COSC344

Database Theory and Applications

$\sigma_{a='c'}(P)$

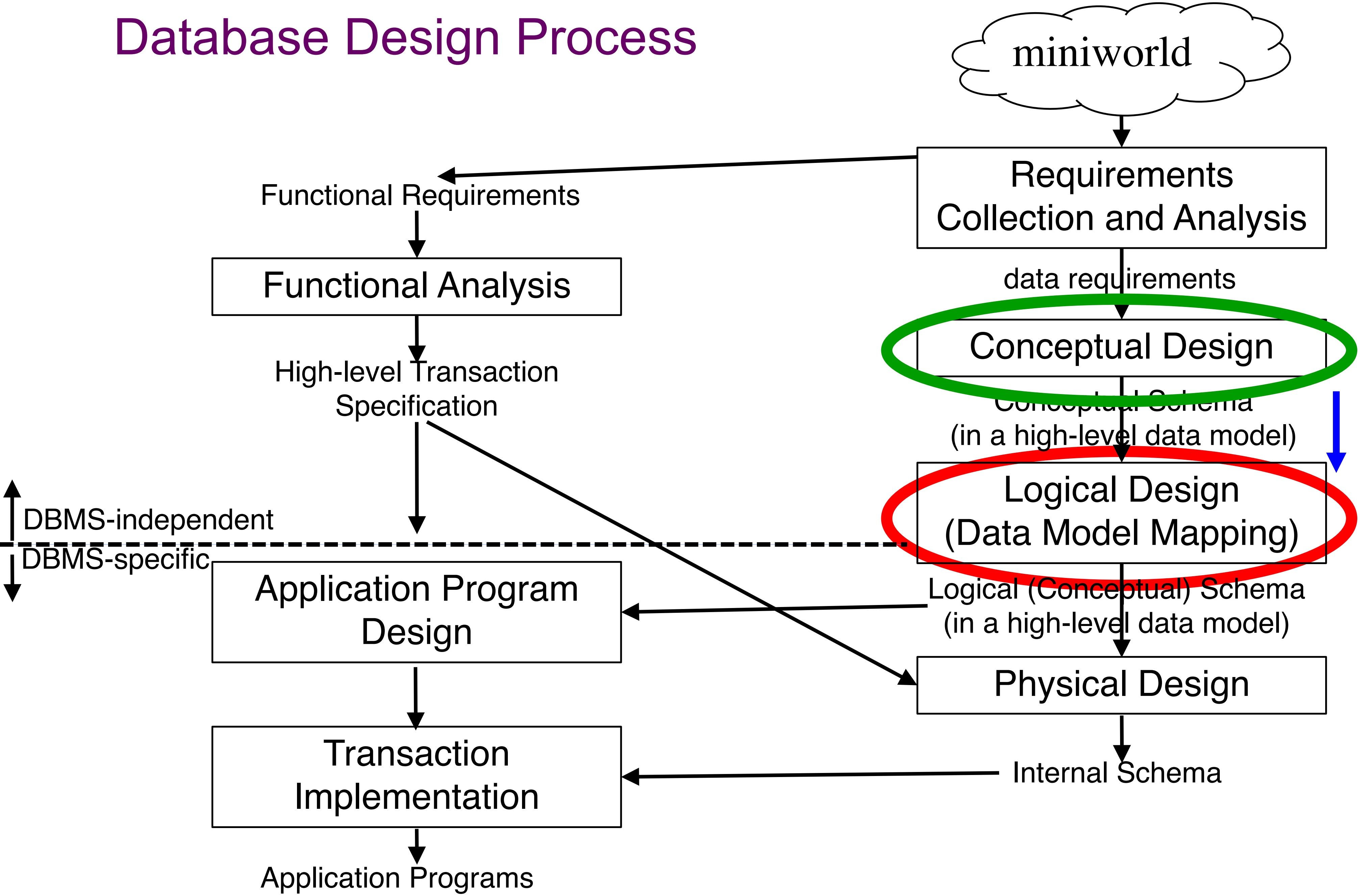
$S \bowtie P$

$\pi_{A, C}(H)$

$P \times Q$

Lecture 4
Relational algebra

Database Design Process



Learning Objectives of Lecture 4

- You should
  - be able to convert an ERD to its relation schema using the seven-step method introduced in this lecture
  - understand the following relational operators
    - Select
    - Project
    - Rename
    - Union
    - Intersection
    - Set difference

- Source
  - Textbook: Chapter 9.1, Chapter 8.1-8.2

ER-to-Relational Mapping

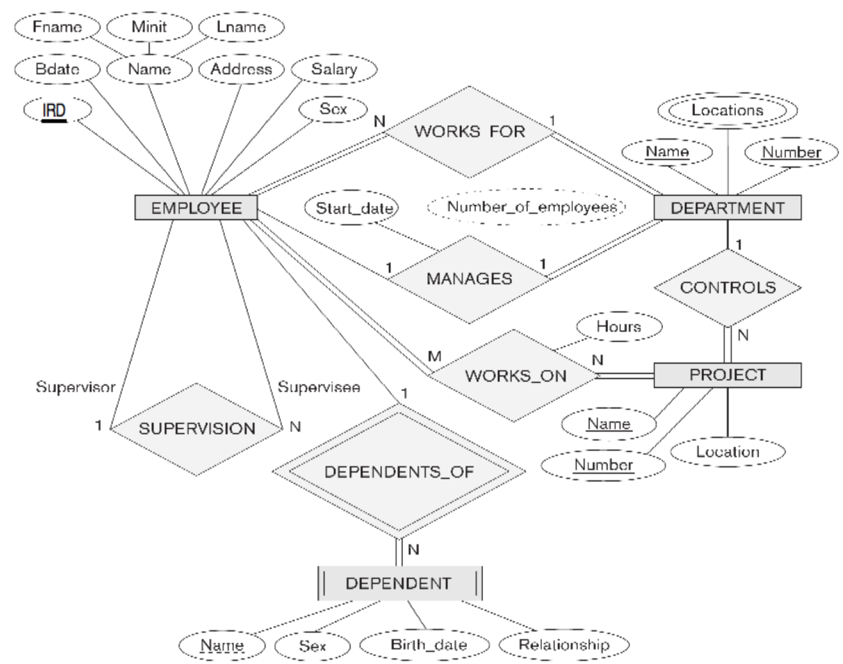


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

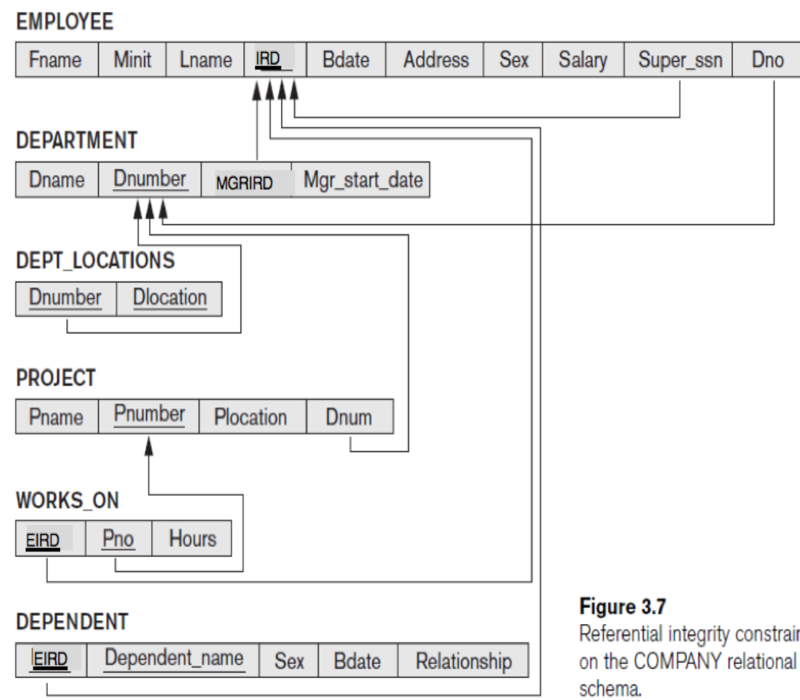


Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

A seven-step algorithm to convert the basic ER diagram into a relation schema

ER-to-Relational Mapping (1)

- Step 1: Mapping of Regular Entity Types
  - For each regular entity type E in the ER schema, **create a relation** R that includes all the **simple** attributes of E.
  - For a **composite attribute**, include its simple component attributes.
  - Choose one of the key attributes of E as the **primary key** for R.
  - If the chosen key was a composite, the set of simple attributes that form it will together be the primary key of R.

EMPLOYEE

Fname	Minit	Lname	Ird	Bdate	Address	Sex	Salary

DEPARTMENT

Dname	Dnumber

PROJECT

Pname	Pnumber	Plocation

ER-to-Relational Mapping (2)

- Step 2: Mapping of Weak Entity Types
  - For each weak entity type W in the ER schema with owner entity type E, **create a relation** R that includes all **simple attributes (or simple components of composites)** of W.
  - Include the primary key attribute of the relation that corresponds to the owner entity type E as **foreign key** attributes of R.
  - The **primary key** of R is the combination of the primary key of the owner and the partial key of the weak entity type, if any.
  - If there is a weak entity type E_2 whose owner is also a weak entity type E_1, then E_1 should be mapped before E_2 to determine its primary key first.

EMPLOYEE

Fname	Minit	Lname	Ird	Bdate	Address	Sex	Salary

DEPARTMENT

Dname	Dnumber

PROJECT

Pname	Pnumber	Plocation

DEPENDENT

Eird	Dependent_name	Bdate	Sex	Relationship

ER-to-Relational Mapping (3)

- Step 3: Mapping of Binary 1:1 Relationship Types
  - Foreign Key Approach (**the most useful approach**)
    - Identify the relations S and T that participate in R
    - Choose one of the relations, say S, and include the primary key of T as a foreign key in S. It is better to choose an entity type with **total participation** in R for the role of S.
    - Include all simple attributes of R as attributes of S
  - Merged relation approach
    - Merge the two entity types and relationship type into one relation
    - Possible when both participations are total
  - Cross-reference approach
    - Set up a third relation to cross-reference primary keys of S and T

EMPLOYEE

Fname	Minit	Lname	Ird	Bdate	Address	Sex	Salary

DEPARTMENT

Dname	Dnumber	Mgrird	Mgr_start_date

PROJECT

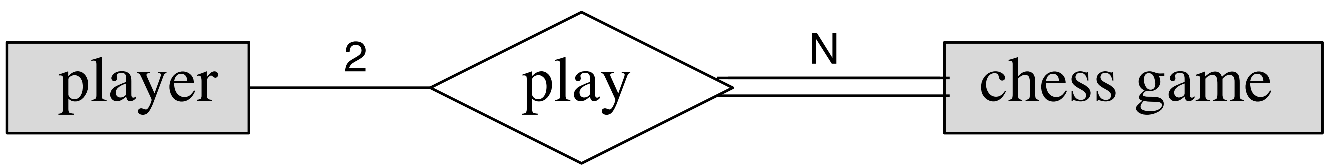
Pname	Pnumber	Plocation

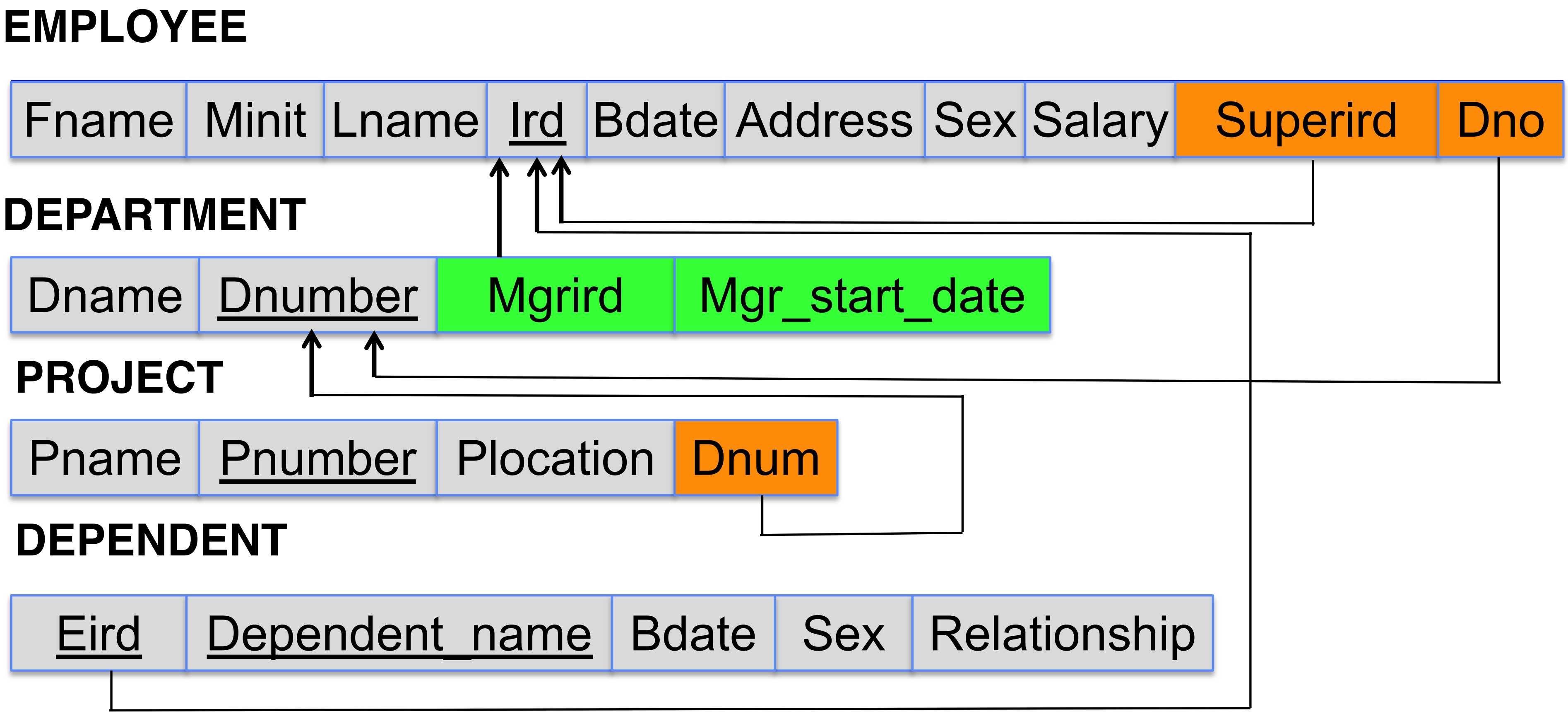
DEPENDENT

Eird	Dependent_name	Bdate	Sex	Relationship

ER-to-Relational Mapping (4)

- Step 4: Mapping of Binary 1:N Relationship Types
  - identify the relation S that represents the entity type participating at the N-side of the relationship.
  - Include as a foreign key in S the primary key of the relation T that represents the other entity type participating in R.
  - Include the simple attributes of R as attributes of S.
- How to map 2:N relationship?

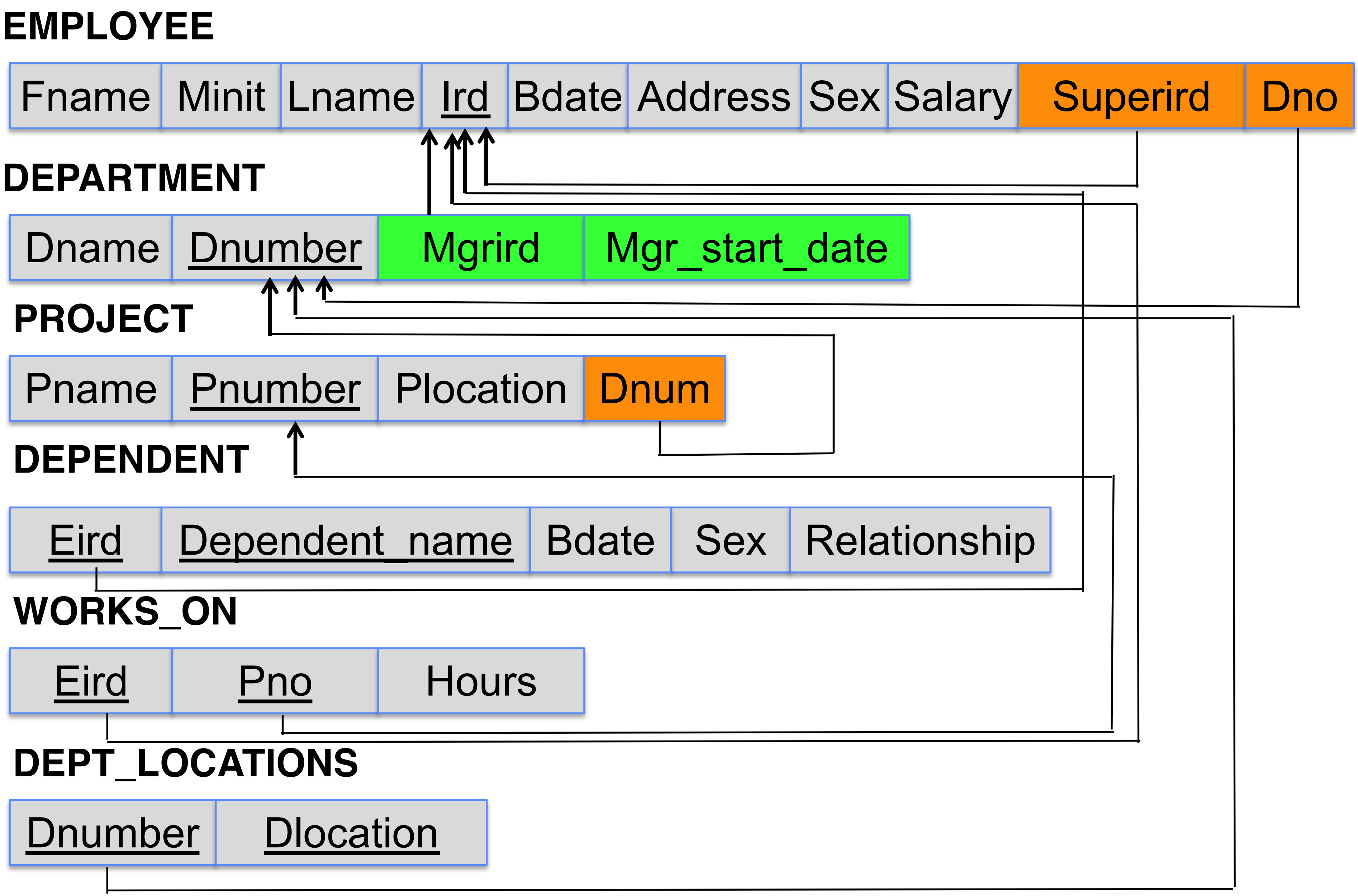



EMPLOYEE
Fname
Minit
Lname
Ird
Bdate
Address
Sex
Salary
Superird
Dno
DEPARTMENT
Dname
Dnumber
Mgrird
Mgr_start_date
PROJECT
Pname
Pnumber
Plocation
Dnum
DEPENDENT
Eird
Dependent_name
Bdate
Sex
Relationship

ER-to-Relational Mapping (5)

- Step 5: Mapping of Binary M:N Relationship Types
  - For each binary M:N relationship type R, create a new relation S to represent R.
  - Include as foreign key attributes in S the primary keys of the participating entity types.
  - Their combination will form the primary key.
  - Include any attributes of R as attributes of S.

- Step 6: Mapping of Multivalued Attributes
  - For each multi-valued attribute A, create a new relation R.
  - R will include an attribute corresponding to A, plus the primary key attribute K of the relation that has A as an attribute.
  - The primary key of R is the combination of A and K.


EMPLOYEE
Fname Minit Lname Ird Bdate Address Sex Salary Superird Dno
DEPARTMENT
Dname Dnumber Mgrird Mgr_start_date
PROJECT
Pname Pnumber Plocation Dnum
DEPENDENT
Eird Dependent_name Bdate Sex Relationship
WORKS_ON
Eird Pno Hours
DEPT_LOCATIONS
Dnumber Dlocation

ER-to-Relational Mapping (6)

- Step 7: Mapping of N-ary Relationship Types
  - For each n-ary relationship type R, where n>2, create a new relation S.
  - Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.
  - Include any attributes of R.
  - The primary key of S is usually a combination of all the foreign keys in S.

SUPPLIER

SNAME	. . .

PROJECT

PROJNAME	. . .

PART

PARTNO	. . .

SUPPLY

SNAME	PROJNAME	PARTNO	QUANTITY

Correspondence between ER and Relational Model

Table 9.1 Correspondence between ER and Relational Models

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

The Relational Data Model

The relational model has 3 core components:

- **Objects (or relations)** – structure of the data organization
- **Integrity** – enforcing constraints and rules
- **Operators** – data manipulation

↓

Relational Algebra

The original relational algebra as described by Codd defines eight operators, in two groups:

special relational operators

- select (or ‘restrict’)
- project
- join
- divide

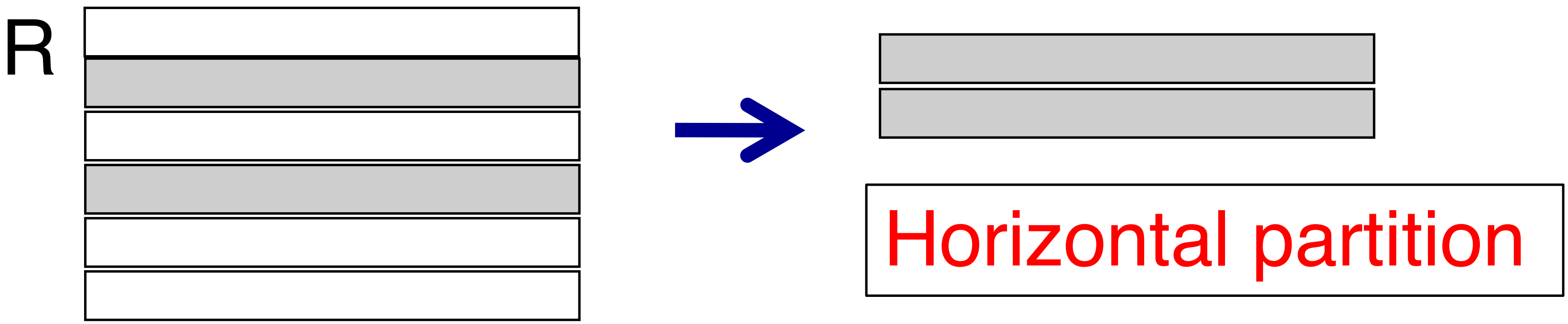
set operators

- union
- intersection
- difference (‘minus’)
- Cartesian product (‘times’)

SELECT (RESTRICT)

$\sigma_{\text{<selection\_condition>}}(\mathbf{R})$

Works on a single relation R and select a ***subset*** of the tuples of R that satisfy the specified ***selection_condition***.



- Selection condition clauses are of the form
  - <attr name> <comparison op> <constant>
  - <attr name> <comparison op> <attr name>
- Comparison operators: <, ≤, >, ≥, =, ≠
- <constant> must be from the attribute domain
- Clauses can be combined with AND, OR and NOT

SELECT (RESTRICT)

Example: **List all employees with a salary no less than 40,000.**

$\sigma_{salary \geq 40000}$ **(EMPLOYEE)**

FNAME	MINIT	LNAME	IRD	BDATE	ADDRESS	SEX	SALARY	SUPERIRD	DNO
Franklin	T	Wong	333445555	8-Dec-1955	638 Voss, Houston, TX	M	40000	888665555	5
Jennifer	S	Wallace	987654321	20-Jun-1941	291 Berry, Bellaire, TX	F	43000	888665555	4

- $\sigma_{DNO = 4}$(EMPLOYEE)
- $\sigma_{IRD \neq '123456789'}$(EMPLOYEE)
- $\sigma_{(DNO=4 \text{ AND } SALARY>25000) \text{ OR } (DNO=5 \text{ AND } SALARY>30000)}$(EMPLOYEE)

SELECT (continued)

- The SELECT operation is applied to each tuple individually
- The degree of the resulting relation is the same as the original relation
- Number of tuples in the resulting relation is always less than or equal to the number of tuples in the original relation
- SELECT is commutative

$$\sigma_{<cond1>}(\sigma_{<cond2>}(R)) = \sigma_{<cond2>}(\sigma_{<cond1>}(R))$$

- A cascade of SELECTs can be combined into a single SELECT

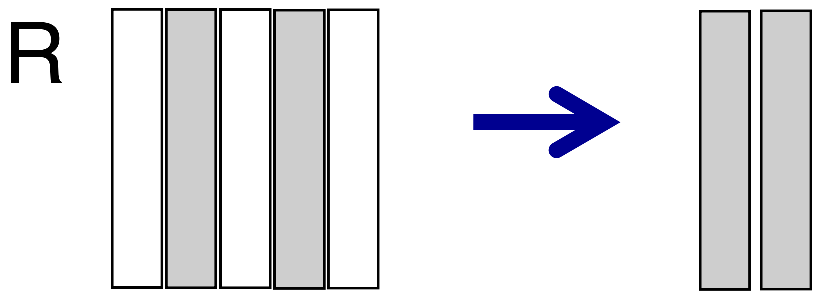
$$\sigma_{<cond1>}(\sigma_{<cond2>}(\ldots(\sigma_{<condn>}(R))$$

$$= \sigma_{<cond1>\ \mathrm{AND}\ <cond2>\ \mathrm{AND}\ \ldots\ \mathrm{AND}\ <condn>}(R)$$

PROJECT

$\Pi_{<attribute\_list>}(\mathbf{R})$

Works on a single relation R and defines a relation that contains **a vertical subset** of R, extracting the values of specified attributes and eliminating duplicates.



Vertical partition

- Order of attributes is the same order as they appear in the <attribute_list>.
- Duplicates are removed.
- $\Pi_{<list1>}(\Pi_{<list2>}(R)) = \Pi_{<list1>}(R)$
  - As long as <list2> contains the attributes in <list1>
- Commutative property does not hold for PROJECT.

Try It

Example: **List the name and number for each department**

$\Pi_{DNAME,\ DNO}$ **(DEPARTMENT)**

DNAME	DNO
Research	5
Administration	4
Headquarters	1
Dummies	0

- $\Pi_{LNAME,\ FNAME,\ SALARY}(EMPLOYEE)$
- $\Pi_{SEX,\ SALARY}(EMPLOYEE)$

Sequences of Operations

- Operations can be
  - Nested as a single relational algebra expression
  - Applied one (or a few) at a time, creating intermediate results
- Example
  - $\Pi_{\text{FNAME, LNAME, SALARY}}(\sigma_{\text{DNO=5}}(\text{EMPLOYEE}))$

Break down a complex sequence of operations into multiple single relational algebra expressions.

Or

- $\text{DEP5\_EMPS} \leftarrow \sigma_{\text{DNO=5}}(\text{EMPLOYEE})$
- $\text{RESULT} \leftarrow \Pi_{\text{FNAME, LNAME, SALARY}}(\text{DEP5\_EMPS})$

- What is the result?

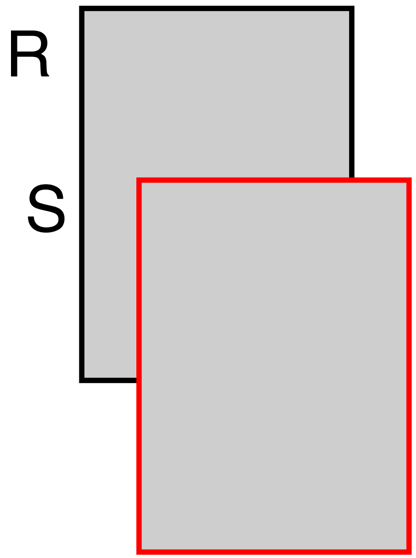
RENAME Operation

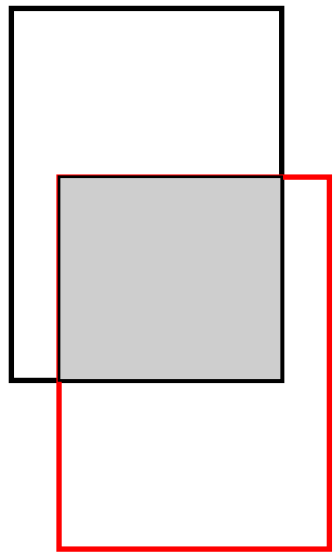
- A formal RENAME operation can rename either the relation name or the attribute names, or both as a unary operator
  - $\rho_S(R)$: rename the relation R to S
  - $\rho_{(B1,\ B2,\ \ldots,\ Bn)}(R)$: rename the attributes of R
  - $\rho_{S(B1,\ B2,\ \ldots,\ Bn)}(R)$: rename both the relation and its attributes

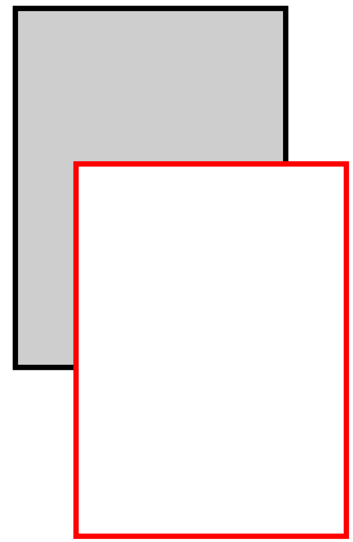
  Where S is the new relation name, and B1, B2, . . ., Bn are the new attribute names.

The set operators

Union	Intersection	Difference
R ∪ S	R ∩ S	R - S







Set Operations

- UNION
  - R ∪ S
  - Includes all tuples that are either in R or in S or in both
  - Duplicate tuples are eliminated
- INTERSECTION
  - R ∩ S
  - Includes every tuple that is simultaneously in both R and S
- SET DIFFERENCE
  - R - S
  - Includes all tuples that are in R but not is S
- UNION and INTERSECTION are associative
  - R ∪ (S ∪ T) = (R ∪ S) ∪ T
  - R ∩ (S ∩ T) = (R ∩ S) ∩ T
- SET DIFFERENCE is not commutative
  - R - S != S – R
- Must be union compatible or type compatible
  - Have the same degree n, and dom(A_i) = dom(B_i) for all i

Examples

(a)

STUDENT	FN	LN
	Susan	Yao
	Ramesh	Shah
	Johnny	Kohler
	Barbara	Jones
	Amy	Ford
	Jimmy	Wang
	Ernest	Gilbert

INSTRUCTOR	FNAME	LNAME
	John	Smith
	Ricardo	Browne
	Susan	Yao
	Francis	Johnson
	Ramesh	Shah

(b)

FN	LN
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert
John	Smith
Ricardo	Browne
Francis	Johnson

(c)

FN	LN
Susan	Yao
Ramesh	Shah

(d)

FN	LN
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

(e)

FNAME	LNAME
John	Smith
Ricardo	Browne
Francis	Johnson

Figure 7.11 Illustrating the set operations UNION, INTERSECTION, and DIFFERENCE. (a) Two union compatible relations. (b) STUDENT ∪ INSTRUCTOR. (c) STUDENT ∩ INSTRUCTOR. (d) STUDENT − INSTRUCTOR. (e) INSTRUCTOR − STUDENT.

Question to Ponder

- How do we tell the DBMS about our database?