# System Analysis – Part 1

Relevant chapter in the core text: Chapter 10

Arish Siddiqui

# Learning objectives

- **After this lecture, you will be able to:**
  - define the importance of conducting the analysis phase to the overall success of the system;
  - choose appropriate techniques for analysing users’ requirements for an information system;
  - construct appropriate textual descriptions and diagrams to assist in summarising the requirements as an input to the design phase.

# Management issues

- **From a managerial perspective, this lecture addresses the following areas**:

    - Which different aspects of the system must be summarised in the requirements document?

    - Which diagramming tools are appropriate to summarise the operation of the existing and proposed systems?

# Systems analysis

**Systems analysis** is about finding out *what* the new system is to do, rather than *how*. There are two basic components to the analysis process:

*fact-finding* – an exercise needs to take place where all prospective users of the new system should contribute to determining requirements.

*documentation* – detailed systems design follows the analysis stage and it needs to be based on unambiguous documentation and diagrams from the analysis stage.

**Systems analysis**: The investigation of the business and user requirements of an information system. Fact-finding techniques are used to ascertain the user's needs and these are summarised using a requirements specification and a range of diagramming methods.

# Factors that will affect the type of analysis

The result of the 'make-or-buy decision':

| Make the software | Off-the Shelf |
|---|---|

**Application complexity**: A very complex system or one where there are linkages to other systems will need very careful analysis to define system and subsystem boundaries, and this will lead to use of more formal techniques when compared with a simple or stand-alone application.

**User versus corporate development**: User development does not lend itself to extensive use of formal analysis tools. However, basic analysis is required

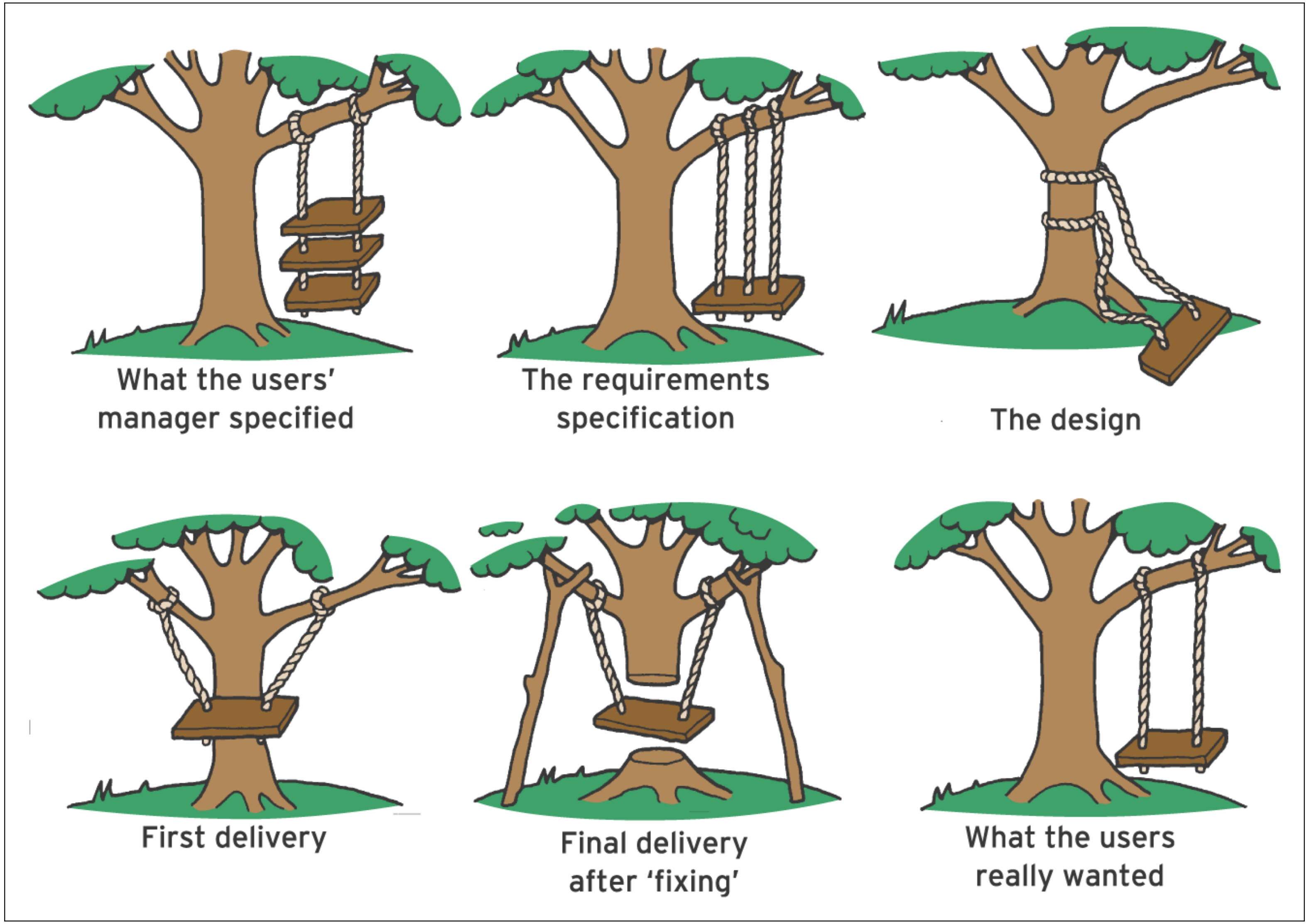



Figure 10.1 Varying interpretations of a user's requirements at different stages in a project

# Analysis technique – Interviewing

**Recommended practice**: A range of staff are interviewed using structured techniques to identify features and problems of the current system and required features of the future system.

**Closed questions**: Closed questions have a restricted choice of answers such as Yes/No or a range of opinions on a scale from 'strongly agree' to 'strongly disagree' (Lickert scale). **Approach is useful for quantitative** analysis of results.

**Open questions:** Asked to elicit opinions or ideas for the new system or identify commonly held views amongst staff. Open questions are not typically used for quantitative analysis, but **can be used to identify a common problem**.

# Interviewing benefits

- The ability to gather detailed information through a two-way dialogue;
- The ability for candid, honest responses to be made;
- valuable insights, especially when open questions are used;
- Responses that can easily be quantified, especially when closed questions are used;
- Being one of the best methods for gathering qualitative data such as opinions, and subjective descriptions of activities and problems.

# Interviewing disadvantages

The analyst's findings may be coloured by his or her perceptions of how other, similar, business operations work. Interviewers need to be especially skilled if this is to be avoided.

The development of a new information system may represent a threat through the risk of deskilling, redundancy or perceived inability to cope with change.

The interviewee may tell the analyst what he or she thinks should happen rather than what actually happens.

An interview at lower organisational levels may not yield as much information as some other methods if staff in this area are not capable of articulating with sufficient clarity.

# Analysis technique – Questionnaires

**Questionnaires** are used to obtain a range of opinion on requirements by targeting a range of staff.

They are open to misinterpretation unless carefully designed.

They should consist of both open and closed questions.

# Success factors – questionnaires

The questions will be framed by the analyst with a clear view of the information that is to be obtained from the completed questionnaires.

The target audience must be carefully considered – a questionnaire designed should match the level of work

The questionnaire should only contain branching (e.g. ‘if the answer to Question 3 was ‘No’, then go to Question 8’) if it is absolutely necessary – multiple branches create confusion and may lead to unusable responses.

# Success factors – Questionnaires (continued)

Questions should be simple and unambiguous so that the respondent does not have to guess what the analyst means.

Multiple-choice, **Lickert-scale-type questions** make the questionnaire easier to fill in and allow the results to be analysed more efficiently.

The questionnaire should contain the required return date and name of the person to whom the questionnaire should be returned.

# Questionnaires – problems

**the inability of respondents to go back to the analyst to seek clarification about what a question means;**

**difficulty in collating qualitative information, especially if the questionnaire contains open-ended questions;**

**the inability to use verbal and non-verbal signals from the respondent as a sign to ask other or different questions;**

**low response rates –**

these can be lower than 20 to 25 per cent when sent to other organisations or customers, which means that a large sample size is needed if the results are to carry any weight. Response rate is not such a problem with internal staff.

# Analysis technique – documentation review

- Uses information on existing systems such as user guides or requirements specifications together with paper or on-screen forms used to collect information such as sales order forms.

# Documentation benefits

If carried out at the beginning of a requirements analysis exercise, it will help provide the analyst with some background information relating to the area under consideration.

It may also help the analyst construct a framework for the remainder of the exercise and enable interviews to be conducted in a more effective way since the analyst has some idea of current business practices and procedures.

If document review is carried out later, it can be used to cross-check the actual business operations with what is supposed to happen.

# Documentation – benefits

THE ABILITY TO SEE HOW DOCUMENTS AND RECORDS ARE ACTUALLY HANDLED AND PROCESSED;

OBSERVATION MAY GIVE A GREATER INSIGHT INTO ACTUAL BUSINESS OPERATIONS THAN SIMPLE PAPER DOCUMENTATION;

IDENTIFICATION OF PARTICULAR OPERATIONS THAT TAKE A LONG TIME;

THE OPPORTUNITY TO SEE HOW DIFFERENT PROCESSES INTERACT WITH EACH OTHER, THUS GIVING THE ANALYST A *DYNAMIC* RATHER THAN A *STATIC* VIEW OF THE BUSINESS SITUATION UNDER INVESTIGATION.

# Documentation – problems

- There can be a large quantity of data for an analyst to process.
- it may take the analyst a long time to identify the documentation that is useful and that which can be ignored.

- Documentation is often out of date. If there is an old computerised system, it is quite possible that the documentation has not been changed for years.

## Analysis technique – Observation

- Useful for identifying inefficiencies in an existing way of working either with a computer-based or manual information system.
- Involves timing how long particular operations take and observing the method used to perform them.

# Analysis technique – Brainstorming

- **Analysis technique – brainstorming**: Brainstorming uses interaction within a group of staff to generate new ideas and discuss existing problems.

- It is the least structured of the fact-finding techniques.

# Requirements specification

- **Requirements specification**: The main output from the systems analysis stage.
  - Its main focus is a description of what all the functions of the software will be.
- Typically includes:
  - *data capture*
  - *preferred data capture methods*
  - *functional requirements*
  - *user interface layout*
  - *output requirements*.

## REQUIREMENTS CATALOGUE ENTRY

| Source | Owner | Requirement ID | Priority |
|---|---|---|---|
| Credit Control Clerk | Credit Control Manager | 5.9 | High |

**Functional Requirements**

Link Sales Order Processing system in with accounting package so that online credit checking is an automatic process when new orders are being processed.

**Non-functional requirements**

| Description | Target Value | Acceptable range | Comments |
|---|---|---|---|
| Response Time | Within 10 seconds | Within 20 seconds | |
| Service Hours | 08:30 to 18:00<br>Monday to Friday | | |
| Availability | 97.5% | Above 92.5% | |

**Benefits**

Will speed up order processing and enable account handlers to spend more time collecting cash rather than continually switching between computer systems when processing orders.

**Comments / suggested solutions**

Either provide a function key to perform the credit check function, or make it an automatic process when an order is entered. Do not allow order to be confirmed if credit check is failed.

**Related documents**

Required System DFD, process box 5.9

**Related requirements**

3.2. Improve cash collection process – more accurate sales ledger data
4.9. Reduce number of bad debts – link to improved aged debtors report

**Resolution**

3.2. Improve cash collection process – more accurate sales ledger data
4.9. Reduce number of bad debts – link to improved aged debtors report

Figure 10.2 Example of a requirements catalogue entry

# Different types of requirements

**FUNCTIONAL REQUIREMENTS**: CONSIST OF REQUIREMENTS THAT PERFORM THE ACTIVITIES THAT RUN THE BUSINESS. EXAMPLES INCLUDE UPDATING MASTER FILES

**NON-FUNCTIONAL REQUIREMENTS**: DEFINE THE PERFORMANCE LEVELS OF THE BUSINESS FUNCTIONS TO BE SUPPORTED. EXAMPLES INCLUDE ONLINE RESPONSE TIMES

**QUANTIFICATION OF REQUIREMENTS**: REFERS TO THE NEED FOR A MEASURE OF QUALITY IF THE BENEFITS ARE TO BE PROPERLY EVALUATED. EXAMPLES MIGHT INCLUDE REDUCING CUSTOMER COMPLAINTS BY 75%

# Data Modelling

Data Modelling – the structuring and organising of data and implement it in a database management system.

Managing large quantities of structured and unstructured data is a primary function of information systems.

Data Models describe structured data for storage in data management systems such as relational databases. They do not usually describe unstructured data

e.g. word processing documents, email messages, pictures, digital audio, and video.

# Data Modelling Diagrams

- Entity Relationship Diagram (ERD)
- Information Flow Diagram (IFD)
- Context Diagram (CD)
- Dataflow Diagram (DD)

# Entity Relationship (ER) Modelling

- ER Modeling is a process to help us understand and document the informational requirements of a system

  e.g. A college system needs to keep information about students; names, addresses, birth dates, courses, enrolment, grades, etc. - these things are documented in the model

# What is an ERD?

A PICTURE SHOWING THE INFORMATION CREATED, STORED, AND USED BY A BUSINESS SYSTEM.

ENTITIES GENERALLY REPRESENT PEOPLE, PLACES, AND THINGS OF INTEREST TO THE ORGANISATION.

LINES BETWEEN ENTITIES SHOW RELATIONSHIPS BETWEEN ENTITIES.

# Entity Relationship (ER) Diagrams

**Designers, programmers and end users view data in different way.**

**Important to get common understanding of how the enterprise operates.**

**ER Diagram - model for communication, should be free of ambiguities**

**Construction - top-down approach:**

firstly identify entities and relationships between the data

add more detail,  such as the information we want to hold about the entities and relationships

# Modelling

A *database* can be modeled as:

- a collection of entities,
- relationship among entities.

An **entity** is an object that exists and is distinguishable from other objects.

- Example: specific person, company, event, plant

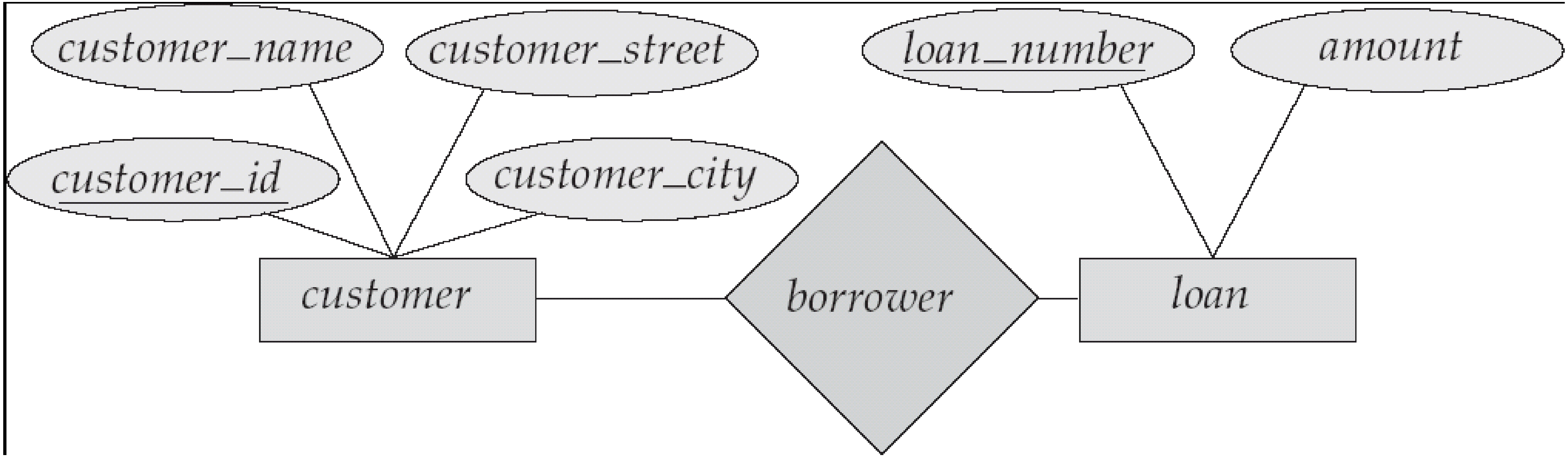
Entities have *attributes*

- Example: people have *names* and *addresses*

An **entity set** is a set of entities of the same type that share the same properties.

- Example: set of all persons, companies, trees, holidays

# ERD


customer_name
customer_street
loan_number
amount
customer_id
customer_city
customer
borrower
loan

# Using the ERD to Show Business Rules

ERD symbols can show when one example of an entity must exist for an example of another entity to exist

- A product must exist before it can be a sold item

ERD symbols can show when one example of an entity can be related to only one or many examples of another entity

- One doctor can have many patients, each patient may have only one primary doctor

# ER Diagrams - Attributes

- Information captured about an entity
- Only those used by the organisation should be included in the model
- Attribute names are nouns
- Sometimes entity name is added at the beginning of the attribute name

# ER Diagrams - Attributes

- Attribute - Property of an entity type of relationship type
- Properties of an entity we want to record
- Example: Employee number, name, address, telephone number
- The attributes could be:
    - EMP_NO, EMP_NAME, EMP_ADDRESS, EMP_TELNO

# ER Diagrams - Relationships

- Associations between entities
- Connected by a line
- Given active verb names
  - One verb can describe relationship in both directions
  - Two verbs can describe each relationship

# ER Diagrams - Relationships

- Entity types - A group of objects with in the real world with the same properties. They may bear relationship to one another
- Example: Employee works in Department
- Recording: Which Department an Employee is in
- The relationship could be:
  - Employee Works in Department
- Relationship occurrence – a uniquely identifiable association, which includes one occurrence from each participating entity type

# Relation Types

- Relation between two entities

  Emp and Dept

- More than one relation between entities

  Lecturer and Student

  Teaches – Personal Tutor

- Relationship with itself

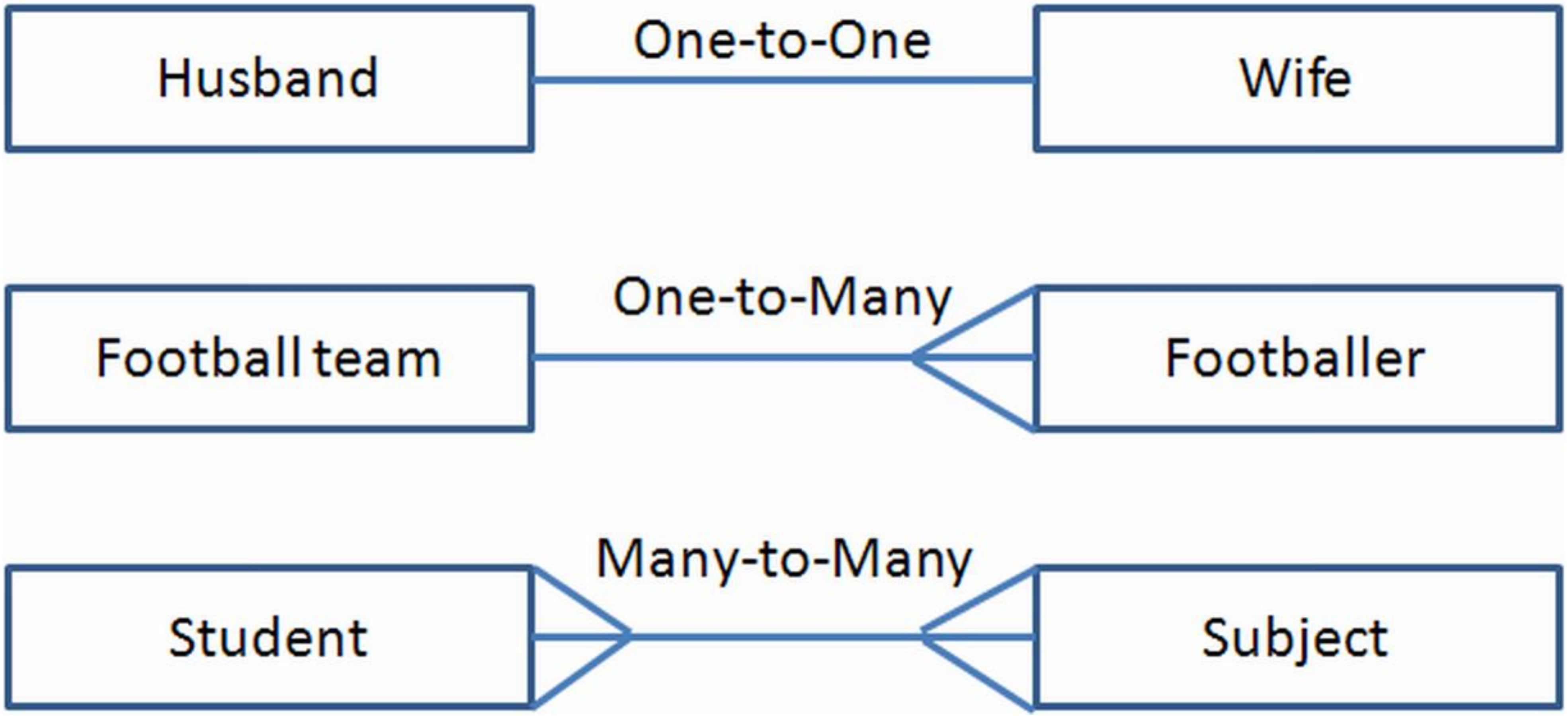
  Called Involuted

  Part made of parts

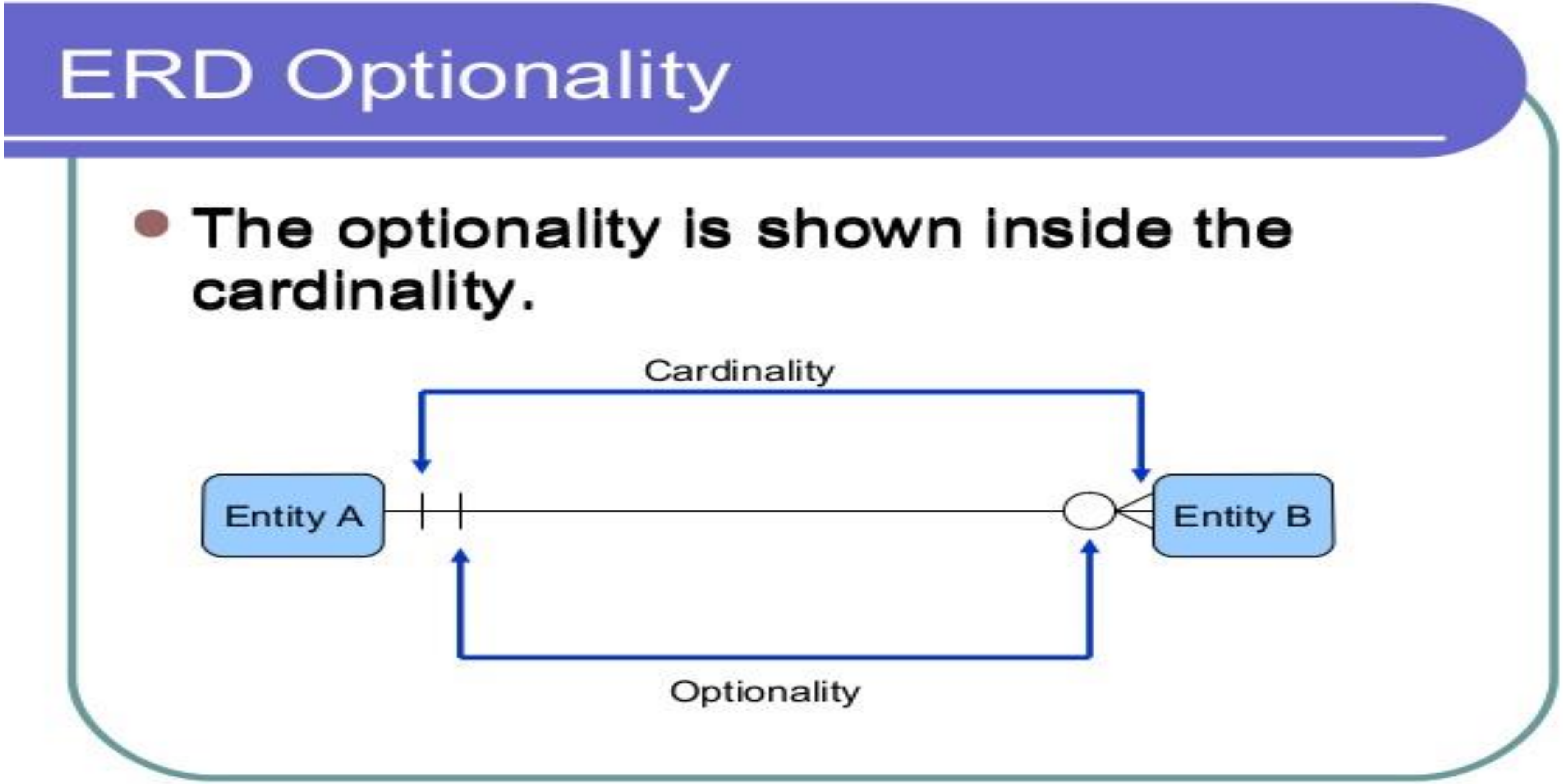
# Degree of Relationship

- Determines the number of occurrences from one entity to another
- Example: Each Dept there are a number of Employees that work in it

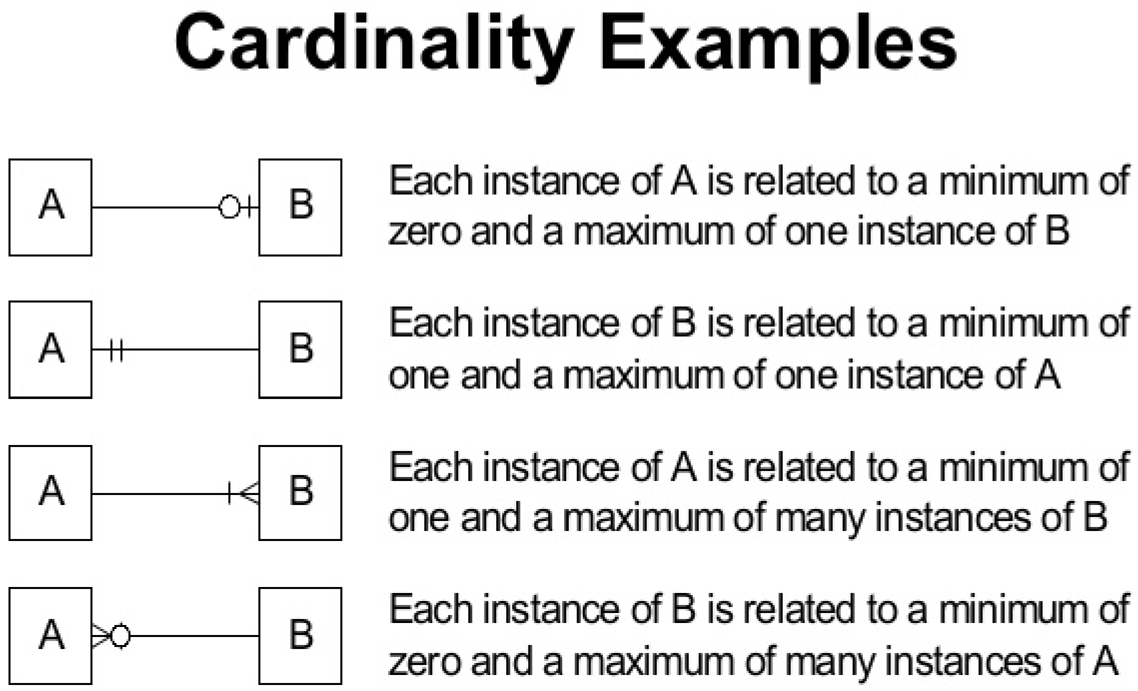
# Three Types of Degree



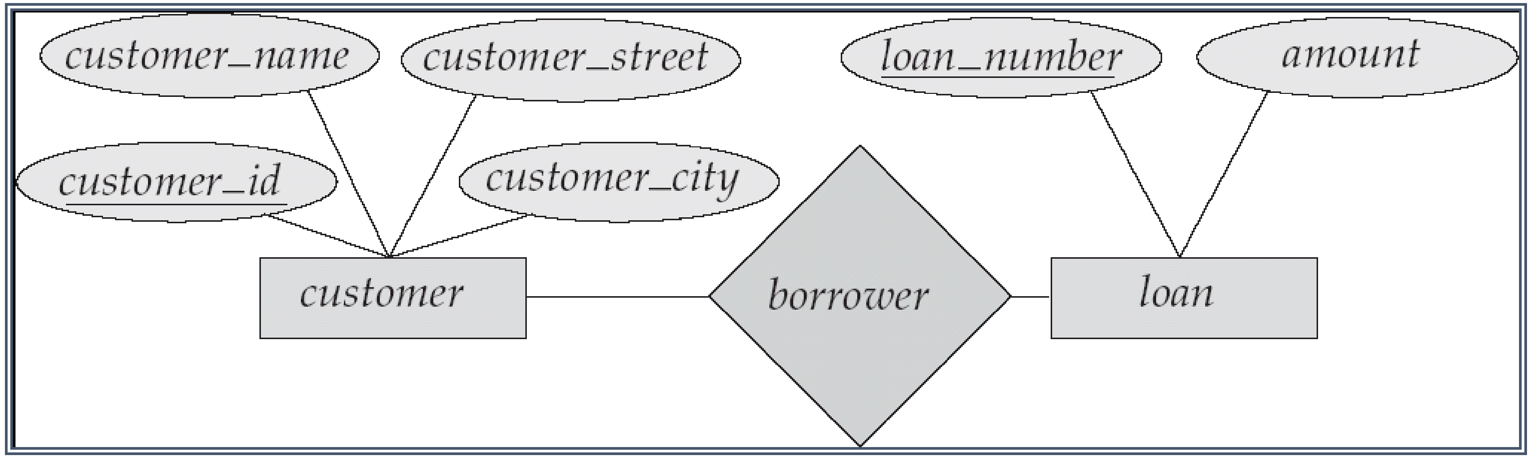
# Cardinality & Optionality



# Mapping Cardinality Constraints

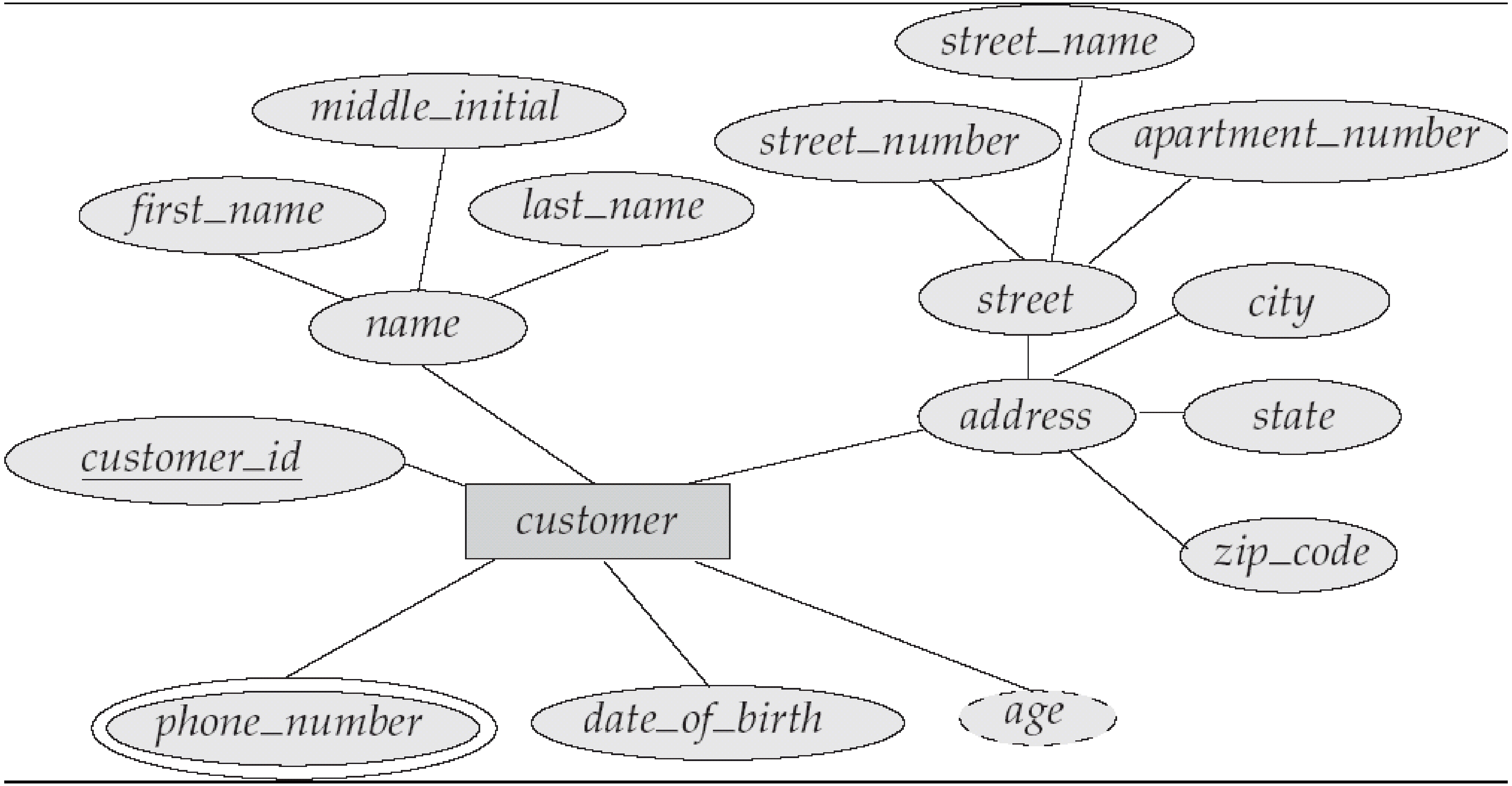


# E-R Diagrams



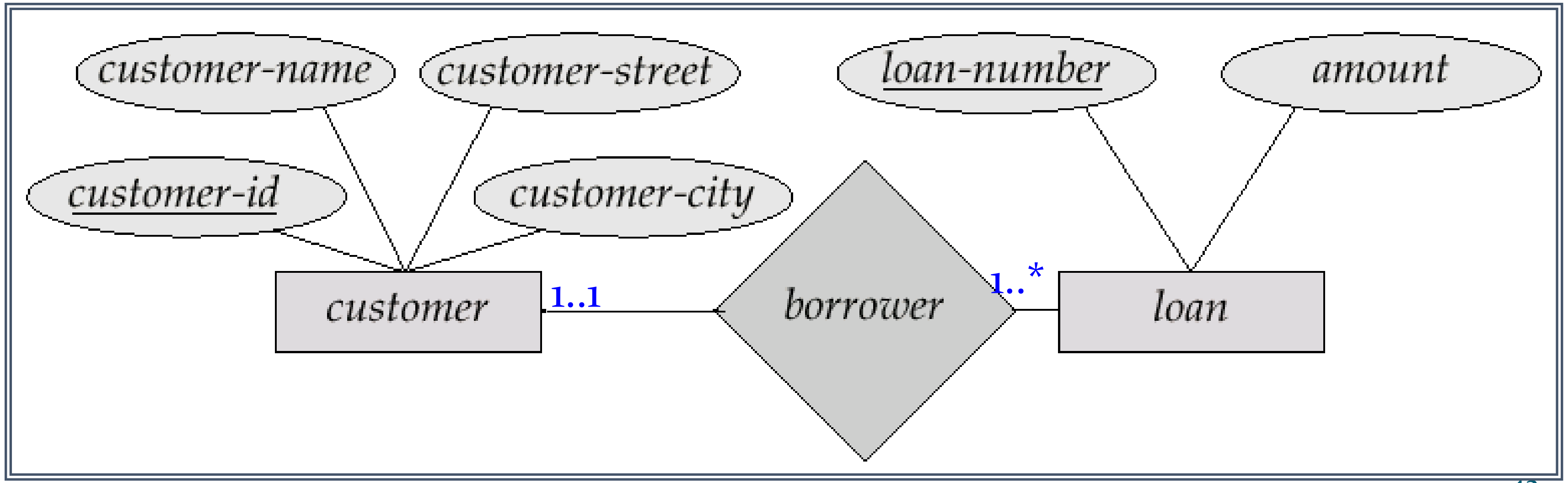

- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Lines link attributes to entity sets and entity sets to relationship sets.
- Ellipses represent attributes
  - Double ellipses represent multivalued attributes.
  - Dashed ellipses denote derived attributes.
- Underline indicates primary key attributes

# E-R Diagram with Attributes



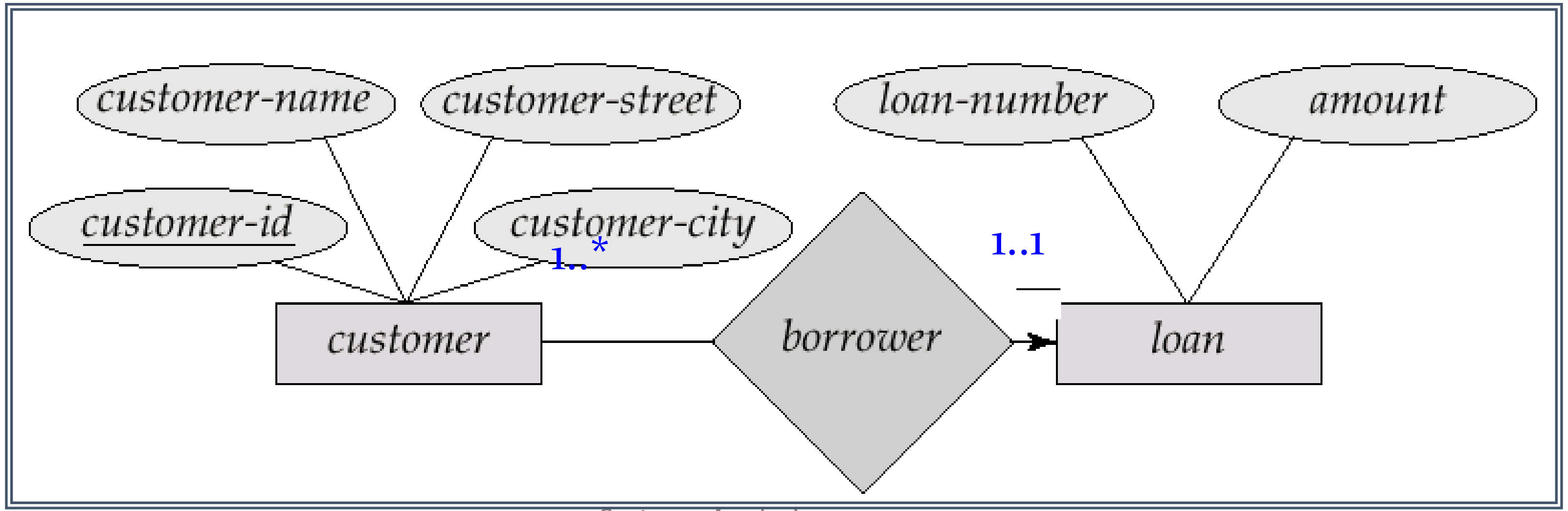
# One-To-Many Relationship

- In the one-to-many relationship a loan is associated with at most one customer via *borrower*, a customer is associated with several loans via *borrower*



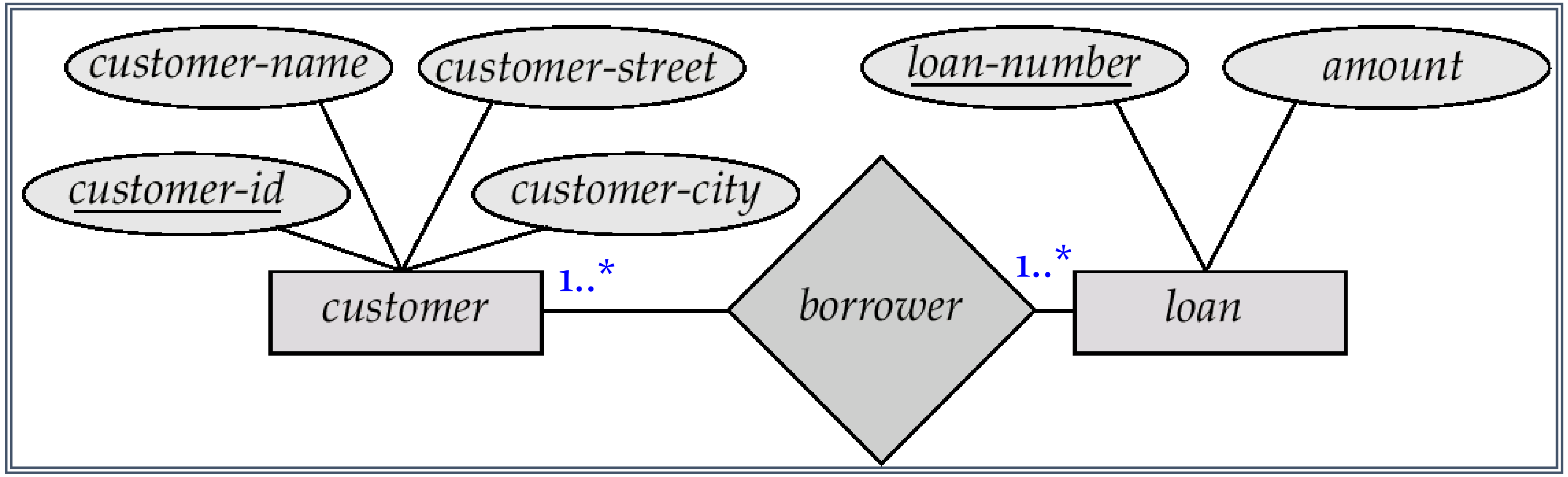
# Many-To-One Relationships

- In a many-to-one relationship a loan is associated with several customers via *borrower*, a customer is associated with at most one loan via *borrower*



# Many-To-Many Relationship

- A customer is associated with several loans via borrower
- A loan is associated with several customers via borrower



# Developing an ER Diagram

- Step 1 Identify Entities
- Step 2 Work out relationships
- Step 3 Identify attributes
- Step 4 Identify cardinality
- Step 5 Identify KEY attributes

(Resolve assumptions)

To be continued ……..

# Questions ?