DATABASDESIGN FÖR INGENJÖRER - 1056F

Sommar 2005

En introduktionskurs i databassystem

http://user.it.uu.se/~udbl/dbt-sommar05/
alt. http://www.it.uu.se/edu/course/homepage/dbdesign/st05/

Kjell Orsborn (Ruslan Fomkin)
Uppsala Database Laboratory
Department of Information Technology, Uppsala University,
Uppsala, Sweden

UPPSALA
UNIVERSITET

Introduction to the Relational Model

Elmasri/Navathe ch 5, 7

Kjell Orsborn &
Ruslan Fomkin

Department of Information Technology
Uppsala University, Uppsala, Sweden

UPPSALA
UNIVERSITET

The Relational Model

- The relational model was introduced by Dr. Edgar (Ted) F. Codd (1924-2003) in 1970.
  - Dr. Codd. a mathematician from Oxford (UK), was at that time working as an IBM researcher in the IBM San Jose Research Laboratory (USA).
- Many DBMS's are based on the relational data model.
- It supports simple declarative, but yet powerful, languages for describing operations on data.
- Operations in the relational model applies to relations (tables) and produce new relations.
  - This means that an operation can be applied to the result of another operation and that several different operations can be combined.
  - Operations are described in an algebraic notation that is based on relational algebra.

Relations as mathematical objects

- In set theory, a relation is defined as a subset of the product set (cartesian product) of a number of domains (value sets).
- The product set of the domains $D_1,D_2,...,D_n$ is written as $D_1 \times D_2 \times .. \times D_n$.
- $D_1 \times D_2 \times ... \times D_n$ constitute the set of all ordered sets $<v_1,v_2,...,v_n>$ such that v_i belongs to D_i for all i.
  - If n=2, D_1={T, F} and D_2={P, Q, R} one gets the product sets:
    $D_1 \times D_2$ = {<T,P>,<T,Q>,<T,R>,<F,P>,<F,Q>,<F,R>}
    $D_2 \times D_1$ = {<P,T>,<P,F>,<Q,T>,<Q,F>,<R,T>,<R,F>}
  - For example, we have the relations:
    $R_1 \subseteq D_2 \times D_1$ R_1= {<P,T>,<Q,T>,<R,T>}
    $R_2 \subseteq D_2 \times D_1$ R_2= {<P,T>,<P,F>}
- Members of a relation is called **tuples**. If the relation is of **degree** n, the tuples are called *n-tuples*.

UPPSALA UNIVERSITET

Relation schema and instance

- $A_1, A_2, \ldots, A_n$ are attributes
- $R (A_1, A_2, \ldots, A_n)$ is a relation schema
  - *Customer-schema(customer-name, customer-street, customer-city)*
- r(R) is a relation on the relation schema R
  - *customer (Customer-schema)*
- The current values (*relation instance*) of a relation are specified by a table.
- An element *t* of *r* is a tuple - represented by a *row* in a table customer

an attribute

customer

customer-name	*customer-street*	*customer-city*
Jones	Main	Harrison
Smith	North	Rye
Curry	North	Rye
Lindsay	Park	Pittsfield

a relation

a tuple

First Normal Form

- Only simple or atomic values are allowed in the relational model.
- Attributes is not allowed to have composite or multiple values.
- The theory for the relational model is based on these assumptions which is called:

The first normal form assumption

Null values

- A special value, **null** or ⊥, can sometimes be used as an attribute value.
- Every occurence of null is unique. Thus, two occurences of null is not considered to be equal even if they are represented by the same symbol.
- null is used:
  - when one does not know the actual value of an attribute.
  - when a certain attribute does not have a value.
  - when an attribute is not applicable.
- Examples of the use of null are showed later.

Keys

- Because relations are sets, all tuples in the relation are different.
- There is usually a subset k of the attributes in a relation schema R, that has the characteristic that if the tuples
  t1, t2 $\in$ r(R) and t1 $\neq$ t2, the following holds:
  t1[k] $\neq$ t2[k] (i.e. the value of k in t1 $\neq$ the value of k in t2)
- Every such subset k is called a **superkey** for R.

Keys - continued . . .

- A superkey k is *minimal* if there is no other superkey k' such that k‘ ⊂ k.
- Every minimal superkey (NOTE! there can be more than one) is called a **candidate key** for R.
- The candidate key chosen by the database designer as the key for R is called R:s **primary key** or just **key**.
- In addition, term **foreign key** is used when a tuple is referenced, from another relation, with its key.

Key examples

- Example superkey:
  - {customer-name, customer- street} and {customer- name} are both superkeys of *Customer*, if no two customers can possibly have the same name.
- Example candidate key:
  - {customer- name} is a candidate key for *Customer* , since it is a superkey (assuming no two customers can possibly have the same name), and no subset of it is a superkey.

Integrity constraints

for a relational database schema

- 1. Domain constraint
  - attribute values for attribute A shall be atomic values from dom(A)
- 2. Key constraint
  - candidate keys for a relation must be unique
- 3. Entity integrity constraint
  - no primary key is allowed to have a null value
- 4. Referential integrity constraint
  - a tuple that refers to another tuple in another relation must refer to an existing tuple
- 5. Semantic integrity constraint
  - e.g. “an employee’s total work time per week can not exceed 40 hours for all projects taken all together”

Steps in translation from E-R model to relational model

- Translation of entity types and their attributes
  - Step 1) Entity types
  - Step 2) Weak entity types
- Translation of relationships
  - Step 3) 1-1 Relationship
  - Step 4) 1-N Relationship
  - Step 5) M-N Relationship
- Translation of multivalued attributes and relationships
  - Step 6) Multivalued attributes
  - Step 7) Multivalued relationships

Determining keys from E-R types

- **Strong entity type**. The primary key of the entity type becomes the primary key of the relation.
- **Weak entity type**. The primary key of the relation consists of the union of the primary key of the strong entity type and the discriminator of the weak entity type.
- **Relationship type**. The union of the primary keys of the related entity types becomes a super key of the relation.
  - For binary many-to-many relationship types, above super key is also the primary key.
  - For binary many-to-one relationship types, the primary key of the "many" entity type becomes the relation's primary key.
  - For one-to-one relationship types, the relation's primary key can be that of either entity type.

Translating entity types and their attributes

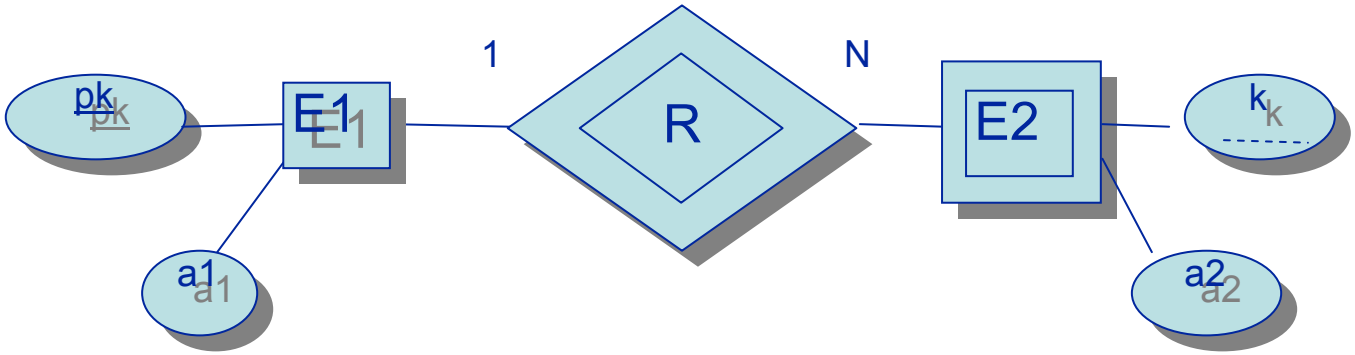
- Step 1: Entity types - a strong entity type reduces to a table with the same attributes.
  - Key attributes (primary key - pk) is made the primary key column(s) for the table. Each attribute gets their own column.
  - Composite attributes are normally represented by their simple components.
  - Example customer schema and table:

Customer(social-security, customer-name, c-street, c-city)

social-security	*customer-name*	*c-street*	*c-city*
321-12-3123	Jones	Main	Harrison
019-28-3746	Smith	North	Rye
677-89-9011	Hayes	Main	Harrison

Translating entity types cont. . .

- Step 2: **Weak entity types** - a weak entity type becomes a table that includes a column for the primary key of the identifying strong entity type .



pk	a1

pk	k	a2

Translating entity types cont. . .

- The table corresponding to a relationship type linking a weak entity type to its identifying strong entity type is redundant.

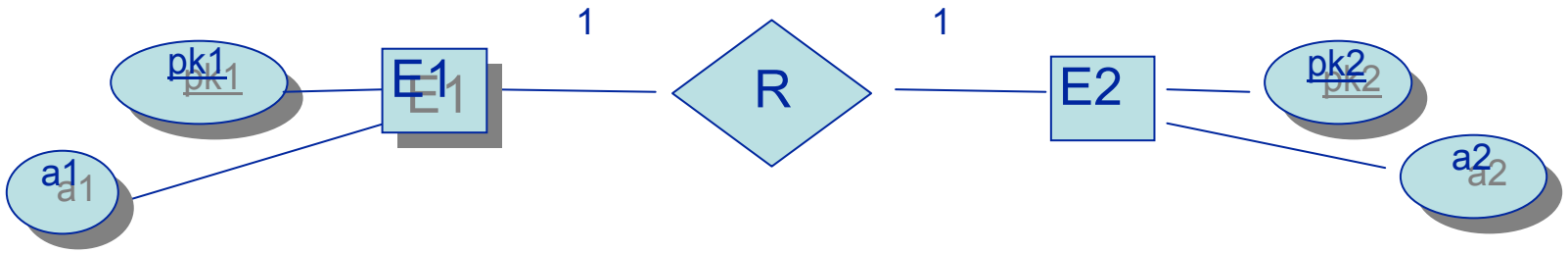
- Example of the payment schema and table:
  - The payment table already contains the information that would appear in the loan-payment table (i.e., the columns loan-number and payment-no).

Payment(loan-number, payment-no, pay-date, amount)

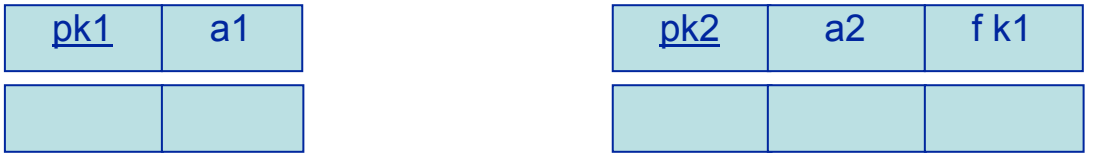
loan-number	*payment-no*	*pay-date*	*amount*
L-17	5	10 May 1996	50
L-23	11	17 May 1996	75
L-15	22	23 May 1996	300

Translating relationship types

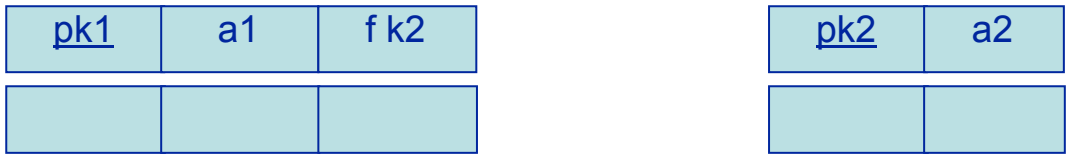
- Step 3: 1-1 Relationship types
  - The foreign key column (fk) is a copy of the other entity's primary key column (pk). The values in a fk-column point to unique row in the other table, and thus implement the relationship.



Alt 1:

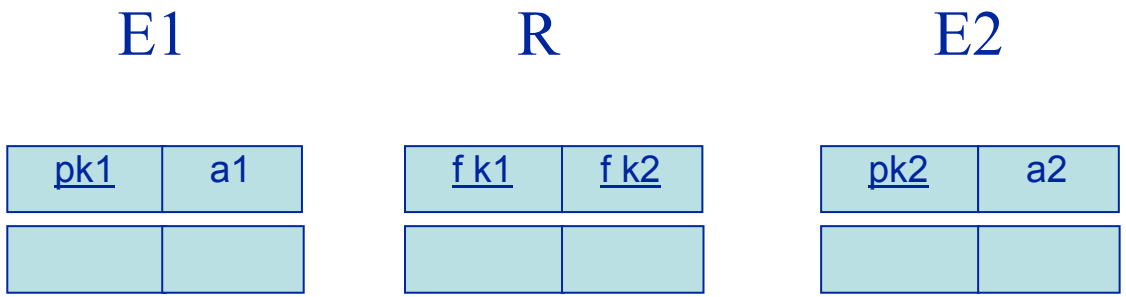


Alt 2:

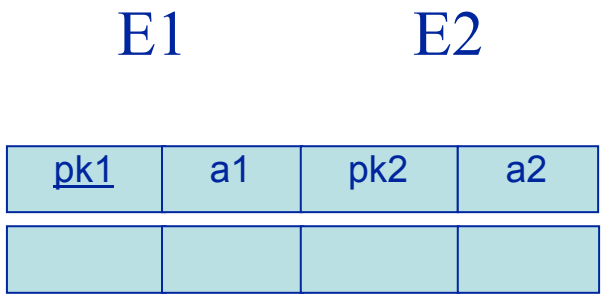


Translating 1-1 relationship types cont. . .

Alt 3:

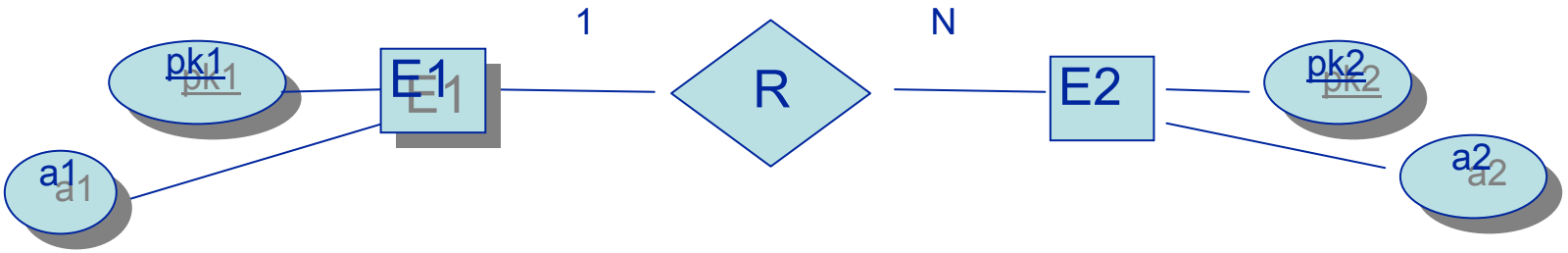


Alt 4:



Translating relationship . . . cont. . .

- Step 4: 1-N Relationship types
  - Include the primary key of the "1-side" as a foreign key on the "N-side", (i.e. the foreign key column is placed on the entity on the N-side).
  - Alternatively, an extra table (R) is created whose primary key is a foreign key composed by the primary key from the N-side.

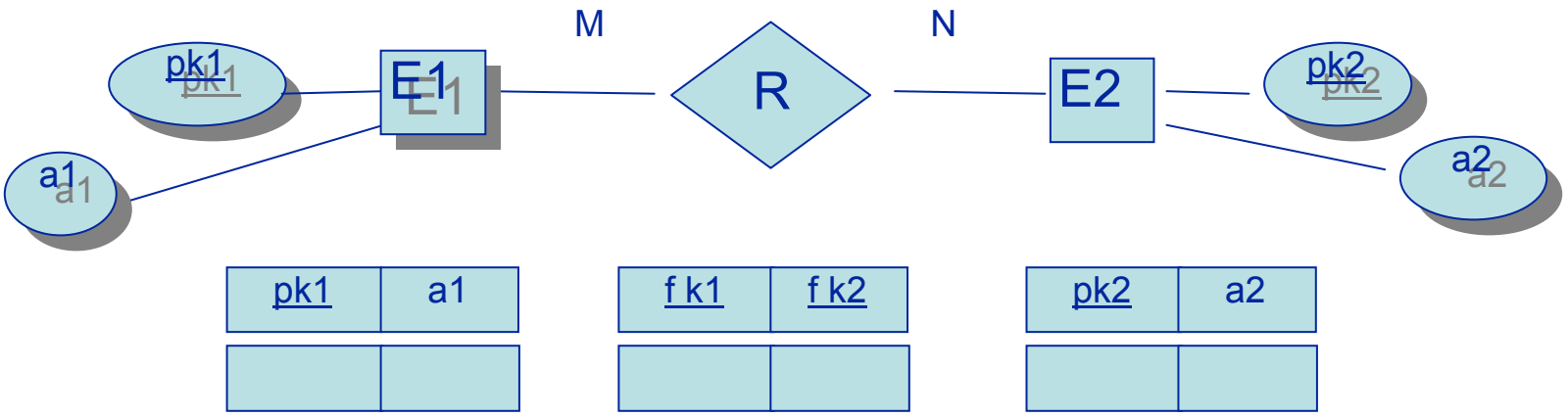


Alt 1:

Alt 2:

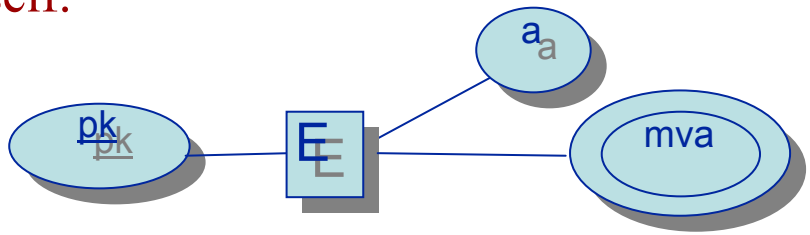
Translating relationship . . . cont. . .

- Step 5: M-N Relationship types
  - Always a separate table with columns for the primary keys of the two participating entity types, and any descriptive attributes of the relationship type.



Translating relationship . . . cont. . .

- Step 6: Multivalued attributes
  - A separate table is created for the multivalued attribute. Its primary key is composed of the owning entity's primary key, and the attribute value itself.



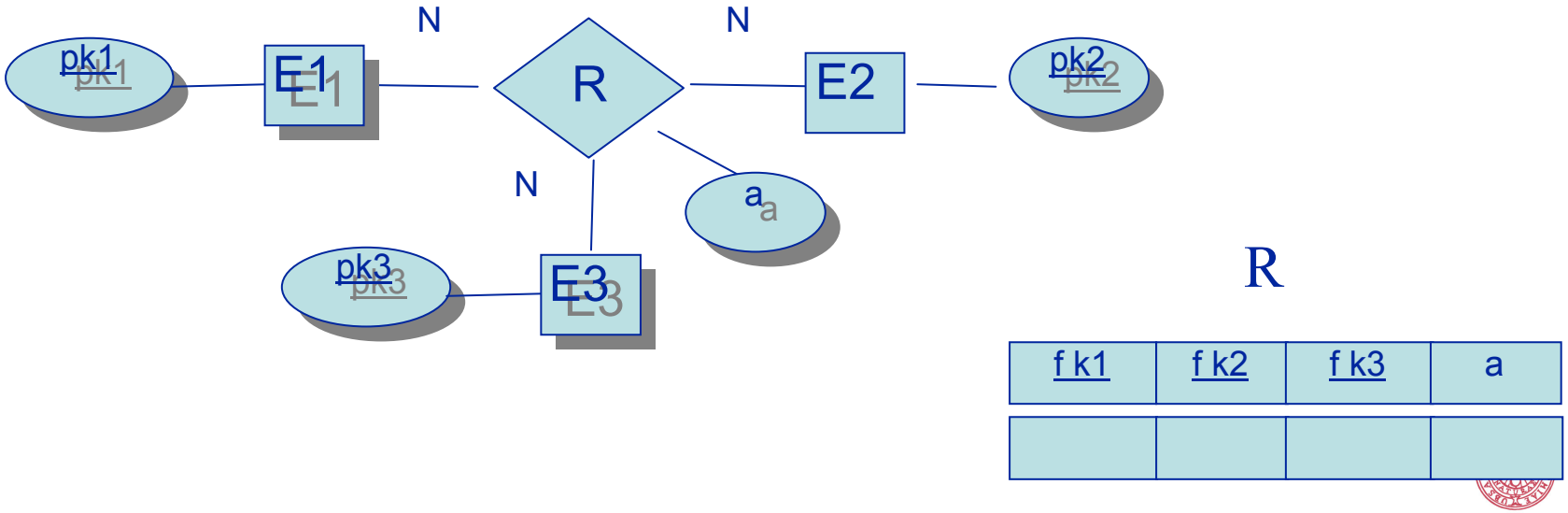
E E-MVA

pk	a

pk	mva

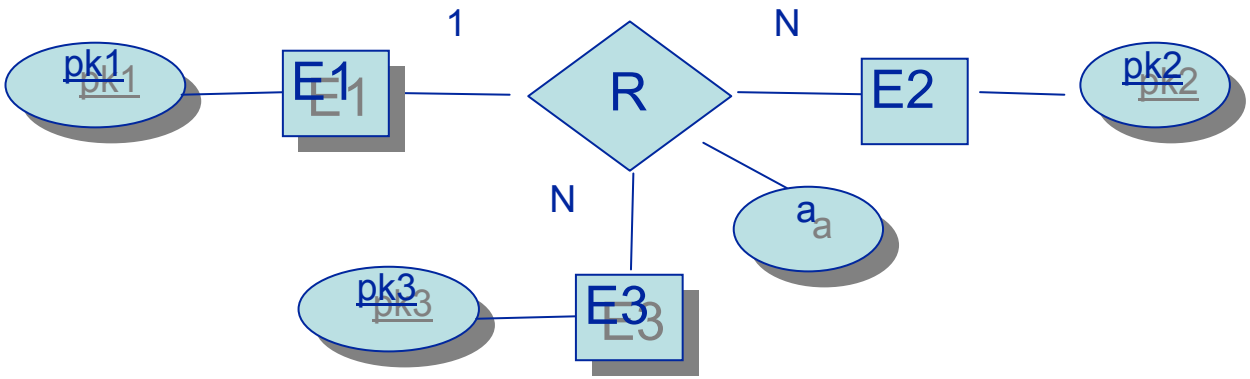
Translating relationship . . . cont. . .

- Step 7: Multivalued relationship types
  - First try to remove multivalued relationships on the E-R model level by model transformation.
  - A separate table is created, with foreign keys to all tables that are included in the relationship. Its primary key is composed of all foreign keys.



Translating relationship . . . cont. . .

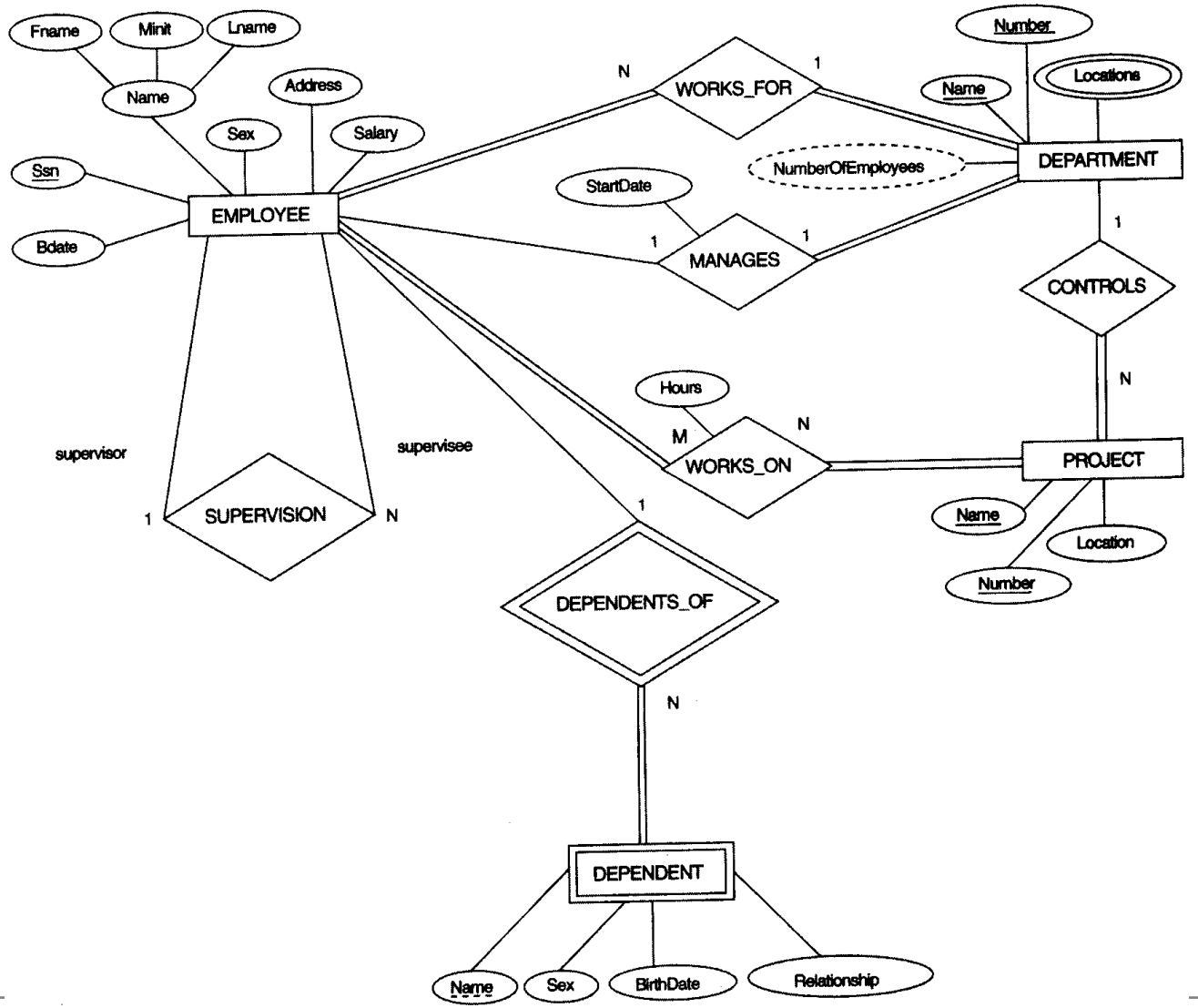
- Step 7: Multivalued relationship types continued
  - In the case where R is 1-N-N, the primary key on R shall not include the fk for the table with cardinality 1.



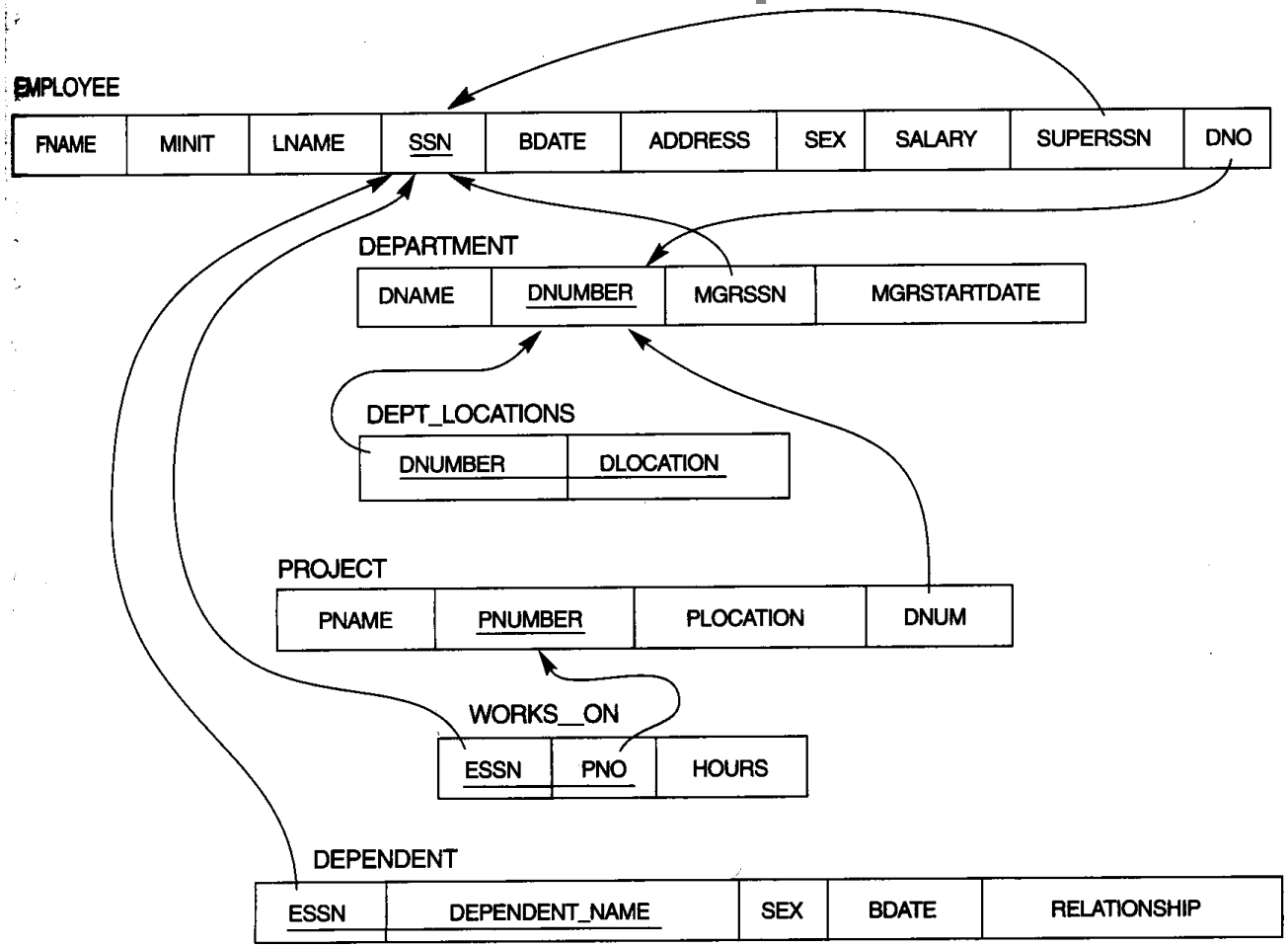
R

f k1	f k2	f k3	a

Example E-R to relational model translation



Relational schemas for the example



Data for the example schema

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	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

DEPT_LOCATIONS

DNUMBER	DLOCATION
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

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

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

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

From E-R to relational model

- The basic procedure defines a set of relational schemas that represent entity and relationship types in the E-R model. This model should further with integrity constraints.
  - Primary keys allow entity types and relationship types to be expressed uniformly as *tables* which represent the contents of the database.
  - A database which conforms to an E-R diagram can be represented by a collection of tables.
  - For each entity type and relationship type there is a unique table which is assigned the name of the corresponding entity type or relationship type.
  - Each table has a number of columns (generally corresponding to attributes), which have unique names.
  - Converting an E-R diagram to a table format is the basis for deriving a relational database design from an E-R diagram.

Summary

- Entity types and their attributes
  - Step 1) Entity types
    - Each entity gets a corresponding table, with the primary key column set to its key attribute.
  - Step 2) Weak entity types
    - The primary key of a weak entity type table has the primary key of the owner table as a component.
- Relationships
  - Step 3) 1-1 Relationship
    - 4 alternatives: fk in E1 or E2, separate R table, common table for E1 & E2
  - Step 4) 1-N Relationship
    - fk i entity on the N-side, separate R table
  - Step 5) M-N Relationship
    - separate R table

Summary cont. . .

- Multivalued attributes and relationships
  - Step 6) Multivalued attributes
    - Separate table for the attribute with its pk composed of the owner pk and the value column.
  - Step 7) Multivalued relationships
    - Separate R table. N-N-N: pk composed of all fk's. 1-N-N: pk is fk to the E1-table.

Short summary E-R -> R

E-R concept	Relational concept
entity type	relation
1:1 relationship type	include one of the primary keys as a foreign key of the other ”entity relation”
1:N relationship type	include the ”1-side” primary key as a foreign key at the ”n-side”
M:N relationship type	relation with two foreign keys
n-ary relationship type (degree > 2)	relation with n foreign keys
simple attribute	attribute
composite attribute	simple attribute components
multivalued attribute	relation anf foreign key
value set	domain
key attribute	primary (or secondary key)