Database Management Systems (DBMS)

Lec 13: Relational database design

Ramesh K. Jallu

IIIT Raichur

Date: 16/03/21

Recap

- Enhanced Entity-Relationship (EER) Model
- The concepts of EER model
  - Subclass and Superclass, Inheritance
  - Specialization and Generalization
  - Union or Category
  - Aggregation

Overview

- Mapping between ER and Relational schema
- A quick recap of keys and constraints in relational schema
- Introduction to normalization

Converting an ER diagram to a relational schema

- Both the ER model and the relational database model are abstract, logical representations of real-world enterprises

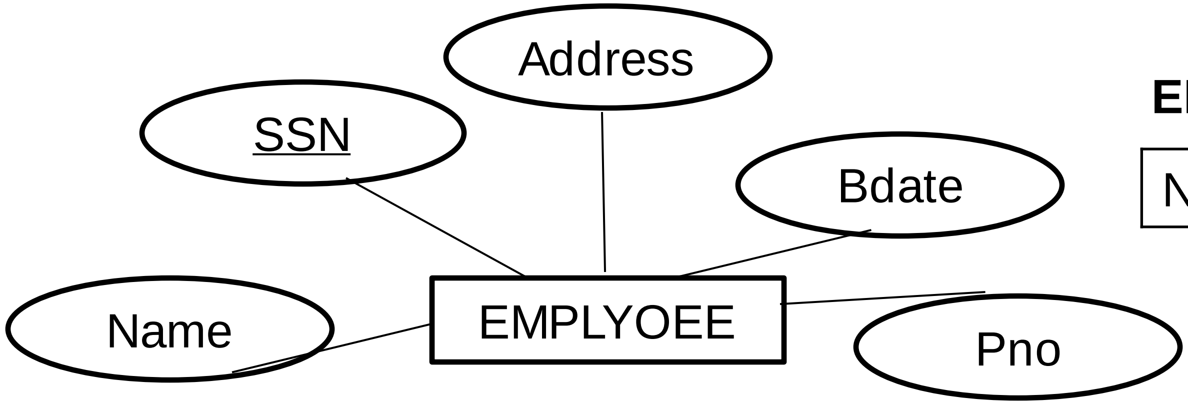
- As the two models employ similar design principles, we can convert an ER design into a relational design

- For each entity set and for each relationship set in the database design, there is a unique relation schema

- The constraints in ER design can be mapped to constraints on relational schemas

1. Representation of strong entity types

- If ***E*** be a strong entity type with attributes $a_1, a_2, \ldots, a_n,$ we represent this entity with a schema called ***E*** with *n* distinct attributes
- Each tuple in the relational schema corresponds an entity in ***E***
- The primary key of the entity type serves as the primary key of the resulting schema

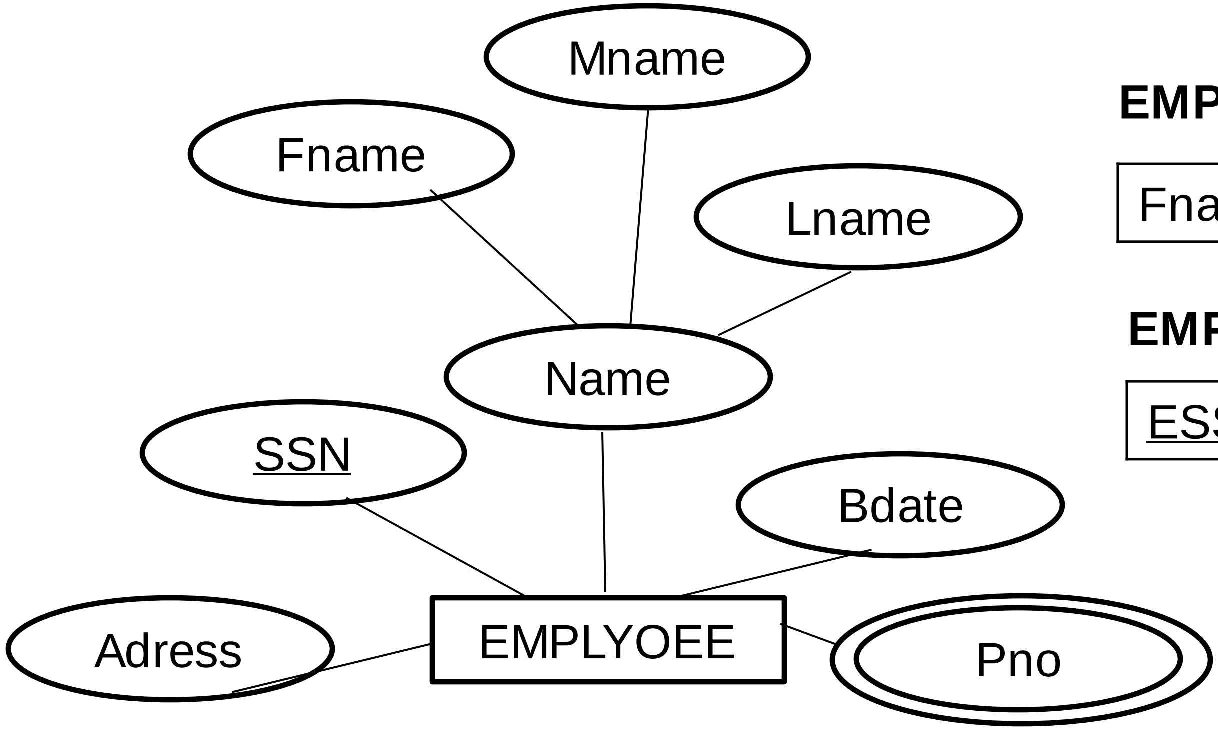


EMPLOYEE

Name	SSN	Address	Bdate	Pno

2. Representation of Strong Entity Sets with non-simple Attributes

- Derived attributes are not explicitly represented in the relational data model
- For a multivalued attribute ***M***, we create a relation schema ***R*** with an attribute ***A*** that corresponds to ***M*** and attributes corresponding to the primary key of the entity set or relationship set of which ***M*** is an attribute



EMPLOYEE

Fname	Mname	Lname	SSN	Address	Bdate

EMP_Pno

ESSN	Ph_no

3. Representation of Weak Entity Sets

- Let ***A*** be a weak entity set with attributes $a_1, a_2, \ldots, a_m$. Let ***B*** be the strong entity set on which ***A*** depends. Let the primary key of ***B*** consist of attributes $b_1, b_2, \ldots, b_n$

- We represent the entity set ***A*** by a relation schema called ***A*** with one attribute for each member of the set $\{a_1, a_2, \ldots, a_m\} \cup \{b_1, b_2, \ldots, b_n\}$

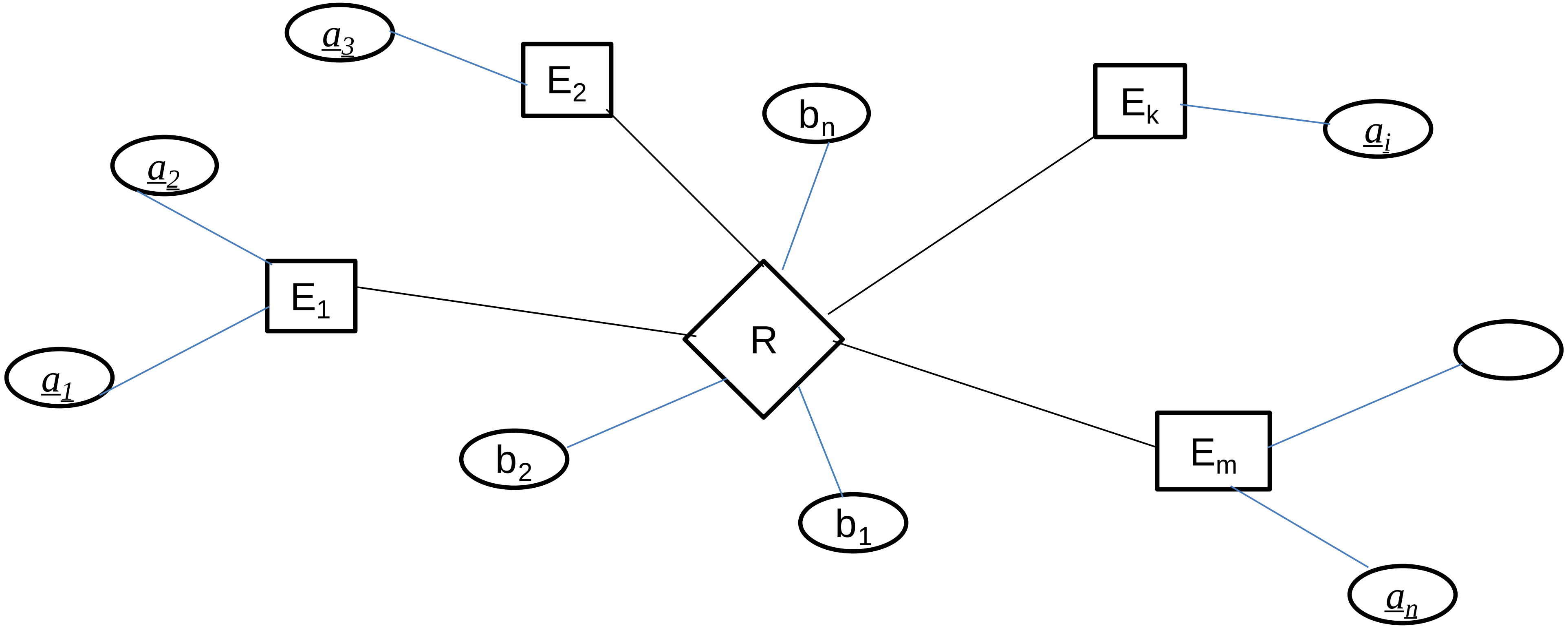
A

$\underline{a_1}$	$\underline{a_2}$	$\underline{\ldots}$	$\underline{a_m}$	$\underline{b_1}$	$\underline{b_2}$	$\underline{\ldots}$	$\underline{b_n}$

4. Representation of relationships

- Let ***R*** be a relationship with descriptive attributes $b_1, b_2, \ldots, b_n$
- Let $a_1, a_2, \ldots, a_m$ be the set of attributes formed by the union of the primary keys of each of the entity sets participating in ***R***
- We represent this relationship set by a relation schema called ***R*** with one attribute for each member of the set $\{a_1, a_2, \ldots, a_m\} \cup \{b_1, b_2, \ldots, b_n\}$
- We also create foreign-key constraints on the relation schema ***R***

4. Representation of relationships



R

a_1	a_2	…	a_m	b_1	b_2	…	b_n

5. Representing cardinality constraints

- For many-to-many relationships, the union of the primary keys is a minimal superkey and is chosen as the primary key

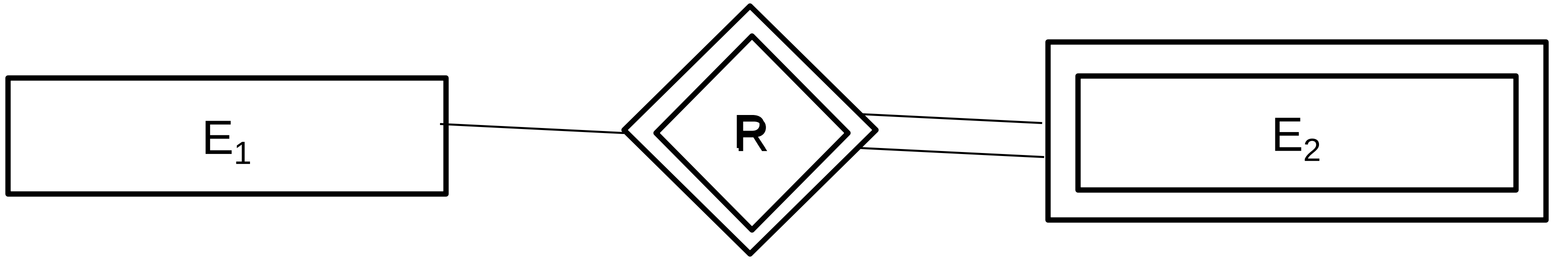
- For one-to-many and many-to-one relationships, the primary key of the “many” side is a minimal superkey and is used as the primary key

- For one-to-one relationships, the primary key of either one of the participating entity sets forms a minimal superkey, and either one can be chosen as the primary key of the relationship set

6. Relationship between weak entity type and strong entity type

- Recall
  - The identifying relationship is many-to-one from the weak entity set to the identifying entity set
  - The participation of the weak entity set in the relationship is total
  - The identifying relationship set should not have any descriptive attributes
  - The of a weak entity set includes the primary key of the strong entity set

- The schema for the relationship set linking a weak entity type to its corresponding strong entity type is ***redundant*** and does not need to be present in a relational database design



Summary

ER MODEL	RELATIONAL MODEL
Entity type	*Entity* relation
1:1 or 1:N relationship type	Foreign key (or *relationship* relation)
M:N relationship type	*Relationship* relation and *two* foreign keys
n-ary relationship type	*Relationship* relation and *n* foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

Relational database design

- The goal of relational database design is to generate a set of relation schemas that allows us to
  - store information without unnecessary redundancy
  - retrieve information easily
  - achieve accuracy and consistency of data

- This is accomplished by designing schemas that are in an appropriate *normal form*

- The *normalization* procedure consists of applying a series of tests to relations and decompose the relations when necessary

Normalization

- Decide if a given relation schema is in “good form” with the aid of normal forms
- If a given relation schema is not in “good form,” then we decompose it into a number of smaller relation schemas, each of which is in an appropriate normal form
  - The decomposition must be a lossless decomposition

Keys (recap)

- Keys are used to distinguish tuples in a given relation
  - **Superkey**: A subset ***K*** of attributes that ***uniquely*** identifies a tuple in the relation
  - If t_i and t_j are any two distinct tuples in a relation, then $t_i[K] \neq t_j[K]$
  - Any relation contains a trivial superkey

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Keys (recap)

- If ***K*** is a superkey, then so is any superset of ***K***
- We are interested in ***minimal*** superkey (a.k.a. Key)
- A relation schema may have more than one key. Such minimal superkeys are called ***candidate keys***
- One of the candidate keys is designated as ***primary key*** and other candidate keys are designated as ***unique keys***

Constraints in relation schema (recap)

1. Key constraints
   - Two distinct tuples cannot have identical values for (all) the attributes

2. Entity integrity constraints
   - Primary key value cannot be NULL

3. Referential integrity constraints
   - Used to maintain consistency among tuples in two relations

Foreign key (recap)

- Let R_1 and R_2 be two relations. A set of attributes K in R_1 is a ***foreign key*** of R_1 that references to relation R_2
  - The attributes in K have the same domain(s) as the primary key attributes of R_2
  - A value of K in a tuple t_i of R_1 either occurs as a value of primary key in some tuple of t_j of R_2 or is **NULL**
  - $t_i[K] = t_j[\text{primary key}]$

- R_1 is called the referencing relation and R_2 is called the referenced relation

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
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

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

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Other constraints in relation schema (recap)

- Semantic integrity constraints
- Functional dependency constraint
  - Establishes a functional relationship among two sets of attributes ***X*** and ***Y***, and is denoted by $X \rightarrow Y$
  - The value of ***X*** determines the value of ***Y*** in all states of a relation
    - I.e., for any two tuples t_1 and t_2 in ***R*** that have $t_1[X] = t_2[X]$, implies that $t_1[Y] = t_2[Y]$
    - In other words, the values of the ***X*** component of a tuple uniquely (or functionally) determine the values of the ***Y*** component

Example: functional dependency (recap)

EXAMPLE

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_2	b_2	c_2	d_2
a_2	b_3	c_2	d_3
a_3	b_3	c_2	d_4

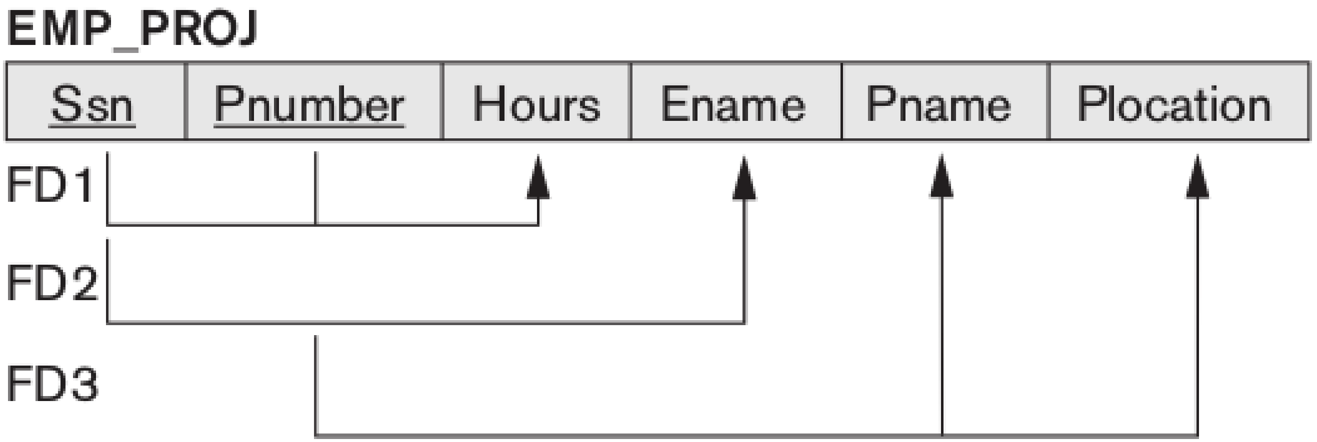
A → ***C*** is satisfied

C → ***A*** is ***not*** satisfied

D → ***{C,A}?***

{A,B} → ***{C,D}?***

Example: functional dependency



1. Ssn $\rightarrow$ Ename
2. Pnumber $\rightarrow$ {Pname, Plocation}
3. {Ssn , Pnumber} $\rightarrow$ Hours

- FDs are the basis to develop a formal methodology for testing and improving relation schemas

Normalization of relations

- The concept of normalization was proposed by Codd in 1972
- Normalization of data we mean analyzing the given relation schemas based on their FDs and primary keys
- The objective of normalization is to (i) minimizing redundancy, and (ii) minimizing the insertion, deletion, and update anomalies
- An unsatisfactory relation schema that does not meet the condition for a normal form is decomposed into smaller relation schemas that meet the tests and possess the desirable properites

In summary

- A formal framework for analyzing relation schemas based on their keys and on the functional dependencies among their attributes

- A series of normal form tests that can be carried out on individual relation schemas so that the relational database can be normalized to any desired degree

Thank you!