Midterm Solutions

Question 1

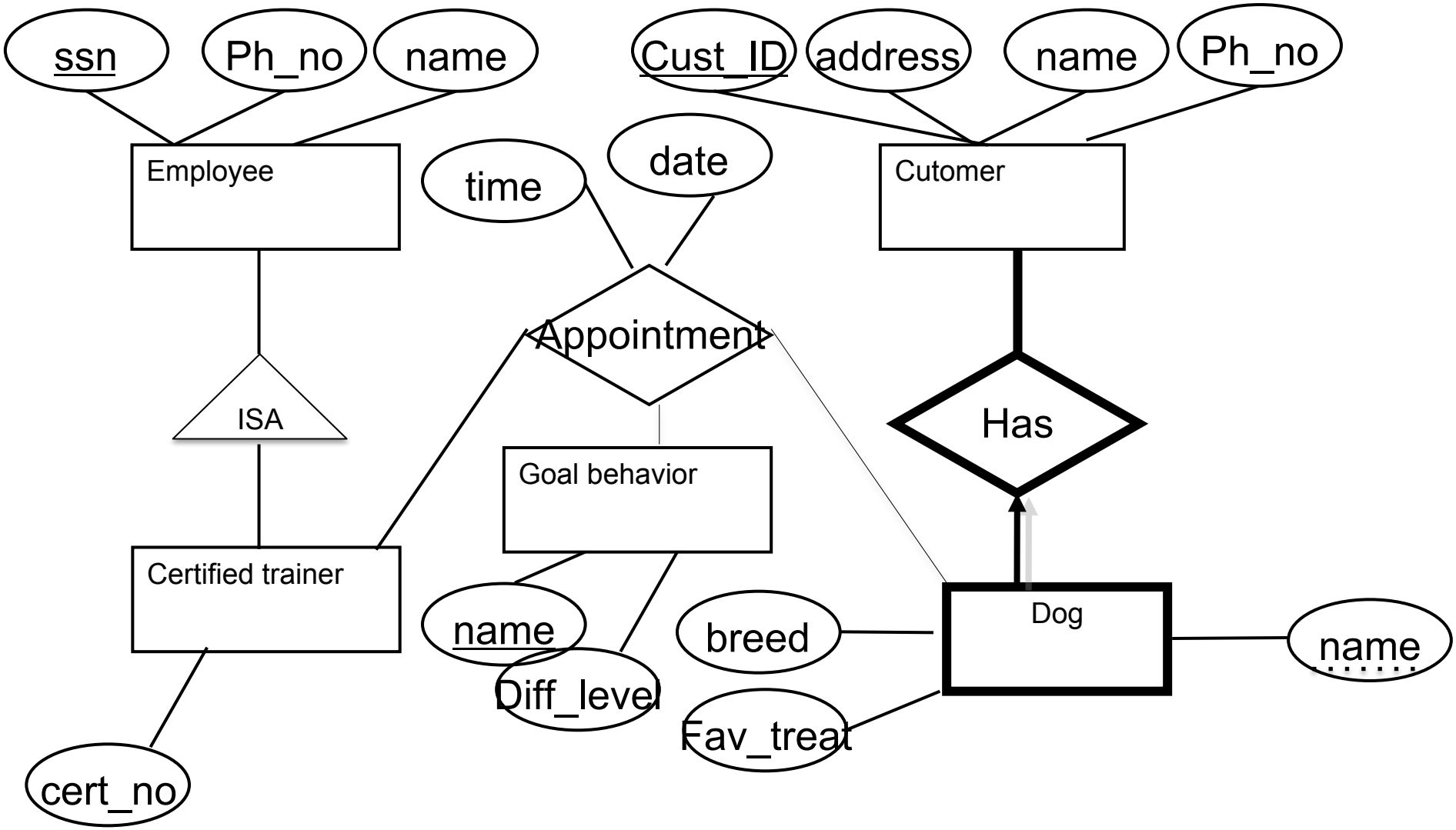
The Perfect Paws dog-training service has asked you to design a database for their business. Perfect Paws has many employees. Some of these employees are certified to train the customer's dogs towards different behavior goals. Each employee has a name, a phone number, and a unique social security number. Employees who are certified trainers also have a unique certification number.

Each customer has a name, a phone number, an address, and a unique customer ID. Each customer owns at least one dog, and each dog has exactly one owner. A dog has a name, a breed, and a favorite treat. Sometimes dogs have the same name, but the combination of customer ID and dog name is always unique (the same owner does not have multiple dogs with the same name).

Certified trainers can have appointments to train dogs on a date at a start time. For each such training appointment, there is a goal behavior that the trainer and dog will work on. Each goal behavior has a difficulty level and a unique name. The same trainer can train the same dog on multiple dates and start times only if the goal behaviors are different. A dog can be trained for the same behavior goal multiple times only if the trainers are different each time

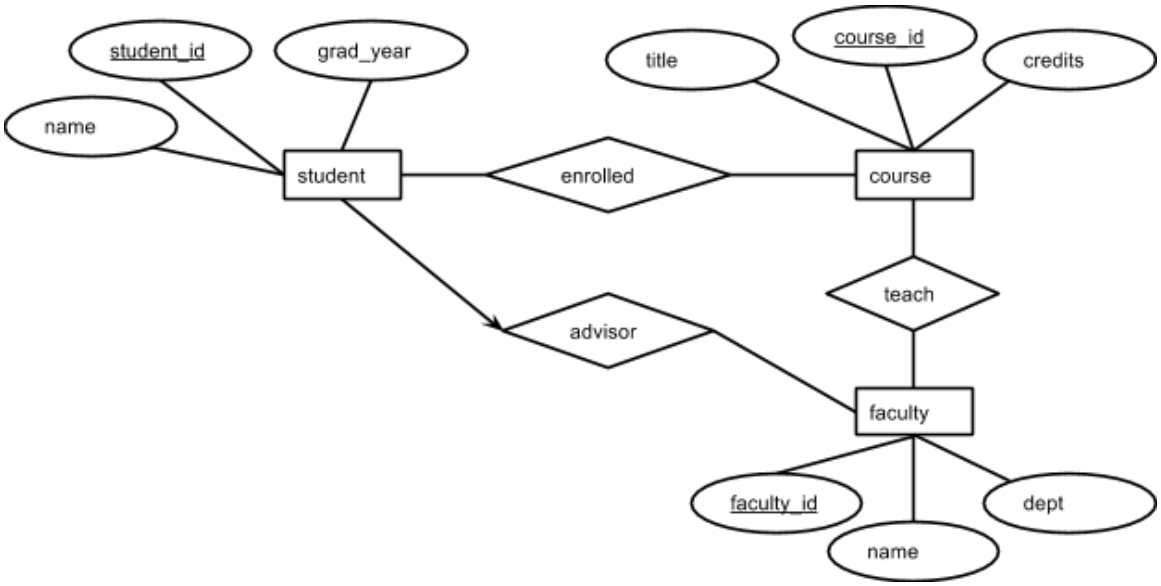
Draw an ER diagram for the Perfect Paws database requirements. Try not to make assumptions; simply draw the diagram for the requirements exactly as written. If you do make any assumptions, please write down what they are.

Question 1 Solution



Question 2. ER to SQL

Write the SQL statements to create all of the tables represented in the ER diagram below. Assume that you will be given the Course_ID for each course, the Faculty_ID for each faculty member, and the Student_ID for each student. You can assume that all strings are less than 50 characters and the primary keys for the entities are all integers.



Question 2 - Solution

Create table student(student_id integer, Grad_year varchar(50), name varchar(50),Primary key(student_id));

create table course (course_id integer, title varchar(50), credit varchar(50), Primary key(course_id));

create table enrolled(student_id integer, course_id integer,
primary key(student_id, course_id),
foreign key(student_id) references Student,
foreign key(course_id) references Course);

create table faculty(faculty_id integer, name varchar(50), dept varchar(50),
Primary key(faculty_id));

create table advisor(faculty_id integer, student_id integer,Primary key(student_id),
Foreign key(student_id) references Student,
Foreign key(faculty_id) references Faculty);

create table teach (faculty_id integer, course_id integer, Primary key(faculty_id, course_id),
Foreign key(student_id) references Student, Foreign key(faculty_id) references Faculty);

Question 2. Part b

Write a statement to add a course with the course ID 484 and title "Database Management Systems" that is worth 4 credits.

Solution: Insert into course values (484, ‘Database Management Systems’, ‘4’);

Question 3: Relational Algebra and calculus

Your new website, Freebook.com, is really taking off. In addition to storing and displaying their personal information, your users would also like the ability to designate other users as "friends." Suppose you create the following tables to store information about users, networks, network memberships, and friend relationships. (Primary keys are underlined.)

User (**uid**, username, age, sex, hometown)
Network (**nid**, description, networkname)
Member (**uid**, **nid**)
Friend (**uid1**, **uid2**)

Each user has an entry in the User table. Users may belong to zero or more networks, and network memberships are listed in the Member table. If the pair (U1,U2) appears in the Friend table, this means that user U1 has identified user U2 as a friend. Notice that unlike many social network sites, in your system this relationship is not symmetric; (U2,U1) means that user U2 has identified user U1 as a friend.

Question 3 – Part a

Write a relational algebra expression to find the names of all users aged 18-24 (inclusive) from hometown ‘Ann Arbor’.

Solution:

$$\Pi_{\text{username}} (\sigma_{\text{hometown='Ann arbor' ^ age>=18 ^ age =< 24}} (\text{User}))$$

Question 3 – Part b

Write a tuple relational calculus query equivalent to your relational algebra expression in Part a.

Solution: {**T** | **∃U** E user (U.hometown=’Ann Arbor’
^ U.age>=18
^ U.age<=24 ^
T.name= U.username}

Question 3 – Part c

Write a SQL query to find the names of all users who are NOT the oldest user in atleast one of their network

Solution1:

```
select U.username
from user U
where exists (select u2.age
                from user u2, member m1, member m2
               where u2.age > u.age and u2.uid = m2.uid
               and u.uid=m1.uid and m1.nid=m2.nid);
```

Question 3 – Part c

Write a SQL query to find the names of all users who are NOT the oldest user in atleast one of their network

Solution2:

Select u1.username
from user u1, member m1, network N, User U2, member m2
where U1.uid=m1.uid and u2.uid=m2.uid and n.nid=m1.nid and n.nid=m2.nid and u1.age<u2.age;

Question 3 – Part d

Write a SQL query to return the network name and the average age of the members in that network, for every network where (a) the network has at least 10 members, and (b) the network's average member age is at least 20% lower than the maximum age of all members across all networks. E.g., if the maximum age of all members is X, then the average age of every returned group should be <= 0.8*X. (Feel free to use views if you need to.)

Question 3 – Part d Solution

```
create view max_age as
Select max(age) as age
From user U, network N, member M
Where U.uid = M.uid and N.nid = M.nid;

select N.networkname, avg(U.age)
from user U, Network N, member M, max_age MA
Where U.uid = M.uid and N.nid = M.nid
group by N.nid, N.networkname
having count(*) >= 10 and avg(U.age) <= 0.8*(MA.age);
```

QUESTION 4 – Part a

Consider a relation R with attributes <A, B, C, D, E> and the following functional dependencies:

- A -> BC
- CD -> E
- B -> D
- E -> A

Part a) Is R in 3NF? Why or Why not?

Ans: The keys are CD, BC, A, E.

So the relation is in 3NF

QUESTION 4 – Part b

Suppose you are given the following relation R: ABCDEF
The following functional dependencies must hold:

- A -> BC
- CD -> E
- B -> D
- E -> A

a) Compute a lossless join decomposition of R into BCNF relations. (You need to show the different iterations of the algorithm and explain why the outcome relations are in BCNF).

Ans: B-> D violates BCNF.
The decomposition
R1: BD, R2: ABCE.
The keys for R1 = B and key for R2 is A, E.

QUESTION 4 – Part c

Suppose you are given the following relation R: ABCDEF
The following functional dependencies must hold:

- A -> BC
- CD -> E
- B -> D
- E -> A

If your decomposition dependency preserving? Why or why not?

Ans: No, it is not dependency preserving, because we cannot get the FD CD-> E.

QUESTION 5 – Part a

Which of the following statements are **NOT** true? (Circle all that apply.)

i. One database can have more than one external schema.
ii. One database can have more than one conceptual schema.
iii. One database can have more than one physical schema.
iv. One database can have more than one user at the same time.

QUESTION 5 – Part a

Which of the following statements are **NOT** true? (Circle all that apply.)

i. One database can have more than one external schema.

ii. One database can have more than one conceptual schema.

iii. One database can have more than one physical schema.

iv. One database can have more than one user at the same time.

QUESTION 5 – Part b

We have decided to create several views on our Sailors table. This is a decision about: (Circle only one)

i. The external schema
ii. The conceptual schema
iii. The physical schema
iv. None of the above

QUESTION 5 – Part b

We have decided to create several views on our Sailors table. This is a decision about: (Circle only one)

i. The external schema
ii. The conceptual schema
iii. The physical schema
iv. None of the above

QUESTION 5 – Part c

Consider the following table.

```
CREATE TABLE Salaries (
    empid integer,
    name char(20),
    salary REAL,
    primary key(empid), check (salary >=0)
  );
```

Given this schema, is it possible for following two queries to return different results? Why?

a) SELECT empid, salary FROM Salaries;

b) SELECT empid, salary FROM Salaries WHERE salary >= 0;

Solution: In cases when ‘null’ is entered for salary, the first query will return the tuple while the second does not.

QUESTION 5 – Part d

In JDBC, what is one reason why you would use a PreparedStatement instead of a Statement?

Solution:

a) Security,

b) Efficiency

Either one is sufficient

Wrong answers: being able to iterate through the results, being able to change the parameters, being able to run the query many times, etc.

QUESTION 6 – Aggregation – Part a

Consider the following instance of relation UmichEmp(name, age, salary).

Name	Age	Salary
Brian	20	80,000
Jackie	40	NULL
Michael	60	20,000
Michael	NULL	40,000
Sarah	NULL	NULL

SELECT sum(salary) from UmichEmp where Age > 30;

Ans: 20000

QUESTION 6 – Aggregation – Part b

Consider the following instance of relation UmichEmp(name, age, salary).

Name	Age	Salary
Brian	20	80,000
Jackie	40	NULL
Michael	60	20,000
Michael	NULL	40,000
Sarah	NULL	NULL

SELECT count(*) from UmichEmp where Age > 30 and Salary < 50000;

Ans: 1

QUESTION 6 – Aggregation – Part c

Consider the following instance of relation UmichEmp(name, age, salary).

Name	Age	Salary
Brian	20	80,000
Jackie	40	NULL
Michael	60	20,000
Michael	NULL	40,000
Sarah	NULL	NULL

SELECT count(*) from UmichEmp group by Name;

Ans: 1
1
2
1

QUESTION 6 – Aggregation – Part d

Consider the following instance of relation UmichEmp(name, age, salary).

Name	Age	Salary
Brian	20	80,000
Jackie	40	NULL
Michael	60	20,000
Michael	NULL	40,000
Sarah	NULL	NULL

SELECT Name from UmichEmp where Salary=20000 AND Salary=40000;

Ans: Empty set

Wrong Answers: **NULL**, Error, Invalid Query, ...

(Note: Null value is a very different concept from an empty set!)