Databases 2022

Darko Bozhinoski,
Ph.D. in Computer Science

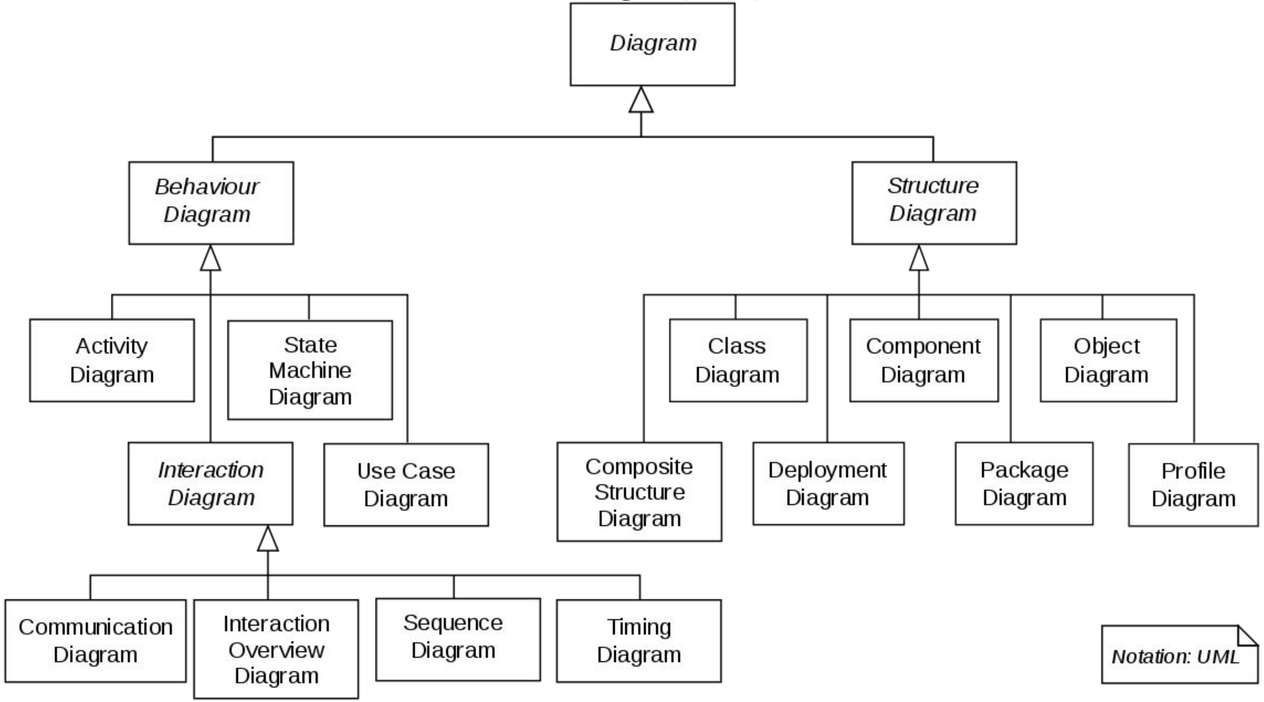
Agenda

- UML
- Data, Information, Knowledge Representation
- Relational Model

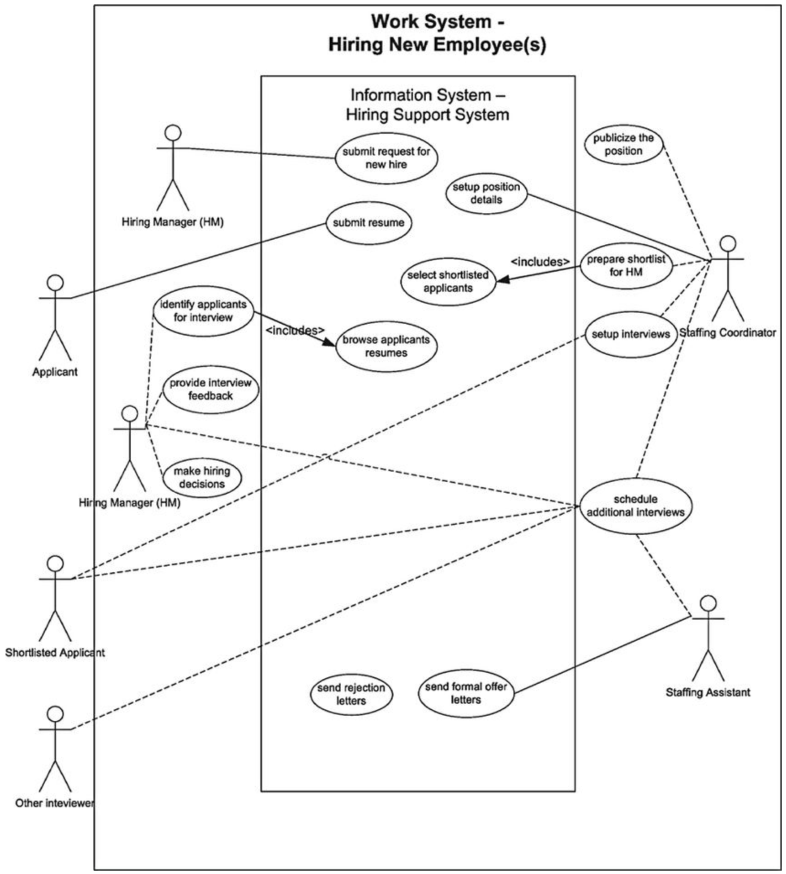
UML - Unified Modeling Language

UML - Unified Modeling Language

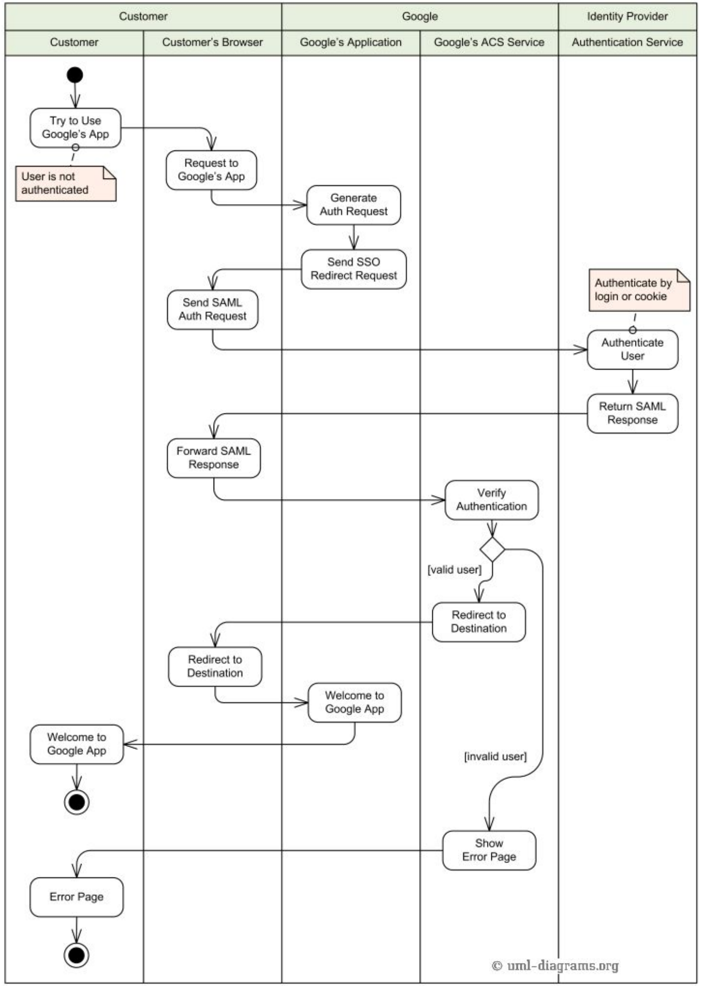
- UML is being used extensively in software design and has many types of diagrams for various software design purposes.



Use Case Diagram



Activity Diagram



Class Diagrams

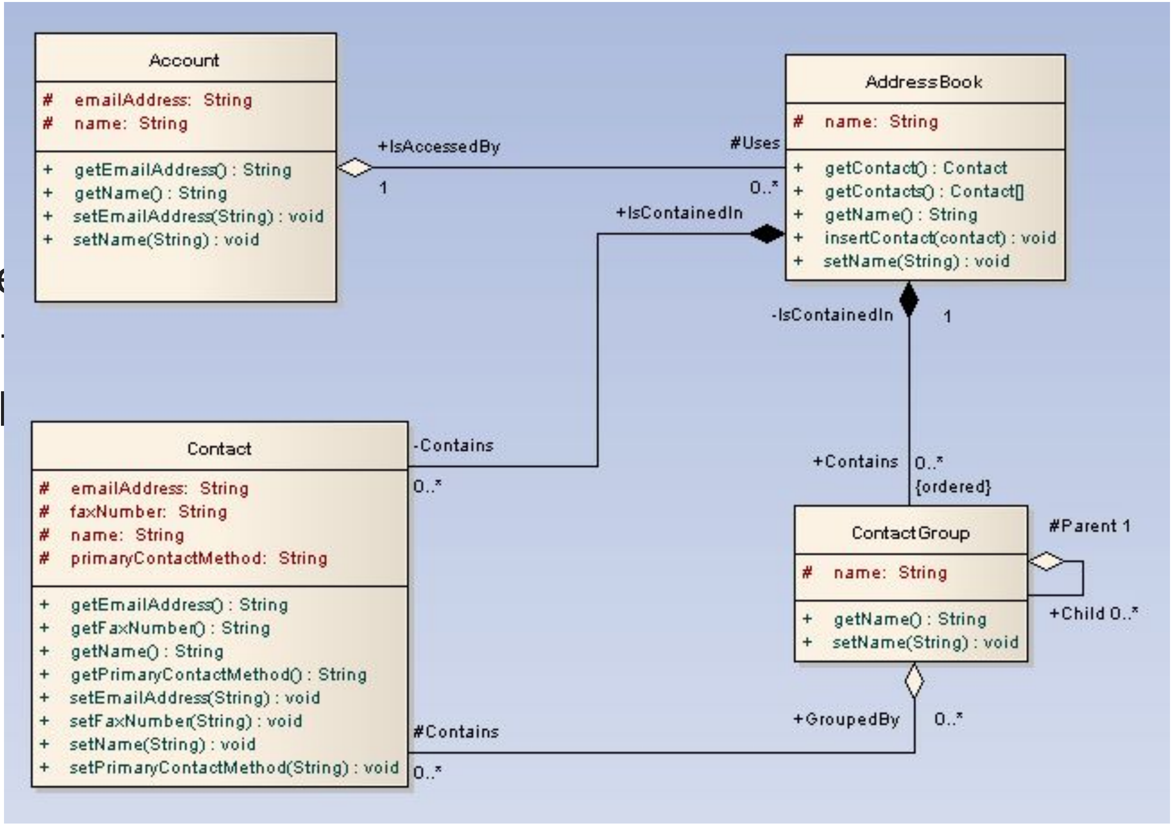
- **Unified Modeling Language (UML) notation** for class diagrams has been used as a standard for conceptual object modeling.

- Class diagrams are used to model the structure of an application, and for detailed modeling and translating models into code. Class diagrams can also be used for data modeling.

Class Diagrams

- **Unified Modeling Lan**
  used as a standard for

- Class diagrams are use
  detailed modeling and
  be used for data model



UML vs. ER Diagram

UML Motivations:

- UML helps to develop efficient, effective and correct Object-Oriented designs
- UML is used for clear communication between project stakeholders

Entity-Relationship diagrams are not UML diagrams

UML: Object Oriented Design

ER: conceptual data modeling

UML class diagrams are similar to ER, but not the same!

ER and UML

When should UML be used? When should ER be used?

Tool support:

- ER models can be mapped to relational schema
- From UML a code skeleton can be generated in several languages

ER and UML

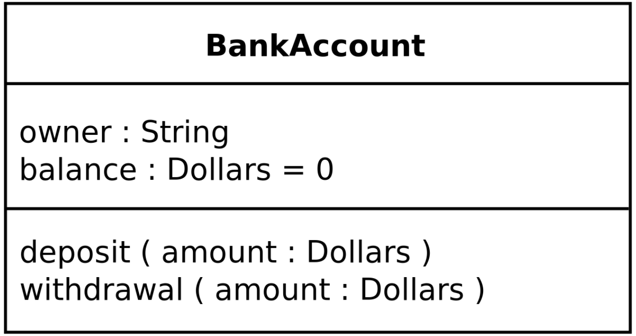
When should UML be used? When should ER be used?

Different modeling languages (Entity-Relation, Unified Modeling Language, and others) are simply notations for communicating a design to stakeholders. Communicating a design is technical communication, and one of the principles of good technical communication is to communicate the information clearly and concisely. Choosing a modeling notation that is understood by your audience and can communicate the desired information clearly is the first step to achieve this principle.

Mapping between UML and ER: Basic concepts

In UML class diagrams, a **class is equivalent to an entity type** in ER.

- The top section gives the **class name** (equivalent to entity type name in ER);
- The middle section includes the **attributes**;
- The last section includes **operations** that can be applied to individual objects (similar to individual entities in an entity set) of the class. Operations are *not* specified in ER diagrams.



Mapping between UML and ER: Basic concepts (2)

- Relationship types (ER) are equivalent to **associations** in UML
- Relationship instances are equivalent to **links**.
- A **binary association** (binary relationship type) is represented as a line connecting the participating classes (entity types), and may optionally have a name. A relationship attribute, called a **link attribute**, is placed in a box that is connected to the association's line by a dashed line.

Mapping between UML and ER: Basic concepts (3)

- The **(min, max) notation to specify relationship constraints** are equivalent to **multiplicities** in UML terminology.
- Multiplicities are specified in the form min..max, and an asterisk (*) indicates no maximum limit on participation
- **Recursive relationship type** is equivalent to **reflexive association** in UML
- **Aggregation:** a relationship between a whole object and its component parts (different notation is used)
- **Association (relationship) names are optional** in UML, and relationship attributes are displayed in a box attached with a dashed line

Mapping between UML and ER: Basic concepts (4)

- Weak entities (in ER) are modeled using the construct qualified association (UML) (e.g. Dependent_name).

- In UML terminology, the partial key attribute is called **discriminator**, because its value distinguishes the objects associated with (related to) the same entity

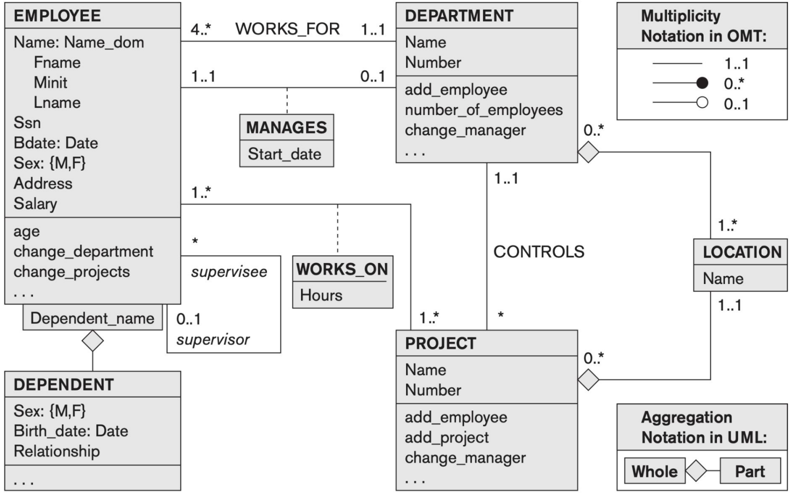


Figure 3.16
The COMPANY conceptual schema in UML class diagram notation.

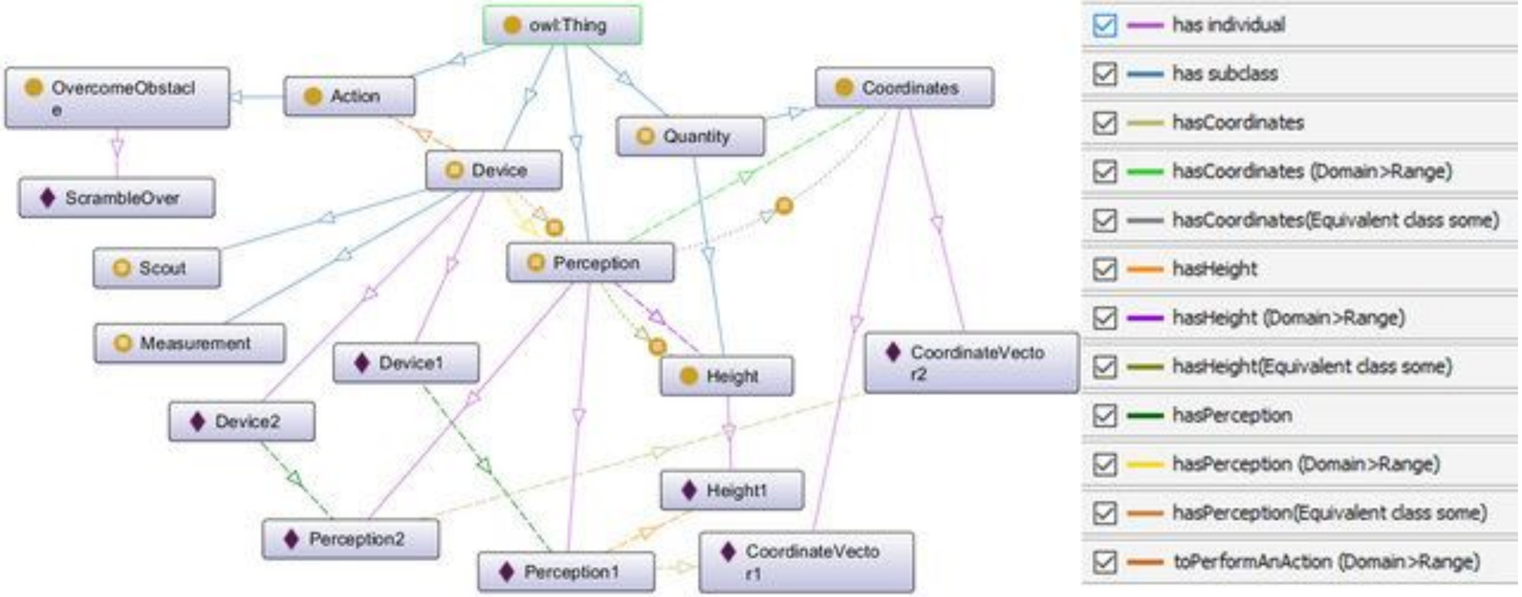
Knowledge Representation and Ontology

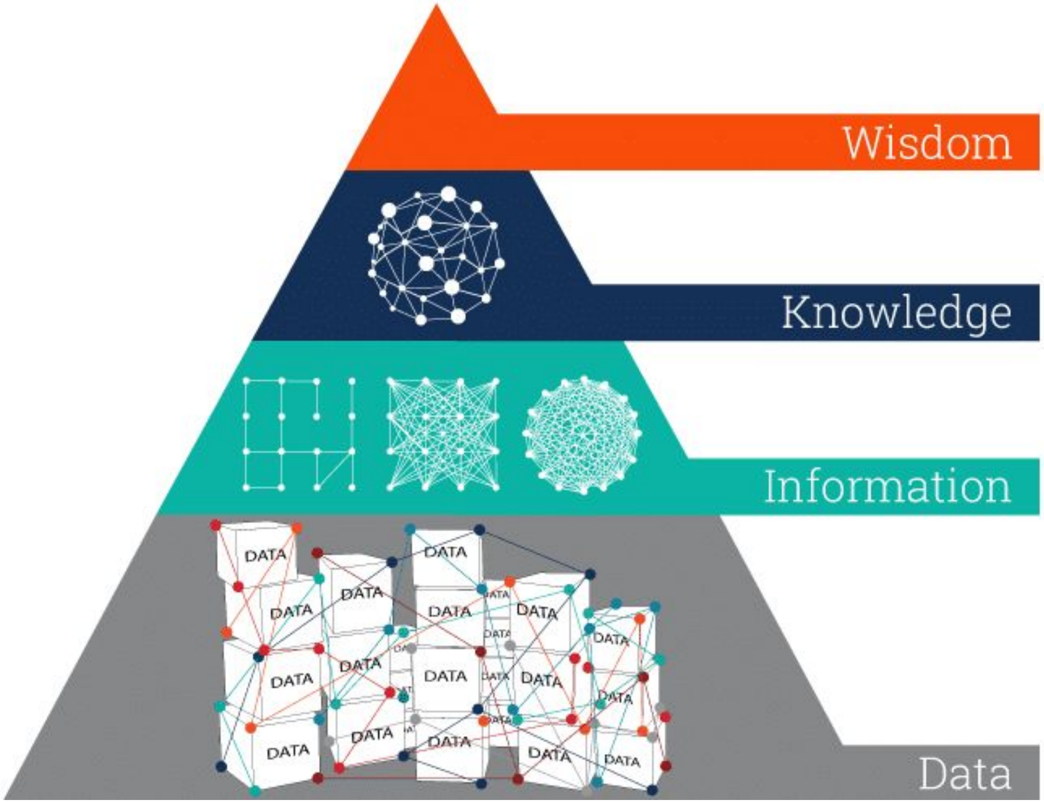
- ❖ Conceptual data modeling
- ❖ Artificial Intelligence (AI)
- ❖ KR techniques: to develop concepts for accurately modeling some domain of knowledge. **Output:** An ontology
- ❖ Ontology: describes the concepts of the domain and how these concepts are interrelated through some common vocabulary.
- ❖ The main difference between an ontology a database schema, is that the schema is usually limited to describing a small subset of the world to store and manage data. An ontology is considered to be general in that it attempts to describe a part of reality or a domain of interest.

Comparison between conceptual modeling and knowledge representation

- KR is broader in scope.
- Different forms of knowledge, such as rules (used in inference, deduction, and search), incomplete and default knowledge, and temporal and spatial knowledge, are represented in KR schemes.
- KR schemes include reasoning mechanisms that deduce additional facts from the facts stored in a database. Most current databases are limited to answering direct queries. Knowledge-based systems using KR schemes can answer queries that involve inferences over the stored data.

Example of robot ontology





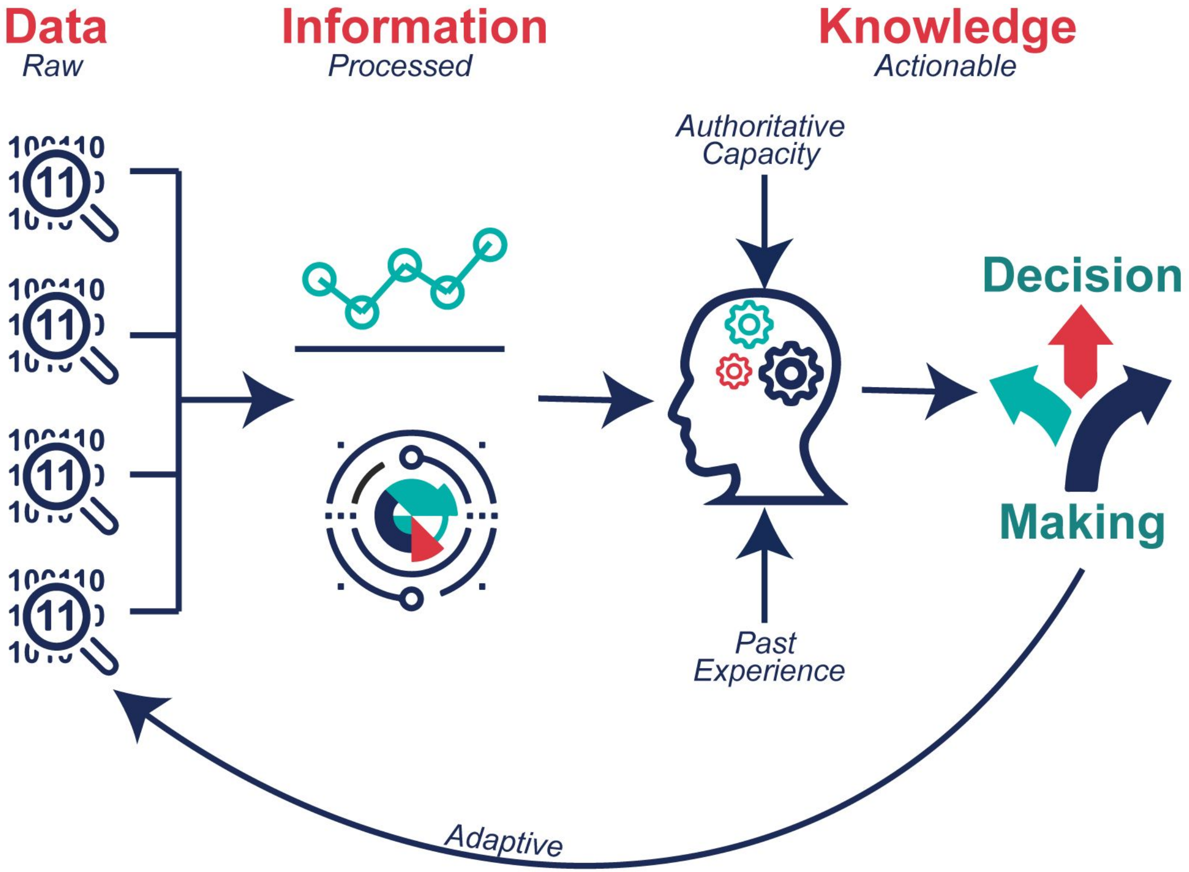
WHY? WHAT IS BEST?

HOW?

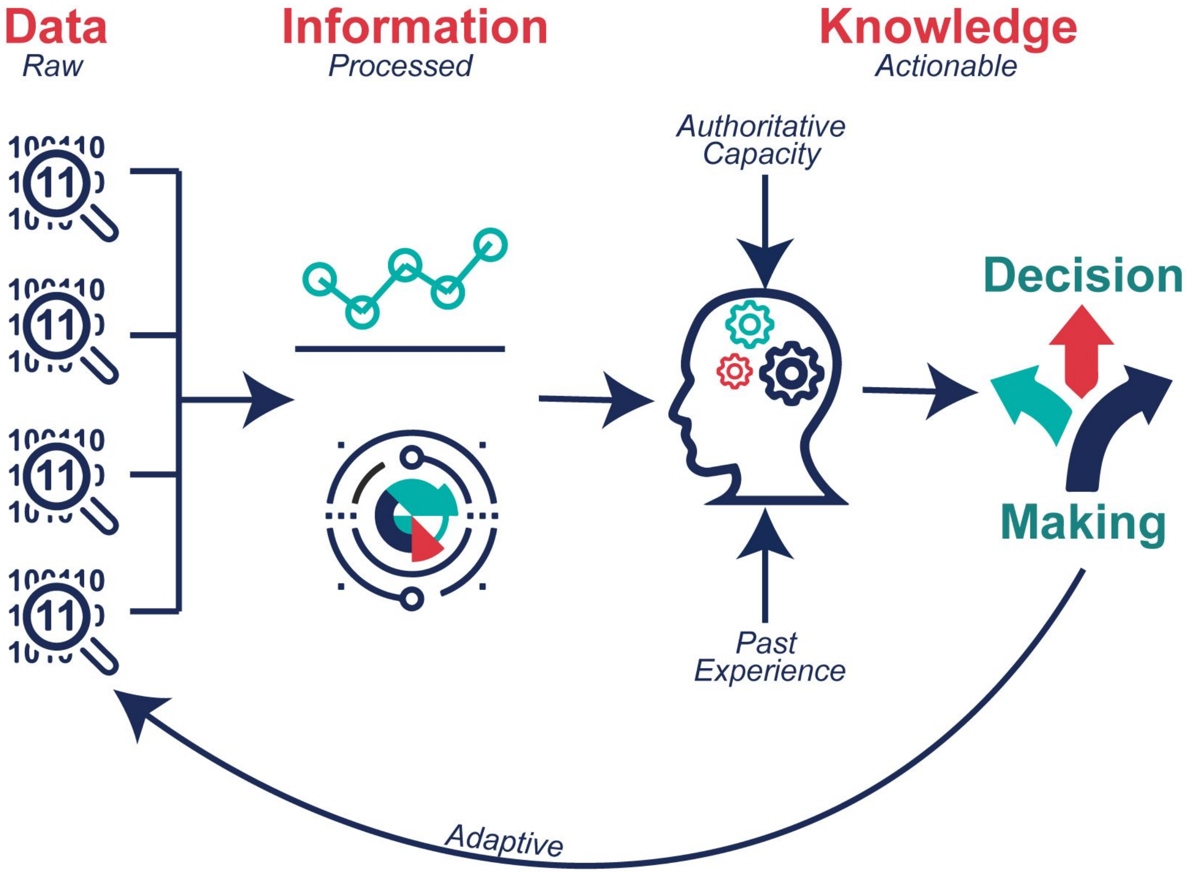
Each step up the pyramid answers questions about and adds value to the initial data.

Data, Information, Knowledge, Wisdom (DIKW) Pyramid

From Data to Decision



From Data to Decision



Data: Facts, description of the World

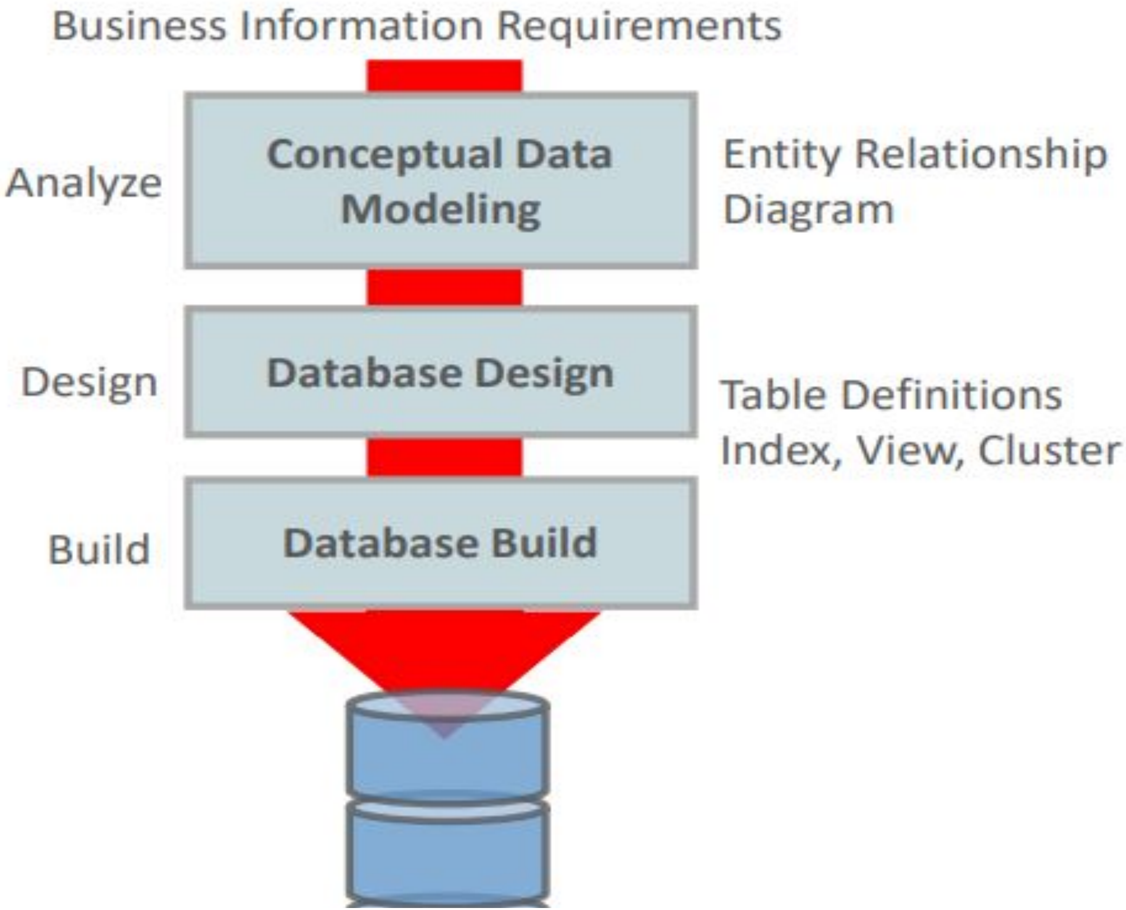
Information: Organized/structured Data

Knowledge: Our personal map/model of the World

Decision: Informed action

Database Design

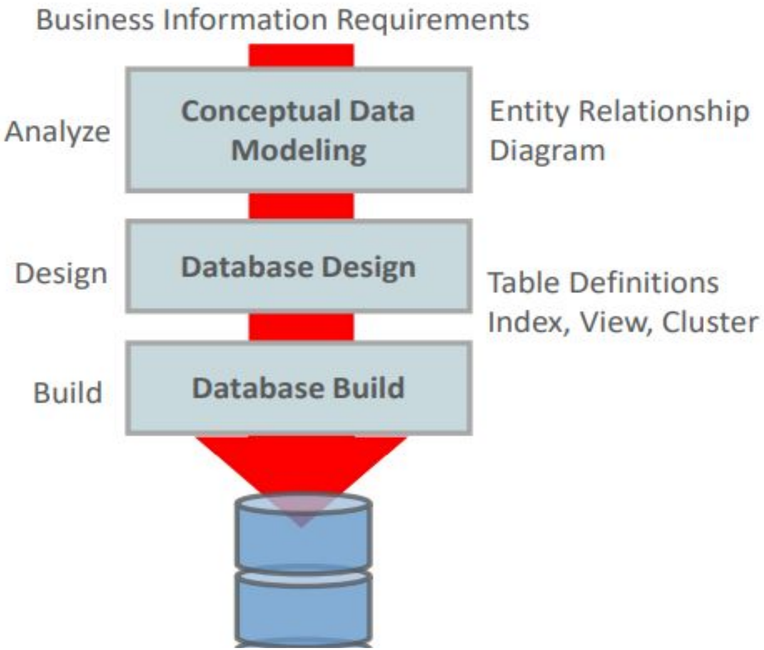
Database Development Process



Conceptual Database Design

- Requirements analysis leads to a high-level description of the data

  Produced artifact: ER model



- Converting the conceptual database design into a database schema in the data model of the chosen DBMS

  Produced artifact: Database schema;

Introduction to Relational Model

Background

- ❖ Proposed by E.F. Codd in 1970 in his seminal paper “A relational model of data for large shared data banks”
- ❖ In the relational model of a database, all data is represented in terms of tuples, grouped into relations.
- ❖ A database organized in terms of the relational model is a relational database.

- ❖ The purpose of the relational model is to provide a declarative method for specifying data and queries: users directly state what information they want from database and let the database management system software take care of describing data structures for storing the data.

Relational (Data Model)

A data model is an abstract model that:

- organizes elements of data
- standardizes how they relate to one another
- and to properties of the real-world entities

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

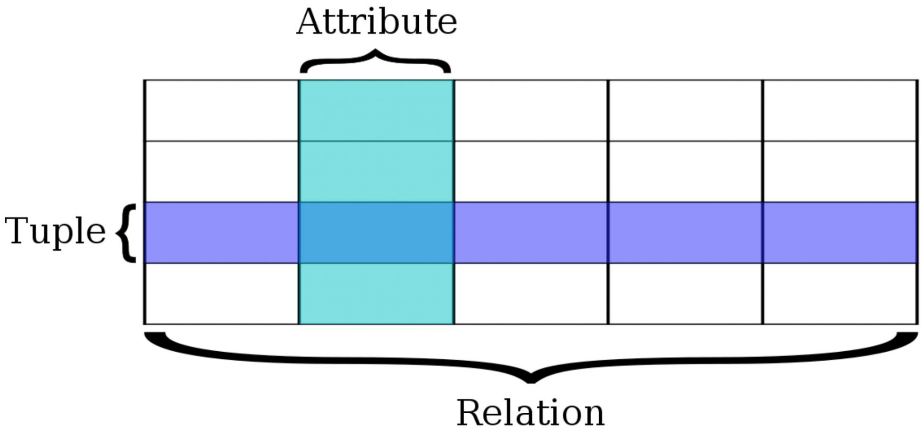
Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

Relational Model Concepts

- ❖ In the relational model, all data must be stored in relations (tables)
  - ➢ It uses the concept of a *mathematical relation*—as its basic building block, and has its theoretical basis in set theory and first-order predicate logic.
- ❖ Each relation consists of rows and columns.
  - ➢ Each row in the table represents a collection of related data values.



Relational Model Concepts (2)

In the formal relational model terminology, a row is called a *tuple,* a column header is called an *attribute,* and the table is called a *relation.*

Relation Name

Attributes

STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Tuples

Relational Model Concepts (3)

Relation schema *R*, denoted by *R*(*A*1, *A*2, ... , *An*), is made up of a **relation name** *R* and a list of attributes, ***A*1, *A*2, ... , *An***. Each **attribute** *Ai* is the name of a role played by some **domain *D*** in the relation schema *R*.

The data type describing the types of values that can appear in each column is represented by a *domain.* A **domain** *D* is a set of atomic values. By **atomic** we mean that each value in the domain is indivisible

The **degree** (or **arity**) of a relation is the number of attributes *n* of its relation schema.

A relation of degree seven would contain seven attributes:

E.g. STUDENT(Name, Ssn, Home_phone, Address, Office_phone, Age, Gpa)

Relational Model Concepts (4)

- A **relation** is a set of n-tuples $r = \{t_1, t_2, ..., t_m\}$. Each **n-tuple** t is an ordered list of n values $t=<v_1,v_2,...,v_n>$,where each value $v_i, 1 \le i \le n$,is an element of dom(A_i) or NULL value.
  - The ith value in tuple t corresponds to the attribute A_i
- **NULL** values represent attributes whose values are unknown, do not exist for some individual tuple or the attribute does not apply to this tuple

Characteristics of Relations

- A relation is defined as a set of tuples
- A tuple is an ordered list of n values
- **Tuple ordering** is not part of a relation definition because a relation attempts to represent facts at a logical or abstract level.
- A tuple in the relation can be interpreted as a fact or a particular instance of the assertion.
- Some relations may represent facts about entities, whereas other relations may represent facts about relationships

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

Relational Model Constraints

- Inherent model-based constraints (implicit constraints):
  - constraints that are inherent in the definition/assumptions of a particular data model hold in every database having that data model as its underpinning (e.g., a relation cannot have duplicate tuples)
- Schema-based constraints (explicit constraints)
  - expressed in the schemas of the data-model
- Application-based constraints(business rules):
  - enforced by the application programs

Which constraints should be considered?

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

Schema-based constraints (explicit constraints)

- ❖ **Domain Constraints**: Each attribute value must be either **null** or drawn from the domain of that attribute.
  - ➢ Data types associated with domains include standard numeric data types for integers (e.g., short integer, integer, long integer) and real numbers (float, double-precision float).

Schema-based constraints (explicit constraints) (2)

- **Key Constraints**: Two distinct tuples in any state of the relation cannot have identical values for (all) the attributes in the key.
  - A super-key specifies a uniqueness constraint that no two distinct tuples in any state r of R can have the same value.
  - Every relation has at least one default super key — the set of all its attributes.
  - A **key k** of a r**elation schema R** is a **superkey of R** with the additional property that removing any attribute A from K leaves a set of attributes K′ that is not a superkey of R any more.
  - **Minimal superkey property:** a superkey from which we cannot remove any attributes and still have the uniqueness constraint hold. This property is required for a key in a relation.
  - Possible to have multiple keys in a relational schema: <u>candidate key</u> and <u>primary key</u>.

Entity Integrity and Referential Integrity

Entity integrity constraint: no primary key value can be NULL

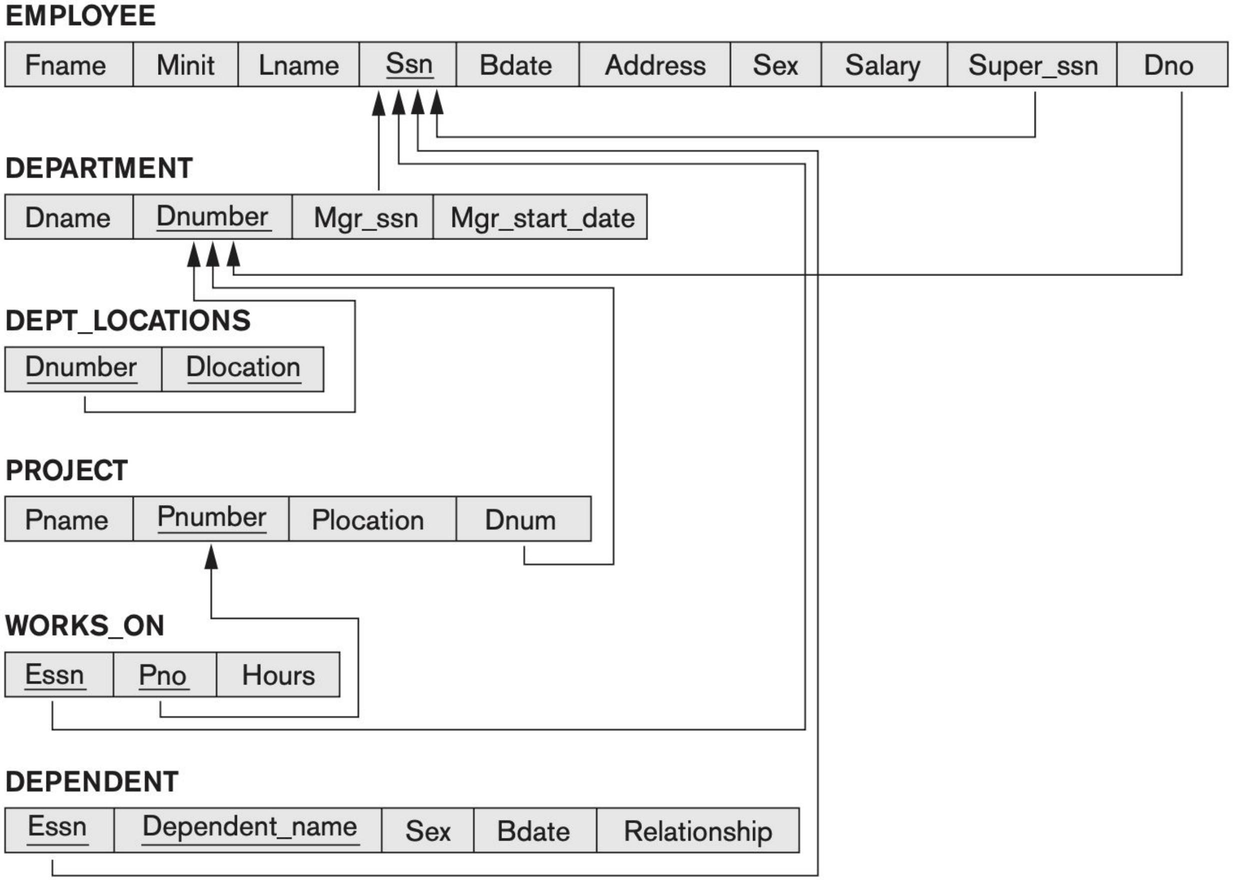
Foreign key: A foreign key of relation R is a set of its attributes intended to be used (by each tuple in R) for identifying/referring to a tuple in some other relation S. (R is called the referencing relation and S the referenced relation.)

- The attributes in the foreign key of R1 have the same domain as the primary key attribute in R2
- A value of foreign key in a tuple t1 of the current state r1 either occurs as a value of PK for some tuple t2 in the current state r2 or is null.

Referential integrity constraint: for every tuple in *R*, the tuple in *S* to which it refers must actually be in *S*.

All integrity constraints should be specified on the relational database schema.

With SQL we can define the primary key, unique key, NOT NULL, entity integrity, and referential integrity constraints among other constraints.



Referential integrity constraints

Exercise

Consider the following relations for a database that keeps track of automobile sales in a car dealership (OPTION refers to some optional equipment installed on an automobile):

CAR(Serial_no, Model, Manufacturer, Price)
OPTION(Serial_no, Option_name, Price)
SALE(Salesperson_id, Serial_no, Date, Sale_price)
SALESPERSON(Salesperson_id, Name, Phone)

Specify the foreign keys for this schema, stating any assumptions you make.

Semantic Integrity Constraints

Semantic Integrity Constraints: application-specific restrictions that are unlikely to be expressible in DDL

- Mechanisms called triggers and assertions can be used in SQL, to specify some of these constraints. It is more common to verify these types of constraints within the application programs.

State vs transition constraints:

- **State constraints:** Define the constraints that a valid state of the database must satisfy.
- **Transition constraints:** deal with state changes in the database

Relational Model Operations

- Operations on the relational model: **retrievals** and **updates**.
  - Update operations.Three basic operations that can change the states of relations in the database: **Insert**, **Delete**, and **Update**
    - Insert is used to insert one or more new tuples in a relation;
    - Delete is used to delete tuples
    - Update is used to change the values of some attributes in existing tuples.
  - Whenever these operations are applied, the **integrity constraints specified on the relational database schema should not be violated**.

Dealing with Constraint Violations through INSERT

INSERT Constraint violations:

- **domain constraint violation:** some attribute value is not of correct domain
- **entity integrity violation:** some attribute within a key of the new tuple has "value" **null**
- **key constraint violation:** key of new tuple is same as key of already-existing tuple
- **referential integrity violation:** foreign key of new tuple refers to non-existent tuple

Dealing with Constraint Violations through INSERT (2)

INSERT Constraint violations:

- **domain constraint violation:** some attribute value is not of correct domain
- **entity integrity violation:** some attribute within a key of the new tuple has "value" **null**
- **key constraint violation:** key of new tuple is same as key of already-existing tuple
- **referential integrity violation:** foreign key of new tuple refers to non-existent tuple

Solution: Reject the attempt to insert!

Dealing with Constraint Violations through DELETE

DELETE Constraint violations:

- **referential integrity violation:** a tuple referring to the deleted one exists.

Dealing with Constraint Violations through DELETE (2)

DELETE Constraint violations:

- **referential integrity violation:** a tuple referring to the deleted one exists.

SOLUTION:

- Reject the deletion
- Attempt to **cascade** (or **propagate**) by deleting any referencing tuples (plus those that reference them, etc., etc.)
- modify the foreign key attribute values in referencing tuples to **null** or to some valid value referencing a different tuple

Dealing with Constraint Violations through UPDATE

UPDATE Constraint violations:

- **Key constraint violation: primary key is changed so as to become same as another tuple**
- **Referential integrity violation:**
  a. foreign key is changed and new one refers to nonexistent tuple
  b. primary key is changed and now other tuples that had referred to this one violate the constraint

Dealing with Constraint Violations through UPDATE (2)

UPDATE Constraint violations:

- **Key constraint violation: primary key is changed so as to become same as another tuple**
- **Referential integrity violation:**
  a. foreign key is changed and new one refers to nonexistent tuple
  b. primary key is changed and now other tuples that had referred to this one violate the constraint

Solution:

- Reject the update;
- Modify the foreign key attribute values in referencing tuples to **null** or to some valid value referencing a different tuple

Transactions

- A transaction is an executing program that includes some database operations, such as reading from the database, or applying insertions, deletions, or updates to the database.
- At the end the transaction must leave the database in a valid or consistent state that satisfies all the constraints specified on the database schema.
- These retrievals and updates will together form an atomic unit of work against the database. For example, a transaction to apply a bank withdrawal will typically read the user account record, check if there is a sufficient balance, and then update the record by the withdrawal amount.


Q & A