SWE 207
Database Management Systems

~ Business Rules, Data Models, Entity Relationship Model ~

CELAL ÇEKEN,

GÖZDE YOLCU ÖZTEL,

VEYSEL HARUN ŞAHİN

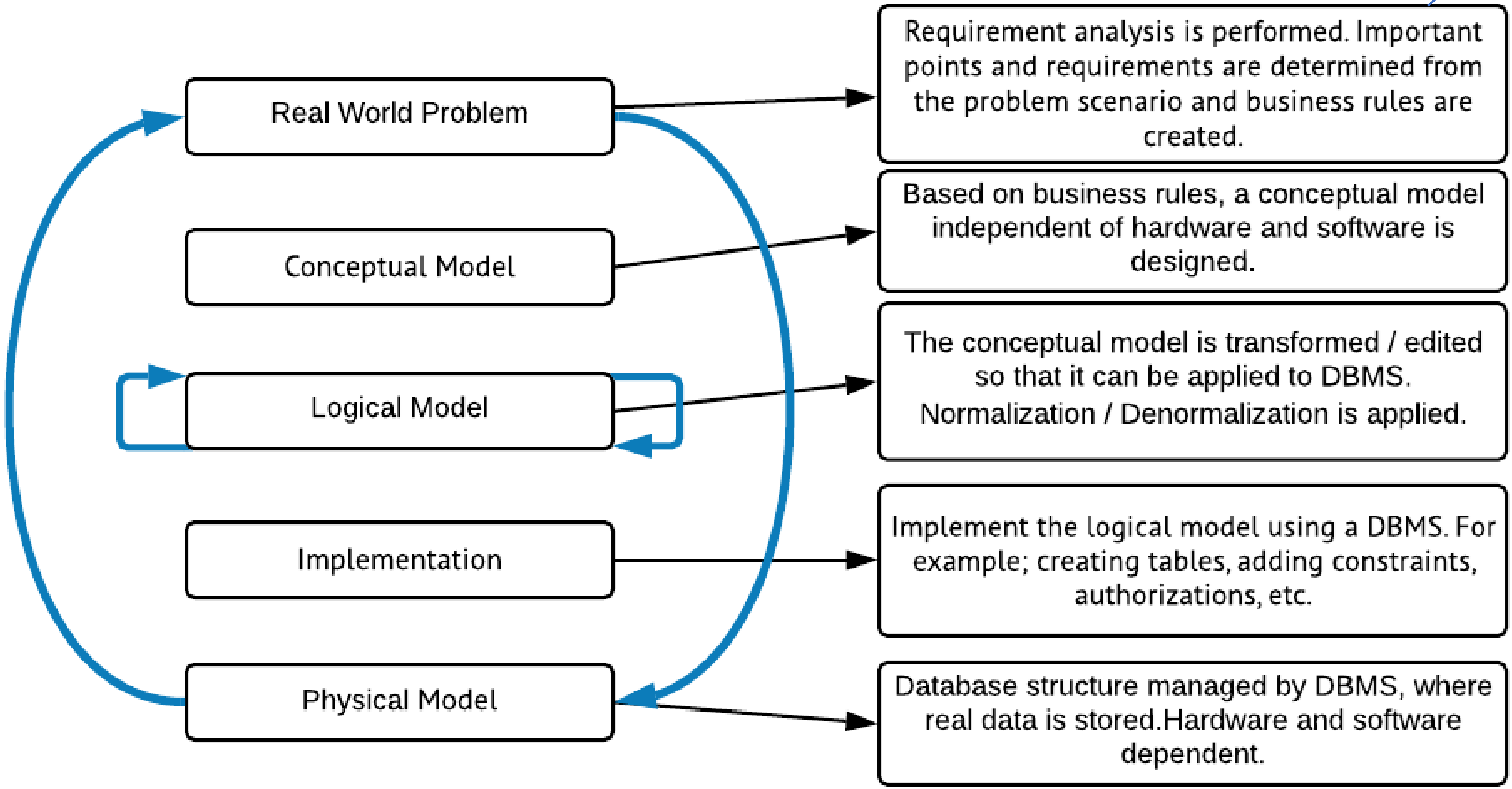
SAKARYA ÜNİVERSİTESİ

Content

- Database development lifecycle
- Business rules
- Basic components of data model
- Converting business rules to a data model
- Development of data models
  - File system, Hierarchical model, Network model, Relational model, Entity Relationship model, Object oriented model,
- Data abstraction
- Entity relationship model
  - Basic concepts, Relationship between entity, Existence dependency, Weak / Strong Relationship, Relationship degrees, Design with entity relationship model (ERM)

Database development lifecycle

Which data?
Relationsh btw data?
Constraints?



Real World Problem

Conceptual Model

Logical Model

Implementation

Physical Model

Requirement analysis is performed. Important points and requirements are determined from the problem scenario and business rules are created.

Based on business rules, a conceptual model independent of hardware and software is designed.

The conceptual model is transformed / edited so that it can be applied to DBMS. Normalization / Denormalization is applied.

Implement the logical model using a DBMS. For example; creating tables, adding constraints, authorizations, etc.

Database structure managed by DBMS, where real data is stored.Hardware and software dependent.

Business rules

- When creating the database (entity, attribute, relationship and constraints), business rules are taken into account.
- Business rules can be defined with the summarized form of the rule or regulation of the organization whose database will be designed.
- Sample business rules:
    - A customer can place many orders.
    - Name, surname and phone number of each customer are requested.
    - Students take one midterm exam and one final exam.
- The source of business rules can be end users, administrators, rule makers and written documents (standards, regulations, etc.).

Business rules

- Meeting directly with the end users to create business rules is a highly effective solution.
- Importance of business rules in terms of database design:
    - It provides communication between users and designers.
    - It enables the designer to understand the nature, importance and scope of the data.
    - It enables the designer to understand the business processes.
    - It helps the designer to develop an accurate data model (allows easy determination of data relationships and constraints).
    - It uniforms the organization's view of data.
- After the business rules are created, the stage of modeling the database is started.

Data model

- Data model: The tool used for simple representation (usually graphical) of complex real-world data structures is called a data model.
- The data model eases communication between database designers, application programmers and end users.
- Database design becomes easier owing to data models.
- Data modeling is an iterative process. First, a simple model is created. Details are added later. Finally, the blueprint template used in database design is obtained.

Basic components of data model

- Entity: An entity is everything that data can be collected and stored about it (student, course, staff, etc.). Refers to objects in the real world. Everything that exists and can be distinguished from its similar.
- Entity set: It is the set of entities of the same type (Students, Lessons, etc.).
- Attribute: The properties of the entity.
- Relationship: Expresses the relationship between entity.

Basic components of data model

- Relationship:
    - One to Many - 1:M
        - A customer can place many orders.
        - Each order is placed by only one customer.
    - Many to Many - M:N
        - Students can take many courses.
        - Each course can be taken by many students.
    - One to One - 1:1
        - A store is managed by an employee.
        - An employee manages a store.

Basic components of data model

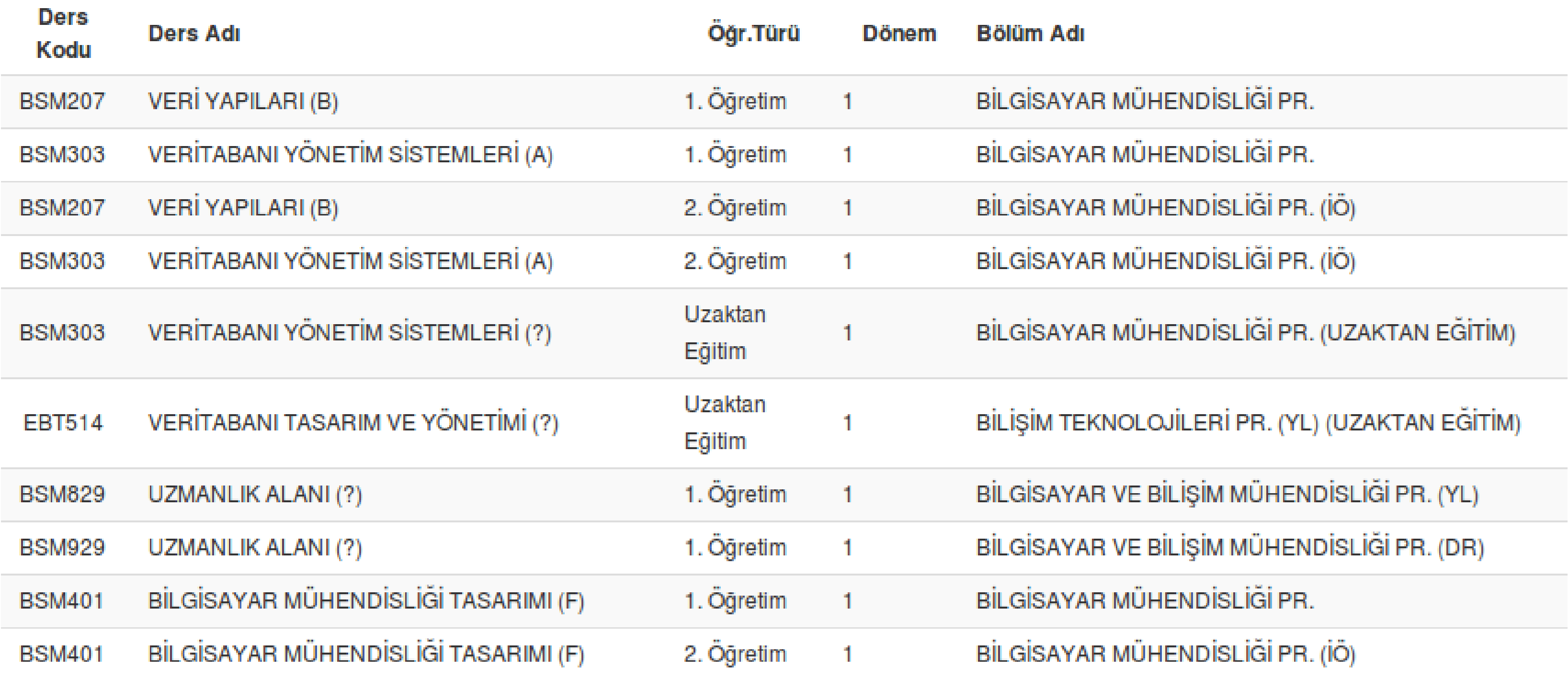
- Constraints:
    - They are the limitations on the data. It is important in terms of ensuring data integrity. For example;
        - Student grade can be between 0-100.
        - The identification number can be 11 characters.
        - The same product cannot be registered more than once.

Development of data models

- File system,
- Hierarchical model,
- Network model,
- Relational model,
- Entity Relationship model,
- Object oriented model,

Development of data models

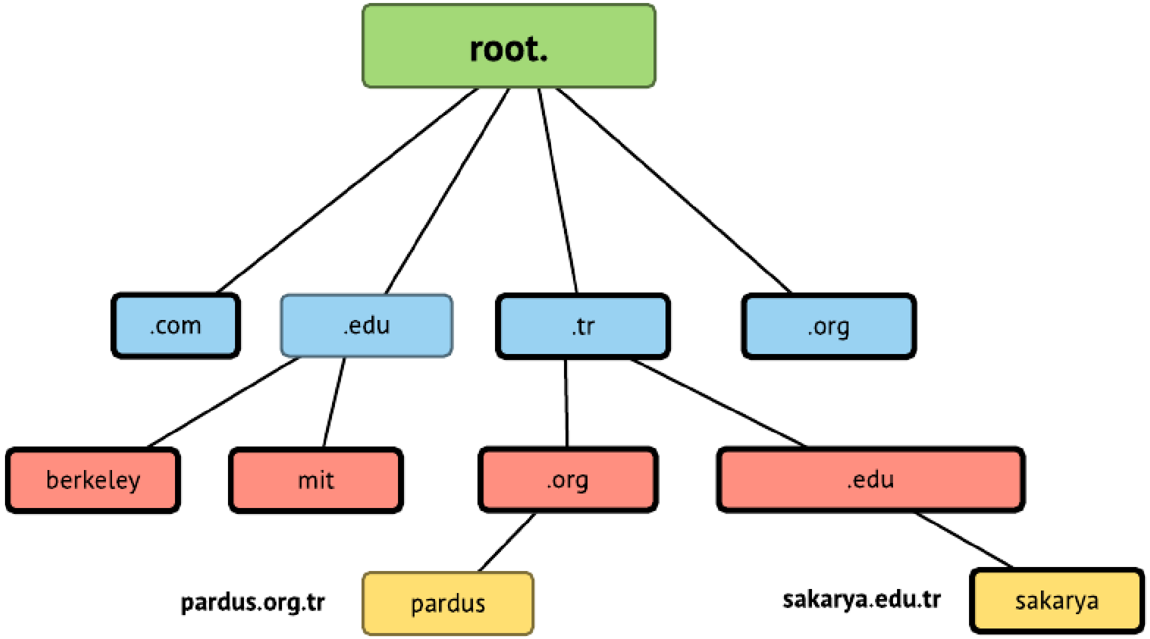
- File system
  - It was mostly used in IBM mainframe systems in the 1960-1970s.
  - There is no relationship between files.
  - A sample file structure can be seen below.



Ders Kodu	Ders Adı	Öğr.Türü	Dönem	Bölüm Adı
BSM207	VERİ YAPILARI (B)	1. Öğretim	1	BİLGİSAYAR MÜHENDİSLİĞİ PR.
BSM303	VERİTABANI YÖNETİM SİSTEMLERİ (A)	1. Öğretim	1	BİLGİSAYAR MÜHENDİSLİĞİ PR.
BSM207	VERİ YAPILARI (B)	2. Öğretim	1	BİLGİSAYAR MÜHENDİSLİĞİ PR. (İÖ)
BSM303	VERİTABANI YÖNETİM SİSTEMLERİ (A)	2. Öğretim	1	BİLGİSAYAR MÜHENDİSLİĞİ PR. (İÖ)
BSM303	VERİTABANI YÖNETİM SİSTEMLERİ (?)	Uzaktan Eğitim	1	BİLGİSAYAR MÜHENDİSLİĞİ PR. (UZAKTAN EĞİTİM)
EBT514	VERİTABANI TASARIM VE YÖNETİMİ (?)	Uzaktan Eğitim	1	BİLİŞİM TEKNOLOJİLERİ PR. (YL) (UZAKTAN EĞİTİM)
BSM829	UZMANLIK ALANI (?)	1. Öğretim	1	BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ PR. (YL)
BSM929	UZMANLIK ALANI (?)	1. Öğretim	1	BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ PR. (DR)
BSM401	BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI (F)	1. Öğretim	1	BİLGİSAYAR MÜHENDİSLİĞİ PR.
BSM401	BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI (F)	2. Öğretim	1	BİLGİSAYAR MÜHENDİSLİĞİ PR. (İÖ)

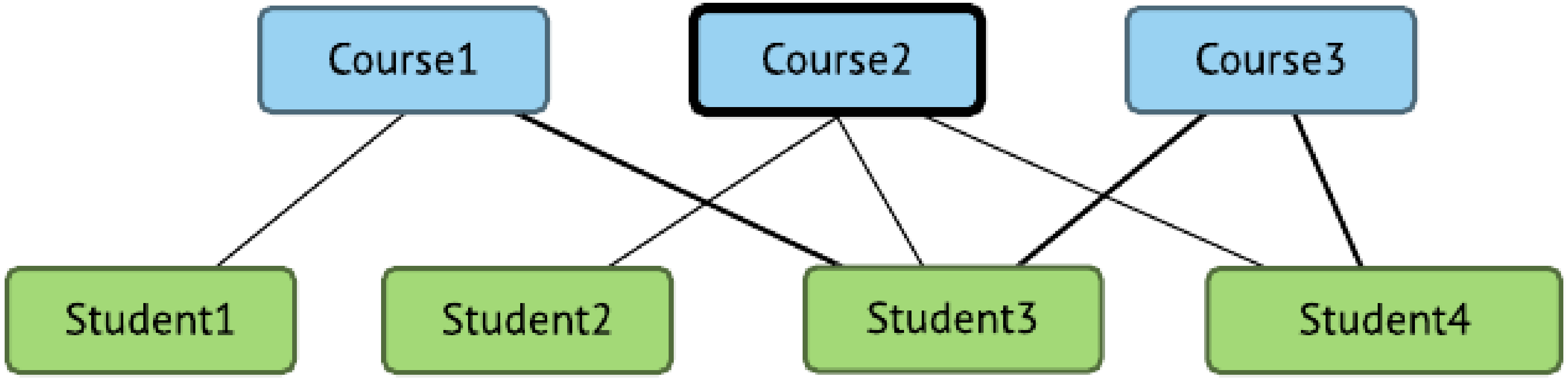
Development of data models

- Hierarchical model
  - It was developed in the 1960s to manage large amounts of data.
  - The data are organized in a tree structure.
  - There is a 1: M relationship between parent-child. Records have only 1 parent record.



Development of data models

- Network model
  - It was developed in the 1970s. More complex relationships are allowed to increase database performance.
  - Unlike the hierarchical model, records can have more than one parent record.



Development of data models

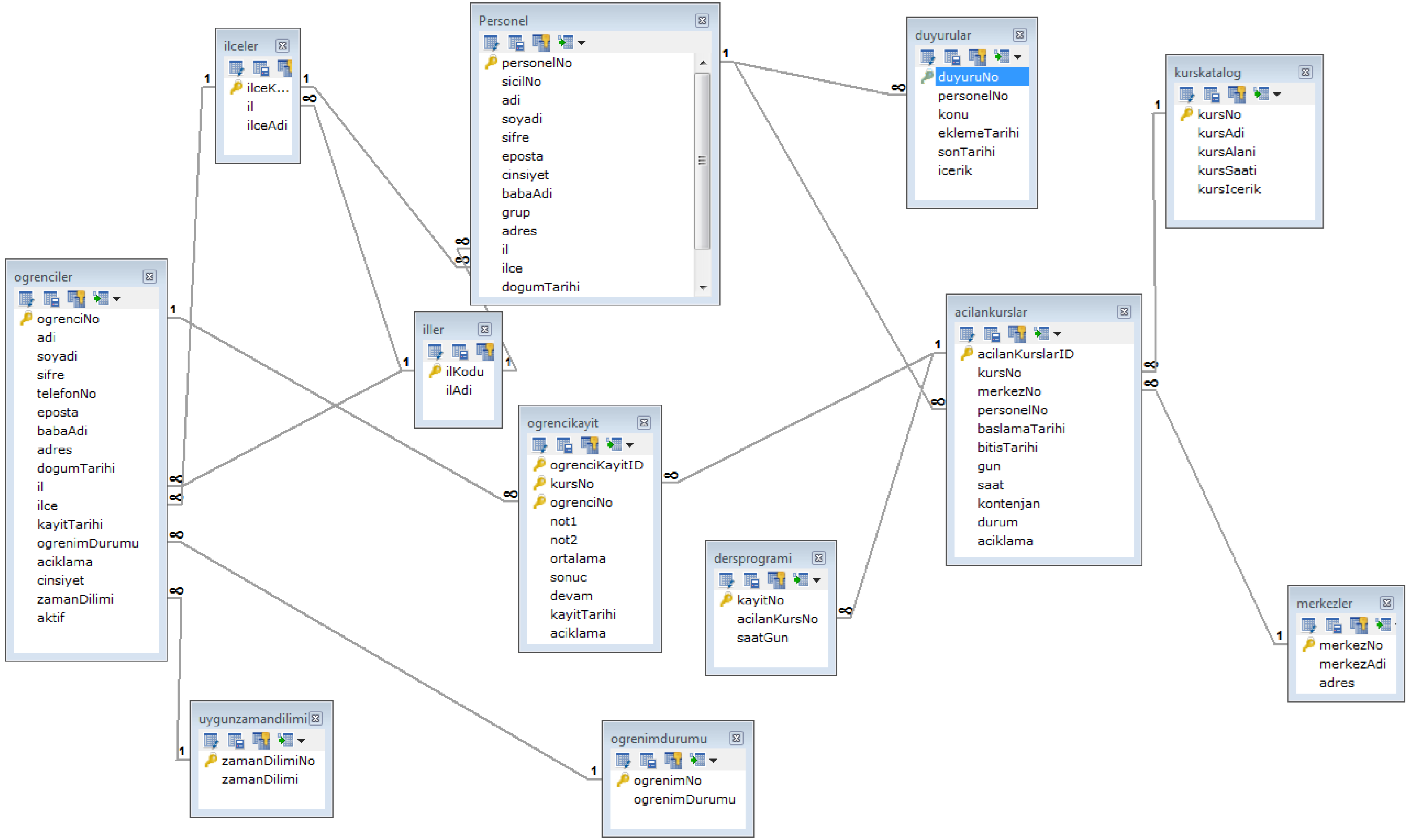
- Network model
  - Below are some of the concepts that emerged and are still used with the network model:
    - Schema: The conceptual organization of the entire database as seen by the database administrator.
    - Sub-schema: The visible part of the database that is produced by the application program.
    - Data manipulation language, DML: The language that enables data updating, new data adding and existing data deleting in the database by performing queries.
    - Data definition language, DDL: The language that enables the definition of the type, structure and constraints of the data in the database.
  - The disadvantage of the network model is that it requires complex program codes, even for very simple queries.

Development of data models

- Relational model
  - It was suggested by E. F. Codd in 1970 (A Relational Model of Data for Large Shared Databanks, Communications of the ACM, June 1970, pp. 377–387).
  - Used by Relational Database Management Systems (RDBMS).
  - One of the most important features of RDBMS is that it hides the complex structure of the relational model from the user.
  - The user sees the relational model as a structure made up of tables containing data.
  - Tables are linked to each other.
  - A relational diagram consists of the representation of entities, attributes of entities, and the relationships between them.

Development of data models

- Relational model



ilceler
ilceK...
il
ilceAdi

Personel
personelNo
sicilNo
adi
soyadi
sifre
eposta
cinsiyet
babaAdi
grup
adres
il
ilce
dogumTarihi

duyurular
duyuruNo
personelNo
konu
eklemeTarihi
sonTarihi
icerik

kurskatalog
kursNo
kursAdi
kursAlani
kursSaati
kursIcerik

ogrenciler
ogrenciNo
adi
soyadi
sifre
telefonNo
eposta
babaAdi
adres
dogumTarihi
il
ilce
kayitTarihi
ogrenimDurumu
aciklama
cinsiyet
zamanDilimi
aktif

iller
ilKodu
ilAdi

ogrencikayit
ogrenciKayitID
kursNo
ogrenciNo
not1
not2
ortalama
sonuc
devam
kayitTarihi
aciklama

acilankurslar
acilanKurslarID
kursNo
merkezNo
personelNo
baslamaTarihi
bitisTarihi
gun
saat
kontenjan
durum
aciklama

dersprogrami
kayitNo
acilanKursNo
saatGun

merkezler
merkezNo
merkezAdi
adres

uygunzamandilimi
zamanDilimiNo
zamanDilimi

ogrenimdurumu
ogrenimNo
ogrenimDurumu

Development of data models

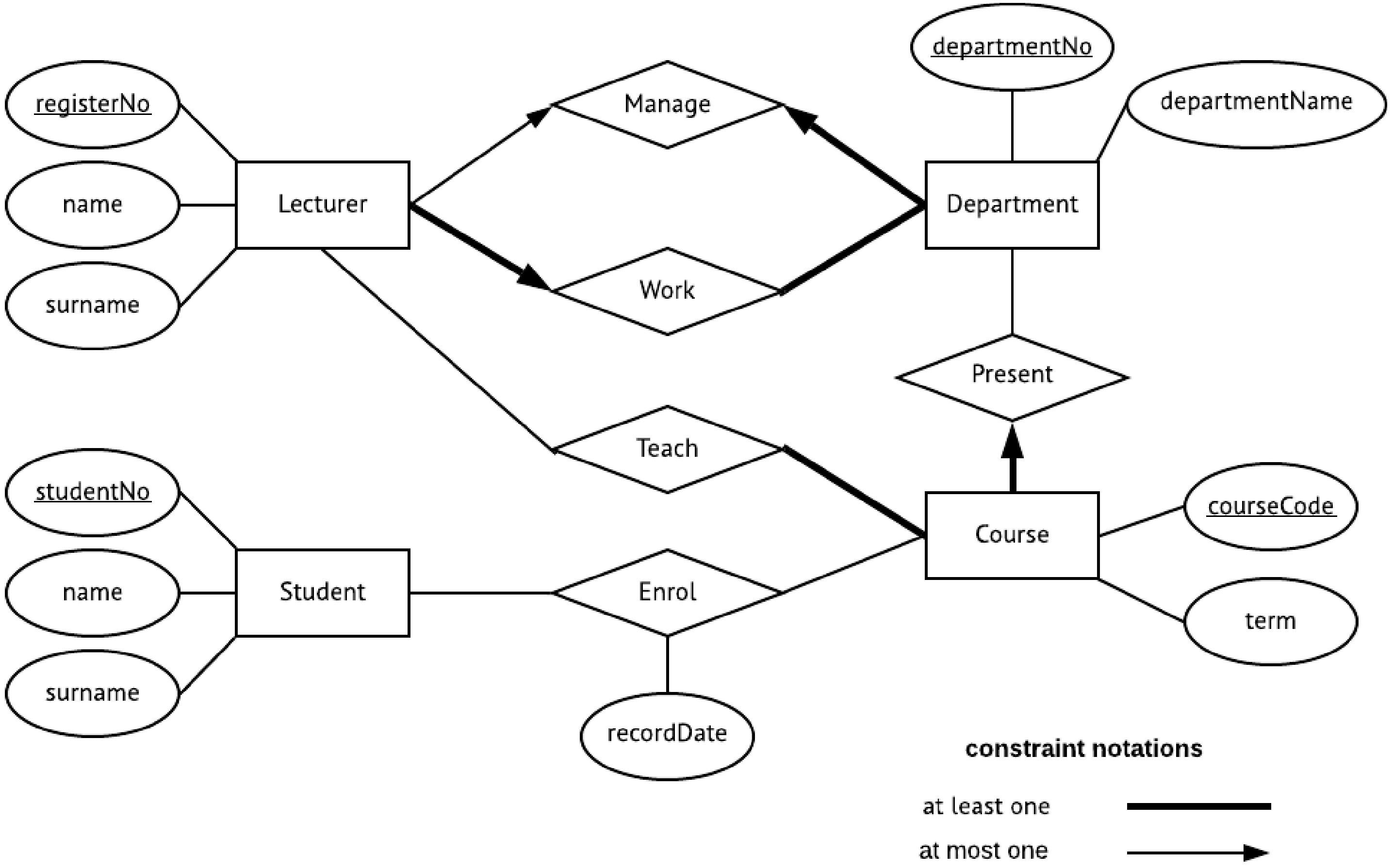
- Relational model
  - One of the strengths of the relational database model is that it uses structured query language (SQL) to manage data.
  - SQL language is a simple language that expresses what should be done instead of explaining how to do it.
  - Therefore, databases are easier to design and manage using SQL.
  - A relational database management system consists of 3 basic components:
    - The database where the data is stored
    - SQL Engine that performs what is desired by compiling SQL commands
    - Interfaces that provide communication with users.

Development of data models

- Entity Relationship model
  - Although the relational model is much more useful than the previous models, the entity relationship model (ERM), which **is the graphical representation of the relational model,** is used more frequently for database design. It was proposed by Peter Chen in 1976.
  - Indepented of software and hardware.
  - Since it is a complement to the relational model, its use has become widespread.
  - The Chen notation and Crow's Foot notation are the frequently used notations.
  - Crow's Foot notation will be used in our examples.

Development of data models

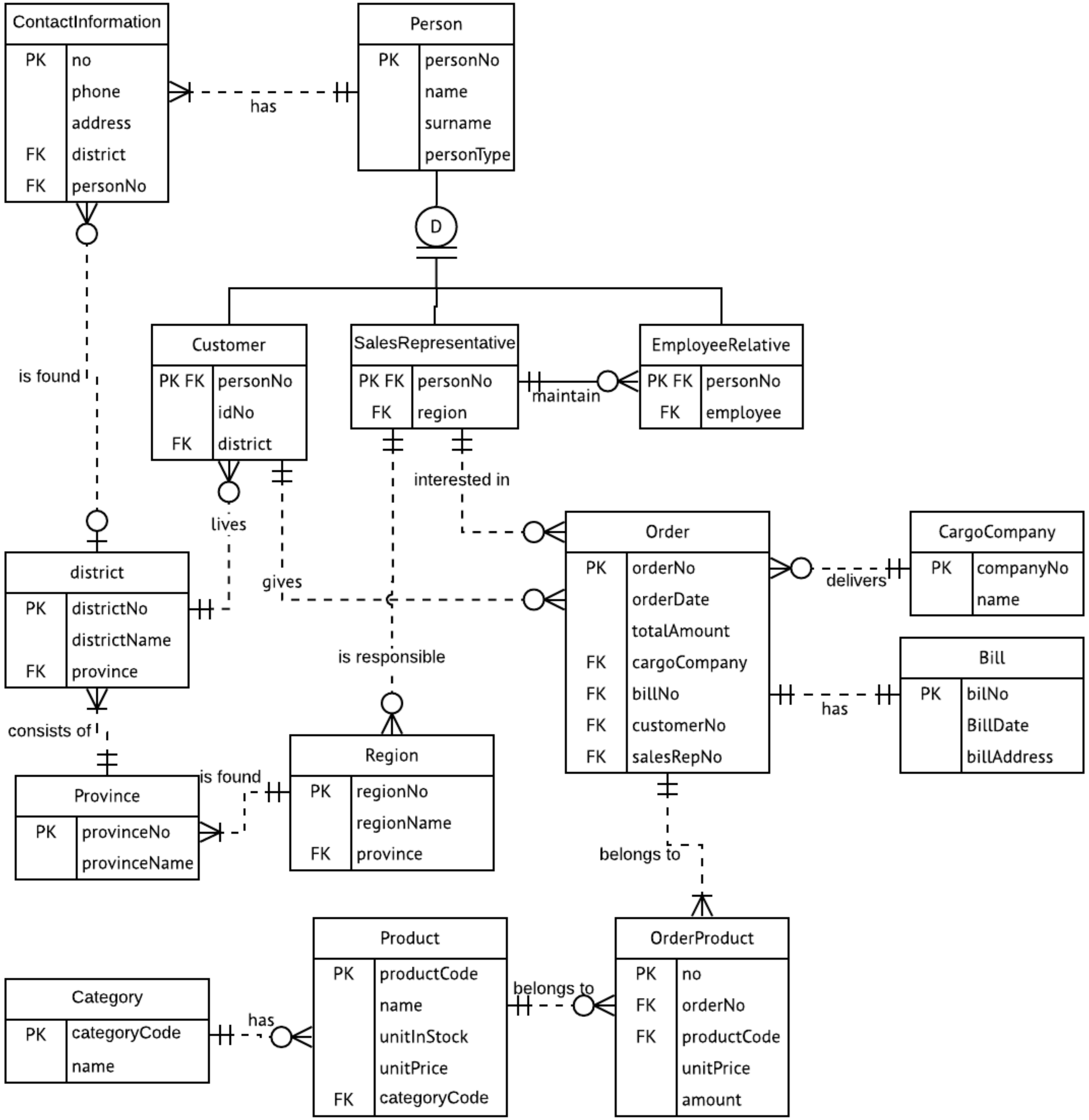
- Entity Relationship model - The Chen notation



Development of data models

- Entity Relationship model –

The Crow's Foot notation



	ContactInformation
PK	no
	phone
	address
FK	district
FK	personNo

	Person
PK	personNo
	name
	surname
	personType

	Customer
PK FK	personNo
	idNo
FK	district

	SalesRepresentative
PK FK	personNo
FK	region

	EmployeeRelative
PK FK	personNo
FK	employee

	district
PK	districtNo
	districtName
FK	province

	Province
PK	provinceNo
	provinceName

	Region
PK	regionNo
	regionName
FK	province

	Order
PK	orderNo
	orderDate
	totalAmount
FK	cargoCompany
FK	billNo
FK	customerNo
FK	salesRepNo

	CargoCompany
PK	companyNo
	name

	Bill
PK	bilNo
	BillDate
	billAddress

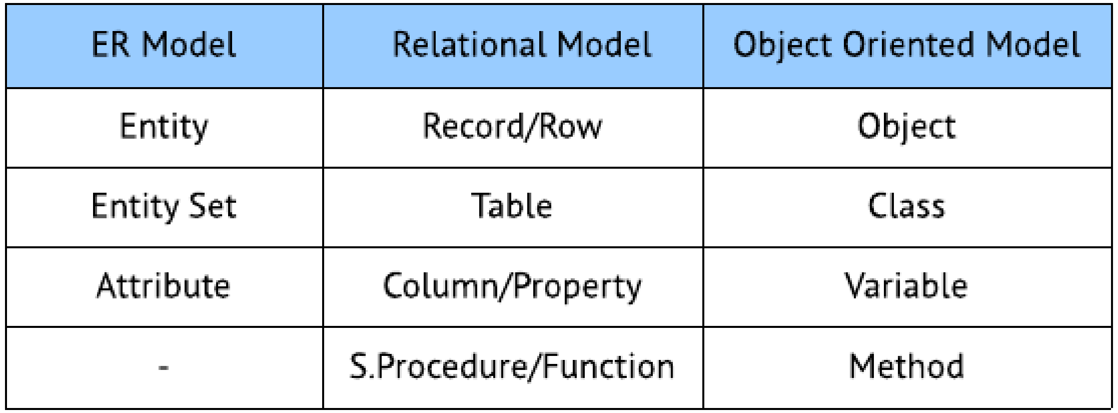
	Category
PK	categoryCode
	name

	Product
PK	productCode
	name
	unitInStock
	unitPrice
FK	categoryCode

	OrderProduct
PK	no
FK	orderNo
FK	productCode
	unitPrice
	amount

Development of data models

- Object oriented model
  - It is a model inspired by the object oriented programming paradigm.
  - The entity in the entity relationship (ER) model is called an object in this model.
  - The information about the object corresponds to the attributes in the ER model.
  - The set of entities is called a class.
  - Unlike the ER model, classes also have member functions. For example: List names



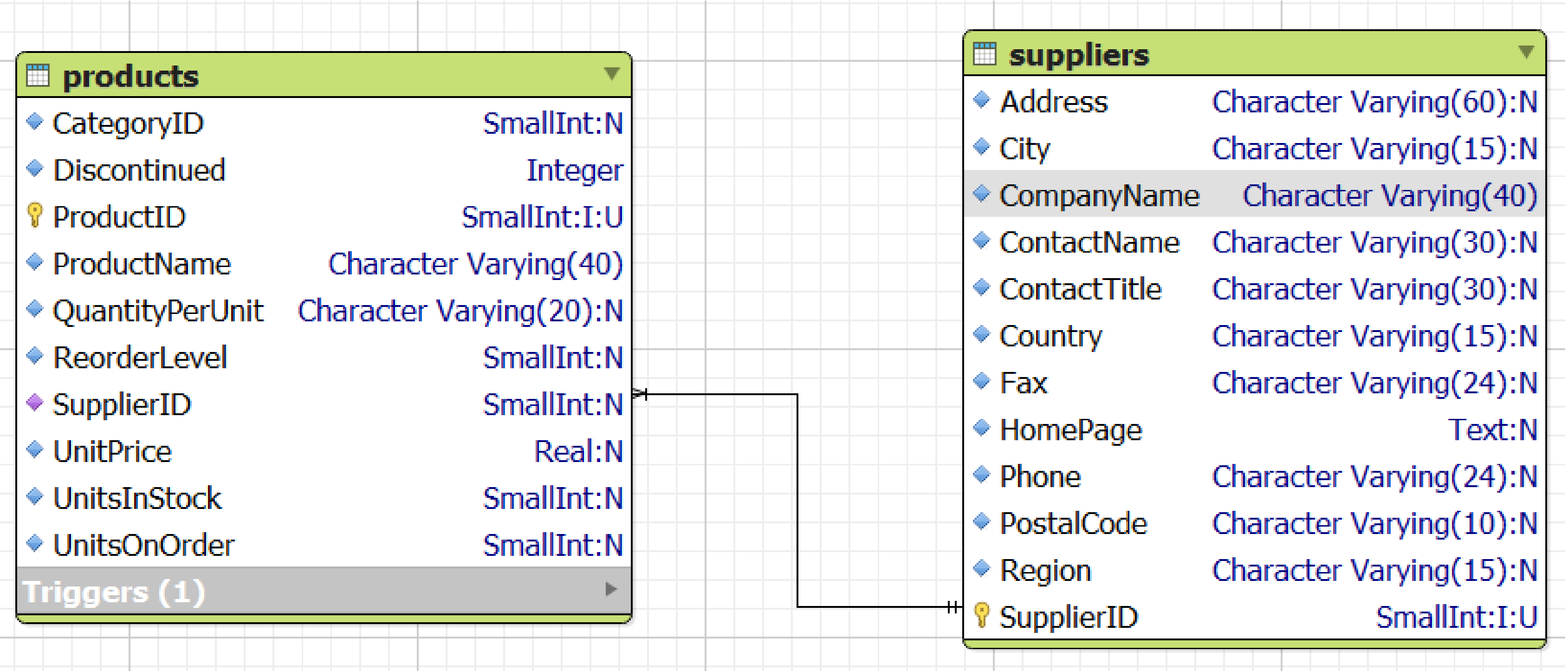
ER Model	Relational Model	Object Oriented Model
Entity	Record/Row	Object
Entity Set	Table	Class
Attribute	Column/Property	Variable
-	S.Procedure/Function	Method

Data Abstraction

- In order to provide a better understanding of data models, ANSI-SPARC defined 3 levels of data abstraction in the early 1970s. (ANSI-SPARC: American National Standards Institute, Standards Planning and Requirements Committee.)
  - External Model
  - Conceptual Model
  - Internal Model

Data Abstraction

- External Model
  - It is the part of the database that is visible to end users. It refers only to the subsections of the database that are relevant to the user.



products	
CategoryID	SmallInt:N
Discontinued	Integer
ProductID	SmallInt:I:U
ProductName	Character Varying(40)
QuantityPerUnit	Character Varying(20):N
ReorderLevel	SmallInt:N
SupplierID	SmallInt:N
UnitPrice	Real:N
UnitsInStock	SmallInt:N
UnitsOnOrder	SmallInt:N
Triggers (1)	

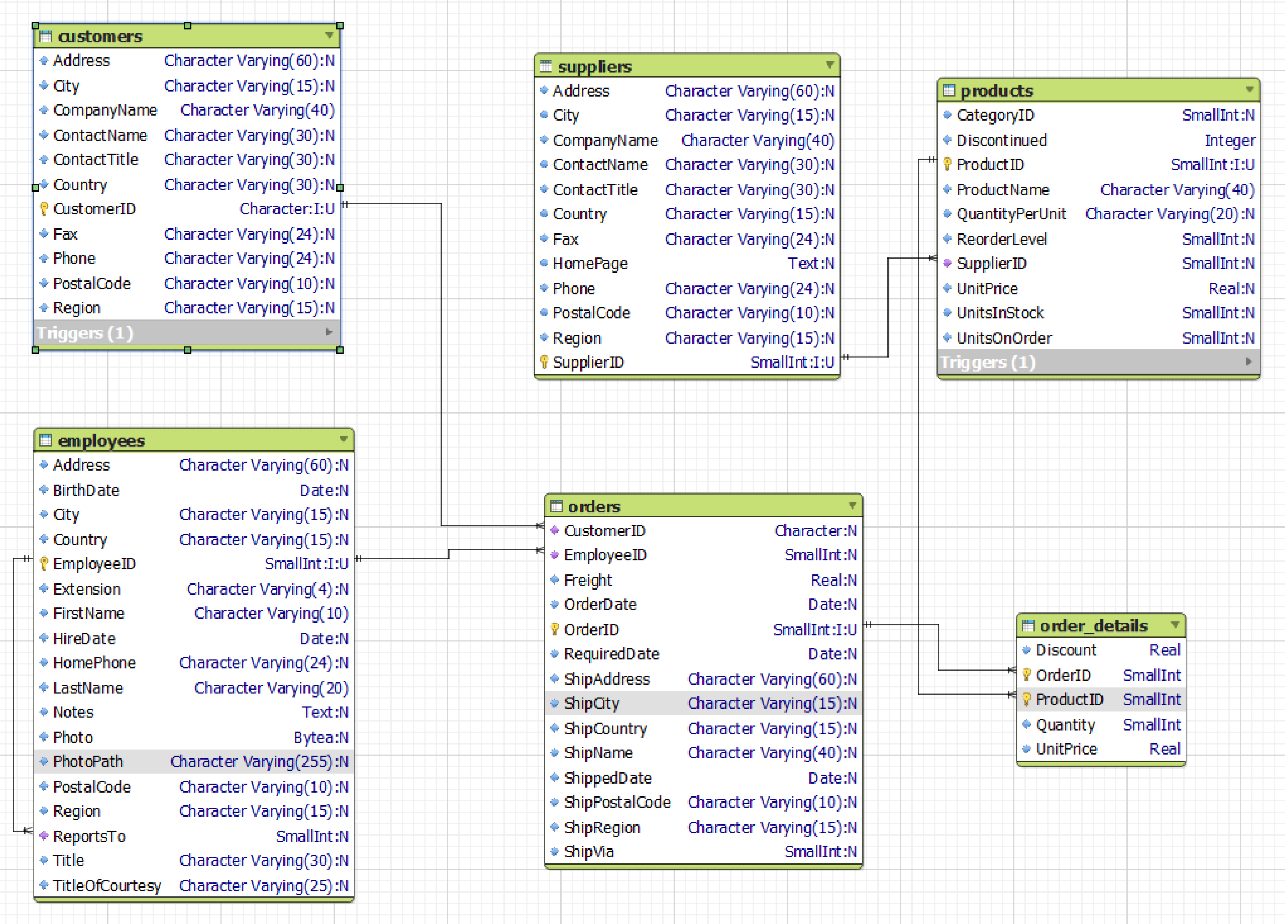
suppliers	
Address	Character Varying(60):N
City	Character Varying(15):N
CompanyName	Character Varying(40)
ContactName	Character Varying(30):N
ContactTitle	Character Varying(30):N
Country	Character Varying(15):N
Fax	Character Varying(24):N
HomePage	Text:N
Phone	Character Varying(24):N
PostalCode	Character Varying(10):N
Region	Character Varying(15):N
SupplierID	SmallInt:I:U

Data Abstraction

- Conceptual Model
  - It is the visible part of the database for the database designer. It combines all the subsections of the database, making it appear globally.
  - It is represented by the Entity Relationship Diagram (ERD). It is independent of software (DBMS) and hardware. Hardware or software changing does not affect the conceptual model design.
  - The conceptual model is also used as a logical view.

Data Abstraction

- Conceptual Model



customers

Address	Character Varying(60):N
City	Character Varying(15):N
CompanyName	Character Varying(40)
ContactName	Character Varying(30):N
ContactTitle	Character Varying(30):N
Country	Character Varying(30):N
CustomerID	Character:I:U
Fax	Character Varying(24):N
Phone	Character Varying(24):N
PostalCode	Character Varying(10):N
Region	Character Varying(15):N

Triggers (1)

suppliers

Address	Character Varying(60):N
City	Character Varying(15):N
CompanyName	Character Varying(40)
ContactName	Character Varying(30):N
ContactTitle	Character Varying(30):N
Country	Character Varying(15):N
Fax	Character Varying(24):N
HomePage	Text:N
Phone	Character Varying(24):N
PostalCode	Character Varying(10):N
Region	Character Varying(15):N
SupplierID	SmallInt:I:U

products

CategoryID	SmallInt:N
Discontinued	Integer
ProductID	SmallInt:I:U
ProductName	Character Varying(40)
QuantityPerUnit	Character Varying(20):N
ReorderLevel	SmallInt:N
SupplierID	SmallInt:N
UnitPrice	Real:N
UnitsInStock	SmallInt:N
UnitsOnOrder	SmallInt:N

Triggers (1)

employees

Address	Character Varying(60):N
BirthDate	Date:N
City	Character Varying(15):N
Country	Character Varying(15):N
EmployeeID	SmallInt:I:U
Extension	Character Varying(4):N
FirstName	Character Varying(10)
HireDate	Date:N
HomePhone	Character Varying(24):N
LastName	Character Varying(20)
Notes	Text:N
Photo	Bytea:N
PhotoPath	Character Varying(255):N
PostalCode	Character Varying(10):N
Region	Character Varying(15):N
ReportsTo	SmallInt:N
Title	Character Varying(30):N
TitleOfCourtesy	Character Varying(25):N

orders

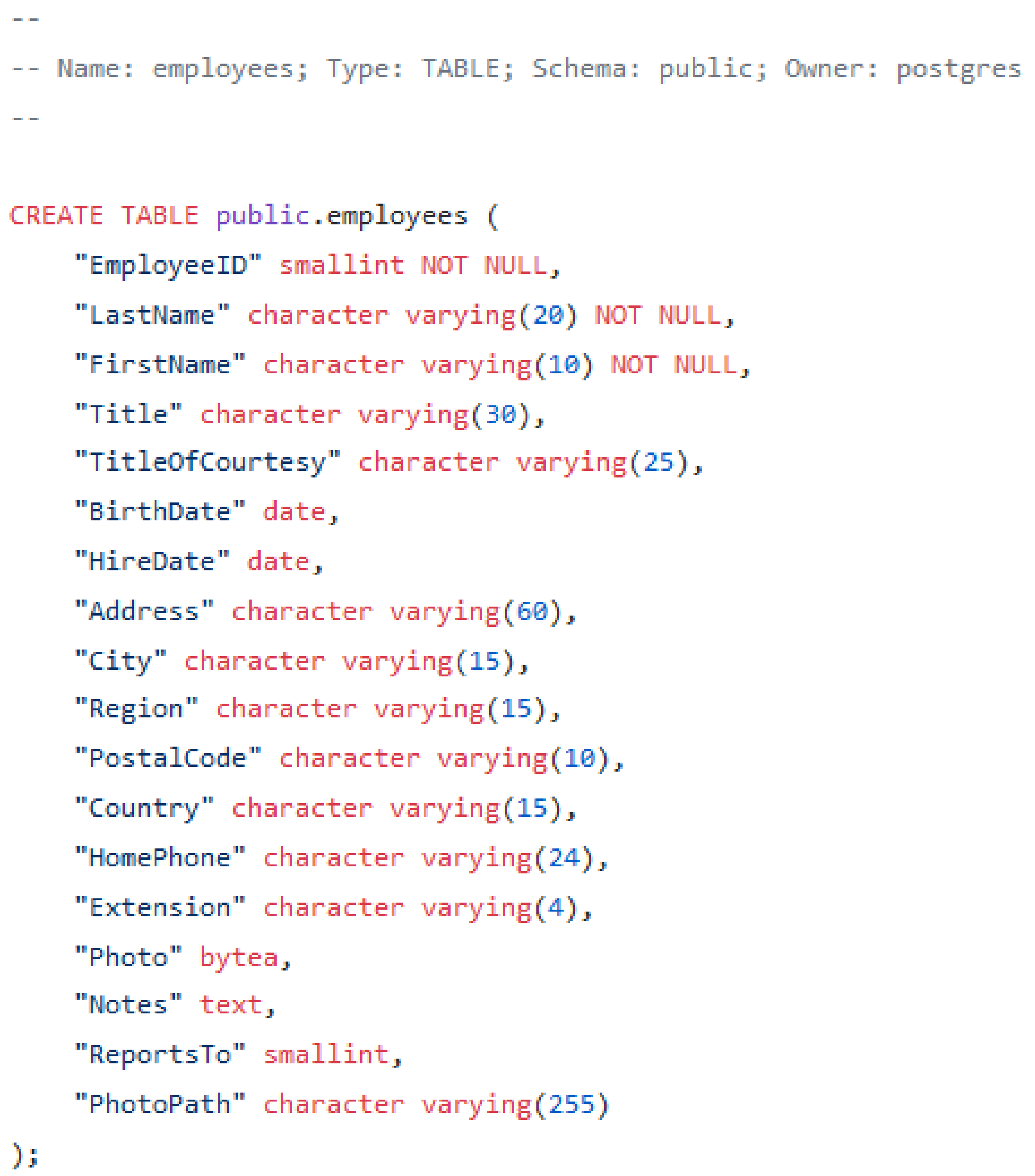
CustomerID	Character:N
EmployeeID	SmallInt:N
Freight	Real:N
OrderDate	Date:N
OrderID	SmallInt:I:U
RequiredDate	Date:N
ShipAddress	Character Varying(60):N
ShipCity	Character Varying(15):N
ShipCountry	Character Varying(15):N
ShipName	Character Varying(40):N
ShippedDate	Date:N
ShipPostalCode	Character Varying(10):N
ShipRegion	Character Varying(15):N
ShipVia	SmallInt:N

order_details

Discount	Real
OrderID	SmallInt
ProductID	SmallInt
Quantity	SmallInt
UnitPrice	Real

Data Abstraction

- Internal Model
  - It is the part of the database that is visible to the Database Management System.
  - Internal model = relational model
  - It is hardware independent, software dependent.



```
--
-- Name: employees; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public.employees (
    "EmployeeID" smallint NOT NULL,
    "LastName" character varying(20) NOT NULL,
    "FirstName" character varying(10) NOT NULL,
    "Title" character varying(30),
    "TitleOfCourtesy" character varying(25),
    "BirthDate" date,
    "HireDate" date,
    "Address" character varying(60),
    "City" character varying(15),
    "Region" character varying(15),
    "PostalCode" character varying(10),
    "Country" character varying(15),
    "HomePhone" character varying(24),
    "Extension" character varying(4),
    "Photo" bytea,
    "Notes" text,
    "ReportsTo" smallint,
    "PhotoPath" character varying(255)
);
```

Entity Relationship Model

- In this section, the entity relationship model is explained in detail:
  - Basic concepts,
  - Relationship between entities,
  - Existence dependency,
  - Weak / Strong Relationship,
  - Relationship degrees,
  - Design with entity relationship model (ERM)

Entity Relationship Model

- Introduction
  - Entity Relationship Diagram (ERD) is a form of representation that enables conceptual modeling of the database.
  - It is independent of software and hardware.
  - It is used to communicate between different users.
  - It is easier to design the database by looking at the ER model.
  - ERD is a form of representation that consists of the basic components of the database, which are entity, attributes, relationships between entities, and constraints.
  - Chen notation emphasizes conceptual modeling.
  - Crow's Foot notation emphasizes more application-oriented approach.
  - UML notation can be used for both conceptual and implementation modeling methods.

Entity Relationship Model

- Converting Business Rules to Entity Relationship Model
  - In general, **nouns** in business rules **are candidates for the entity**, and **verbs are candidates for the relationship between entities.**
  - While nouns or noun phrases with information about them are candidates for the entity, those **without information are candidates for attributes belonging to the entity**.
    - The customer's **name**, **surname**, **number**, and **address** information are stored.
    - Order’s **number, price** information are stored.
    - A *customer* can <u>give</u> many orders.

Entity Relationship Model

- Converting Business Rules to Entity Relationship Model
  - Relationships are bilateral.

    - **1 faculty member** can teach many (4) **lectures.**
    - **1 lecture** can be given by only **1 faculty member.**

    - **1 person** can be the manager of **1 department.**
    - **1 department** can only be managed by **1 person**.

    - **1 student** can enroll in many **courses**.
    - **1 course** can be taken by many **students.**

Entity Relationship Model

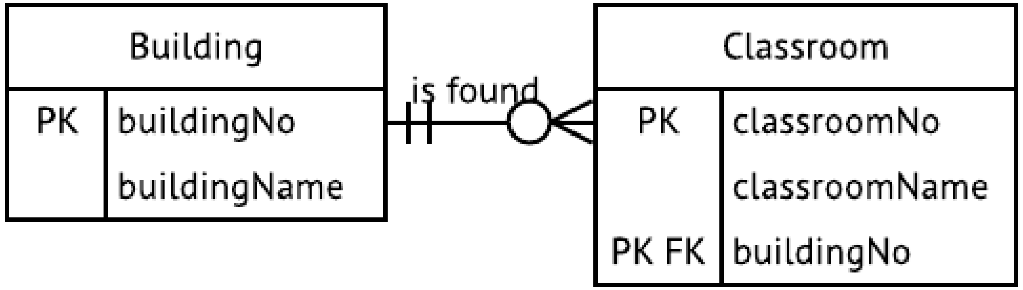
- Converting Business Rules to Entity Relationship Model
  - Primary Key: Attribute(s) that allow us to describe each entity in a set of entities uniquely. **(unique and not null)**
    - It can be a single attribute (field) or a combination of more than one attribute.
      - productCode
      - courseCode + studentNumber
  - Combined attributes:
    - address: street, city, country, zip code, etc.
    - They should be converted to simple featured fields for detailed queries.
  - Simple attributes:
    - age, name, surname, gender, etc.

Entity Relationship Model

- Converting Business Rules to Entity Relationship Model
  - Derived attributes:
    - Value of attributes is **generated using other attributes.**
      - SELECT AVG(AGE(dateOfBirth)) FROM Person
      - Should it be stored or calculated?
      - In Case of Storage:
        - Advantage: Less processing power required, faster access to data, can be used for historical information.
        - Disadvantage: Must be constantly monitored for actual value, takes extra space
      - In Case of Calculation:
        - Advantage: It saves space. It becomes current value every moment.
        - Disadvantage: Requires a lot of processing power, slower access to data, more complex queries.

Entity Relationship Model

- Relationship between entities
  - One to Many relationship
    - **1 classroom** can only be found in **1 building**.
    - **1 building** can include **many classroom**.



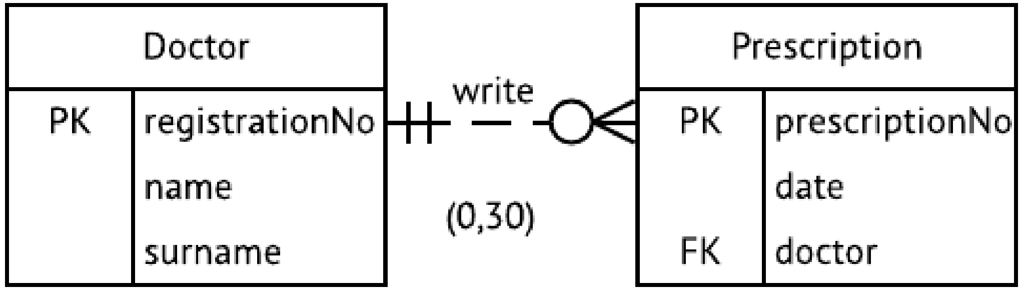
Building

PK	buildingNo
	buildingName

is found

Classroom

PK	classroomNo
	classroomName
PK FK	buildingNo



Doctor

PK	registrationNo
	name
	surname

write

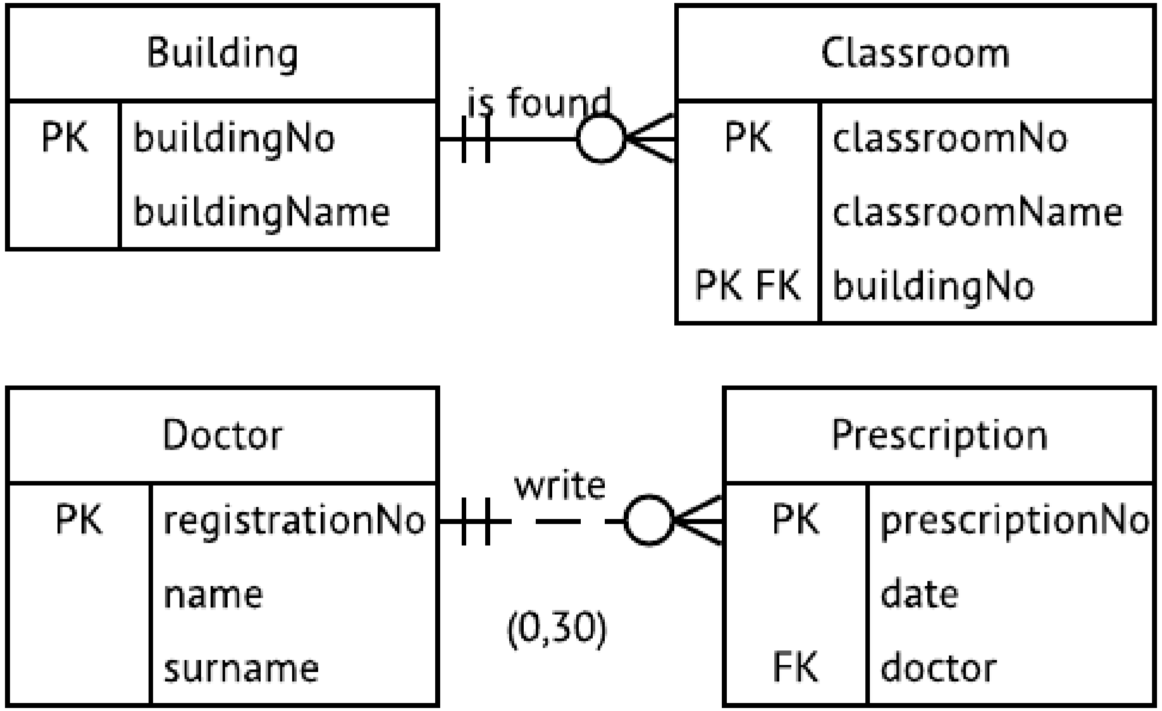
(0,30)

Prescription

PK	prescriptionNo
	date
FK	doctor

Entity Relationship Model

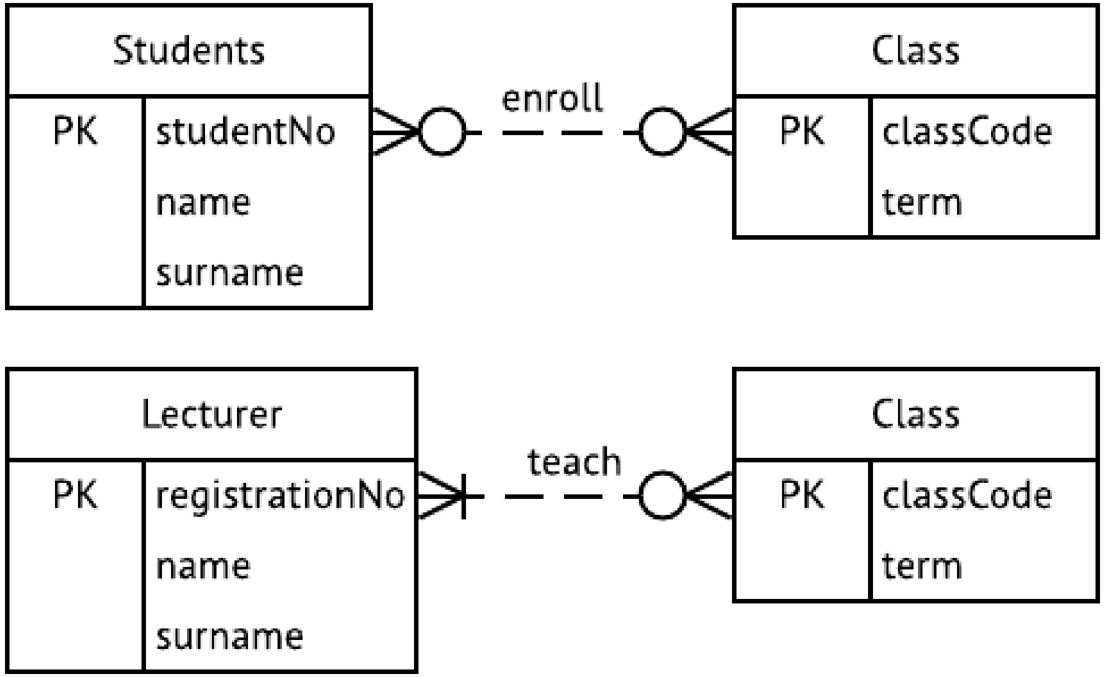
- Relationship between entities
  - One to Many relationship



- Normally, the number of records in the table cannot be limited. Application software or triggers can be used for this purpose.
- Displaying the record counts is very useful when developing application software. (A minimum of 10 and a maximum of 30 registrations are required to open a class. A doctor can write a maximum of 30 prescriptions in a day ...)
- Numbers are determined by looking at business rules.

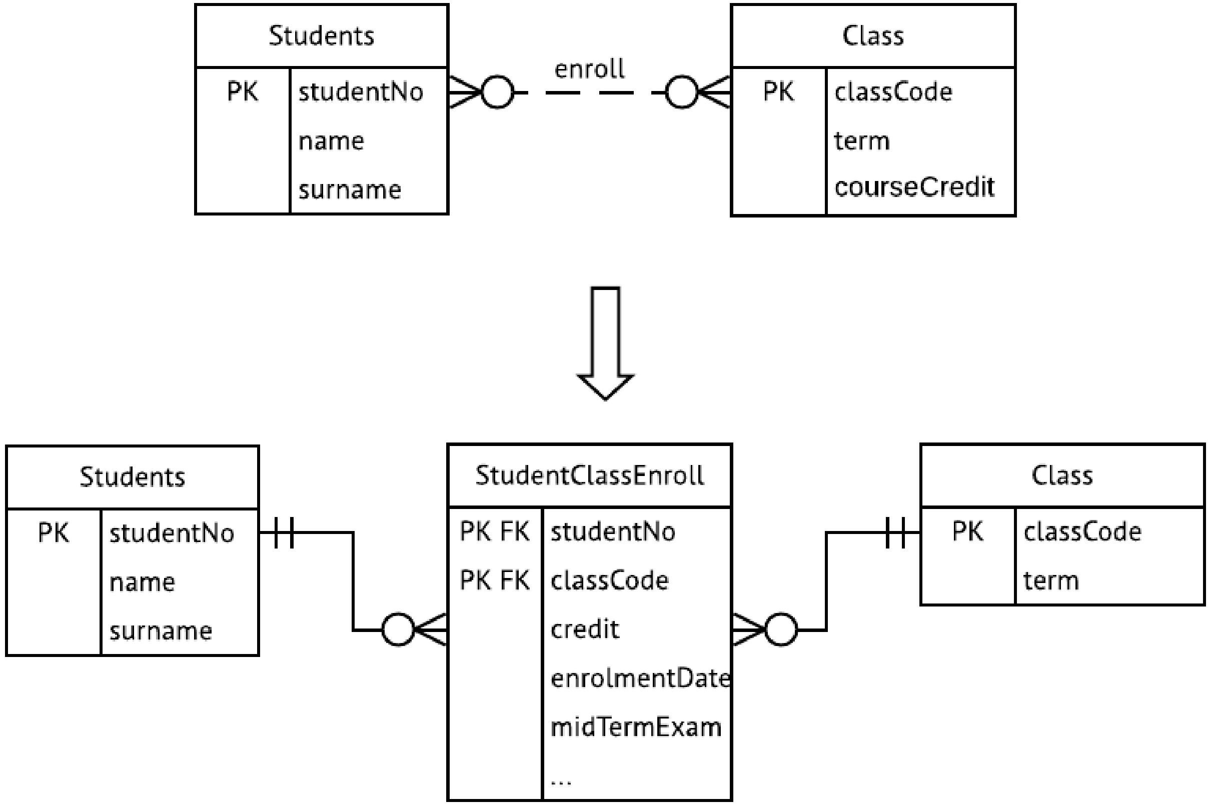
Entity Relationship Model

- Relationship between entities
  - Many to Many relationship
    - **1 student** can enroll in **many classes**.
    - **1 class** can be taken by **many students**.



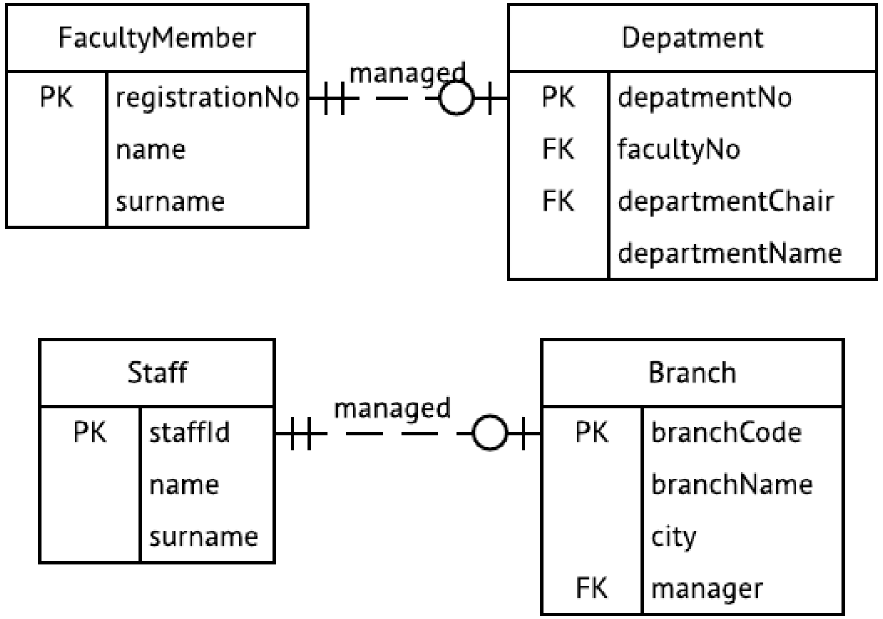
Entity Relationship Model

- Relationship between entities
  - Many to Many relationship
    - **1 student** can enroll in **many classes**.
    - **1 class** can be taken by **many students**



Entity Relationship Model

- Relationship between entities
  - One to One relationship
    - **1 faculty member** can manage **1 department.**
    - **1 department** can be managed by **1 faculty member.**



FacultyMember	
PK	registrationNo
	name
	surname

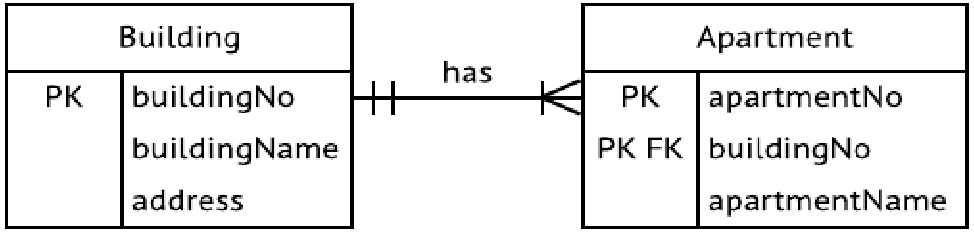
Depatment	
PK	depatmentNo
FK	facultyNo
FK	departmentChair
	departmentName

Staff	
PK	staffId
	name
	surname

Branch	
PK	branchCode
	branchName
	city
FK	manager

Entity Relationship Model

- Existence Dependency
  - In a Building management system, there are Building and Room entities.
  - In such a system, there is the rule that "**a room cannot exist without a building.** "
  - For example, **we cannot** and should not **register a room that does not belong to any building** entity set.
  - In this example, there is said to be **an existing dependency between Room and Building.**
  - In this case, **the Building the superior entity and the Room is the dependent entity.**

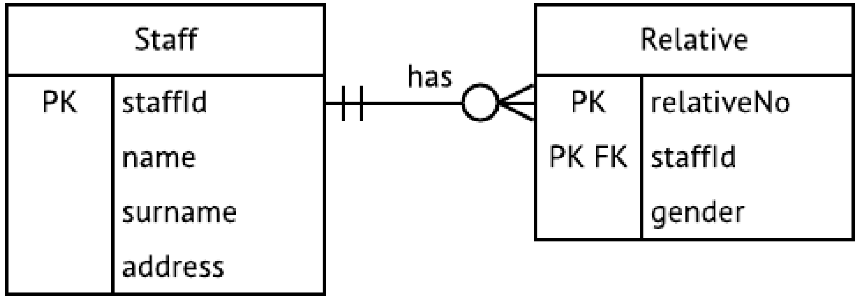


Building	
PK	buildingNo
	buildingName
	address

has

Apartment	
PK	apartmentNo
PK FK	buildingNo
	apartmentName

Entity Relationship Model



Staff

PK staffId

name

surname

address

has

Relative

PK relativeNo

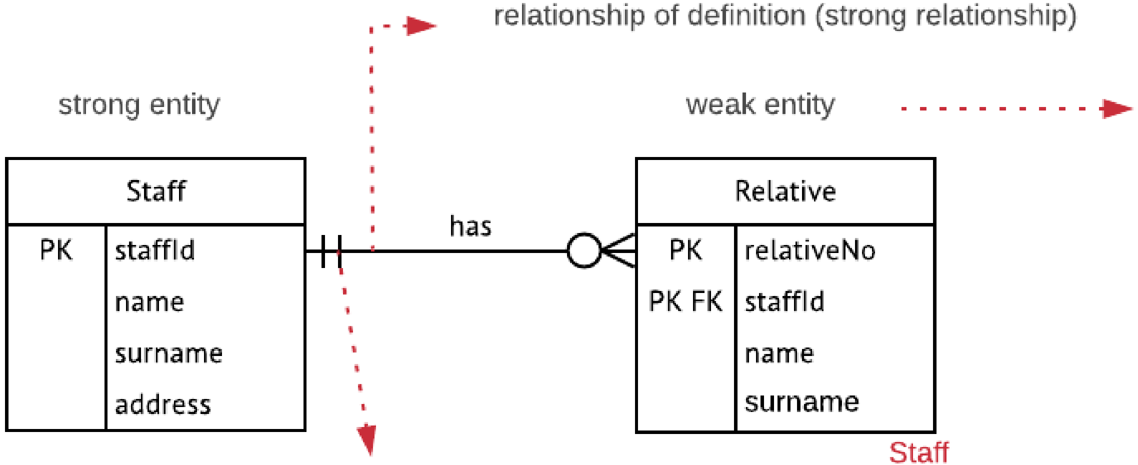
PK FK staffId

gender

- Existence Dependency
  - Consider a Personnel and Relative entity sets (For example: child, wife, etc.) in a **personnel information system.**
  - In such a system, there is a rule that **"there can not be relative entity that is not related to a staff member.** "
  - For example, **we cannot or should not register a child who is not related to any staff member** to the Relative entity set.
  - In this example, there is said to be an **existing dependency between the Relative and the Staff.**
  - In this case, **the staff is the superior entity, and the Relative is the dependent entity.**

Entity Relationship Model

- Existence Dependency



relationship of definition (strong relationship)

strong entity

weak entity

Staff	
PK	staffId
	name
	surname
	address

Relative	
PK	relativeNo
PK FK	staffId
	name
	surname

1. The strong entity's primary key is needed for defining.
2. In order to exist, there must be a relevant staff.

Existence Dependency
*staffId (FK) in weak entity must be "not null"
*Since FK is also PK, there is always an existence dependency for the weak entity.

Staff

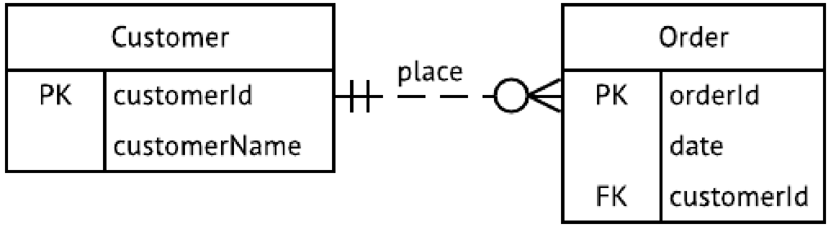
staffId	name	surname	address
1	...	...	...
2	...	...	...

Relative

relativeNo	staffId	name ...
1	1	...
2	1	...
1	2	

Entity Relationship Model

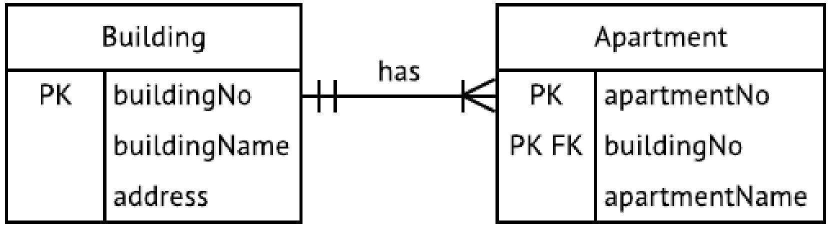
- Weak / Strong Relationship
  - While a relationship is established between the two entities, the primary key of one of the entities can take place as the primary key on the other.
  - The relationship, in this case, is called the strong relationship.
  - Otherwise, the relationship is a weak relationship.



Customer
PK customerId
customerName

place

Order
PK orderId
date
FK customerId



Building
PK buildingNo
buildingName
address

has

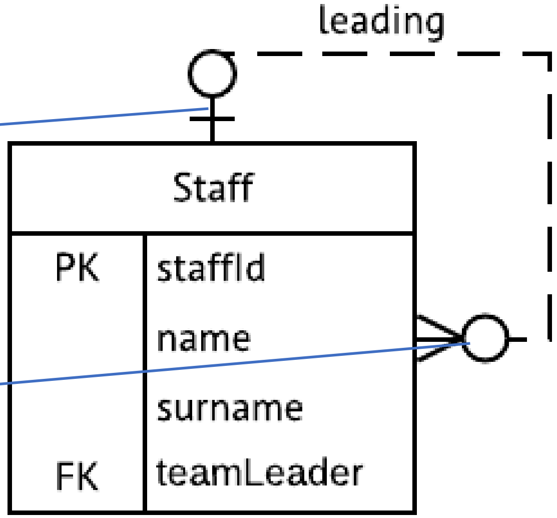
Apartment
PK apartmentNo
PK FK buildingNo
apartmentName

Entity Relationship Model

- Relationship degrees
  - Unary relationship
  - Binary relationship
  - Ternary relationship

Entity Relationship Model

- Relationship degrees
  - Unary relationship
    - If an entity has a relationship to itself, such a relation is called a unary relationship.
    - For example, a staff member in a Staff table is the manager of zero or more staff at the same time. A staff member must have either zero or one manager.
    - class-subclass, category-subcategory

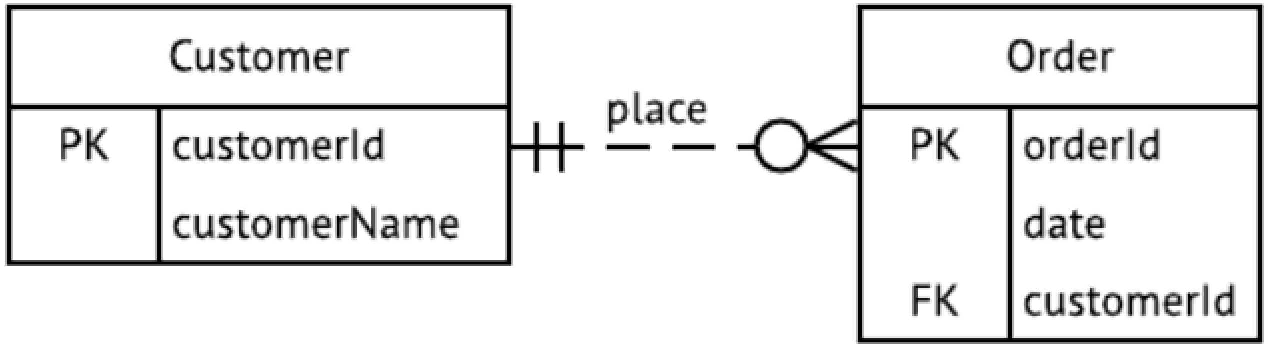


One staff can have one team
leader or no teamleader

One team leader (staff) can
leads no staff or many stafs

Entity Relationship Model

- Relationship degrees
  - Binary relationship
    - The relationship of two entities is called the binary relationship.
    - Most common

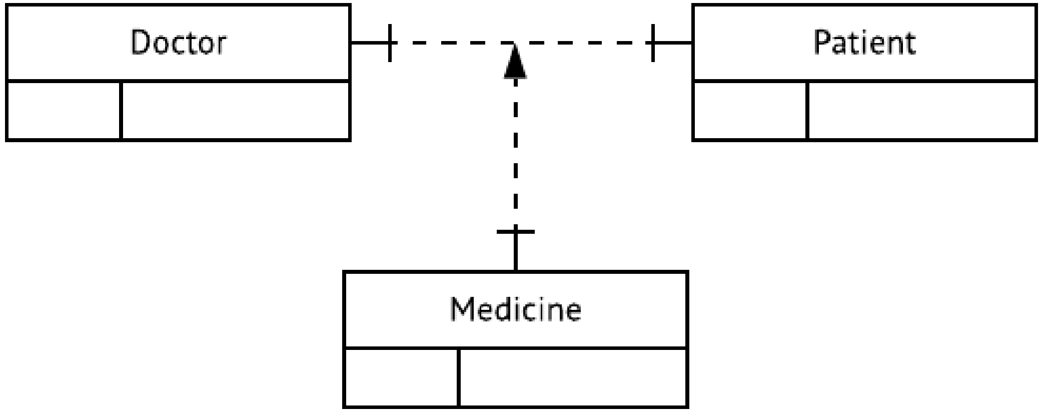


Customer	
PK	customerId
	customerName

Order	
PK	orderId
	date
FK	customerId

Entity Relationship Model

- Relationship degrees
  - Ternary relationship
    - If 3 entities have relationship at the same time, this type of relationship is called a ternary relationship.
    - Although there are 3 entities in a conceptual design, a 4th entity is required to perform this relationship.



Entity Relationship Model

- Design with entity relationship model (ERM)
  - Database design is an iterative (spiral model or iterative model in software development) process rather than linear or sequential (like the waterfall model in software development).
  - Iterative means "do it over and over again". An iterative process is based on repeating processes and procedures. (attributes can turn entities, relationships can change)
  - Developing an entity relationship diagram usually involves the following steps:

Entity Relationship Model

- Design with entity relationship model (ERM)
  - A detailed scenario containing the description of the work of the institution is created. The scenario will be much more realistic and effective, especially if it is developed in consultation with the role representatives within the organization.
  - Business rules are created based on the descriptions of the works in the scenario.
  - Entities and relationships between the entities are created based on business rules.
  - The first entity relationship diagram is developed.
  - The attributes and primary keys that clearly defined.
  - The Entity Relationship Diagram is reviewed and if necessary, the above steps are repeated until the desired state is reached.