CHAPTER 2

——EXERCISES

1

Describe the differences in meaning between the terms relation and relation schema.

- Answer:

A relation schema is a type definition, and a relation is an instance of that schema.

For example, student (ss#, name) is a relation schema and

ss#	name
123-45-6789	Tom Jones
456-78-9123	Joe Brown

is a relation based on that schema.

2

employee(person_name, street, city)

works(person_name, company_name, salary)

company(company_name, city)

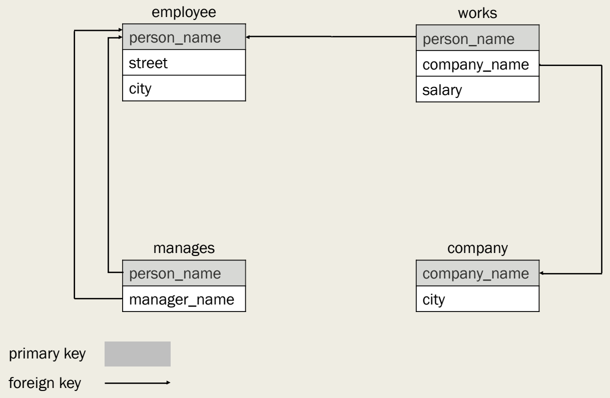
manages(person_name, manager_name)

3

Which attributes should be choose as primary key for each relation schema?

Which attributes are foreign keys based on the primary keys?

4



5

- employee(person_name, street, city)
- works(person_name, company_name, salary)
- company(company_name, city)
- manages(person_name, manager_name)

6

【T0001】*Consider the relational database of the figure. Give an expression in the relational algebra to express each of the following queries:*

1. Find the names of all employees who work for First Bank Corporation.
2. Find the names and cities of residence of all employees who work for First Bank Corporation.
3. Find the names, street address, and cities of residence of all employees who work for First Bank Corporation and earn more than $10,000 per annum.
4. Find the names of all employees in this database who live in the same city as the company for which they work.
5. Find all companies located in the city in which Small Bank Corporation is located.

7

1. Find the names of all employees who work for First Bank Corporation.

$$\Pi_{person\_name}(\sigma_{company\_name=\text{“Fist Bank Corporation”}}(works))$$

2. Find the names and cities of residence of all employees who work for First Bank Corporation.

$$\Pi_{employee.person\_name,\ city}(\sigma_{employee.person\_name=workspersonl\_name}(employee\times(\sigma_{company\_name=\text{“Fist Bank Corpoation”}}(works))))$$

8

3. Find the names, street address, and cities of residence of all employees who work for First Bank Corporation and earn more than $10,000 per annum.

$\Pi_{person\_name,\, street,city}(\sigma_{employee.person\_name=works.person\_name}$
$(\sigma_{(company\_name=\text{“Fist Bank Coration”} \wedge salry>10000)}(works) \times employee\ ))$

4. Find the names of all employees in this database who live in the same city as the company for which they work.

$\Pi_{employee.person\_name}(\sigma_{company.\text{city}=\text{employee.}city}(company \times$
$\sigma_{employee.person\_name=works.person\_name}(employee \times works)))$

9

5. Find all companies located in the city in which Small Bank Corporation is located.

$\Pi_{comapny.company\_name}\ (\sigma_{company.city=SBC.city}(company \times$
$\rho_{SBC}(\Pi_{city}(\sigma_{company\_name=\text{“Small Bank Corporation”}}(company))))$

10

student(ID, name, dept_name, credits)

instructor(ID, name, dept_name, salary)

course(course_id, title, dept_name, credits)

section(course_id, sec_id, semester, year, building, room_number, time_slot_id)

takes(ID, course_id, sec_id, semester, year, grade)

teaches(ID, course_id, sec_id, semester, year)

advisor(s_ID, i_ID)

11

【T0005】Consider the relation *advisor* in the figure, *s_id* is the primary key of *advisor*. Suppose one student could be advised by multiple instructors. Therefore, is *s_id* still the primary key of *advisor*? If not, which is the new primary key of *advisor*?

12

Consider the relation *advisor* in the figure, *s_id* is the primary key of *advisor*. Suppose one student could be advised by multiple instructors. Therefore, is *s_id* still the primary key of *advisor*? If not, which is the new primary key of *advisor*?

s_id is a foreign key in *advisor* to *student*. As one student could be advised by multiple instructors, only use s_id could not distinct different tuples. In addtion, one instructor could advise multiple students. So the primary key of *advisor* is {s_id, i_id}.

13

student(ID, name, dept_name, credits)

instructor(ID, name, dept_name, salary)

course(course_id, title, dept_name, credits)

section(course_id, sec_id, semester, year, building, room_number, time_slot_id)

takes(ID, course_id, sec_id, semester, year, grade)

teaches(ID, course_id, sec_id, semester, year)

advisor(s_ID, i_ID)

14

【T0004】Explain the meaning of following expressions:

- $\sigma_{year \geq 2009}(takes) \bowtie \text{student}$
- $\sigma_{year \geq 2009}(takes \bowtie \text{student})$
- $\Pi_{ID,name,course\_id}(student \bowtie takes)$

15

- $\sigma_{year \geq 2009}(takes) \bowtie \text{student}$

For each student who takes at least one course in 2009, display the students information along with the information about what courses the student took. The attributes in the result are:
ID, name, dept name, tot cred, course id, section id, semester, year, grade

- $\sigma_{year \geq 2009}(takes \bowtie \text{student})$

Same as above; selection can be done before the join operation.

- $\Pi_{ID,name,course\_id}(student \bowtie takes)$

Provide a list of consisting of
ID, name, course id
of all students who took any course in the university.

16

loan

loan_number	branch_name	amount
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

borrower

customer_name	loan_number
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

customer_name	customer_street	customer_city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

customer

17

【T0007】Consider the relation in the figure, give an expression in the relational algebra for the query “Find the names of all customers who have a loan at the bank.” Rewrite the query to include not only the name, but also the city of residence for each customer. Observe that now customer Jackson no longer appears in the result, even though Jackson does in fact have a loan from the bank.

1. Explain why Jackson does not appear in the result.
2. Suppose that you want Jackson to appear in the result. How would you modify the database to achieve this effect?

18

- Find the names of all customers who have a loan at the bank

$$\Pi_{customer\_name}(\text{borrower})$$

- Find the names of all customers who have a loan at the bank and their city

$$\Pi_{customer\_name,customer\_city}(\text{customer} \bowtie \text{borrower} \bowtie \text{loan})$$

19

1. Explain why Jackson does not appear in the result.

Answer: Although Jackson does have a loan, no address is given for Jackson in the customer relation. Since no tuple in customer joins with the Jackson tuple of borrower, Jackson does not appear in the result.

2. Suppose that you want Jackson to appear in the result. How would you modify the database to achieve this effect?

Answer: The best solution is to insert Jackson’s address into the customer relation. If the address is unknown, null values may be used. If the database system does not support nulls, a special value may be used (such as unknown) for Jackson’s street and city. The special value chosen must not be a plausible name for an actual city or street.

20

employee(person_name, street, city)

works(person_name, company_name, salary)

company(company_name, city)

manages(person_name, manager_name)

21

4. Find the names of all employees in this database who live in the same city as the company for which they work.

$$\Pi_{peperson\_name}(\sigma_{company.city=employee.city}(company \times \sigma_{employee.person\_name=works.person\_name}(employee \times works)))$$

$$\Pi_{peperson\_name}(company \bowtie employee \bowtie works)$$

22

【T0006】 Consider the relational database in the figure. Give an expression in the relational algebra for each request:

1. Assume the companies may be located in several cities. Find all companies located in cities in which Small Bank Corporation is located.
2. Give all employees of First Bank Corporation a 10 percent salary raise.
3. Give all managers in this database a 10% salary raise, unless the salary would be greater than $100,000. In such cases, give only a 3 percent raise.

23

1. Assume the companies may be located in several cities. Find all companies located in cities in which Small Bank Corporation is located.

$$\Pi_{comapny.company\_name}\ (company \div \Pi_{city}(\sigma_{company\_name=\text{“Small Bank Corporation”}}(company))$$

2. Give all employees of First Bank Corporation a 10 percent salary raise.

$$works \leftarrow \Pi_{person\_name,comany\_name,salary*1.1}(\sigma_{company\_name=\text{“First Bank Corporation”}}(works) \cup (works - \sigma_{company\_name=\text{“First Bank Corporation”}}(works))$$

24

3. Give all managers in this database a 10% salary raise, unless the salary would be greater than $100,000. In such cases, give only a 3 percent raise.

$t_1 \leftarrow \Pi_{works.person\_name,company.company\_name,salry}$
$(\sigma_{works.person\_name=manger}(works \times managers))$
$t_2 \leftarrow \Pi_{works.person\_name,company.company\_name,salry*1.03}(\sigma_{t_1.salry*1.1>10000}(t_1)$
$t_2 \leftarrow t_2 \cup \Pi_{works.person\_name,company.company\_name,salry*1.1}(\sigma_{t_1.salry*1.1\leq 10000}(t_1)$
$works \leftarrow (works - t_1) \cup t_2$

25

employee(person_name, street, city)

works(person_name, company_name, salary)

company(company_name, city)

manages(person_name, manager_name)

26

【T0048】Consider the relational database in the figure. Give an expression in the relational algebra for each request:

1. Find the company with the most employees.
2. Find the company with the smallest payroll.
3. Find those companies whose employees earn a higher salary, on average, than the average salary at First Bank Corporation.
4. Delete all tuples in the *works* relation for employees of Small Bank Corporation.

27

1. Find the company with the most employees.

$\mathrm{t}_1 \leftarrow company\_name\mathcal{G}_{\boldsymbol{count-distinct}(person\_name)\ as\ number}\ (works)$
$result \leftarrow \Pi_{company\_name}(company\_name\ \mathcal{G}_{\boldsymbol{Max}(number)}(t_1))$

2. Find the company with the smallest payroll.

$\mathrm{t}_1 \leftarrow company\_name\mathcal{G}_{\boldsymbol{count}(salary)\ as\ payroll}\ (works)$
$result \leftarrow \Pi_{company\_name}(company\_name\ \mathcal{G}_{\boldsymbol{Min}(number)}(t_1))$

3. Find those companies whose employees earn a higher salary, on average, than the average salary at First Bank Corporation.

$\mathrm{t}_1 \leftarrow company\_name\ \mathcal{G}_{\boldsymbol{avg}(salary)\ as\ payroll}\ (works)$
$FBC \leftarrow \sigma_{company_{name}=\text{“FisrtBank Corporation”}}\ (\mathrm{t}_1)$
$result \leftarrow \Pi_{company\_name}(\sigma_{t_1.payroll>FBC.payroll}\ (t_1 \times \mathrm{FBC}))$

28

4. Delete all tuples in the *works* relation for employees of Small Bank Corporation.

$$works \leftarrow works - \sigma_{company_{name}=\text{“Small Bank Corporation”}}(works)$$

29

student(ID, name, dept_name, credits)

instructor(ID, name, dept_name, salary)

course(course_id, title, dept_name, credits)

section(course_id, sec_id, semester, year, building, room_number, time_slot_id)

takes(ID, course_id, sec_id, semester, year, grade)

teaches(ID, course_id, sec_id, semester, year)

advisor(s_ID, i_ID)

30

- Consider the relational database of the figure. Write the following queries in relational algebra, using the university schema.

  1. *Find the titles of courses in the Comp. Sci. department that have 3 credits.*
  2. *Find the IDs of all students who were taught by an instructor named Einstein.*
  3. *Find the highest salary of any instructor.*
  4. *Find all instructors earning the highest salary (there may be more than one with the same salary).*
  5. *Find the enrollment of each section that was offered in Autumn 2009.*
  6. *Find the maximum enrollment, across all sections, in Autumn 2009.*
  7. *Find the sections that had the maximum enrollment in Autumn 2009*

31

CHAPTER 3

——EXERCISES

1

employee(<u>person_name</u>, street, city)

works(<u>person_name</u>, company_name, salary)

company(<u>company_name</u>, city)

manages(<u>person_name</u>, manager_name)

2

【0008】Give an SQL schema definition for the employee database of figure. Choose an appropriate domain for each attribute and an appropriate primary key for each relation schema.

3

- **create table** *employee*(
  employee_name **char**(20),
  street **char**(30),
  city **char**(30),
  primary key (*employee_name*));
- **create table** *works*(
  employee_name **char**(20),
  company_name **char**(20),
  salary **numeric**(8,2),
  primary key (*employee_name*));
- **create table** *company*(
  company_name **char**(20),
  city **char**(30),
  primary key (*company_name*));
- **create table** *manages*(
  employee_name **char**(20),
  manager_name **char**(20),
  primary key (*employee_name*));

4

【0009】Let the following relation schemas be given:

R=(A, B, C)
S=(D,E,F)

Let relations *r*(*R*) and *s*(S) be given. Give an expression in SQL that is equivalent to each of the following queries.

a. $\Pi_A(r)$
b. $\sigma_{B=17}(r)$
c. $r \times s$
d. $\Pi_{A,F}(\sigma_{C=D}(r \times s))$

5

R=(A, B, C)
S=(D,E,F)

a. $\Pi_A(r)$

```
select distinct A
from r
```

b. $\sigma_{B=17}(r)$

```
select distinct *
from r
where B=17
```

c. $r \times s$

```
select distinct *
from r, s
```

d. $\Pi_{A,F}(\sigma_{C=D}(r \times s))$

```
select distinct A, distinct F
from r, s
where C=D
```

select $\equiv \boldsymbol{\Pi}$
from $\equiv \times$
where $\equiv \boldsymbol{\sigma}$

6

employee(<u>person_name</u>, street, city)

works(<u>person_name</u>, company_name, salary)

company(<u>company_name</u>, city)

manages(<u>person_name</u>, manager_name)

7

【0010】*Consider the employee database in the figure, where the primary keys are underlined. Give an expression in SQL for each of the following queries.*

1. Find the names of all employees who work for First Bank Corporation.
2. Find all employees in the database who live in the same cities as the companies for which they work.
3. Find all employees in the database who live in the same cities and on the same streets as do their managers.

8

employee(person_name, street, city)
works(person_name, company_name, salary)
company(company_name, city)
manages(person_name, manager_name)

1. Find the names of all employees who work for First Bank Corporation.

- **select** *person_name*
  from *works*
  where *company_name* = 'First Bank Corporation'

9

employee(person_name, street, city)
works(person_name, company_name, salary)
company(company_name, city)
manages(person_name, manager_name)

2. Find all employees in the database who live in the same cities as the companies for which they work.

- **select** *employee.person_name*
  from *employee*, *works*, *company*
  where *employee.person_name* = *works.person_name*
  and *employee.city* = *company.city*
  and *works.company_name* = *company.company_name*

10

employee(person_name, street, city)
works(person_name, company_name, salary)
company(company_name, city)
manages(person_name, manager_name)

2. Find all employees in the database who live in the same cities as the companies for which they work.

- **select** *e.person_name*
  from *employee* **as** *e*, *works* **as** *w*, *company* **as** *c*
  where *e.person_name* = *w.person_name* **and** *e.city* = *c.city*
  and *w.company_name* = *c.company_name*

11

employee(person_name, street, city)
works(person_name, company_name, salary)
company(company_name, city)
manages(person_name, manager_name)

3. Find all employees in the database who live in the same cities and on the same streets as do their managers.

- **select** *P.person_name*
  from *employee* **as** *P*, *employee* **as** *R*, *manages* **as** *M*
  where *P.person_name* = *M.person_name* **and**
  M.manager_name = *R.person_name* **and**
  P.street = *R.street* **and** *P.city* = *R.city*

12

■ 【0011】 Let R = (A,B,C), and let r_1 and r_2 both be relations on schema *R*. Give an expression in SQL that is equivalent to each of the following queries.

1. $r_1 \cup r_2$
2. $r_1 \cap r_2$
3. $r_1 - r_2$
4. $\Pi_{AB}(r_1) \bowtie \Pi_{BC}(r_2)$

13

1. $r_1 \cup r_2$

```
(select * from r1)
union
(select * from r2)
```

2. $r_1 \cap r_2$

```
(select * from r1)
intersect
(select * from r2)
```

```
select *
from r1
where (A, B, C) in (select *
                    from r2)
```

3. $r_1 - r_2$

```
(select * from r1)
except
(select * from r2)
```

```
select *
from r1
where (A, B, C) not in (select *
                        from r2)
```

4. $\Pi_{AB}(r_1) \bowtie \Pi_{BC}(r_2)$

```
select r1.A, r2.B, r3.C
from r1, r2
where r1.B = r2.B
```

14

student(ID, name, dept_name, credits)

instructor(ID, name, dept_name, salary)

course(course_id, title, dept_name, credits)

section(course_id, sec_id, semester, year, building, room_number, time_slot_id)

takes(ID, course_id, sec_id, semester, year, grade)

teaches(ID, course_id, sec_id, semester, year)

advisor(s_ID, i_ID)

15

■ 【0012】 *Write the following queries in SQL, using the university schema.*

1. Find the names of all students who have taken at least one Comp. Sci. course; make sure there are no duplicate names in the result.
2. For each department, find the maximum salary of instructors in that department. You may assume that every department has at least one instructor.
3. Find the IDs and names of all students who have not taken any course offering before Spring 2009.
4. Find the lowest, across all departments, of the per-department maximum salary computed by the preceding query.

16

1. Find the names of all students who have taken at least one Comp. Sci. course; make sure there are no duplicate names in the result.

```
select distinct name
from student, takes, course
where student.ID=takes.ID  and takes.course_id=course.course_id
        and course.dept = 'Comp. Sci.'
```

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

```
select dept, max(salary)
from instructor
group by dept
```

17

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

```
(select id, name
 from student )
except
(select id, name
 from student, takes
 where student.ID=takes.ID  and year < 2009 )
```

4. Find the lowest, across all departments, of the per-department maximum salary computed by the preceding query.

```
select min(maxsalary)
from (select dept, max(salary) as maxsalary
        from instructor group by dept)
```

18

employee(<u>person_name</u>, street, city)

works(<u>person_name</u>, company_name, salary)

company(<u>company_name</u>, city)

manages(<u>person_name</u>, manager_name)

19

【0017】 *Consider the employee database in the figure, where the primary keys are underlined. Give an expression in SQL for each of the following queries.*

1. Find all employees who earn more than the average salary of all employees of their company.
2. Find the company that has the smallest payroll.

20

1. **Find all employees who earn more than the average salary of all employees of their company.**

employee(<u>person_name</u>, street, city)

works(<u>person_name</u>, company_name, salary)

company(<u>company_name</u>, city)

manages(<u>person_name</u>, manager_name)

- **select** *person_name*
  from *works* **as** *T*
  where *salary* > (**select avg** (*salary*)
  from *works* **as** *S*
  where *T.company_name* = *S.company_name*)

21

2. **Find the company that has the smallest payroll.**

employee(<u>person_name</u>, street, city)

works(<u>person_name</u>, company_name, salary)

company(<u>company_name</u>, city)

manages(<u>person_name</u>, manager_name)

- **select** *company_name*
  from *works*
  group by *company_name*
  having sum (*salary*) <= **all** (**select sum** (*salary*)
  from *works*
  group by *company_name*)

22

【0015】 *Write an SQL query, without using a* ***with*** *clause, to find all branches where the total account deposit is less than the average total account deposit at all branches.*

a. Using a nested query in the **from** clause.

b. Using a nested query in a **having** clause.

23

find all branches where the total account deposit is less than the average total account deposit at all branches.

- **Using a nested query in the *from* clause.**

```
select branch_name, tot_balance
from (select branch_name, sum (balance)
      from account
      group by branch_name) as branch_total(branch_name, tot_balance)
where tot_balance <
      ( select avg (tot_balance)
      from ( select branch_name, sum (balance)
            from account
            group by branch_name) as branch_total(branch_name, tot_balance) )
```

24

find all branches where the total account deposit is less than the average total account deposit at all branches.

- Using a nested query in the ***having*** clause.

```
select branch_name, sum (balance)
from account
group by branch_name
having sum (balance) <
        ( select avg (tot_balance)
        from ( select branch_name, sum (balance)
                from account
                group by branch_name) as branch total(branch_name, tot_balance) )
```

25

【0019】Consider the relational database in the figure. Give an expression in SQL for each of the following queries.

1. **Give all employees of First Bank Corporation a 10 percent raise.**
2. **Give all managers of First Bank Corporation a 10 percent raise.**
3. **Delete all tuples in the *works* relation for employees of Small Bank Corporation.**

26

employee(<u>person_name</u>, street, city)

works(<u>person_name</u>, company_name, salary)

company(<u>company_name</u>, city)

manages(<u>person_name</u>, manager_name)

27

1. Give all employees of First Bank Corporation a 10 percent raise.

employee(<u>person_name</u>, street, city)

works(<u>person_name</u>, company_name, salary)

company(<u>company_name</u>, city)

manages(<u>person_name</u>, manager_name)

```
update works
set salary = salary * 1.1
where company name = 'First Bank Corporation'
```

28

2. Give all managers of First Bank Corporation a 10 percent raise.

employee(person_name, street, city)
works(person_name, company_name, salary)
company(company_name, city)
manages(person_name, manager_name)

update *works*
set *salary* = *salary* * *1.1*
where *employee name* **in** (**select** *manager name*
from *manages*)
and *company name* = ’First Bank Corporation

29

3. Delete all tuples in the *works* relation for employees of Small Bank Corporation.

employee(person_name, street, city)
works(person_name, company_name, salary)
company(company_name, city)
manages(person_name, manager_name)

delete *works*
where *company name* = ’Small Bank Corporation’

30

【0064】Suppose we have three relations *r*(*A*, *B*), *s*(*B*, *C*), and *t*(*B*, *D*), with all attributes declared as **not null.**

1. Give instances of relations *r*, *s*, and *t* such that in the result of
   (*r* ***natural left outer join*** *s*) ***natural left outer join*** *t*
   attribute *C* has a null value but attribute *D* has a non-null value.
2. Are there instances of *r*, *s*, and *t* such that the result of
   r ***natural left outer join*** (*s* ***natural left outer join*** *t*)
   has a null value for *C* but a non-null value for *D*? Explain why or why not.

31

1. Give instances of relations *r*, *s*, and *t* such that in the result of
   (*r* **natural left outer join** *s*) **natural left outer join** *t*
   attribute *C* has a null value but attribute *D* has a non-null value.

Consider r=(a, b),s=(b1,c1) and t=(b, d). With these relation instances, the expression “(r **natural left outer join** s) **natural left outer join** t” would give (a, b, null, d).

2. Are there instances of *r*, *s*, and *t* such that the result of
   r **natural left outer join** (*s* **natural left outer join** *t*)
   has a null value for *C* but a non-null value for *D*? Explain why or why not.

Since *s* **natural left outer join** *t* is computed first, the absence of nulls is both s and *t* implies that each tuple of the result can have *D* null, but *C* can never be null.

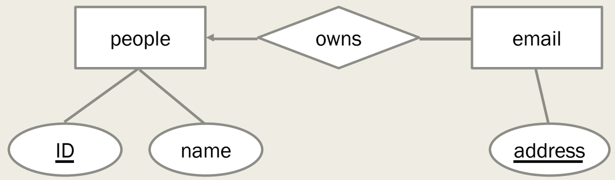
32

CHAPTER 5

——EXERCISES

1

Draw an ER diagram based on the following description: Suppose we have two entity sets, People and Email. Suppose we also use a relationship Owns, which connects these two entities. A person may own multiple email accounts, but an email account can only be owned by a single person.

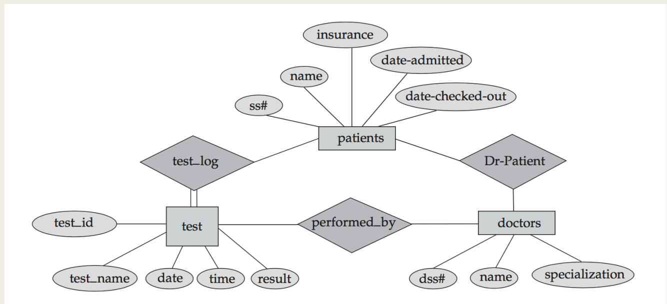


2

Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted. Assume :

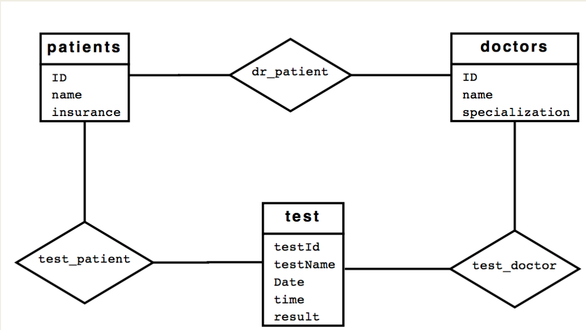
- *doctors* have attributes: *dss#, name, specialization*
- *patients* have attributes: *patient_id, name, insurance, date-admitted, date-checked-out*
- *test* have attributes: *test_id, testname, date, time, result*

3



E-R Diagram for a Hospital

4



E-R Diagram for a Hospital

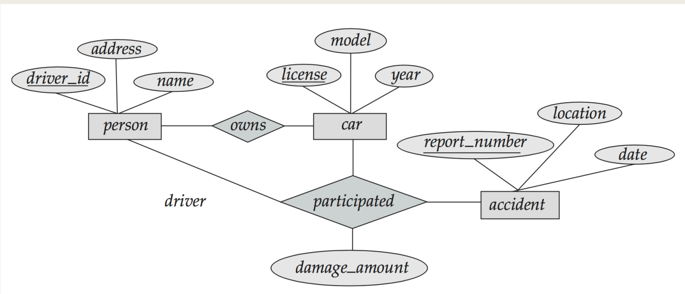
5

【0022】Construct an E-R diagram for a car insurance company. Each customer has at least one car. Associate with each car any number of accidents(include 0).

Assume:

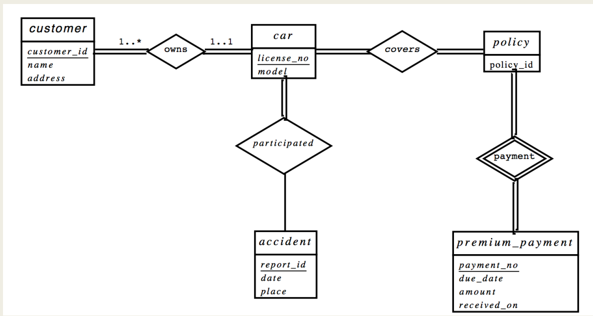
- *person: driver_id, name, address*
- *car: license, year, model*
- *accident: report_number, date, location*

6



E-R Diagram for a Car Insurance Company

7

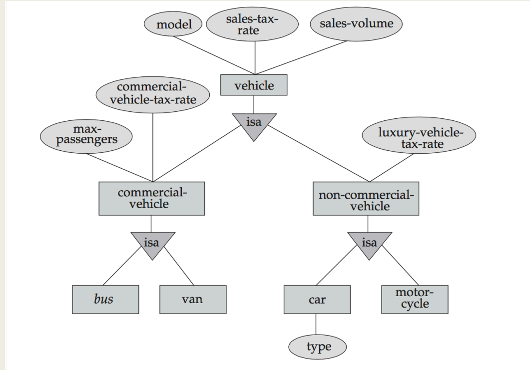


E-R Diagram for a Car Insurance Company

8

【0023】Design a generalization–specialization hierarchy for a motor-vehicle sales company.
The company sells motorcycles, passenger cars, vans, and buses. Justify your placement of attributes at each level of the hierarchy. Explain why they should not be placed at a higher or lower level.

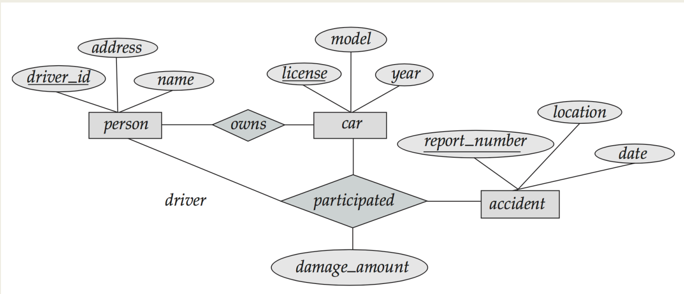
9



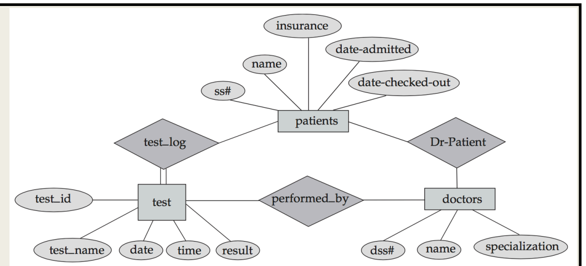
E-R Diagram for motor-vehicle sales company

10

【0025】Construct appropriate tables for the E-R diagrams



11

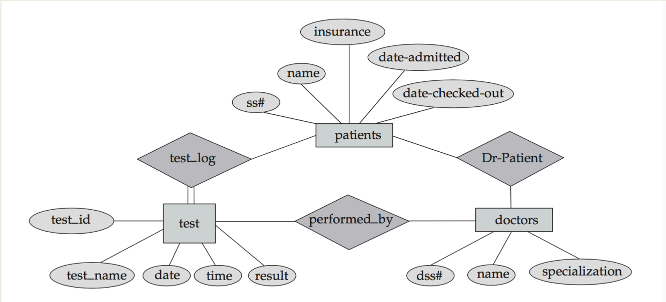


Car insurance tables:

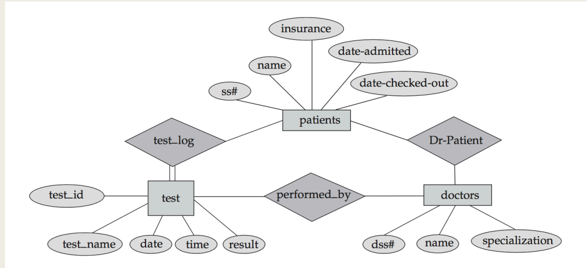
- person (<u>driver-id</u>, name, address)
- car (<u>license</u>, year, model)
- accident (<u>report-number</u>, date, location)
- participated(<u>driver-id</u>, <u>license</u>, <u>report-number</u>, damage-amount)

12

【0026】Construct appropriate tables for the following E-R diagrams



13



Hospital tables:

- patients (<u>ss#</u>, name, insurance, date-admitted, date-checked-out)
- doctors (<u>dss#</u>, name, specialization)
- test (<u>test_id</u>, testname, date, time, result)
- doctor-patient (<u>patient-id</u>, <u>doctor-id</u>)
- test-log (<u>testid</u>, <u>patient-id</u>)
- performed-by (<u>testid</u>, <u>dss#</u>)

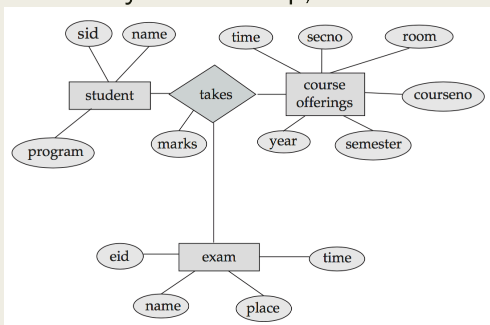
14

【0024】Consider a database used to record the marks that students get in different exams of different course offerings (or sections).

- Construct an E-R diagram that models *exams* as entities, and uses a ternary relationship, for the database.
- Construct an alternative E-R diagram that uses only a binary relation- ship between *student* and *section*. Make sure that only one relationship exists between a particular *student* and *section* pair, yet you can represent the *marks* that a *student* gets in different *exams*.

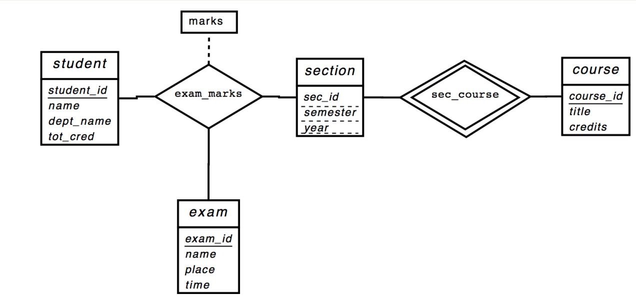
15

Construct an E-R diagram that models *exams* as entities, and uses a ternary relationship, for the database.



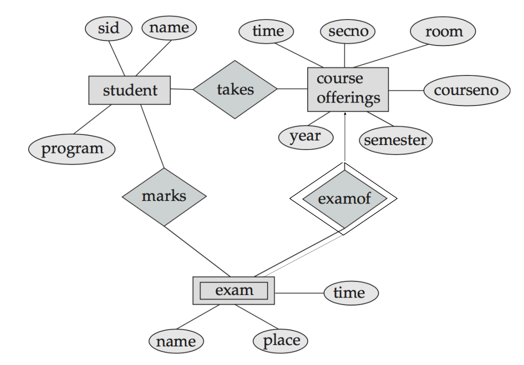
16

Construct an E-R diagram that models *exams* as entities, and uses a ternary relationship, for the database.



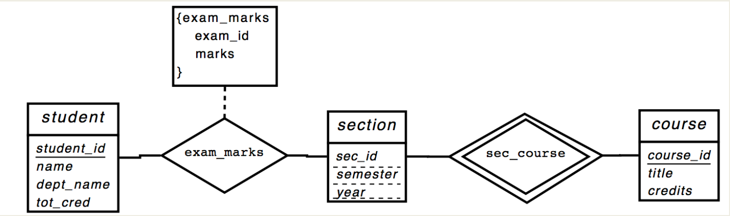
17

Construct an alternative E-R diagram that uses only a binary relation- ship between *student* and *section*.



18

Construct an alternative E-R diagram that uses only a binary relation- ship between *student* and *section*.



19

CHAPTER 6

——EXERCISES

1

Consider relation *R* with five attributes {*A B C D E*}, and the following functional dependencies: *A* →*B; BC*→*D; D*→*E*. Give the complete closure for {*A C*}

2

Consider relation *R* with five attributes {*A B C D E*}, and the following functional dependencies:

A →*B; BC*→*D; D*→*E*.

Give the complete closure for {*A C*}

- $\{A,C\}^+ = \{A,C\}$;
- Since *A*→*B* holds, we add B into results. Then $\{A,C\}^+ = \{A, B, C\}$;
- Since *BC*→*D* holds, $\{A,C\}^+$ changes to $\{A, B, C, D\}$;
- Since *D*→*E* holds, $\{A,C\}^+$ changes to $\{A, B, C, D, E\}$;
- Therefore, the closure for $\{A,C\}$ is $\{A,C\}^+ = \{A, B, C, D, E\}$.

3

Compute the closure of the following set F of functional dependencies for relation schema R = (A, B, C, D, E).

A→BC
CD→E
B→D
E→A
compute the canonical cover Fc.

4

Compute the closure of the following set F of functional dependencies for relation schema R = (A, B, C, D, E).
A⟶BC; CD⟶E; B⟶D; E⟶A
compute the canonical cover Fc.

- F_c={A⟶BC; CD⟶E; B⟶D; E⟶A}
- To test whether B is extraneous in A⟶BC, Consider F_1={A⟶C; CD⟶E; B⟶D; E⟶A}

  Since $\{A\}^+$={A, C} under F_1, B is not extraneous in A⟶BC.
- To test whether Cis extraneous in A⟶BC, Consider F_2={A⟶B; CD⟶E; B⟶D; E⟶A}

  Since $\{A\}^+$={A, B, D} under F_2, C is not extraneous in A⟶BC.
- To test whether C is extraneous in CD⟶E, Consider γ={CD}-C={D}. Since $\{\gamma\}^+$={D} under F, C is not extraneous in CD⟶E.
- To test whether D is extraneous in CD⟶E, Consider γ={CD}-D={C}. Since $\{\gamma\}^+$={C} under F, D is not extraneous in CD⟶E.
- Therefore, F_c={A⟶BC; CD⟶E; B⟶D; E⟶A}.

5

Consider a relation with schema R(*A, B, C, D, E. F*) and the set of functional dependency *F*={*E*⟶ *D, C*⟶*B, CE* ⟶*F, B* ⟶*A*} holds on relation *R*

- Give all candidate keys of this relation, motivate.
- Is this relation R in 3NF? If it is not, decompose it into relations in 3NF.

6

R(*A, B, C, D, E. F*)   *F*={*E*⟶ *D, C* ⟶*B, CE* ⟶*F, B* ⟶*A*}

- Since $\{CE\}^+$={A,B,C,D,E,F}, the candidate key of R is (C, E).
- Since *E*⟶ *D, C* ⟶*B, B* ⟶*A*, R is not in 3NF, we need to decompose R.

  There is no extraneous attribute in CE⟶F, Therefore

  F_c ={*E*⟶ *D*， *C* ⟶*B, CE* ⟶*F, B* ⟶*A*}

  We first generate four relational schemas {E, D},{C, B}, {C,E,F} and {B,A}.

  Since {C, E, F} already contains candidate key, the decomposing result is {E, D},{C, B}, {C,E,F} and {B,A}.

7

Consider a relation with schema R(*A, B, C, D, E, G*) and the set of functional dependency F={ A⟶B， B⟶C， AD⟶G， D⟶E }

- Give all candidate keys of this relation, motivate.
- Give the closure of attribute sets {G},{AD},{CD} and {BC},respectively.
- Is this relation R in BCNF？ If it is not, decompose it into relations in BCNF.

8

R(*A, B, C, D, E, G*) ,F={ A→B， B→C， AD→G， D→E }

- Since $(AD)^{+}$={A, B, C, D, E, G}, the candidate key is {AD}
- 1. $\{G\}^{+}$={G}　　2. $\{AD\}^{+}$={A,B,C,D,E,G}

  3. $\{C,D\}^{+}$={C, E, D}　　4. $\{B,C\}^{+}$={B, C}
- Since A→B, B→C, and D→E, R is not in 3NF.

  F_c={A→B， B→C， AD→G， D→E }

We first generate four relational schemas {A, B}, {B, C}, {A, D, G}, and {D, E }.

Since {A, D, G} already contains candidate key, the decomposing result is schemas {A, B}, {B, C}, {A, D, G}, and {D, E }

9

R(*A, B, C, D, E, G*) ,F={ A→B， B→C， AD→G， D→E }

- Since $(AD)^{+}$={A, B, C, D, E, G}, the candidate key is {AD}
- 1. $\{G\}^{+}$={G}　　2. $\{AD\}^{+}$={A,B,C,D,E,G}

  3. $\{C,D\}^{+}$={C, E, D}　　4. $\{B,C\}^{+}$={B, C}
- Since A→B, B→C, and D→E, R is not in BCNF.

  result={A, B, C, D, E, G}

  Since A→B and A →R not in F^{+}, we decompose R. result={A, C, D, E, G}∪{A, B}

  Since A → C and A →{A,C,D,E,G} not in F^{+}, we decompose to result={A, D, E, G}∪{A, B}∪{A, C}

  Since D→E and D →{A, D, E, G} not in F^{+}, we decompose to result={A, D, G}∪{A, B}∪{A, C}∪{D, E}

  Final, all the subschemas are in BCNF. Therefore, the decomposing result is A, D, G},{A, B},{A, C} and {D, E}.

10