Week 1

Friday, 24 April 2020 6:33 PM

- Conceptual Database
- Entity relationship model (Entities and their attributes can be described with Entity-Relationship Diagrams)
- Relational Data model

What is data?
Data :
◦ known facts that can be recorded and
have explicit meaning . . .

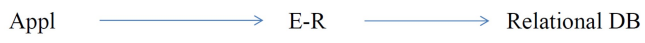
What is a database?
◦ . . . a collection of related data . . .

What is a database management system (DBMS)?
◦ DBMS: . . . a collection of programs that enables users to create and maintain a database . . .
◦ Database system: . . . The database and DBMS together . . .

--
Database system provides facilities to:
◦ Define a database - specifying the data items to be stored and their types,
◦ Construct a database - loading the data items and storing them on some storage medium (usually disk),
◦ Manipulate a database
◦ querying - i.e. retrieving relevant data,
◦ updating - i.e. adding, deleting or modifying data items:
◦ from one “correct” state to another “correct” state,
◦ reporting

--
Database system must be
◦ Timely - e.g. an airline database (fast response), a CAD system (must be interactive),
◦ Multi-user - e.g. trading system,
◦ Modifiable - must be able to be extended or reorganised, e.g. to cope with new laws, requirements, business conditions,
◦ Secure - different classes of users may need different levels of access,
◦ No redundancy,
◦ Robust - e.g. power failure during an update - must be able to recover to a consistent state.

1. Conceptual Database Design



Database Design

1.1 Entity and Attributes
• Entities represent things in the real word.
Each entity has values for each attribute.

• An entity type usually has a key: a set of attributes that uniquely identifies an entity. For example:
– {payroll number} is a key of RESEARCHER,
– {name} is a key of DEPARTMENT.

• Some entity types do not have a key of their own. Such entity types are called weak entity types.
• Entities of a weak entity type can be identified by a partial key and by being related to another entity type - owner.

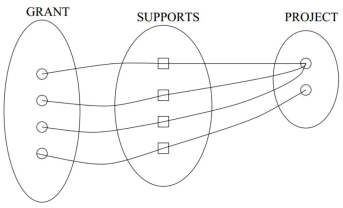
• Attributes describe properties of entities.
• Attributes may be
– simple(atomic) e.g. sex = 'Female', or
– composite e.g. name consists of title (Dr), Initials (C.C.), family name (Chen).
– single-valued e.g. student number, name, or
– multivalued e.g. keywords = neural networks, computer graphics, databases.
• An attribute can have a null value if, for example:
– there is no suitable value e.g. a student may have no interests: keywords = NULL
– the true value is not known e.g. the marriage date of a person is not known: marriage date = NULL.
• A derived attribute is one whose value can be derived from other attributes and entities. e.g. number of students.

• An important constraint is the key constraint: in any extension of the entity type, there cannot be two entities having the same values for their key attributes.

Constraints on relationship types:

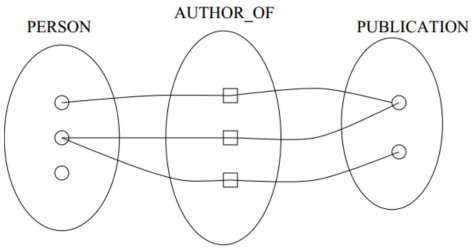
-- *Cardinality ratio constraint:* specifies the number of relationship instances an entity can participate in.
• Example: A research grant supports only one research project, but a research project may be supported by many grants. PROJECT:GRANT is a 1 : N relationship.

- This is illustrated in the occurrence diagram below:



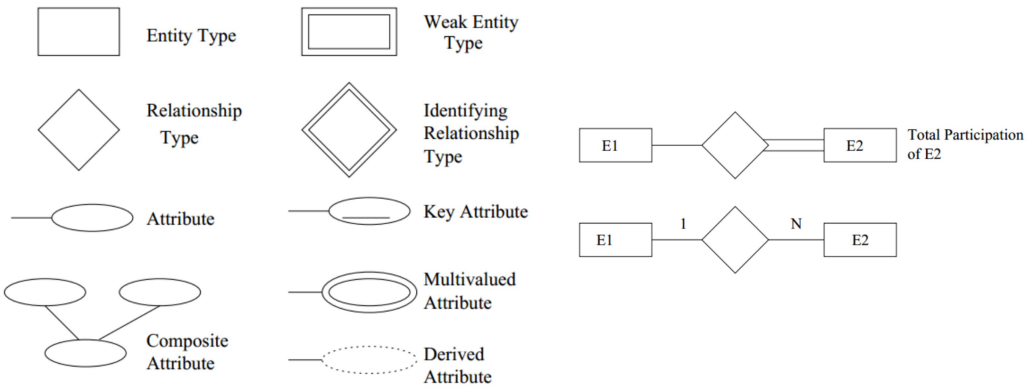
- We can also show this in an ERD:

-- *Participation constraint:* participation of an entity in a relationship can be:
• total: every entity must participate e.g. every publication has an author.
• partial: not necessarily total. e.g. not every person has publications.



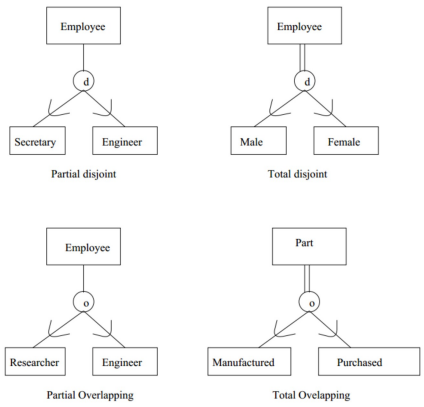
- This can be shown with an ERD like the one below:

- The notation used for ERDs is summarised in Elmasre/Navathe Figure 3.15.



• A subclass inherits all the attributes of the superclasses.
Many extensions of ER model:
– Specialisation: the process of defining a set of subclasses of an entity type; this entity type is called the superclass of the specialization.
– Generalisation: a reverse process of specialisation.

--A specialisation:
– may be either total or partial; and
– may be either disjoint or overlapping



Design Principles
• Faithfulness: reflect reality.
• Avoid redundancy.
• Picking the right kind of element.

The Relational Data Model

2.1 Structures

- In the relational model, everything is described using relations.
- A relation can be thought of as a named table.
- Each column of the table corresponds to a named attribute.
- The set of allowed values for an attribute is called its domain.
- Each row of the table is called a tuple of the relation.
- N.B. There is no ordering of column or rows.
- composite and multivalued attributes are allowed in ER model, but not allowed in relational data model.

- Keys are used to identify tuples in a relation.
- A superkey is a set of attributes that uniquely determines a tuple.
- A candidate key is a superkey,最小不可再分超键。
- A primary key is a designated candidate key.
- Foreign keys are used to refer to a tuple in another relation.

2.2 Integrity constraints

- There are several kinds of integrity constraints that are an integral part of the relational model:
- 2.2.1 Key constraint: candidate key values must be unique for every relation instance.
- 2.2.2 Entity integrity: an attribute that is part of a primary key cannot be NULL.
- 2.2.3 Referential integrity: The third kind has to do with “foreign keys”.

<2, Dr. V. Ciesielski>

insert

RESEARCHER

Person#	Name
1	Dr.C.C.Chen
2	Dr.R.G.Wilkinson

STUDENT

Person#	Name
1	Dr.C.C.Chen
3	Ms.K.Juliff
4	Ms.J.Gledill
5	Ms.B.K.Lee

COURSE

Department	Name
Psychology	Ph.D.
Comp.Sci.	Ph.D.
Comp.Sci.	M.Sc.
Psychology	M.Sc.

ENROLMENT

Enrolment#	Supervisee	Supervisor	Department	Name
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

2/14/20 13

- *Insertions*: When inserting, we need to check
  - that the candidate keys are not already present,
  - that the value of each foreign key either
    - –is all null, or
    - – is all non-NULL and occurs in the referenced relation.

Examples:

1. Insert < 2, *Dr.V.Ciesielski* > into RESEARCHER

Allowed? No. Violates a key constraint.

Action? Reject or allow the user to correct.

2/14/20 17

<Comp.Sci., NULL>

insert

RESEARCHER

Person#	Name
1	Dr.C.C.Chen
2	Dr.R.G.Wilkinson

STUDENT

Person#	Name
1	Dr.C.C.Chen
3	Ms.K.Juliff
4	Ms.J.Gledill
5	Ms.B.K.Lee

COURSE

Department	Name
Psychology	Ph.D.
Comp.Sci.	Ph.D.
Comp.Sci.	M.Sc.
Psychology	M.Sc.

ENROLMENT

Enrolment#	Supervisee	Supervisor	Department	Name
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

2/14/20 14

2. Insert < *Comp.Sci.,NULL* > into COURSE

Allowed? No. Violates the entity integrity constraint.

Action: Reject or correct.

3. Insert < 5, 6, 2, *Psychology, Ph.D.* > into ENROLMENT

Allowed? No. Violates a referential integrity constraint (There is no person number 6).

Action: Reject, correct or accept after insertion of person number 6.

2/14/20 18

<5, 6, 2, Psychology, Ph.D>

insert

RESEARCHER

Person#	Name
1	Dr.C.C.Chen
2	Dr.R.G.Wilkinson

STUDENT

Person#	Name
1	Dr.C.C.Chen
3	Ms.K.Juliff
4	Ms.J.Gledill
5	Ms.B.K.Lee

COURSE

Department	Name
Psychology	Ph.D.
Comp.Sci.	Ph.D.
Comp.Sci.	M.Sc.
Psychology	M.Sc.

ENROLMENT

Enrolment#	Supervisee	Supervisor	Department	Name
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

2/14/20 15

RESEARCHER

Person#	Name
1	Dr.C.C.Chen
2	Dr.R.G.Wilkinson

STUDENT

Person#	Name
1	Dr.C.C.Chen
3	Ms.K.Juliff
4	Ms.J.Gledill
5	Ms.B.K.Lee

COURSE

Department	Name
Psychology	Ph.D.
Comp.Sci.	Ph.D.
Comp.Sci.	M.Sc.
Psychology	M.Sc.

ENROLMENT

Enrolment#	Supervisee	Supervisor	Department	Name
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

- *Deletions*: When deleting, we need to check referential integrity – check whether the primary key occurs in another relation.

Examples:

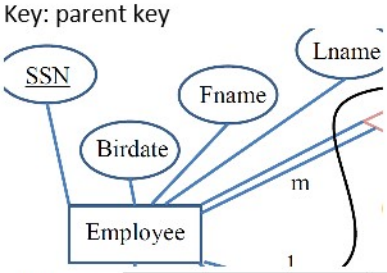
1. Delete tuple with Person# 2 from RESEARCHER

Allowed? No. Violates the referential integrity.

Action: Reject, correct or modify the ENROLMENT tuple by

- deleting it (note that the this requires another integrity check, possibly causing a cascade of deletions), or
- setting the foreign key value to NULL (note this can't be done if it is part of a primary key), or
- setting the foreign key value to another acceptable value.

There are 7 steps to do: ER to the relational data model
Step 1:
RD:
Writing down all of the entities and their simple attributes, assign a label to primary key.
1a: for each specialised entity
Create new relation
Attributes: put its parent key + its simple attribute
Key: parent key



Employee

SSN	Fname	Lname	Birdate

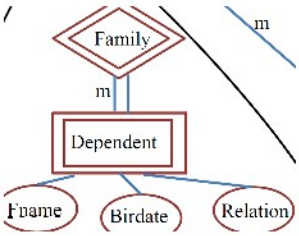
Department

Name	Location

Project

Pname	Pnumber

Step 2:
For each weak entity type with owner entity type, writing down all simple attributes.
RD:
Create a relation. i.e. Department
Key: The foreign key plus the partial key of the weak entity.



Employee

SSN	Fname	Lname	Birdate

Department

Name	Location

Project

Pname	Pnumber

Dependent

SSN	Fname	Birdate	Relation

Step 3
For each 1:1 relationship
RD: add the attributes to corresponding relations (participate totally)

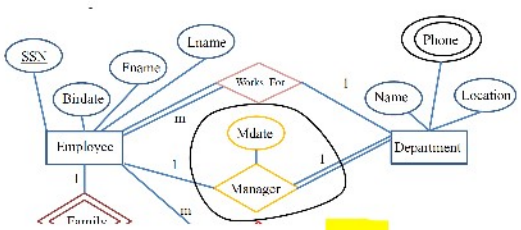
Step 3

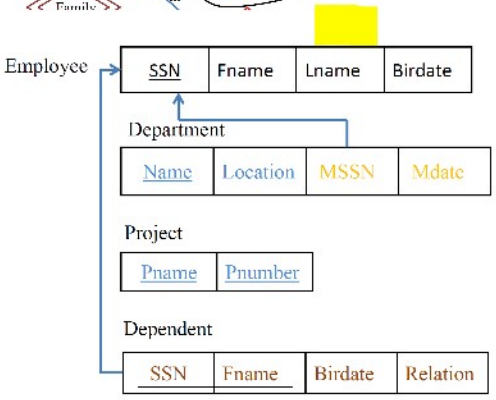
For each 1:1 relationship

RD: add the attributes to corresponding relations (participate totally)

add the attribute of the primary key of Employee to Department as foreign key.

Add the simple attributes.





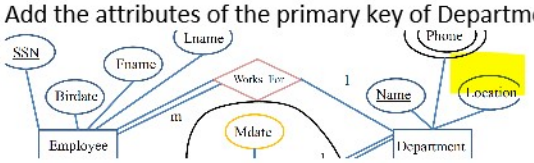
Step 4

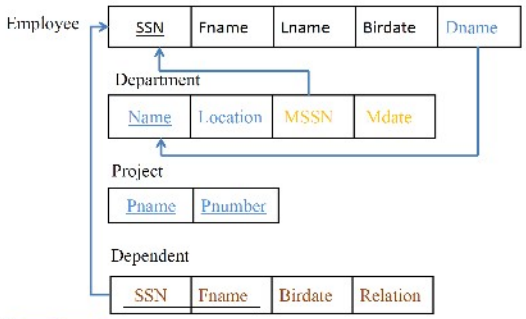
For each regular 1:N relationship

RD:

Add the attributes in employee (N)

Add the attributes of the primary key of Department to Employee as foreign key





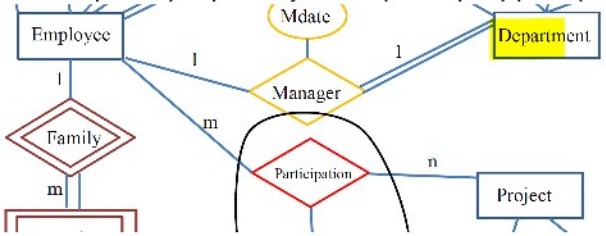
Step 5

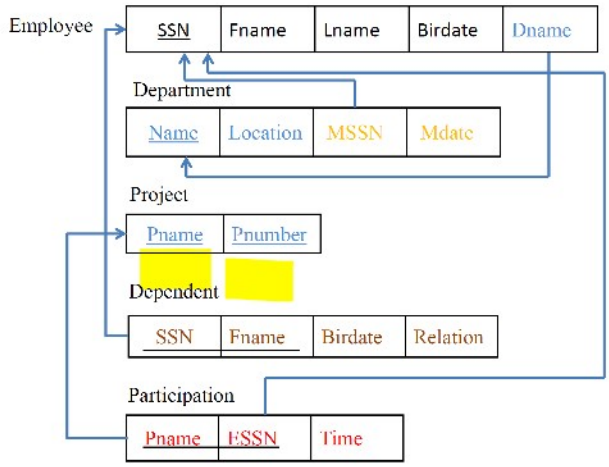
For each N:M relationship

RD:

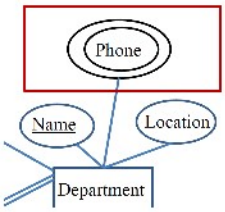
Create new relation. i.e. Participation

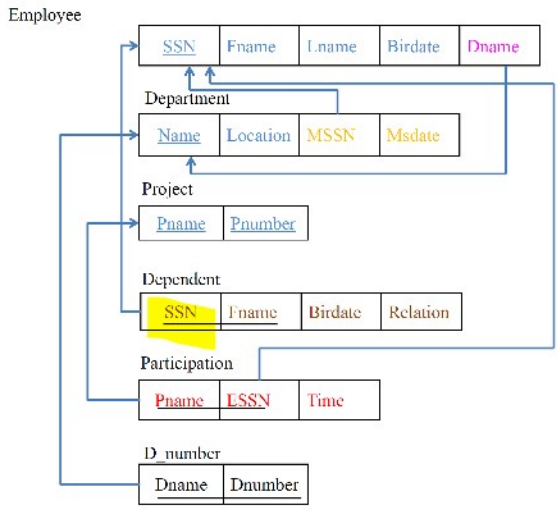
Add the primary key of Project and primary key pf Employee as foreign key





Step 6
For each multivalued attribute
RD:
Create a new relation i.e.D_number for Phone
Add this attribute together with the key of Department as a foreign key to department.





Step 7
For each n-ary relationship type (n > 2). Create a new relation with
As for step 5.

- Example: ER—RDB