of money using a card at a certain time (consider both date and time).

Of course, a card can be used in several transactions at the same ATM or at different ATMs and at an ATM multiple transactions can be done with multiple cards.

Requirements:

1. Write an SQL script that creates the corresponding relational data model. (4p) ✓
2. Implement a stored procedure that receives a card and deletes all the transactions related to that card. (1p) ✓
3. Create a view that shows the card numbers which were used in transactions at all the ATMs. ~~(2p)~~
4. Implement a function that lists the cards (number and CVV code) that have the total transactions sum greater than 2000 lei. (2p) 1 P



10.01.2019 – 921/1

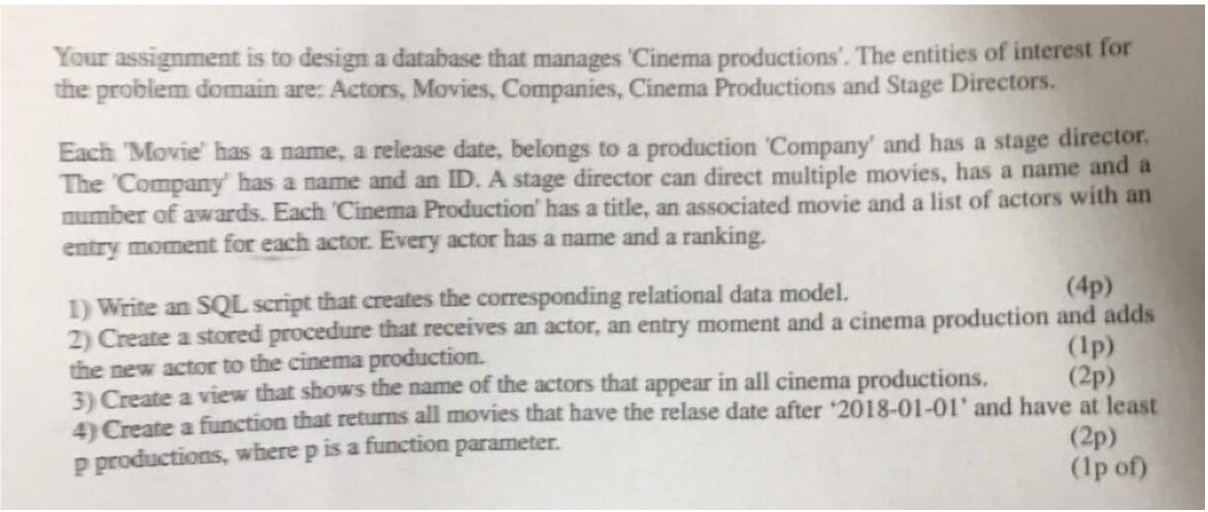
Create a database for women shoes.

a) The entities of the problem domain are: *Shoes*, *Shoe Models*, *Presentation Shops* and *Women*.
b) Each presentation shop has a name and a city.
c) Each woman has a name and a maximum amount to spend.
d) Each shoe has a price and it is part of a shoe model. Each shoe model is characterized by a name and a season. A shoe model contains one or more shoes.
e) A shoe can be found in one or more presentation shops and in a presentation shop can be one or more shoes, characterized also by the number of available shoes.
f) A woman will buy one or more shoes and a shoe will be bought by one or more women, knowing also the number of shoes bought and the spent amount.

TO DOs:

1) Write an SQL script that creates the corresponding relational data model. (4p)
2) Create a stored procedure that receives a shoe, a presentation shop and the number of shoes and adds the shoe to the presentation shop. (1p)
3) Create a view that shows the women that bought at least 2 shoes from a given shoe model. (2p)
4) Create a function that lists the shoes that can be found in at least ***T*** presentation shops, where ***T>=1*** is a function parameter. (2p)

(1p of)

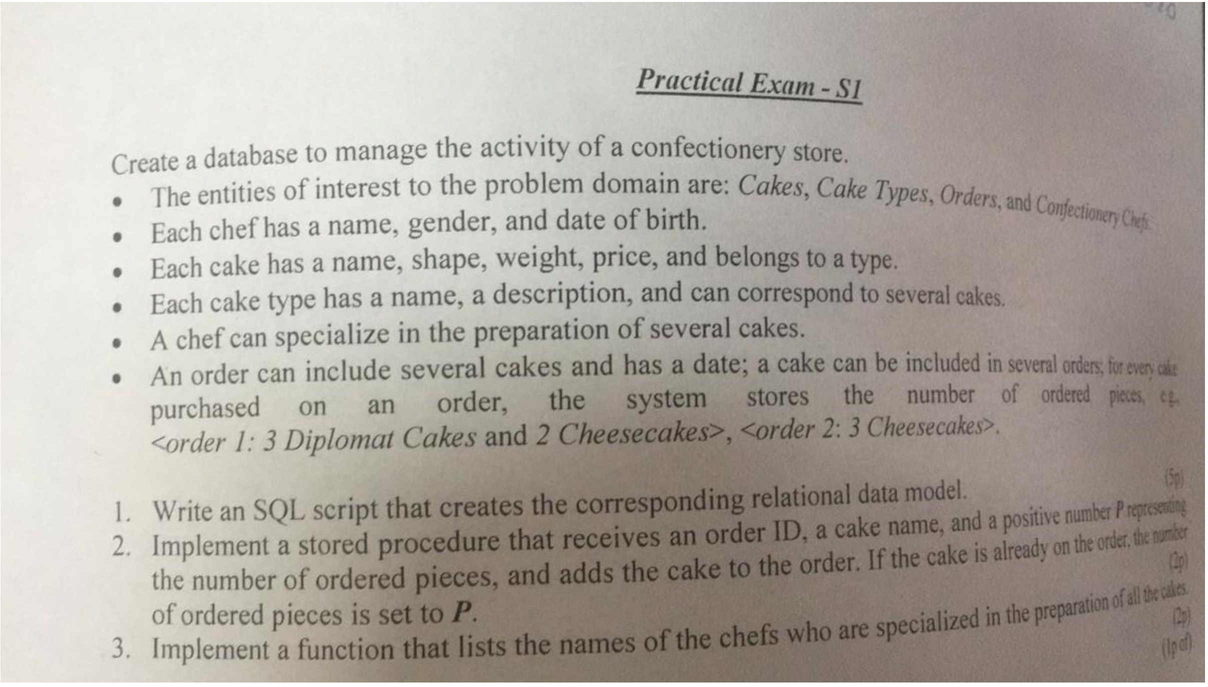


Your assignment is to design a database that manages 'Cinema productions'. The entities of interest for the problem domain are: Actors, Movies, Companies, Cinema Productions and Stage Directors.

Each 'Movie' has a name, a release date, belongs to a production 'Company' and has a stage director. The 'Company' has a name and an ID. A stage director can direct multiple movies, has a name and a number of awards. Each 'Cinema Production' has a title, an associated movie and a list of actors with an entry moment for each actor. Every actor has a name and a ranking.

1) Write an SQL script that creates the corresponding relational data model. (4p)
2) Create a stored procedure that receives an actor, an entry moment and a cinema production and adds the new actor to the cinema production. (1p)
3) Create a view that shows the name of the actors that appear in all cinema productions. (2p)
4) Create a function that returns all movies that have the relase date after '2018-01-01' and have at least p productions, where p is a function parameter. (2p)

(1p of)



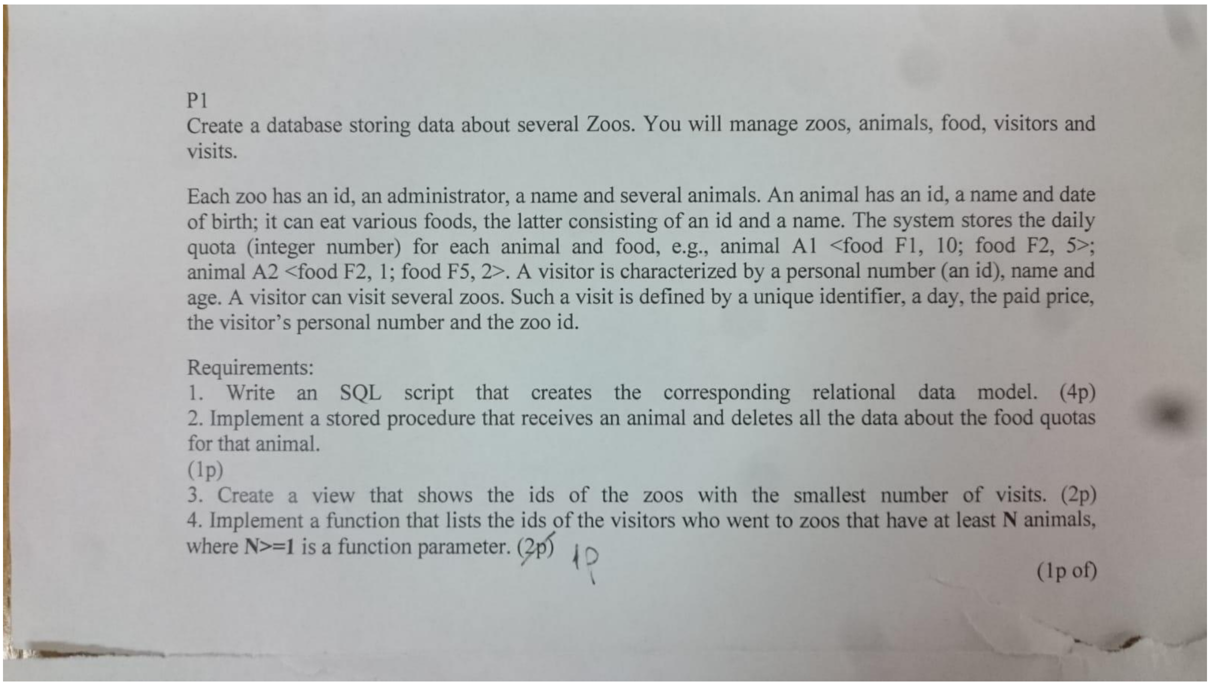
Practical Exam - S1

Create a database to manage the activity of a confectionery store.

- The entities of interest to the problem domain are: *Cakes*, *Cake Types*, *Orders*, and *Confectionery Chefs*.
- Each chef has a name, gender, and date of birth.
- Each cake has a name, shape, weight, price, and belongs to a type.
- Each cake type has a name, a description, and can correspond to several cakes.
- A chef can specialize in the preparation of several cakes.
- An order can include several cakes and has a date; a cake can be included in several orders; for every cake purchased on an order, the system stores the number of ordered pieces, e.g., *<order 1: 3 Diplomat Cakes* and *2 Cheesecakes>*, *<order 2: 3 Cheesecakes>*.

1. Write an SQL script that creates the corresponding relational data model. (5p)
2. Implement a stored procedure that receives an order ID, a cake name, and a positive number ***P*** representing the number of ordered pieces, and adds the cake to the order. If the cake is already on the order, the number of ordered pieces is set to ***P***. (2p)
3. Implement a function that lists the names of the chefs who are specialized in the preparation of all the cakes. (2p)

(1p of)



P1

Create a database storing data about several Zoos. You will manage zoos, animals, food, visitors and visits.

Each zoo has an id, an administrator, a name and several animals. An animal has an id, a name and date of birth; it can eat various foods, the latter consisting of an id and a name. The system stores the daily quota (integer number) for each animal and food, e.g., animal A1 <food F1, 10; food F2, 5>; animal A2 <food F2, 1; food F5, 2>. A visitor is characterized by a personal number (an id), name and age. A visitor can visit several zoos. Such a visit is defined by a unique identifier, a day, the paid price, the visitor's personal number and the zoo id.

Requirements:

1. Write an SQL script that creates the corresponding relational data model. (4p)
2. Implement a stored procedure that receives an animal and deletes all the data about the food quotas for that animal.
(1p)
3. Create a view that shows the ids of the zoos with the smallest number of visits. (2p)
4. Implement a function that lists the ids of the visitors who went to zoos that have at least **N** animals, where **N>=1** is a function parameter. (~~2~~p) 1p

(1p of)