Carlo Batini
University of Milano Bicocca, Italy
batini@disco.unimib.it

Anatomy of an assignment in conceptual design

Introduction to this presentation - 1

Dear user of my book on «Database Modeling and Design», I have produced this Power Point presentation to analyze in detail the outcome of an exam assignement organized in April 2016 for my introductory course in Data Bases. 45 minutes were given to students to provide a solution. About 130 students attended the exam.

The exam assignment (for short, assignment in the following) refers to the design of an Entity Relationship schema. The domain of interest is an information system for managing a bike sharing service in a municipality. Requirements provided to students are reproduced in the next page.

Text of the assigment

A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking lot and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type). Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes). When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

In order to use bikes, users have to register themselves. A registered user is characterized by an identifier (e.g. a fiscal code, or a social security number), a progressive numeric code (first person registered, second person registered, etc.) a name, a surname, date of birth, municipality of birth, with code, name of municipality and region (we assume that municipalities are located in regions).

Everytime a user picks up and subsequently returns a bike, such an event (pick up and return of a bike by a user) has to be recorded. The event is identified by a progressive number in the year (e.g. first event in 2015, second event in 2015, etc.) and duration (e.g. 12 minutes, 47 minutes); such duration has to be explicitely represented in the data base. The minute of pick up and the minute of return have to be associated to the event, besides, of course, the bike used and the user involved.

One or more credit cards are associated to users, with type (e.g. Visa, Diners), progressive code in the card type, and daily maximum amount of withdrawal (that depends on the card and on the person).

You have to represent above requirements with an Entity Relationship Schema, in terms of entities, relationships, attributes, internal and external idetitifers, generalizations, is-a relationships, minimum and maximum cardinalities.

Introduction to this presentation - 2

Most relevant difficulties in the assignment concerns the various relationships among the physical part of the bike sharing service, namely the bike, and all other organizational, spatial and temporal issues related to the bike sharing service.

In my lessons, I had used this year as background material the six parts of my course on «Database Modeling and Design», freely downloadable at http://hdl.handle.net/10281/97114

The complexity of the assignment and the diversification of student solutions lead me to consider this case as a potentially very effective training ground, that allows me to highlight typical errors and difficulties inherent in conceptual design. This presentation can be used by students as «yet-another exercise» in conceptual design, or also as self assessment. It is also useful for instructors, as an example of feedback on the difficulties that student face in understanding the concepts of the ER Model.

Goals of the presentation

- For the student
  - Discuss a challenging exercise in conceptual design.
  - Apply methods shown in Part 4: Conceptual Design of the referenced book downloadable at http://hdl.handle.net/10281/97114
  - Make a self assessment in conceptual design, that allows the student to fix the level of maturity achieved in the discipline.
- For the instructor
  - Assign to students a case study that can be discussed as a class work.
  - Apply a model that allows to measure the complexity of an exercise in conceptual design, in such a way to be able to properly size an exam assignment.

Contents of the following slides

1. Exercise proposed to students
2. Background knowledge of students when they made the assignment
3. The diagrammatic representation adopted for the Entity Relationship Model
4. Requirements and reference correct solution
5. A process of incremental design of the correct solution(s), made of several phases.
6. For each phase and related subschema produced, a set of alternative correct solutions, and a set of wrong solutions, and for each of them the decrease in score associated in comparison to the maximum score, expressed in a [0..30] scale.
7. Statistics on some types of errors and related involved qualities.
8. Some indicators to measure and benchmark the complexity of the assignment.

1. Exercise

- Try to produce the ER schema based on previous requirements in 45 minutes. In this way you will experiment the same difficulties that had my students…..
- When you finish, go to the next page →

2. Background knowledge of students when they made the assignment

When making the assignment, students had attended Part 0 to Part 4 of the course on Database Modeling and Design

Part 1 - Basic Concepts
- Data
- Information
- Basic Structure of a Computer
- Traditional Data Management
  - Redundancy
  - Inconsistency
- Data Base - 1
- Data Base Model
  - Relational Model
    - Relation
    - Attribute
- Data Base - 2
  - Schema
  - Instance
- Data Base Technology
  - Data Base Management System
  - SQL-Data Description Language
  - SQL-Data Manipulation Language
- Data Base Design
  - Design Phase
    - Conceptual Design
      - Entity Relationship Model
    - Logical Design
  - Set up, Query and Update Phase
- Methodology

Part 2 - Entity Relationship Model
- Diagrammatic Representation
- Modeling Construct
  - Entity
  - Attribute of Entity
  - Relationship
  - Attribute of Relationship
  - Min/Max Cardinality of an Entity in a Relat.
  - Is-a Relationship between two Entities
  - Generalization Hierarchy among n Entities
  - Inheritance Property
  - Identifier of an Entity
    - Internal Identifier
    - External Identifier

Part 3 - Relational Model
- Relation
  - Schema
  - Instance
  - Tuple
- Attribute
  - Domain of an Attribute
  - Value of an Attribute
- Data Base
  - Schema
  - Instance
- Incomplete information
- Integrity Constraint
  - Intra Relation Int. Constraint
    - Domain Constraint
    - Tuple Constraint
    - Key
    - Primary Key
  - Inter Relation Int. Constraint
    - Referential Integrity
- Rel. Data Base Quality: Normalization
  - Boyce Codd Normal Form
  - Normalization Process

Part 4 - Conceptual Design
- Schema Quality Dimension
  - Correctness with respect to the Model
  - Correctness with respect to Requirements
  - Minimality (Redundancy)
  - Completeness
  - Pertinence
  - Readability
    - Diagrammatic Readability
    - Compactness
  - Normalization
- Design Strategy
  - Bottom Up
  - Top Down
  - Oil Stain
  - Mixed

Part 5 - Logical Design
- Phase of Logical Design
  - Simplification and Optimization
  - Translation
- Simplification Phase
  - Removing Generalizations
- Optimization Phase
  - Partitoning of Entities/Relationships
  - Merging of Entities/Relationships
- Translation Step
  - (Translation of an) Entity
  - Many to Many Relationship
  - One to Many Relationship
  - Ternary Relationship
  - Recursive Relationship
  - (1,1) to (1,1) Relationship
  - (0,1) to (1,1) Relationship
  - (0,1) to (0,1) Relationship

Concepts learnt in Part 2

Part 2 - Entity Relationship Model

Diagrammatic Representation

Modeling Construct

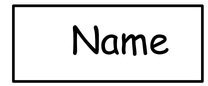
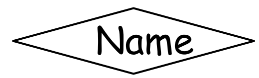
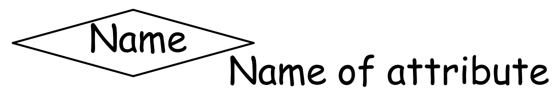
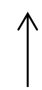
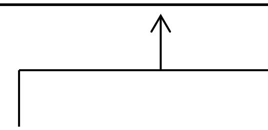
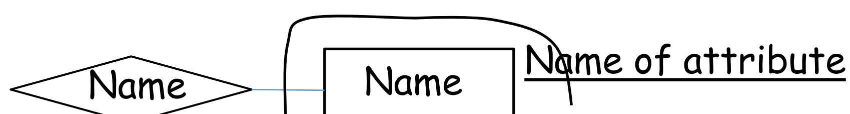
- Entity
- Attribute of Entity
- Relationship
- Attribute of Relationship
- Min/Max Cardinality of an Entity in a Relationship
- Is-a Relationship between two Entities
- Generalization Hierarchy among n Entities
- Inheritance Property
- Identifier of an Entity
  - Internal Identifier
  - External Identifier

Schema Quality Dimensions learnt in Part 4

- Correctness with respect to the Model
- Correctness with respect to Requirements
- Minimality (Redundancy)
- Completeness
- Pertinence
- Readability
  - Diagrammatic Readability
  - Compactness

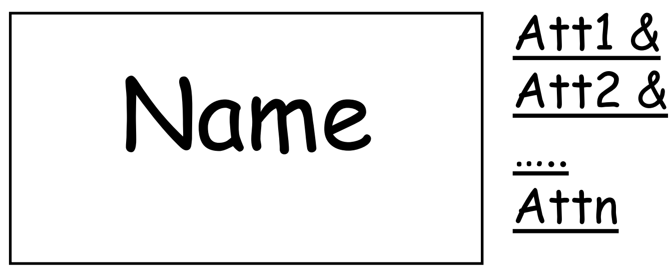
3. Diagrammatic representation adopted

Diagrammatic representation

Construct of the ER model	Diagrammatic representation
Entity	
Attribute of entity	
Relationship	
Attribute of relationship	
Is-a hierarchy	
Generalization hierarchy	
Internal Identifier	
External Identifier	

Furthermore…

- When the internal identifier is made of two/n attributes we will write:



4. Requirements and reference correct solution

Text of the assigment

A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking lot and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type). Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes). When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

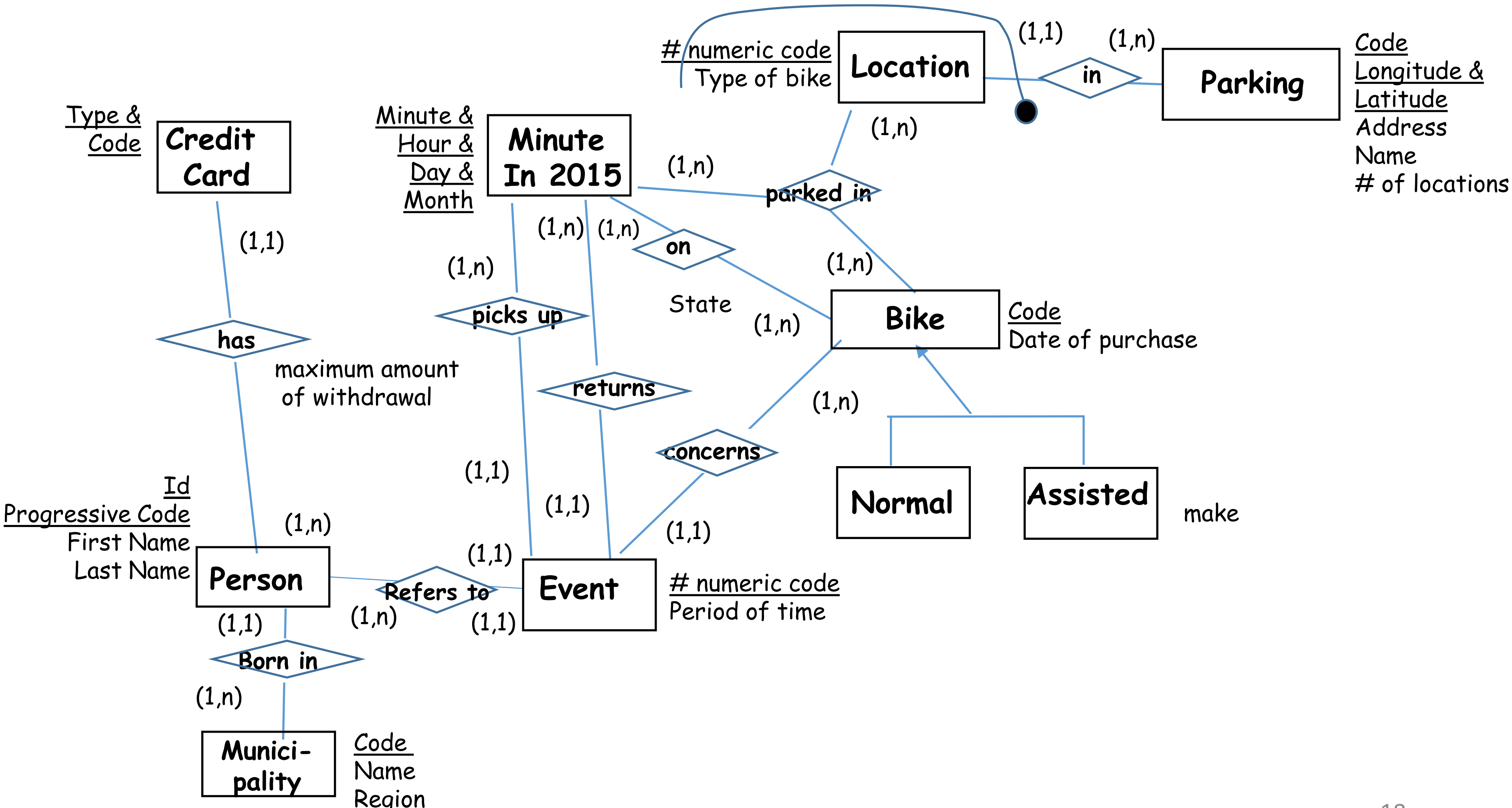
In order to use bikes, users have to register themselves. A registered user is characterized by an identifier (e.g. a fiscal code, or a social security number), a progressive numeric code (first person registered, second person registered, etc.) a name, a surname, date of birth, municipality of birth, with code, name of municipality and region (we assume that municipalities are located in regions).

Everytime a user picks up and subsequently returns a bike, such an event (pick up and return of a bike by a user) has to be recorded. The event is identified by a progressive number in the year (e.g. first event in 2015, second event in 2015, etc.) and duration (e.g. 12 minutes, 47 minutes); such duration has to be explicitely represented in the data base. The minute of pick up and the minute of return have to be associated to the event, besides, of course, the bike used and the user involved.

One or more credit cards are associated to users, with type (e.g. Visa, Diners), progressive code in the card type, and daily maximum amount of withdrawal (that depends on the card and on the person).

You have to represent above requirements with an Entity Relationship Schema, in terms of entities, relationships, attributes, internal and external idetitifers, generalizations, is-a relationships, minimum and maximum cardinalities.

One possible correct solution



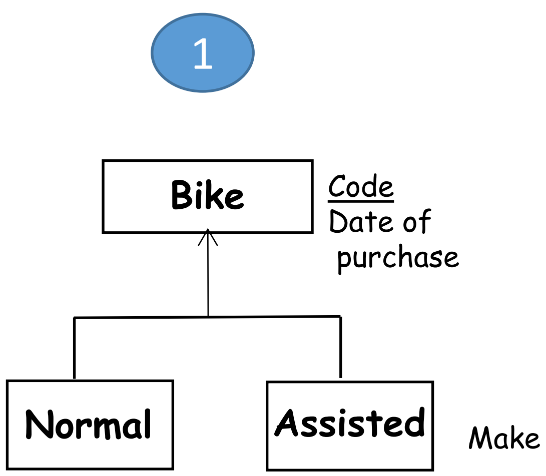
5. A process of incremental design of the correct solution(s), made of several phases.
6. For each phase and related subschema produced, a set of alternative correct solutions, and a set of wrong solutions.

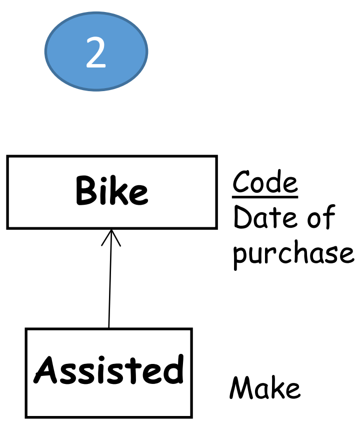
More on Self Assessment

- In the following I provide correct solutions for the different parts of the schema, discuss variants, and provide also wrong solutions (obviously, I cannot provide all possible cases….). Assuming that a perfect solution has grade 30 (as in the italian rules), and the correponding passing grade is 18, a numeric evaluation is assigned to each error, that has to be subtracted from 30.
- The above procedure <u>models errors as independent</u>, while usually errors are logically related. So it is to be seen as an approximate quantitative procedure, with a treshold of approximation equal to, say, +/- 10%.

First of all, what does it mean that two solutions are equivalent?

- Let us introduce the concept of equivalent solutions with a simple example. Consider the following requirements:
- Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes).
- They can be modeled, among others, with the following schemas:





Schemas 1 and 2 are said to be equivalent, since both of them represent correctly requirements above , although with different modeling solutions.

Let us start the design of the ER Schema....

- First of all read carefully two or three times the requirements…….

Text of the assigment

A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking lotand from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type). Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes). When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

In order to use bikes, users have to register themselves. A registered user is characterized by an identifier (e.g. a fiscal code, or a social security number), a progressive numeric code (first person registered, second person registered, etc.) a name, a surname, date of birth, municipality of birth, with code, name of municipality and region (we assume that municipalities are located in regions).

Everytime a user picks up and subsequently returns a bike, such an event (pick up and return of a bike by a user) has to be recorded. The event is identified by a progressive number in the year (e.g. first event in 2015, second event in 2015, etc.) and duration (e.g. 12 minutes, 47 minutes); such duration has to be explicitely represented in the data base. The minute of pick up and the minute of return have to be associated to the event, besides, of course, the bike used and the user involved.

One or more credit cards are associated to users, with type (e.g. Visa, Diners), progressive code in the card type, and daily maximum amount of withdrawal (that depends on the card and on the person).

You have to represent above requirements with an Entity Relationship Schema, in terms of entities, relationships, attributes, internal and external idetitifers, generalizations, is-a relationships, minimum and maximum cardinalities.

Requirements are too complex to be modeled in a single step, with a one-shot strategy. So we have to choose one suitable strategy, or a mix of strategies, to perform the process.

Before chosing a strategy, let us try to segment requirements.

Requirements segmentation

- Let us segment requirements according to the simplest criterium, corresponding to put together sentences that are in the same paragraph, and split sentences in different paragraphs. See next page.

Structure of requirements in four parts

R1 – A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type. Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes). When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

R2 – In order to use bikes, users have to register themselves. A registered user is characterized by an identifier (e.g. a fiscal code, or a social security number), a progressive numeric code (first person registered, second person registered, etc.) a name, a surname, date of birth, municipality of birth, with code, name of municipality and region (we assume that municipalities are located in regions).

R3 – Everytime a user picks up and subsequently returns a bike, such an event (pick up and return of a bike by a user) has to be recorded. The event is identified by a progressive number in the year (e.g. first event in 2015, second event in 2015, etc.) and duration (e.g. 12 minutes, 47 minutes); such duration has to be explicitely represented in the data base. The minute of pick up and the minute of return have to be associated to the event, besides, of course, the bike used and the user involved.

R4 – One or more credit cards are associated to users, with type (e.g. Visa, Diners), progressive code in the card type, and daily maximum amount of withdrawal (that depends on the card and on the person).

You have to represent above requirements with an Entity Relationship Schema, in terms of entities, relationships, attributes, internal and external idetitifers, generalizations, is-a relationships, minimum and maximum cardinalities.

Design strategies compared
see C. Batini – Course on Database Modeling and Design – Part 4: Conceptual Design

Strategies / Pros and cons	Pros	Cons
Bottom up	Intuitive and «easy»	Forces restructurings
Oil stain	Compromise between bottom up and top down	Some restructuring
Top down	Need for an holistic view – difficult to achieve	No restructuring

Strategy chosen to produce the solution

- Since only 45 minutes were given to students, we accordingly decide to adopt a bottom up strategy (see C. Batini – Course on Database Modeling and Design – Part 4: Conceptual Design), producing separately several parts of the final schema, and subsequently connecting such parts in a whole.
- As a consequence, we focus in sequence on requirements R1 to R4. When requirements are two complex to be modeled one shot, we split them recursively into simpler parts.

Text of the assigment

R1 – A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type. Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes). When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

R2 – In order to use bikes, users have to register themselves. A registered user is characterized by an identifier (e.g. a fiscal code, or a social security number), a progressive numeric code (first person registered, second person registered, etc.) a name, a surname, date of birth, municipality of birth, with code, name of municipality and region (we assume that municipalities are located in regions).

R3 – Everytime a user picks up and subsequently returns a bike, such an event (pick up and return of a bike by a user) has to be recorded. The event is identified by a progressive number in the year (e.g. first event in 2015, second event in 2015, etc.) and duration (e.g. 12 minutes, 47 minutes); such duration has to be explicitely represented in the data base. The minute of pick up and the minute of return have to be associated to the event, besides, of course, the bike used and the user involved.

R4 – One or more credit cards are associated to users, with type (e.g. Visa, Diners), progressive code in the card type, and daily maximum amount of withdrawal (that depends on the card and on the person).

You have to represent above requirements with an Entity Relationship Schema, in terms of entities, relationships, attributes, internal and external idetitifers, generalizations, is-a relationships, minimum and maximum cardinalities.

Requirements R1

Requirements R1

R1 – A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type. Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes). When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

Requirements R1 are probabily the most complex to be modeled. We split them into two parts →

Requirements R1

R1.1 – A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type. Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes).

R1.2 When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

Let us consider separaterly R1.1. and R1.2.

Requirements R1.1

Requirements R1

R1.1 – A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type. Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes).

Let us split again requirements R1.1. in two distinct parts.

Requirements R1.1

R1.1.1 – A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type.

R1.1.2 Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes).

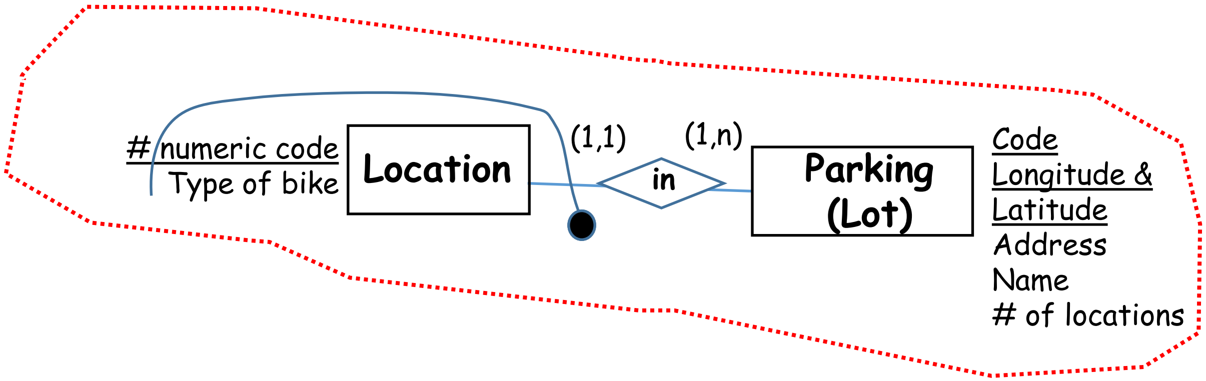
Requirements R1.1.1

Requirements R1.1.1

R1.1.1 – A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type.

Try to produce a reasonable schema.

Schema related to R1.1.1

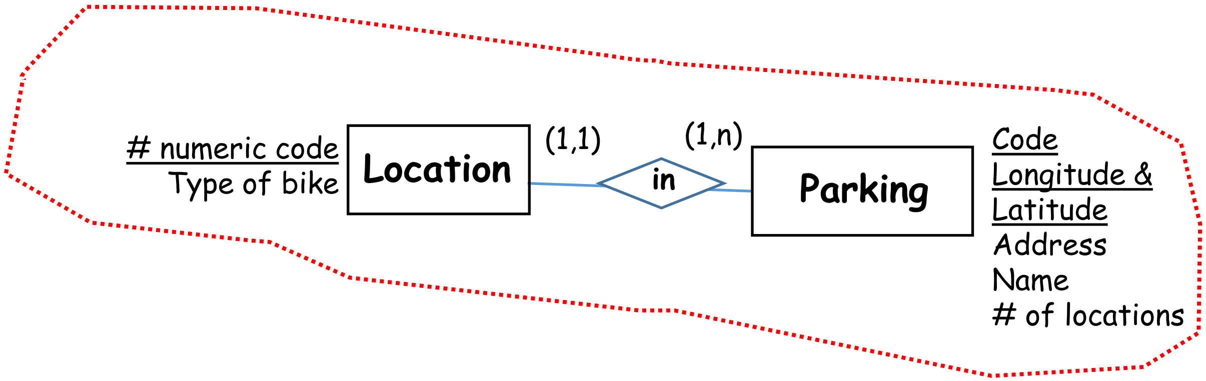


Comments

1. Notice the external identifier associated to Location (<Each location is identified by a progressive number in the parking lot…..>).
2. The pair <Longitude + Latitude> can be considered, besides Code, a second internal identifier of Entity Parking.
3. The attribute # of locations is redundant. This redundancy should not be removed during conceptual design, we have simply to point out it.

Wrong solutions for requirements R1.1.1

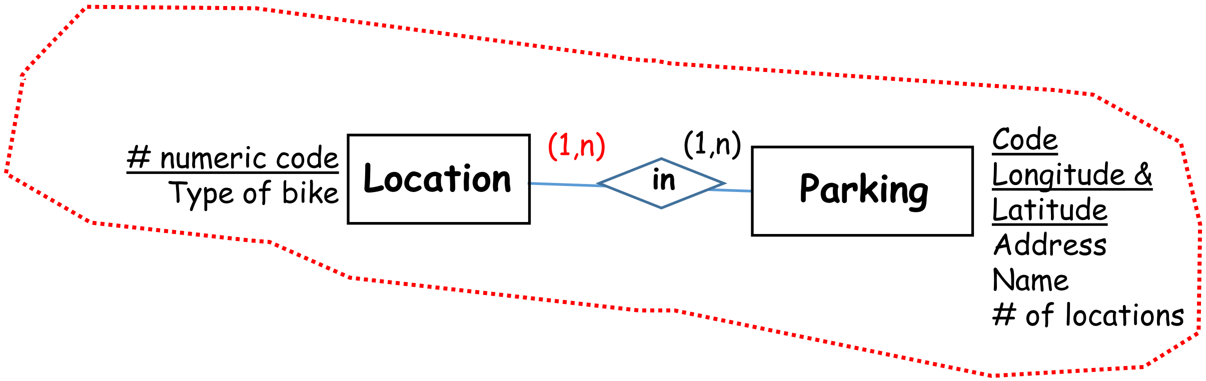
Wrong solution (subtract 1.5)



Comments

1. The identifier of Location is wrong (while cardinalities are correct) since numeric codes are the same for the different parking lots.

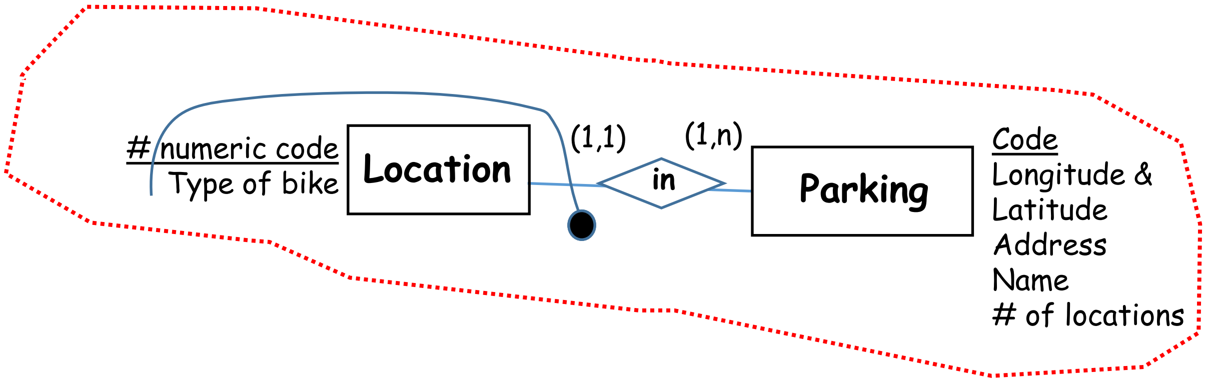
Wrong solution (subtract 2)



Comments

1. Here the maximum cardinality of Location is wrong too, so the error is more serious.

Wrong solution (subtract 0,2)



Comments

1. The second identifier of Parking (Latitude + Longitude) is not represented. As we will see in the statistics of errors made by students, the great majority of students made this mistake, probably for the short time available for checking and fine tuning, so the error in this case is negligible.

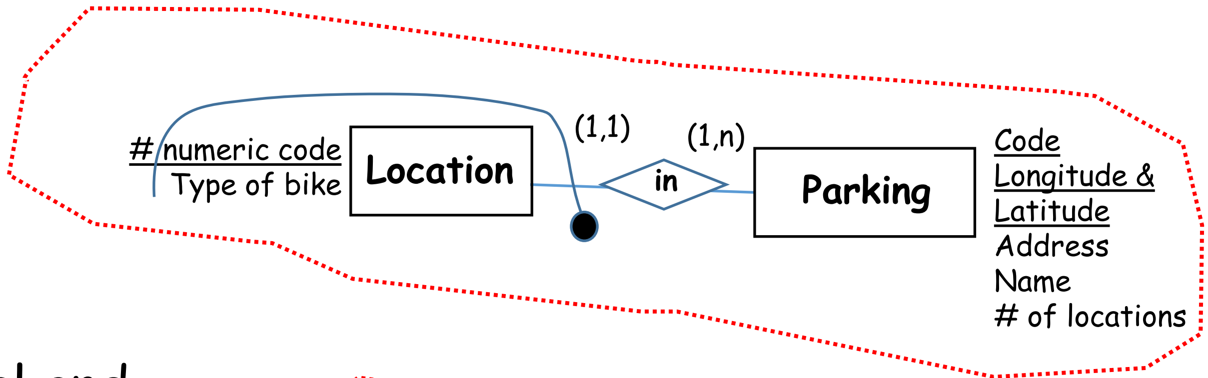
Requirements R1.1.2

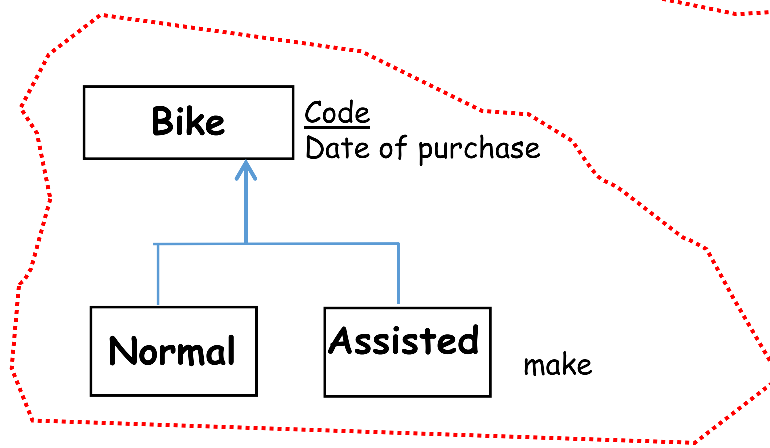
Requirements R1.1.2

R1.1.1 – A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type.

R1.1.2 Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes).

Schema related to R1.1.1 + R1.1.2

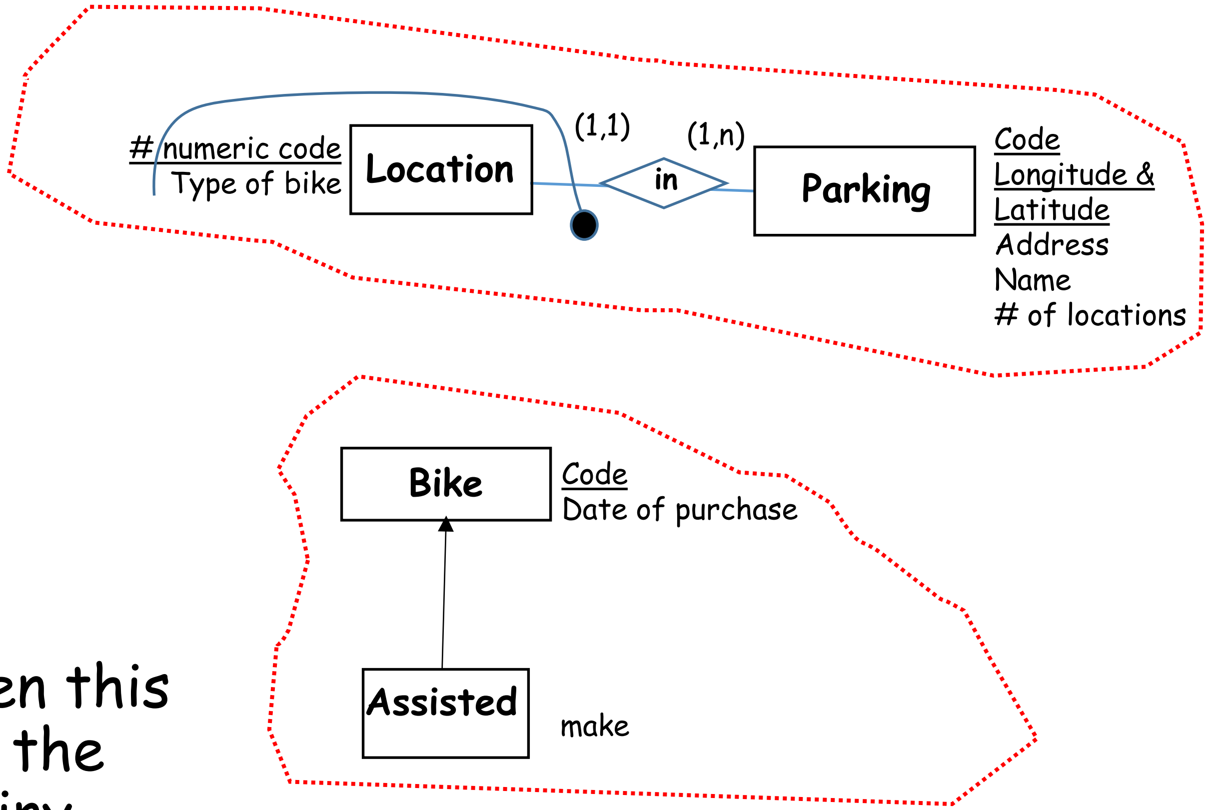




Comments

1. We have modeled normal and assisted bikes as a generalization, since assisted bikes are characterized by the make attribute.
2. Another equivalent choice corresponds to model only bikes and assisted bikes, without representing normal bikes, see next page (see an in depth analysis in the slides starting with «Discussion on entity Bike»)

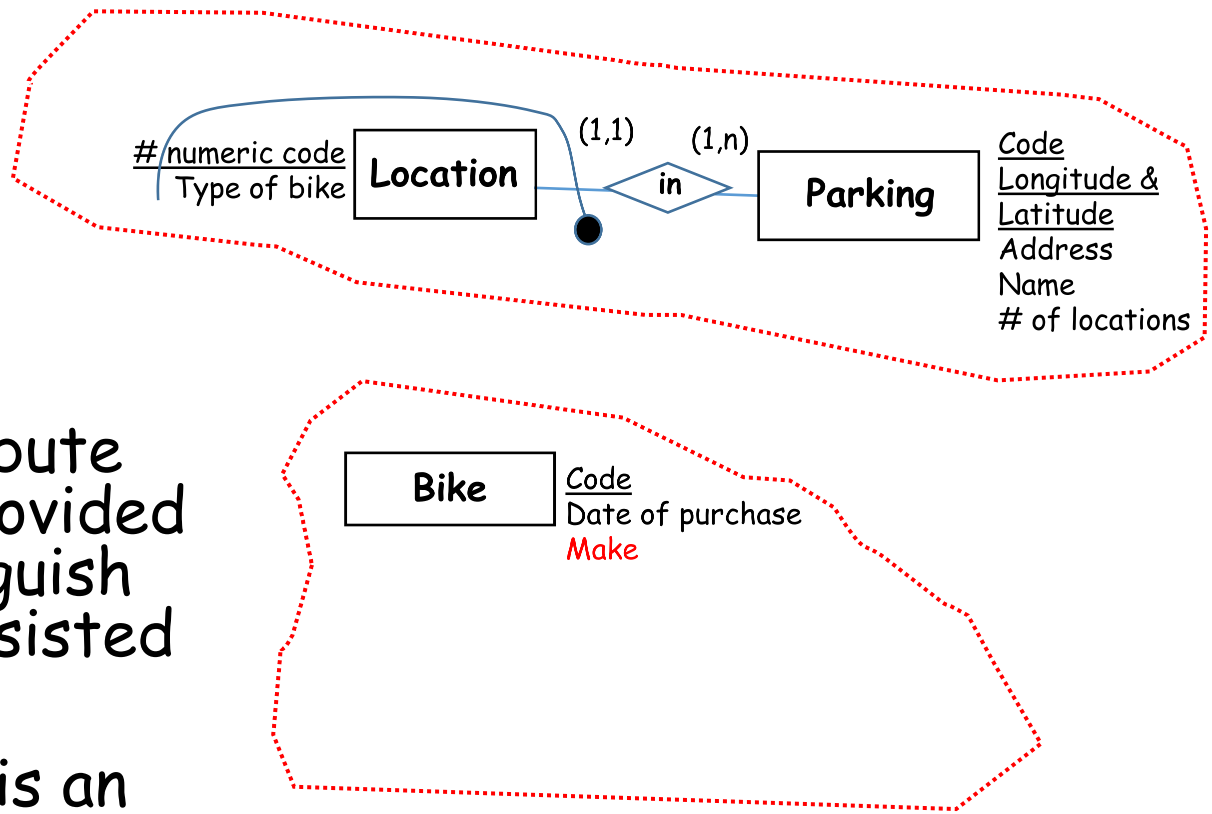
Bike and Assisted bike as an is-a relationship



Comment

- The difference between this solution and the one in the previous page is very tiny, both can be considered as correct and readable solutions.

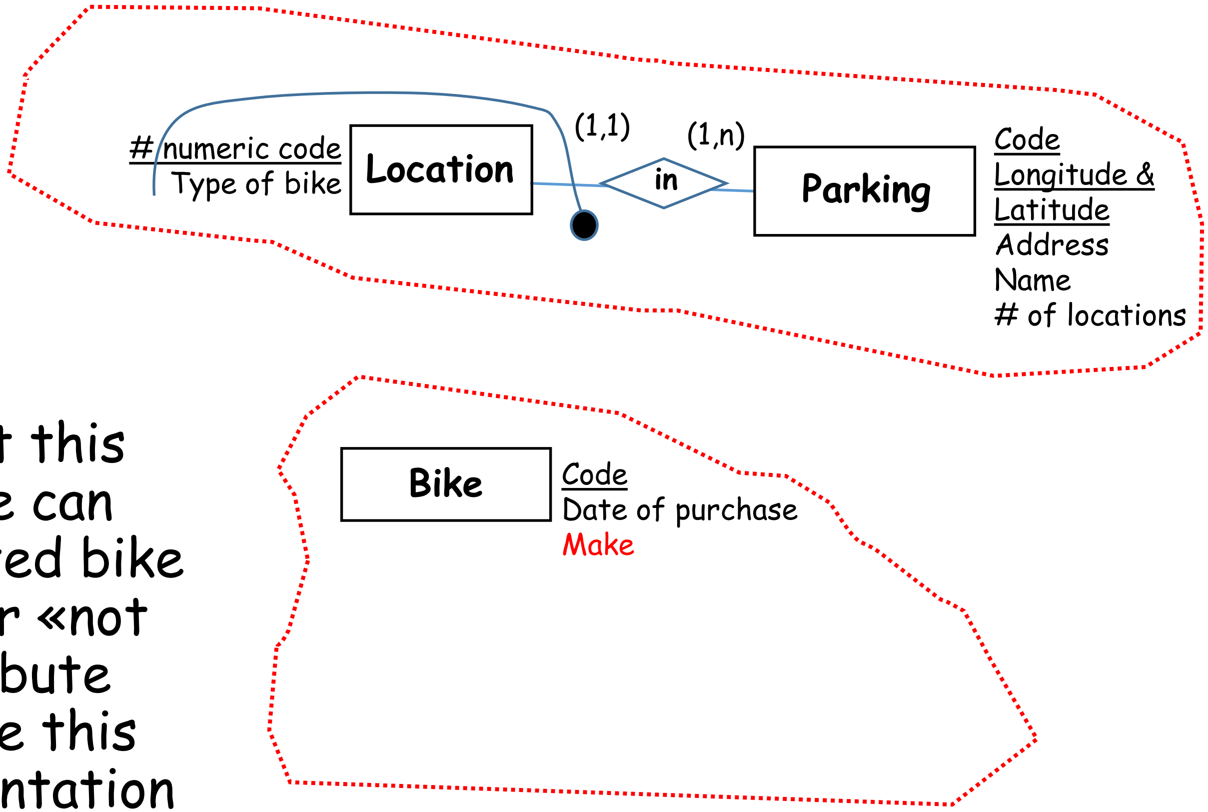
Wrong solution (subtract 1)



Comment

- In this case no attribute such as «Type» is provided that allows to distinguish among normal and assisted bikes.
- Furthermore, there is an avoidable redundancy in the schema, since a value of Make = null is assigned to normal bikes.

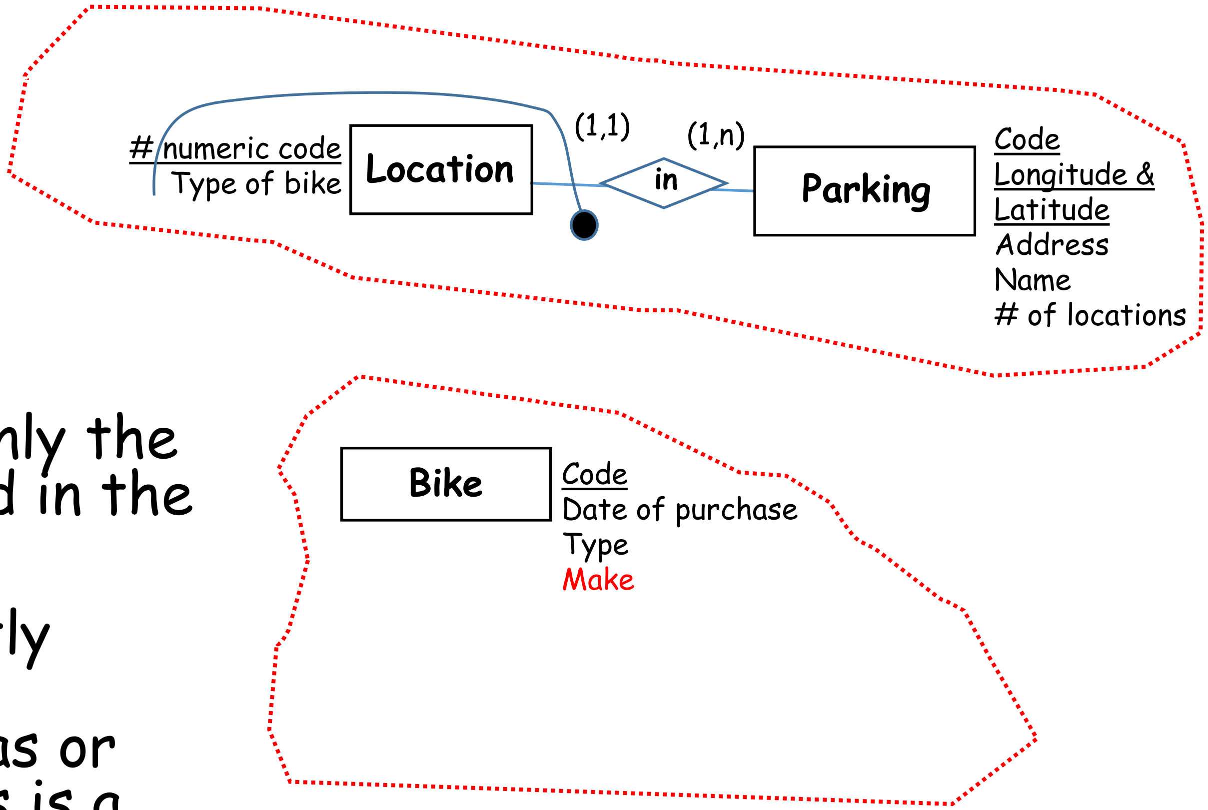
Position of some students



Comment

Some students argued that this schema is correct, since we can identify Normal and Assisted bike according to being «null» or «not null» the value of the attribute Make. I do not agree, since this choice is a sort of implementation trick, and the type of bike is hidden and not explicit as in the attribute Type.

Wrong solution (subtract 0,6)



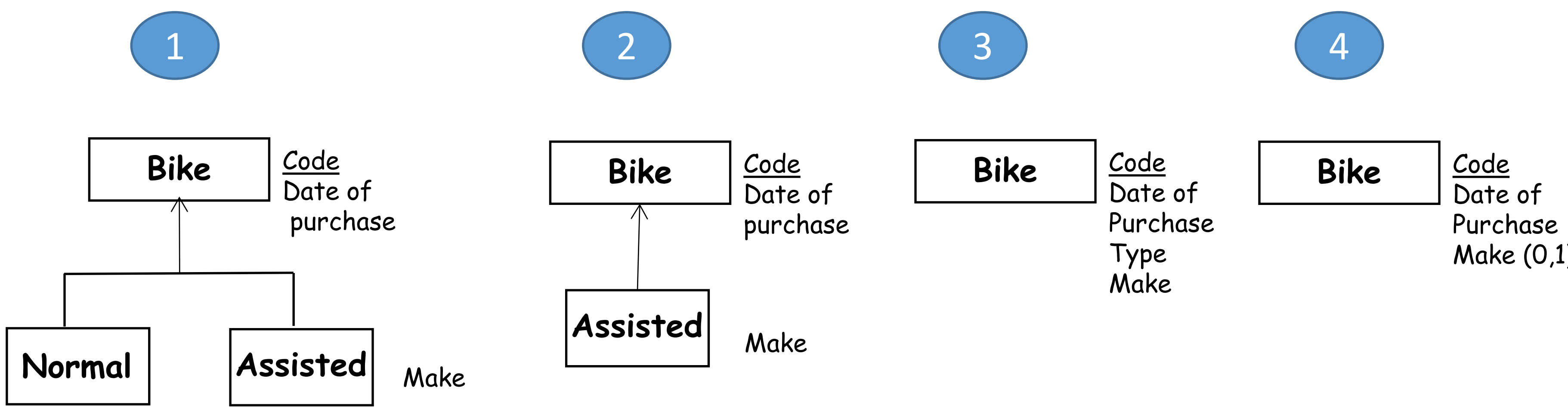
Comment

- In this case there is only the redundancy highlighted in the previous slide.
- The ER model inherently allows to avoid such redundancies using is-as or generalizations, so this is a case of inaccuracy with respect to the model. See also the discussion in the following slides.

Discussion on entity Bike

The entity Bike and related schemas

Requirements on Bikes say: each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes). So we can represent Bikes as in the following four cases. Schemas 1 and 2 are «almost» equivalent, in schema 1 we have an entity without properties (Normal), but its name is informative, and so we can consider equally readable and informative as schema 2. Schema 3 is equivalent to Schemas 1 and 2, provided than we explicitely represent the type, that has as domain [Normal, Assisted]. As a consequence the fact that bikes are of two types is not directly understandable from the schema. Furthermore in Schema 3 we do not express the fact that make is significant only for Assisted bikes; this rule is partially expressed in Schema 4, where it is expressed the property that the value of Make is optional.



Requirements R1.2

Requirements R1.2 – first part

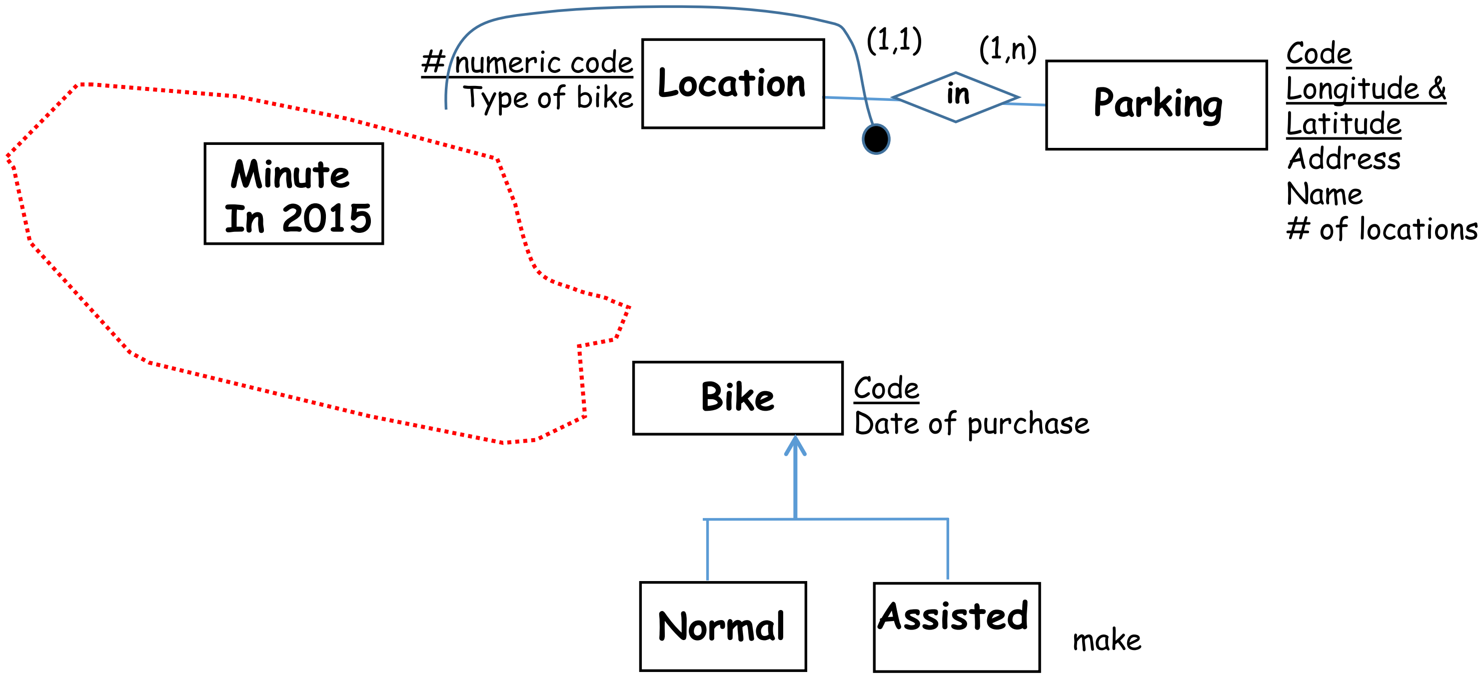
R1.2 When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

This part of requirements is quite complex to be modeled. Anyhow, if we read carefully requirements two or three times, there are some clues in the text that help in modeling the conceptual schema, they are underlined in the text. More specifically let us consider the sentence

It is strongly recommended to represent the concept of minute with an Entity

Let us apply the recommendation…. →

Schema related to R1.1 + R1.2

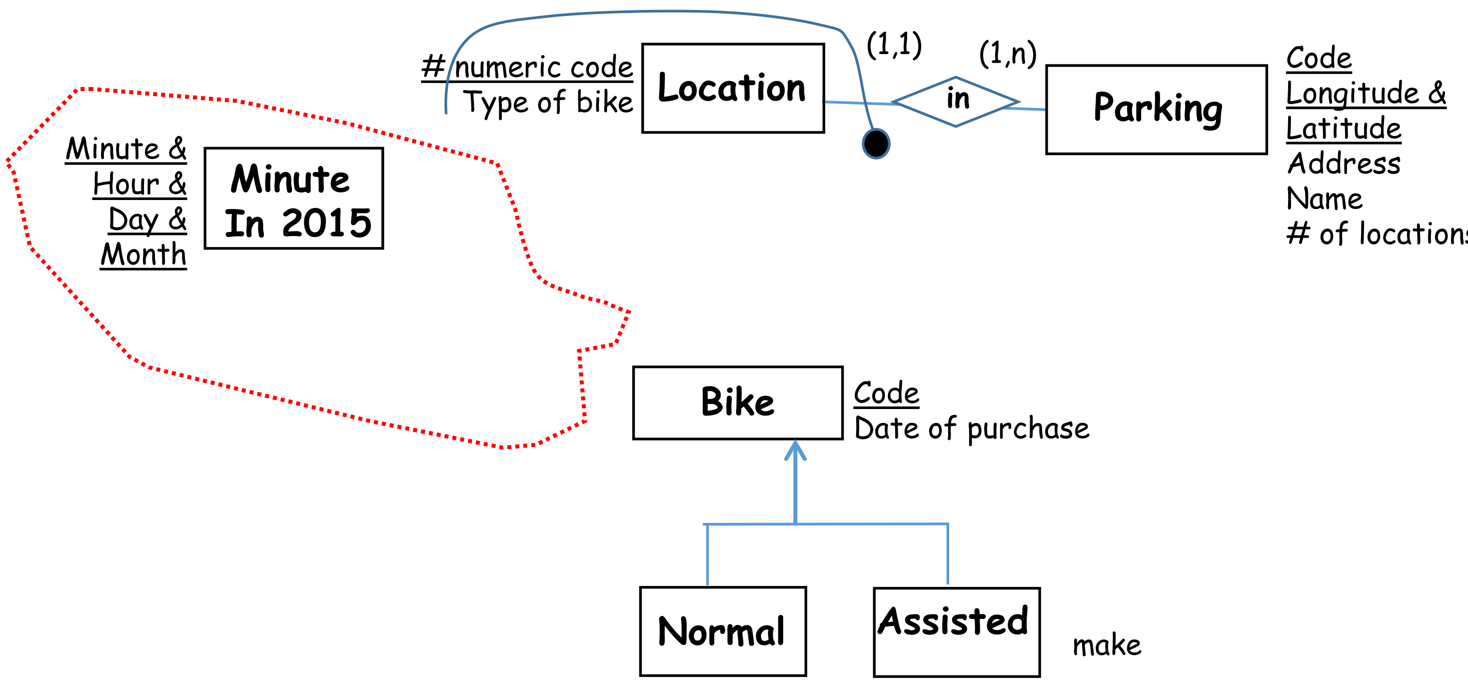


Requirements R1.2 – first part

R1.2 When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

When representing minute as an entity, we have to focus first on the attributes that make up the identifier: to identify a minute in one year, which are the properties that allow to distinguish the minute from all other minutes in the year? Propose a solution and then look at the next page.

Schema related to R1.1 + R1.2 first part

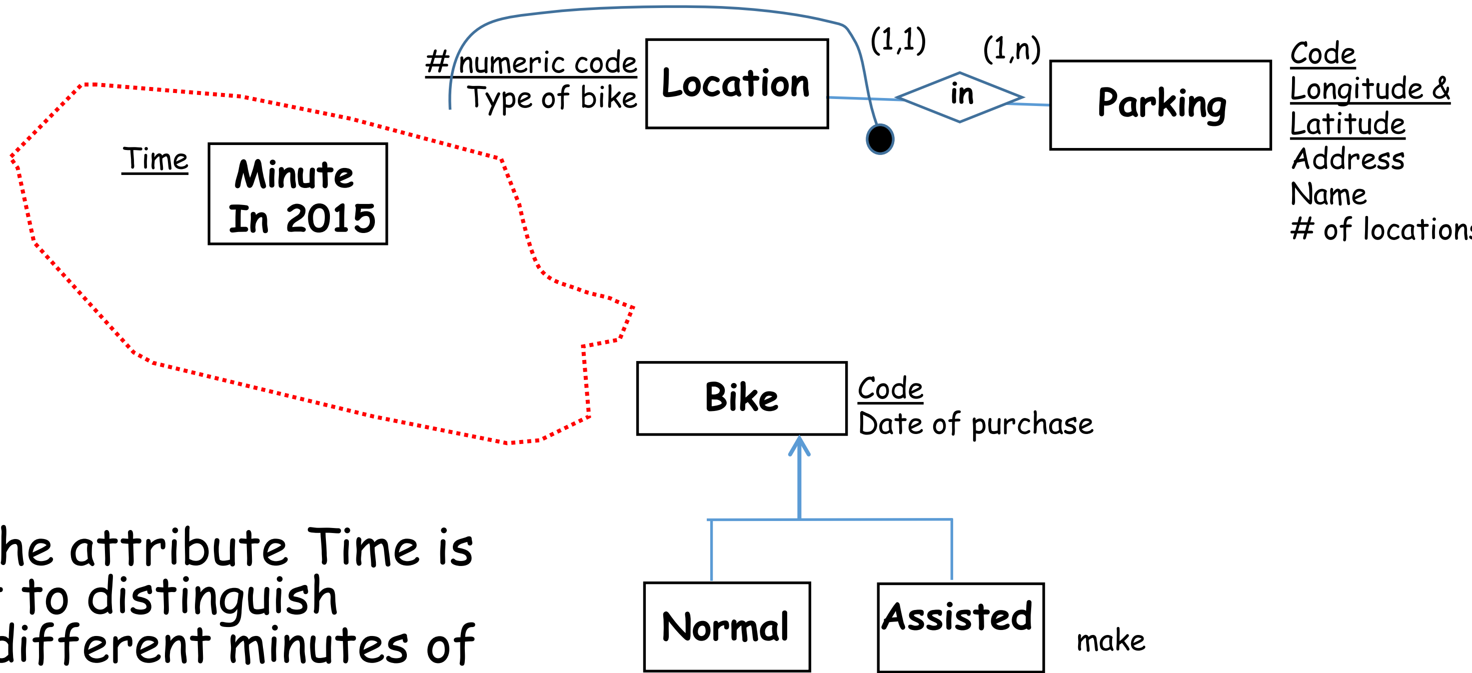


Let us focus on other possible solutions for the Entity Minute

R1.2 When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

Some students have wrongly modeled the name and attributes in the identifier of the Entity Minute as in the next two slides.

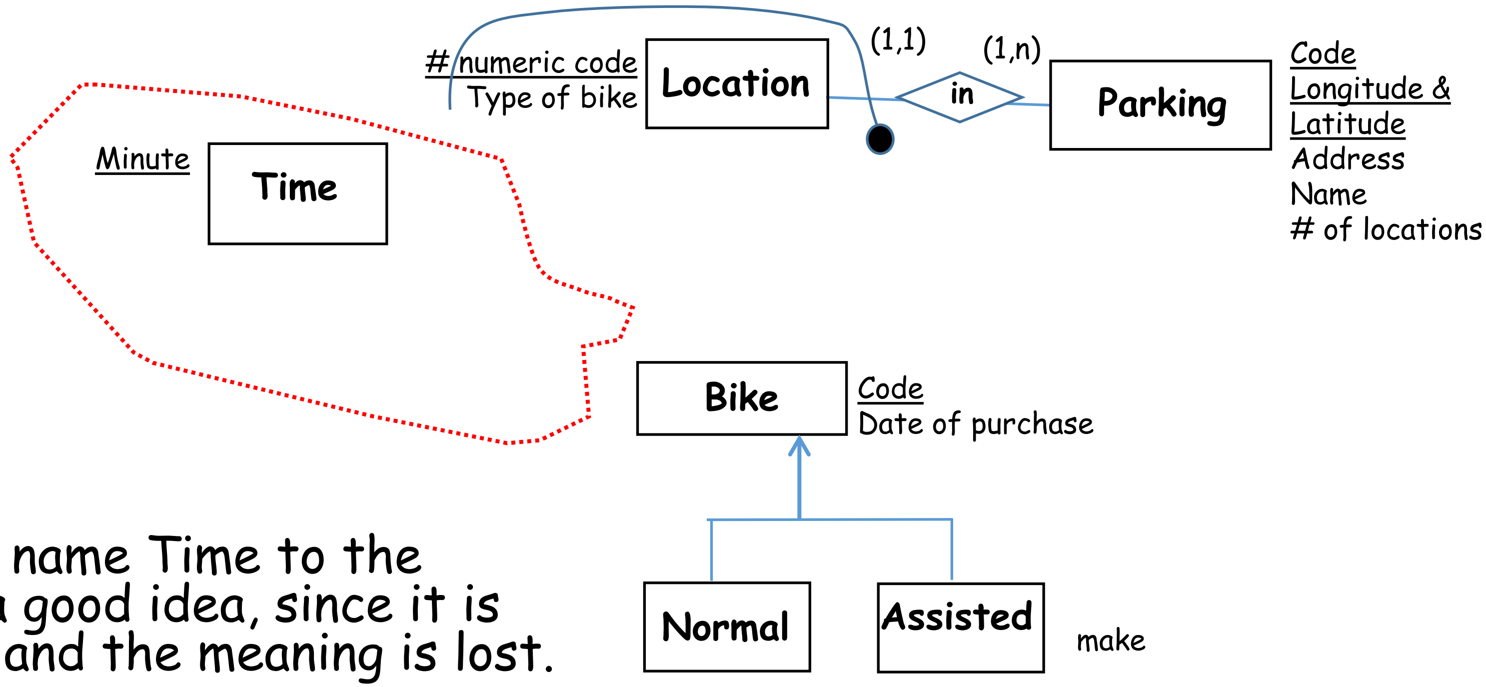
Lacking quality: correctness with respect to requirements (subtract 0.5)



Comment

- In this case the attribute Time is not sufficient to distinguish between the different minutes of the year, unless it is an aggregate attribute, where the aggregation is in terms of attributes minute, day, etc., but this is not represented in the schema, so correctness with respect to requirements is not achieved.

Lacking quality: correctness with respect to requirements (subtract 0.5)



Comment

- Assigning the name Time to the entity is not a good idea, since it is too abstract, and the meaning is lost.
- As for the identifier Minute, it is wrong, since it is only part of the information we have to provide to identify a minute in the year; so, also in this case, correctness with respect to requirements is not achieved.

In depth analysis
on possible identifiers
for the Entity Minute

Requirements associated to the concept Minute (in black)

- R1.2 When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

Correct solutions provided by students, with number of occurrences and percentage on the total of solutions

Correct Solution		occurrences	percentage
	Minute, Day, Hour, Month	23	23,71%
	Minute, Hour, Date (composed of Day, Month)	16	16,49%
	Minute, Hour, Day, Date	1	1,03%

Wrong solutions, ranked by distance from the correct solution, with number of occurrences and percentage

	Minute, Day, Hour, Month, Year	6	6,19%
	Day (composed of Year, Month, Day), Hour (composed of Hour, Minute)	3	3,09%
	Date (composed of Year, Month, Day), Hour (composed of Hour, Minute, Second)	1	1,03%
	Minute, Hour, Day	2	2,06%
	Minute, Hour, Date	2	2,06%
	Day, Hour, Month	1	1,03%
	Day, Month, Year	1	1,03%
	Hour, Date, Year	1	1,03%
	Minute, hour, Month	1	1,03%
	Minute, Hour, Year	1	1,03%
	Minute, Hour	3	3,09%
	Hour, Date	9	9,28%
	Day, Month	2	2,06%
	Day, Year	1	1,03%
	Hour, Year	1	1,03%
	Value, Day, Month	1	1,03%
	Id Minute, Minute	1	1,03%
	Time, Date	3	3,09%
	Second, Number	1	1,03%
	Minute	1	1,03%
	Hour	1	1,03%
	Year	2	2,06%
	Value	1	1,03%
	Number	2	2,06%
	Nothing	9	9,28%
	Total	97	

Comment on solutions for the identifier of Minute

- The great diversification of solutions provided by students reveals that examples of «minute» provided in requirements did not avoid wrong, incomplete, imprecise solutions produced by the majority of students.
- From the other side, this is an interesting case of requirements where the autonomous «creativity» of students can be tested and the errors committed can be quantified as distance from correct solution(s).

Requirements R1.2 – second part

R1.2 When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

Now, before modeling requirements, let us:

1. Split remaining requirements in R1.2 into two different paragraphs and
2. Highlight the concepts in requirements that have been modeled so far as entities.

Requirements R1.2 – second part

R1.2.1 - When bikes are not used, they are parked in locations of parking lots. As a consequence, for each Bike and for each Minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the Bike, namely if it is parked or it is in motion.

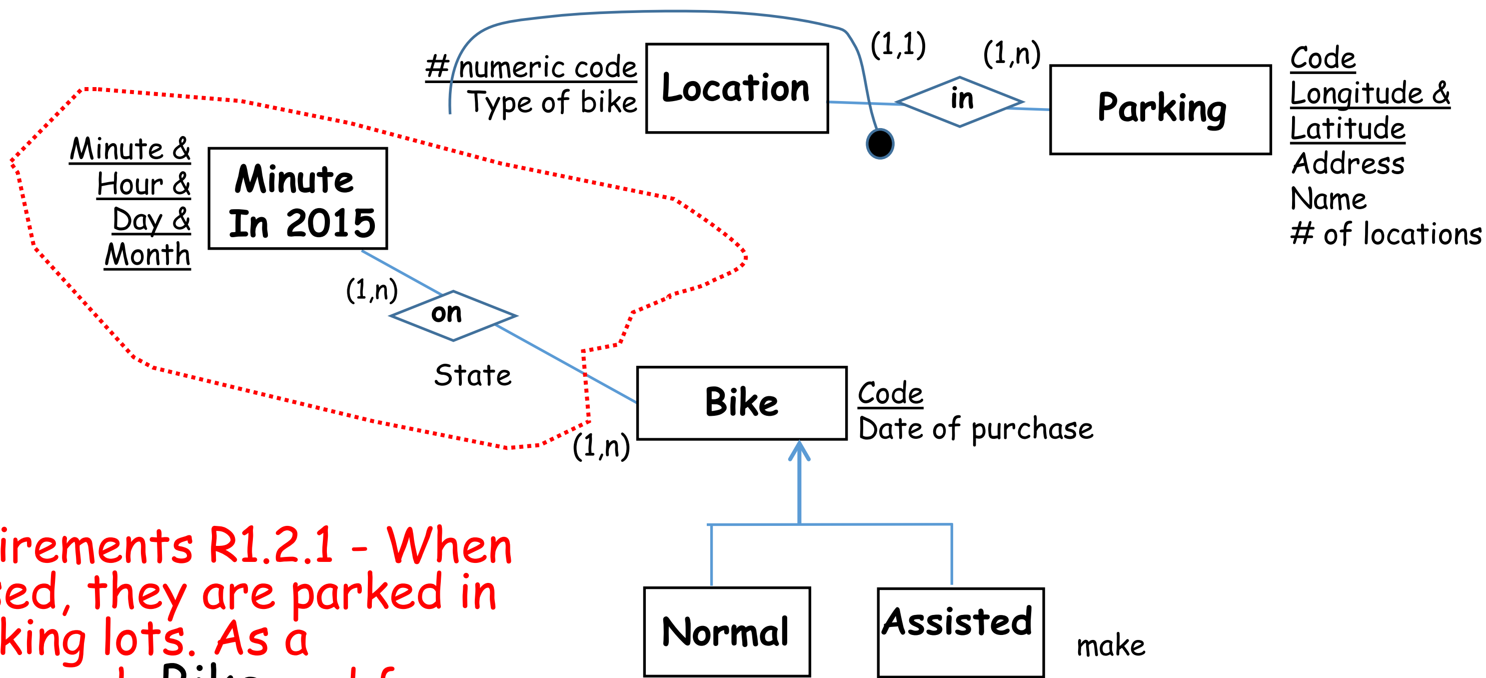
R1.2.2 - Furthermore, for each Minute in the year and each Location in a Parking lot, we want to represent, separately, which Bike is possibly parked in the Location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given Minute a Bike is in motion, the corresponding Location in a Parking lot does not exist, and, therefore, will not be represented.

Requirements R1.2.1 and R1.2.2 correspond to two different relationships:

1. Requirements R1.2.1 can be modeled in terms of a **binary** relationship between Bike and Minute.
2. Requirements R1.2.2 can be modeled in terms of a **ternary** relationship among Bike, Minute and Location.

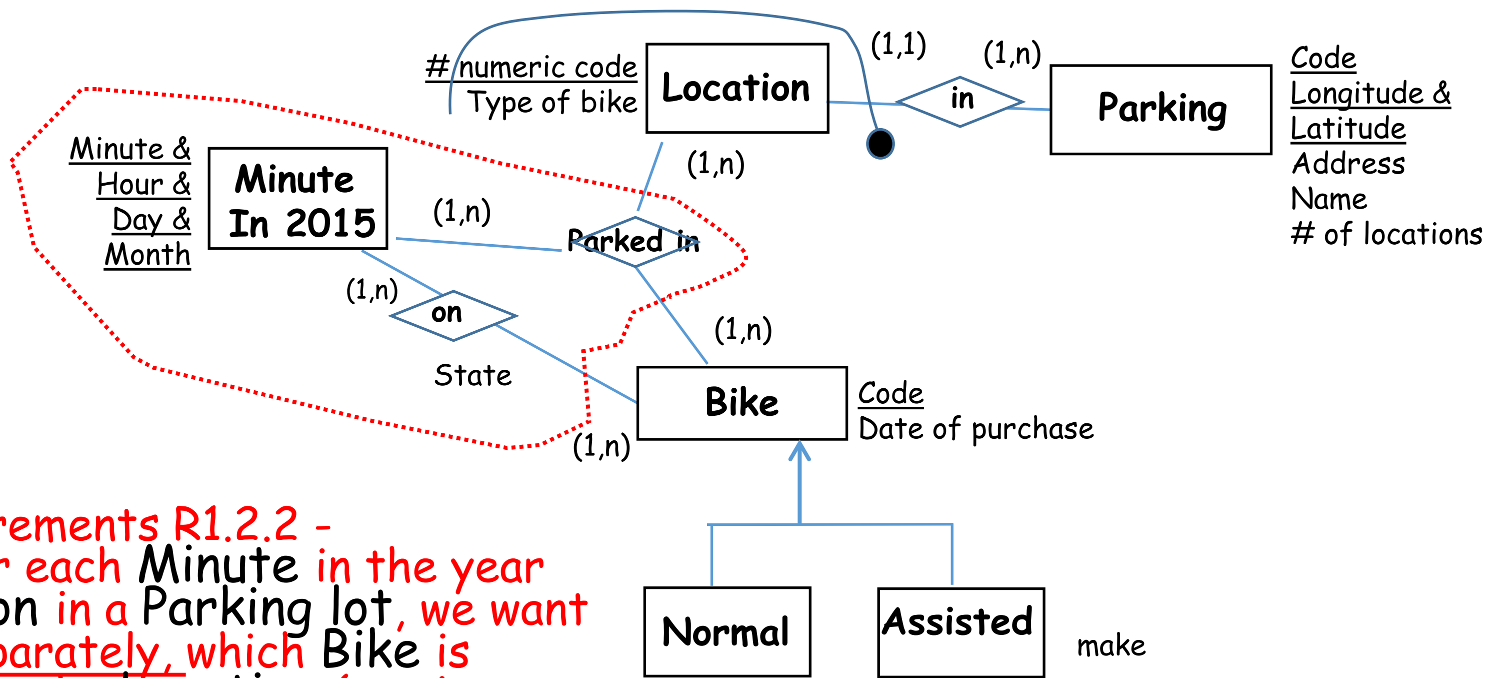
The word separately is an evident suggestion to model requirements as separate relationships in the ER schema.

Schema related to R1.1.1 + R1.1.2 + R1.2.1



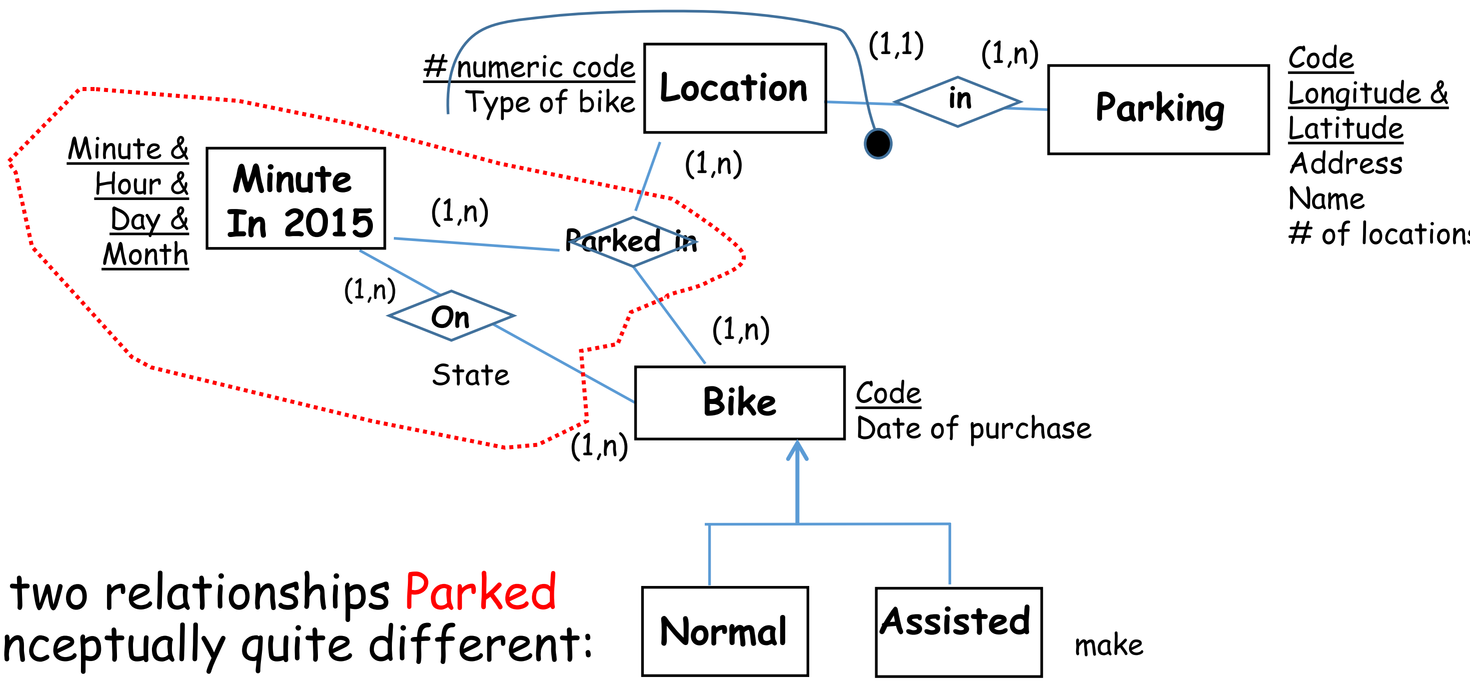
Remember requirements R1.2.1 - When bikes are not used, they are parked in locations of parking lots. As a consequence, for each Bike and for each Minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the Bike, namely if it is parked or it is in motion.

Schema related to R1.1.1 + R1.1.2 + R1.2.1 + R1.2.2



Remember requirements R1.2.2 - Furthermore, for each Minute in the year and each Location in a Parking lot, we want to represent, separately, which Bike is possibly parked in the Location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given Minute a Bike is in motion, the corresponding Location in a Parking lot does not exist, and, therefore, will not be represented.

Schema related to R1.1.1 + R1.1.2 + R1.2.1 + R1.2.2



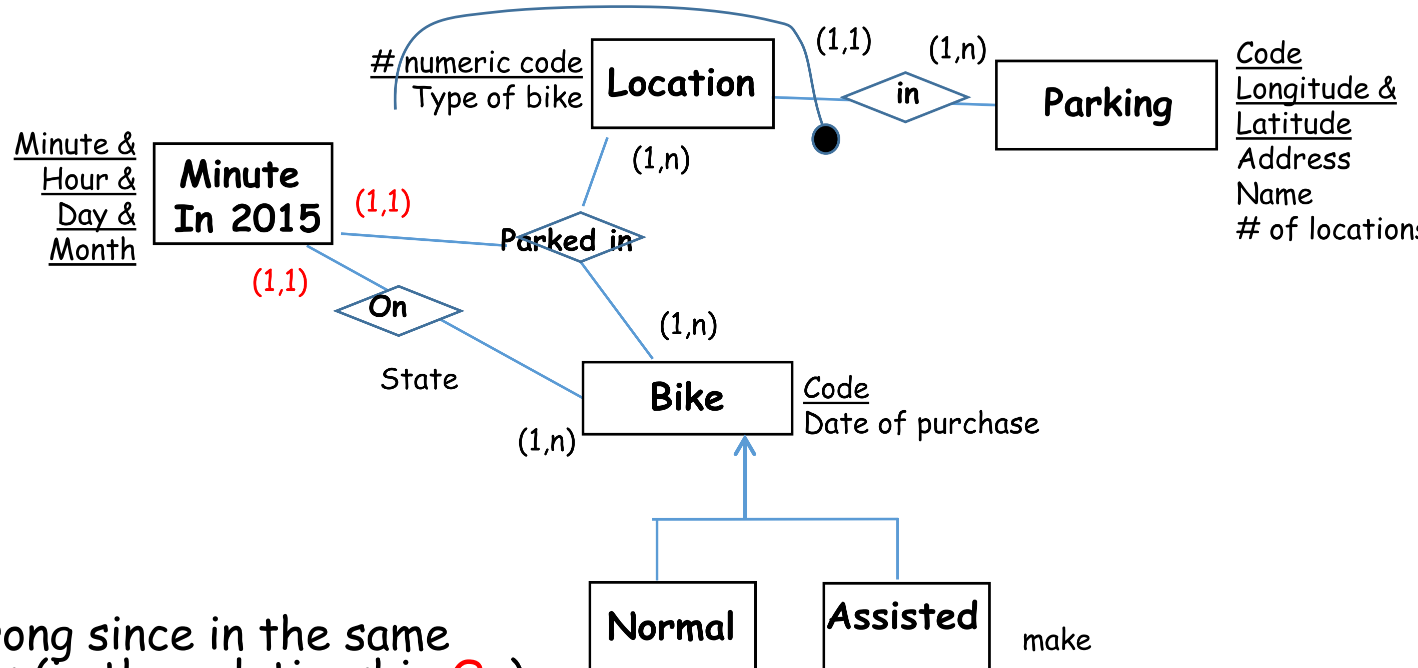
Comment

Notice that the two relationships Parked in and On are conceptually quite different:

- Relationship On represents the state of the bike in each Minute of the year
- Relationship Parked in represent the specific location in which a Bike is parked only for Minutes of the year in which the bike is parked.

Most frequent errors in modeling R1.2

Wrong cardinality of the Entity Minute (subtract 0.7 for each wrong cardinality)

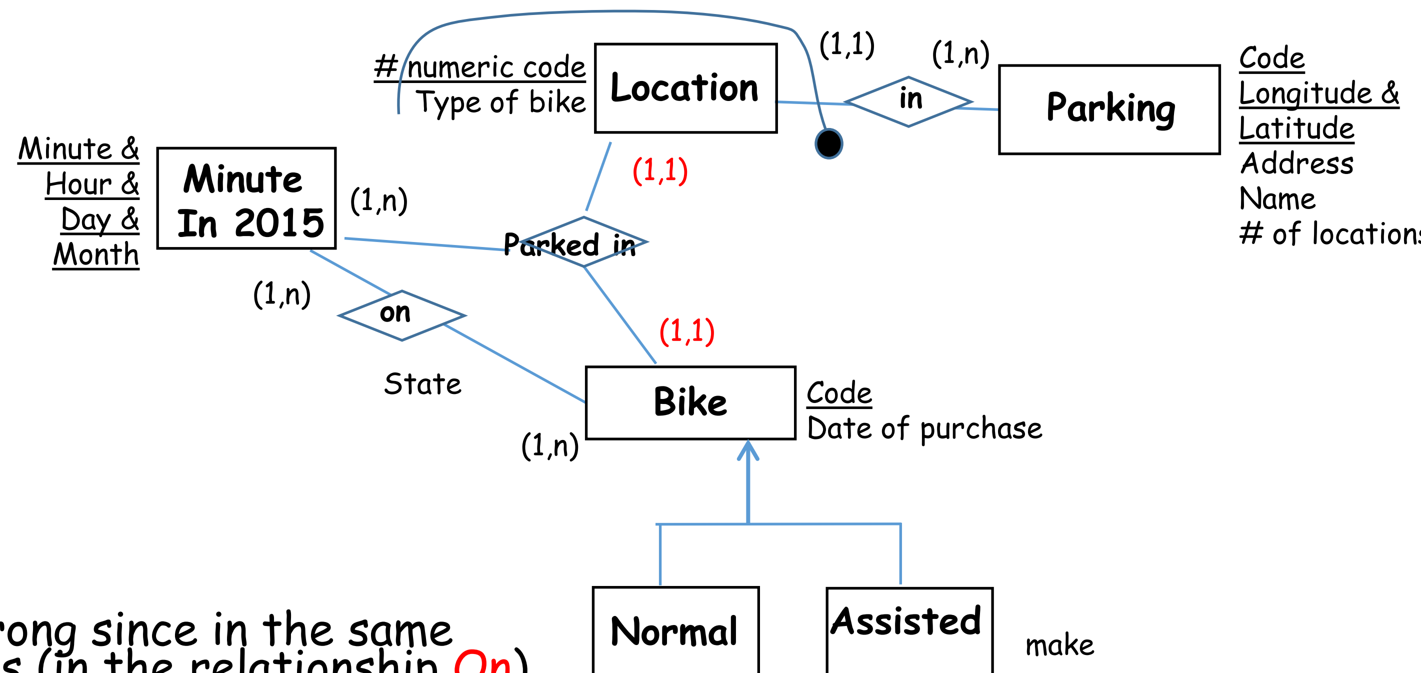


Comment

Cardinalities are wrong since in the same minute several bikes (in the relationship On) or else several bikes in locations can appear in the relationship instances.

When you have to assign cardinalities to an Entity in the schema, and you are in trouble, think to the meaning of the relationship in the real world, and, in case, conceive a generic instance of the relationship.

Wrong cardinalities of Entities Location and Bike (subtract 0.7 for each cardinality)

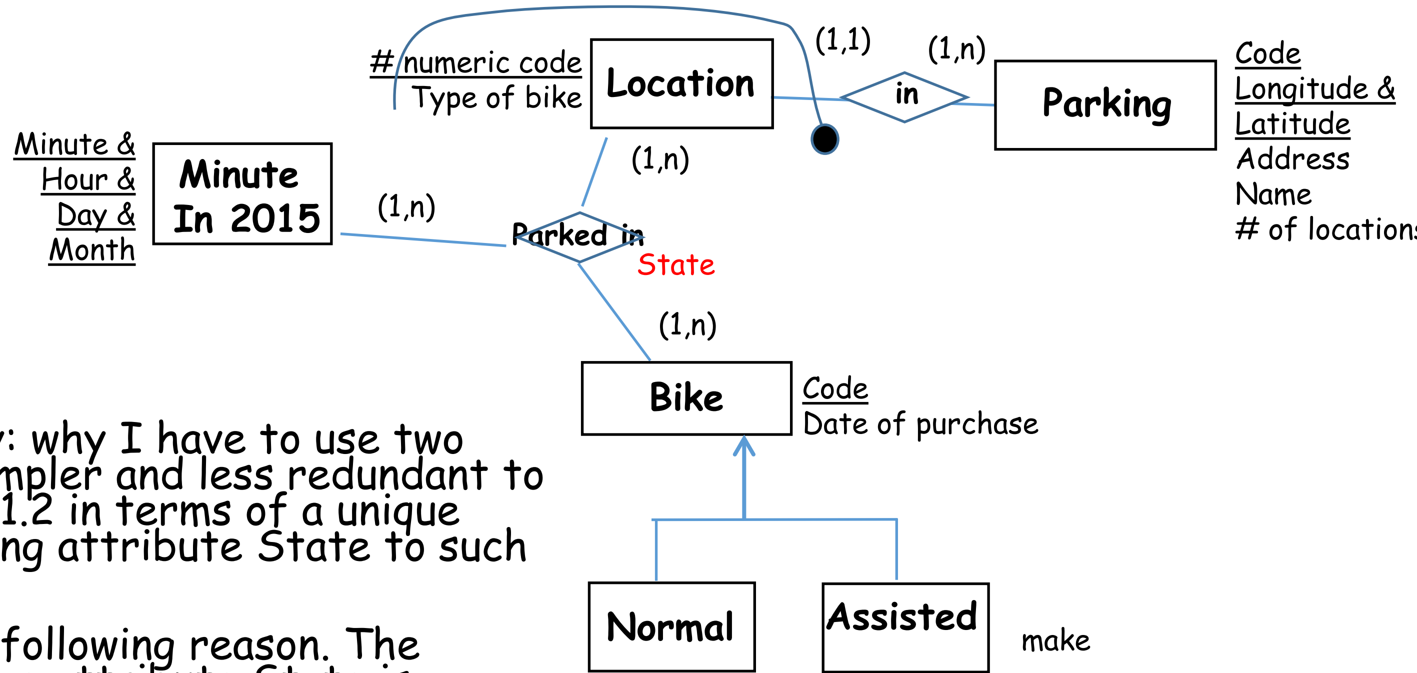


Comment

Cardinalities are wrong since in the same minute several bikes (in the relationship On) or else several bikes in locations (relationship Parked in) can appear in the the relationship instances.

When you have to assign cardinalities to an Entity in the schema, and you are in trouble, think to the meaning of the relationship in the real world, and, in case, conceive a generic instance of the relationship.

Wrong modeling in terms of a unique relationship instead of two relationships (subtract 3)



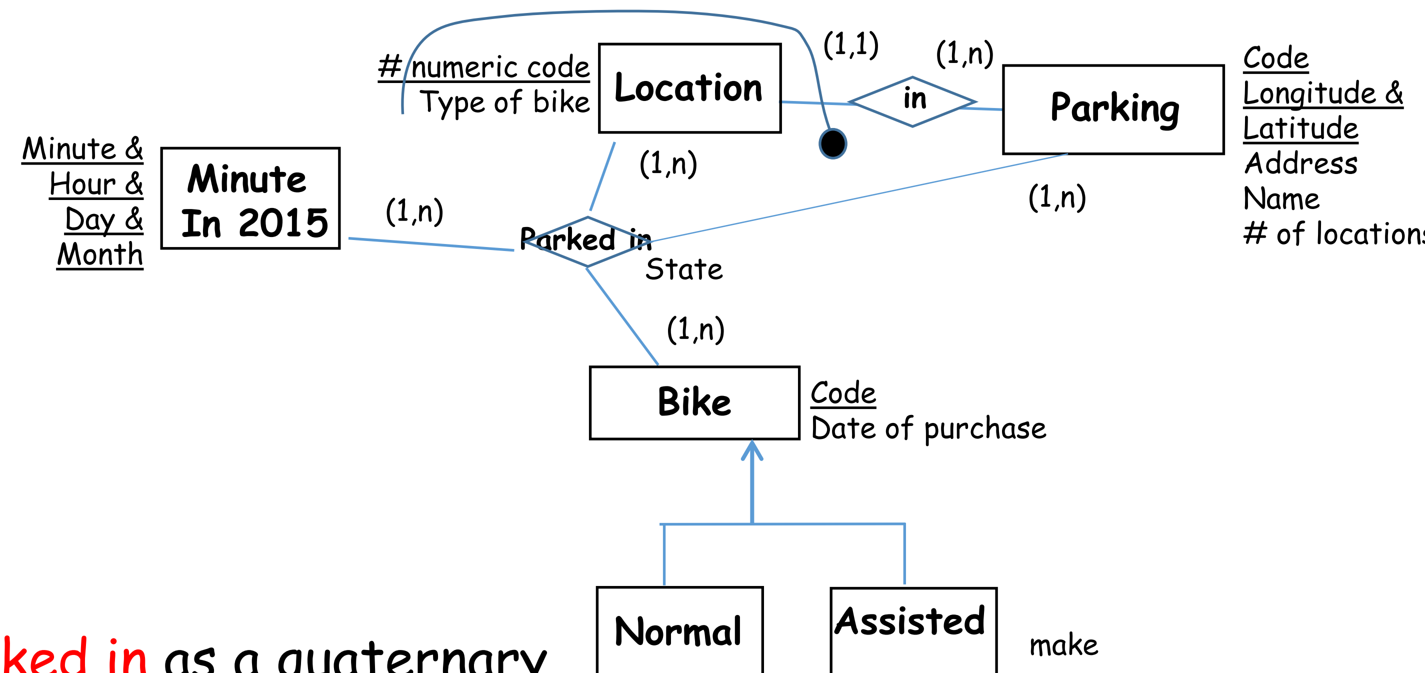
Comment

One is tempted to say: why I have to use two relationships? It is simpler and less redundant to model requirements R1.2 in terms of a unique relationship, associating attribute State to such relationship.

This is wrong for the following reason. The attribute domain of the attribute State is without no doubt [Parked, not Parked], no other meaning can be associated to State.

As a consequence, now the relationship has changed its meaning, and represents all possible triples <minute, location, bike>, in the cartesian product of minutes in 2015, locations in parking, and bikes available, so a very huge number…! Such relationships makes no sense!

Wrong modeling of Parking as an Entity involved in the relationship Parked in (subtract 2)



Comment

Representing Parked in as a quaternary relationship is wrong, since every location is located in a specific parking lot, and so including Parking in the entities involved in Parking in is useless, redundant, in simple words is wrong.

Requirements R2

Text of the assigment

R1 – A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type. Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes). When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

R2 – In order to use bikes, users have to register themselves. A registered user is characterized by an identifier (e.g. a fiscal code, or a social security number), a progressive numeric code (first person registered, second person registered, etc.) a name, a surname, date of birth, municipality of birth, with code, name of municipality and region (we assume that municipalities are located in regions).

R3 – Everytime a user picks up and subsequently returns a bike, such an event (pick up and return of a bike by a user) has to be recorded. The event is identified by a progressive number in the year (e.g. first event in 2015, second event in 2015, etc.) and duration (e.g. 12 minutes, 47 minutes); such duration has to be explicitely represented in the data base. The minute of pick up and the minute of return have to be associated to the event, besides, of course, the bike used and the user involved.

R4 – One or more credit cards are associated to users, with type (e.g. Visa, Diners), progressive code in the card type, and daily maximum amount of withdrawal (that depends on the card and on the person).

You have to represent above requirements with an Entity Relationship Schema, in terms of entities, relationships, attributes, internal and external idetitifers, generalizations, is-a relationships, minimum and maximum cardinalities.

Requirements R2

R2 – In order to use bikes, users have to register themselves. A registered user is characterized by an identifier (e.g. a fiscal code, or a social security number), a progressive numeric code (first person registered, second person registered, etc.) a name, a surname, date of birth, municipality of birth, with code, name of municipality and region (we assume that municipalities are located in regions).

This part of requirements poses less critical issues than modeling requirements R1.

Let us first fix the entities that are present in requirements.

Entities in Requirements R2

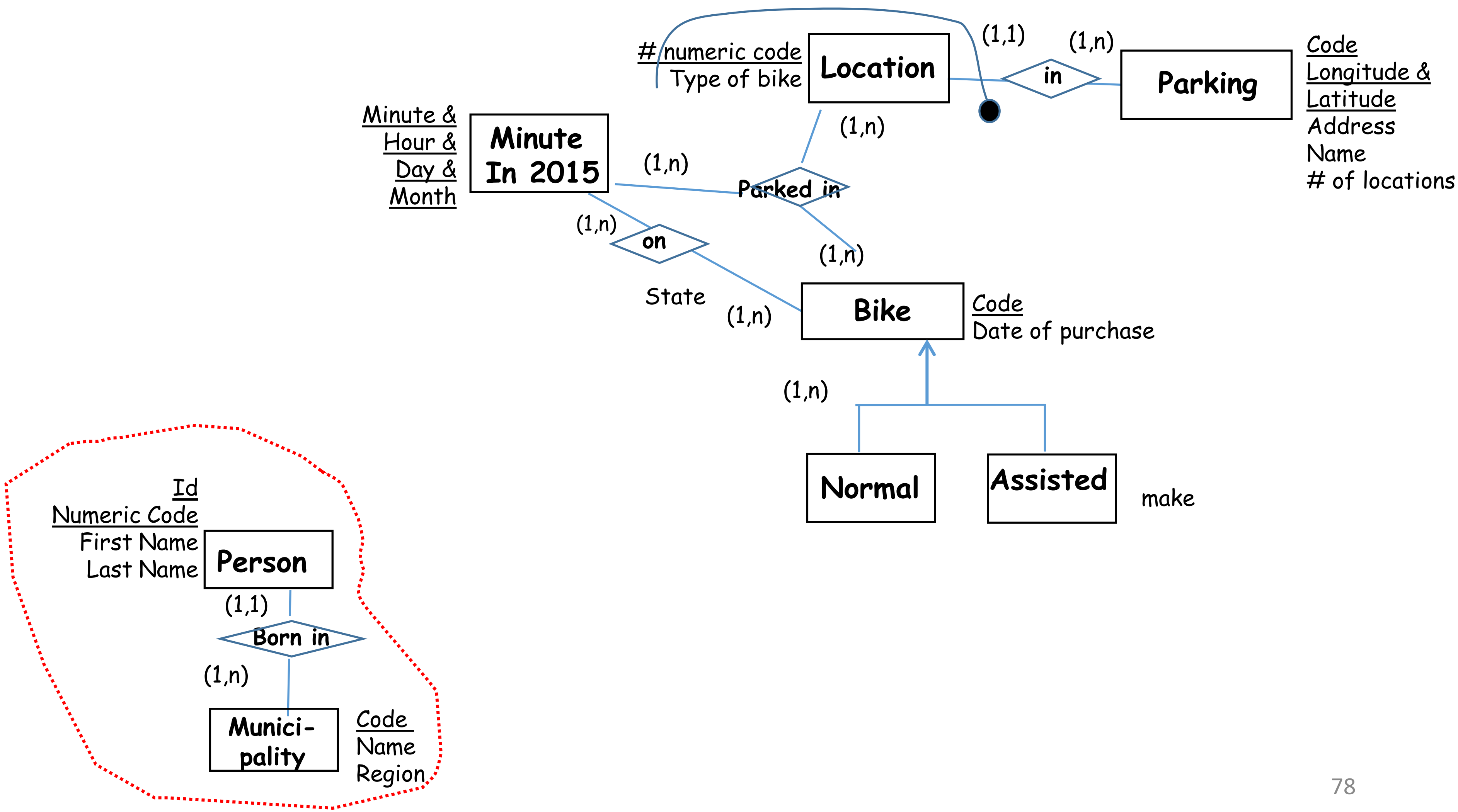
R2 – In order to use bikes, users have to register themselves. A registered user is characterized by an identifier (e.g. a fiscal code, or a social security number), a progressive numeric code (first person registered, second person registered, etc.) a name, a surname, date of birth, municipality of birth, with code, name of municipality and region (we assume that municipalities are located in regions).

Let us first fix the entities that are present in requirements:

1. Entity User has several attributes, and two identifiers, respectively a. e.g. Fiscal code (we choose the italian identifier), and b. Progressive numeric code.
2. Entity Municipality has as identifier the Code.

Furthermore, User and Municipality are related by a relationship Born in.

One possible correct solution

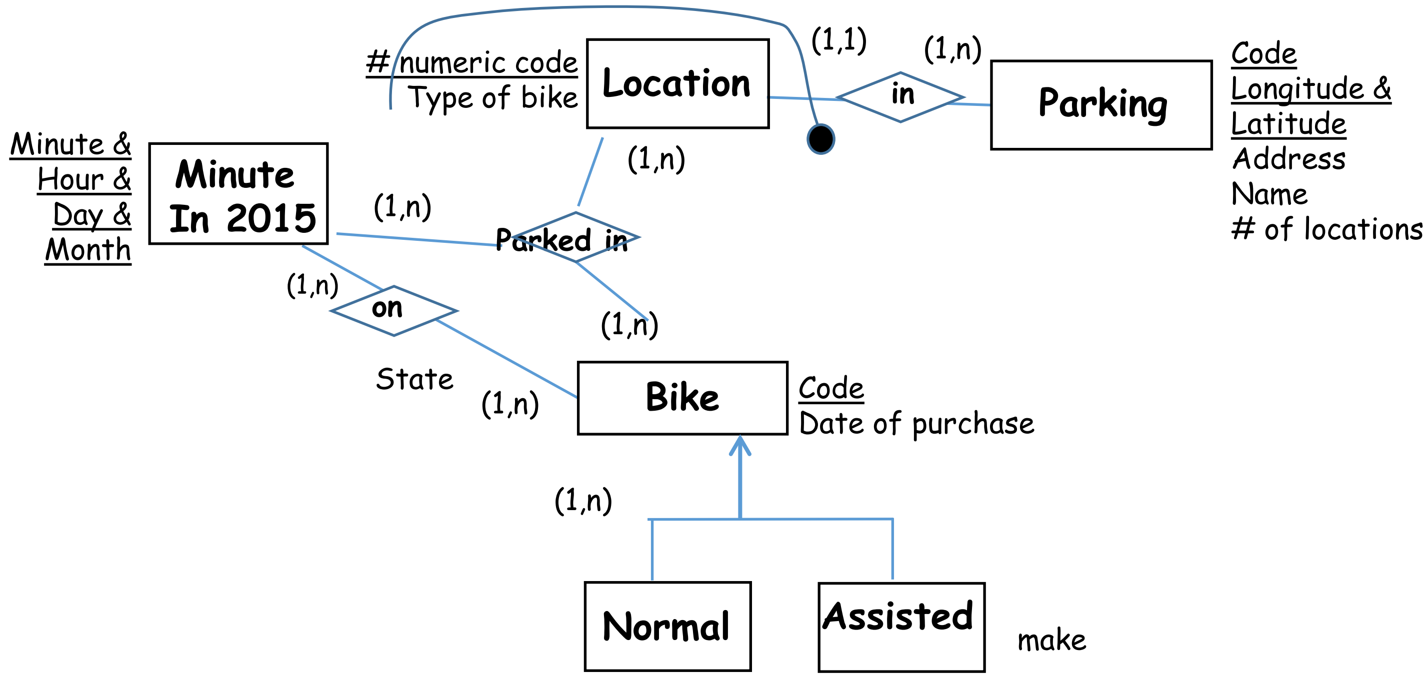


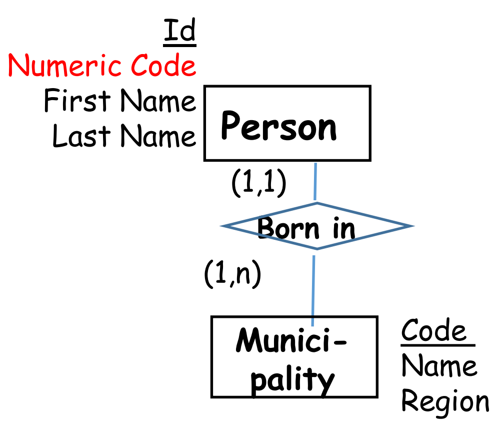
Possible errors in modeling Requirements R2

Only one identifier for Person (-0,5)

Comment

In this solution Numeric Code is not chosen as (second) internal identifier

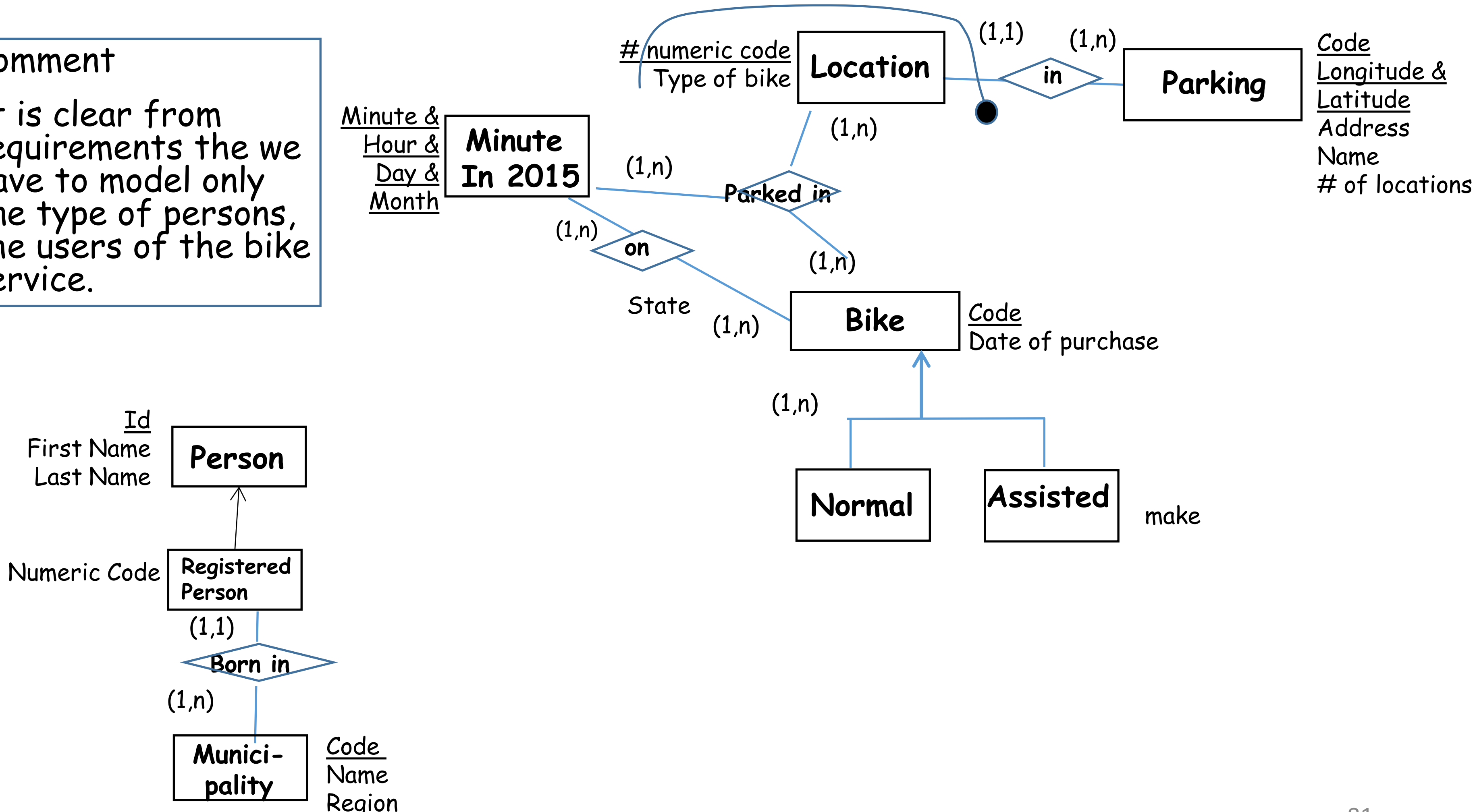




Distinguish two types of Persons (-0,5)

Comment

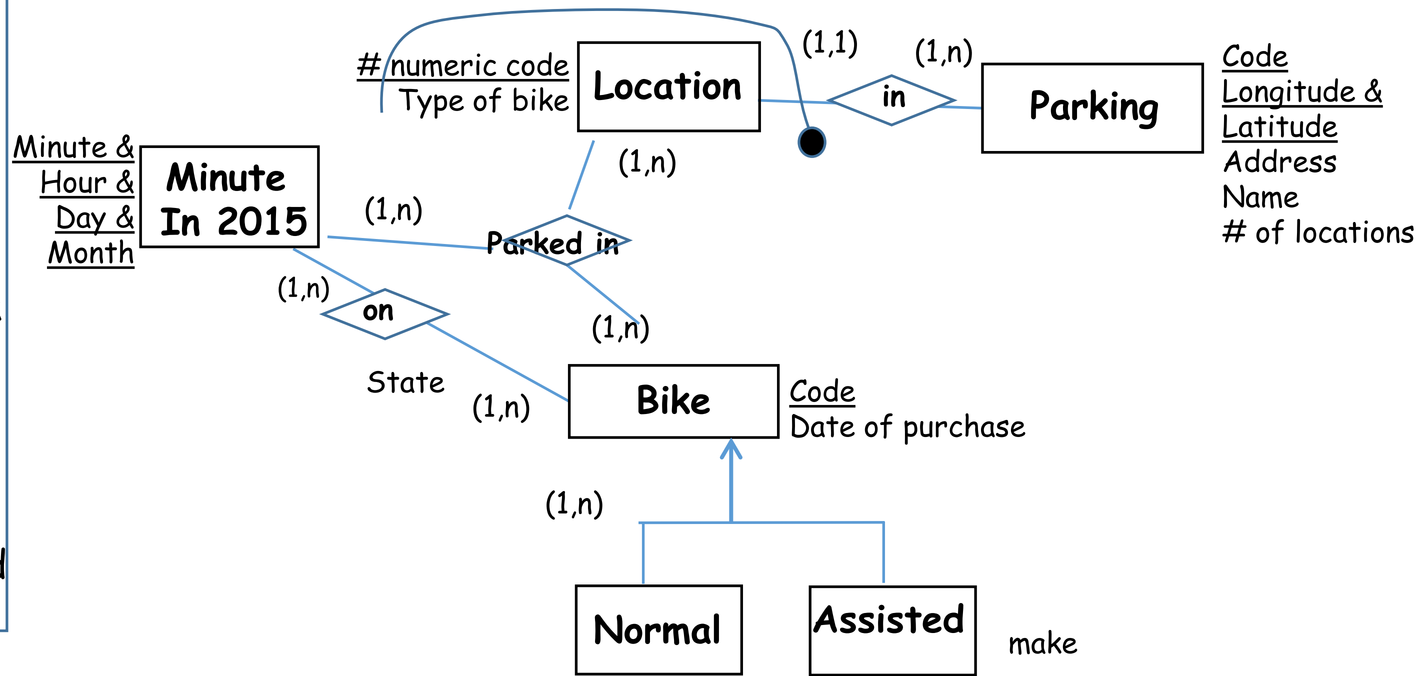
It is clear from requirements the we have to model only one type of persons, the users of the bike service.

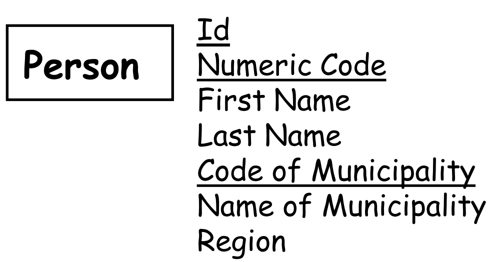


Only one entity between Person and Municiplity of Birth (-1,5)

Comment

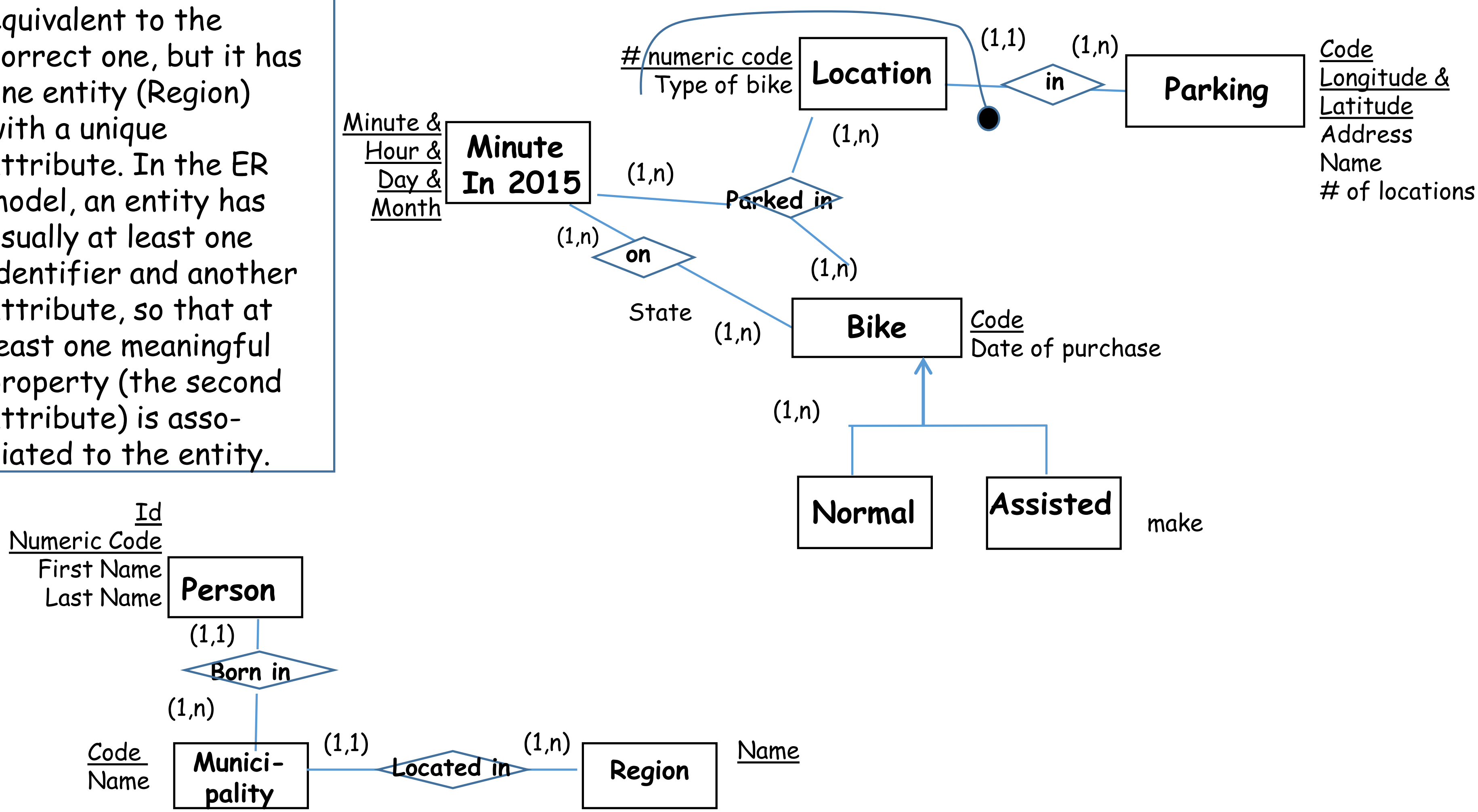
This is an error, since several properties pertain to Municipality, so that municipality matches the definition of Entity. Furthermore, instances of the schema are inherently redundant, because, if several users share the same municipality of birth, the correspoinding name and region are duplicated.





Entity Region with a unique attribute (-1,5)

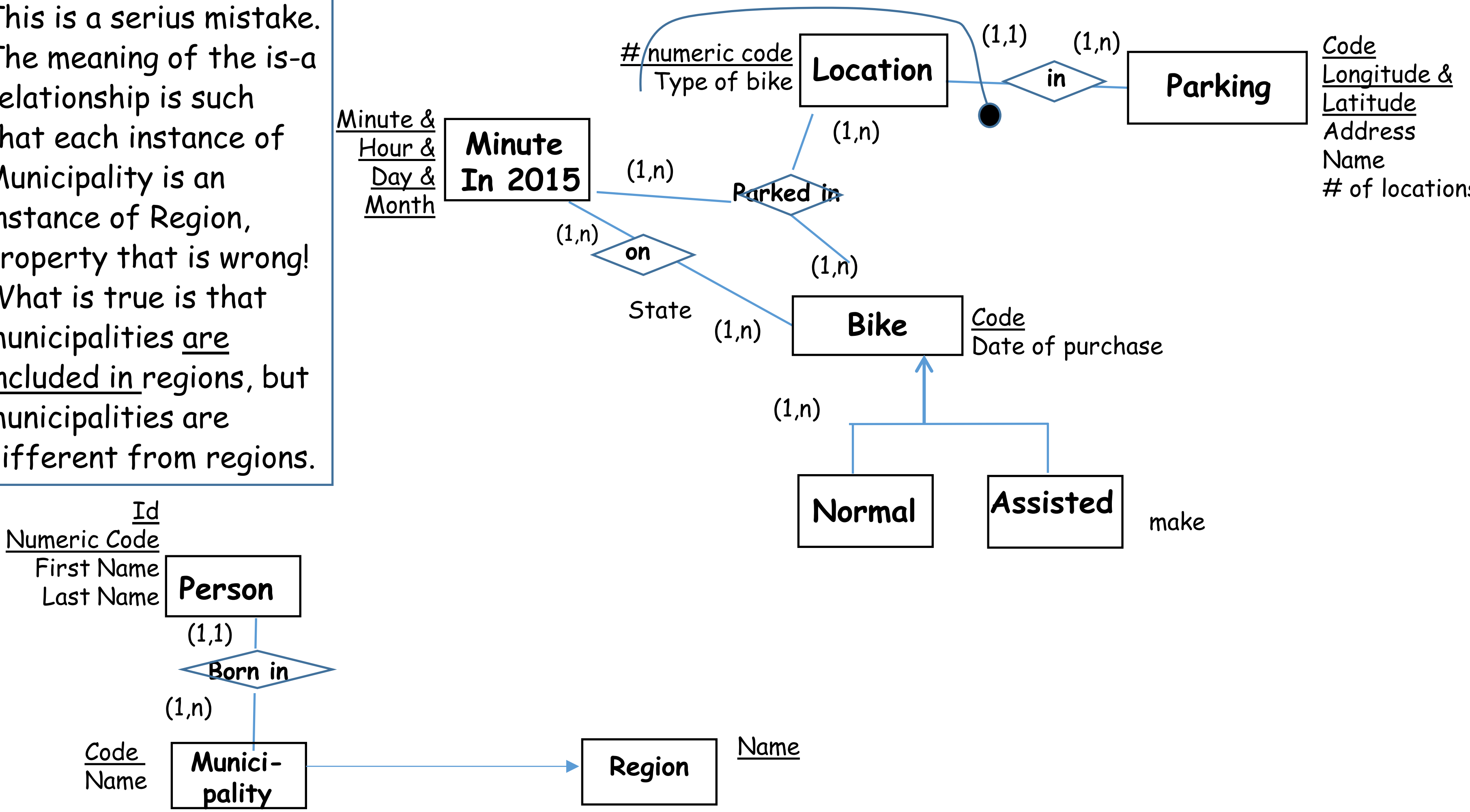
Comment
This solution is equivalent to the correct one, but it has one entity (Region) with a unique attribute. In the ER model, an entity has usually at least one identifier and another attribute, so that at least one meaningful property (the second attribute) is associated to the entity.



Entity Region in is-a with Entity Municipality (-4)

Remark

This is a serius mistake. The meaning of the is-a relationship is such that each instance of Municipality is an instance of Region, property that is wrong! What is true is that municipalities are included in regions, but municipalities are different from regions.



Requirements R3

Text of the assigment

R1 – A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type. Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes). When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

R2 – In order to use bikes, users have to register themselves. A registered user is characterized by an identifier (e.g. a fiscal code, or a social security number), a progressive numeric code (first person registered, second person registered, etc.) a name, a surname, date of birth, municipality of birth, with code, name of municipality and region (we assume that municipalities are located in regions).

R3 – Everytime a user picks up and subsequently returns a bike, such an event (pick up and return of a bike by a user) has to be recorded. The event is identified by a progressive number in the year (e.g. first event in 2015, second event in 2015, etc.) and duration (e.g. 12 minutes, 47 minutes); such duration has to be explicitely represented in the data base. The minute of pick up and the minute of return have to be associated to the event, besides, of course, the bike used and the user involved.

R4 – One or more credit cards are associated to users, with type (e.g. Visa, Diners), progressive code in the card type, and daily maximum amount of withdrawal (that depends on the card and on the person).

You have to represent above requirements with an Entity Relationship Schema, in terms of entities, relationships, attributes, internal and external idetitifers, generalizations, is-a relationships, minimum and maximum cardinalities.

Requirements R3

R3 – Everytime a user picks up and subsequently returns a bike, such an event (pick up and return of a bike by a user) has to be recorded. The event is identified by a progressive number in the year (e.g. first event in 2015, second event in 2015, etc.) and duration (e.g. 12 minutes, 47 minutes); such duration has to be explicitely represented in the data base. The minute of pick up and the minute of return have to be associated to the event, besides, of course, the bike used and the user involved.

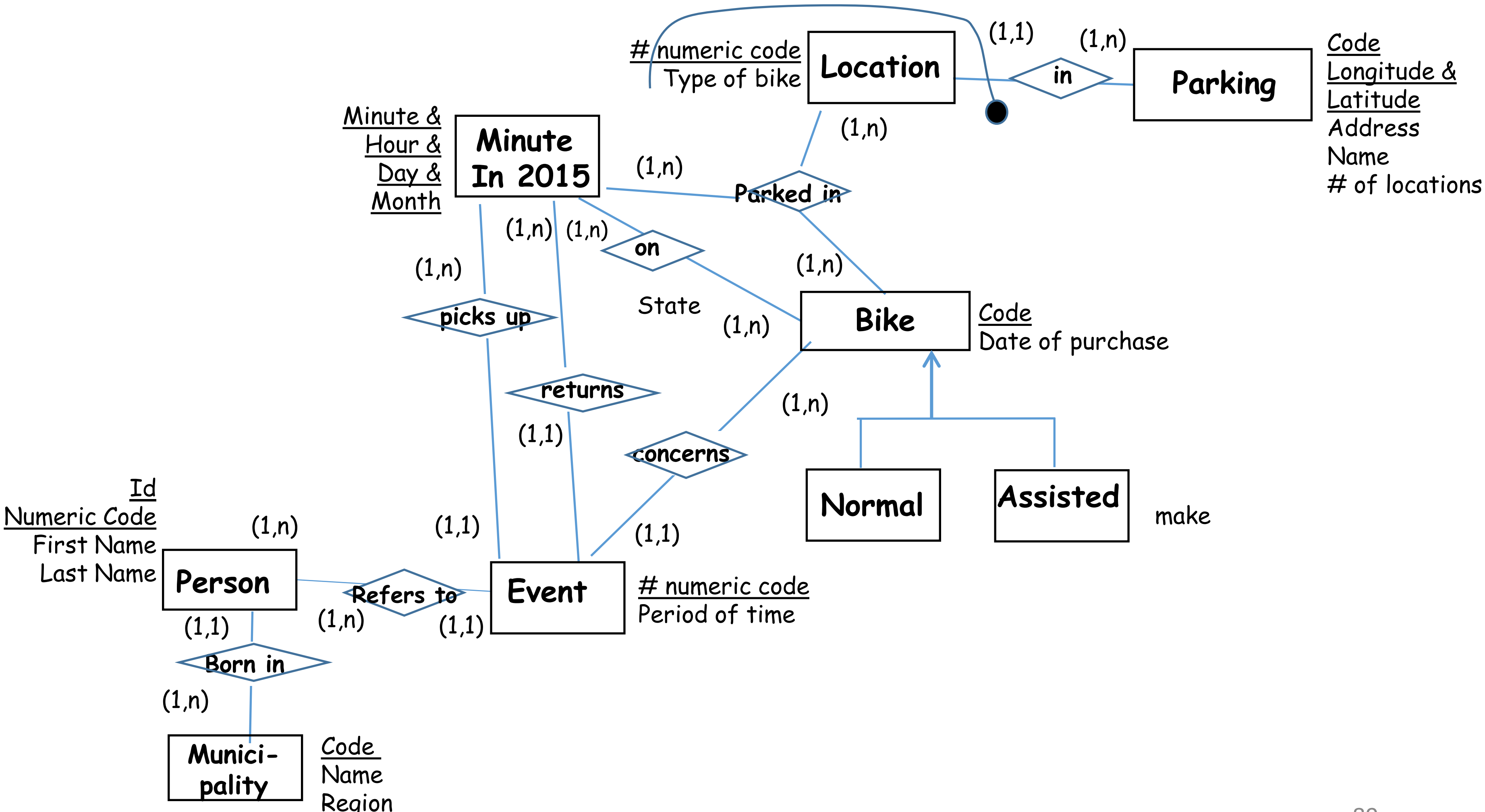
Event is by far the most complex concept in requirements, due to its both temporal and spatial character. Several concepts present in the schema defined so far, have to be related to the concept «event». The representation of such conceptual relationships changes, depending from the choice to model event either a. as an entity, or b. as a relationship.

Let us see the two cases separaterly.

Requirements R3
Event modeled as an Entity

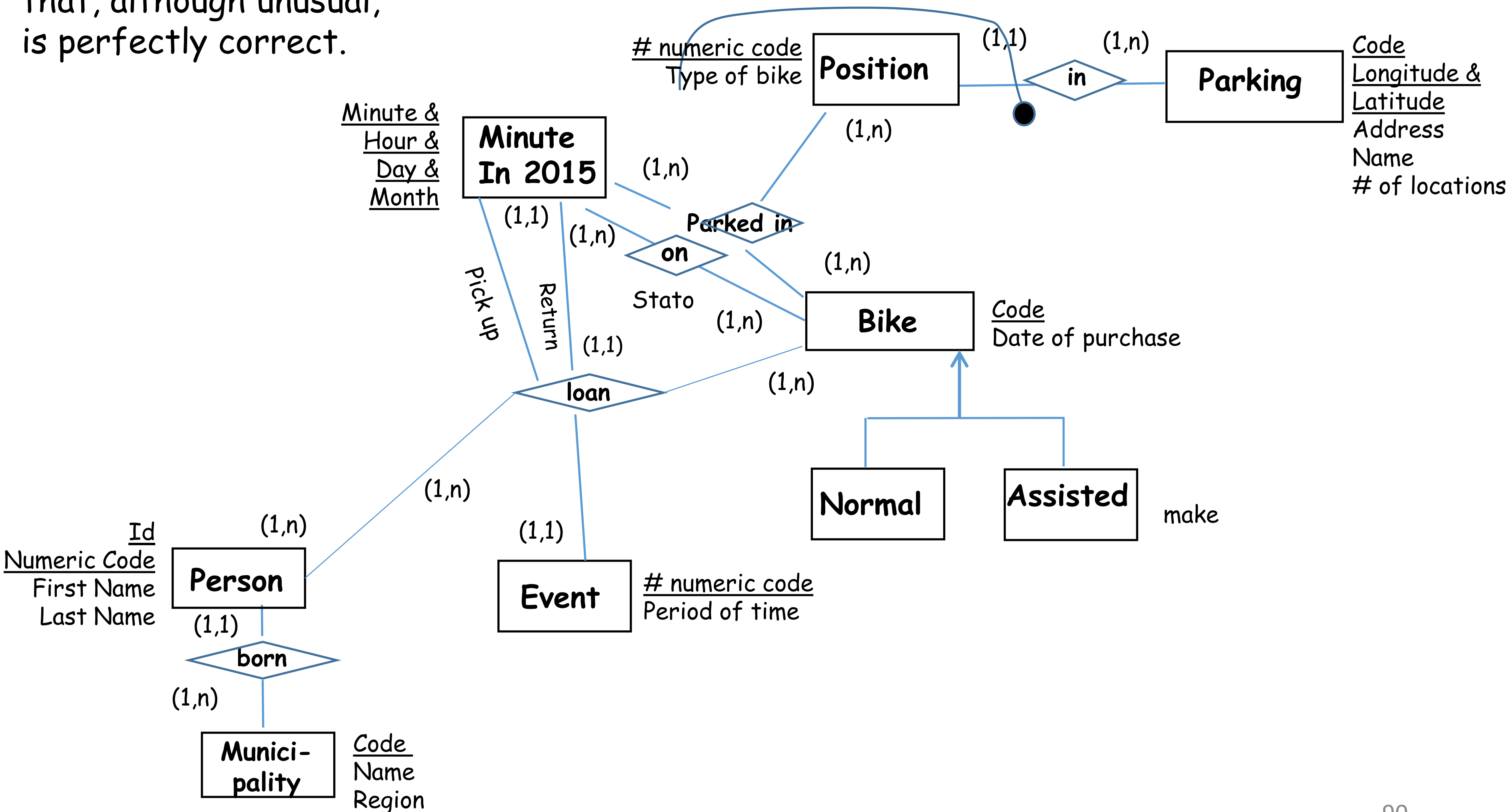
Solution with Event as an Entity

This is the most «natural» modeling of Event, because in requirements it is said that Event is characterized by a progressive numeric code and a Period of time. Such modeling solution leads to model with binary relationships, of type 1 to n its relationship with a. user, b. the two pick up and return minutes, and c. bike.



Second solution with event as Entity

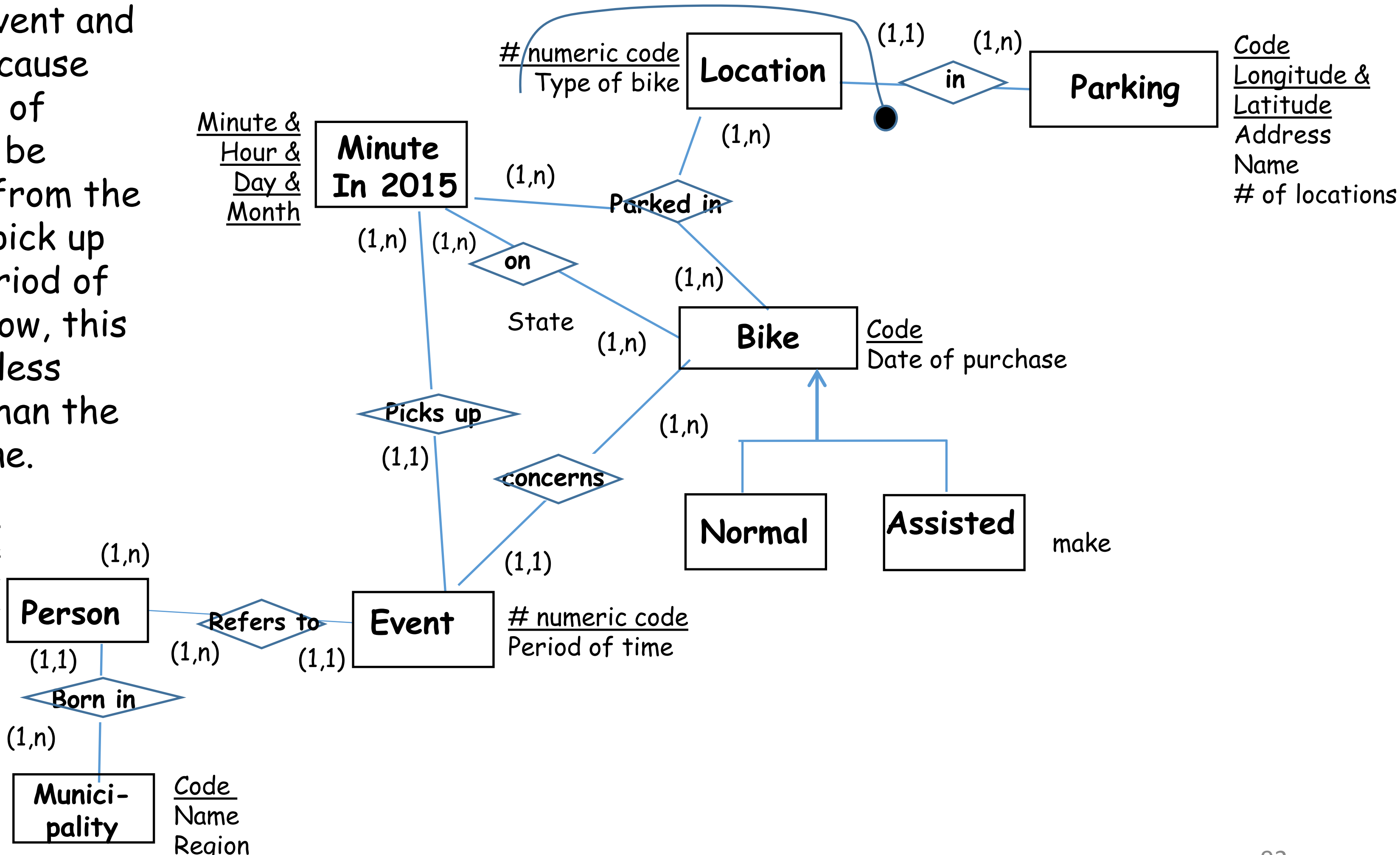
Comment
In this case we have
a relationship with degree n = 5,
that, although unusual,
is perfectly correct.



Requirements R3
Alternative solutions
with event modeled as entity

Alternative solution with Event as an Entity

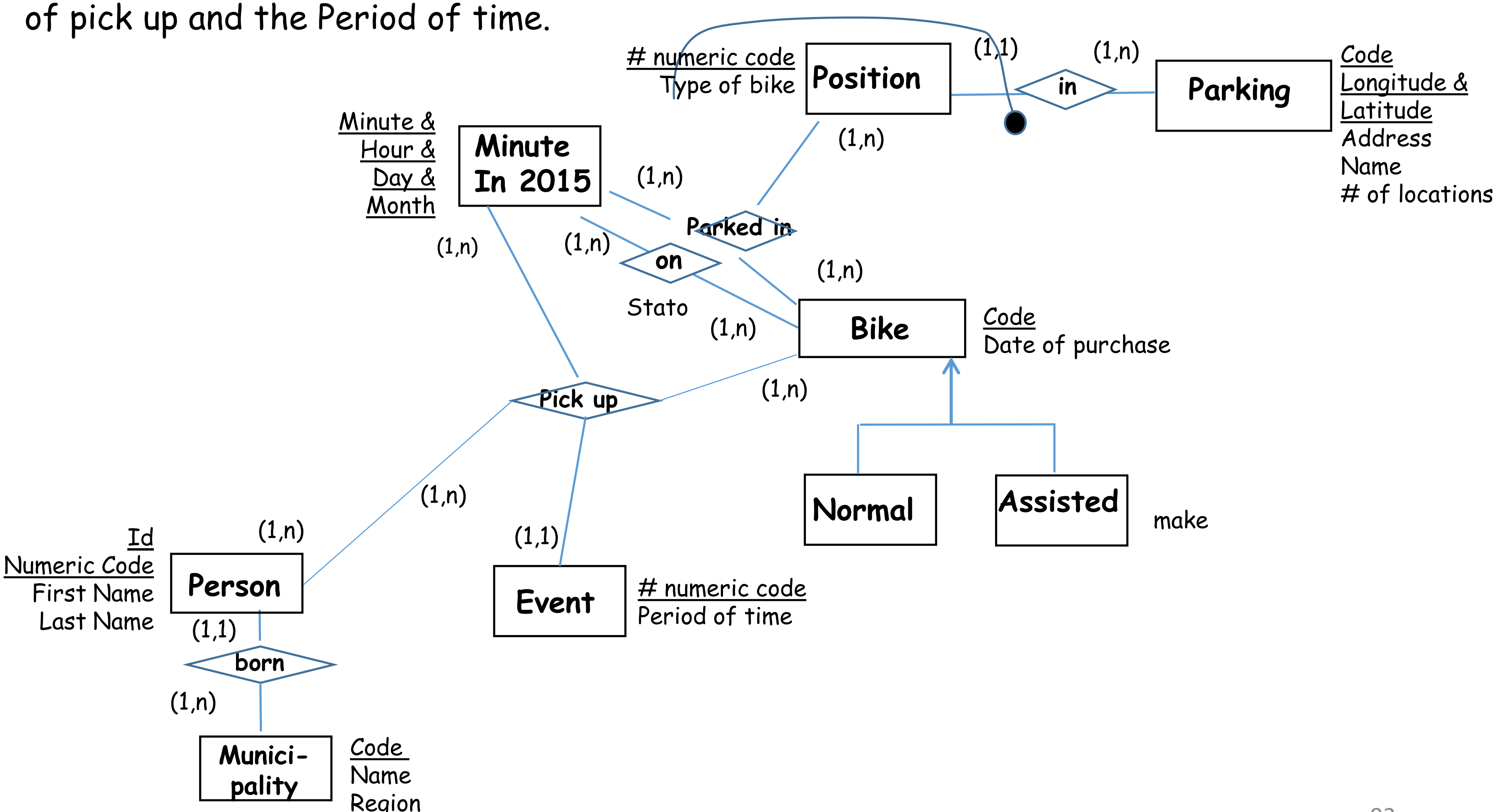
Comment
In this case, we do not need two relationships between Event and Minute, because the minute of return can be computed from the minute of pick up and the Period of time. Anyhow, this solution is less readable than the previous one.



Alternative solution with event as Entity

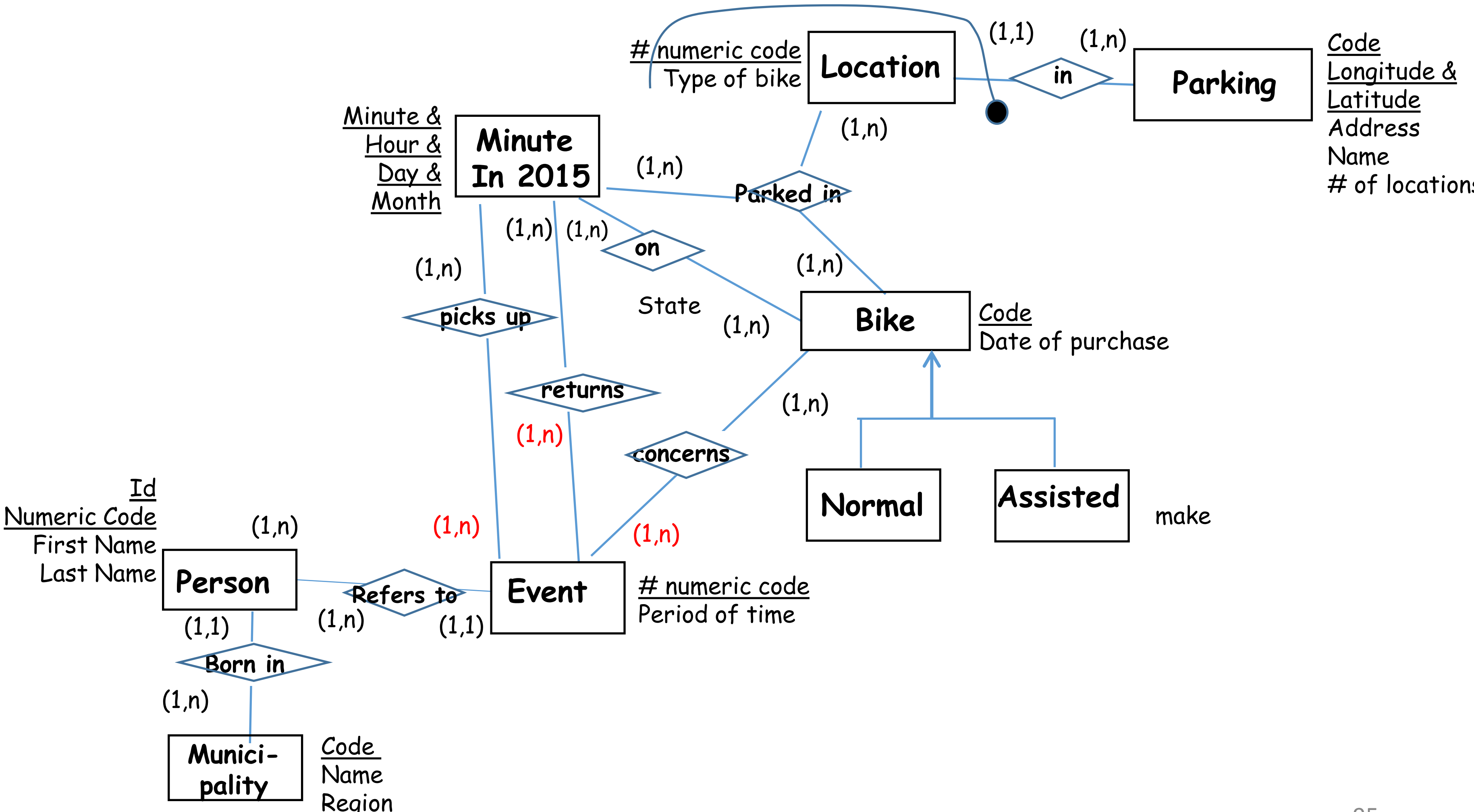
Comment

Similarly, we do not necessarily need a relationships with degree n=5. We can use a quaternary relationship, because the minute of return can be computed from the minute of pick up and the Period of time.

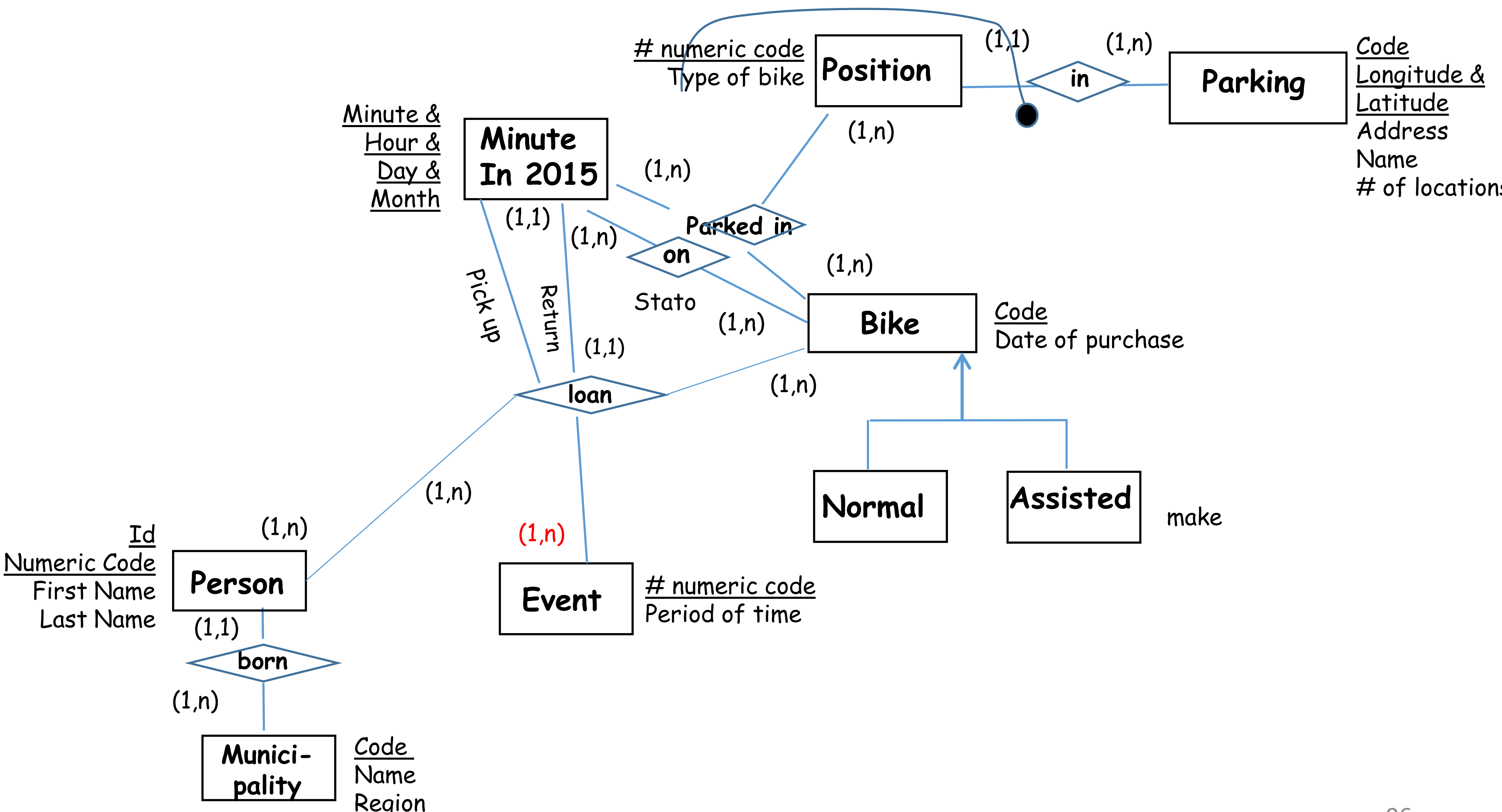


Requirements R3
Wrong solutions
with event modeled as entity - cardinalities

Wrong Solution with Event as an Entity (- 1)

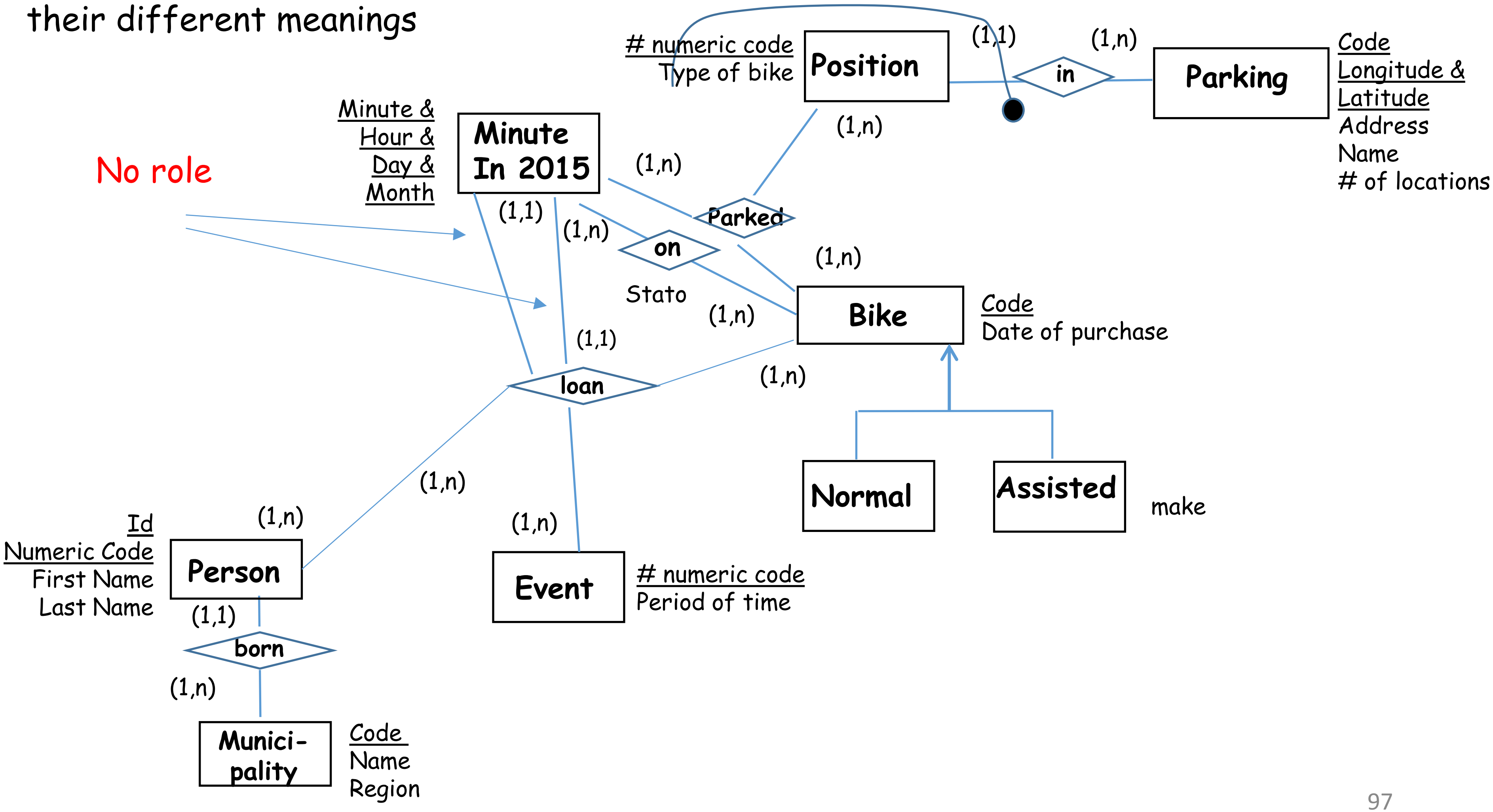


Second solution with event as Entity (-1)



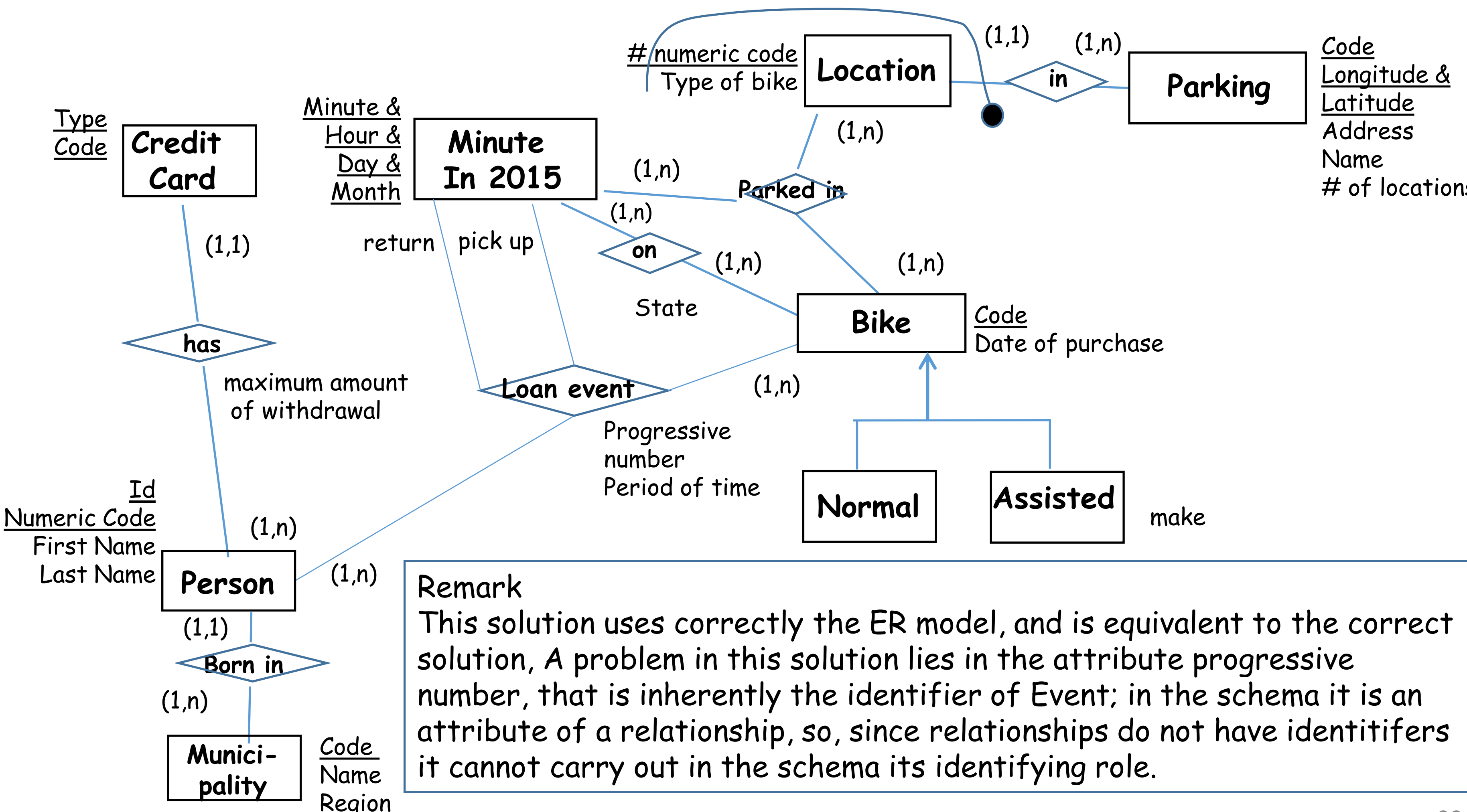
Wrong solution with event as Entity (-0,5)

Comment
without a role defined for the two relationships, we cannot distinguish their different meanings



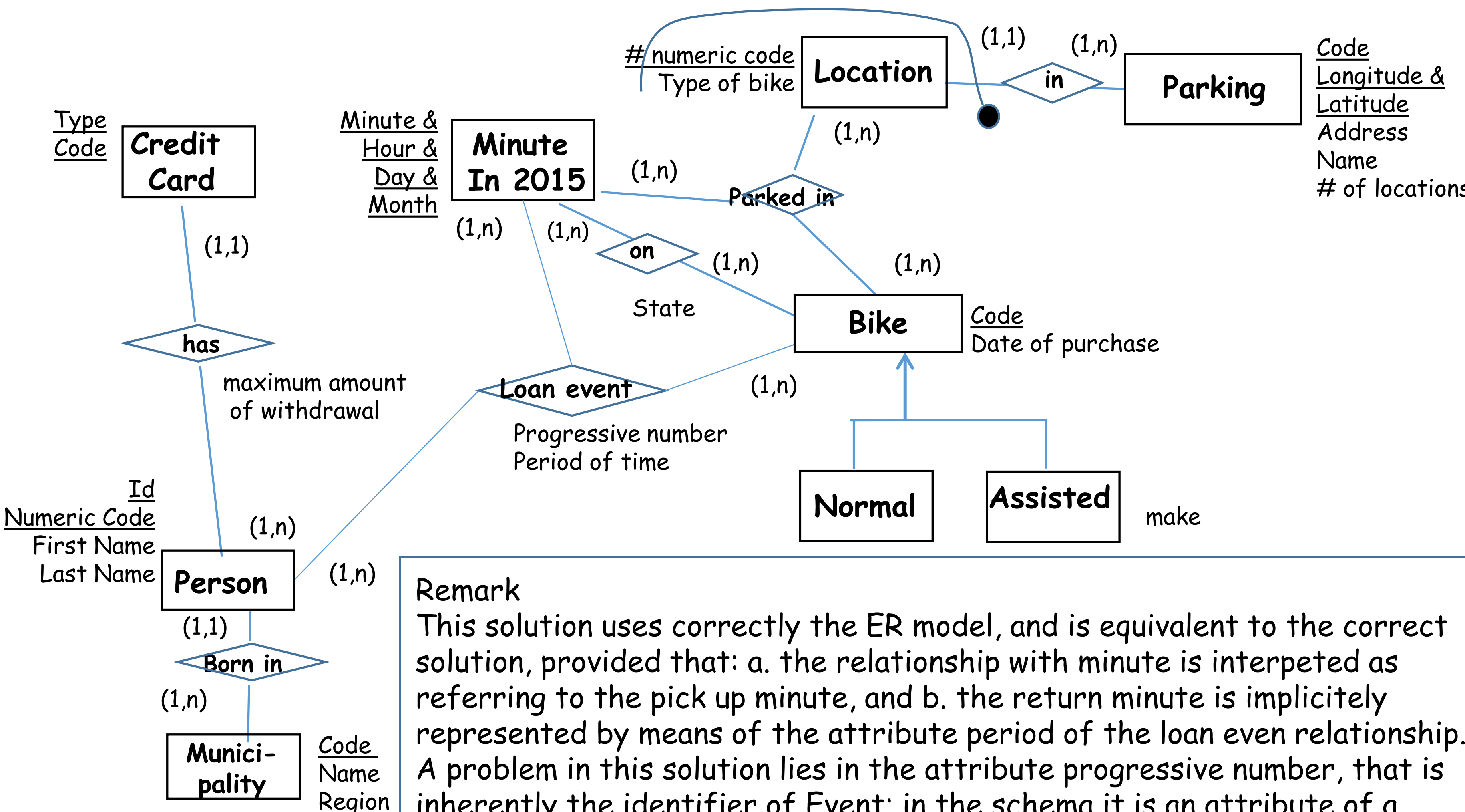
Requirements R3
Event modeled with a relationship

Solution 1 with event modeled with a relationship



Remark
This solution uses correctly the ER model, and is equivalent to the correct solution, A problem in this solution lies in the attribute progressive number, that is inherently the identifier of Event; in the schema it is an attribute of a relationship, so, since relationships do not have identitifers it cannot carry out in the schema its identifying role.

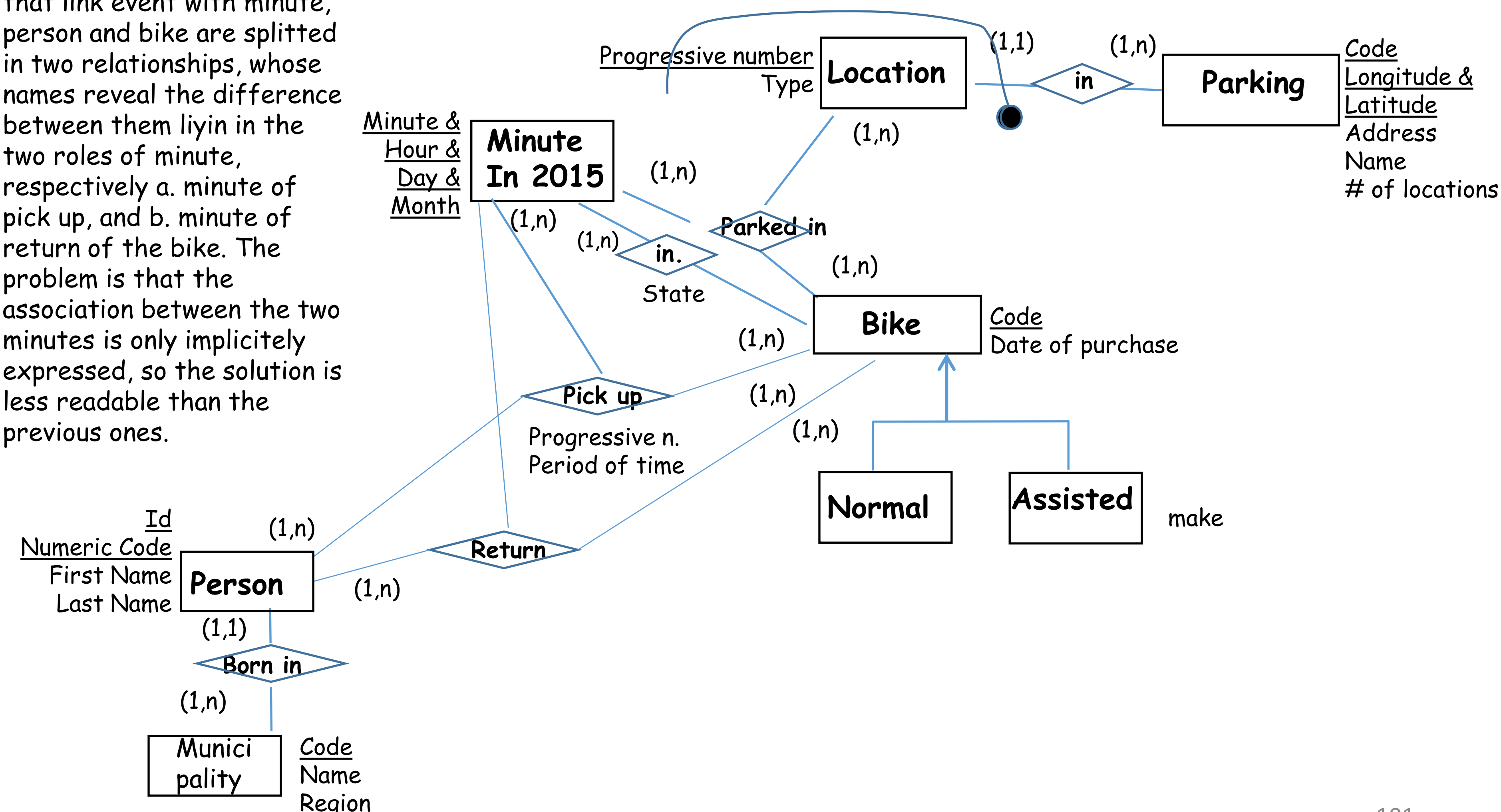
Solution 2 with event modeled with a relationship



Remark
This solution uses correctly the ER model, and is equivalent to the correct solution, provided that: a. the relationship with minute is interpeted as referring to the pick up minute, and b. the return minute is implicitely represented by means of the attribute period of the loan even relationship. A problem in this solution lies in the attribute progressive number, that is inherently the identifier of Event; in the schema it is an attribute of a relationship, so, since relationships do not have identitifers it cannot carry out in the schema its identifying role.

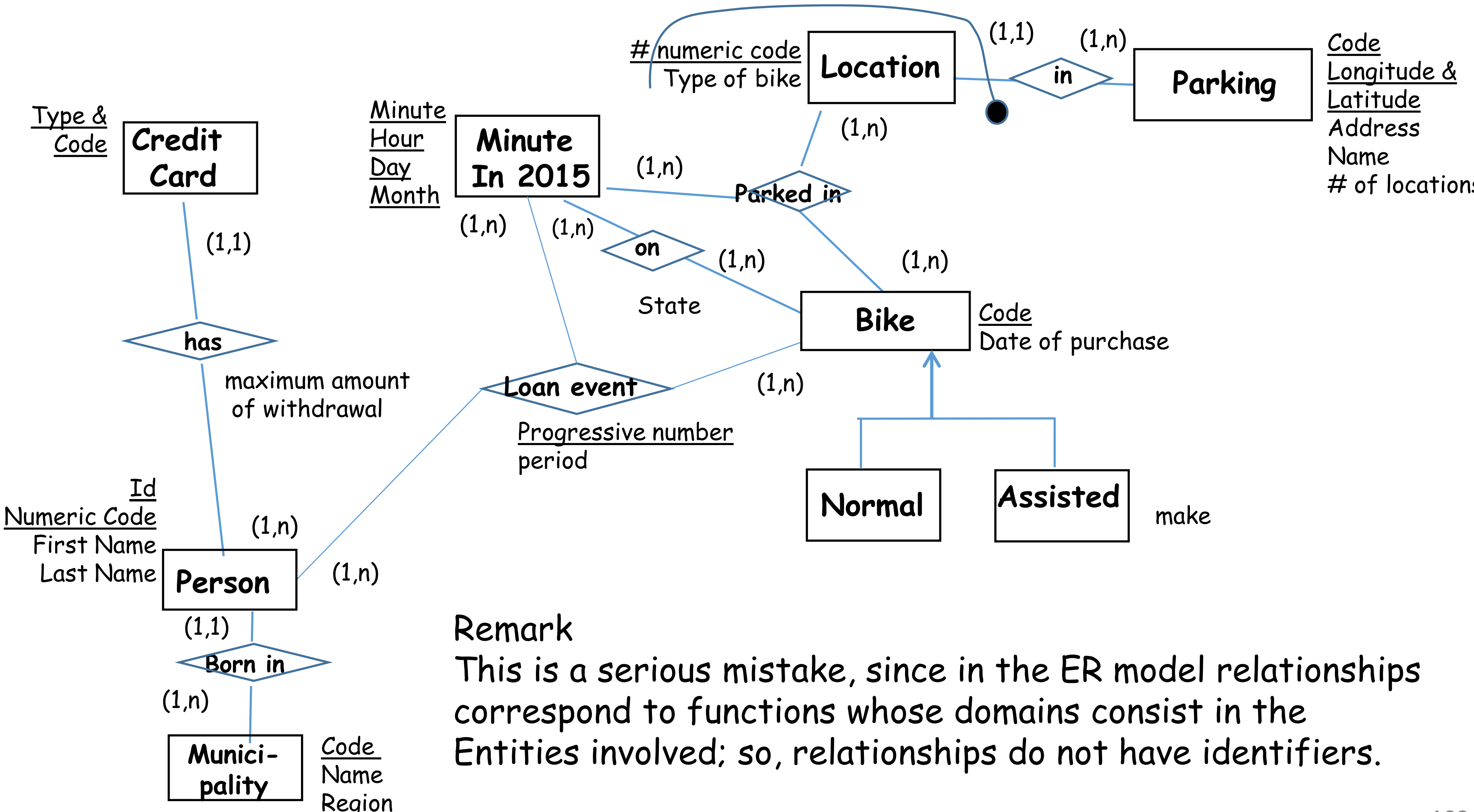
Solution 3 with two ternary relationships for Event

Remark
This solution is a variant of the previous one; here, the diverse semantic relations that link event with minute, person and bike are splitted in two relationships, whose names reveal the difference between them liyin in the two roles of minute, respectively a. minute of pick up, and b. minute of return of the bike. The problem is that the association between the two minutes is only implicitely expressed, so the solution is less readable than the previous ones.



Wrong solutions for Requirements R3

Solution with an identifier for the relationship (-3)



Remark

This is a serious mistake, since in the ER model relationships correspond to functions whose domains consist in the Entities involved; so, relationships do not have identifiers.

Requirements R4

Requirements R4

R1 – A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type. Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes). When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

R2 – In order to use bikes, users have to register themselves. A registered user is characterized by an identifier (e.g. a fiscal code, or a social security number), a progressive numeric code (first person registered, second person registered, etc.) a name, a surname, date of birth, municipality of birth, with code, name of municipality and region (we assume that municipalities are located in regions).

R3 – Everytime a user picks up and subsequently returns a bike, such an event (pick up and return of a bike by a user) has to be recorded. The event is identified by a progressive number in the year (e.g. first event in 2015, second event in 2015, etc.) and duration (e.g. 12 minutes, 47 minutes); such duration has to be explicitely represented in the data base. The minute of pick up and the minute of return have to be associated to the event, besides, of course, the bike used and the user involved.

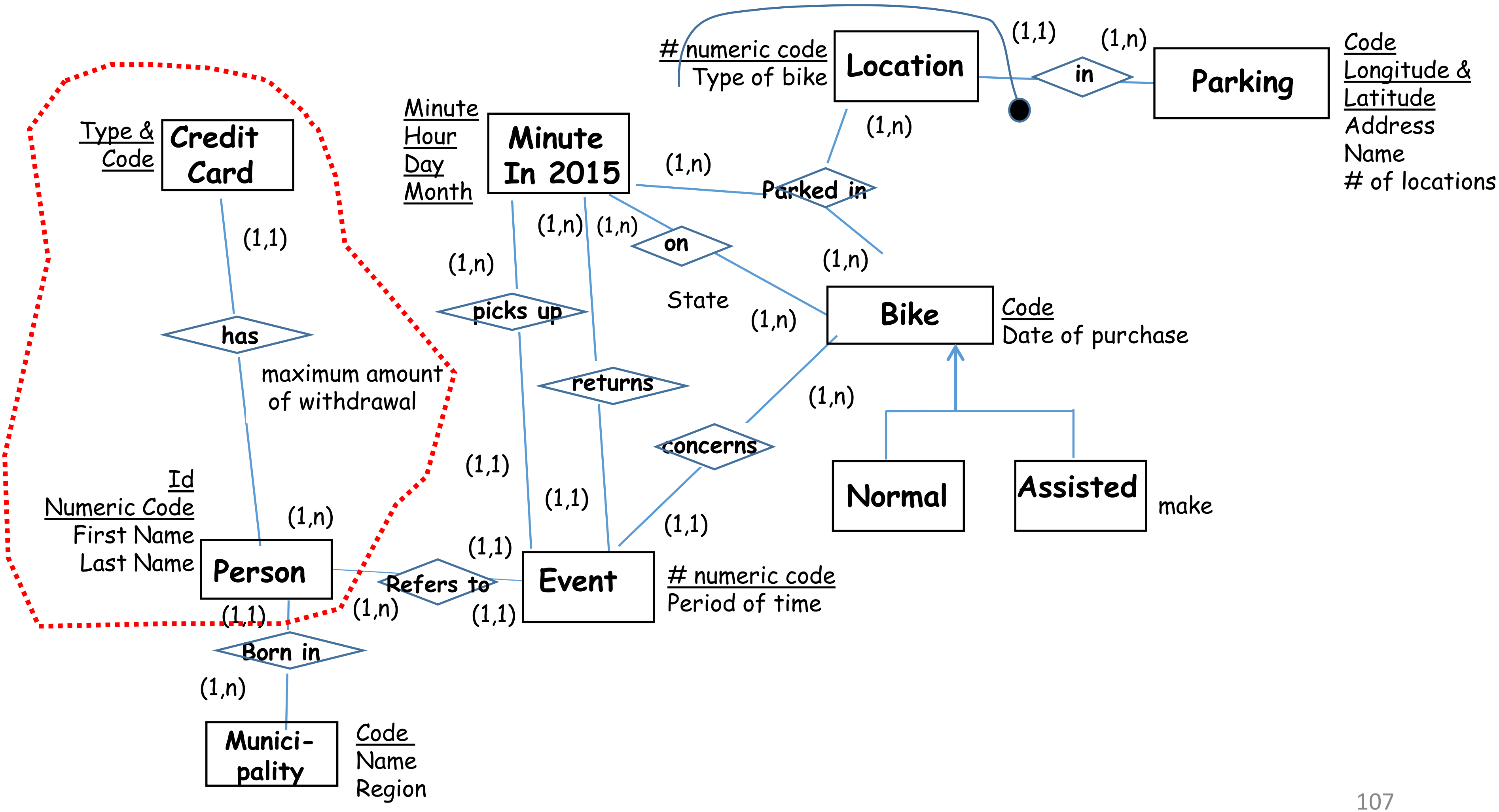
R4 – One or more credit cards are associated to users, with type (e.g. Visa, Diners), progressive code in the card type, and daily maximum amount of withdrawal (that depends on the card and on the person).

Requirements R4

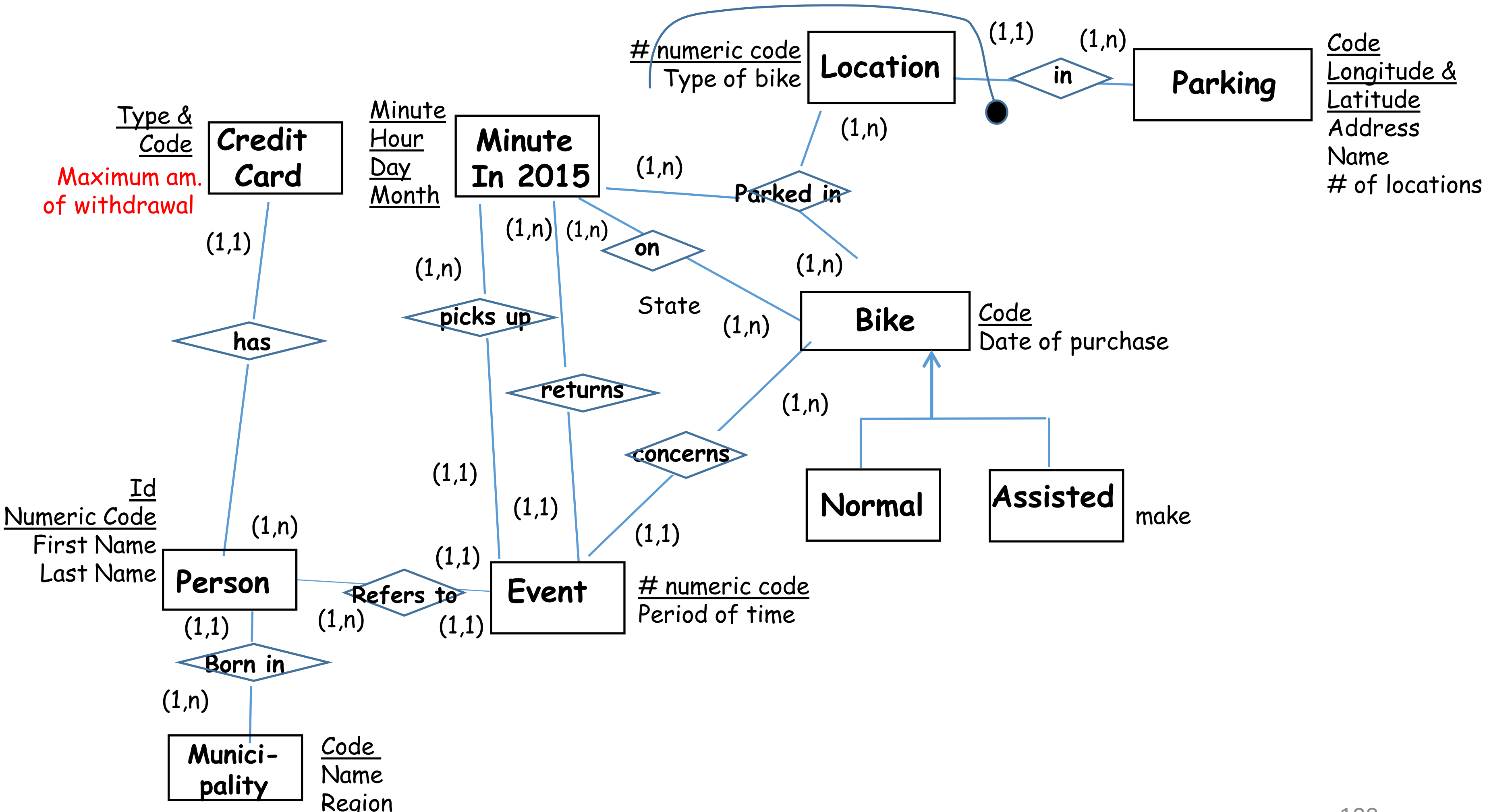
R4 – One or more credit cards are associated to users, with type (e.g. Visa, Diners), progressive code in the card type, and daily maximum amount of withdrawal (that depends on the card and on the person).

See solutions in the following slides.

Modeling Requirements R4

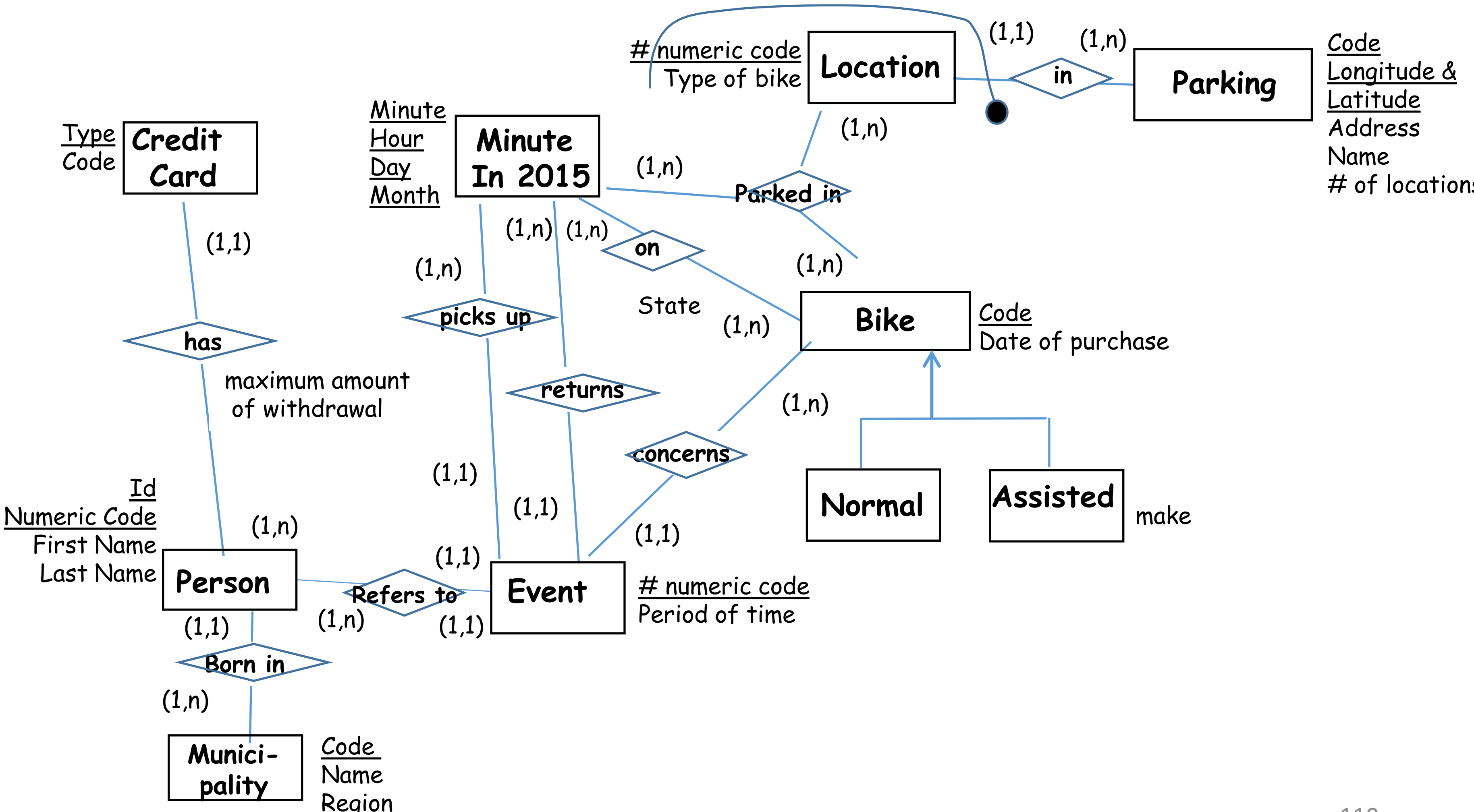


Alternative attribution of attribute



Wrong solutions in modeling Requirements R4

Identifier of Credit Card made of only one attribute (-1,5)



More on Entity Event and Entity Bike

Entities Event and Bike and their possibile generalizations

As we have seen in previous discussion, and as it is evident from schemas representing requirements, entities Event and Bike play an important role in the schema.

Some students have proposed generalizations for Event and Bike that have not been discussed so far. We deal with them in the following.

Entity Event

Modeling choices for Event as a generalization

Assume we want to extend Event with a generalization. The natural choice is to choose Pick up Event and Return Event as child entities (this has been also the proposed solution of some students). If we think to the meaning of Event in this case, we have that the instances of events corresponding e.g. to the one with event number = 345 are two:

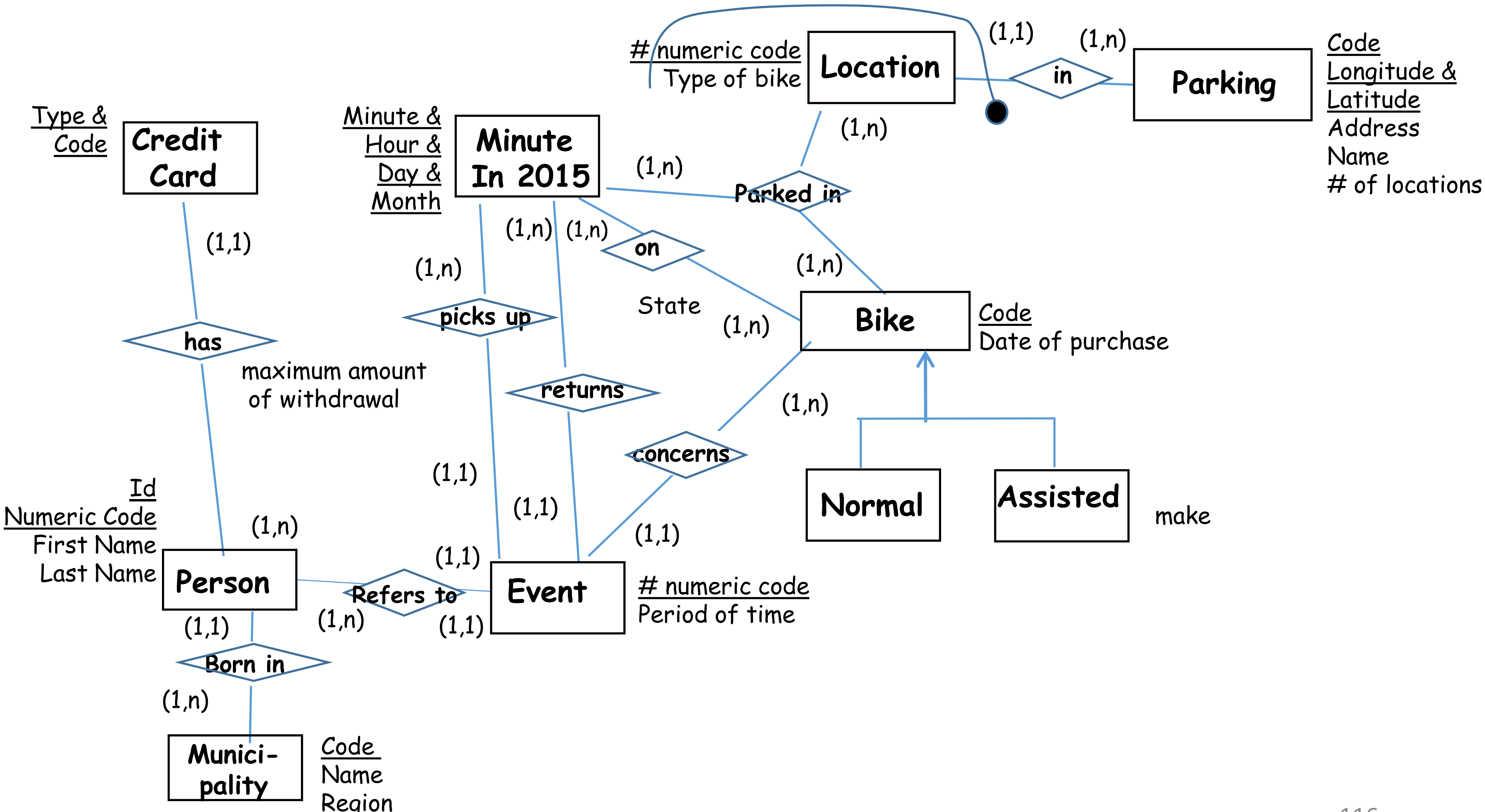
a. the pick up event related to event number 345 and

b. the corresponding return event.

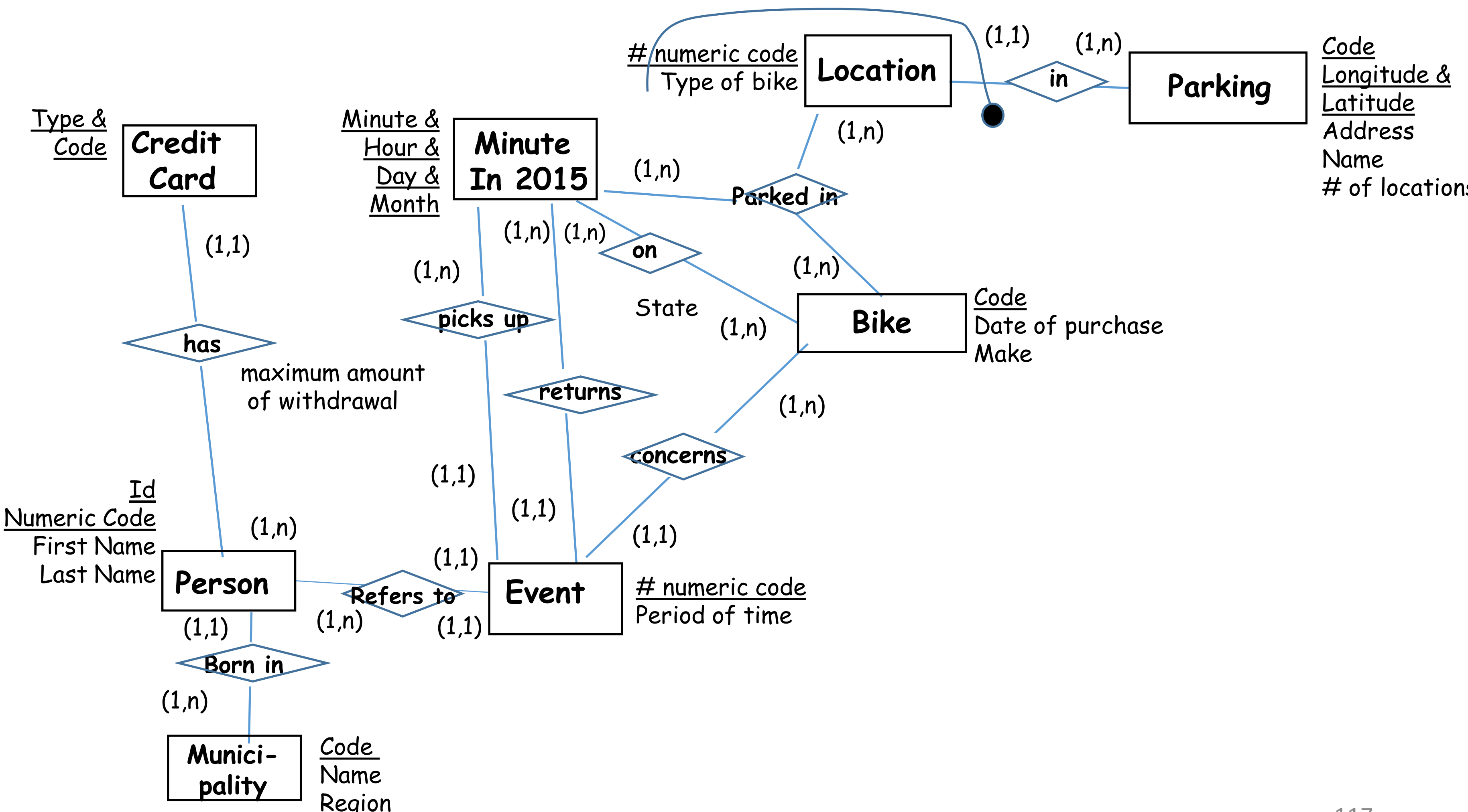
This is a case of schema that is wrongly related to requirements. A priori, this cannot be considered an error, we have to investigate which are the consequent changes to the schema, and check if we produce an equivalent schema; in this case we have to compare the previous schema and the new schema from the point of view of schema qualities. Being the equivalent, the only comparison quality that can be used here is readability.

First case

Let us come back to the reference solution

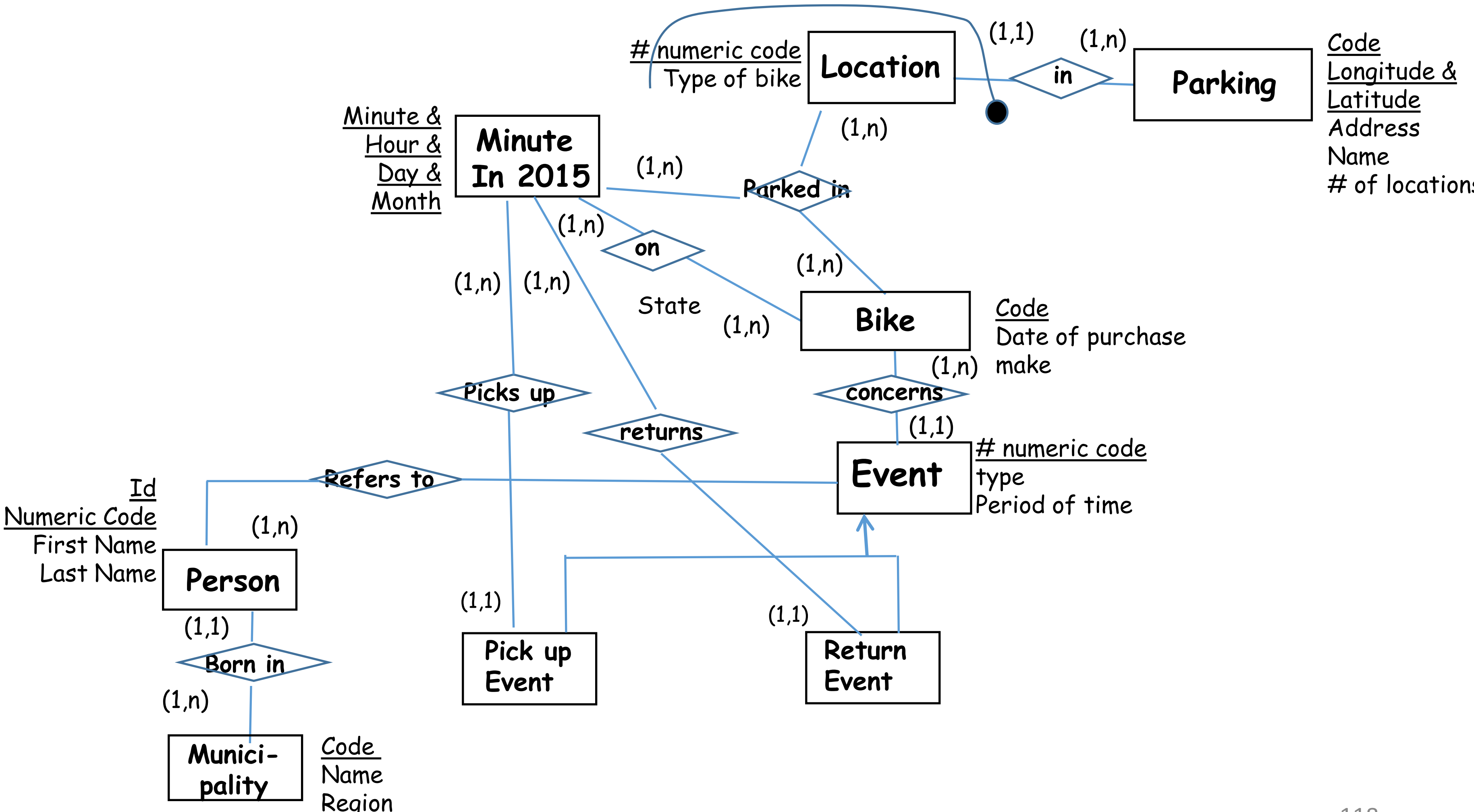


For sake of clarity, we remove from Bike the generalization

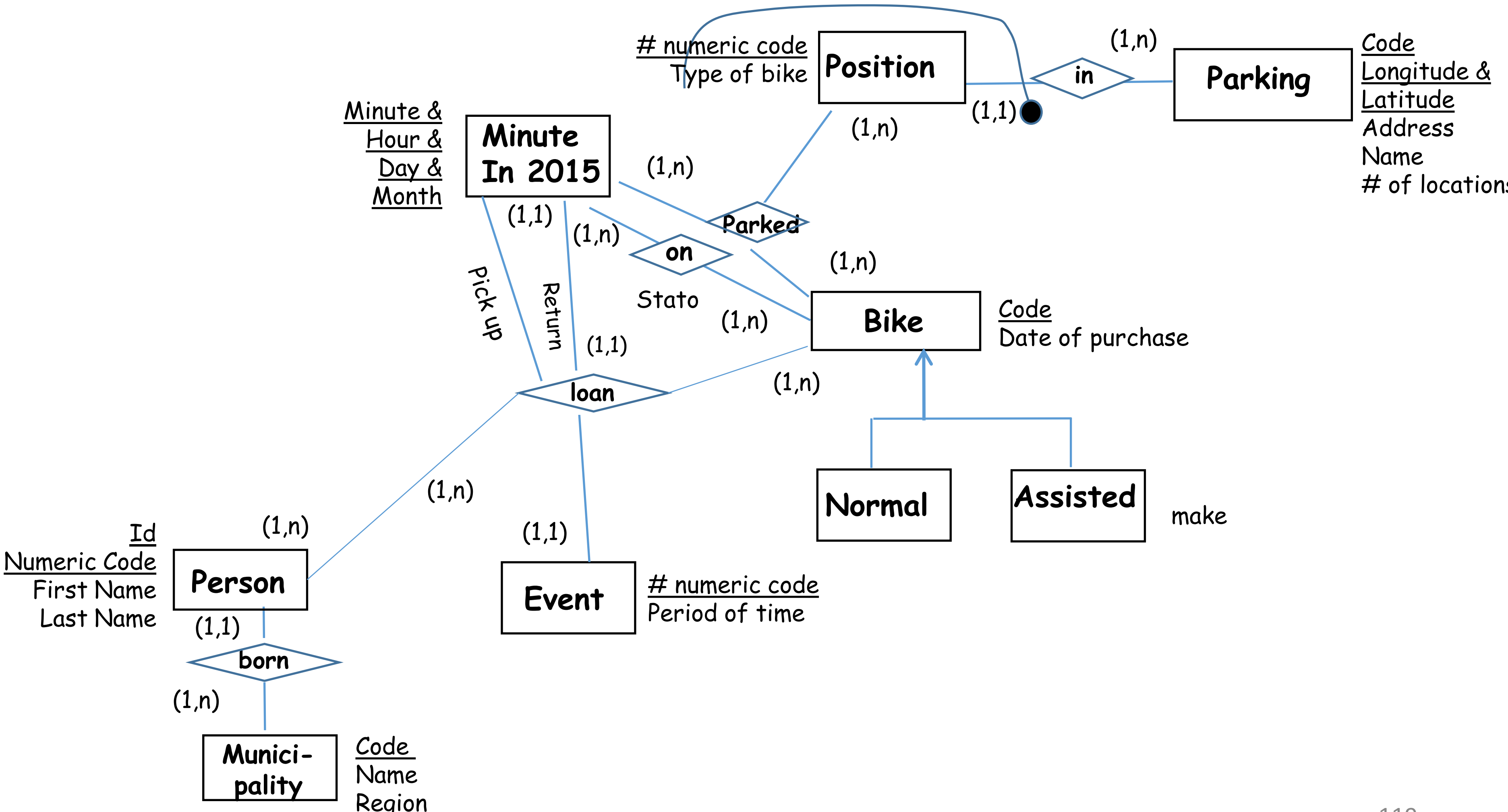


Result of changing the «new» entity Event into a generalization first case

Comment – To keep the same meaning of the original schema, we have to attribute the two Pick up and Return relationships to the two new entities in the generalization. Attributes of Event remain associated to event. It does not seem that we have enhanced the readability of the schema, we had only to introduce new concepts, reducing the overall compactness.



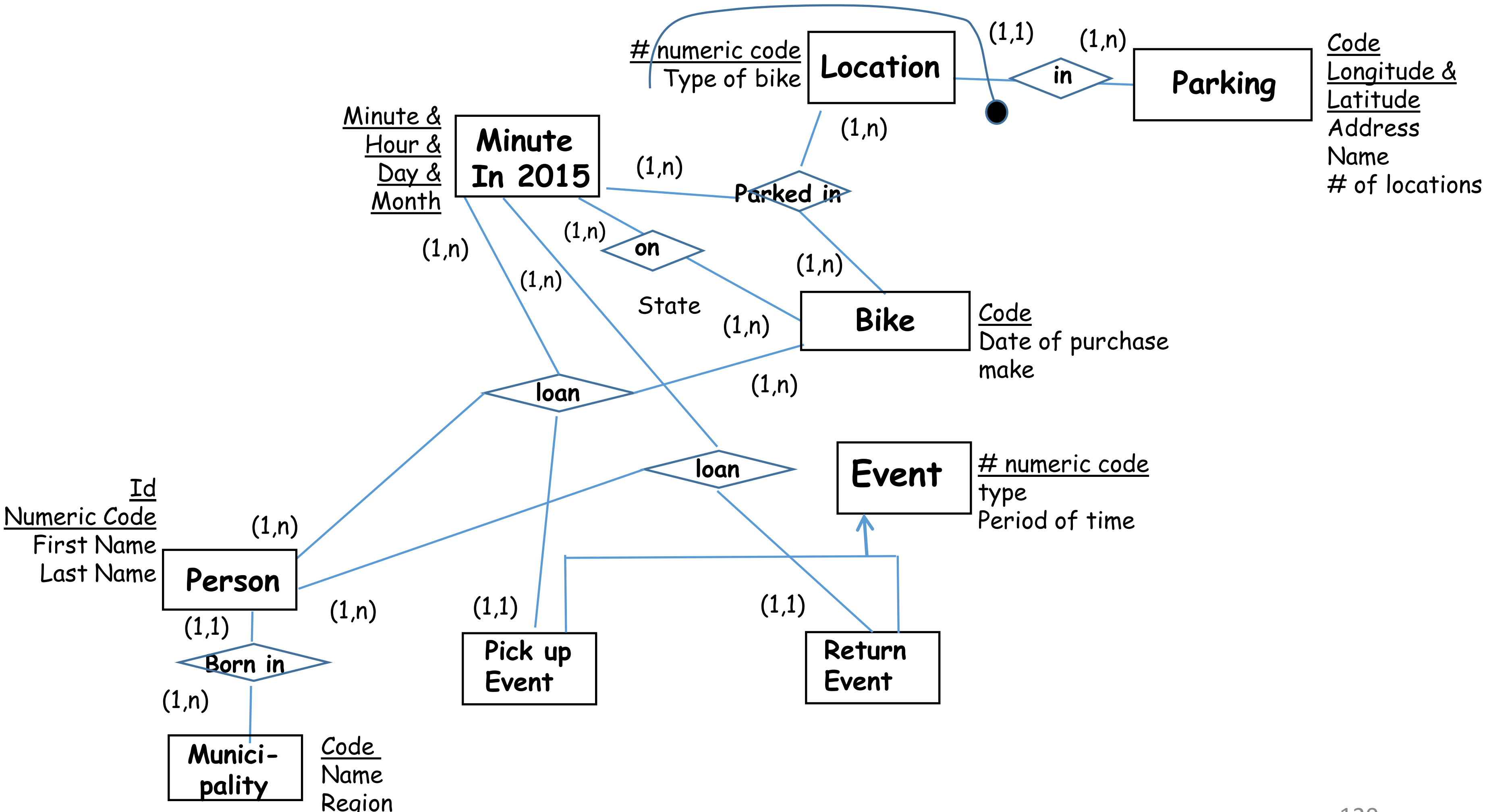
Second case



Code
Longitude &
Latitude
Address
Name
# of locations
Parking
(1,n)
in
(1,1)
numeric code
Type of bike
Position
(1,n)
Minute &
Hour &
Day &
Month
Minute
In 2015
(1,n)
Parked
(1,1)
(1,n)
on
(1,n)
Stato
(1,n)
Pick up
Return
(1,1)
Bike
Code
Date of purchase
loan
(1,n)
(1,n)
Normal
Assisted
make
Id
Numeric Code
First Name
Last Name
(1,n)
Person
(1,1)
Event
numeric code
Period of time
(1,1)
born
(1,n)
Munici-
pality
Code
Name
Region

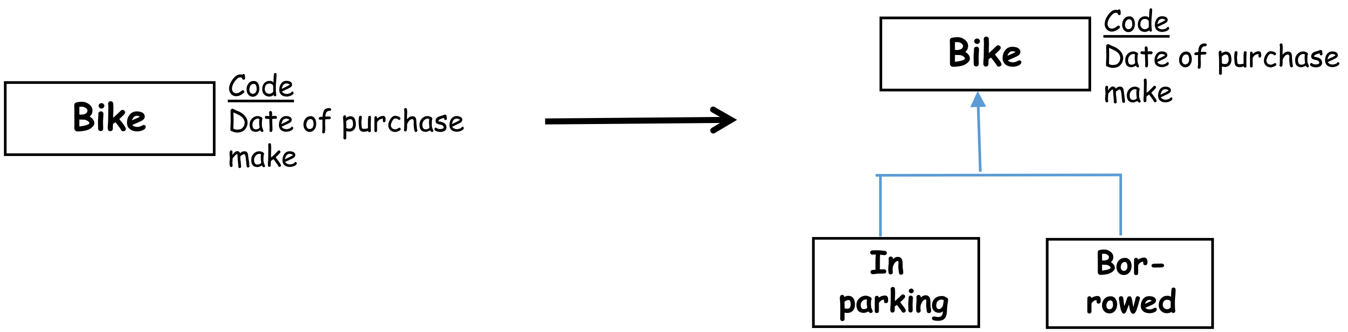
Result of changing the «new» entity Event into a generalization - second case

Comment – Also in this case, the introduction of new entities results in the need to split the previous relationship into two relationship, without no evident gain in readability.



Entity Bike

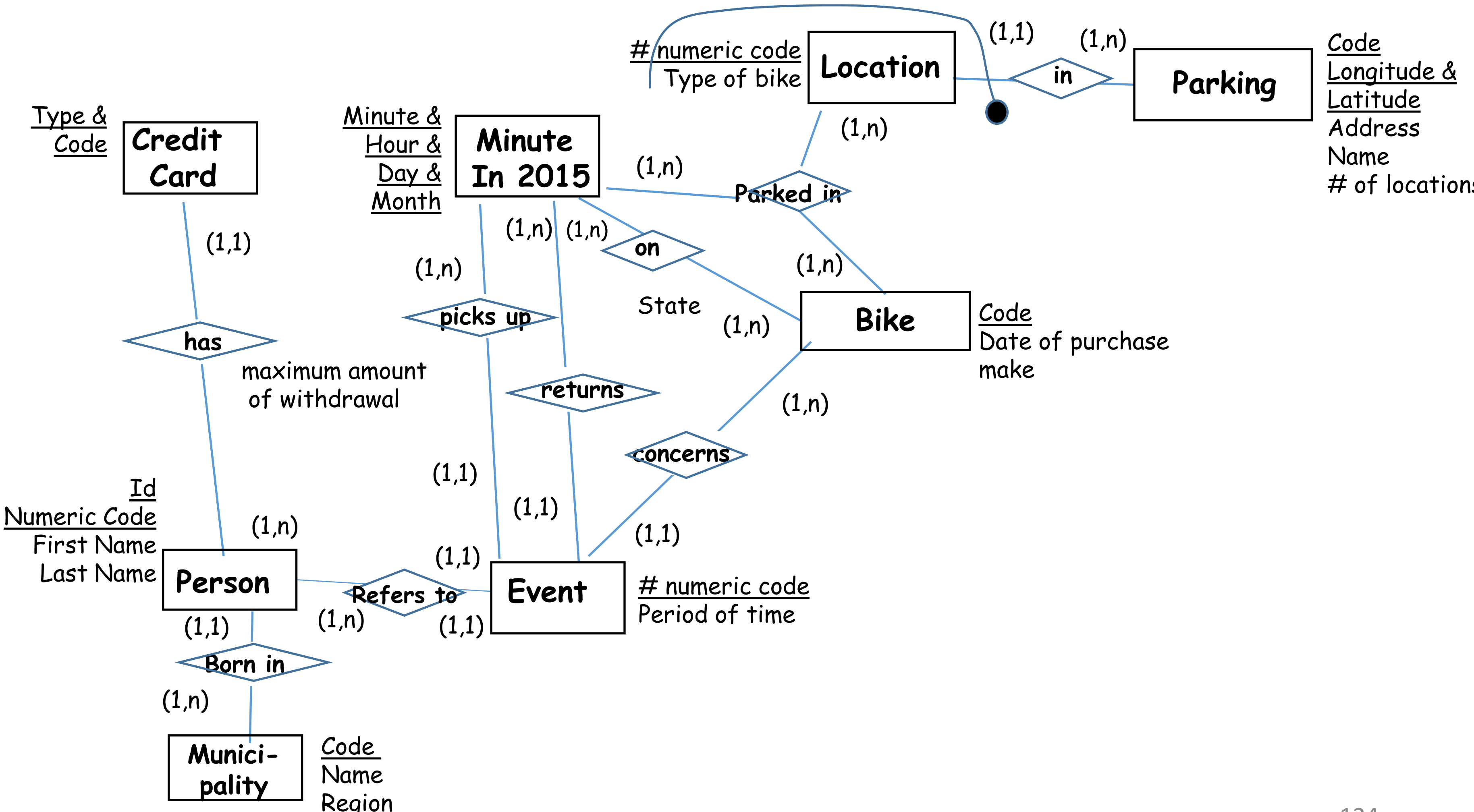
The new generalization



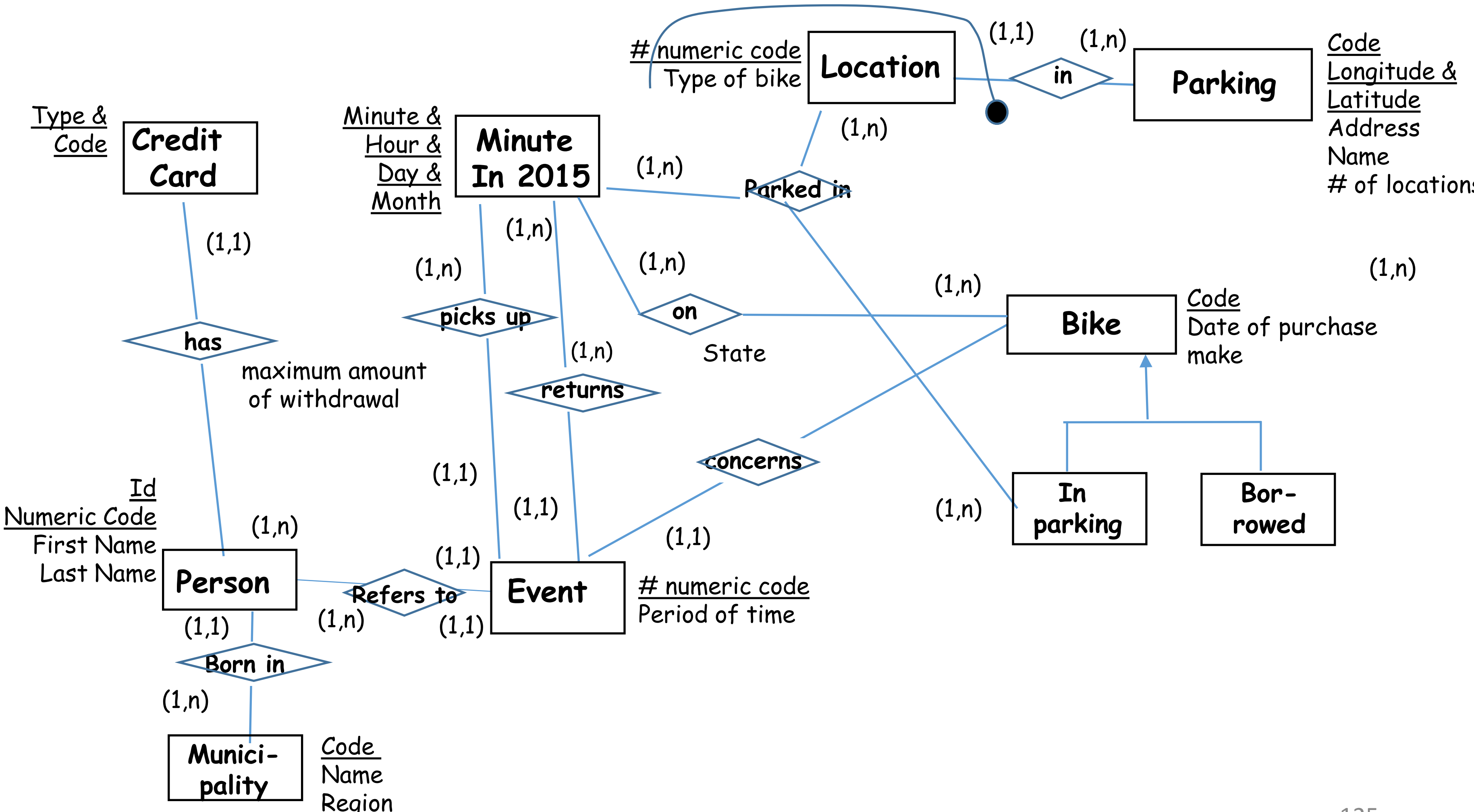
Consequences of the introduction of the new generalization

- The introduction of the new generalization leads to consequences that need for some comment. We first show the new schemas resulting from the generalization, then we comment the solution.

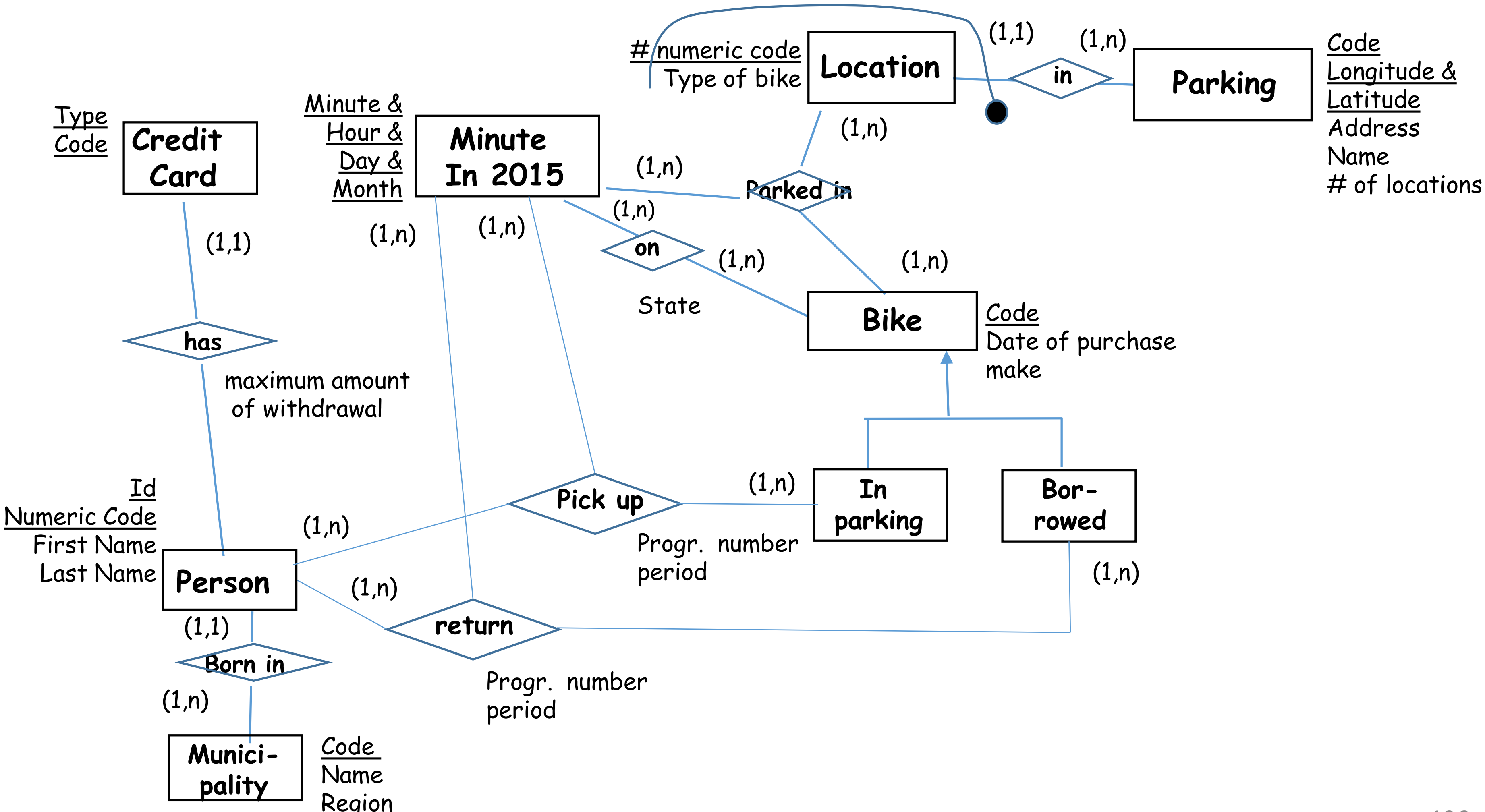
Again, let us come back to the first reference solution



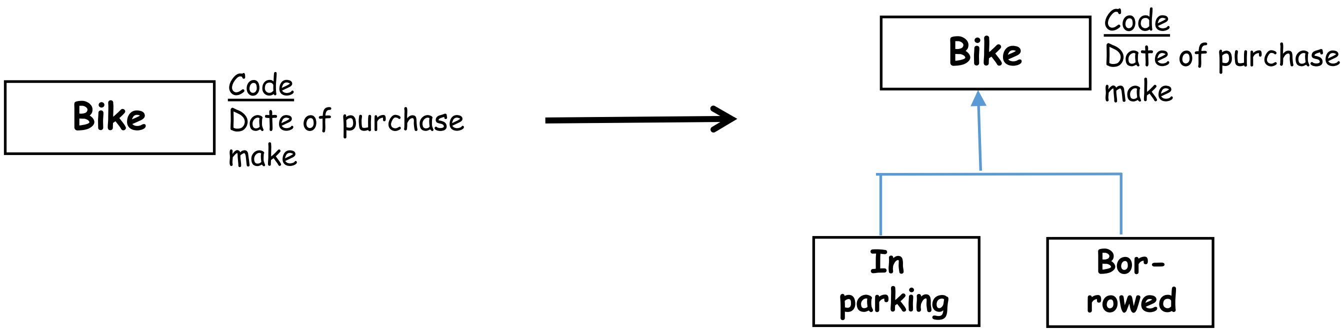
New solution with event modeled as an entity



Solution 2 with event modeled with a relationship



Consequences of the introduction of the generalization



Here, as a result of the introduction of the two child entities, the instances of bike do not change. The difference w.r.t to the previous case is that, due to the presence of the new two entities, every time an (instance of) bike changes its state from «in parking» to «borrowed» or vice versa, it changes the entity, namely the class of instances to which it belongs. There are two strong objections to this choice:

a. Entities are defined as classes of objects that have an identification, share common properties, and are of interest for the application. As such, their instances are basically stable in time; this does not mean that they never change in time, but that change with a low frequency.

b. Although during conceptual design we are not concerned with efficiency issues, if we model entities whose instances continuosly change, due to translation rules in logical design rules, there is a concrete possibility that we model the two entities with two tables, in such a way that at run time we have to frequently delete/insert tuples corresponding to bikes in the two tables. This is the exact counterpart at run time of the above a. argument.

Diagrammatic readability more in depth

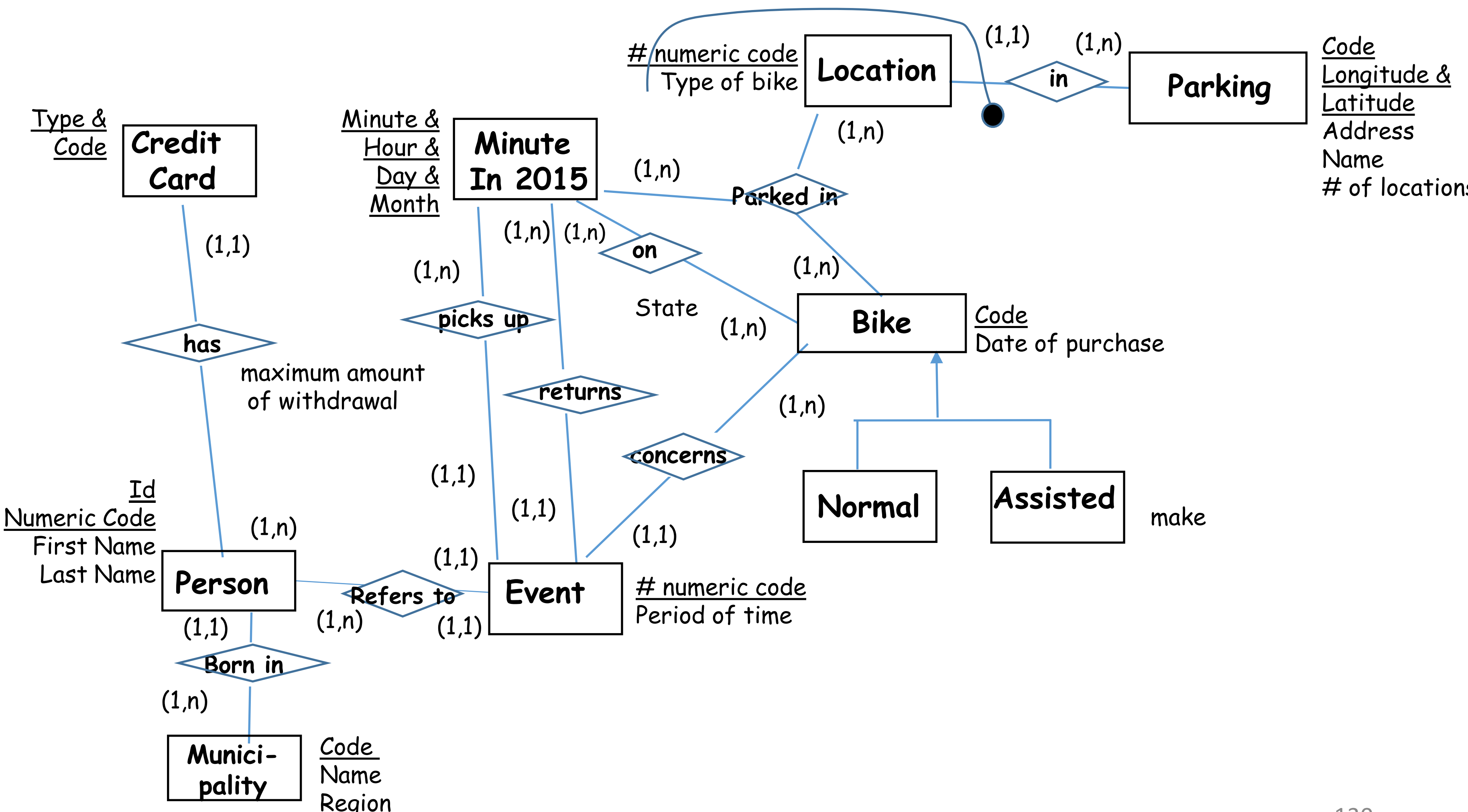
Diagrammatic readability

We first reconsider the diagram shown in previous slides, then we improve its diagrammatic readability by

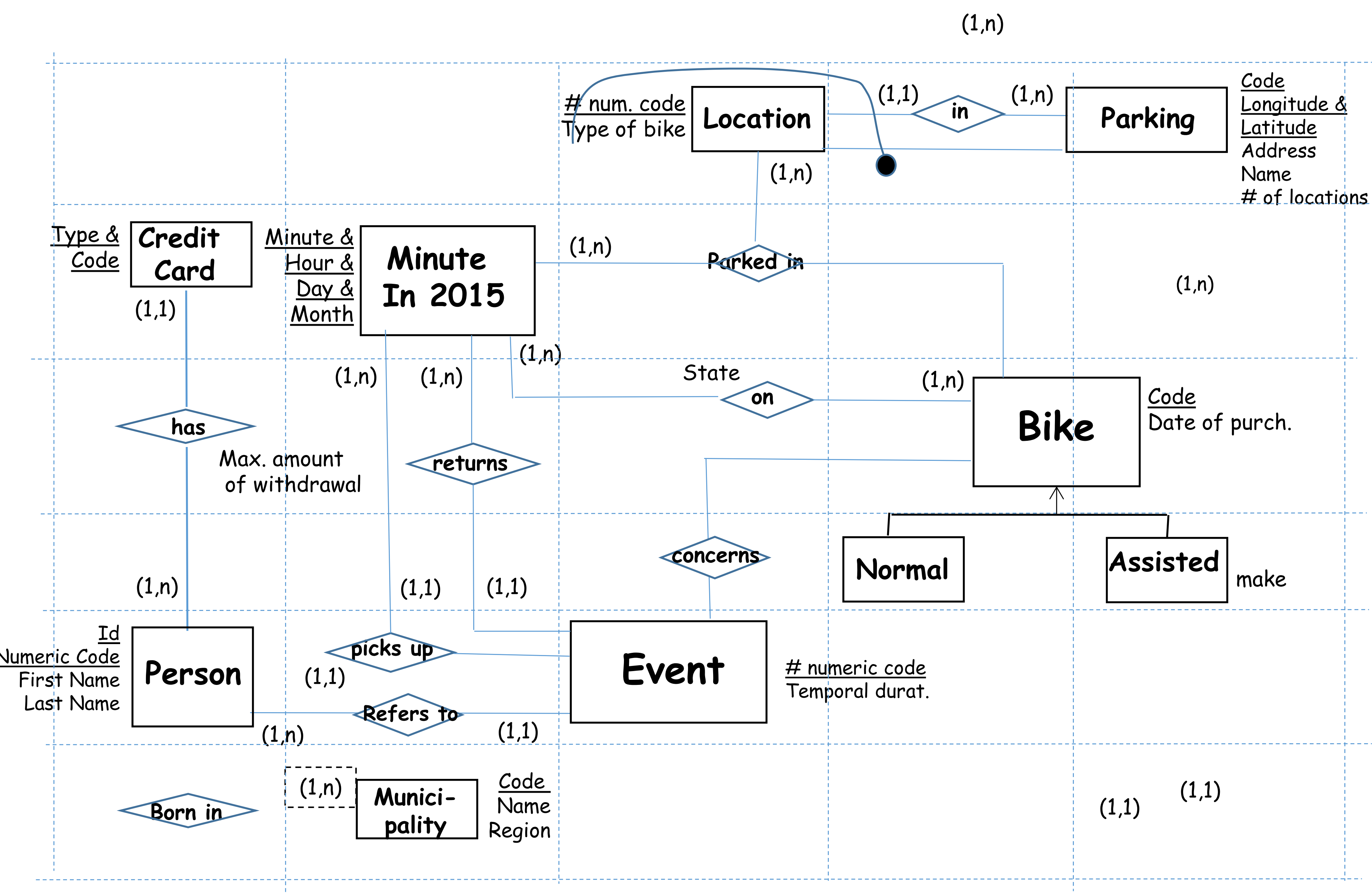
- framing the diagram in a grid, and
- representing all connecting lines as horizontal or vertical lines.

- Then we show two diagrams produced by students, one with good diagrammatic readability and the second with low readability

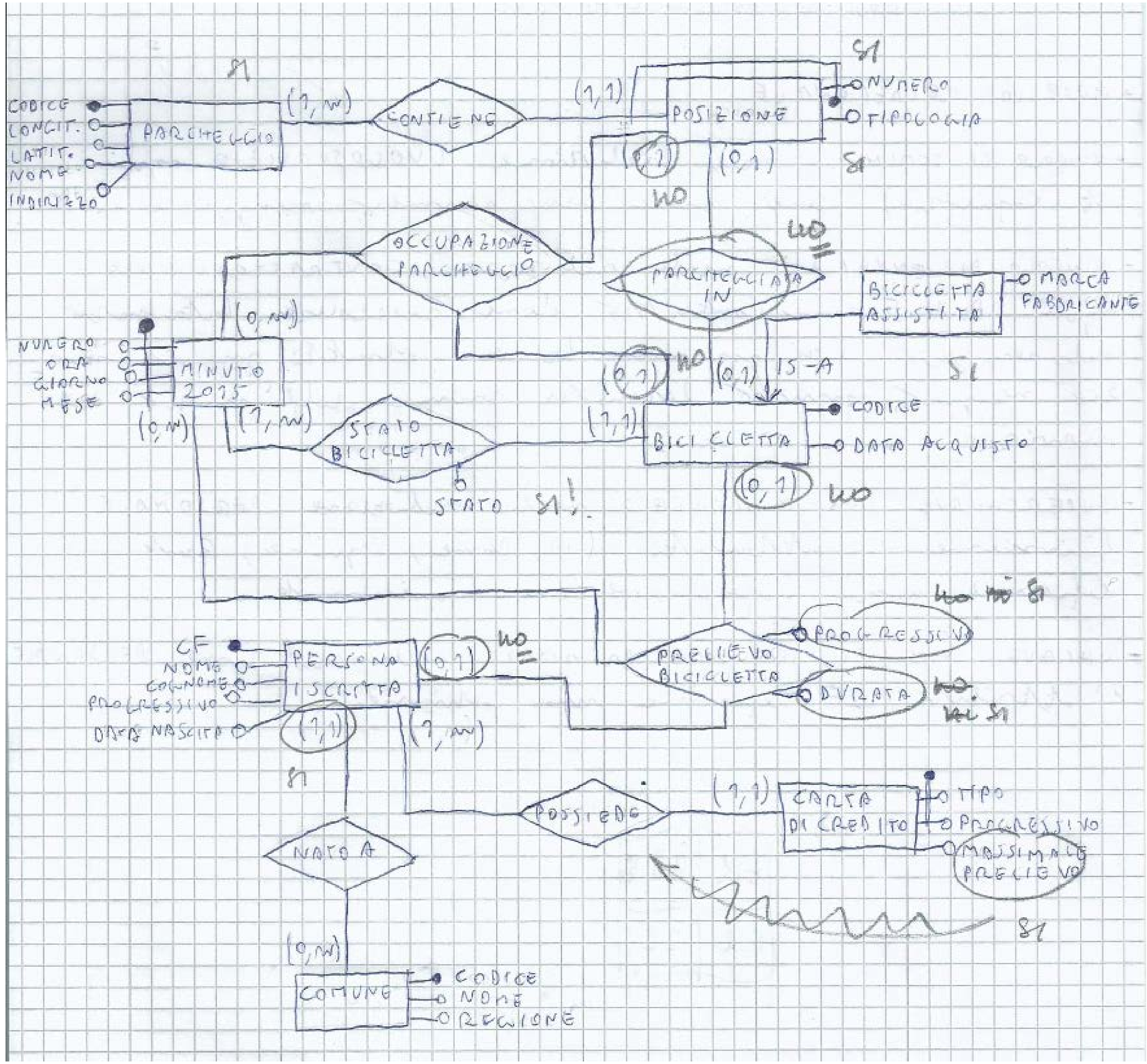
Diagram shown in previous slides



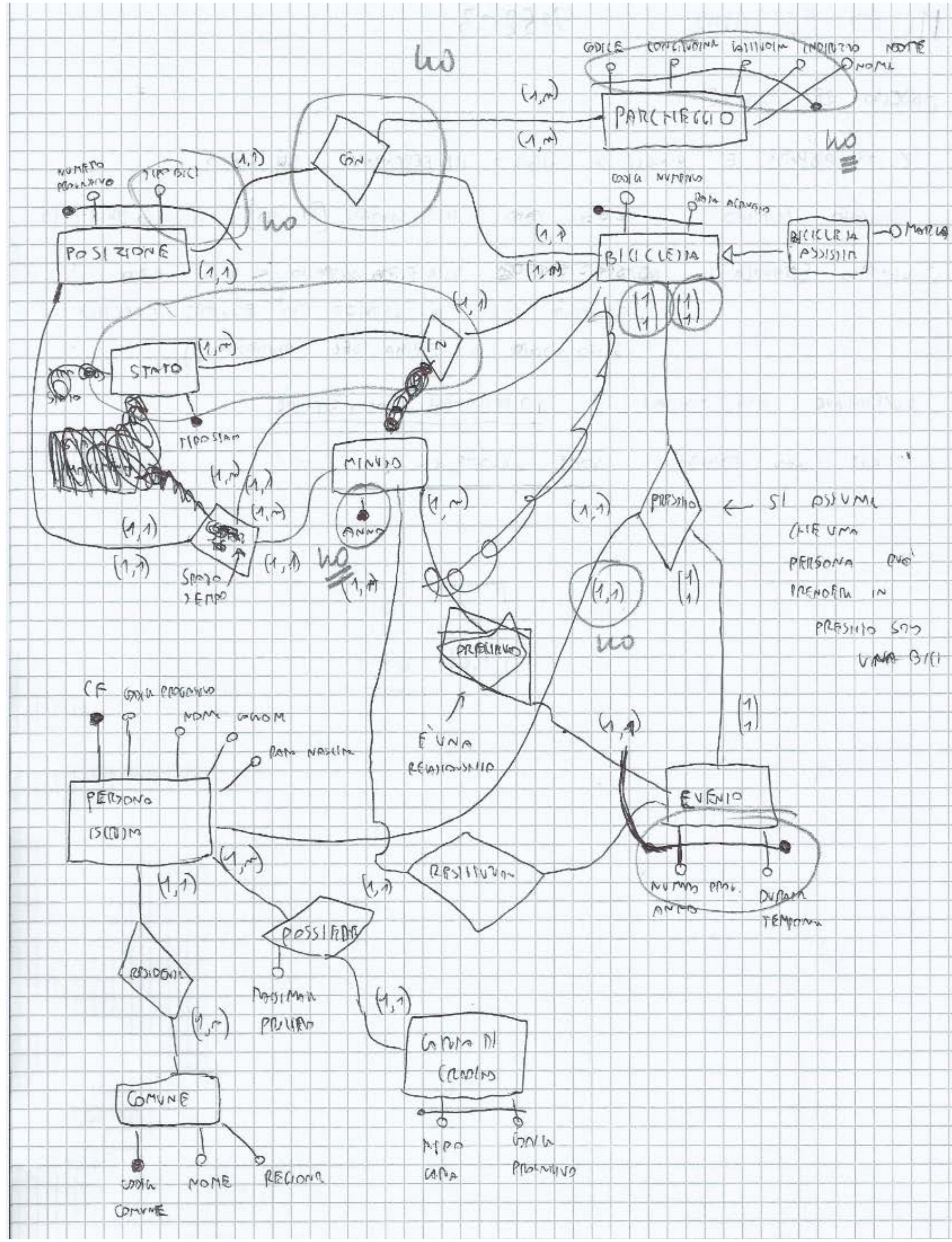
A diagram with enhanced readability



Good readability



Poor readability



7. Statistics on some types of errors and related involved qualities

Statistics on most frequent errors, involved concepts in the schema, quality that is violated and error rate

Type of error	Case in the schema	Not fullfilled quality	Wrong	Total	Error rate
Missing External identifier	Entity Position	CORRECTNESS W.R.T. REQUIREMENTS	43	110	39%
Wrong cardinalities: (1,n) instead of (1,1) in ternary relationship	Entity Event	CORRECTNESS W.R.T. REQUIREMENTS	34	60	57%
Missing or partially missing cardinalities	Whole schema	CORRECTNESS W.R.T. REQUIREMENTS	21	113	19%
Wrong cardinalities: (1,n) instead of (1,1) in binary relationship	Entity Credit Card in relat. with Person	CORRECTNESS W.R.T. REQUIREMENTS	23	115	20%
Wrong is-a or generalization	E.g. Minute is-s Year	CORRECTNESS W.R.T. THE MODEL	12	120	10%
Missing ternary relationship	Relationship Minute - Position - Bike	CORRECTNESS W.R.T. REQUIREMENTS	35	111	32%
Entity with unique attribute	Several entities	CORRECTNESS W.R.T. THE MODEL	21	113	19%
Identifier in a relationship	Several relationships	CORRECTNESS W.R.T. THE MODEL	9	120	8%
Generalization among bikes, parked and in motion bikes	Entity Bike Service	CORRECTNESS W.R.T. THE MODEL	3	120	3%
An entity represented with another entity with different meaning	Entity Event represented as Minute	CORRECTNESS W.R.T. REQUIREMENTS	33	102	32%
Missed double identifier in an entity	Entity Person	CORRECTNESS W.R.T. REQUIREMENTS	108	120	90%
Missed double identifier in an entity	Entity Parking lot	CORRECTNESS W.R.T. REQUIREMENTS	118	120	98%
Presence of an entity with unique instance in requirements	Entity Bike-Service	PERTINENCE	17	120	14%
Presence of an entity with unique instance in requirements	Entity (Organizing) Municipality	PERTINENCE	9	120	8%
Wrong external identifier	Several entities	CORRECTNESS W.R.T. REQUIREMENTS	28	120	23%
Wrong internal identifier (e.g. one attribute instead of two)	Entity Credit Card	CORRECTNESS W.R.T. REQUIREMENTS	38	106	36%
Presence of crossings in lines	Whole ER Diagram	READABILITY	24	120	20%

8. Methods to measure and compare the complexity of an assignment

On assignment complexity

- When conceiving a set of requirements for an assignment in conceptual database design, an instructor risks to produce requirements that are either too complex or too simple to manage.
- In this part of the presentation we:
  - Define a model that, given a set of requirements, allows measuring the complexity of the ER schema design process.
  - Provide a method for the instructor that allows, when conceiving a new assignment, to rate the complexity of the solution with reference to exam assignments arranged in the past, and modify suitabily the assignment to fit a reasonable complexity.

1. Measuring the complexity of an assignment

Measuring the complexity of an assignement - 1

- Having available a measure of complexity of an assignment, the instructor has the possiiblity to concieve more homogeneous assignments in the years; consequently, the grades of students attending the course in the years are comparatively more homogeneous.
- In the next slides we define a global indicator to measure the complexity of an assigniment. Such global indicator is based on complexity ratings associated to the different concepts defined in the ER model (e.g. an entity, an attribute, etc.)

Measuring the complexity of an assignment - 2

- A generic ER schema is made of entities, relationships, etc.
- When modeling an ER schema, the different concepts defined in the ER Model have unequal complexity of usage. E.g. for a student, identifying a ternary relationship is fairly more complex than identifying a binary relationship.
- We define now for each concept a complexity weight, based on the experience accumulated in the years. See next page.

Complexity weights of concepts in the ER Model

Entity Relationship Concept	Complexity weight (cw)
Entities	1
Binary Relationship	1,5
Ternary Relationship	2,5
Quaternary Relationship	4
Attribute of Entity	0,2
Attribute of Relationship	0,8
External Identifier	1,8
Generalizations	1,5
Is-a Relationship	1,5
Cardinalities (each pair)	0,5

Method to evaluate the complexity of an assignment in conceptual design

1. Given a set of requirements in conceptual design, find an ER schema that is a reasonable solution for requirements.
2. Count the number of occurrences of the different ER concepts in the schema.
3. The global complexity indicator is given by:

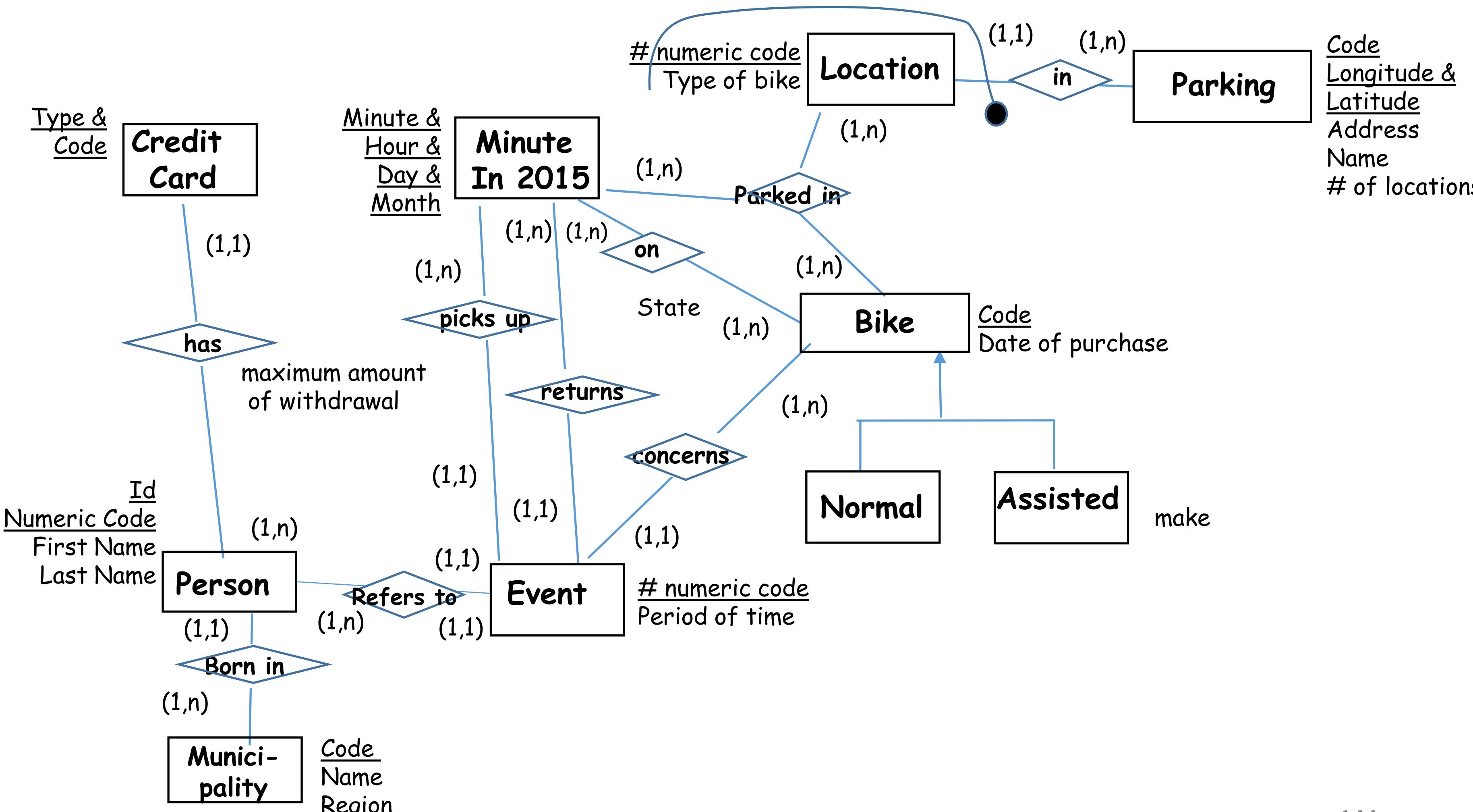
   Global indicator = $\sum nj * cwj$, where

   - nj is the number of occurrences of the generic concept j in the schema and
   - cwj is the complexity weight of concept j

Exercise

Evaluate the global complexity indicator for the ER schema in the next slide.

ER schema of the assignment



Occurrences of concepts in the ER schema

Entity Relationship Concept	Occurrences
Entities	10
Binary Relationship	6
Ternary Relationship	2
Quaternary Relationship	0
Attribute of Entity	28
Attribute of Relationship	1
External Identifier	1
Generalizations	1
Is-a Relationship	0
Cardinalities	18

Occurrences and complexity weights associated to concepts

Entity Rel. concept	Occurrences	Weight
Entities	10	1
Binary Relationship	6	1,5
Ternary Relationship	2	2,5
Quaternary Relationship	0	4
Attribute of Entity	28	0,2
Attribute of Relationship	1	0,8
External Identifier	1	1,8
Generalizations	1	1,5
Is-a Relationship	0	1,5
Cardinalities	18	0,5

Evaluation of the global complexity indicator

Concept	#	CF	Total
Entities	10	1	10
Binary Relationship	6	1,5	9
Ternary Relationship	2	2,5	5
Quaternary Rel.	0	3,5	0
Attribute of Entity	28	0,2	5,6
Attribute of Relationship	1	0,8	0,8
External Identifier	1	1,8	1,8
Generalizations	1	1,5	1,5
Is-a Relationship	0	1,5	0
Cardinalities	18	0,5	9
Global complexity indicator			42,7

Weights for Derived Indicators to measure the complexity of the assigment

Notice that we could enrich indicators with another set of derived indicators. In the following table we exemplify some of the with their values in the ER schema. We do not inglobe them at the moment in the method.

Derived Indicator	Weight
# of identifiers / # of entities with identifiers	1,25
# of relationships / # of entities	1,125
# of attributes in identifiers / # of attributes	0,5

2. Conceiving and fine tuning a new assignment, based on a complexity comparison

Method to conceive requirements for a new assignment

1. Conceive a set of requirements, and produce a reasonable solution, in terms of an ER schema ers.
2. Trace correspondences between sentences and words in requirements and related concepts in the schema ers.
3. Evaluate the global complexity indicator gci for the ER schema.
4. Compare the value of gci with the historical series of global indicators for assignments in previous years.
5. If gci is significantly higher than the average of previous indicators, then change the schema by removing concepts in the schema and suitably adapting sentences in requirements, driven by complexity weights of ER concepts.
6. If gci is significantly lower than the average of previous indicators, then change the schema adding concepts in the schema and suitably adapting sentences in requirements, driven by complexity weights of ER concepts.

Collaborative work

I will appreciate contributions from the readers of this text, oriented to enrich with comments and extensions the topics addressed. My e-mail is batini@disco.unimib.it

Appendix

Final segmentation

R1.1.1 – A Municipality (e.g. the municipality of Milan, Italy) aims at managing in a given year (e.g. in 2015) a bike sharing service. The service is organized in terms of parking lots, each one identifed by a code, a longitude, a latitude, an address, a name (e.g. Piazzale Susa), and by a number of locations for bycicles. Each location is identified by a progressive number in the parking and from the type of bike that can be hosted in the location, namely a. normal or b. assisted (namely, equipped with an electric motor). We assume that in a specific location of a parking lot it is always parked a bike of a given type.

R1.1.2 Each bike is identified by a numeric code, and a date of purchase; assisted bikes have a make, that may change among bikes (we do not represent the makes of normal bikes).

R1.2 When bikes are not used, they are parked in locations of parking lots. As a consequence, for each bike and for each minute in the year (e.g. minute 34 of hour 10 a.m. of march 7th) we want to represent the state of the bike, namely if it is parked or it is in motion. Furthermore, for each minute in the year and each location in a parking lot, we want to represent, separately, which bike is possibly parked in the location (e.g. in minute 29 of hour 11 a.m. of march 21th bike 73 is parked in location 15 of parking lot 28, while in the same minute bike 130 is parked in location 6 of parking lot 112). Of course, if in a given minute a bike is in motion, the corresponding location in a parking lot does not exist, and, therefore, will not be represented. It is strongly recommended to represent the concept of minute with an Entity.

R2 – In order to use bikes, users have to register themselves. A registered user is characterized by an identifier (e.g. a fiscal code, or a social security number), a progressive numeric code (first person registered, second person registered, etc.) a name, a surname, date of birth, municipality of birth, with code, name of municipality and region (we assume that municipalities are located in regions).

R3 – Everytime a user picks up and subsequently returns a bike, such an event (pick up and return of a bike by a user) has to be recorded. The event is identified by a progressive number in the year (e.g. first event in 2015, second event in 2015, etc.) and duration (e.g. 12 minutes, 47 minutes); such duration has to be explicitely represented in the data base. The minute of pick up and the minute of return have to be associated to the event, besides, of course, the bike used and the user involved.

R4 – One or more credit cards are associated to users, with type (e.g. Visa, Diners), progressive code in the card type, and daily maximum amount of withdrawal (that depends on the card and on the person).

You have to represent above requirements with an Entity Relationship Schema, in terms of entities, relationships, attributes, internal and external idetitifers, generalizations, is-a relationships, minimum and maximum cardinalities.