Chapter 6

Exercise 6.1

We wish to automate the manager of loans in a library. The specification of the application, acquired through an interview with the librarian, is show in Figure 6.15.
Analyze the specification, filter the ambiguities and then group them according to type. Pay particular attention to the difference between the concept of book and copy of book.
Identify the logical links between the various groups of specifications thus obtained.

The Library
A reader who uses the library has an identity card on which is written his or her code, name and address. The users makes requests for the loan of books catalogued in the library. Each book has a title and a list of authors and there can be many copies of any book. Each book in the library is identified by a code. Following a request, the archive of available books is first consulted (that is. those not out on loan at present). If the book is available, we look for the book on the shelves. *Once the book is found it is given to the reader. The text is then classified as one on loan. When the reader has finished, the book is returned, put back on the shelves and re-classified as available. For each loan the times and date of taking out and returning are recorded.*

Figure 6.15

Solution:

Term	Description	Synonym	Links
Reader	A person who loans copy of books in the library	User	Copy of Book, Loan
Book	Type of book present in the library. The library has one or more copies of the same book.		Copy of Book
Copy of book	Each copy of book present in the library. May be loaned by a reader	Book, Text	Book, Reader, Loan
Loan	A loan made by a reader; each loan refers to one copy of book		Reader, Copy of Book

PHRASES RELATED TO READERS

A reader who uses the library has an identity card on which his or her code, name and
Address are written.
The readers make request for the loan of copy of books catalogued in the library.

PHRASES RELATED TO BOOKS

Each book has a title and a list of authors, and there can be many copies of any book.

PHRASES RELATED TO BOOK COPIES

Each copy of a book in the library is identified by a code.
Following a request, the archive of available copy of books is first consulted.
The archive of available copy of books contains all copy of books which are not out on loan.
If the copy of book is available, we look for the copy of book on the shelves.

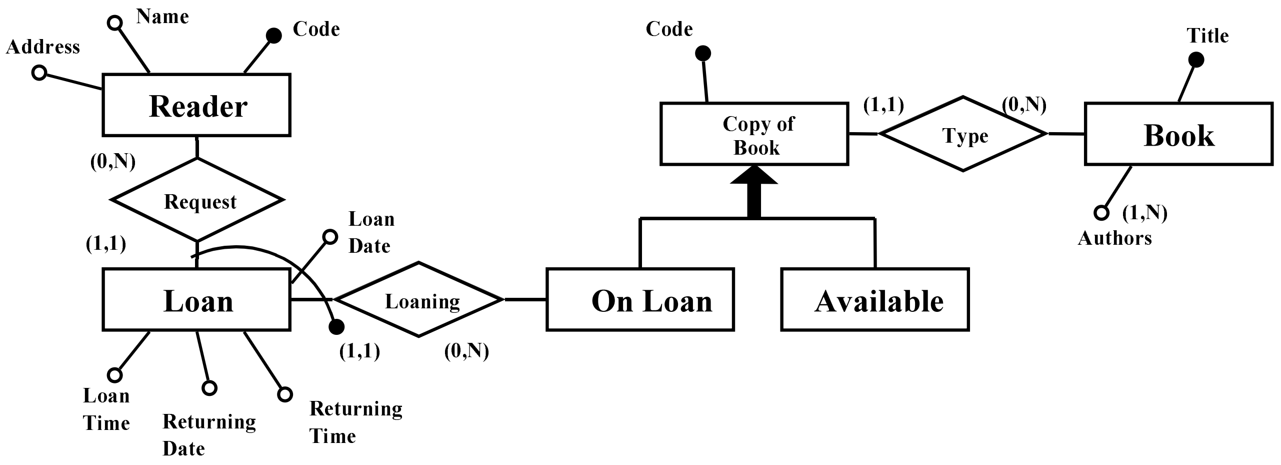
PHRASES RELATED TO LOANS

Once the copy of book is found, it is given to the reader.
The copy of book is then classified as a loan.
For each loan the times and dates of taking out and returning are recorded.

Exercise 6.2

Represent the specification of the previous exercise using the E-R model.

Solution:



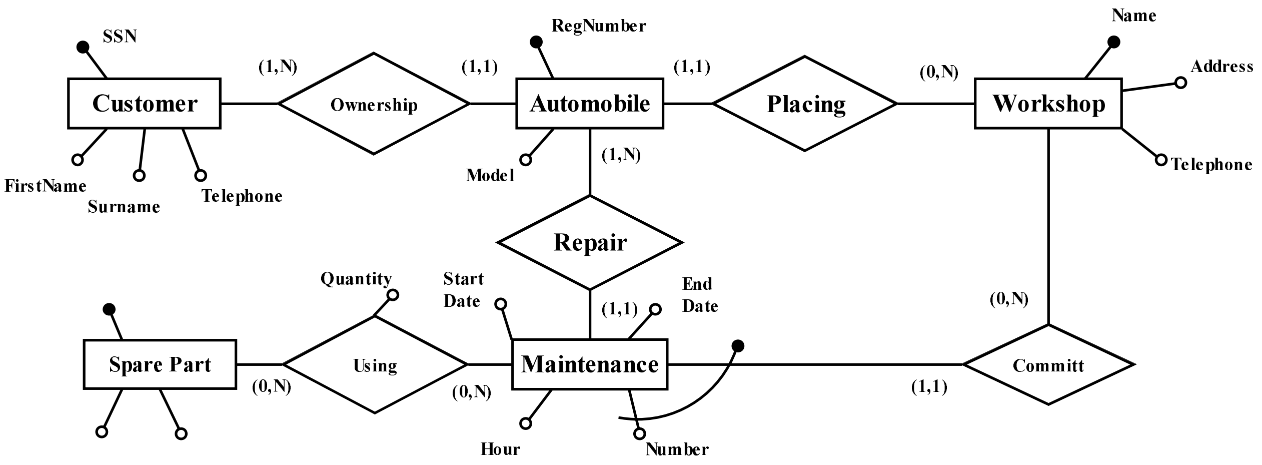
Exercise 6.3

Define an Entity-Relationship schema that describes the data of an application relating to a chain of automobile workshop. We are interested in:

- The workshop, with name (identifying), address and telephone number;
- The automobiles, with registration number (identifying) and model (a string of characters with no further structure) and owner;
- The customers (automobile owners), with social security number, surname, first name and telephone; each client can be the owner of more than one automobile;
- The maintenance work carried out in a workshop, with a number (unique in a particular workshop), start date and end date, parts replaced (with quantities) and number of hours labour;
- The spare parts available with code, name and unit cost.

Show the cardinalities of the relationships and (at least) one identifier for each entity.

Solution:



Exercise 6.4

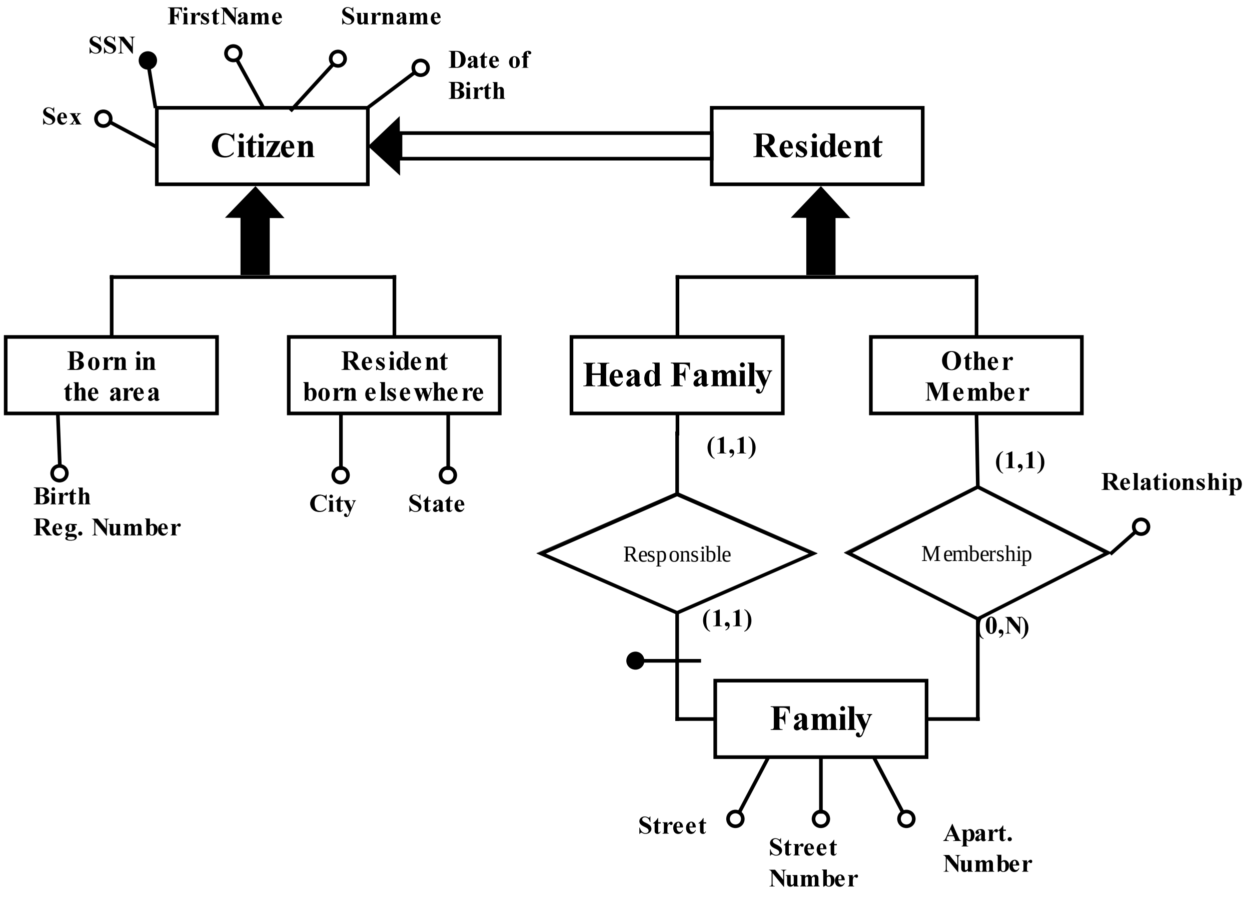
Define an E-R schema that describes the data of an application relating to the electoral roll of the city of WhoKnowWhere, with citizens and families. The following are stored:

- Information on the citizen born in area and those resident in the area; each citizen is identified by a social security number, and has surname, first name, sex and date of birth; furthermore:
  - for anyone born in the area, the birth registration number is also stored;
  - for anyone born in another area, the city and state of birth are stored.

- Information on resident families, each of which has one and one only head of the family and zero ore more other members; for each of them, the relationship to the head of the family is recorded (spouse, child, parent or other); each resident citizen belongs to one and one family only; all the members of a family have the same residence (street, street number, apartment number).

Try to use the inside-out strategy. At the end, verify the quality of the schema obtained.

Solution:



Using the inside-out strategy, the creation of this schema starts with the entity CITIZEN; around it, we can add the two specializations and the specializations of RESIDENT (this entity represents all the residents, wherever they are born).

Finally, we can add the entity FAMILY with the two relationship RESPONSIBLE and MEMBERSHIP.

This schema is correct and complete, because it represents all the specifications, with correct constructs.

Exercise 6.5

Analyze the specifications relating to matches of a soccer tournament shown In Figure 6.16 and construct a glossary of terms.

Soccer Tournament
For each match, we store the series and the day on which it takes place, which match it is (e.g. first match, second match etc.) the date with day, month, year, the teams involved in the match with the name of the city and the trainer, and finally for each team whether played at home. We store the name and the surname of each player in each team with his date of birth and main position. We store, for each day, how many points each time has and we also store, for each match, the players of each team who played and in which position each player played (the position can change from one game to another). For each match, we store the referee, with first name, surname, city and region of birth. The matched played as scheduled must be distinguished from those postponed. For a postponed match, we store the date in which it is actually played. We also identify the matches played in a city other than that of the home team; for each of these, we store the city in which it took place, as well as the reason for the variation venue. For each player, we are interested in the city of birth.

Figure 6.16

Solution:

<u>Glossary of terms</u>

Term	Description	Synonym	Links
Match	A match played in the tournament; may be postponed or played on neutral ground.	Game	Player, Team, Tournament Day, Referee
Tournament day	A tournament day; it includes many matches; this term is very different from ‘day’ of a date; each tournament day has a date (day, month, year).	Day	Match, Team
Team	A team which plays in the tournament.		Player, Match, Tournament Day
Player	A player who plays with a team; It is important to know in which match he played and in which positions.		Team, Match
Referee	A referee who referees match in the tournament		Match

Exercise 6.6

Having organized the specifications of Exercise 6.5 into groups of similar type, show them using an Entity-Relationship model, using a top-down strategy starting from an initial skeleton schema. Note that the schema in Figure 5.26 represents a possible solution to this exercise.

Solution:

PHRASES RELATING THE MATCH AND TOURNAMENT DAY

For each match, we store the series and the tournament day on which it takes place, which match it is (e.g. first match, second match etc.) the date with day, month, year.
The matched played as scheduled must be distinguished from those postponed. For a postponed match, we store the date in which it is actually played.
We also identify the matches played in a city other than that of the home team; for each of these, we store the city in which it took place, as well as the reason for the variation venue.

PHRASES RELATING TO THE REFEREE

For each match we store the referee, with first name, surname, city and region of birth.

PHRASES RELATING TO THE TEAMS

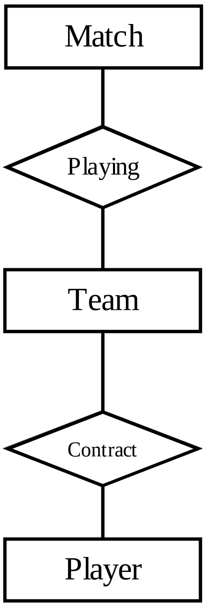
For each match we store the teams involved, with the name of the cities and trainers, and for each team whether played at home.
We store, for each tournament day, how many points each team has.

PHRASES RELATING TO PLAYERS

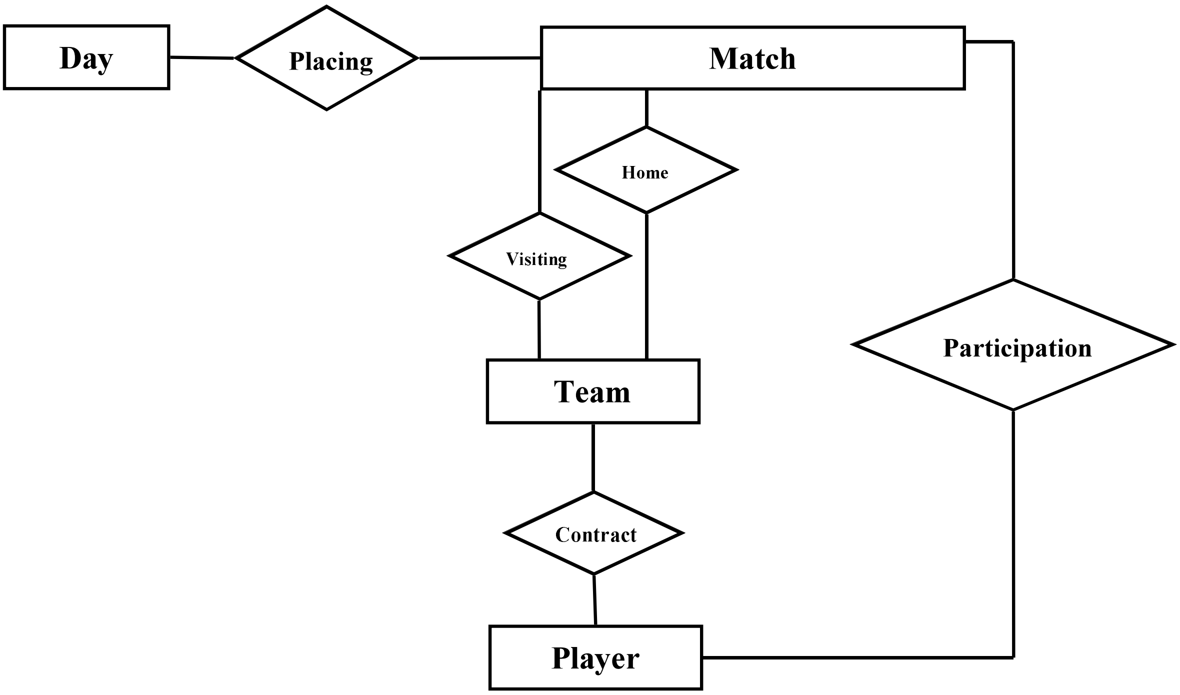
We store the name and the surname of each player in each team, with his date of birth and main position. For each match, we store the players of each team who played, and in which position each player played (the position can change from one game to another).
For each player we are interested in the city of birth.

The following schemas represents the steps of the construction of the final schema, using a top-down strategy.

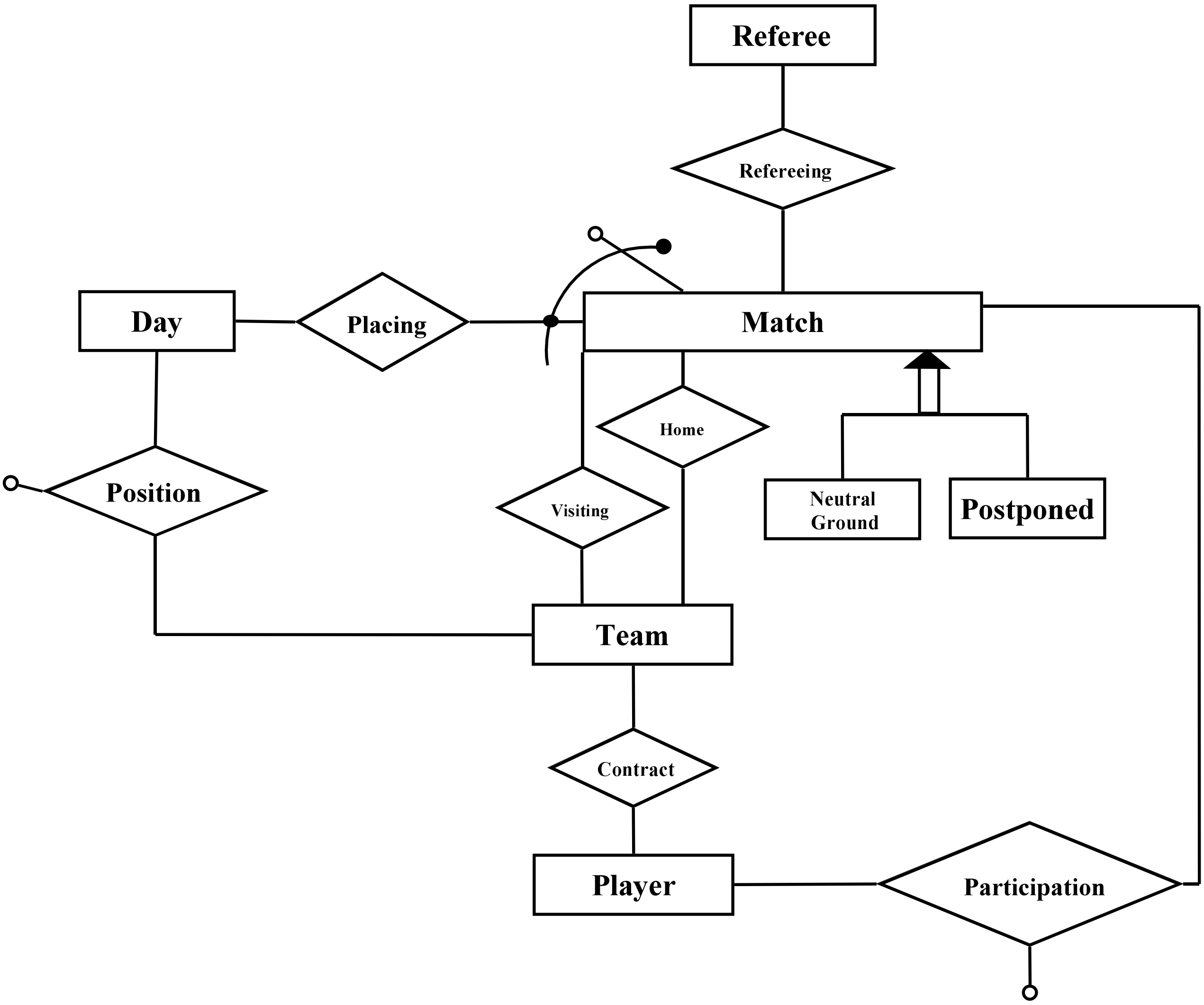
1) Skeleton Schema



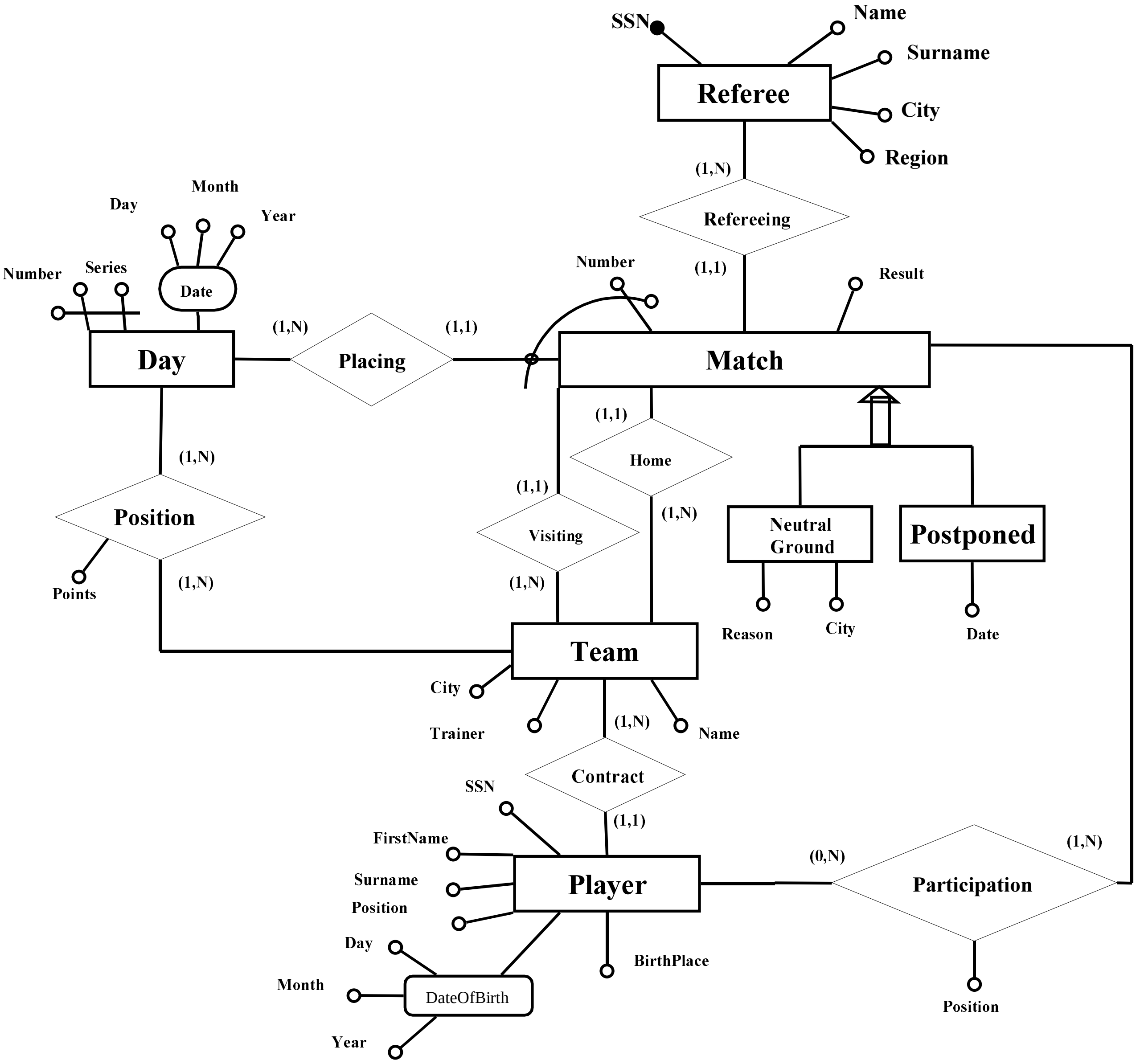
2)


Day
Placing
Match
Home
Visiting
Participation
Team
Contract
Player

3)


Referee
Refereeing
Day
Placing
Match
Home
Neutral Ground
Postponed
Position
Visiting
Team
Contract
Player
Participation

4)


SSN
Name
Referee
Surname
City
Region
(1,N)
Refereeing
(1,1)
Month
Day
Year
Number
Series
Date
Number
Result
(1,N)
(1,1)
Day
Placing
Match
(1,1)
Home
(1,N)
(1,1)
Position
Visiting
(1,N)
Neutral Ground
Postponed
Points
(1,N)
(1,N)
Reason
City
Date
Team
City
Trainer
(1,N)
Name
Contract
SSN
(1,1)
FirstName
Surname
Position
Player
(0,N)
Participation
(1,N)
Day
BirthPlace
Month
DateOfBirth
Position
Year

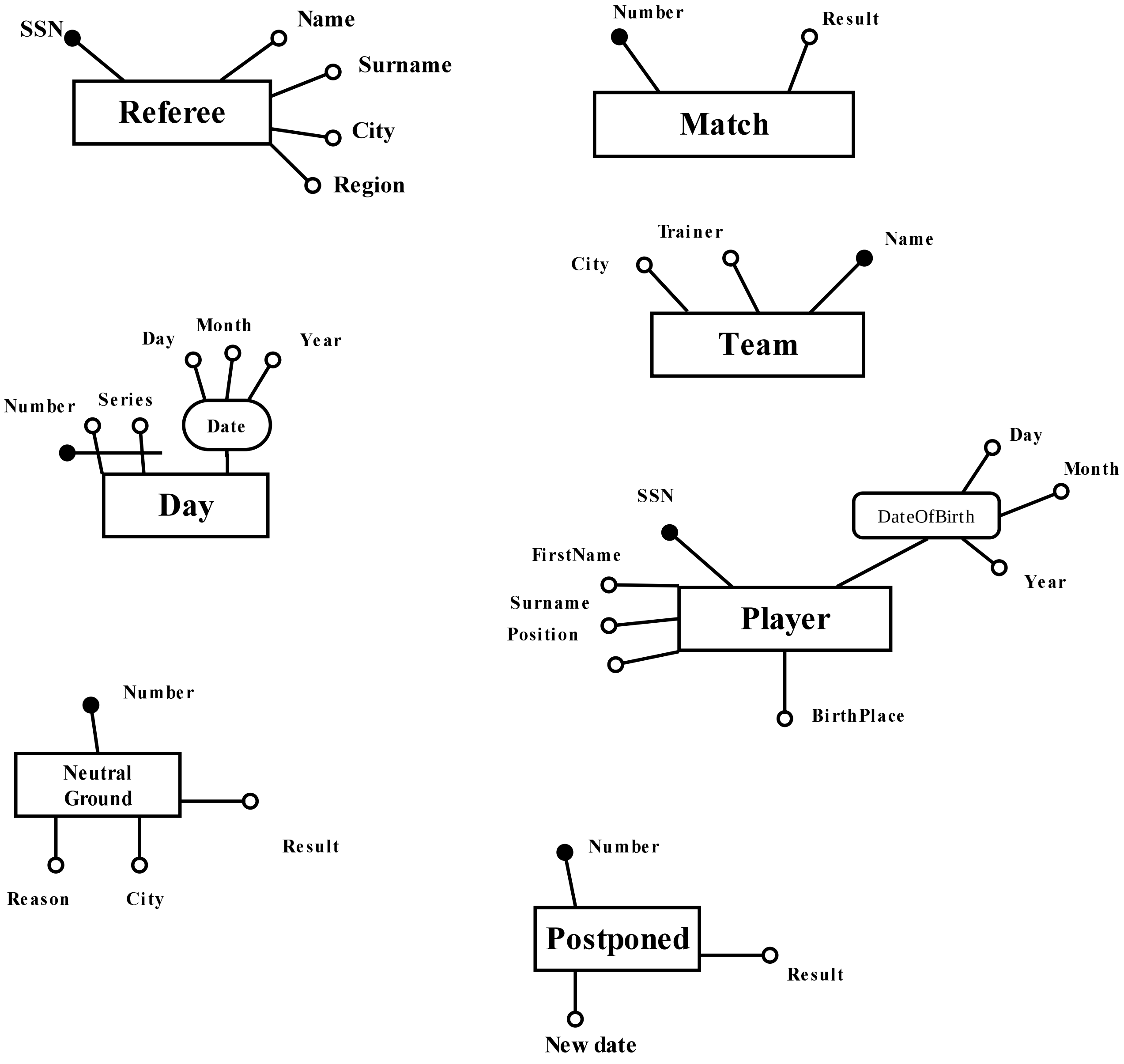
Exercise 6.7

Try the represent again the specification in Figure 6.16 with an Entity-Relationship schema, using a bottom-up strategy this time. Construct separate fragments of the schema that describe the various homogenous components of the specification and then proceed with the integration of the various schemas. Compare the result with the schema obtained from Exercise 6.6.

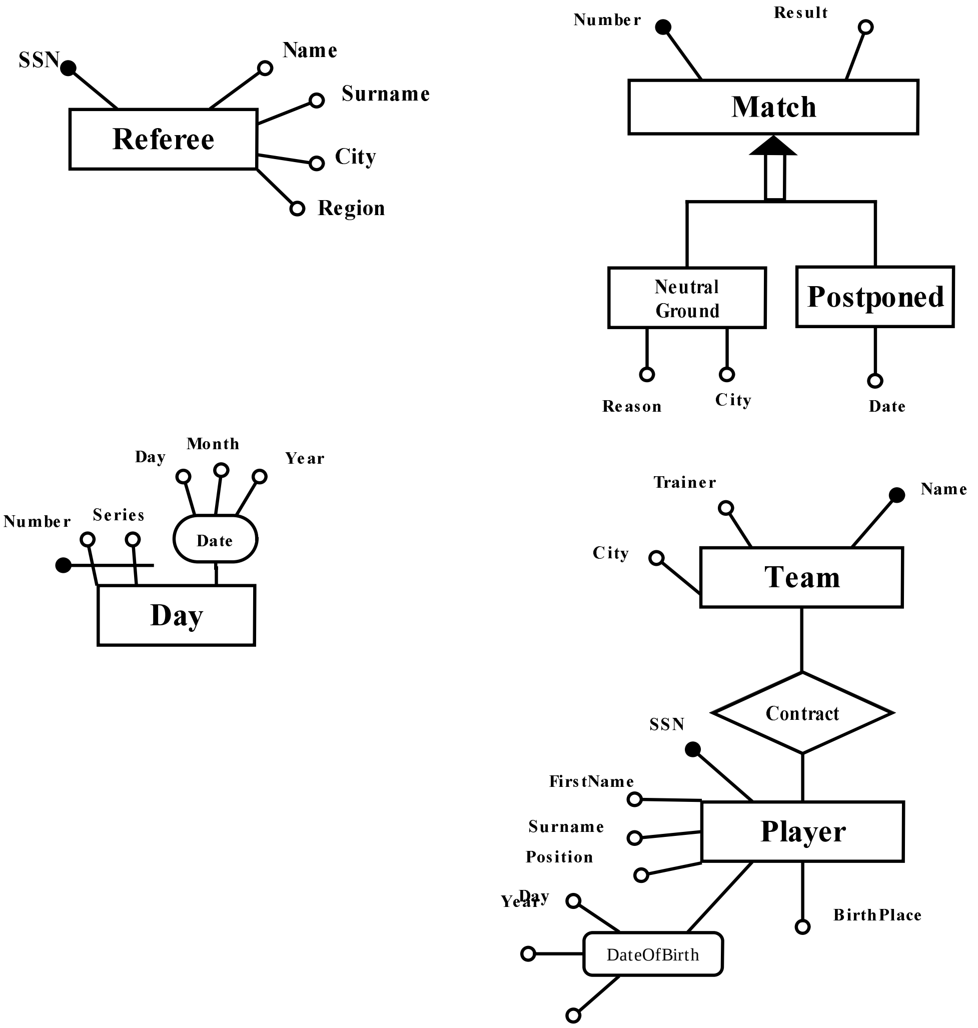
Solution:

The following steps may represent the creation of the final schema using a bottom-up strategy. The schema so obtained is the same as Exercise 6.6.

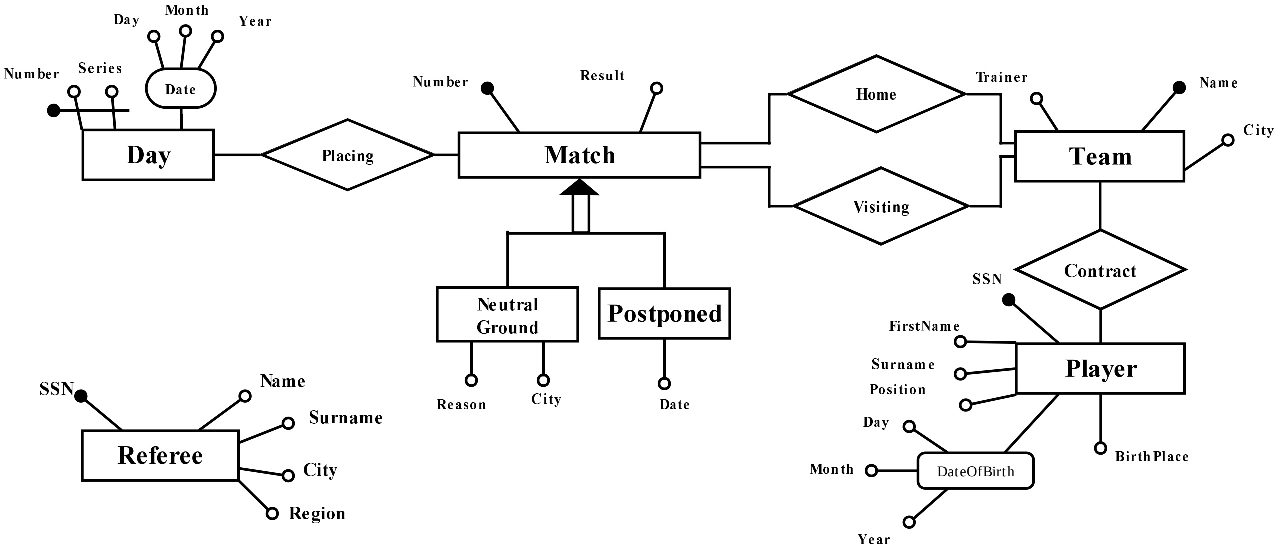
1)



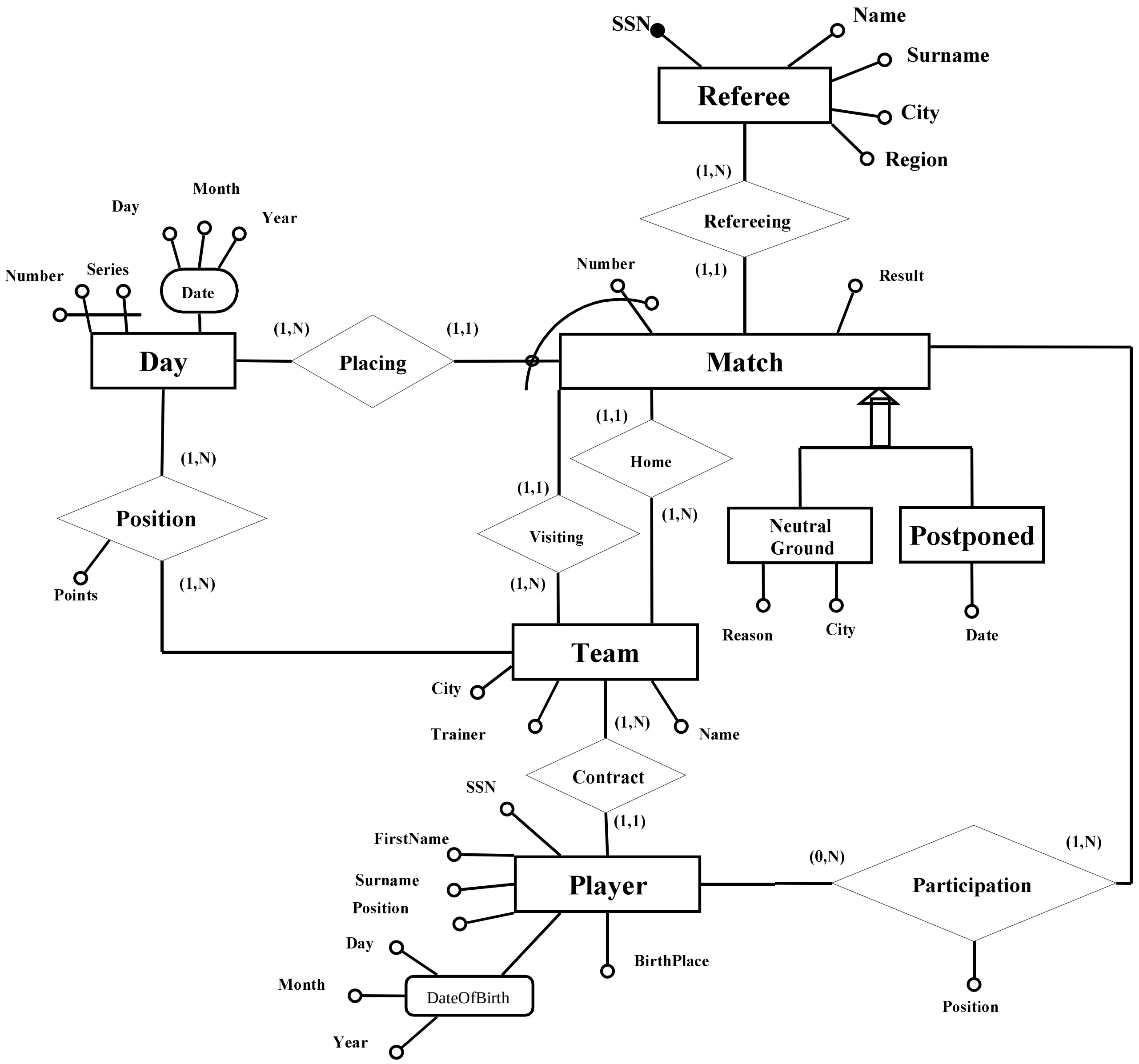
2)


SSN
Name
Surname
City
Region
Referee
Number
Result
Match
Neutral Ground
Postponed
Reason
City
Date
Day
Month
Year
Number
Series
Date
Day
Trainer
Name
City
Team
Contract
SSN
FirstName
Surname
Position
Player
Year
Day
BirthPlace
DateOfBirth

3)


Day
Month
Year
Number
Series
Date
Day
Placing
Number
Result
Match
Home
Visiting
Trainer
Name
City
Team
Neutral Ground
Postponed
Reason
City
Date
Contract
SSN
FirstName
Surname
Position
Player
Day
Month
DateOfBirth
BirthPlace
Year
SSN
Name
Surname
Referee
City
Region

4)


SSN
Name
Surname
Referee
City
Region
(1,N)
Refereeing
(1,1)
Month
Day
Year
Number
Series
Date
Number
Result
(1,N)
Placing
(1,1)
Day
Match
(1,1)
Home
(1,N)
(1,1)
Position
Visiting
(1,N)
Neutral Ground
Postponed
Points
(1,N)
(1,N)
Reason
City
Date
Team
City
Trainer
(1,N)
Name
Contract
SSN
(1,1)
FirstName
Surname
Player
(0,N)
Participation
(1,N)
Position
Day
BirthPlace
Month
DateOfBirth
Position
Year

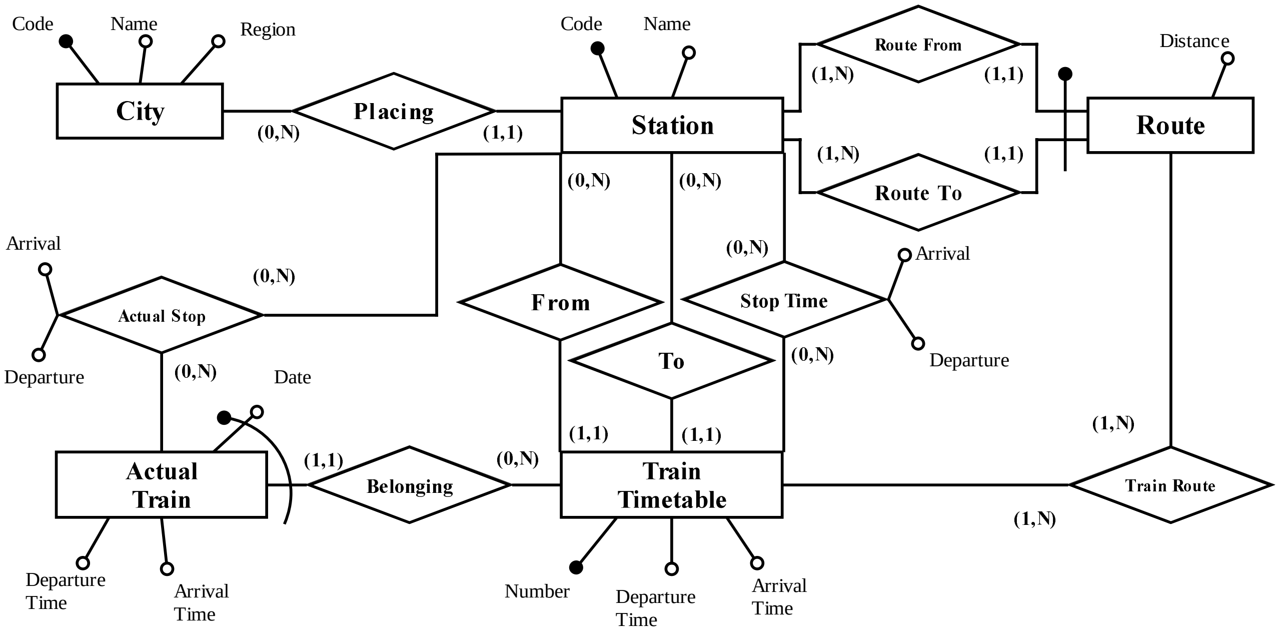
Exercise 6.8

We wish to carry out a reverse engineering operation. That is, given a relational database, we wish to construct its conceptual representation using the E-R model. The database is for an application concerning trains and railway stations and is made up of the following relations:

- STATION (Code, Name, City), with a referential constraint between the attribute **City** and CITY relation;
- CITY (Code, Name, Region);
- ROUTE (From, To, Distance), with referential constraints between the attributes **From** and the relation STATION and between the attribute **To** and the relation STATION; this relation contains all and only the pairs of stations connected directly by a route (that is without intermediate stations);
- TRAINTIMETABLE(Number, From, To, DepartureTime, ArrivalTime) with referential constraints between the attributes **From** and the relation STATION and between the attribute **To**, and the relation STATION;
- TRAINROUTE(TrainNumber, From, To) with referential constraints between the attribute **TrainNumber** and the relation TRAINTIMETABLE and between the attributes **From** and **To** and the relation STATION
- STOPTIME(TrainNumber, Station, Arrival, Departure) with referential constraints between the attribute TrainNumber and the relation TRAINTIMETABLE and between the attribute **Station** and the relation STATION;
- ACTUALTRAIN(TrainNumber, Date, DepartureTime, ArrivalTime) with a referential constraints between the attribute **TrainNumber** and the TRAINTIMETABLE relation;
- ACTUALSTOP (TrainNumber, Date, Station, Arrival, Departure) with a referential constraints between the two attributes **TrainNumber** and **Station** and the STOPTIME relations.

Indicate possible redundancies.

Solution:



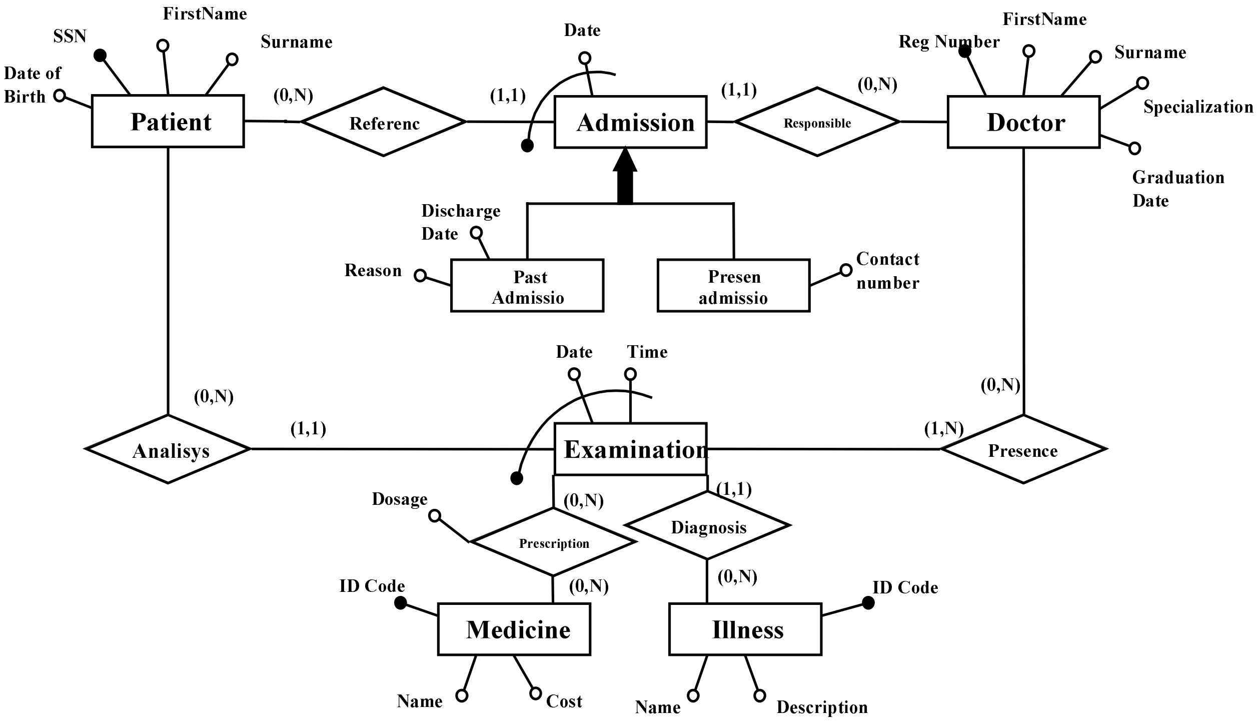
The relationship TRAINROUTE contains redundant information, because the train routes can be obtained from the relationships FROM, TO and STOP. However this operation could be quite difficult, and information about train routes could be often requested in such a database; so the redundancy is useful to improve the performance (the optimization of a database will be treated in Chapter 7); moreover the concept of route is in any case important in this context, and so it is correct to represent it in the E-R schema.

Exercise 6.9

Define an E-R schema that describes the data of an application for a hospital ward. We are interested in:

- Each patient, with social security number, first name, surname and date of birth;
- The admission of each patient, with admission date (an identifier for each admission of a patient) and doctor in charge of the case: as well as, for a discharge, the date of discharge, the date of discharge and reason (cure, transfer etc.), and, for a patient in the ward at present, the contact number of a relative (which we can assume is simply a string);
- Each doctor, with registration number, surname, first name, specialization and graduation date;
- Each examination, with date, time, doctor present, medicines prescribes (with dosages) and the illness diagnosed – each examination is identified by the patient involved and by the date and time;
- For each medicine, the identification code, the name and cost;
- For each illness, an identifying code, a name and a description.

Solution:

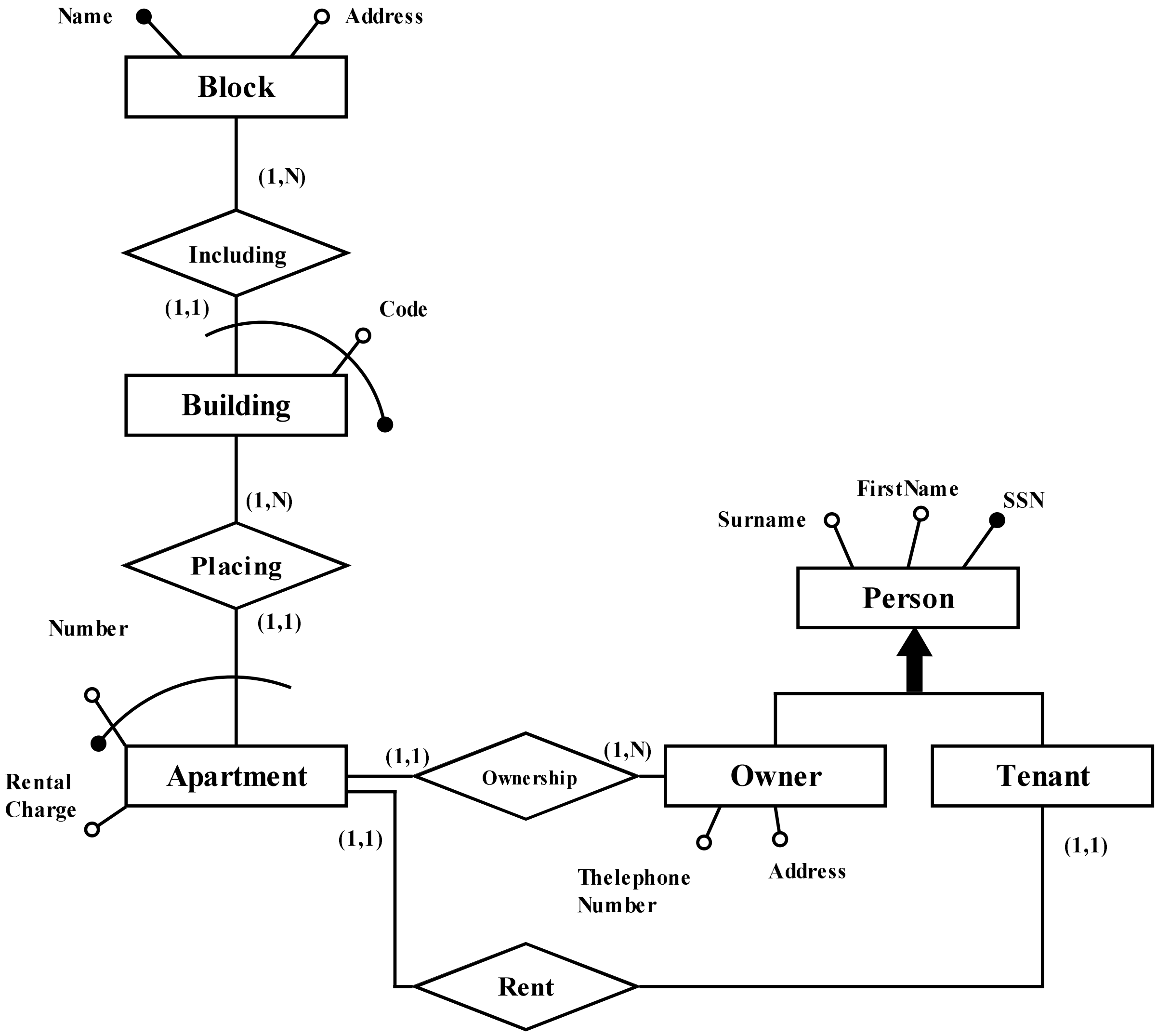


Exercise 6.10

Define an Entity-Relationship schema that describes the data for an application for the management of apartment blocks, according to the following specification:

- Each apartment block has a name (which identifies it) and an address, and has one or more buildings, each of which contains a number of apartments;
- If the apartment block has more than one building, each of them has a code (e.g. building 'A') which identifies it together with the name of the block;
- Each apartment is identified, in its respective block, by the building (if many exist) and by a number (the apartment number) and has a rental charge;
- Each apartment has an owner and a tenant; for both of them, we are interested in the name, surname, social security number; for the owners we want to store also their address and telephone numbers; each person can be the owner of many apartments but can be the tenant of one apartment only.

Solution:



The generalization is not exclusive, because owners of apartments can be also tenant of other apartments.