This is a closed book exam: the only resources allowed are blank paper, pens, and your head. Explain your reasoning. Write clearly, in the sense of logic, language, and legibility. The clarity of your explanations affects your grade. Good luck!

Write your name and student number on *all* solution sheets and here. Name: .....
At the end of the exam, hand in *all* sheets that you received, including this one. Student number: .....

Problem 1 [30%] Design the Entity-Relationship schema of an information system relating to a car rental agency. Of each *car* owned by the agency, we are interested in the license plate (identifier), the brand, the model, the year of registration, and the rental category. Of each *rental category*, identified by a code, we are interested in the daily rental price and the rental price per km. Of each car rental *customer*, we are interested in the tax code (identifier), the name, and, if known, the date of birth. There are exactly two types of customers: occasional customers and registered customers. Each *occasional customer* has a credit card whose number and circuit are of interest. Of *registered customers* we are interested in the registration-card number (which identifies a registered customer) and the residence. Each registered customers benefits from a personalized discount on one or more rental categories, and the amount of this discount is of interest. In addition, of each registered customer we are interested in the (possibly empty) set of cars on which he has expressed a preference for possible rentals. Each *rental* is made by one and only one customer and is related to one and only one car. Of each rental we are also interested in the rental type (kilometer, days, etc..), the start date, and the expected duration in hours. Of each *completed rental*, we are also interested in the actual duration in hours and covered kilometers. Please note that the same car cannot be rented several times with the same start date. Finally, each car may be serviced, and of each *service work* we are also interested in the month in which it was carried out and the cost. The rule is that each car is subject to at most one service work per month.

Problem 2 [42%] Carry out the logical design of the database, producing the complete relational schema with constraints, taking into account the following indications: *(i)* The access to registered customers is mainly through the registration-card number. *(ii)* We want to avoid null values in the database. *(iii)* when accessing a customer, we want to know if it is an occasional or a registered customer.

As steps in your design you should produce:

1. [7%] the restructured ER schema (possibly with external constraints),
2. [25%] the direct translation into the relational model (possibly with external constraints), and
3. [10%] the restructured relational schema (again with constraints).

Motivate explicitly how the above indications affect your design.

Problem 3 [15%] Consider the relation `Purchase(code,customer,product,expense)`, which collects information about purchases made by customers and of which we know that it does not contain any null value. Each tuple stores the code of the purchase, the customer who made it, the product purchased and the amount of money spent by the customer. Write a SQL query that for each customer and each product calculates the average amount spent by the customer on purchases of that product that have recorded an expense greater than 1000 Euros, considering that this number must be 0 if the customer has not made purchases of that product that have recorded an expense greater than 1000 Euros.

Problem 4 [13%] Consider the database D consisting only of the relation R shown in Figure 1, and say which is the result obtained by executing the SQL queries Q_1 and Q_2 on the database D:

A	B	C	D
20	5	5	10
40	5	5	30
70	2	5	NULL
50	2	6	NULL

Figure 1: Relation R

Q_1:
```
SELECT T1.A, T1.B, T2.C, T2.D
FROM (SELECT A, B FROM R) T1,  (SELECT C, D FROM R) T2
WHERE T1.B = T2.C  AND  T2.D < T1.A
```

Q_2:
```
SELECT T1.A, T1.B, T2.C, T2.D
FROM (SELECT A, B FROM R) T1,  (SELECT C, D FROM R) T2
WHERE T1.B = T2.C  AND  T2.D < T1.A  AND
      T1.A NOT IN (SELECT D FROM R)
```

Hint: Recall that "`NOT IN`" is equivalent to "`<> ALL`".