CSL 451 Introduction to Database Systems

SQL (3.5-3.7)

Department of Computer Science and Engineering
Indian Institute of Technology Ropar

Narayanan (CK) Chatapuram Krishnan

Summary

- Set Operations
  - ‘union’
  - ‘intersect’
  - ‘except’
  - In the presence of duplicates
    - ‘all’
- Null Values
  - Arithmetic expressions involving null
  - Comparisons involving null
    - ‘unknown’ value
  - ‘is null’
  - ‘is not null’

Summary

- Aggregate Functions
  - ‘avg’
  - ‘min’, ‘max’
  - ‘sum’
  - ‘count’
- Handling of duplicates
- Aggregation with Grouping
  - ‘group by’
    - Relation between attributes appearing in ‘select’ clause and ‘group by’ clause
  - ‘having’
    - Relation between attributes appearing in ‘having’ clause and ‘group by’ clause
- Order of execution with ‘group by’ and ‘having’ clauses
- Aggregation with null and Boolean values

3.8.a Find all customers of the bank who have an account but not a loan

branch (branch_name, branch_city, assets)

customer (customer_name, customer_street, customer_city)

loan(loan_number, branch_name, amount)

borrower (customer_name, loan_number)

account (account_number, branch_name, balance)

depositor (customer_name, account_number)

3.11.b Find the IDs and names of all students who have not taken any course offering before Spring 2009

3.11.c For each department, find the maximum salary of instructors in that department. You may assume that every department has at least one instructor

EN Q18 List the name of all employees who do not have supervisors

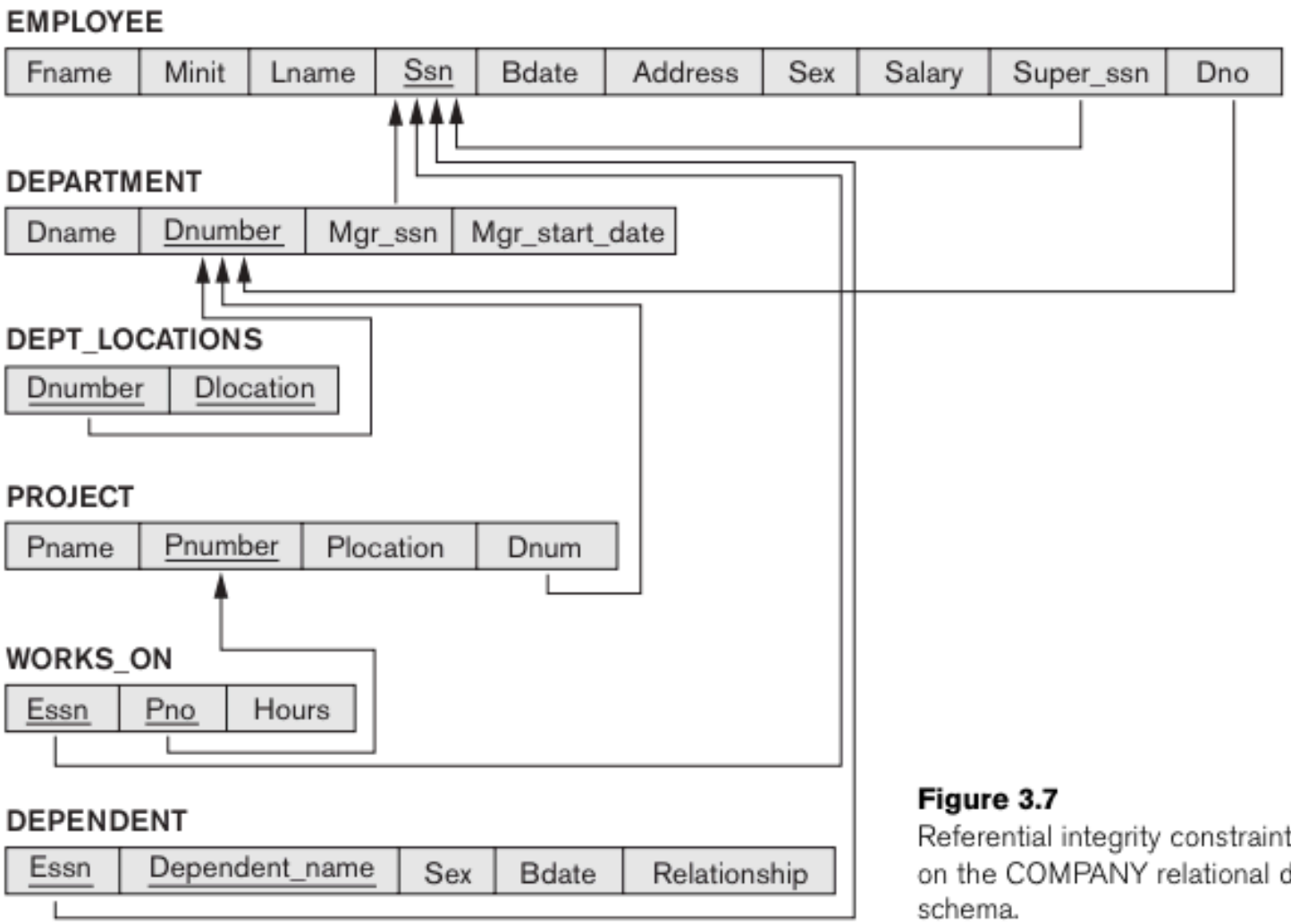

Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

EN Q4A Make a list of all project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department controls the project.

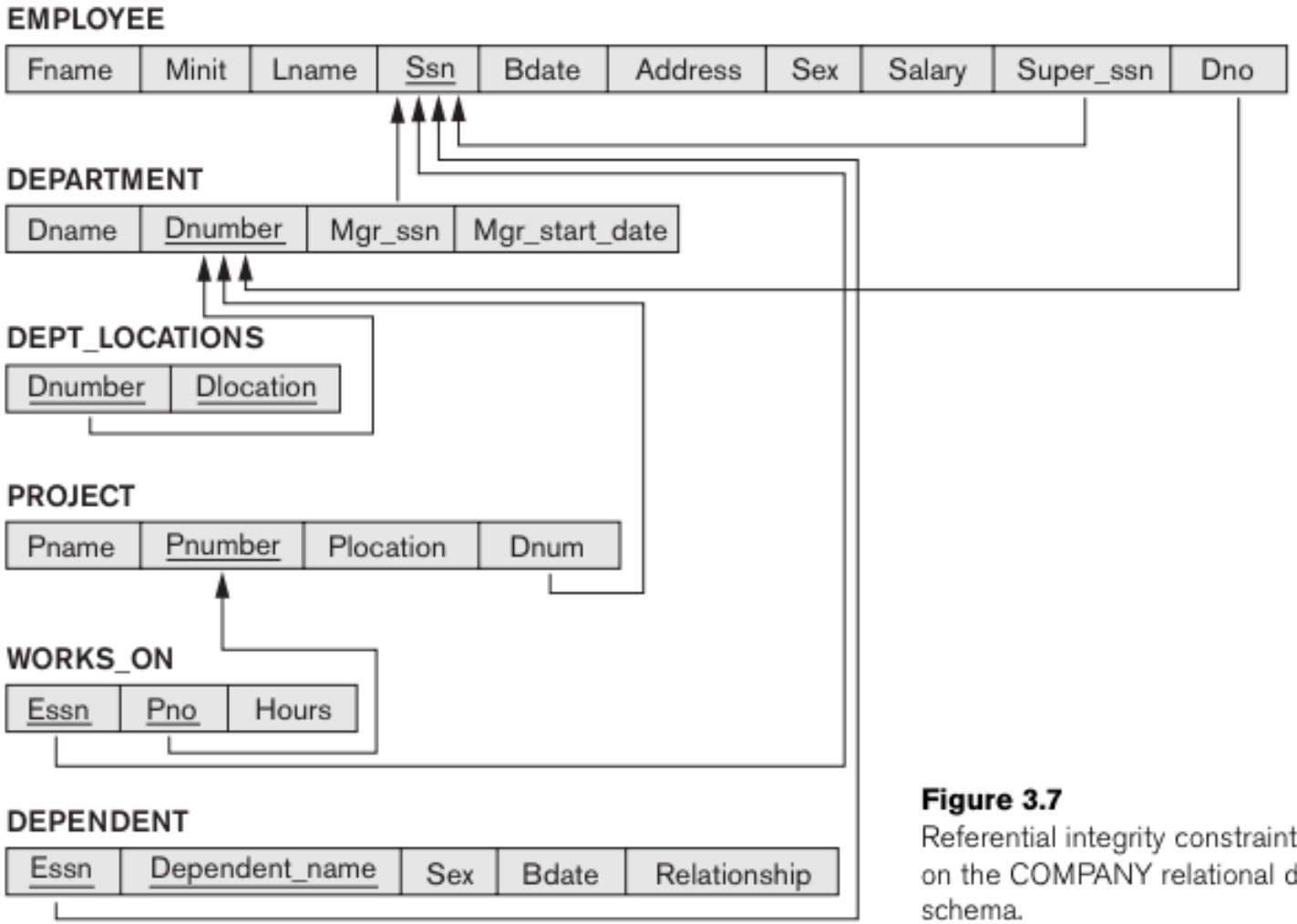

Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

3.14.a Find the number of accidents in which the cars belonging to John Smith were involved

person (*driver id*, *name*, *address*)

car (*license*, *model*, *year*)

accident (*report number*, *date*, *location*)

owns (*driver id*, *license*)

participated (*report number*, *license*, *driver id*, *damage_amount*)

SQL – Nested Queries (Membership)

- subquery
  - select, from or where clause
- Set Membership
  - 'in' - presence
  - 'not in' - absence

select a
from *r1*
where *p* **and** *b* **in** (**select** *c* **from** *r1* **where** *p*)

- Find all the courses taught in both the Fall 2009 and Spring 2010 semesters

(**select** *course_id*
from *section*
where *semester*='Fall' **and** *year*=2009)
intersect
(**select** *course_id*
from *section*
where *semester*='Spring' **and** *year*=2010)

SQL- Nested Queries (Membership)

- Find all the courses taught in Fall 2009 but not in Spring 2010

(**select** *course_id*
from *section*
where *semester*='Fall' **and** *year*=2009)
except
(**select** *course_id*
from *section*
where *semester*='Spring' **and** *year*=2010)

EN Q4A Make a list of all project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department controls the project.

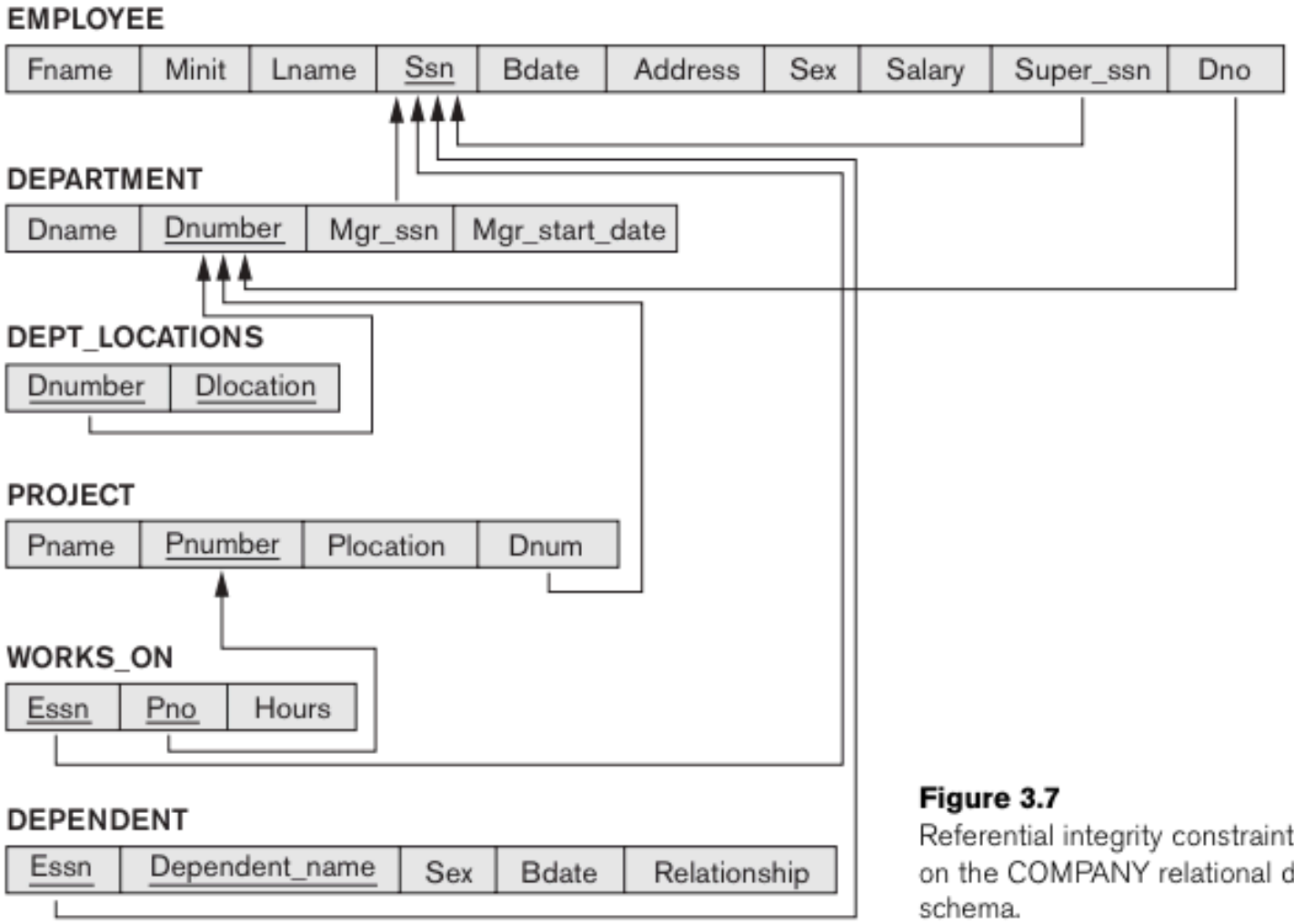

Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

EN Q16 Retrieve the name of each employee who has a dependent with the same first name and is the same sex as the employee

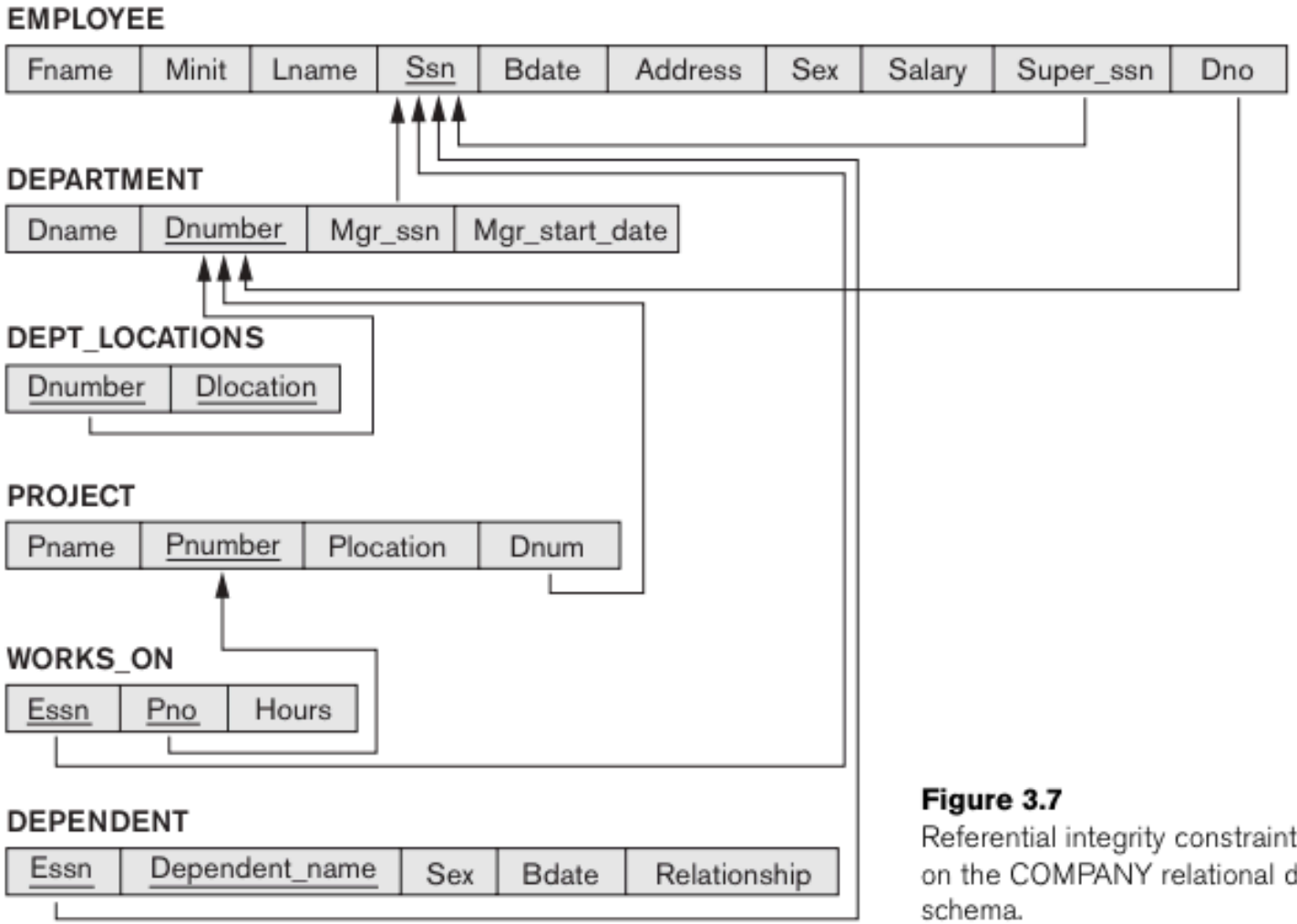

Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

SQL-Nested Queries (Comparison)

- 'some'

  select *a*
  from *r1*
  where *p* **and** *b* **op some** (**select** *c* **from** *r1* **where** *p*)

  – op can be <, >, <=, >=, =,< >
- Find the names of all instructors whose salary is greater than at least one instructor in the Biology department

```
select distinct T.name
from instructor as T, instructor as S
where T.salary > S.salary and S.dept_name='Biology'
```

- Find all the courses taught in both the Fall 2009 and Spring 2010 semesters

```
select distinct course_id
from section
where semester='Fall' and year=2009 and
      course_id in (select course_id
                    from section
                    where semester='Spring' and year=2010)
```

= some ?

- Find all the courses taught in Fall 2009 but not in Spring 2010 semesters

```
select distinct course_id
from section
where semester='Fall' and year=2009 and
      course_id not in (select course_id
                        from section
                        where semester=Spring' and year=2010)
```

<>some ?

SQL-Nested Queries (Comparison)

- 'all'

  select *a*
  from *r1*
  where *p* **and** *b* **op all(select** *c* **from** *r1* **where** *p*)

  – op can be <, >, <=, >=, =,< >

- Find the names of all instructors whose salary is greater than every instructor in the Biology department

select name
from instructor
where salary **> all** (
select salary
from instructor
where dept_name='Biology')

- Find all the courses taught in both the Fall 2009 and Spring 2010 semesters

```
select distinct course_id
from section
where semester='Fall' and year=2009 and
      course_id in (select course_id
                    from section
                    where semester='Spring' and year=2010)
```

= all ?

- Find all the courses taught in Fall 2009 but not in Spring 2010 semesters

```
select distinct course_id
from section
where semester='Fall' and year=2009 and
      course_id not in (select course_id
                        from section
                        where semester=Spring' and year=2010)
```

<> all?

- Find the departments that have the highest average salary?

3.9.d Find all employees in the database who earn more than each employee of 'Small Bank Corporation'

employee (employee_name, street, city)

works (employee_name, company_name, salary)

company (company_name, city)

manages (employee_name, manager_name)

SQL-Nested Queries (Test for Empty Relations)

- 'exists'
  - returns true if the argument subquery is nonempty

Find all the courses taught in both the Fall 2009 and Spring 2010 semesters

select distinct S.course_id
from section **as** S, section **as** T
where S.semester='Fall' **and** S.year=2009 **and** T.semester='Spring' **and** T.year=2010 **and** S.course_id=T.course_id)

SQL-Nested Queries (Test for Empty Relations)

- 'exists'
  - returns true if the argument subquery is nonempty

Find all the courses taught in both the Fall 2009 and Spring 2010 semesters

select distinct S.course_id
from section **as** S, section **as** T
where S.semester='Fall' **and** S.year=2009 **and** T.semester='Spring' **and** T.year=2010 **and** S.course_id=T.course_id)

```
select distinct S.course_id
from section as S
where S.semester='Fall' and S.year=2009 and
        exists (select *
                from section as T
                where T.semester='Spring' and T.year=2010 and
S.course_id=T.course_id);
```

correlated subquery

EN Q7 List the name of managers who have at least one dependent

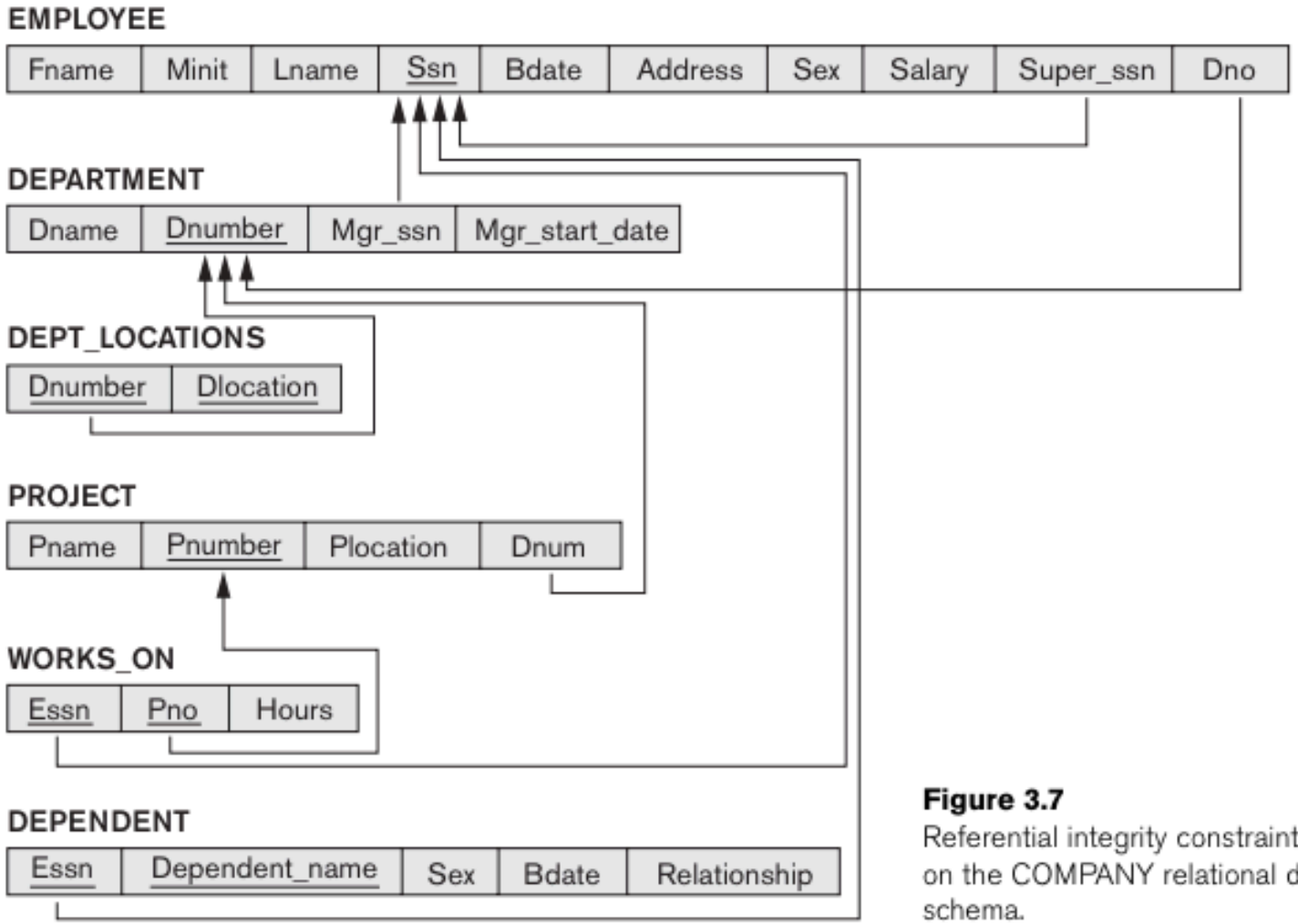

Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

SQL-Nested Queries (Test for Empty Relations)

- 'not exists'
  - returns true if the argument subquery is empty

Find all the students who have taken all courses offered in the Biology department

SQL-Nested Queries (Test for Empty Relations)

- 'not exists'
  - returns true if the argument subquery is empty

Find all the students who have taken all courses offered in the Biology department

```
select T.ID, T.name
from student as T
where not exists ((select course_id
    from course
    where dept_name='Biology')
    except
    (select S. course_id
     from takes natural join course as S where S.ID=T.ID))
```

- Find all the students who have taken all courses offered in the Biology department and these are the only courses that they have taken.

- Find all the students who have taken all courses offered in the Biology department and these are the only courses that they have taken.

```
select T.ID, T.name
from student as T
where (not exists ((select course_id
        from course
        where dept_name='Biology')
        except
        (select S. course_id
        from takes natural join course as S where S.ID=T.ID))
    and
    not exists (select S.dept_name
        from course natural joint takes as S
        where S.ID = T.ID and S.dept_name<>'Biology'))
```

3.9.e Assume that the companies may be located in several cities. Find all companies located in every city in which 'Small Bank Corporation' is located

employee (employee_name, street, city)

works (employee_name, company_name, salary)

company (company_name, city)

manages (employee_name, manager_name)

3.2.a Find the total grade points earned by the student with ID 12345, across all courses taken by the student

grade.points(grade, points)

3.2.b Find the grade point average earned by the student with ID 12345, across all courses taken by the student

grade.points(grade, points)

3.2.c Find the ID and GPA of every student

grade.points(grade, points)

3.9.f Find the companies that has the most employees

employee (employee_name, street, city)

works (employee_name, company_name, salary)

company (company_name, city)

manages (employee_name, manager_name)

3.1.d Find all instructors earning the highest salary

3.1.e Find the enrollment of each section that was offered in Autumn 2009

3.1.e Find the maximum enrollment, across all sections, in Autumn 2009.