Homework-2

Name: ABID ALI

Roll : 2019380141

Chapter 2 Introduction to the Relational Model

Question No:2.2

Consider the foreign key constraint from the dept name attribute of instructor to the department relation. Give examples of inserts and deletes to these relations, which can cause a violation of the foreign key constraint

Answer No:2.2

• Inserting into the instructor table, where the department table does not have the department Comp_Science, would violate the foreign key constraint.

For example insert into tuple: (12345, ABID, Comp_Science, 4000)

• Deleting the tuple: (Comp_Science, JOE, 2000) from the department table, where at least one student or instructor tuple has dept name as Comp_Science, would violate the foreign key constraint.

Question No:2.7

Consider the relational database of Figure 2.14. Give an expression in the relational algebra to express each of the following queries

employee (*person_name*, *street*, *city*)
works (*person_name*, *company_name*, *salary*)
company (*company_name*, *city*)

Figure 2.14 Relational database for Exercises 2.1, 2.7, and 2.12.

Answer No:2.7

a. Find the names of all employees who live in city “Miami”.

πperson_name (σcity = “Miami” (employee))

b. Find the names of all employees whose salary is greater than $100,000.

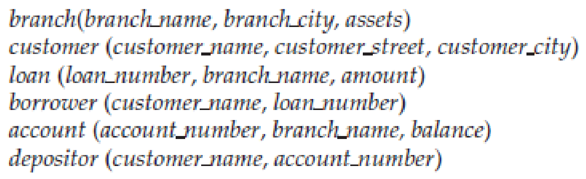
πperson_name (σsalary > 100000 (works))

c. Find the names of all employees who live in “Miami” and whose salary is greater than $100,000.

πperson_name (σcity = “Miami” ∧ salary>100000 (employee ⋈ works))

Question No:2.8

Consider the bank database of Figure 2.15. Give an expression in the relational algebra for each of the following queries.



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*)

Figure 2.15 Banking database for Exercises 2.8, 2.9, and 2.13.

Answer No:2.8

a. Find the names of all branches located in “Chicago”.

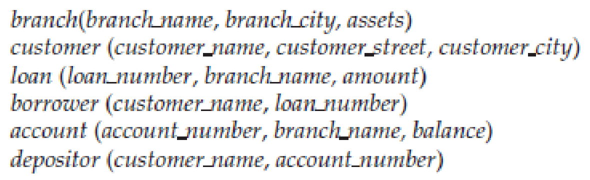
πbranch_name (σbranch city = “Chicago” (branch))

b. Find the names of all borrowers who have a loan in branch “Down-town”

πcustomer_name (σbranch name = “Downtown” (borrower ⋈ loan))

Question No:2.9

Consider the bank database of Figure 2.15.
a. What are the appropriate primary keys?
b. Given your choice of primary keys, identify appropriate foreign keys.



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*)

Figure 2.15 Banking database for Exercises 2.8, 2.9, and 2.13.

Answer No:2.9(a)

The primary keys of the various schema are underlined. Although in a real bank the customer name is unlikely to be a primary key , since two customers could have the same name, we use a simplified schema where we assume that names are unique. We allow customers to have more than one account, and more than one 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)

Answer No:2.9(b)

The foreign keys are as follows
i. For loan: branch name referencing branch.

ii. For borrower: Attribute customer name referencing customer andloan number referencing loan

iii. For account: branch name referencing branch.

iv. For depositor: Attribute customer name referencing customer and account number referencing account

Question No:2.12

Consider the relational database of Figure 2.14. Give an expression in the relational algebra to express each of the following queries:

a. Find the names of all employees who work for “First Bank Corporation”.
b. Find the names and cities of residence of all employees who work for “First Bank Corporation”.
c. Find the names, street address, and cities of residence of all employees who work for “First Bank Corporation” and earn more than $10,000.

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

Figure 2.14 Relational database for Exercises 2.1, 2.7, and 2.12.

Answer No:2.12

a. πperson_name (σcompany_name =“First Bank Corporation”(works))

b. πperson_name,city (employee⋈(σcompany_name = “First Bank Corporation”(works)))

c. πperson_name, street,city (σcompany_name = “First Bank Corporation” Λ salary>10000(works⋈employee))

Question No:2.13

Consider the bank database of Figure 2.15. Give an expression in the relational
algebra for each of the following queries:
a. Find all loan numbers with a loan value greater than $10,000.

b. Find the names of all depositors who have an account with a value greater than $6,000.

c. Find the names of all depositors who have an account with a value greater than $6,000 at the "Uptown" branch.

Answer No:2.13

a. πloan_number (σamount >10000 (loan))

b. πcustomer_name(σbalance> 6000"(depositor⋈account))

c. πcustomer_name(σbalance> 6000 ∧ branch_name>"Uptown" (depositor⋈account))

Chapter 3 Introduction to SQL

Question No 3.3

Write the following inserts, deletes or updates in SQL, using the university schema.
a. Increase the salary of each instructor in the Comp. Sci. department by 10%.
b. Delete all courses that have never been offered (that is, do not occur in the section relation).
c. Insert every student whose tot cred attribute is greater than 100 as an instructor in the same department, with a salary of $10,000.

Answer No 3.3

a. Increase the salary of each instructor in the Comp. Sci. department by 10%.

```
update instructor

set salary = salary * 1.10
where dept_name = ‘Comp. Sci.’
```

b. Delete all courses that have never been offered (that is, do not occur in the section relation).

```
delete from course
where course id not in
(select course_id from section)
```

c. Insert every student whose tot cred attribute is greater than 100 as an instructor in the same department, with a salary of $10,000.

```
insert into instructor
select ID, name, dept_name, 10000
from student
where tot_cred > 100
```

Question No 3.4

Consider the insurance database of Figure 3.18, where the primary keys are underlined. Construct the following SQL queries for this relational database.
a. Find the total number of people who owned cars that were involved in accidents in 2009.
b. Add a new accident to the database; assume any values for required attributes.
c. Delete the Mazda belonging to “John Smith”.

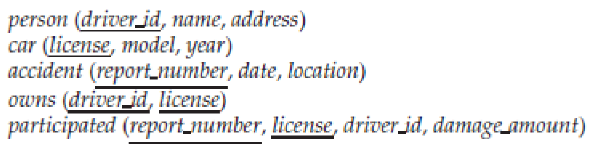

person (*driver_id*, *name*, *address*)
car (*license*, *model*, *year*)
accident (*report_number*, *date*, *location*)
owns (*driver_id*, *license*)
participated (*report_number*, *license*, *driver_id*, *damage_amount*)

Figure 3.18 Insurance database for Exercises 3.4 and 3.14.

Answer No 3.4(a)

```
select count (distinct name)

from accident, participated, person

where accident.report number = participated.report number
and participated.driver id = person.driver id

and date between date ’1989-00-00’ and date ’1989-12-31’
```

Answer No 3.4(b)

```
insert into accident
values (4007,‘2001-09-01’,‘Berkeley’)
insert into participated
select o.driver_id, c.license, 4007, 3000
from person p, owns o, car c
where p.name = ‘Jones’ and p.driver_id = o.driver_id and
```

o.license = c.license and c.model = ‘Suzuki’

Answer No 3.4(c)

delete car

where model = ‘Mazda’ and license in

(select license

from person p, owns o

where p.name = ’John Smith’ and p.driver_id = o.driver_id)

Question No 3.7

Consider the SQL query

select distinct

p.a1 from p, r1, r2

where p.a1 = r1.a1 or p.a1 = r2.a1

Under what conditions does the preceding query select values of p.a1 that are either in r1 or in r2? Examine carefully the cases where one of r1 or r2 may be empty

Answer No 3.7

The query selects those values of p.a1 that are equal to some value of r1.a1 or r2.a1 if and only if both r1 and r2 are non-empty. If one or both of r1 and r2 are empty, the cartesian product of p, r1 and r2 is empty, hence the result of the query is empty. Of course if p itself is empty, the result is as expected, i.e. empty.

Question No 3.8

Consider the bank database of Figure 3.19, where the primary keys are underlined. Construct the following SQL queries for this relational database.
a. Find all customers of the bank who have an account but not a loan.

b. Find the names of all customers who live on the same street and in the same city as “Smith”.

c. Find the names of all branches with customers who have an account in the bank and who live in “Harrison”

Answer No 3.8

a. (select customer_name

from depositor) except

(select customer_name

from borrower)

b. select F.customer_name

from customer F join customer S using(customer_street, customer_city)

where S.customer_name = ‘Smith’

c. select distinct branch_name

from account natural join depositor natural join customer

where customer_city = ‘Harrison’

Question No 3.9

Consider the employee database of Figure 3.20, where the primary keys are underlined. Give an expression in SQL for each of the following queries. a. Find the names and cities of residence of all employees who work for "First Bank Corporation.
c. Find the names, street addresses, and cities of residence of all employees who work for "First Bank Corporation" and earn more than $10,000.
d. Find all employees in the database who do not work for "First Bank Corporation".
e. Find all employees in the database who earn more than each employee of "Small Bank Corporation".
f. Assume that the companies may be located in several cities. Find all companies located in every city in which "Small Bank Corporation" is located.
g. Find the company that has the most employees.
h. Find those companies whose employees earn a higher salary, on average, than the average salary at "First Bank Corporation".

Answer No 3.8

a. select e.employee_name, city

from employee e, works w

where w.company_name = 'First Bank Corporation' and

w.employee_name = e.employee_name

b. select*

from employee

where employee name in

(select employee_name

from works

```
where company_name = ‘First Bank Corporation’ and salary > 10000)
```

```
c.select employee_name
from works
where company_name ≠ ‘First Bank Corporation
```

```
d. select employee_name
from works
where salary > all
(select salary
from works
where company_name = ‘Small Bank Corporation’)
```

```
e. select S.company_name
from company S
where not exists ((select city
from company
where company_name = ’Small Bank Corporation’)
except
(select city
from company T
where S.company_name = T.company_name))
```

```
f. select company_name
from works
group by company_name
having count (distinct employee_name) >= all
(select count (distinct employee_name)
from works
group by company_name
```

```
g. select company_name
from works
group by company
name having avg (salary) > (select avg (salary)
from works
where company_name = 'First Bank Corporation')
```

Question No 3.11

Write the following queries in SQL, using the university schema.
a. 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.

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

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

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

Answer No 3.11

```
a) SELECT name
FROM student NATURAL JOIN takes NATURAL JOIN course
WHERE course.dept ='Comp.Sci ;
```

```
b)
SELECT id, name
FROM student
EXCEPT(
SELECT id, name
FROM student NATURAL JOIN takes
WHERE year<2009);
```

```
c)
SELECT dept_name, MAX(salary)
FROM instructor
GROUP BY dept_name;
```

```
d)
SELECT MIN(maximum_salary) FROM(
SELECT dept_name,MAX(salary)maximum_salary
FROM instructor
GROUP BY dept_name
);
```

Question No 3.12

Write the following queries in SQL, using the university schema.
a. Create a new course “CS-001”, titled “Weekly Seminar”, with 0 credits.

b. Create a section of this course in Autumn 2009, with sec id of 1.

c. Enroll every student in the Comp. Sci. department in the above section. d. Delete enrollments in the above section where the student’s name is Chavez.

e. Delete the course CS-001. What will happen if you run this delete statement without first deleting offerings (sections) of this course.

f. Delete all takes tuples corresponding to any section of any course with the word “database” as a part of the title; ignore case when matching the word with the title.

Answer No 3.12

a)

INSERT INTO course(course_id,title,dept_name, credits) VALUES(‘CS-001’,’Weekly Seminar’,’Comp.Sci’;0);

b)

INSERT INTO SECTION(course_id,sec_id,semester, YEAR VALUES(‘CS-001’,’1’,’Autumn’,2009);

c)

INSERT INTO takes(ID,course_id,sec_id,semester, YEAR)

SELECT ID, ‘CS-0011’,’Autumn’,2009

FROM student

```
WHERE dept_name ='Comp.Sci';
```

d)

```
delete from takes
where(course_id='CS-
001')and(sec_id='1')and(semester='Autumn')and(year=2009)
and (ID in
(select iD
from student
where name ='Chavez
)
);
```

Question No 3.13

Write SQL DDL corresponding to the schema in Figure 3.18. Make any reasonable assumptions about data types, and be sure to declare primary and foreign keys.

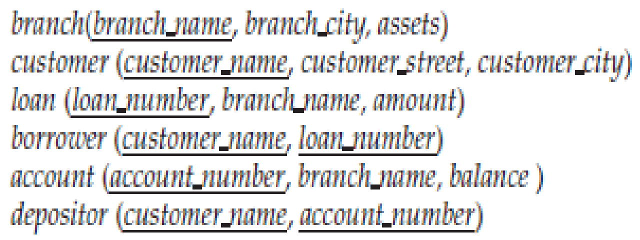
Answer No 3.13

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)

```
create table person(driver id varchar(50),
name varchar(50),
address varchar(50),

primary key (driver id))
create table car(license varchar(50),
model varchar(50),
year integer,
primary key (license))

create table accident(report number integer,
date date,
location varchar(50),
primary key (report number))


create table owns(driver id varchar(50),
license varchar(50),
primary key (driver id,license)
foriegn key (driver id)references personforiegn key (license)references car


create table participated(report number integer,license varchar(50),
driver id varchar(50),
damage amountinteger,
primary key (report number,license)foriegn key (license)references
carforiegn key (report number)references accident))
```

<u>Question No 3.15</u>

Consider the bank database of Figure 3.19, where the primary keys are underlined. Construct the following SQL queries for this relational database.
a. Find all customers who have an account at all the branches located in "Brooklyn".
b. Find out the total sum of all loan amounts in the bank.
c. Find the names of all branches that have assets greater than those of at least one branch located in "Brooklyn".



branch(<u>*branch_name*</u>, *branch_city*, *assets*)
customer (<u>*customer_name*</u>, *customer_street*, *customer_city*)
loan (<u>*loan_number*</u>, *branch_name*, *amount*)
borrower (<u>*customer_name*</u>, <u>*loan_number*</u>)
account (<u>*account_number*</u>, *branch_name*, *balance*)
depositor (<u>*customer_name*</u>, <u>*account_number*</u>)

Figure 3.19 Banking database for Exercises 3.8 and 3.15.

<u>Answer No 3.15</u>

a.)

```
with branchcount as(select count(*)
branchwhere branch city = ’Brooklyn’)
select customer namefrom customer c
where branchcount =(select count(distinct branch name)
from (customer natural join depositor natural join accountnatural join branch)
as d
where d.customer name = c.customer name)
```

b.)

select sum(amount)from loan

c.)

select branch namefrom branch

where assets >some

(select assetsfrom branchwhere branch city = ’Brooklyn’)

Question No 3.24

Consider the query:

with dept total (dept name, value) as

(select dept name, sum(salary)

from instructor

group by dept name),

dept total avg(value) as

(select avg(value)

from dept total)

select dept name

from dept total,

dept total avg where dept total.value >= dept total avg.value;

Rewrite this query without using the with construct.

Answer No 3.24

```
select dept_name
from(select dept_name,sum(salary)as value
from instructor
group by dept name)as dept_total,
(select avg(value as value1)
from dept_total) as dept_total_avg
where dept_total.value >= dept_total_avg.value1;
```