**Instructions:**

**In a word document, answer the following questions.**

**Keep question in the exact order they are listed in.**

**Questions should be in bold font.**

**Answers should be in non-bold font.**

**All questions are worth the same points**

**Question 1 (chapter 3) Consider the following set of requirements for a UNIVERSITY database that is used to keep track of students' transcripts.**

**(a) The university keeps track of each student's name, student number, social security number, current address and phone, permanent address and phone, birthdate, sex, class (freshman, sophomore, ..., graduate), major department, minor department (if any), and degree program (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city, state, and zip of the student's permanent address, and to the student's last name. Both social security number and student number have unique values for each student.**

**(b) Each department is described by a name, department code, office number, office phone, and college. Both name and code have unique values for each department.**

**(c) Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of course number is unique for each course.**

**(d) Each section has an instructor, semester, year, course, and section number. The section number distinguishes different sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester.**

**(e) A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3,**

**4 for F, D, C, B, A, respectively).**

**Design an ER schema for this application, and draw an ER diagram for that schema.**

**Specify key attributes of each entity type and structural constraints on each relationship type. Note any unspecified requirements, and make appropriate assumptions to make the specification complete.**

\* indiciates unique

Obj indicates object that the attribute is pointing to

Student

name|number\*|SSN\*|CurrAddr|(Obj)PermAddr|(DateTime)Birthday|(Char)Sex|(Str)Class|(Obj)MajorDept|(Obj)MinorDept|(Str)DegreeProgr

PermAddr&Phone

(Str)City|(Str)State|(int)Zip|(int)Phone

Name

(Str)FirstName|(Str)LastName|(Char)MiddleName

Department

(Str)Name\*|(Str/long)DeptCode\*|(int)OfficeNum|(int)OfficePhone|(Str)College

Course

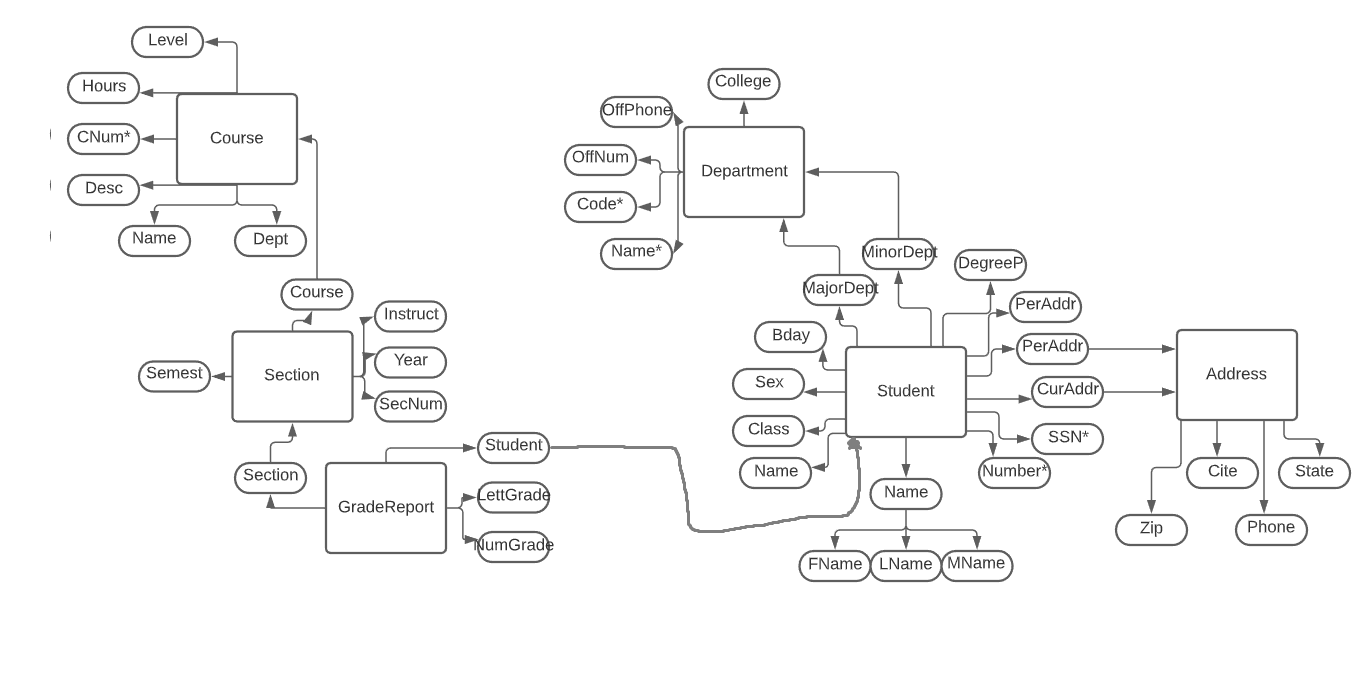
(Str)Name|(Str)Desc|(int)CourseNum\*|(int)CredHours|(int)Level|(Obj)OfferingDept

Section

(Str)Instruct|(Str)Semester|(int)Year|(Obj)Course|(int)SecNum

GradeReport

(Obj)Student|(Obj)Section|(Char)LettGrade|(int)NumGrade



**Question 2 (chapter 3) Consider the ER diagram of the following figure, which shows a simplified schema for an airline reservations system. Extract from the ER diagram the requirements and constraints that resulted in this schema. Try to be as precise as possible in your requirements and constraints specification.**

**An e r diagram of an Airport database schema. The entity, Airplane is related to Airplane type by type. Airplane has a key attribute, Airplane i d and Total no of seats. Double lines connect the entity Airplane with Type relationship box. Cardinality ratio from Airplane to Airplane Type is N:1. Airplane Type has the attributes, Type name, Maximum seats, and Company. Type name is the key attribute. Airplane type is related to airport by can land attribute. Airport has the attributes, Airport code, City, State and Name. Cardinality ratio from Airplane Type and Airport is N: M. Airport is linked to Flight leg through the attributes, Departure Airport and Arrival Airport. Departure airport has the attribute scheduled d e p time. Arrival airport has the attribute, scheduled arrival time. The relation, Departure Airport between Airport and Flight leg has a cardinality ratio 1: N. The relation, Arrival Airport between Airport and Flight leg has the cardinality ratio, 1:N. Flight leg is placed within double rectangles and has a key attribute, leg number. Double lines connect to the Flight leg from Departure and Arrival attributes. Flight Leg is related to Flight by an identifying relationship, Legs. The cardinality ratio from Flight leg to Flight is N: 1. Flight is related to Fare by the identifying relationship fares. Flight has the attributes, number, airline, and weekdays. Number is the key attribute. Cardinality ratio from Flight to Fare is 1: N. Fare has the attributes, Code, Amount, and Restrictions. Code is the key attribute. An instance of flight leg is assigned to Airplane from the Leg instance entity. Leg instance is a weak entity. It has the attributes date and Number of available seats. Date being the key attribute. The cardinality ratio from Flight Leg to Leg instance is 1:N. Cardinality ratio from Airplane to Leg instance is 1:N. Leg instance is related to airport by Departure time and Arrival time. Cardinality ratio from Leg instance to Airport through the Arrival time relation is N:1 and ratio from Leg instance to airport through departure time relation is also N:1. Entity seat is related to Reservation and the cardinality ratio is N. It has a key attribute of seat number. The reservation entity has the attributes, customer name and customer phone.   The relationship box is related to leg instance and has a cardinal ratio of 1.  A note below the diagram reads, A Leg is a nonstop portion of a flight. A LEG INSTANCE is a particular occurrence of a LEG on a particular date.**

Airport is required to have one departure and arrival airports.

These DEPT and ARRIV airports can have N flight\_legs.

Flight\_legs can have N legs, but each legs is only allowed one flight.

Each Flight is allowed 1 Fares, but Fares is allowed N FARE.

Flight\_LEG also breaks into INSTANCE\_OF which it's allwoed one.

This instance of is a LEG\_INSTANCE, which is allowed N departs and arrives. It's also allowed N ASSIGNED airplanes and 1 RESERVATION.

RESERVATION can have N seats to it.

ASSIGNED can only have 1 AIRPLANE to it.

AIRPLANE can be of N TYPE, but TYPE is only allowed 1 AIRPLANE\_TYPE

AIRPLANE\_TYPE can be N CAN\_LAND

CAN\_LAND can be M AIRPORT.

Other small things are to note that for a FLIGHT\_LEG to exist, it requires a minimum of 1 INSTANCE\_OF.

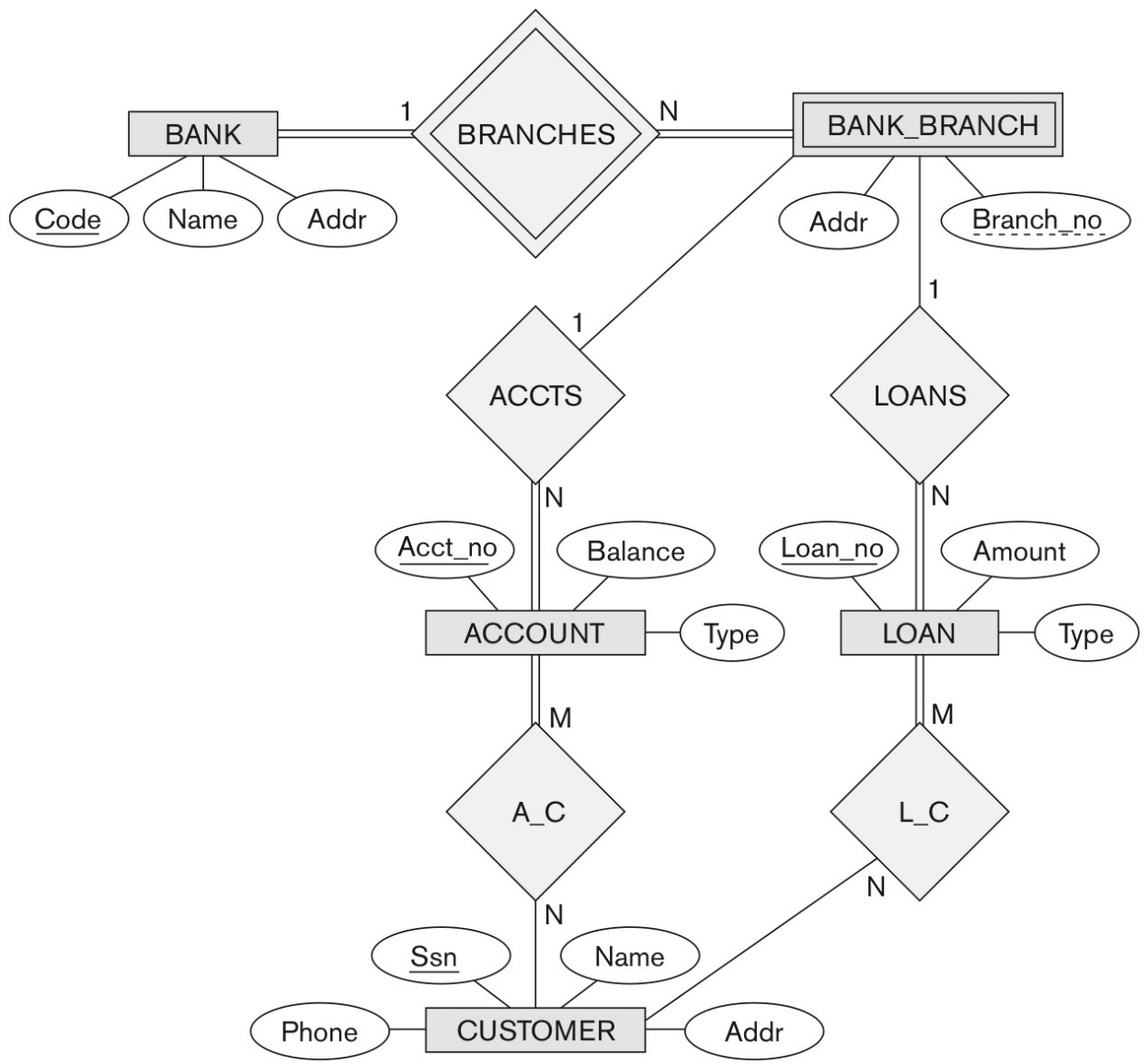
The same is true for AIRPORT requiring DEPARTURE and ARRIVAL.

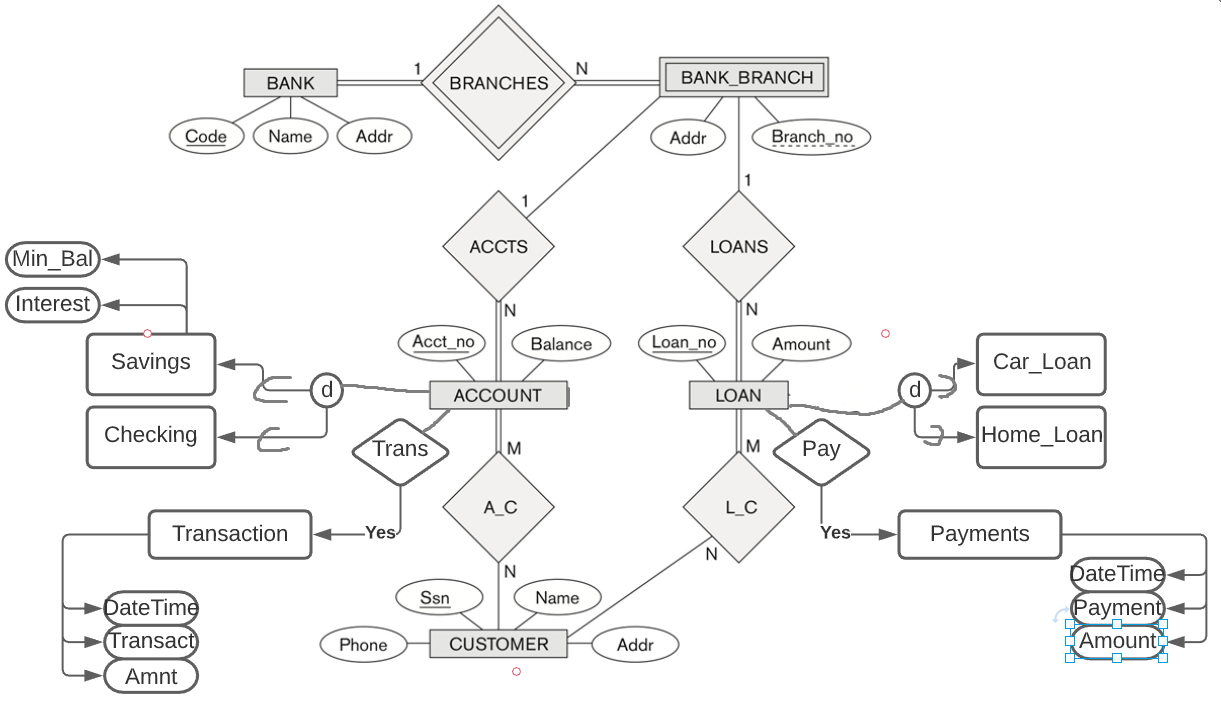
AIRPORT requires 1 DEPARTS and 1 ARRIVES, but that can be N number of LEG\_INSTANCEs.

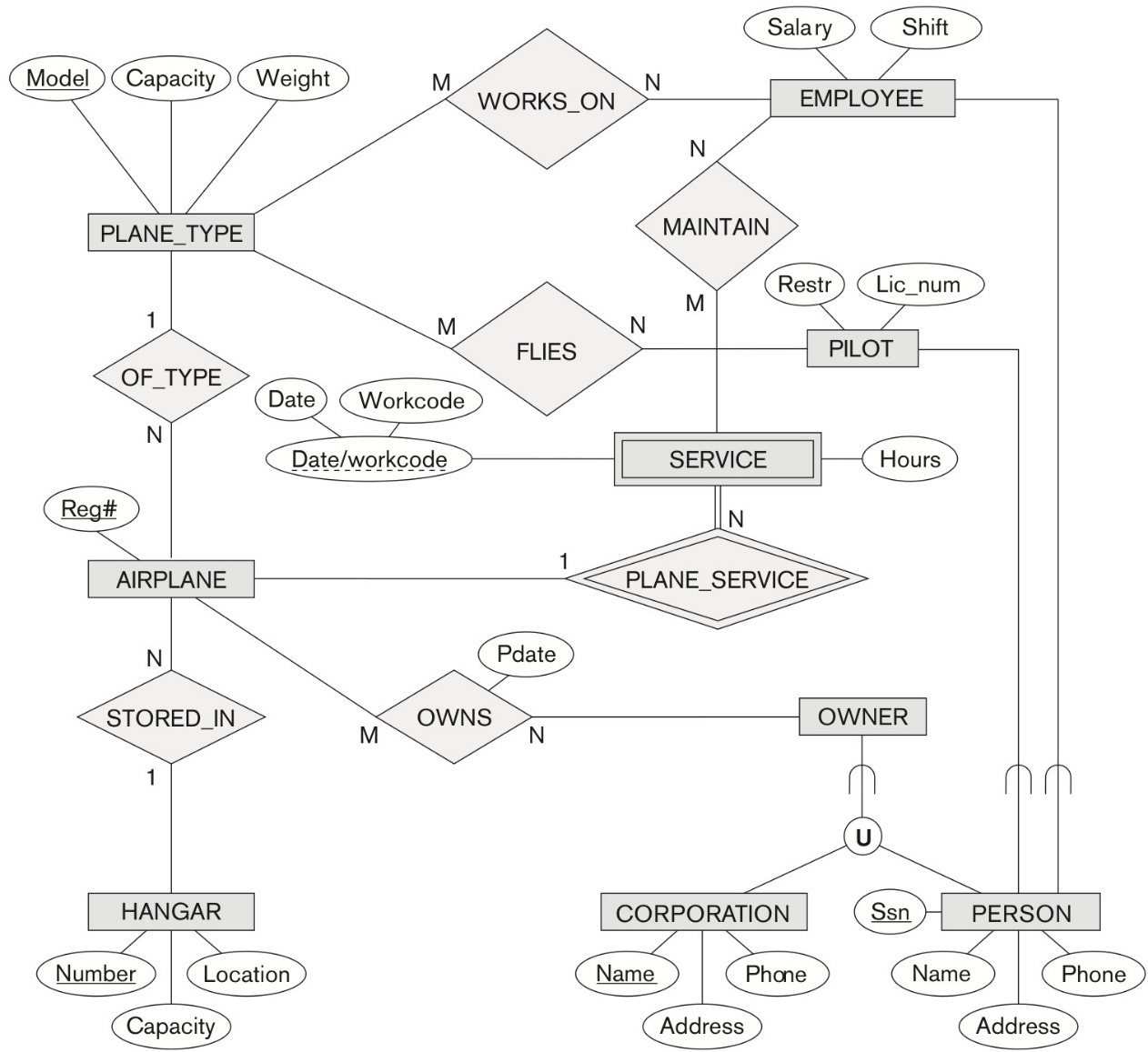
Each LEG\_INSTANCE requires 1 RESERVATION, but each RESERVATION can be N seats.

**Question 3 (chapter 4) Consider the BANK ER schema of following figure, and suppose that it is necessary to keep track of different types of ACCOUNTS (SAVINGS\_ACCTS, CHECKING\_ACCTS,**

**...) and LOANS (CAR\_LOANS, HOME\_LOANS, ...). Suppose that it is also desirable to keep track of each account's TRANSACTIONs (deposits, withdrawals, checks, ...) and each loan's PAYMENTs; both of these include the amount, date, time, ... Modify the BANK schema, using ER and EER concepts of specialization and generalization. State any assumptions you make about the additional requirements.**

****

**Question 4(Chapter 4) The following figure shows an example of an EER diagram for a small private airport database that is used to keep track of airplanes, their owners, airport employees, and pilots. From the requirements for this database, the following information was collected: Each airplane has a registration number [Reg#], is of a particular plane type [of\_type], and is stored in a particular hangar [stored\_in]. Each plane\_type has a model number [Model], a capacity [Capacity], and a weight [Weight]. Each hangar has a number [Number], a capacity [Capacity], and a location [Location]. The database also keeps track of the owners of each plane [owns] and the employees who have maintained the plane [maintain]. Each relationship instance in owns relates an airplane to an owner and includes the purchase date [Pdate]. Each relationship instance in maintain relates to an employee to a service record [service]. Each plane undergoes service many times; hence, it is related by [plane\_service] o a number of service records. A service record includes as attributes the date of maintenance [Date], the number of hours spent on the work [Hours], and the type of work done [Workcode]. We use a weak entity type [service] to represent airplane service, because the airplane registration number is used to identify a service record. An owner is either a person or a corporation. Hence, we use a union type (category) [owner] that is a subset of the union of corporation [Corporation] and person [Person] entity types. Both pilots {Pilot] and employees [Employee] are subclasses of person. Each pilot has specific attributes license number [Lic\_Num] and restrictions [Restr]; each employee has specific attributes salary [Salary] and shift {Shift]. All person entities in the database have data kept on their social security number [Ssn], name [Name], address [Address], and telephone number [Phone]. For corporation entities, the data kept includes name [Name], address [Address], and telephone number [Phone]. The database also keeps track of the types of planes each pilot is authorized to fly [Flies] and the types of planes each employee can do maintenance work on [Works\_on]. Show how the small airport EER schema may be represented in UML notation.**

******

