DataBases - tutorials

Marek Kozlowski

m.kozlowski@ii.pw.edu.pl

markozlow@gmail.com

Info

The course introduces students into the area of databases. Basic concepts and notions of database management systems (DBMS) and their architecture will be presented. Respective data structures, as well as DDL and DML language constructions, will be discussed. Mathematical fundamentals of relational databases will be presented. Principles of designing databases will be outlined. Relational query languages, such as QBE, SQL and PL/SQL, will be presented.

Tutorials@Labs

Tutors will be organised collectively with the Labs as a „One Activity” (the tutorials and labs are related to each other), according to that there is a one unified agenda, of course some changes can be applied by Lab Supervisors

The default tools suggested for Labs

Requirements for ERD, Relational Schema: draw.io (a web-based tool for a data modelling)

Requirements for Physical Model (DDL scripts creating physical model), SQL (Selects, DML commands), Db objects as Functions, Stored Procedures, Indexes: download and install SQL DEVELOPER (https://www.oracle.com/pl/tools/downloads/sqldev-v192-downloads.html)

How to configure SQL Developer tool: http://galera.ii.pw.edu.pl/oracle_en/ (you need to get from administrative staff a username and password to a database, the access to a database is fully remotely without any restrictions)

I organized the tutorial agenda into the sequence of tasks, of course some of them can be joined or adjusted by Lab Supervisors. Each task, explained by examples on tutorials, has its own lab scenario and output, the labs and their outputs are scored by Lab Supervisors.

Tutorials@Labs - calendar

Kalendarz semestru: 22Z

Poprzedni semestr | Semestr 22Z | Następny semestr

2022/2023	Październik					Listopad				Grudzień				Styczeń					Luty	
Poniedziałek		3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	6	13
Wtorek		4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	31	7	14
Środa		5	**12**	19	26	2	9Pi	16	23	30	7	14	21	28	4	11	18	25	1	8	15
Czwartek		6	13	20	27	3	10	17	24	1	8	15	22	29	5Pi	12	19	26	2	9	16
Piątek		7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	3	10	17
Sobota	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	4	11	18
Niedziela	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	5	12	19
		N/P	P	N	P	N	N/P	P	N	P	N	P	N		P	N	P	N			

Tasks #1-3

Task#1: Create/Figure out a idea for your personal data base that you work on during the Labs (e.g. database for hospital, renting car company, university schedule planning)

Output: 1-2 paragraphs textual description of the data base idea, each entity that you plan to persist should be precisely described with the attributes, also some basic business activities that you would like to model in a database

Task#2: ERD - Entity Relationship Diagram - create ERD diagram for your database project

Default Output: the ERD diagram in draw.io created or using other tools

Task#3: Relational Schema plus Normalization Forms - convert ERD into Relational schema and check the 3 normalisation forms

Default Output: the Relational schema in draw.io created, describe clearly that Normalization Forms are met in the schema

Tasks #4-7

Task#4: Physical Model - DDL (Data Definition Language) - write DDLl sql script that creates database in Oracle environment

Default Output: ddl sql script

Task#5: SQL - DML (Data Manipulation Language) - write DML sql script that populates the tables with a random data

Default Output: dml sql script

Task#6: SQL - queries - write few select queries using inner/outer joins, group by+having, order by, subquries

Default Output: selects sql script

Task#7: Oracle Database Objects - write few commands creating example of function, stored procedure, index

Default Output: commands creating those objects

Conceptualization phase – the current one

Task#1: Create/Figure out a idea for your personal data base that you work on during the Labs (e.g. database for hospital, renting car company, university schedule planning)

Output: usually a 1-2 paragraphs textual description of the data base idea, each entity that you plan to persist should be precisely described with the attributes, also some basic business activities that you would like to model in a database

ERD Diagrams - introduction

- An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.
- By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.
- ER diagrams are used to sketch out the design of a database.

ERD Diagrams – Best Practices

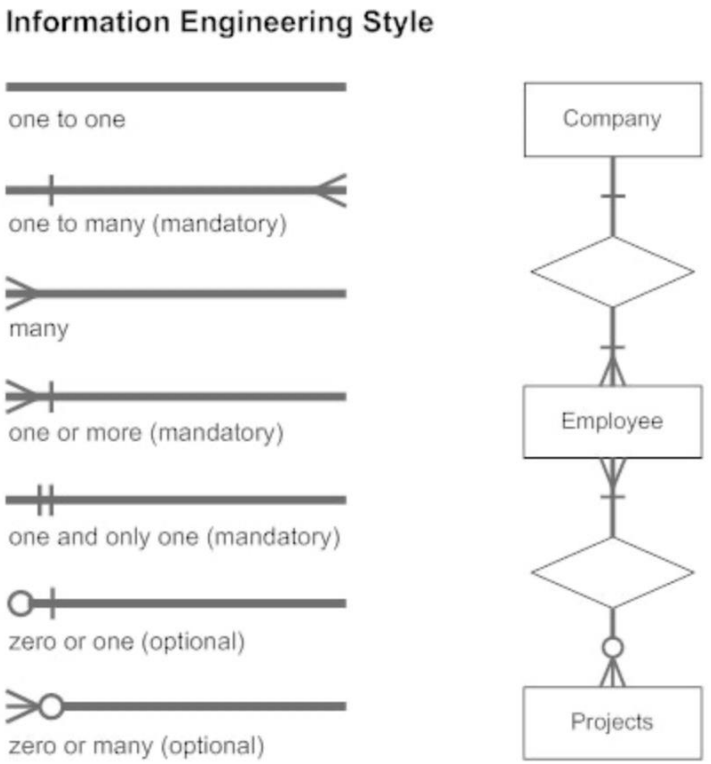
Identify the entities. The first step in making an ERD is to identify all of the entities you will use. An entity is nothing more than a rectangle with a description of something that your system stores information about. This could be a customer, a manager, an invoice, a schedule, etc. Draw a rectangle for each entity you can think of on your page. Keep them spaced out a bit.

Add attributes. Any key attributes of entities should be added using oval-shaped symbols.

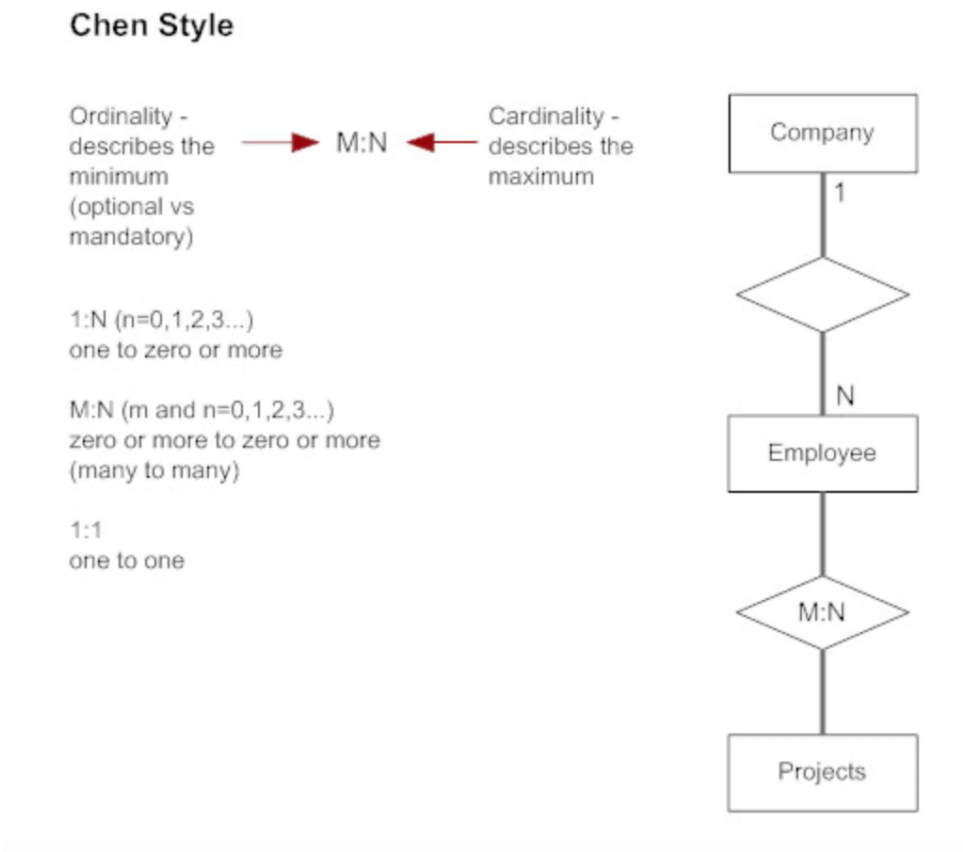
Identify&Desribe relationships. Look at two entities, are they related? If so draw a solid line connecting the two entities, in this step you are also deciding about cardinality, optionality, name od relationship

ERD Diagrams

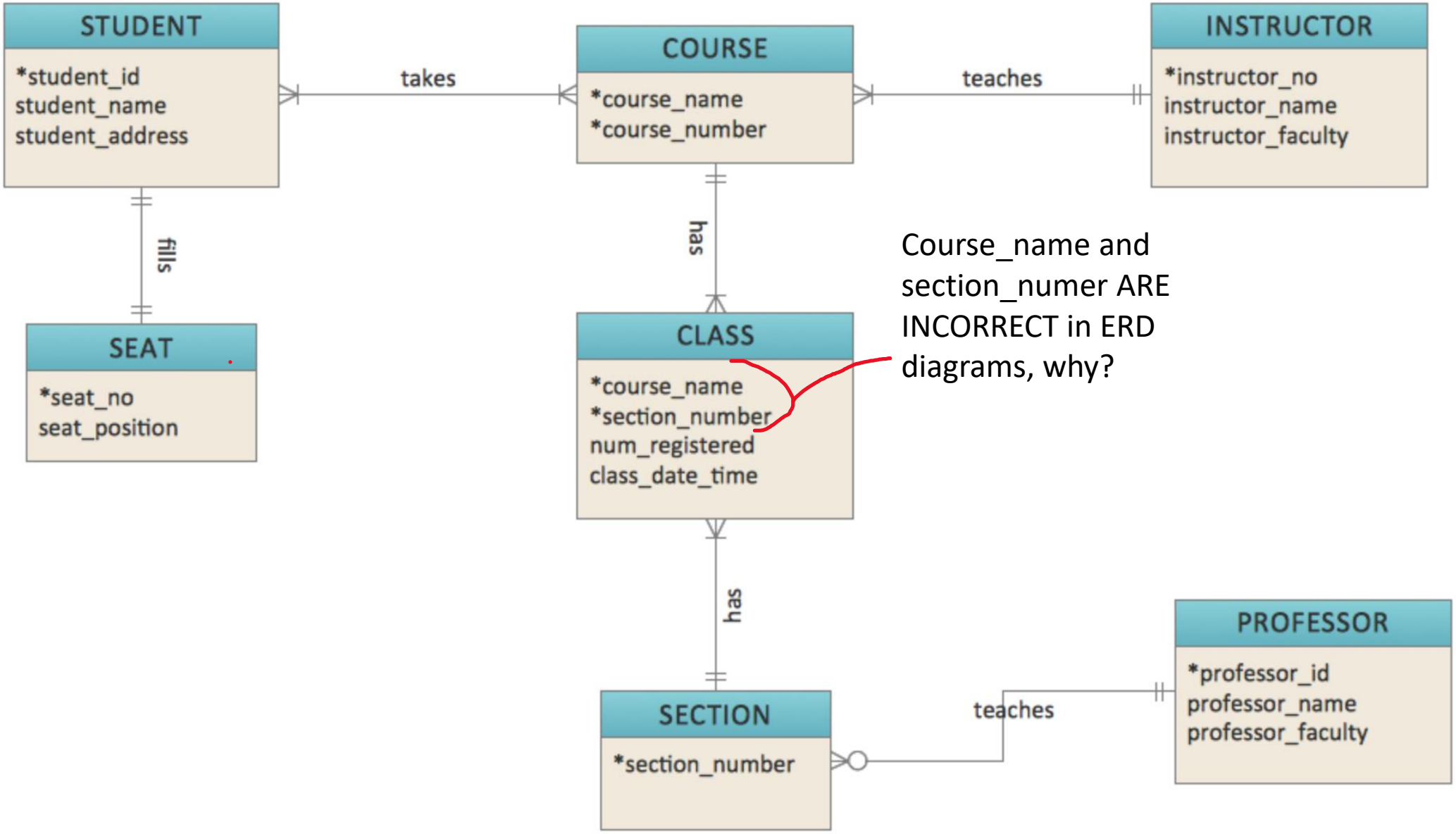
There are many notation styles that express cardinality.
Information Engineering Style



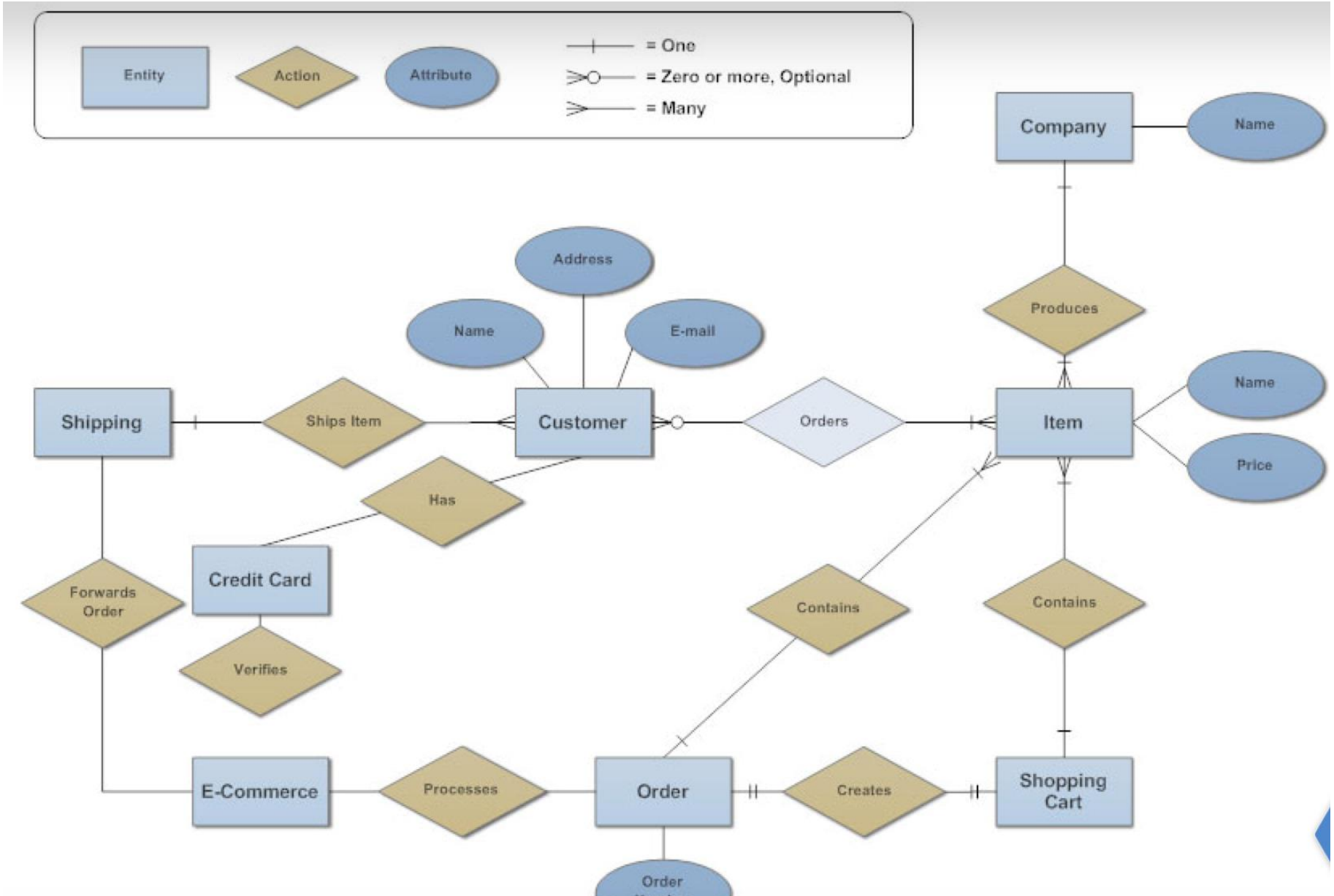
ERD Diagrams



ERD Diagrams - example



ERD Diagrams – example using Chens notations



ERD – task to practice on-site

University – simple version

A database includes information pertaining to students, lecturers (scientists) and lectures.

We store first names, last names and correspondence addresses for each person.

A lecture is described by its name, the semester number, and the maximum number of attendees.

Each student may enroll in many lectures and may have a supervisor.

Each scientist may be a supervisor of many students, but not more than 10 and has to give at least one lecture.