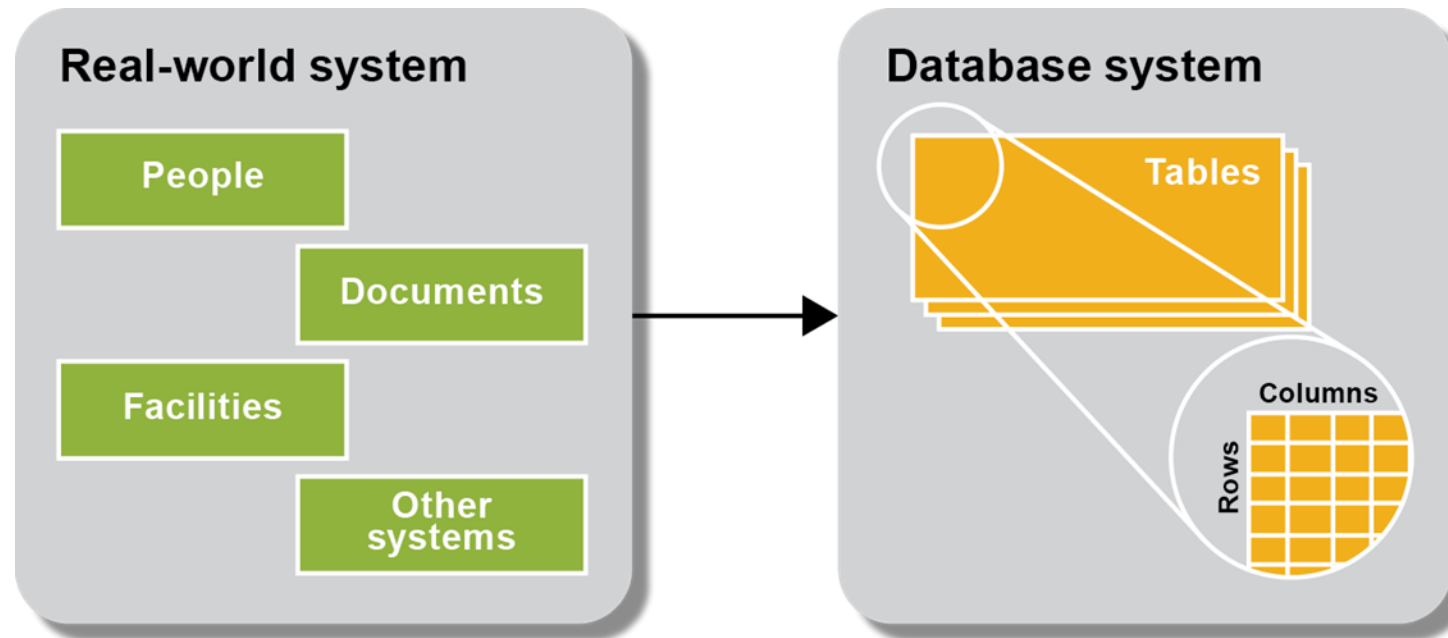


Database Design

- **Introduction to Data Management**
 - Some definitions to get started and for future reference
 - Note: terminology varies across texts, platforms and applications – concept is the same
 - Database basics
- **Database Design**
 - Use database specifications/real-world models to identify tables, columns and keys for a database
 - (E)ER diagrams, translating ER diagrams to relational DB schema
- **Data Normalization**
 - Rules of data normalization
 - Steps for normalizing database schema
 - Concepts of “tidy” data, transforming raw data to processed data

Database Design

A database is a model of a real-world system



Database Design

Acme Fabrication, Inc.		
Custom Contraptions, Contrivances and Confabulations		Invoice Number: I01-1088
1234 West Industrial Way East Los Angeles California 90022		Invoice Date: 10/05/11
800.555.1212 fax 562.555.1213 www.acmefabrication.com		Terms: Net 30

Part No.	Qty.	Description	Unit Price	Extension
CUST345	12	Design service, hr	100.00	1200.00
457332	7	Baling wire, 25x3ft roll	79.90	559.30
50173	4375	Duct tape, black, yd	1.09	4768.75
328771	2	Rubber tubing, 100ft roll	4.79	9.58
CUST281	7	Assembly, hr	75.00	525.00
CUST917	2	Testing, hr	125.00	250.00
		Sales Tax		245.20

Your salesperson:	Ruben Goldberg, ext 4512
Accounts receivable:	Inigo Jones, ext 4901

\$7,557.83

PLEASE PAY THIS AMOUNT

Thanks for your business!

The data elements on the invoice document

Vendor name	Item extension
Vendor address	Vendor sales contact name
Vendor phone number	Vendor sales contact extension
Vendor fax number	Vendor AR contact name
Vendor web address	Vendor AR contact extension
Invoice number	Invoice total
Invoice date	
Invoice terms	
Item part number	
Item quantity	
Item description	
Item unit price	

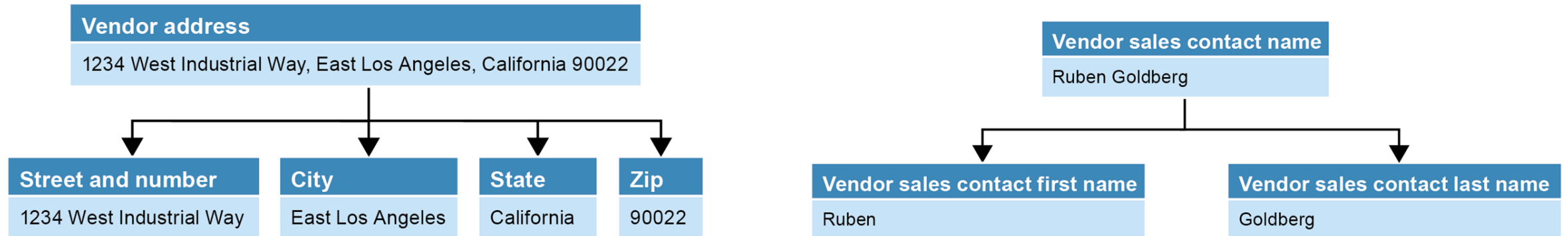
Actual invoice



List of data elements

Database Design

- Consider dividing some data fields into their components



Database Design

The data elements on the invoice document

Vendor name	Item extension
Vendor address	Vendor sales contact name
Vendor phone number	Vendor sales contact extension
Vendor fax number	Vendor AR contact name
Vendor web address	Vendor AR contact extension
Invoice number	Invoice total
Invoice date	
Invoice terms	
Item part number	
Item quantity	
Item description	
Item unit price	

Vendors	Invoices
Vendor name	Invoice number
Vendor address	Invoice date
Vendor city	Invoice terms
Vendor state	Invoice total
Vendor zip code	Line item part number(s)
Vendor phone number	Line item quantity(ies)
Vendor contact name	Line item description(s)
	Line item unit price(s)
	Line item extension(s)

List of data elements



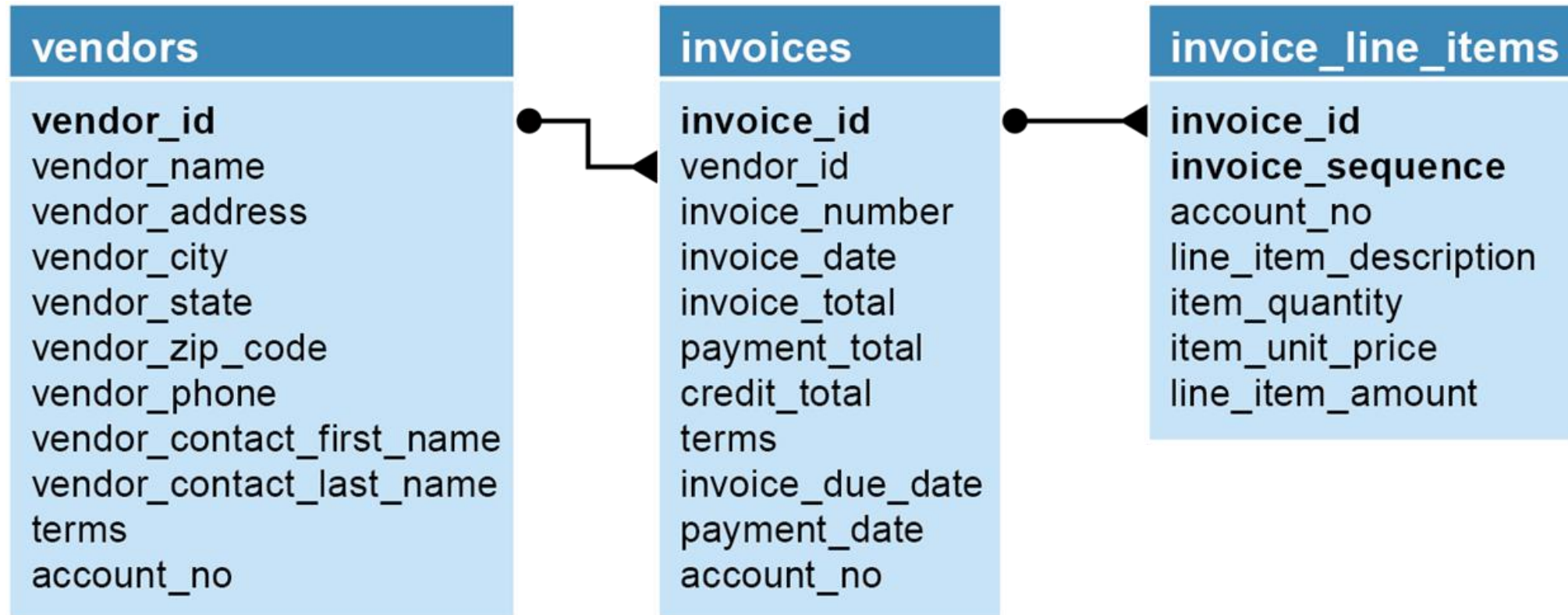
Tables and columns for DB

Database Design

Vendors	Invoices	Invoice Line Items
Vendor name	Invoice number	Invoice number
Vendor address	Invoice date	Line item part number
Vendor city	Invoice terms	Line item quantity
Vendor state	Invoice total	Line item description
Vendor zip code		Line item unit price
Vendor phone number		Line item extension
Vendor contact name		

Split Invoice data into two tables – one invoice may have several line items

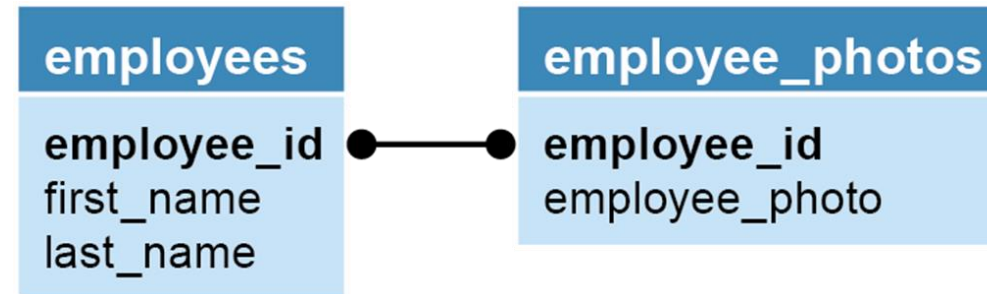
Database Design



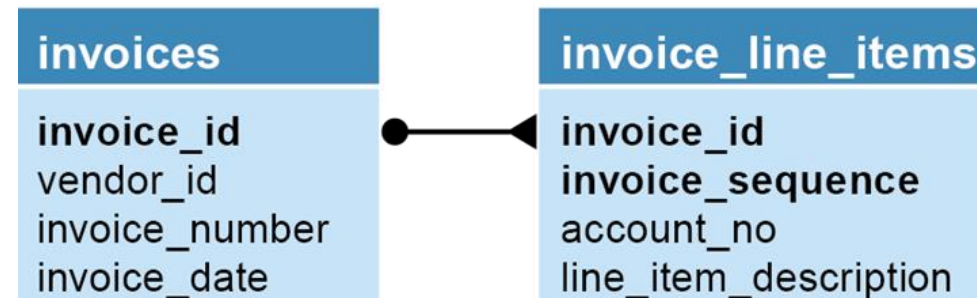
Add fields where needed (vendor_id, invoice_sequence, etc.) for identification and to link tables

Table Relationships

- One-to-one



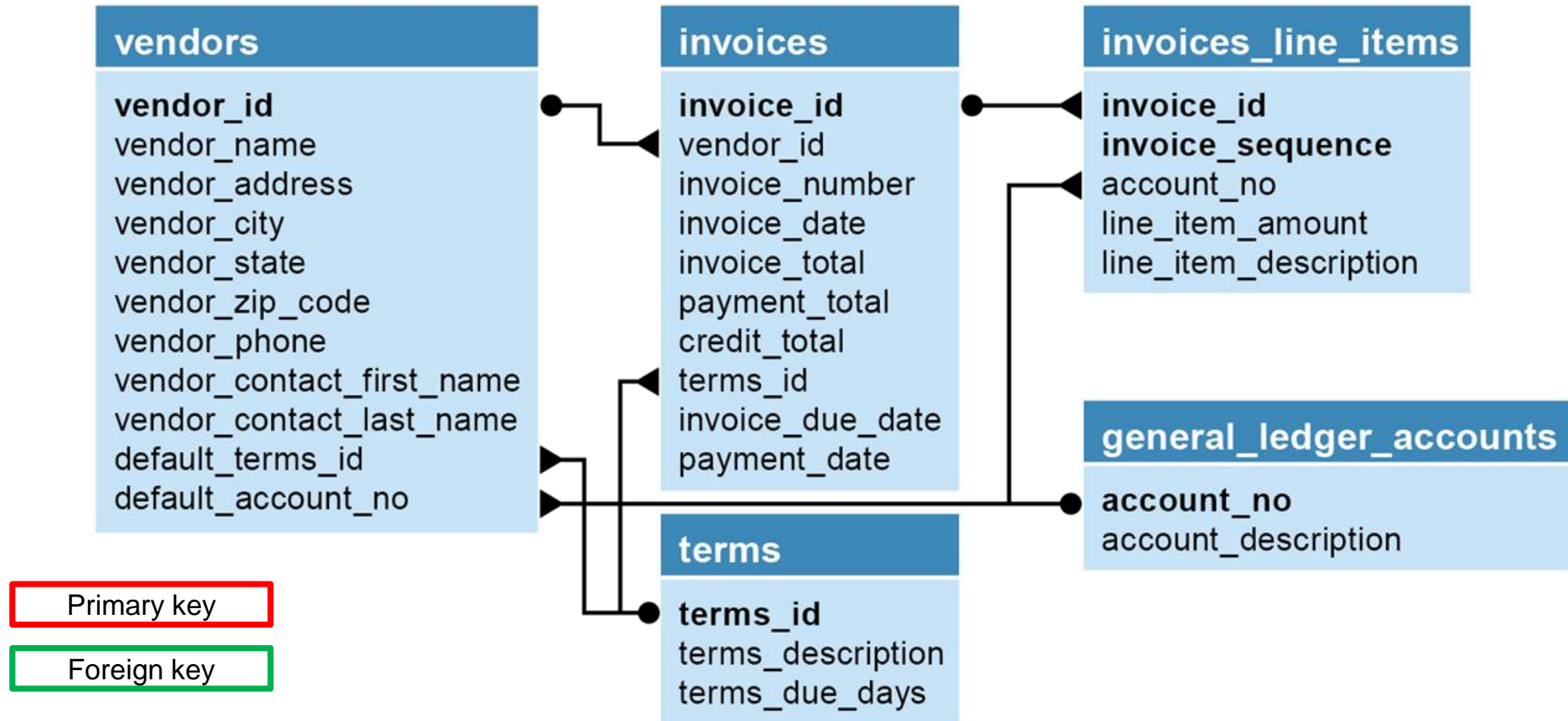
- One-to-many



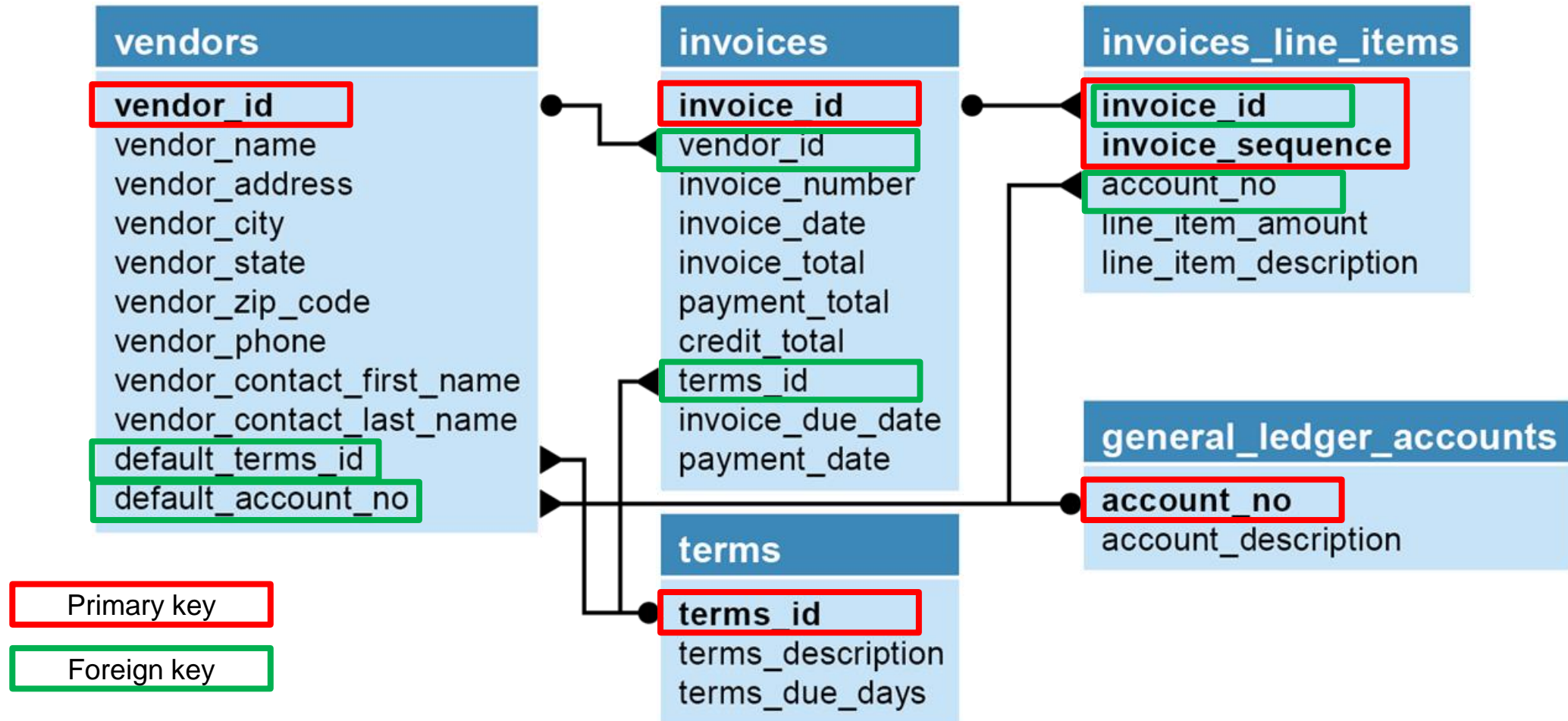
- Many-to-many



Primary Keys & Foreign Keys



Primary Keys & Foreign Keys



Referential Integrity

- Referential integrity can be violated when
 - A row from the primary key table is deleted or the value of a primary key is updated
 - A row in the foreign key table is inserted or the value of a foreign key is updated
- MySQL enforces referential integrity (called “Declarative Referential Integrity”) using primary and foreign key constraints
 - Cause an error on insert/update/delete operation when necessary on primary/foreign keys.
 - Other behavior is possible using the ON DELETE clause in the CONSTRAINT
 - default: RESTRICT (same as NO ACTION), other options: SET NULL, CASCADE

Keys vs. Indexes

- Recall: keys are used to identify rows in table and to facilitate linking across tables
- Keys (indexes with constraints)
 - Primary – values in column are unique & not-null
 - Unique – values in column are unique (null values are allowed)
 - Foreign – values in column reference a column (primary key) in another table
- Indexes (set of ordered references to rows of a table)
 - All keys are automatically indexed by MySQL
 - Additional indexes can be defined to improve performance
 - ♦ Read-efficiency vs. write-efficiency considerations

Some Considerations Regarding Keys

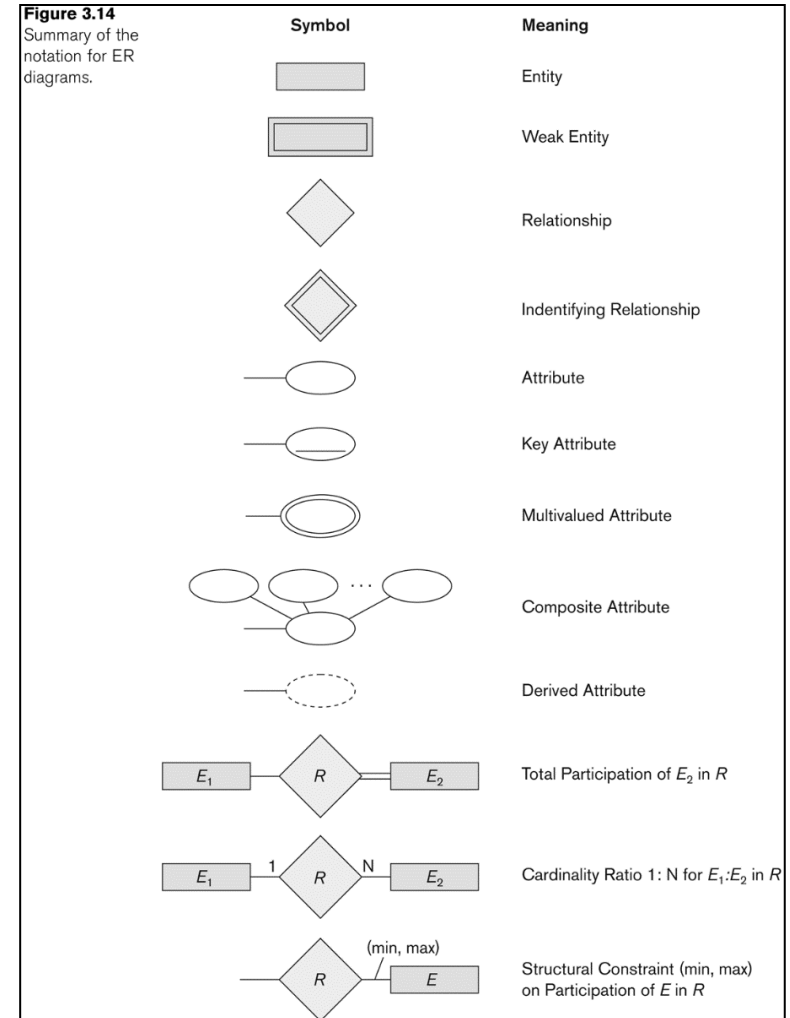
- What field (or combination of fields) uniquely identifies a row in your table?
 - Watch for potential duplicates – two students with the same last name? last name and first name? Birthdate? Etc.
- Which rows in separate tables will need to be linked later for reporting/analysis purposes?

Database Design:

ER Diagram -> Relational DB Schema

Consider

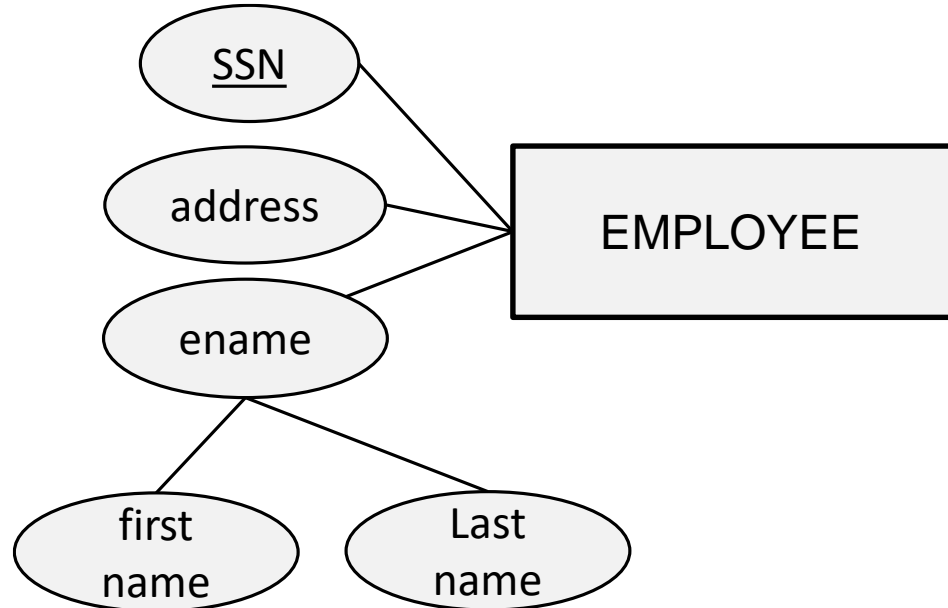
- Entities
- Attributes
- Relationships



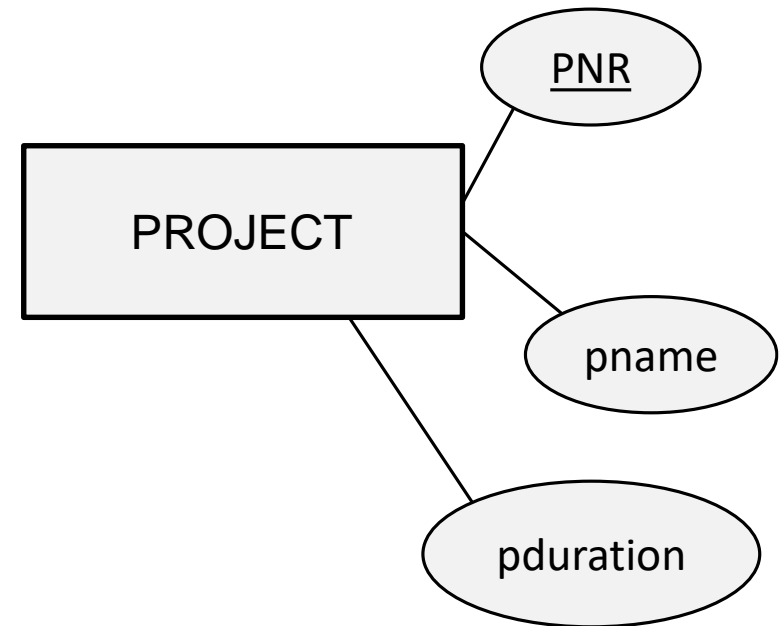
Vocabulary Cross-Reference

EER model	Relational Model
Entity type (or just Entity)	Relation, Table
Entity	Tuple, Row, Record
Attribute type (or just Attribute)	Column, Field
Attribute	Cell
Relationship	Primary & Foreign Keys

ER Entities & Attributes



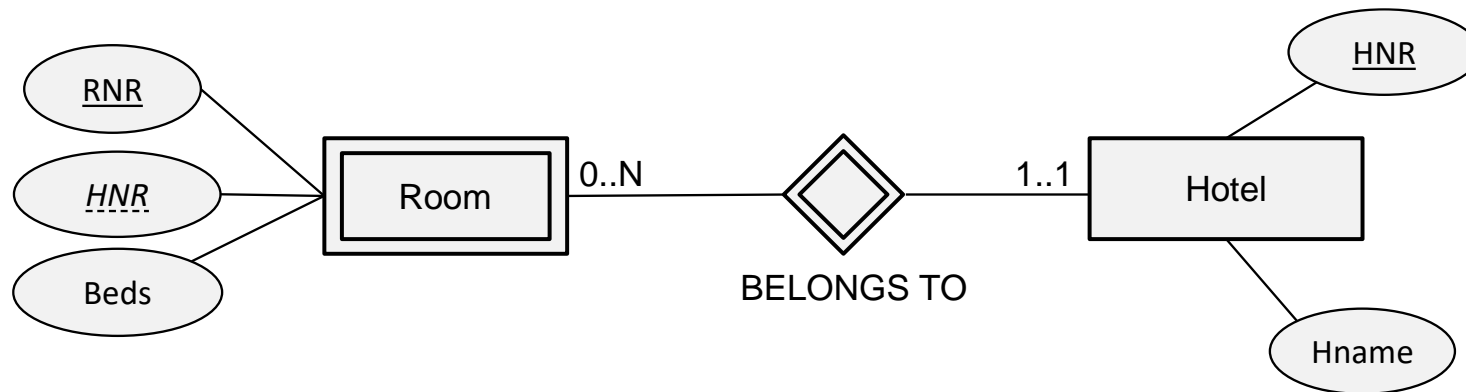
EMPLOYEE
SSN
address
first name
last name



PROJECT
PNR
pname
pduration

ER Relationships and Cardinalities

- Relationship cardinalities specify the minimum or maximum number of relationship instances that an individual entity can participate in
- Minimum cardinality can be 0 or 1
 - If 0: partial participation
 - If 1: total participation or existence dependency
- Maximum cardinality can be 1 or N
- Relationship types are often characterized by their maximum cardinalities
 - 4 options for binary relationship types: 1:1, 1:N, N:1 and M:N.



University DB Example

- University model has
 - Students who each have a name and a student ID. Students may register for multiple classes.
 - Courses which have a course number, course title and credit hours. Courses may have one prerequisite course.
 - Course Sections which are instances of Courses that have assigned days of the week and room numbers.
 - Professors who each have a name and department. Professors may teach multiple classes.

University DB Example

1. Entities

Students

Courses

Professors

2. Attributes

3. Relationships

University DB Example

1. Entities

2. Attributes

Students	Courses	Course Sections	Professors
Student ID Name	Course Number Course Title Credit Hours Prerequisite	Course Number Days of Week Room Number	Faculty ID Name Department

3. Relationships

University DB Example

1. Entities
2. Attributes
3. Relationships

Students
Student ID
First Name
Last Name

Course Section
Section Number
Course ID
Days
Room
Faculty ID

Course
Course Number
Title
Credit Hours
Prerequisite Course

Professor
Faculty ID
First Name
Last Name
Department

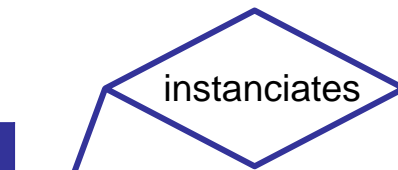
University DB Example

1. Entities
2. Attributes
3. Relationships

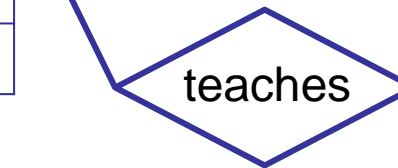
Students
Student ID
First Name
Last Name



Course Section
Section Number
Course ID
Days
Room
Faculty ID



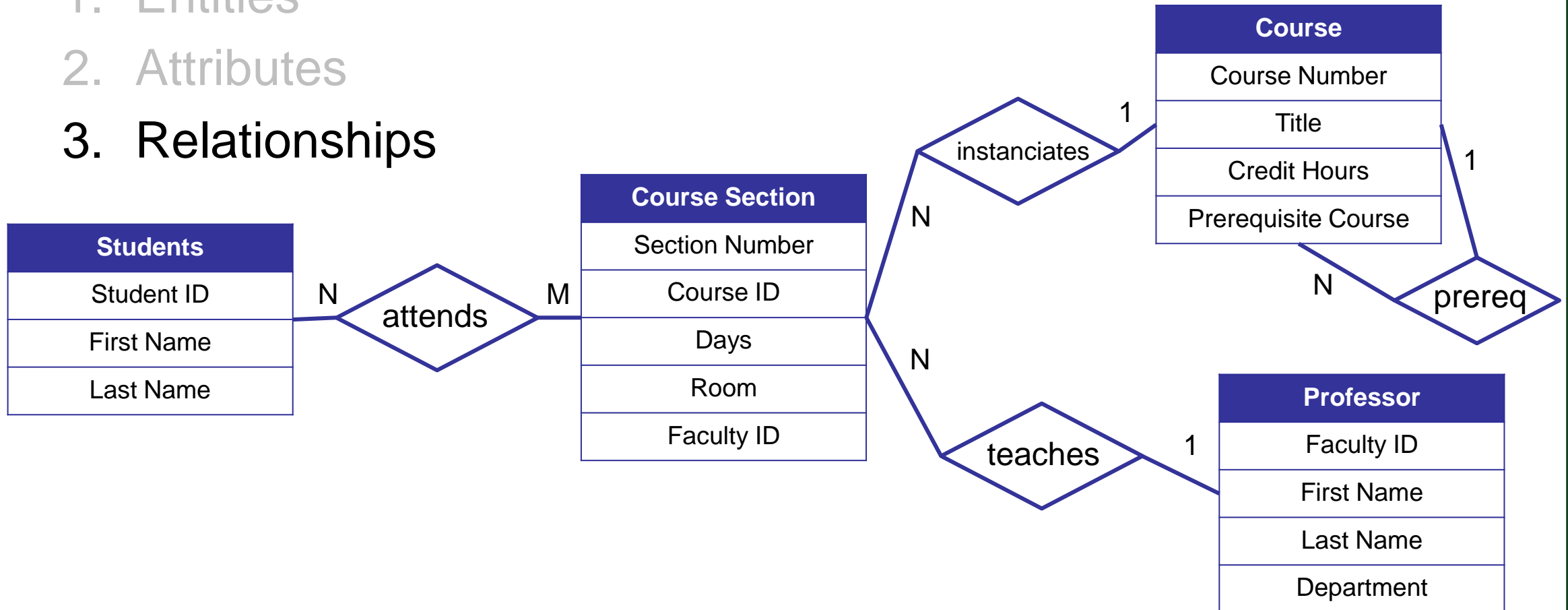
Course
Course Number
Title
Credit Hours
Prerequisite Course



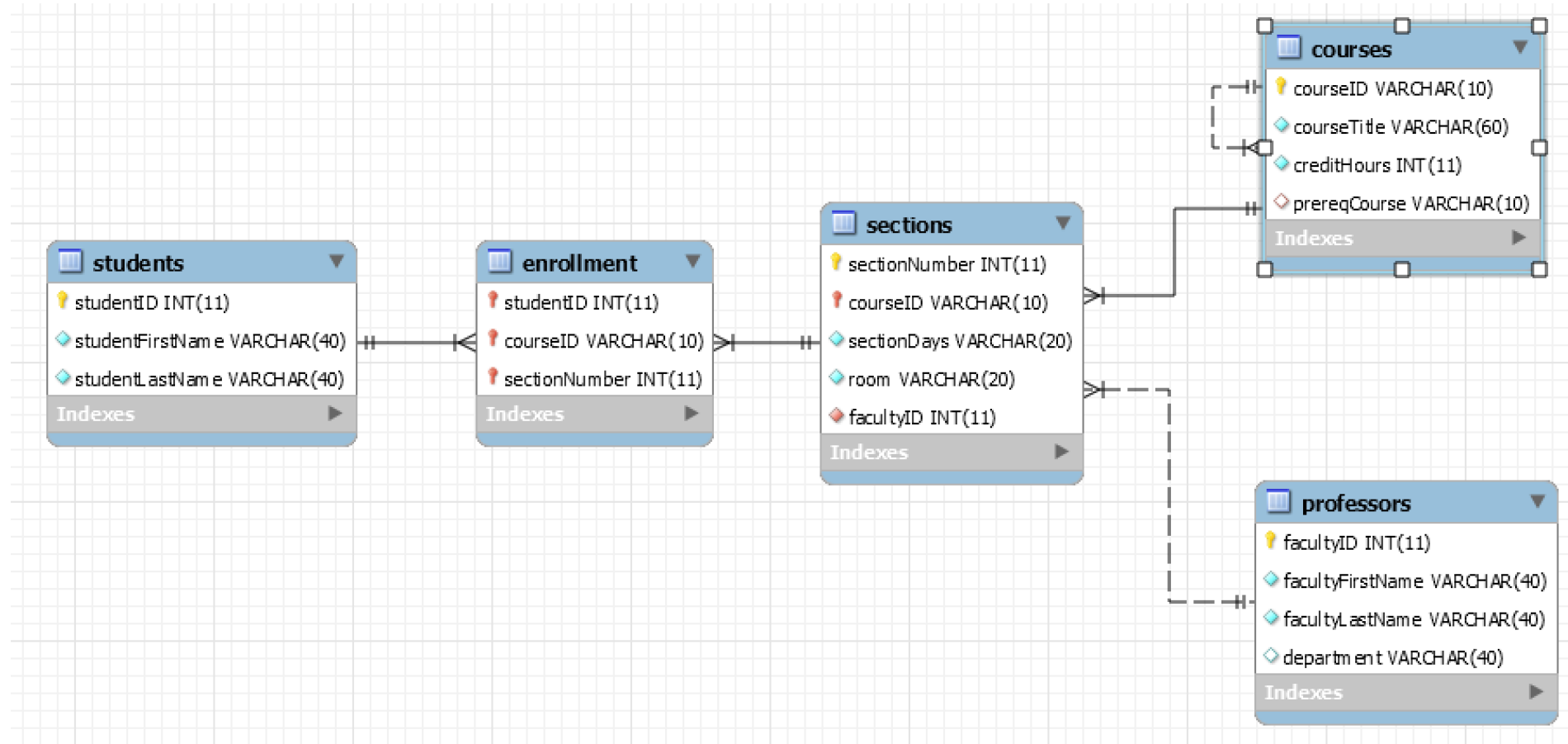
Professor
Faculty ID
First Name
Last Name
Department

University DB Example

1. Entities
2. Attributes
3. Relationships



University DB Example

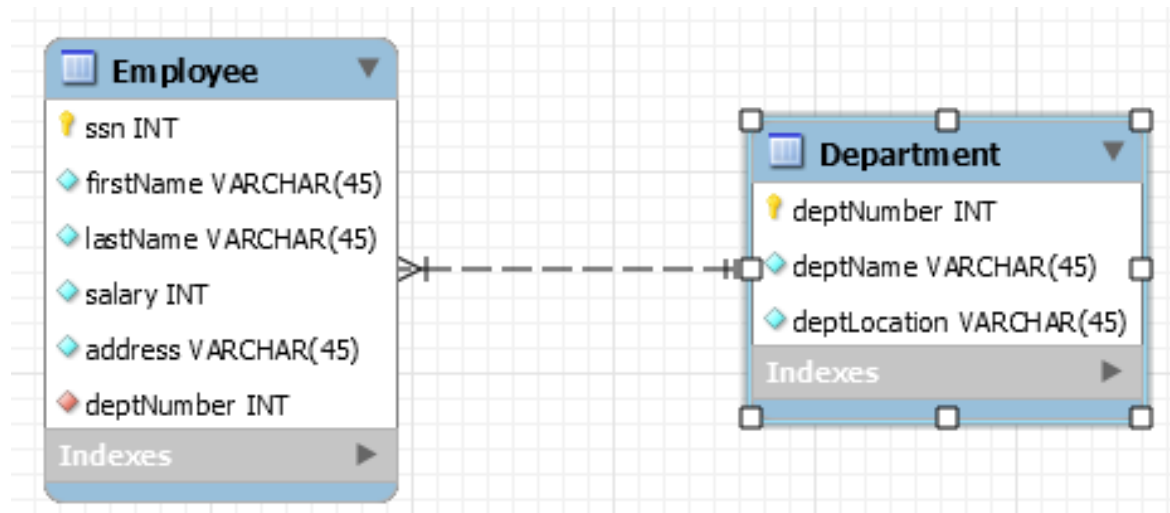
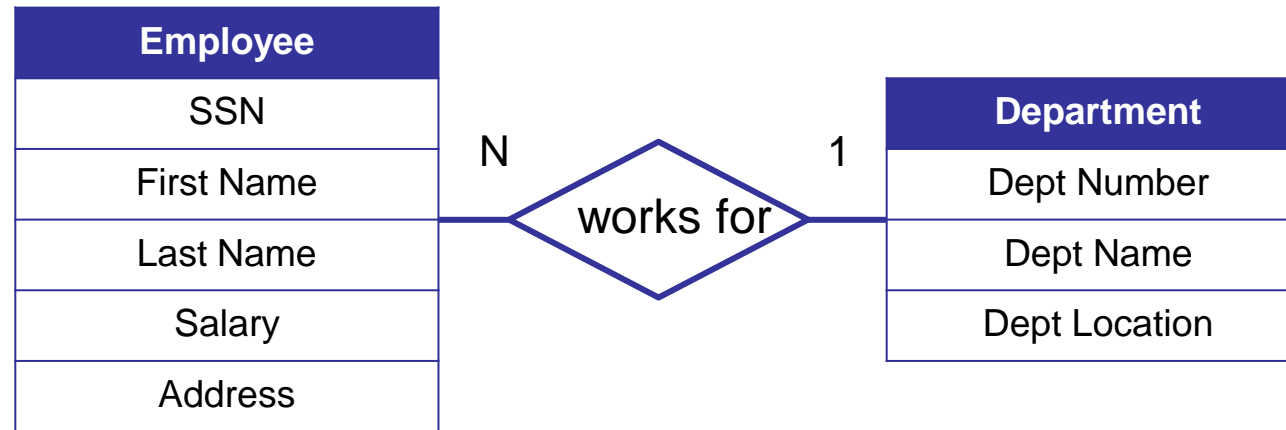


ER -> relational DB schema: many-to-many relationships become junction ("linking") tables

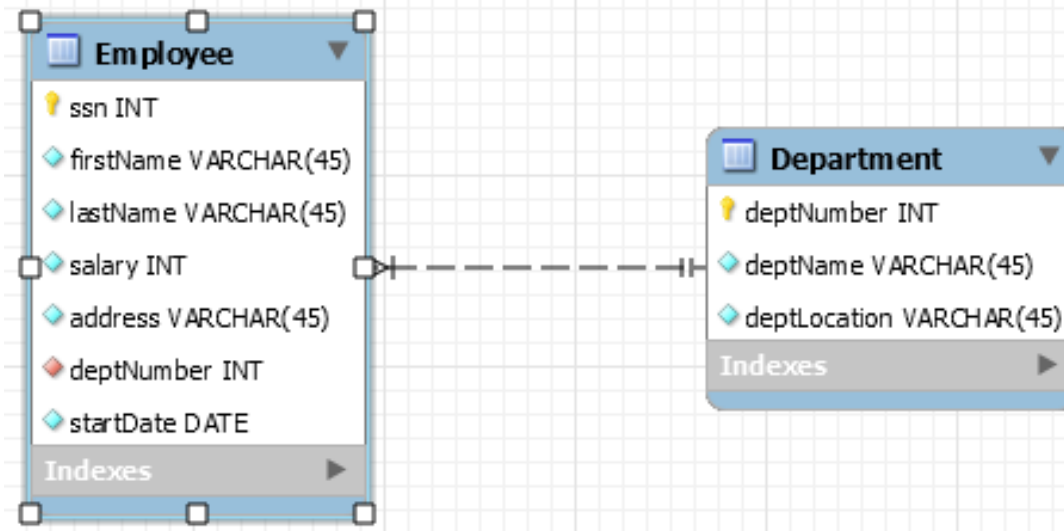
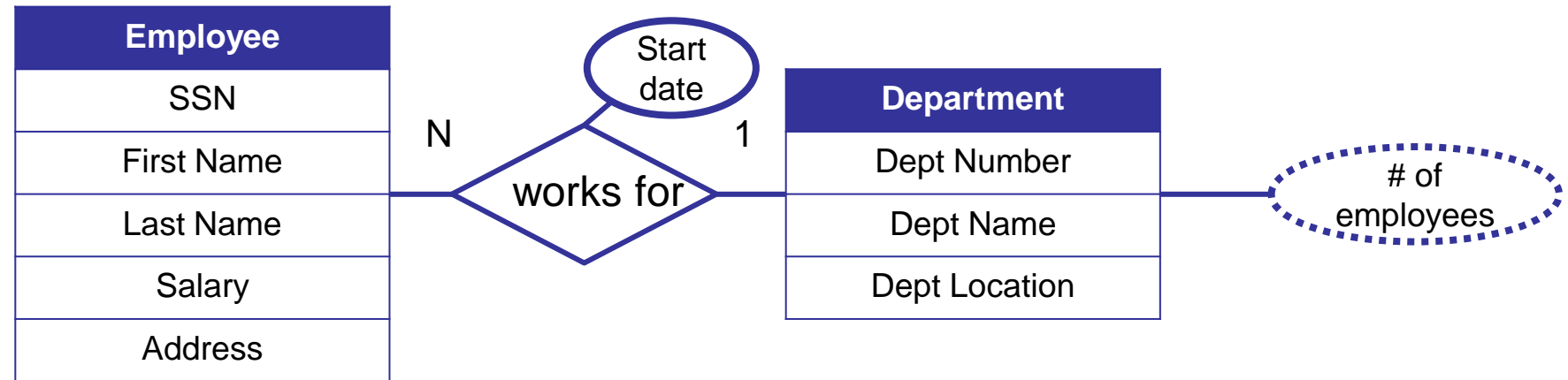
ER Diagram -> Relational DB Schema

- Entities become tables
- Entity attributes become table columns
- Relationships
 - Become foreign key constraints
 - Many-to-many relationships become junction (“linking”) tables
- Relationship attributes
 - Can become columns in either entity table for one-to-one relationships
 - Can become columns on the N-side entity table for many-to-one relationships
 - Must become columns in the junction table for many-to-many relationships

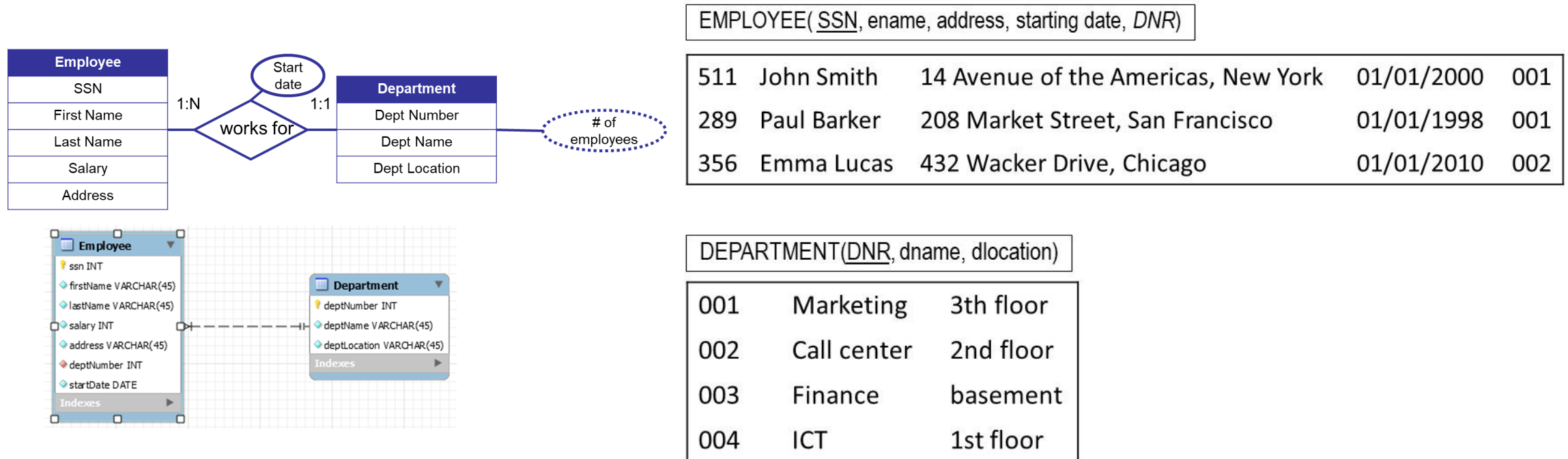
1:N Relationships -> Relational DB Schema



1:N Relationships -> Relational DB Schema

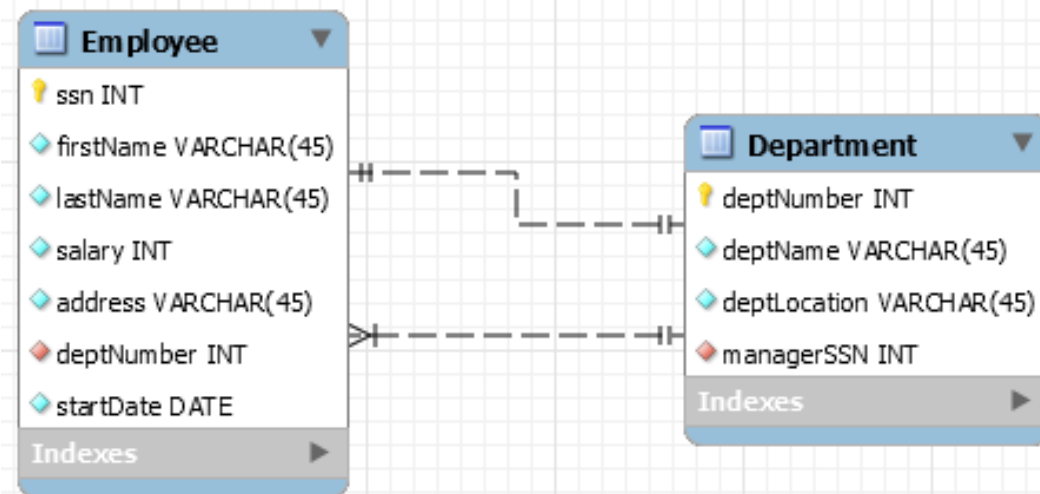
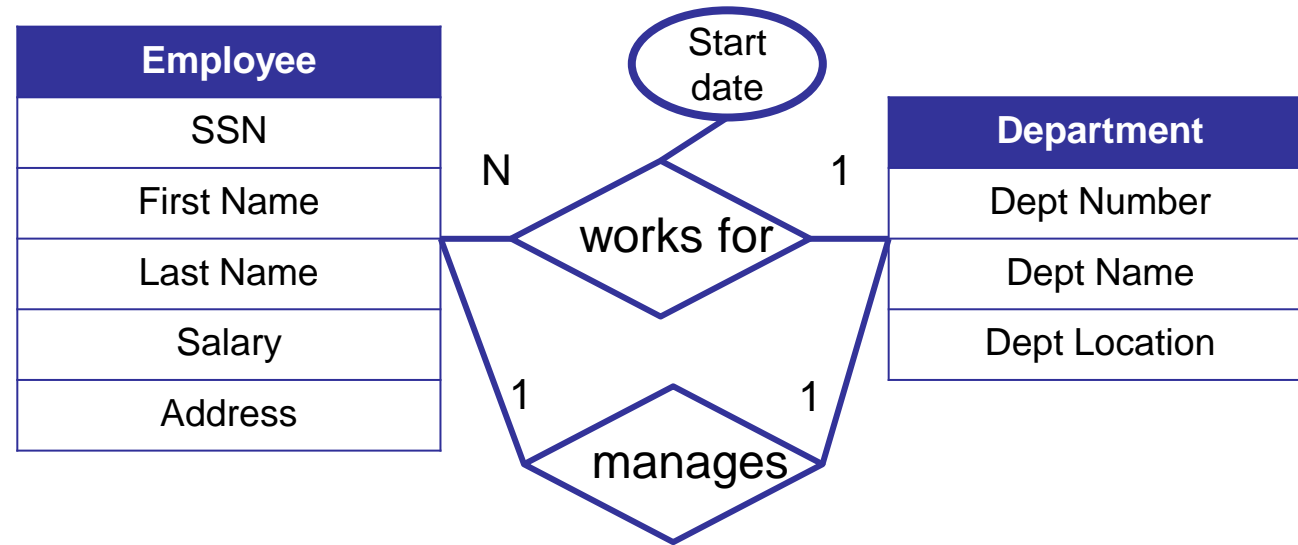


1:N Relationships -> Relational DB Schema



- Foreign key in Employee table (N-side of the relationship) links to primary key in Department (1-side)
- Foreign key declared NOT NULL to capture minimum of one department per employee
- Semantics lost: minimum of one employee per department

1:1 Relationships -> Relational DB Schema



1:1 Relationships -> Relational DB Schema

EMPLOYEE(SSN, ename, address, DNR)

511	John Smith	14 Avenue of the Americas, New York	001
289	Paul Barker	208 Market Street, San Francisco	003
356	Emma Lucas	432 Wacker Drive, Chicago	NULL
412	Michael Johnson	1134 Pennsylvania Avenue, Washington	NULL
564	Sarah Adams	812 Collins Avenue, Miami	001

DEPARTMENT(DNR, dname, dlocation)

001	Marketing	3th floor
002	Call center	2nd floor
003	Finance	basement
004	ICT	1st floor

EMPLOYEE(SSN, ename, address)

511	John Smith	14 Avenue of the Americas, New York
289	Paul Barker	208 Market Street, San Francisco
356	Emma Lucas	432 Wacker Drive, Chicago

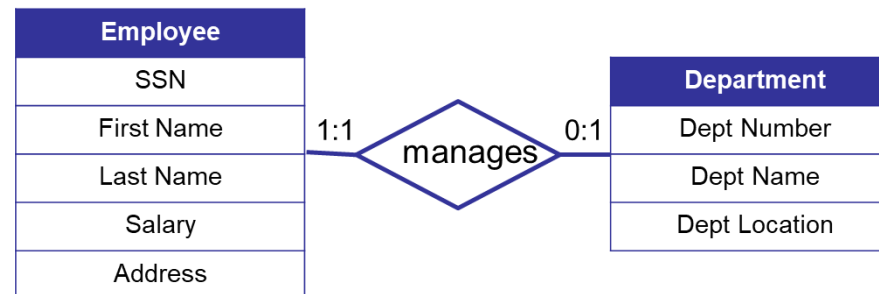
DEPARTMENT(DNR, dname, dlocation, SSN)

001	Marketing	3th floor	511
002	Call center	2nd floor	511
003	Finance	basement	289
004	ICT	1st floor	511

VS.

Semantics lost:

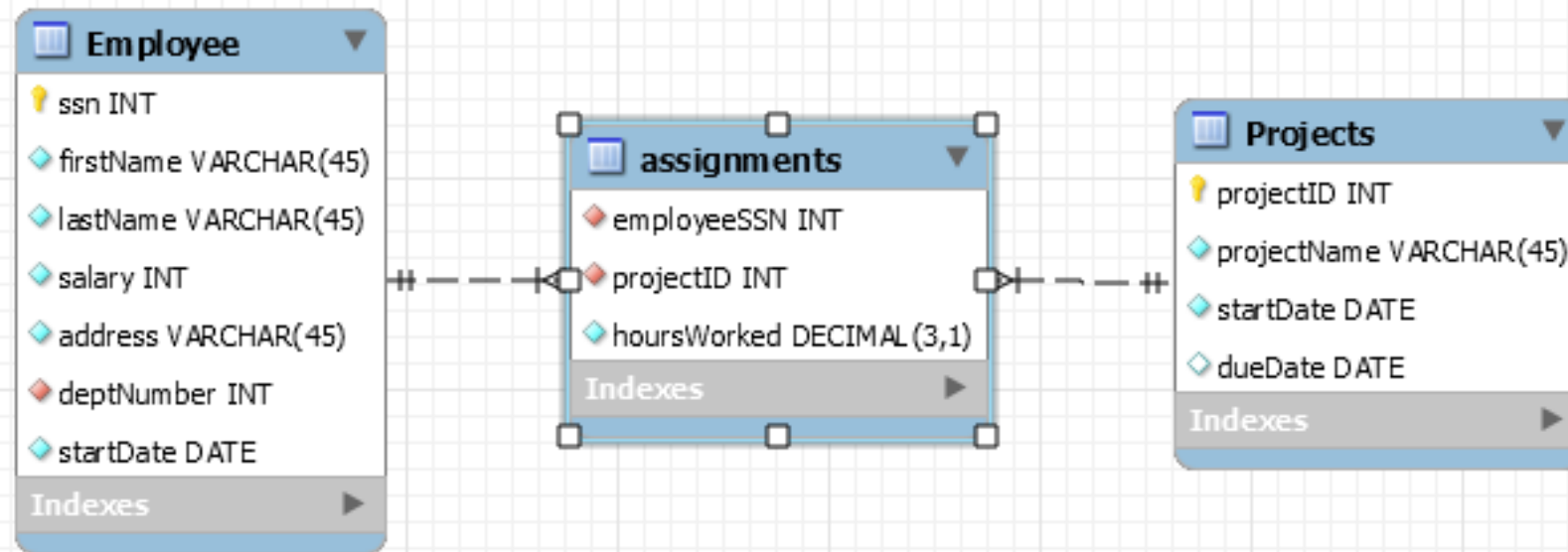
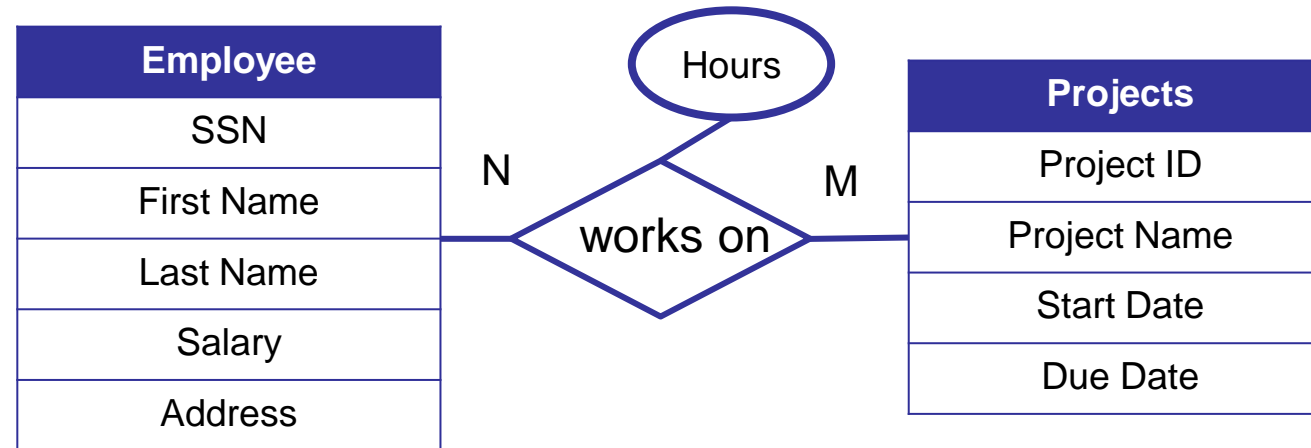
- max of one manager per dept
- min of 1 manager per dept



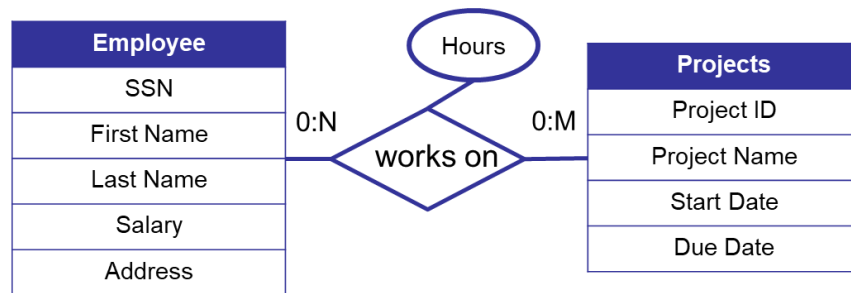
Semantics lost:

- max of one dept per manager

N:M Relationships -> Relational DB Schema



N:M Relationships -> Relational DB Schema



EMPLOYEE(SSN, ename, address, DNR)

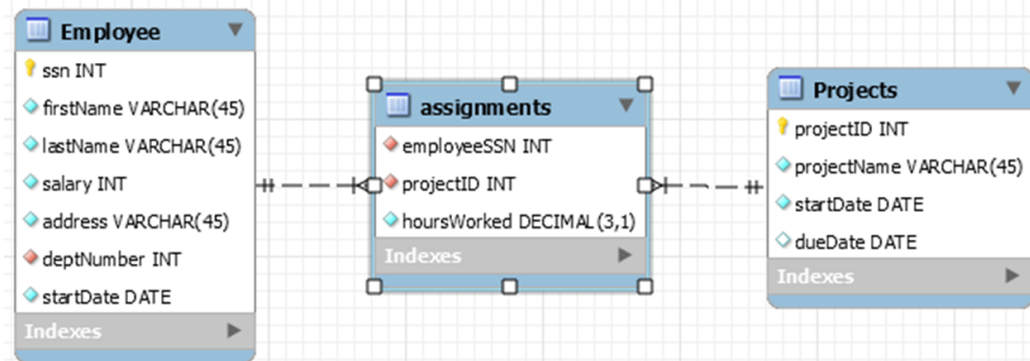
511	John Smith	14 Avenue of the Americas, New York	001
289	Paul Barker	208 Market Street, San Francisco	001
356	Emma Lucas	432 Wacker Drive, Chicago	002

PROJECT(PNR, pname, pduration)

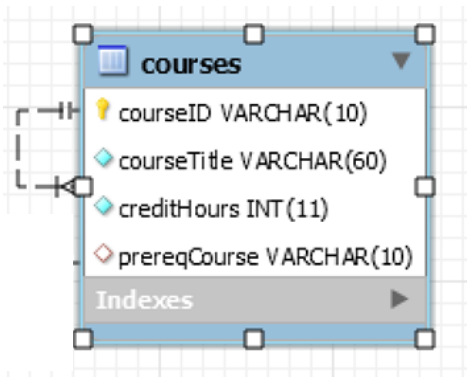
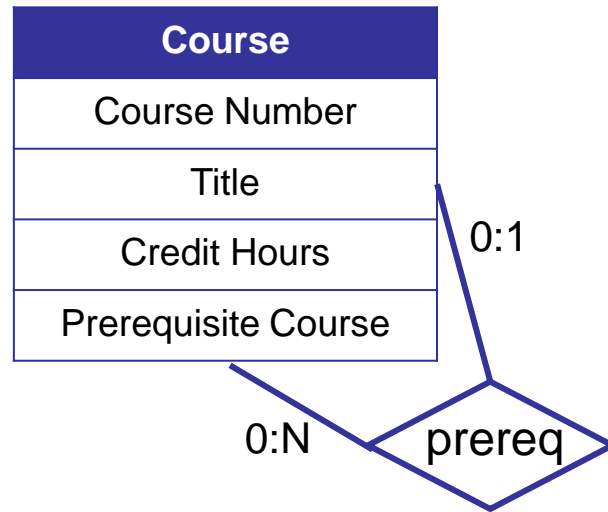
1001	B2B	100
1002	Analytics	660
1003	Web site	52
1004	Hadoop	826

WORKS_ON(SSN, PNR, hours)

511	1001	10
289	1001	80
289	1003	50

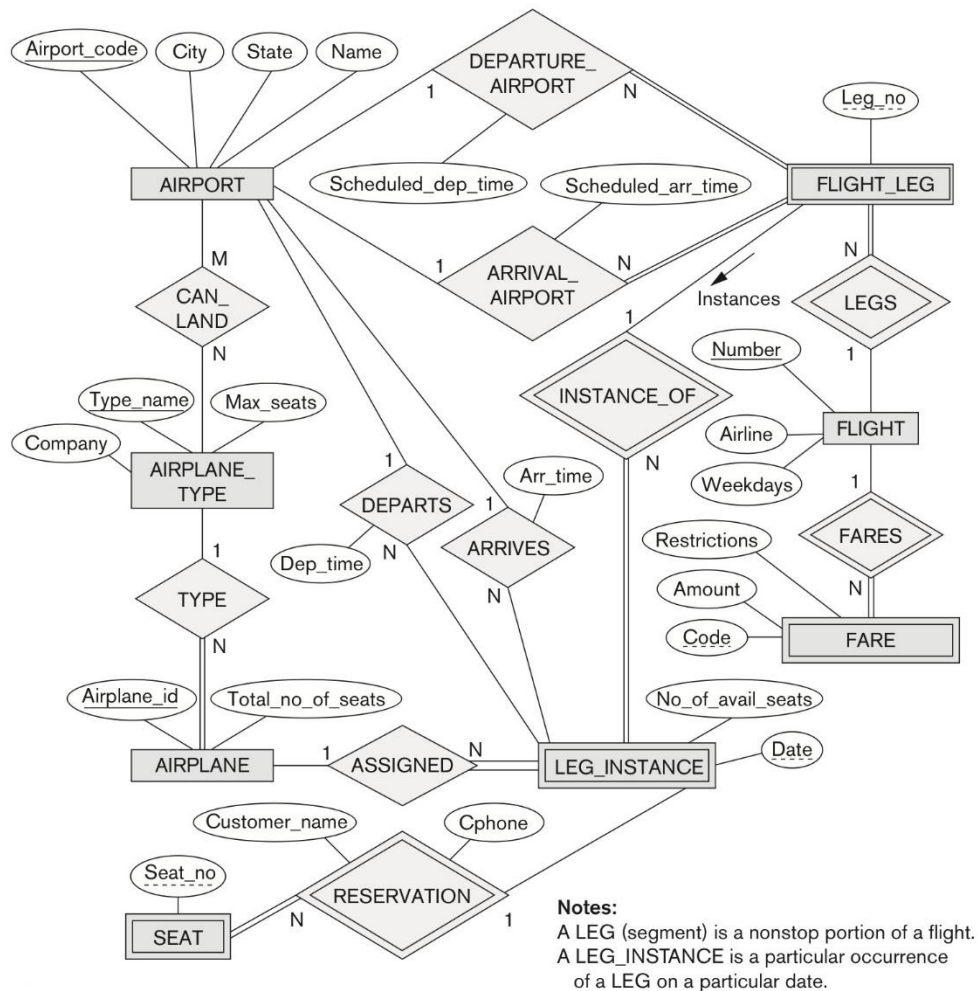


Unary Relationships -> Relational DB Schema



courseID	courseTitle	creditHours	prereqCourse
CAS100	Effective Speech	3	
CHEM101	Introductory Chemistry	3	
CHEM111	Experimental Chemistry I	1	CHEM101
ENGL015	Rhetoric and Composition	3	
ENGL202	Effective Writing	3	
HDFS129	Introduction to Human Development...	3	
MICRB106	Elementary Microbiology	3	
MICRB107	Elementary Microbiology Laboratory	1	
NURS200	Introduction to Nursing Research	3	
NURS205	Introduction to Pharmacological Con...	3	
NURS215	Health: Introduction to Wellness	3	
NURS225	Health: Introduction to Illness	3	
NURS230	Health: Nursing Process	4	
NURS245	Violence and the Impact on Society	3	
NURS301	Nursing Care of Client through the Ad...	4	NURS225
NURS310	Nursing Care of the Elderly	4	NURS225
NURS320	Nursing Care of Young Adults	4	NURS225
NURS352	Advanced Health Assessment for the ...	3	

Airline Reservations System



- Each AIRPORT has a unique AirportCode, the AIRPORT Name, and the City and State in which the AIRPORT is located.
- Each airline FLIGHT has a unique number, the Airline for the FLIGHT, and the Weekdays on which the FLIGHT is scheduled (for example, every day of the week except Sunday can be coded as X7).
- A FLIGHT is composed of one or more FLIGHT LEGs (for example, flight number CO1223 from New York to Los Angeles may have two FLIGHT LEGs: leg 1 from New York to Houston and leg 2 from Houston to Los Angeles). Each FLIGHT LEG has a DEPARTURE AIRPORT and Scheduled Departure Time, and an ARRIVAL AIRPORT and Scheduled Arrival Time.
- A LEG INSTANCE is an instance of a FLIGHT LEG on a specific Date (for example, CO1223 leg 1 on July 30, 1989). The actual Departure and Arrival AIRPORTs and Times are recorded for each flight leg after the flight leg has been concluded. The Number of available seats and the AIRPLANE used in the LEG INSTANCE are also kept.
- The customer RESERVATIONS on each LEG INSTANCE include the Customer Name, Phone, and Seat Number(s) for each reservation.
- Information on AIRPLANES and AIRPLANE TYPEs are also kept. For each AIRPLANE TYPE (for example, DC-10), the TypeName, manufacturing Company, and Maximum Number of Seats are kept. The AIRPORTs in which planes of this type CAN LAND are kept in the database. For each AIRPLANE, the AirplaneId, Total number of seats, and TYPE are kept.