Data modeling

Introduction

Motivating questions

- Why should we store our data in a relational database?
- How should we organize our data?
- Why do we need data models to design a database?
- What makes a good data model?

Single table example

- Consider the spreadsheet, Departments.xlsx:

DeptNbr	DeptName	DeptType	DeptStatus
930	Receiving	Mfg	Active
378	Assembly	Mfg	Active
372	Finance	Adm	Active
923	Planning	Adm	Active
483	Construction	Plant	Inactive

- This sheet stores information about the concept of a department within a company

- This would be a table in a relational database

Relational model definitions

Relational model definitions

- Entity: object, concept or event
- Attribute (column): a characteristic of an entity
- Record or tuple (row): the specific characteristics or attribute values for one example of an entity
- Entry: the value of an attribute for a specific record
- Table: a collection of records
- Database: a collection of tables

Single table example

Entity: Departments

Table: A collection of records about the entity (departments)

Record: Information about department 372

Entry: Value of DeptNbr for the construction department

Departments

DeptNbr	DeptName	DeptType	DeptStatus
930	Receiving	Mfg	Active
378	Assembly	Mfg	Active
372	Finance	Adm	Active
923	Planning	Adm	Active
483	Construction	Plant	Inactive

Attribute: DeptName – the names of the departments

Database: CompanyDatabase, includes **tables** such as: Departments, Employees, Sales

Deeper dive on tables and attributes

- Tables
  - Tables represent entities, which are usually plural nouns
  - Tables are often named as exactly what they represent (typically plural nouns, without spaces):
    - e.g. Companies, Customers, Vehicles, Orders, etc.
- Attributes
  - Characteristics of an entity (table), typically nouns
  - Examples in the form of: Table (Attr1, Attr2, ... AttrN)
    - Vehicles (VIN, Color, Make, Model, Mileage)
    - Drivers (SSN, Fname, Lname, Address)
    - DriverLicenses (Type, Start_date, Expiration_date)

Entity types and entity occurrences

Entity type

Departments

DeptNbr
DeptName
DeptType
DeptStatus

Entity occurrence

Departments

DeptNbr	DeptName	DeptType	DeptStatus
930	Receiving	Mfg	Active
378	Assembly	Mfg	Active
372	Finance	Adm	Active
923	Planning	Adm	Active
483	Construction	Plant	Inactive

- When developing a data model, entity type descriptions should be as extensive as possible

Example entity type descriptions

- Poor description (seen lots of these)
  - Vendors: Someone we buy products from.
- Exemplary description (never seen one like this in real life)
  - Vendors: US corporations we have reviewed with respect to their qualifications for providing products to our company. Vendors are rated based on price, quality, delivery performance and financial stability. Each vendor is classified by one vendor status: approval pending, approved, rejected or inactive. This approval decision is made in a weekly meeting among purchasing, manufacturing and finance. Purchasing requests that rejected vendors be kept in the database for future reference. Purchasing expects 400 vendors will be maintained at any one time. Of these, 200 will be active, 25 pending, 75 inactive and 100 rejected. Contact Joan Smith in Purchasing for more information.

Data models

- When designing a database to store and analyze data, you first need to develop a data model
- The data model describes the data that is stored in the database and how to access it
- The data model defines the tables and attributes in the database
  - Each important concept/noun in the data is defined as a table in the database

Key points from lesson

- Data in relational databases are organized into tables, which represent entities
- Single tables within a database are like spreadsheets, but we use different vocabulary to talk about the rows and columns
- Entity types should be described as part of the data modeling process, this will help with the documentation and determination of business rules

Data modeling

Solutions: Entity and attribute

Identify which are entities and which are attributes:

- Instructor (E)
- Teaching assistant (TA) (E)
- Course section number (A)
- Building name (A)
- Course number (A)
- Textbook price (A)
- Teaching asst (TA) name (A)
- Instructor ID (A)
- Textbook author (A)
- Course title (A)
- Textbook (E)
- Classroom (E)
- Textbook ISBN (A)
- Section days (A)
- Instructor office hours (A)
- Textbook title (A)
- Classroom number (A)
- TA student ID (A)
- Instructor name (A)
- Textbook publisher (A)
- Section capacity (A)
- Course objective (A)
- Copyright date (A)
- Building number (A)
- Course section (E)
- Course (E)
- Building (E)
- Section time (A)
- Classroom capacity (A)

Designing a data model

- Data models help specify each entity in a table in a standardized way
- Data models allow administrator to impose rules, constraints, and relationships on the data that are stored
  - Enables users to understand business rules and effectively process and analyze data
- Acts as a schematic for building the database

Rules of the relational data model

- Each attribute (column) has a unique name within a table
- All entries or values in the attribute are examples of that attribute
- Each record (row) is unique in a good database
- Ordering of records and attributes is unimportant

Departments

DeptNbr	DeptName	DeptType	DeptStatus
930	Receiving	Mfg	Active
378	Assembly	Mfg	Active
372	Finance	Adm	Active
923	Planning	Adm	Active
483	Construction	Plant	Inactive

Characteristics of a good data model

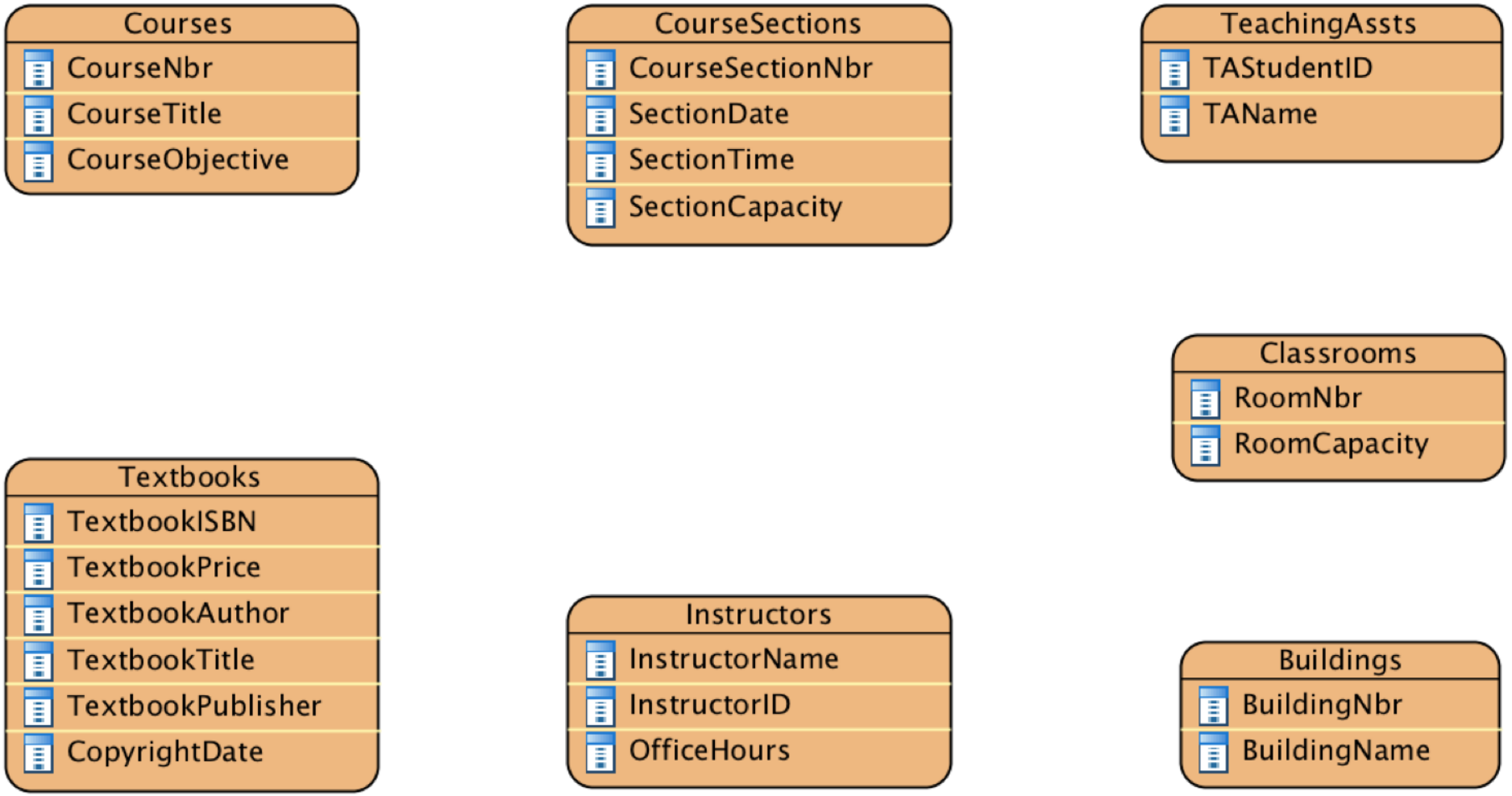
- Complete: Is all necessary data represented?
- No redundancy: Is the same fact recorded more than once?
- Enforcement of rules: How accurately does it enforce business rules?
- Reusability: Can the database be used for different applications (e.g. web application, enterprise analytics, etc.?)
- Flexibility: Can the model cope with possible changes to the business rules or data requirements?

Key points from lesson

- The data model describes the data that is stored in the database and how to access it
- Each record is unique in a good database
- Data models enable users to understand business rules and effectively process and analyze data

Relationships and cardinality

Solutions: Entity type and attribute

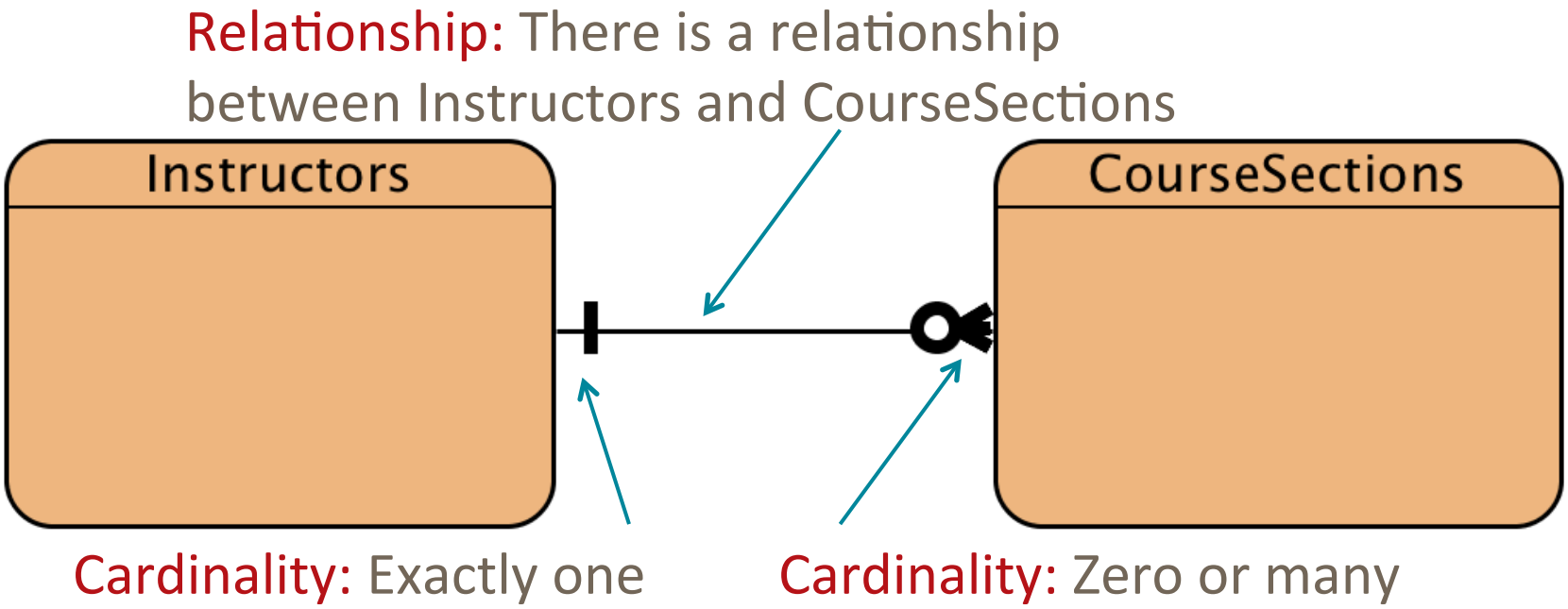


How to draw an entity-relationship diagram (ERD)

- ERD or entity-relationship diagram is a schematic of the database
- Entities are drawn as boxes
- Relationships between entities are indicated by lines between these entities
- Cardinality describes the expected number of related occurrences between the two entities in a relationship and is shown using crow's foot notation

Relationships + cardinality = business rules

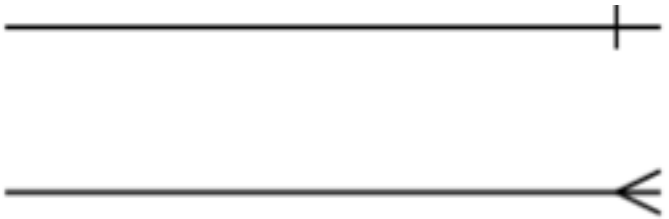
ERD for Instructors and CourseSections



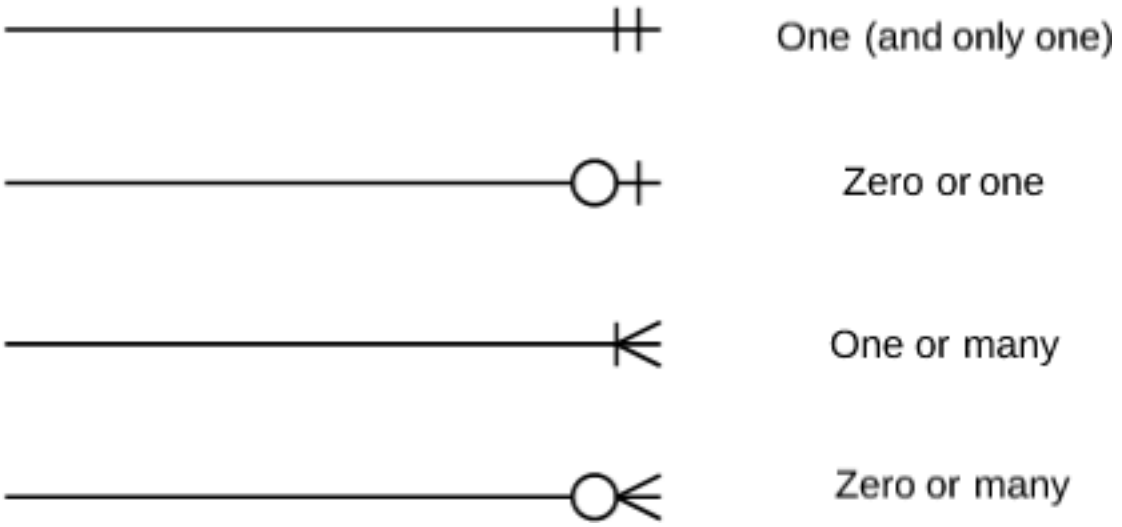
- Business rules defined through relationships and cardinality:
  - There is exactly one instructor for each course section
  - Each instructor may teach zero, one or many course sections (shortened to zero or many)

Cardinality – crow's foot notation

- General meanings:



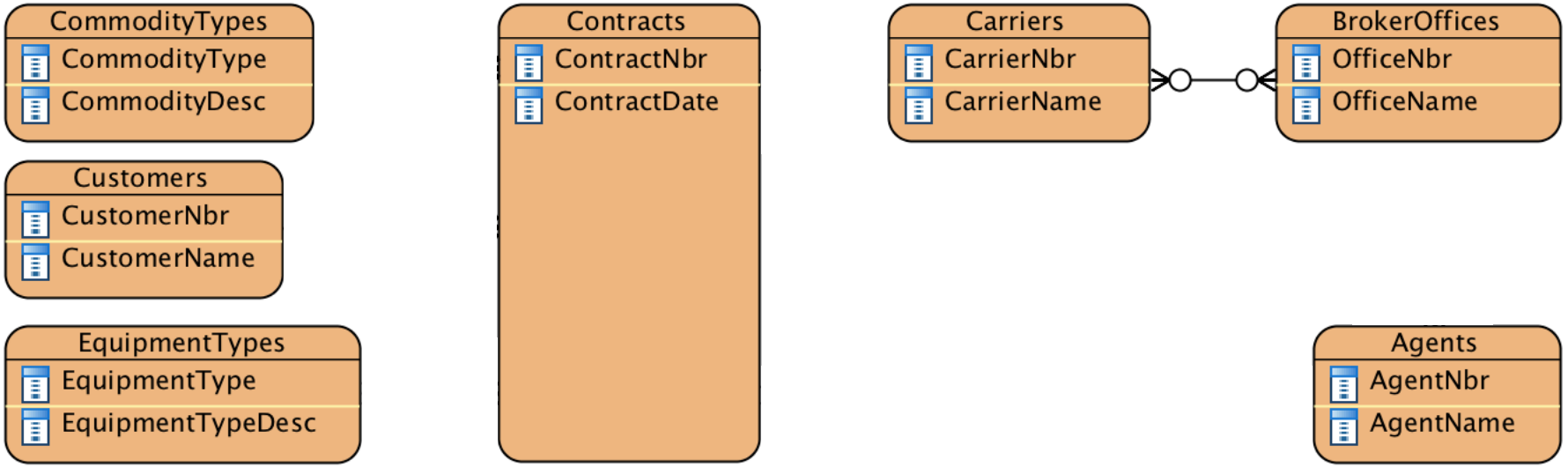
- Mandatory vs. optional:



Transportation broker example

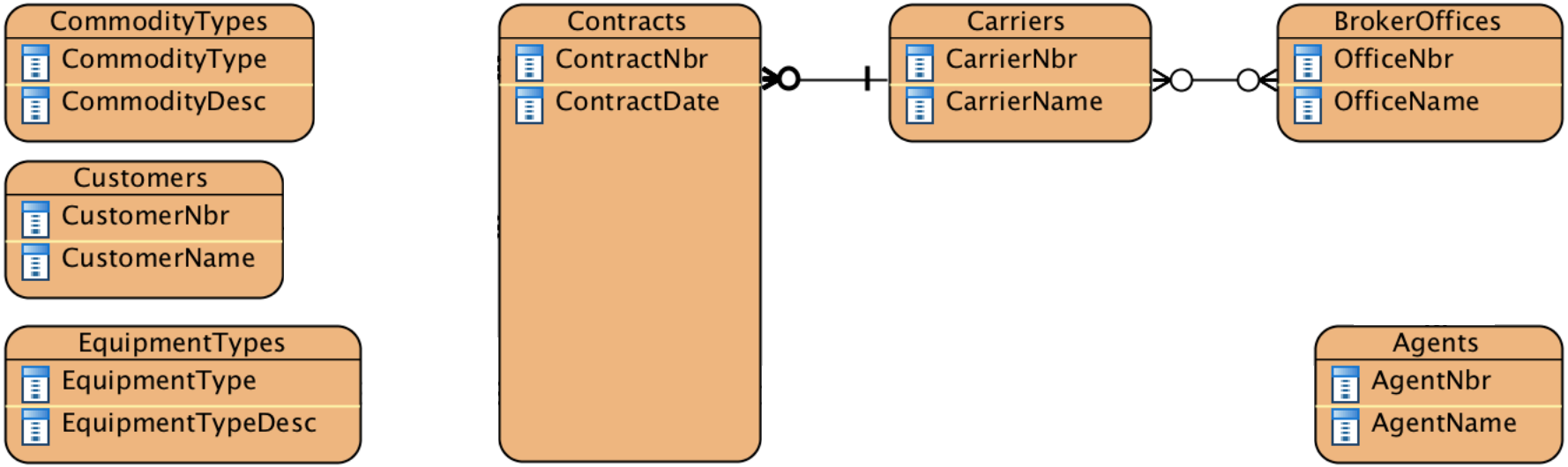
- On the next slide there is a small data model for a freight shipping broker
- Captures underlying rules or logic of broker's business
- Provides information about how the database should be structured

Transportation broker data model



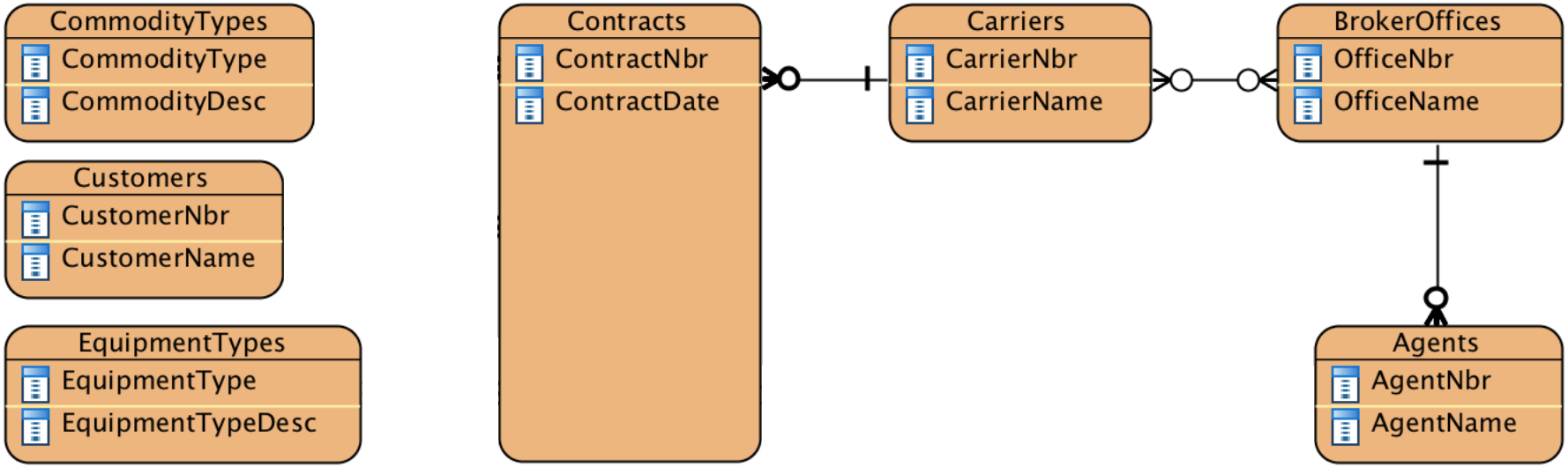
- A carrier can be associated with many offices
- An office can be associated with many carriers

Transportation broker data model



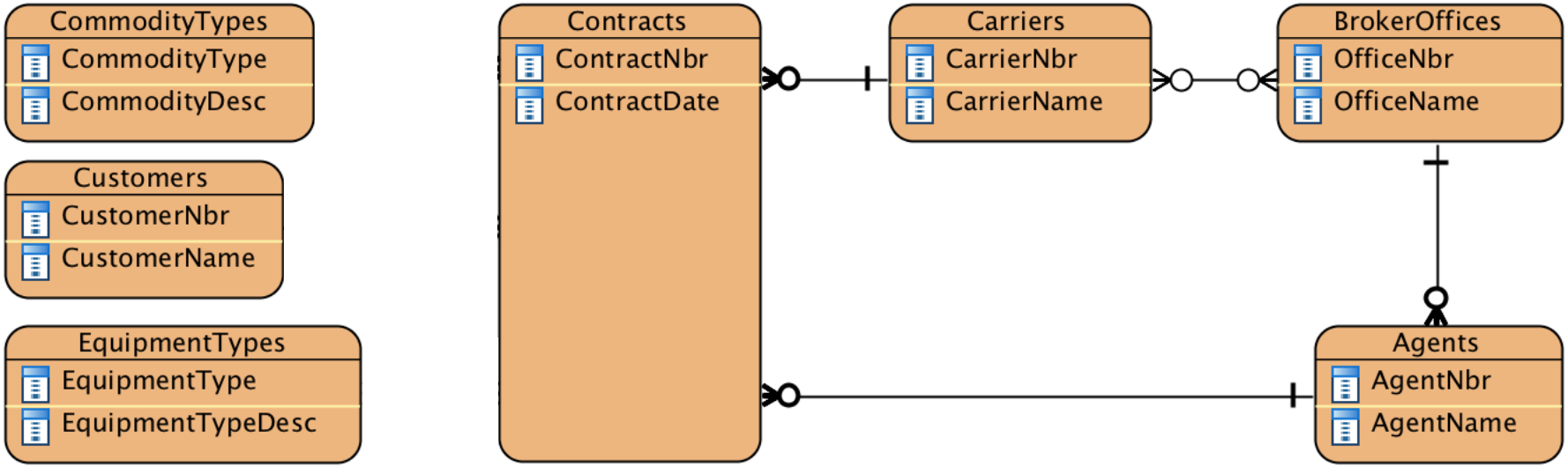
- A carrier can issue many contracts
- A contract is issued by one carrier

Transportation broker data model



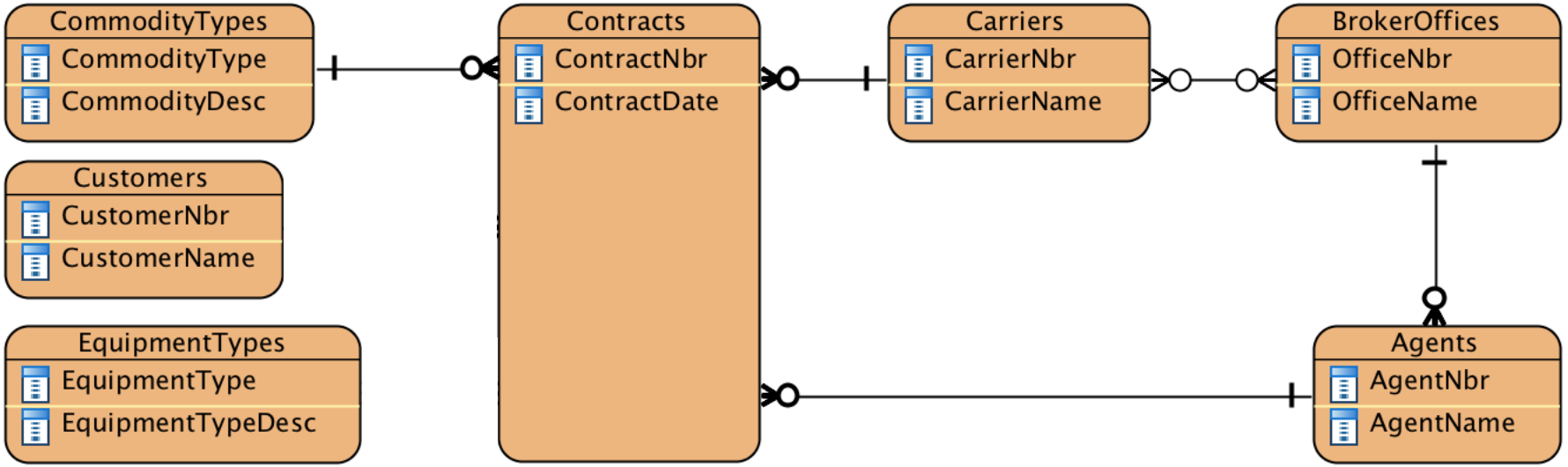
- An office can employ many agents
- An agent is employed by one office

Transportation broker data model



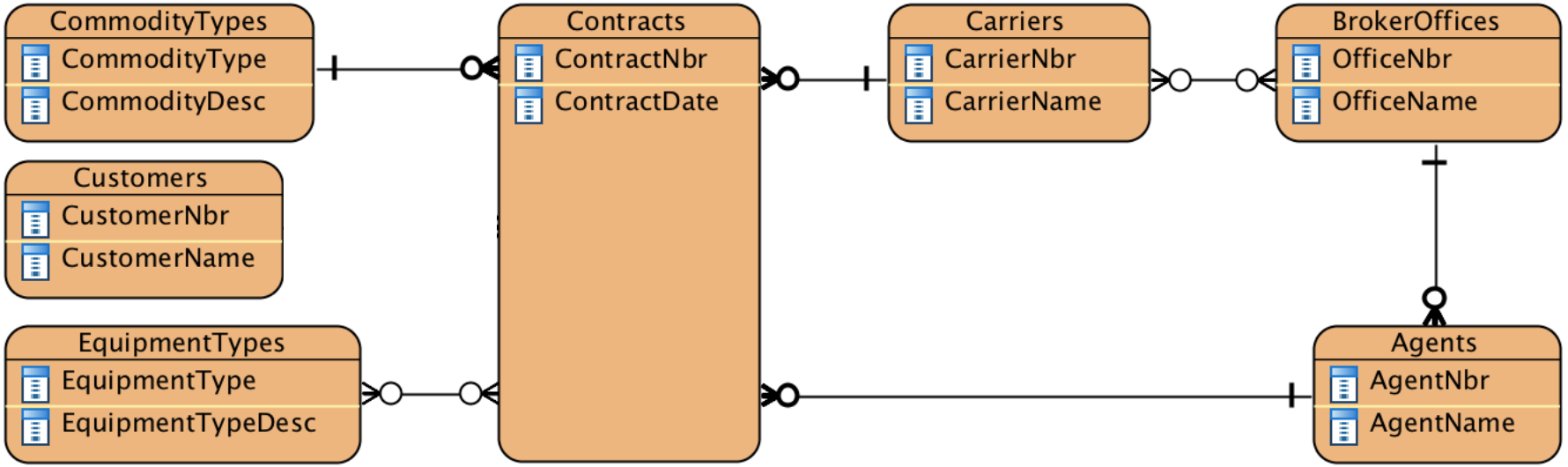
- An agent can sell many contracts
- A contract is serviced by only one agent

Transportation broker data model



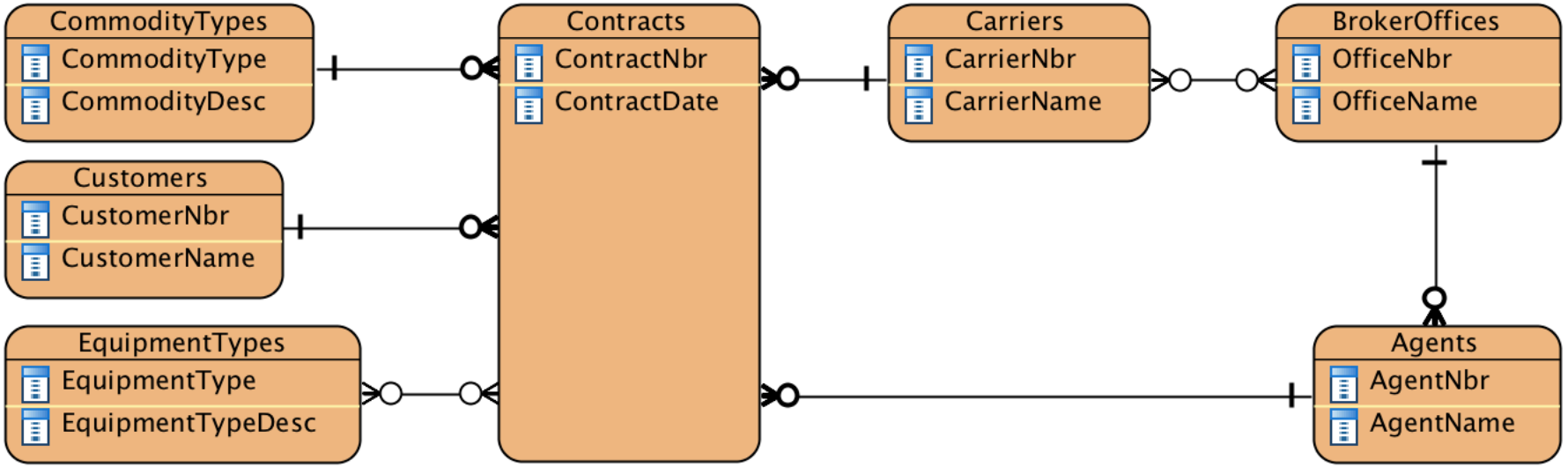
- A contract can serve to carry only one commodity type
- A commodity type can be carried under many contracts

Transportation broker data model



- A contract can be associated with many equipment types
- An equipment type can be associated with many contracts

Transportation broker data model



- A customer can be served by many contracts
- A contract covers one customer

Is there always only one solution for a data model?

- Several solutions may exist
- Often, these will describe different underlying business processes or rules
- These often depend on the application requirements or business needs

Domain validation entities

- Also called pick lists or validation lists
- Used to standardize data in a database

Department			
DeptNbr	**DeptName**	**DeptType**	**DeptStatus**
930	Receiving	Mfg	Active
378	Assembly	Mfg	Active
372	Finance	Adm	Active
923	Planning	Adm	Active
483	Construction	Plant	Inactive

Domain validation entity

ValidDeptTypes
Mfg
Adm
Plant
Sales
Operations

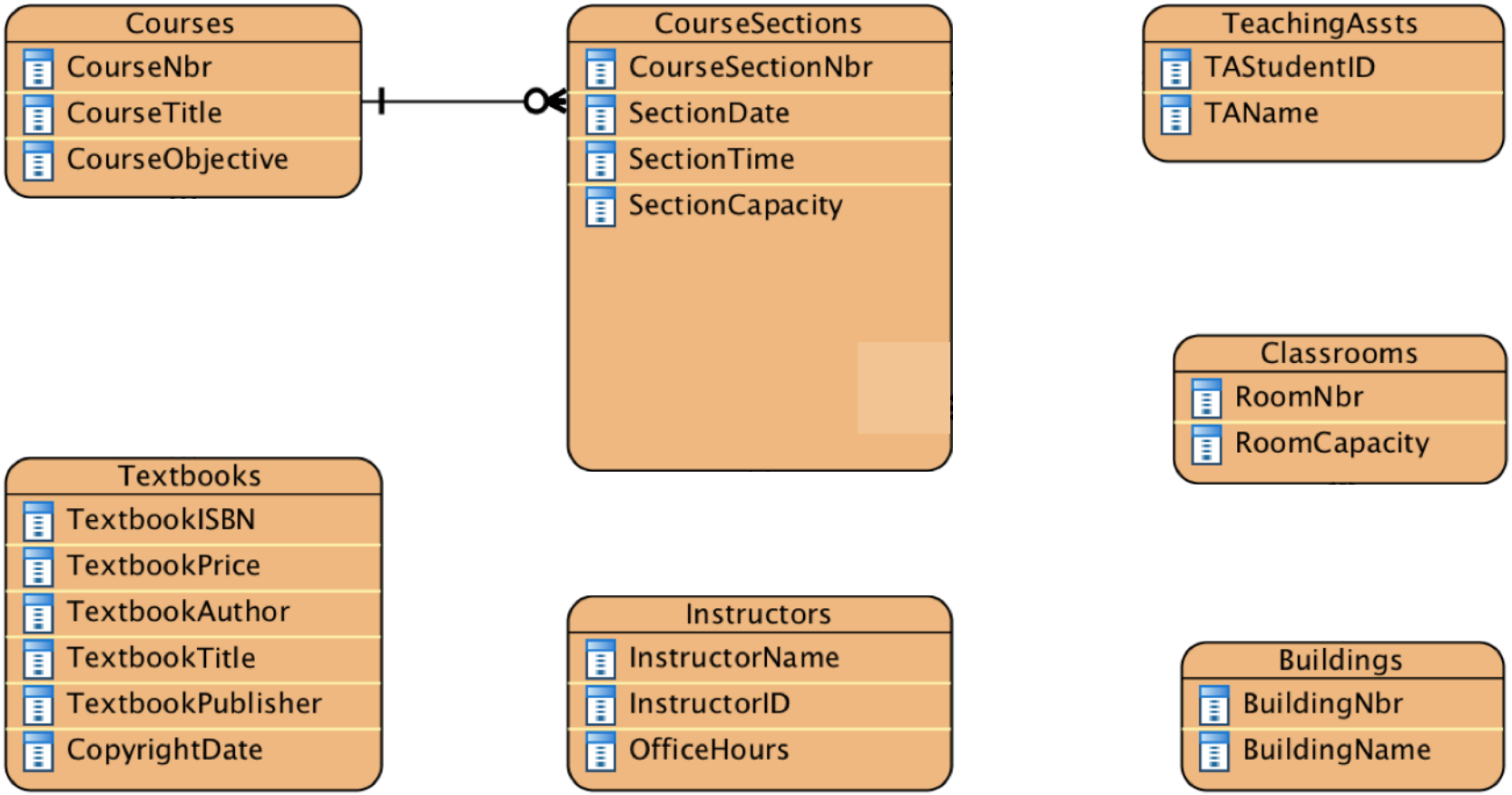
- Domain validation entity: table with a single attribute, enforces values of attribute in related table
- Requires that any new department type must be on a list of existing department types in the table "ValidDeptTypes"

Key points from lesson

- Business rules are imposed on the database through relationships and cardinality
- Business rules are also understood based on relationship and cardinality
- Domain validation entities restrict entries to a set of specified values
- Data models may vary for a given dataset as business logic evolves

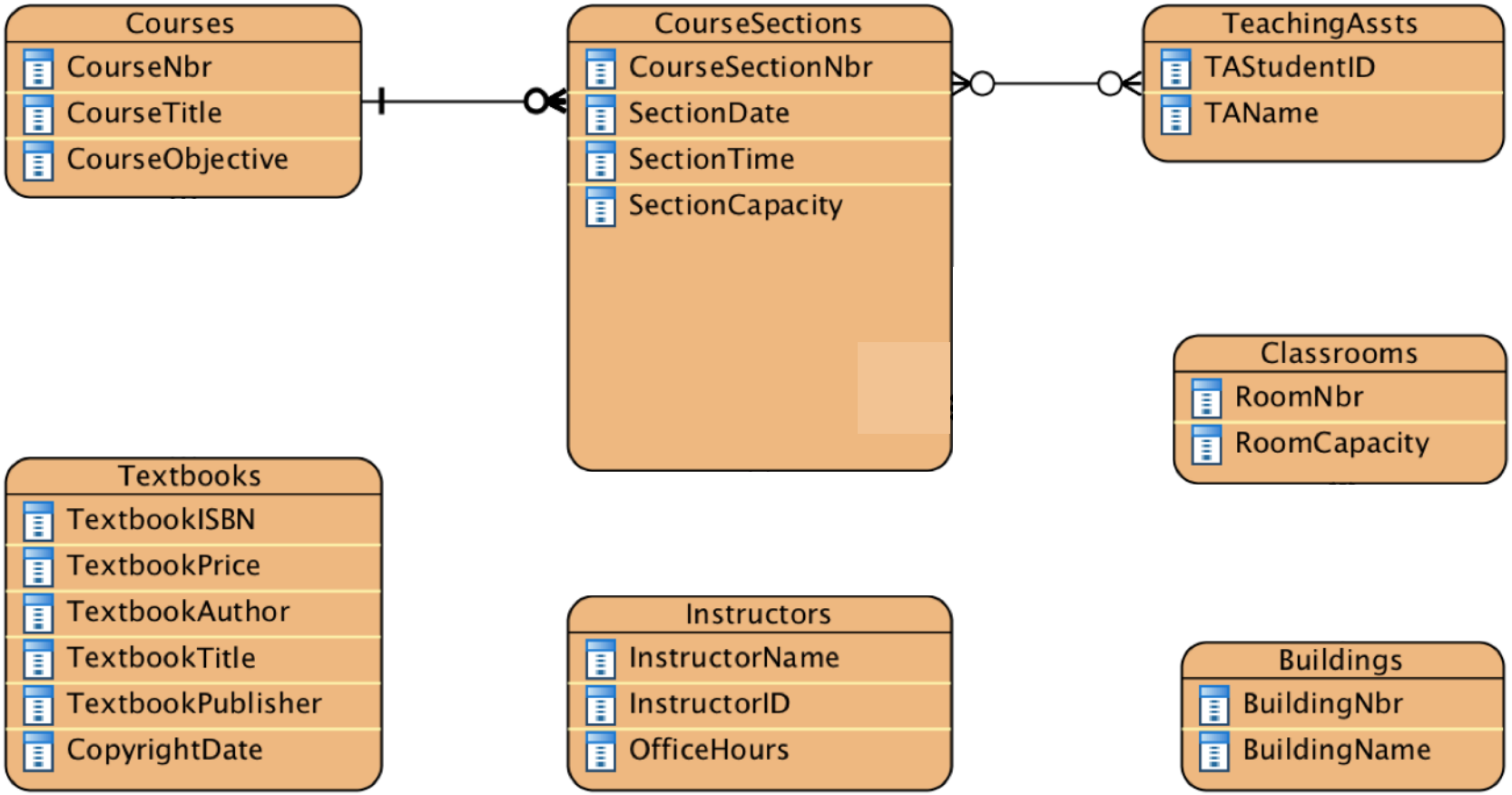
Keys

Solutions: Relationships and cardinality



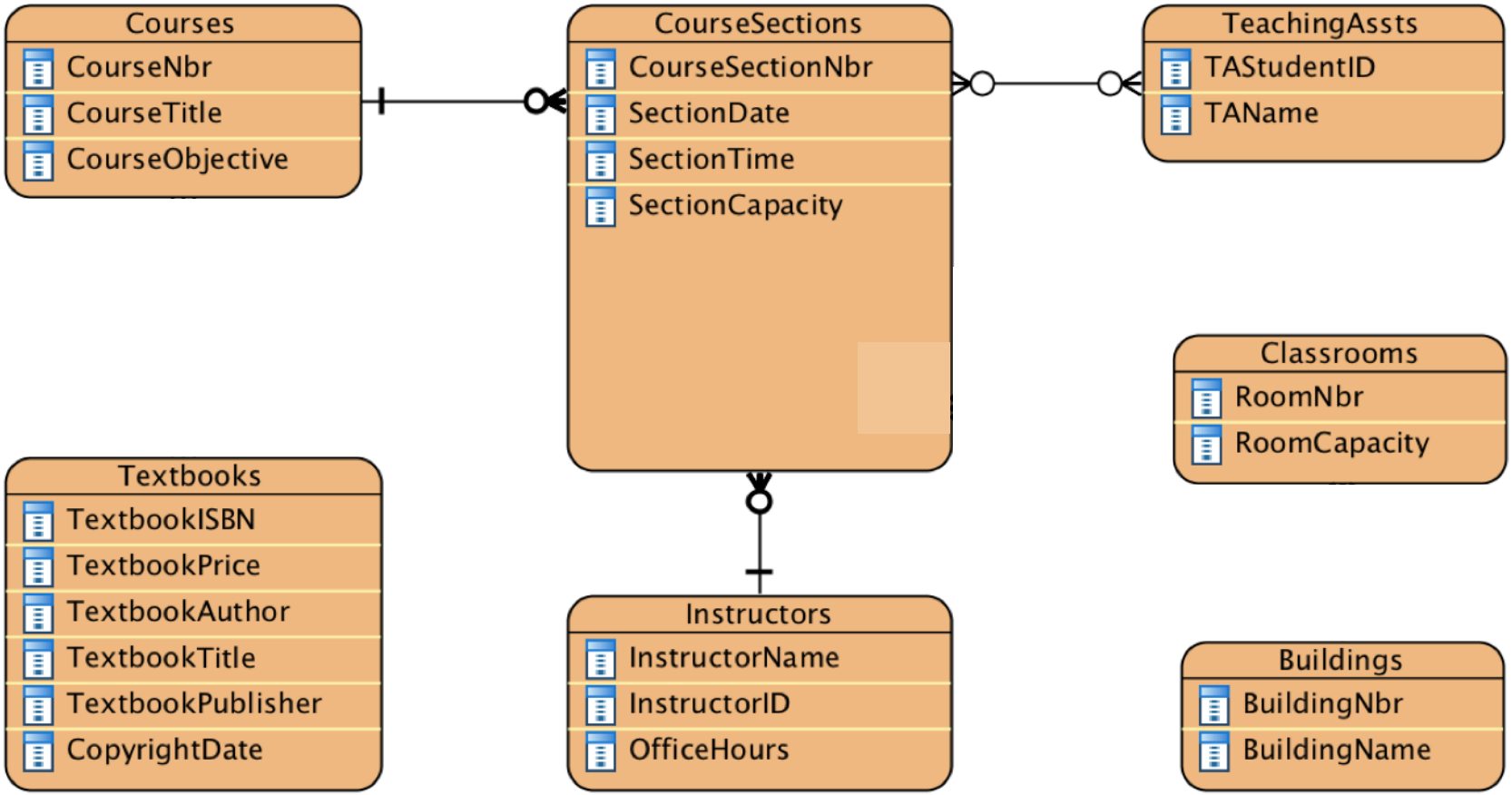
- Course may be offered in many (0,1 or more) sections
- Course section must be associated with a course

Solutions: Relationships and cardinality



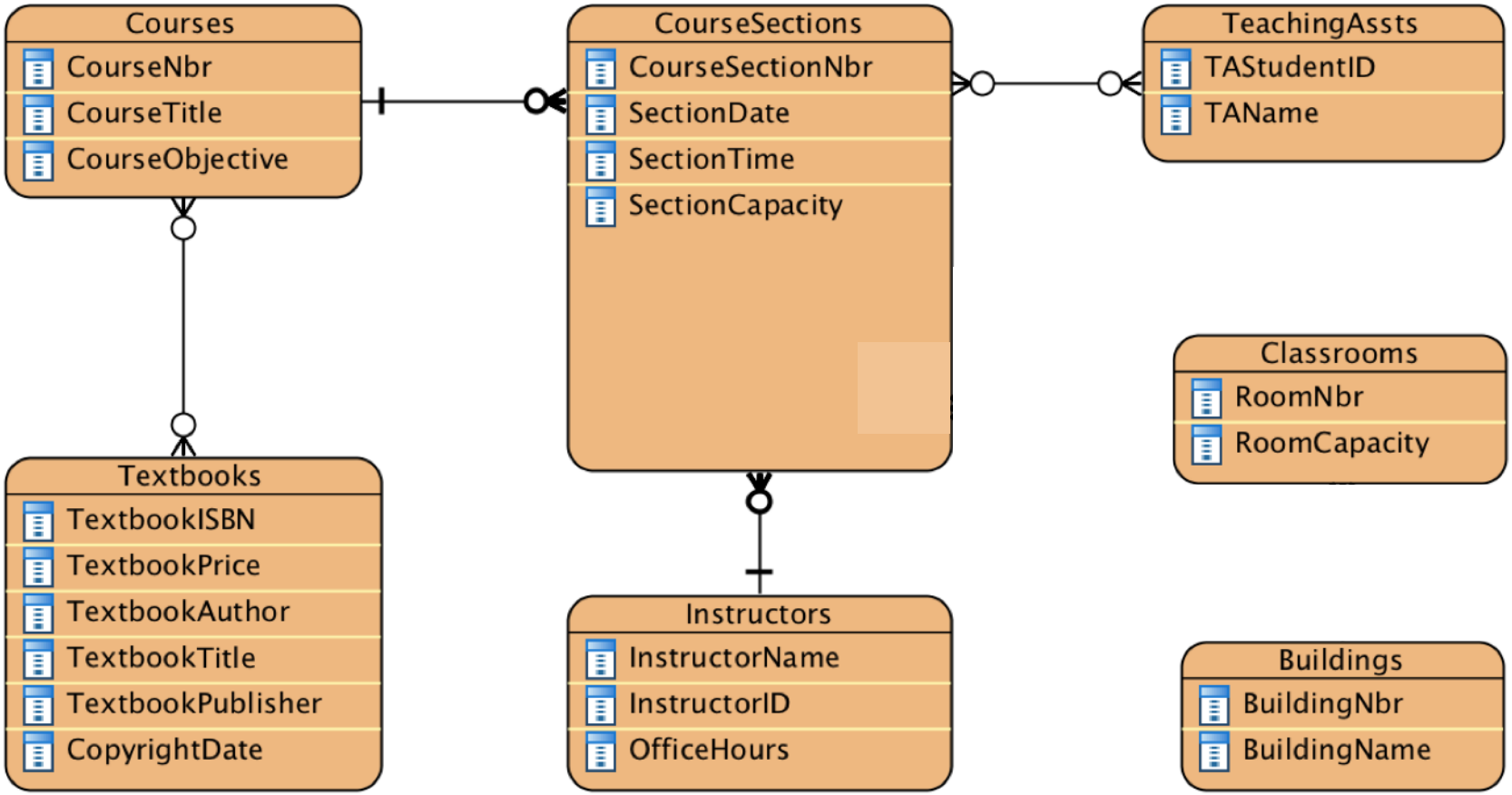
- Course section may be taught by many (0,1 or more) TAs
- TA may teach many (0, 1 or more) course sections

Solutions: Relationships and cardinality



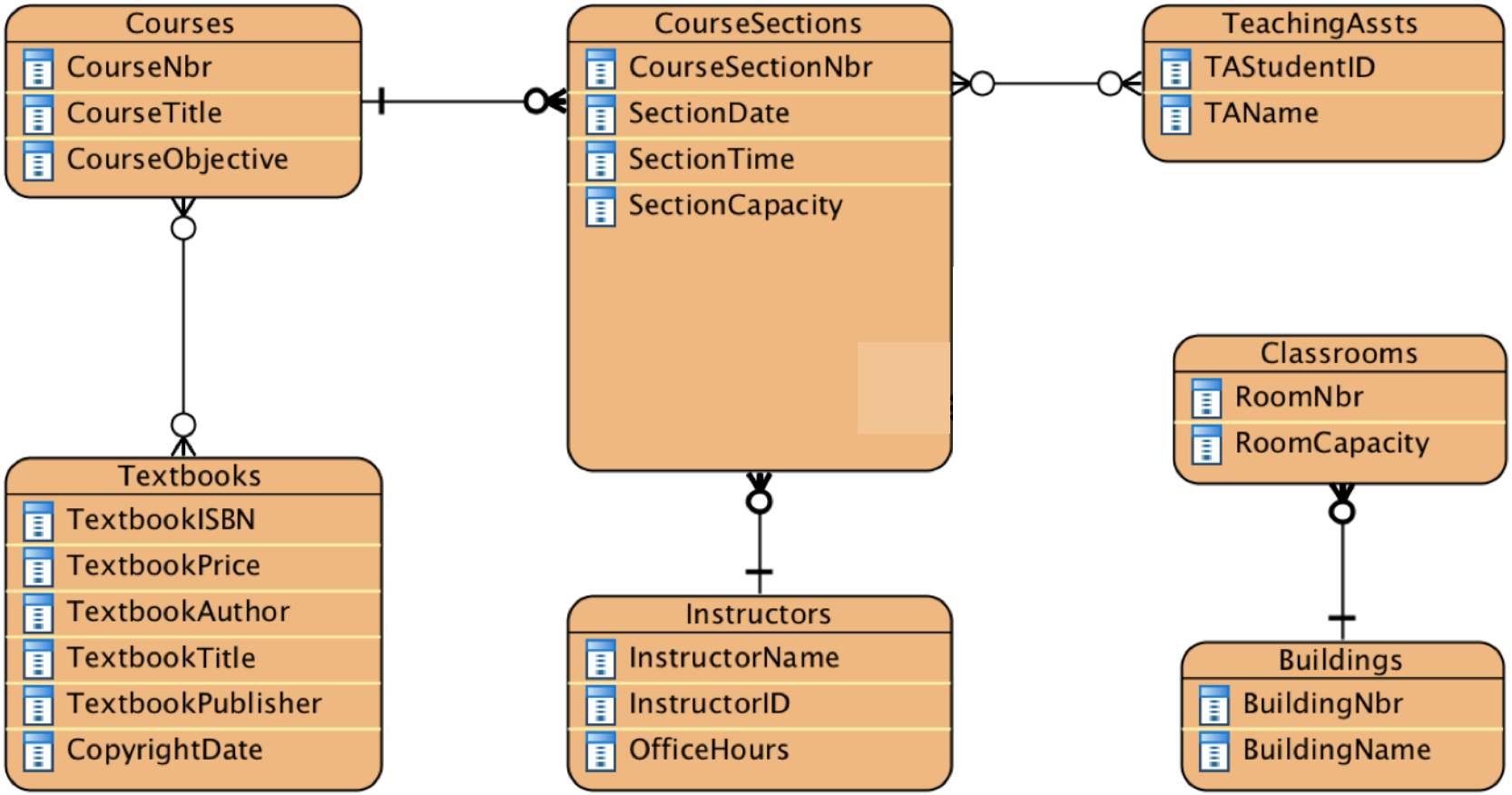
- Course section must be taught by 1 instructor (??)
- Instructor may teach many sections

Solutions: Relationships and cardinality



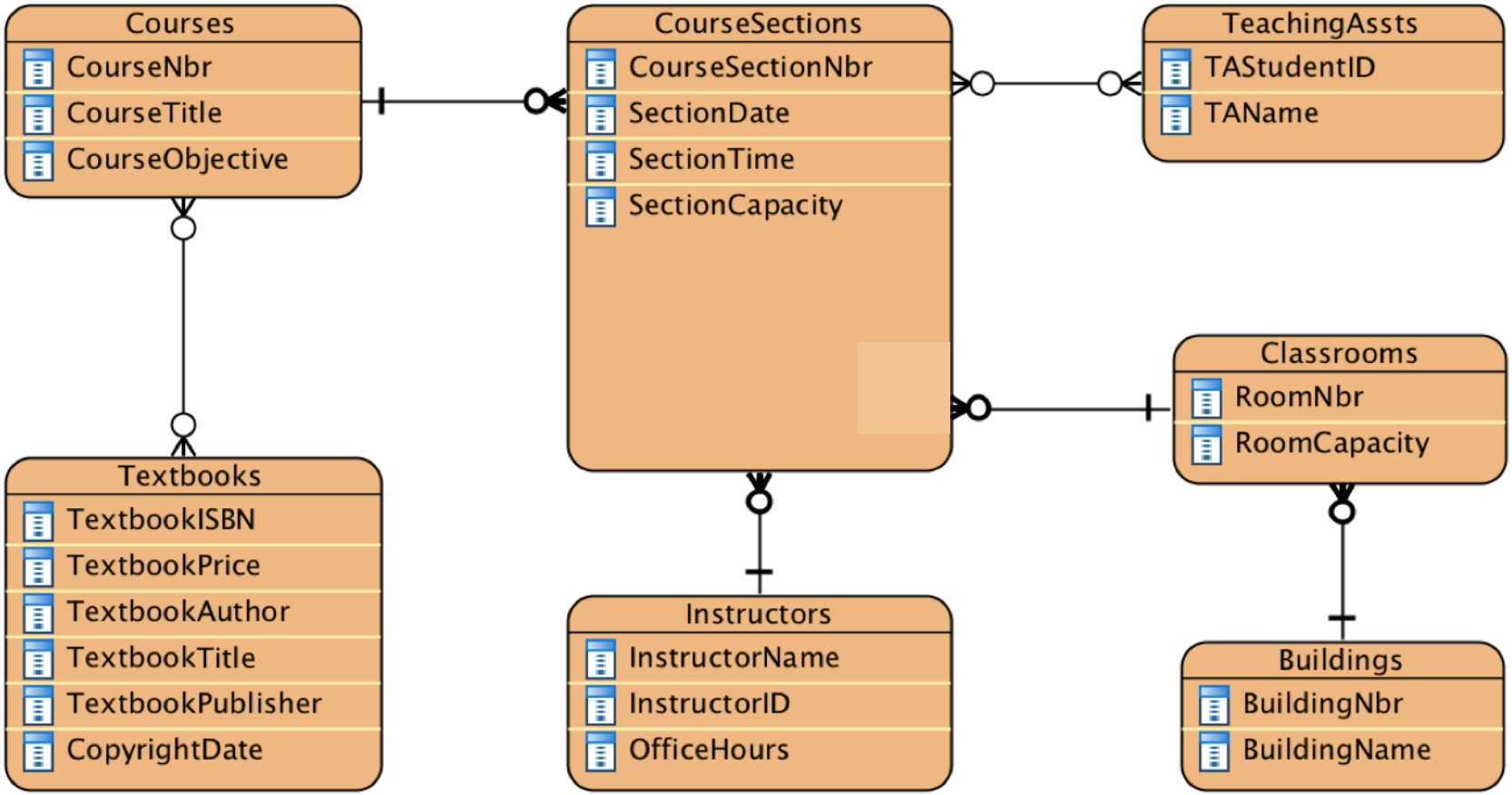
- Course may use many textbooks (all sections use same)
- Textbook may be used in many courses

Solutions: Relationships and cardinality



- Building may contain many rooms
- A room is in only one building

Solutions: Relationships and cardinality



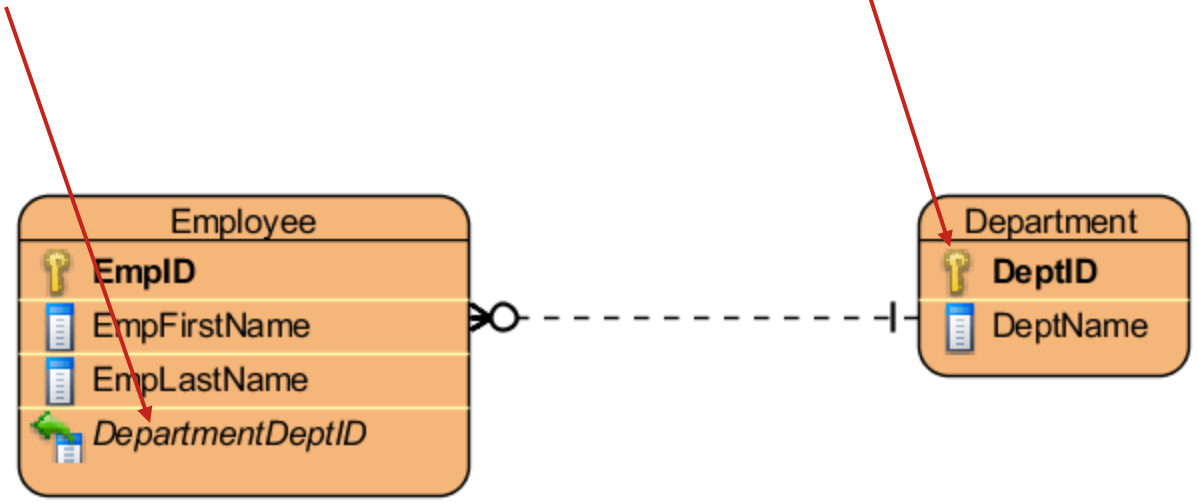
- A course section may use a room
- A room may be used by many course sections (not at same time)

Primary and foreign keys

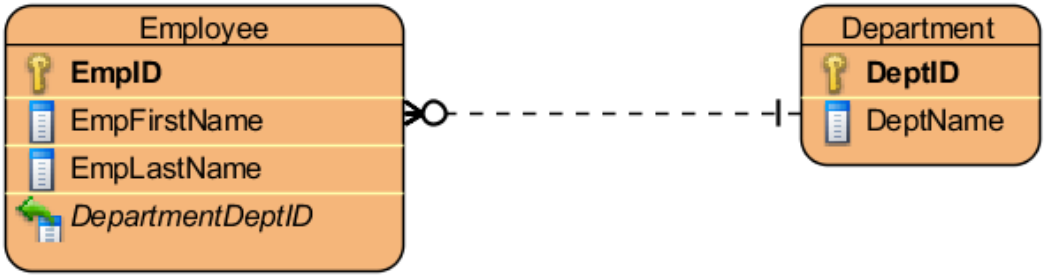
- Primary key: one or more attributes that uniquely identify a record
- What would you use in a customer database of 100,000 people and no unique customer id?
  - Name not unique
  - Add birthdate, but not guaranteed to be unique
  - Address can change
  - Can use social security number, but not everyone has one
  - Privacy is an issue

Primary and foreign keys

- Primary key of the independent or parent entity type is maintained as a non-key attribute in the related, dependent or child entity type, this is known as the foreign key



Foreign keys



Employee

EmpID	DeptID	EmpLastName	EmpFirstName
4436	483	Brown	John
4574	483	Jones	Helen
5678	372	Smith	Jane
5674	372	Crane	Sally
9987	923	Black	Joe
5123	923	Green	Bill
5325	483	Clinton	Bob

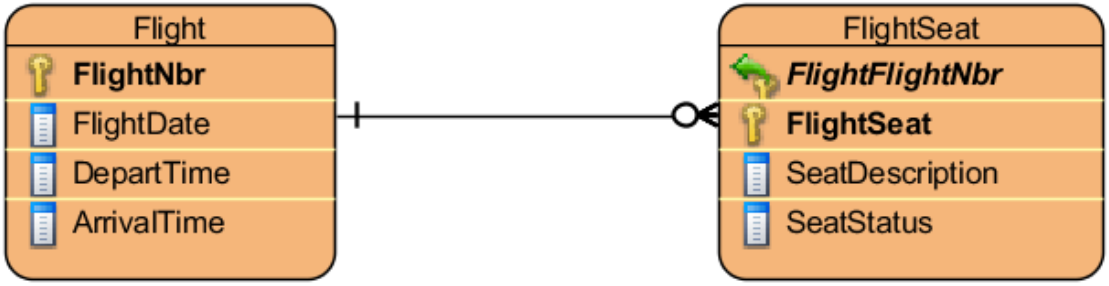
Department

DeptID	DeptName
930	Receiving
378	Assembly
372	Finance
923	Planning
483	Construction

- Database requires a valid department number (or null) when employee is added
- Employee ID is the unique identifier of employees; department number is not needed as part of the employee primary key

Composite keys

- A composite key is a primary key that consists of more than one attribute



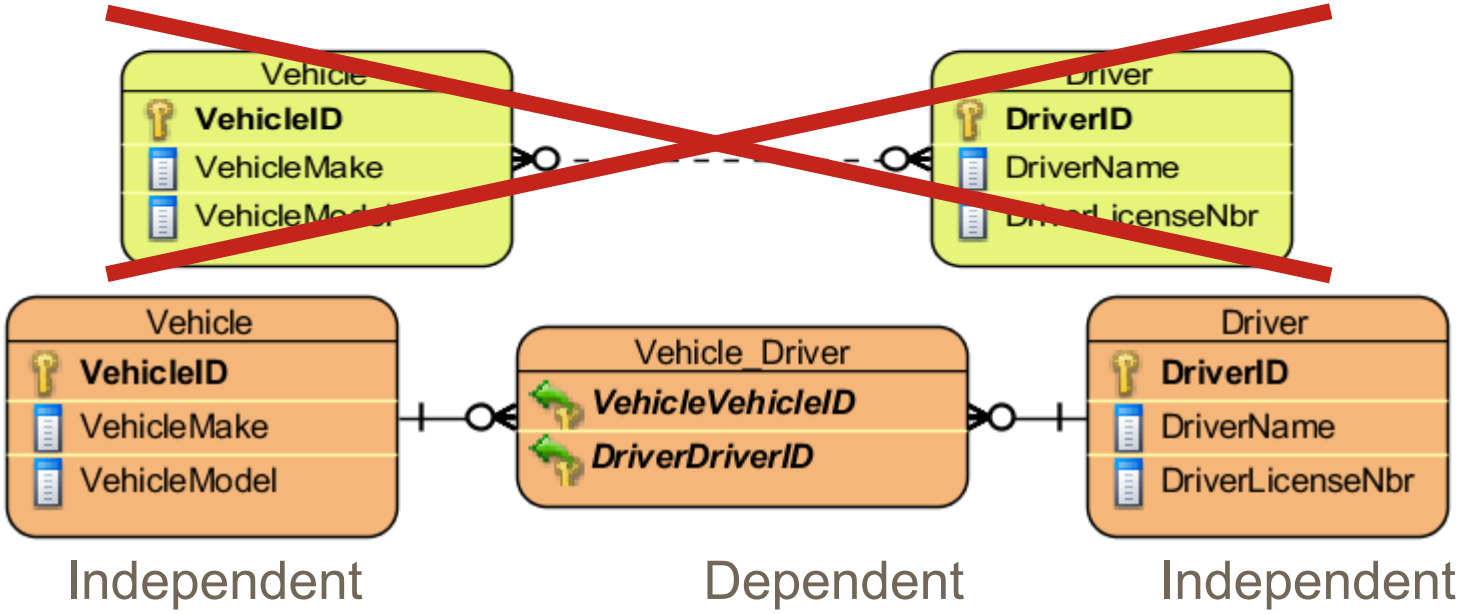
- Consider a charter airline: every flight has a different number

Flight			
FlightNbr	FlightDate	DepartTime	ArrivalTime
243	9/24	9:00am	11:00am
253	9/24	10:00am	12:30pm
52	9/24	11:00am	2:00pm

FlightSeat			
FlightNbr	SeatNbr	SeatStatus	Seat Description
243	8A	Confirmed	Window
243	7D	Reserved	Aisle
243	14E	Open	Center
253	1F	Open	Window
253	43A	Confirmed	Window

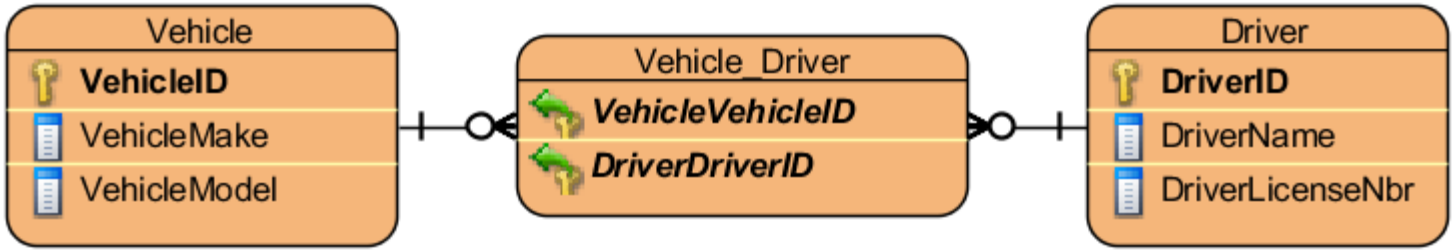
Many to many relationships

- Vehicle can be driven by many drivers; driver can drive many vehicles
- How can we get vehicle information for a driver from the database?



- Associative table (entity), aka junction table
- Primary key of parent is used in primary key of child

Many to many relationships



Vehicle

VehicleID	VehicleMake	VehicleModel
35	Volvo	Wagon
33	Ford	Sedan
89	GMC	Truck

Driver

DriverID	DriverName	DriverLicenseNbr
253	Ken	A23423
900	Jen	B89987

VehicleDriver

VehicleID	DriverID
35	900
35	253
89	900

- Never create an entity with vehicle1, vehicle2, etc. as attributes!

Referential integrity

- Referential integrity maintains the validity of foreign keys when the primary key in the parent table changes
  - Every foreign key either matches a primary key (or is null)
  - For example: cannot add an employee to an invalid department

- Cascade rules: choose among delete options
  - Cascade restrict: Rows in the primary key table can't be deleted unless all corresponding rows in the foreign key tables have been deleted
  - Cascade delete: When rows in the primary key table are deleted, associated rows in foreign key tables are also deleted

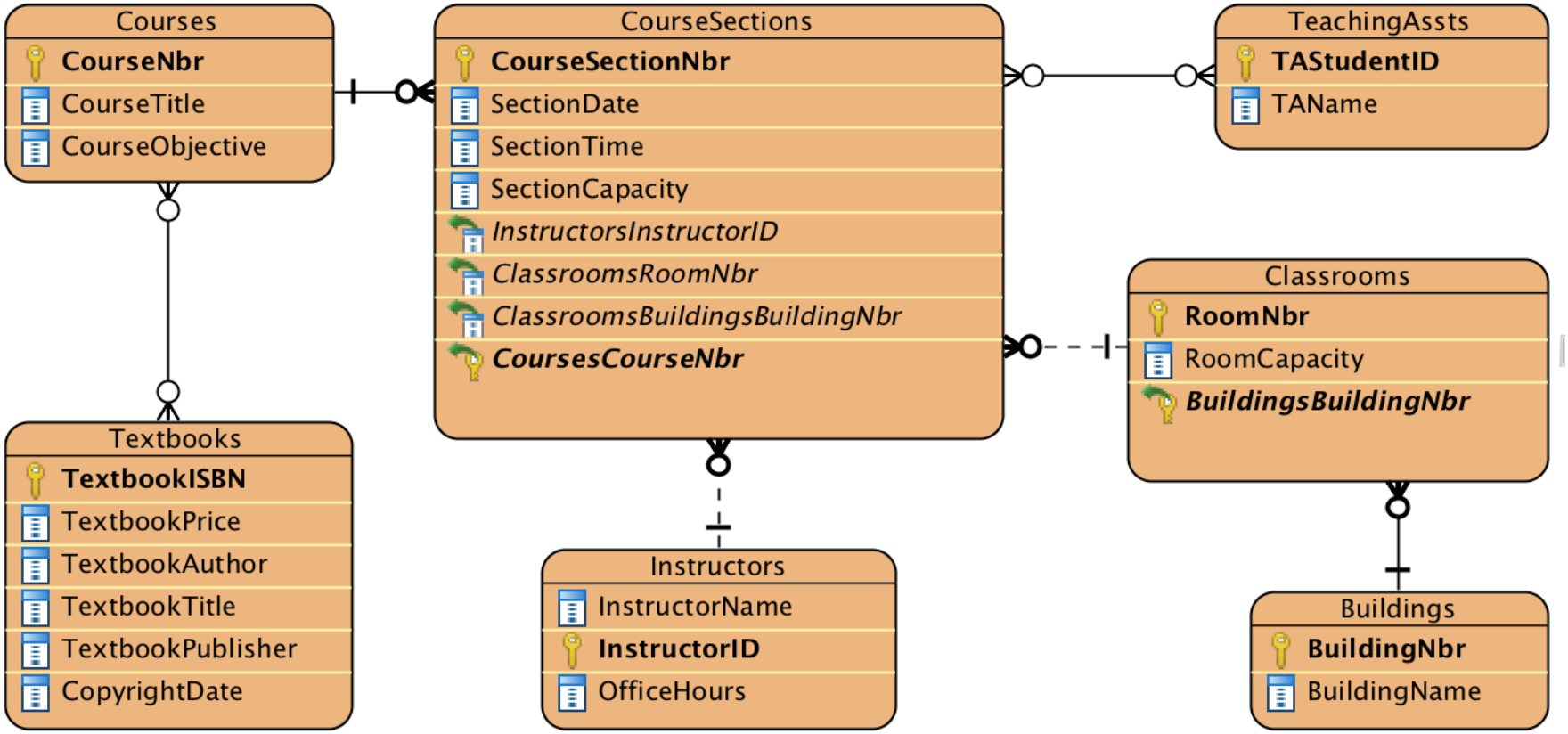
Key points from lesson

- Primary keys are attributes used to uniquely identify a record
- Foreign keys are attributes stored in a dependent entity which show how records in the dependent entity are related to an independent entity
- Data model consists of:
  - Entities and attributes
  - Primary keys
  - Foreign keys
  - Relationships and cardinality
  - Referential integrity and cascade rules

edX example

Solutions: Primary and foreign keys

- We're getting there with this ERD: we've defined entities, attributes, relationships and keys



Introduction to edX data modeling exercise

- We have all of the user data from the first year of running edX
- Download the dataset resource and inspect it
  - 10 percent of the data has been randomly selected for use
- Flat file: a table with all attributes and records from which we will design the database using a relational data model
- What are the most appropriate entities? What concepts should be represented in the database table?

Exercise motivation and approach

- Why is it useful to do this exercise?
  - Scenario: Customer or client hands you data for analysis in one or more giant Excel sheet
  - Must understand business rules underpinning the dataset

- Look at the single-table dataset
- How could data structure be improved with a relational database?
  - Identify major entities
  - Identify attributes of those entities
  - Identify relationships between entities

edX Dataset File (from website)

	A	B	C	D	E	F	G	H	I
1	course_id	Course_Short	Course_Long	userid_DI	registered	viewed	explored	certified	Country
2	HarvardX/CB	HeroesX	The Ancient (	MHxPC1304	1	0	0	0	Germany
3	HarvardX/CB	HeroesX	The Ancient (	MHxPC1305	1	1	0	0	United S
4	HarvardX/CB	HeroesX	The Ancient (	MHxPC1303	1	0	0	0	United S
5	HarvardX/CB	HeroesX	The Ancient (	MHxPC1303	1	1	0	0	United S
6	HarvardX/CB	HeroesX	The Ancient (	MHxPC1303	1	0	0	0	China
7	HarvardX/CB	HeroesX	The Ancient (	MHxPC1303	1	1	0	0	United K
8	HarvardX/CB	HeroesX	The Ancient (	MHxPC1303	1	1	0	0	United S
9	HarvardX/CB	HeroesX	The Ancient (	MHxPC1305	1	0	0	0	United S
10	HarvardX/CB	HeroesX	The Ancient (	MHxPC1302	1	0	0	0	Other Af
11	HarvardX/CB	HeroesX	The Ancient (	MHxPC1303	1	0	0	0	United S
12	HarvardX/CB	HeroesX	The Ancient (	MHxPC1303	1	1	0	0	Greece
13	HarvardX/CB	HeroesX	The Ancient (	MHxPC1303	1	0	0	0	United S
14	HarvardX/CB	HeroesX	The Ancient (	MHxPC1301	1	1	0	0	United S
15	HarvardX/CB	HeroesX	The Ancient (	MHxPC1303	1	1	0	0	Colombia
16	HarvardX/CB	HeroesX	The Ancient (	MHxPC1300	1	1	0	0	United S

Data dictionary

Column Name	Description
course_id	three-part identifier for a course
Course_Short_Title	Short title for the course
Course_Long_Title	Long title for the course
userid_DI	Individual user ID
registered	Whether the user is registered (1/0)
viewed	Whether the user has viewed the contents (1/0)
explored	Whether the user has explored the course (1/0)
certified	Whether the user is certified (1/0)
Country	User's country of origin
LoE_DI	User's level of education
YoB	User's year of birth
Age	User's age
gender	User's gender
grade	User's grade in the course
nevents	Number of events the user has done on the site
ndays_act	Number of actions taken by the user
nplay_video	Number of video plays done by the user
nchapters	Number of chapters read by the user
nforum_posts	Number of forum posts made by the user
roles	Any roles the user has
incomplete_flag	Whether the user has an incomplete for the course

Most obvious entities and keys

- Entity: Users
  - Primary key: userid_DI
- Entity: Courses
  - Primary key: course_id

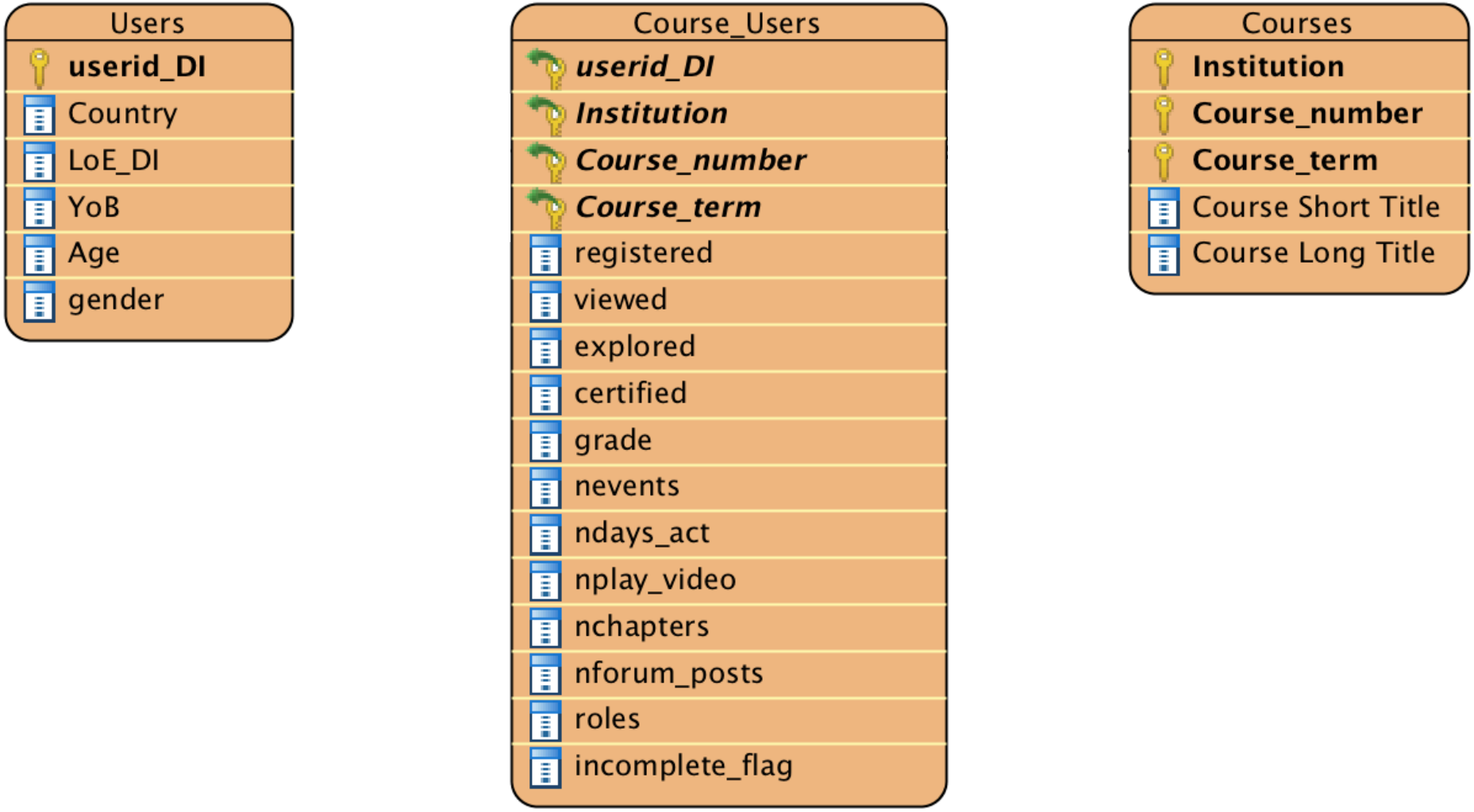
Primary key for courses

- course_id: MITx/6.00x/2013_Spring

- Split one attribute into three attributes for ease of querying:
  - Institution: MITx
  - Course_number: 6.00x
  - Course_term: 2013_Spring

- Updated data dictionary:
  - Institution: organization responsible for the course
  - Course_number: numbers and letters identifying the course
  - Course_term: season and year of course session

edX entity-relationship diagram



Key points from lesson

- Selection of entities and associated attributes from a flat file is not always obvious
- The data modeling process may reveal inconsistencies or errors in the data which will have to be corrected before importing into a database
- Foreign keys can be used as primary keys in a dependent entity if the keys uniquely identify records in the dependent entity