Database technology

Lecture 1: Introduction
Lecture 3: Enhanced entity-relationship (EER) modelling

Jose M. Peña
jose.m.pena@liu.se

Database applications

- Traditional applications:
  - Numeric and textual databases
- More recent applications:
  - Bioinformatics
  - Multimedia databases
  - Geographic information systems (GIS)
  - Data warehouses
  - Real-time and active databases
  - Many other applications

What is a database?

- A database represents some aspect of the real world, i.e. a mini world.
- A database consists of a logical coherent collection of data with an underlying meaning.
- A database is designed, built and filled with data with respect to an underlying purpose.

Example of a database

- Mini-world for the example:
  - Part of a UNIVERSITY environment.
- Some mini-world *entities*:
  - STUDENTs
  - COURSEs
  - SECTIONs (of COURSEs)
  - (academic) DEPARTMENTs
  - INSTRUCTORs
- Some mini-world *relationships*:
  - SECTIONs *are of specific* COURSEs.
  - STUDENTs *take* SECTIONs.
  - COURSEs *have prerequisite* COURSEs.
  - INSTRUCTORs *teach* SECTIONs.
  - COURSEs *are offered by* DEPARTMENTs.
  - STUDENTs *major in* DEPARTMENTs.

Example of a database

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Basic definitions

- Database:
  - A collection of related data.
- Data:
  - Known facts that can be recorded and have an implicit meaning.
- Mini-world:
  - Some part of the real world about which data is stored in a database. For example, student grades and transcripts at a university.
- Database management system (DBMS):
  - A software package/ system to facilitate the creation and maintenance of a computerized database.
- Database system:
  - The DBMS software together with the data itself. Sometimes, the applications are also included.

Database system environment

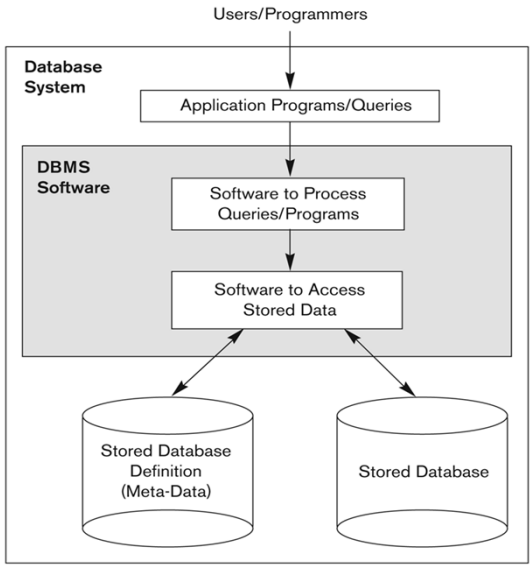


Figure 1.1
A simplified database system environment.

Typical DBMS functionality

- Define a particular database in terms of its data types, structures, and constraints.
- Construct or load the initial database contents on a secondary storage medium.
- Manipulate the database:
  - Retrieval: Querying, generating reports.
  - Modification: Insertions, deletions and updates to its content.
  - Accessing the database through web applications.
- Process and share by a set of concurrent users and application programs – yet, keeping all data valid and consistent.

Main characteristics of the database approach

- Self-describing nature of a database system:
  - A DBMS catalog stores the description of a particular database (e.g. data structures, types, and constraints).
  - The description is called meta-data.
  - This allows the DBMS software to work with different database applications.
- Insulation between programs and data:
  - Called program-data independence.
  - Allows changing data structures and storage organization without having to change the DBMS access programs.
- Data abstraction:
  - A data model is used to hide storage details and present the users with a conceptual view of the database.
  - Programs refer to the data model constructs rather than data storage details.
- Support of multiple views of the data:
  - Each user may see a different view of the database, which describes only the data of interest to that user.

Database design process

- Two main activities:
  - Database design.
  - Applications design.
- Focus in this course on database design.
  - To design the conceptual, logical and physical model for a database application.
- Applications design focuses on the programs and interfaces that access the database.
  - Generally considered part of software engineering.

Database design process

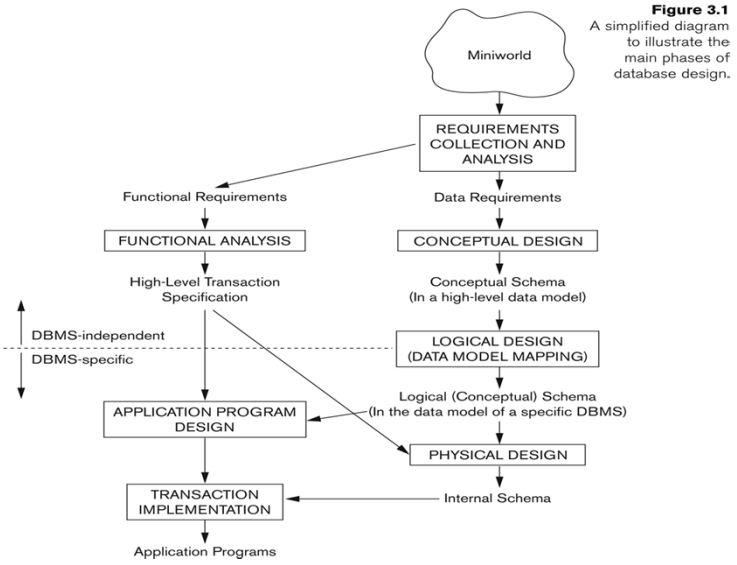
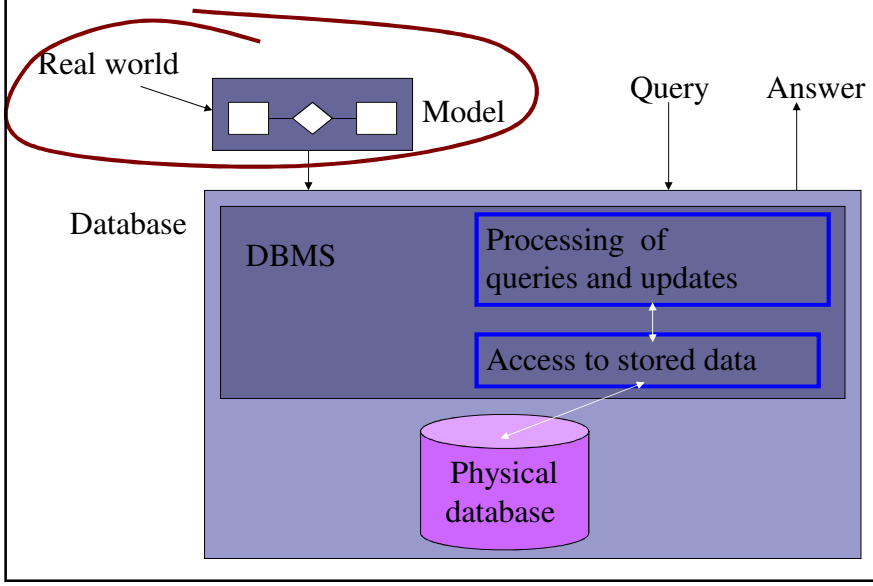


Figure 3.1
A simplified diagram to illustrate the main phases of database design.

Course goals

- Understand the important concepts within databases and database terminology.
- Design a database for a given application.
  - EER-modelling.
- Design and use a relational database.
  - Concept of relations.
  - Use MySQL.
  - Decipher a new relational database system.
- Theoretical foundations behind relational databases.
  - Normalization.
- Understand how the database is stored on the computer.
  - Basic technology, file structures, indexing, etc.
  - Impact on database performance.
  - B-trees, hashing, etc.
- Understand how databases can support multiple users.
  - What problems occur.
  - Views.
  - Transactions.
  - Serialisation.
- Understand how persistency can be guaranteed.
  - Recovery.

Overview



13

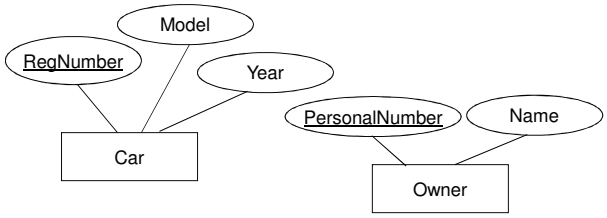
Entity-relationship (ER) model

- High-level conceptual data model.
  - An overview of the database.
  - Easy to discuss with non-database experts.
  - Easy to translate to data model of DBMS.
- ER diagram.

14

Entity and entity type

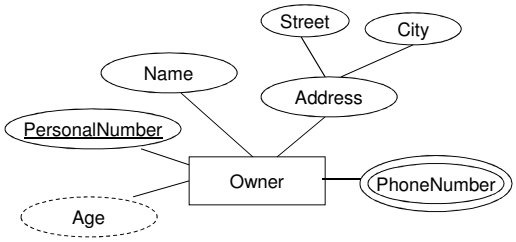
- Entity: A ”thing” in the real world with an independent existence.
- Attributes: Properties that describes an entity.
- Entity type: A collection of entities that have the same set of attributes.



15

Attributes

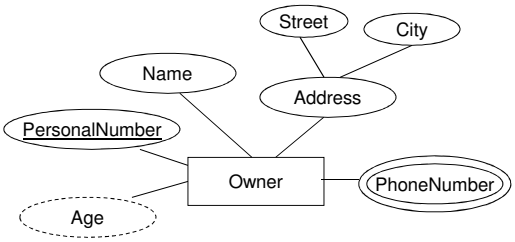
- Simple vs. composite.
- Single-valued vs. multivalued.
- Stored vs. derived.



16

Constraints on attributes

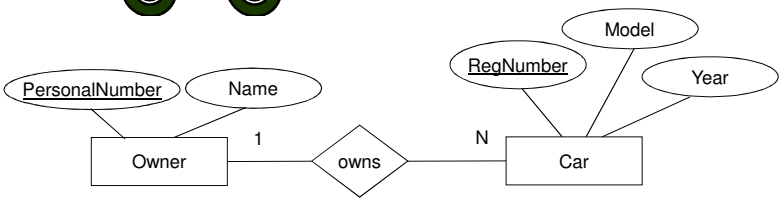
- Value sets (domains) of attributes.
- Key attributes.



17

Relationship type

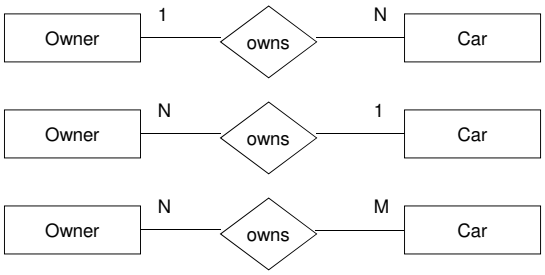
- Relationship type: Association among entity types.



18

Constraints on relationship types

- Cardinality ratio: **Maximum** number of relationships an entity can participate in.
- Possible cardinality ratio: 1:1, 1: N, N:1, and N:M

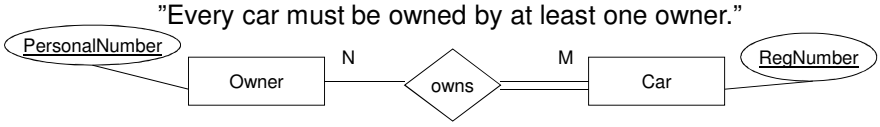


19

Constraints on relationship types

- Participant constraint.
  - Total participation: Every entity participates in **at least** one relationship with another entity.

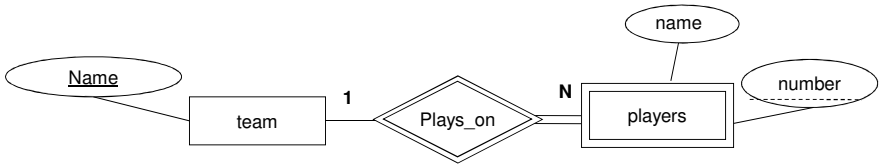
”Every car must be owned by at least one owner.”



20

Constraints on relationship types

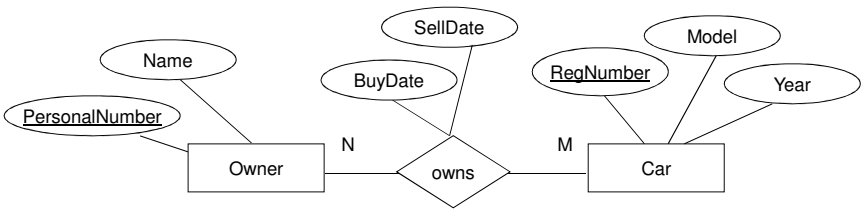
- Weak entity types: They do not have key attibutes of their own.
- A weak entity can be identified uniquely by being related to another entity (together with its own attributes).



21

Attributes of relationship types

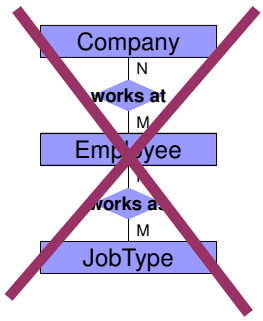
”Store information on who owned which car and during which period of time”

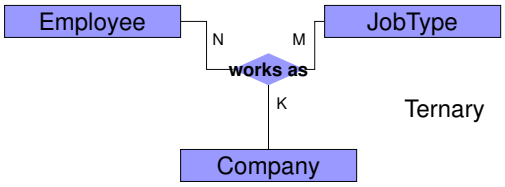


22

N-ary relationships

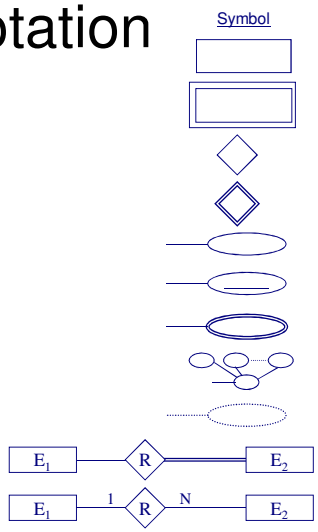
- Example. A person works as an engineer at one company and as a gym instructor at another company.





23

ER Notation



Meaning
ENTITY TYPE
WEAK ENTITY TYPE
RELATIONSHIP TYPE
IDENTIFYING RELATIONSHIP TYPE
ATTRIBUTE
KEY ATTRIBUTE
MULTIVALUED ATTRIBUTE
COMPOSITE ATTRIBUTE
DERIVED ATTRIBUTE
TOTAL PARTICIPATION OF E_2 IN R
CARDINALITY RATIO 1:N FOR E_1:E_2 IN R

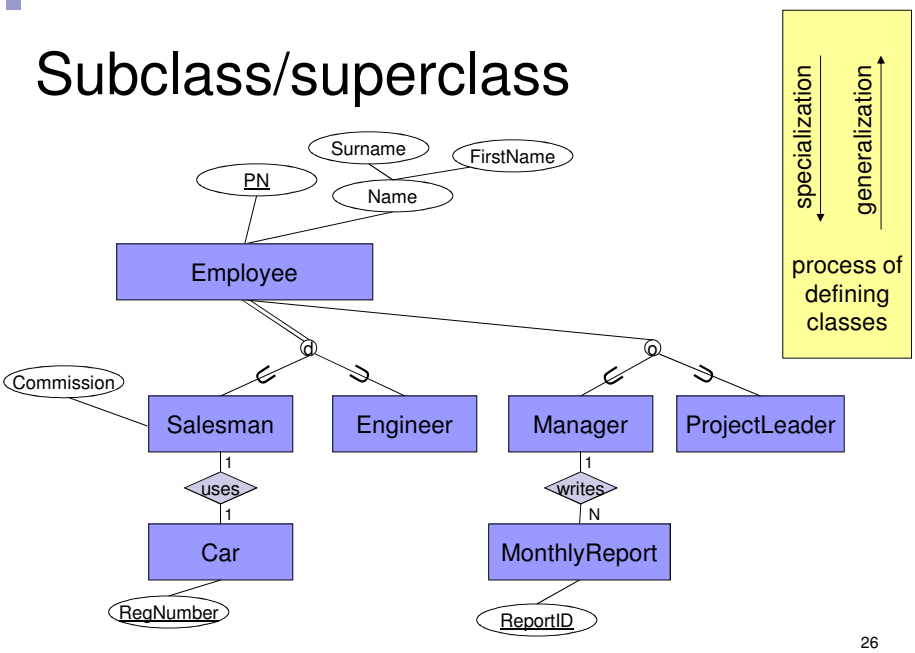
24

Enhanced ER (EER) model

- Why more? To comply with more complex data requirements.
  - Example. Only some employees can use a company car, only managers have to write a monthly report, but all employees have assigned personal number, salary account and a place in the office.
- Subclass/superclass, specialization/generalization, union/category, and attribute and relationship inheritance.

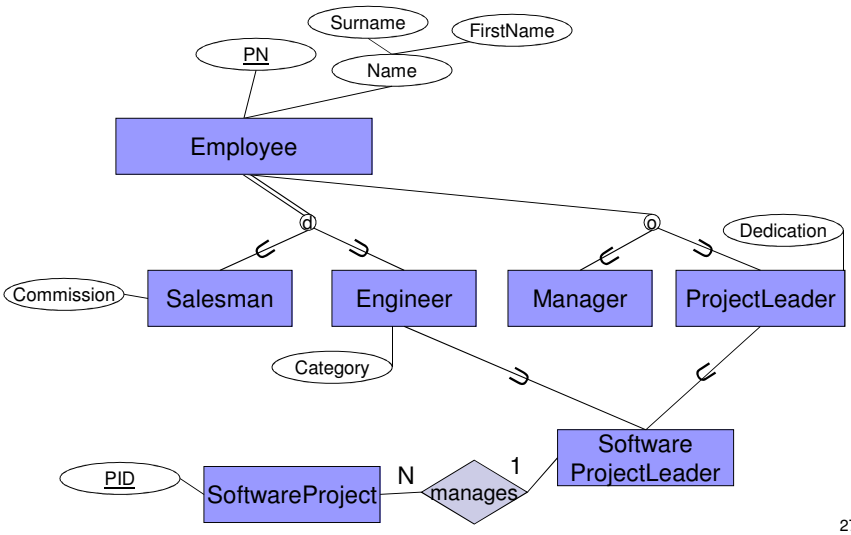
25

Subclass/superclass



26

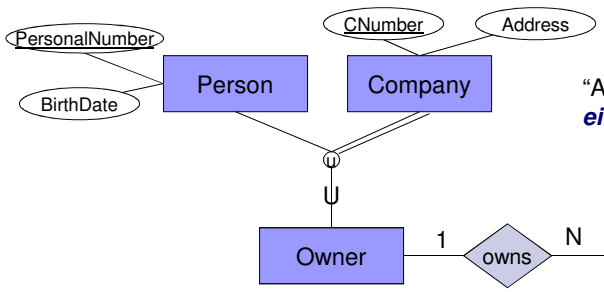
Single vs. multiple inheritance



27

Union/category

- A UNION subclass represents a collection of entities that is a subset of the UNION of the entities of the superclasses.



"An owner of a car is ***either*** a person ***or*** a company."

28

Example

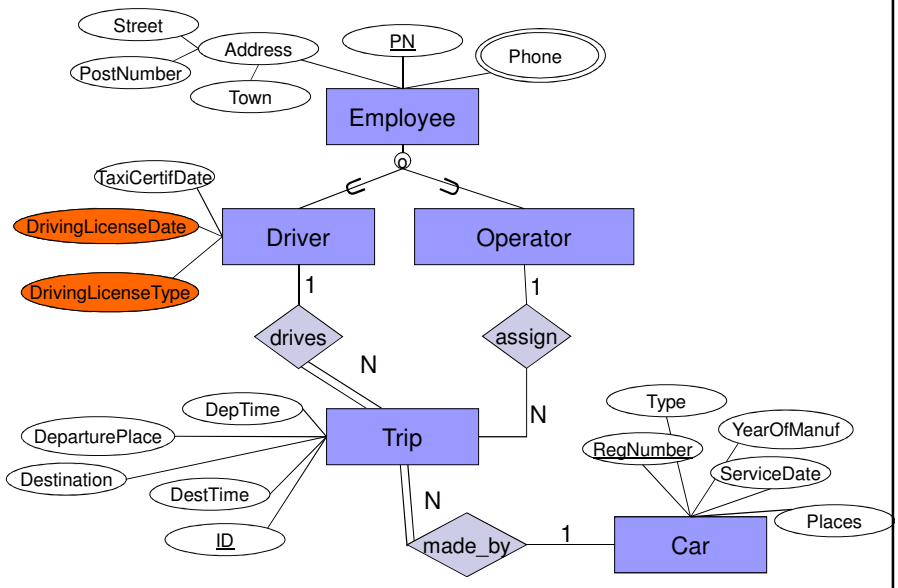
A taxi company needs to model their activities.

There are two types of employees in the company: drivers and operators. For drivers it is interesting to know the date of issue and type of the driving license, and the date of issue of the taxi driver's certificate. For all employees it is interesting to know their personal number, address and the available phone numbers.

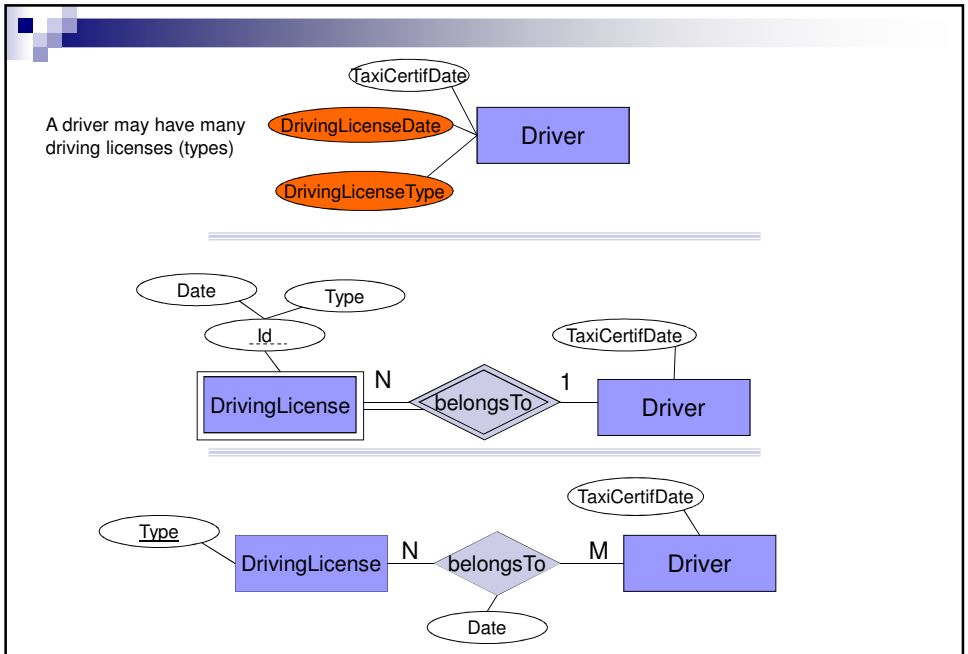
The company owns a number of cars. For each car there is a need to know its type, year of manufacturing, number of places in the car and date of the last service.

The company wants to have a record of car trips (körningar). A taxi may be picked on a street or ordered through an operator who assigns the order to a certain driver and a car. Departure and destination addresses together with times should also be recorded.

29



30



31

Summary

- Entity-relationship (ER) model: A graphical way to model the world.
- Main concepts: Entity type, relationship type, and attributes.
- Different types of constraints.
- Enhanced ER model.

32