

Midterm I

CS425 - Database Organization

Instructions

- Try to answer all the questions using what you have learned in class
- **When writing a query, write the query in a way that it would work over all possible database instances and not just for the given example instance!**
- Exam includes 4 relational algebra questions, one DDL, 2 SQL Queries and one update SQL and an er diagram and reduction
- I will work on my own on the exam and I will not share my answers or discuss it with anyone even after completing the exam.
- Signature

Consider the following database schema and example instance storing information about medical care:

hospital

Hospital	Doctor	County
Dupage	Bob	Dupage
GoodSam	Bob	Dupage
Northwesertn	Joan	Cook

procedure

pName	Recovery time	Anesthesia
Heart	15	Yes
Bladder	5	No
Lung	5	Yes

patient

pName	insurance	age	County
Sam	HMO	13	Dupage
Paul	PPO	87	Cook

takeCare

patient	pName	doctor	hospital
Sam	Lung	Bob	Dupage
Paul	Heart	Bob	GoodSam

doctor

docName	insurance	rate
Joan	HMO	15,000
Joan	PPO	20,000
Bob	HMO	12,000
Bob	PPO	14,000

Hints:

- Attributes with black background form the primary key of a relation.
 - The attribute *pName* of relation *takeCare* is a foreign key to relation *procedure*.
 - The attribute *patient* of relation *takeCare* is a foreign key to relations *patient*.
 - The combination of attribute *doctor* and *hospital* of relation *takeCare* is the foreign key to relations *hospital*.
 - The attribute *doctor* of relation *takeCare* stores doctors. However, it is not a foreign key to relation *doctor*, because the primary key of that relation also includes insurance information.
-

Part 1 Relational Algebra

Question 1.1 (3 points)

Write a relational algebra expression that return the patient's name along with the cost that had hearts or lungs procedure.

$\Pi_{\text{Patient.pName}, \text{Doctor.rate}}(\sigma_{\text{pName} = \text{'Heart'} \vee \text{pName} = \text{'Lung'}}(\text{Doctor} \bowtie (\text{Patient.insurance} = \text{Doctor.insurance}) \wedge \text{TakeCare.doctor} = \text{Doctor.docName} \text{ TakeCare}))$

Question 1.2 (2.5points)

Write a relational algebra expression that return the patient's name that had the procedure performed in hospital located in a different county than their residence.

$\Pi_{\text{Patient.pName}}(\text{Patient} - \text{Patient} \bowtie \text{Hospital})$

Question 1.3 (2.5 points)

Write a relational algebra expression that return the patient's age for all patient with lungs disease (had a lungs procedure)

$\Pi_{\text{Patient.age}}(\sigma_{\text{TakeCare.pName} = \text{'Lung'}}(\text{Patient} \bowtie_{(\text{Patient.pName} = \text{TakeCare.patient})} \text{TakeCare}))$

Question 1.4 (2.5 points)

Write a relational algebra expression that return the patient's name and recovery time needed for each procedure they have done.

$\Pi_{\text{Patient.pName}, \text{procedure.recoveryTime}}(\text{Patient} \bowtie_{(\text{Patient.pName} = \text{TakeCare.patient})} \text{TakeCare} \bowtie \text{Procedure})$

Part 2.1 SQL DDL

Question 2.1 (2.5 Points)

Write an SQL statement that creates a new table *treatment* that stores the *hospitalname* and the *procedure* and the *patientname*. Furthermore, we want to store a *hospitalFee* for each such assignment. The combination of procedure, hospital, and patient uniquely identifies an assignment. Each assignment has a *hospitalFee* that is bigger than 0 and smaller than 1,000,000 dollar. When *hospitalname* is removed from the hospital table it get deleted from the treatment table

```
CREATE TABLE treatment (hospitalName varchar (30), Doctor varchar(30)
(HospitalName, Doctor) foreign key references hospital (hospital, doctor) ONDELETE CASCADE, // no reduction for not
mentioning doctor, however you have to reference the entire primary key of the hospital table
procedure varchar(30) foreign key references procedure,
patientName varchar(30) foreign key references patient,
hospitalFee INTEGER CHECK(hospitalFee > 0 AND hospitalFee < 1000000),
PRIMARY KEY (hospitalName, patientName, procedure));
```

Part 2.2 Query

Question 2.2.1 (2 Points)

Write an SQL query that returns county for which the average age of patient is below 35.

select county FROM

select country, avg (age) from patient GROUP BY county HAVING av(age) < 35;

Question 2.2.1 (2.5 Points)

Return the name and rate for all doctors that support HMO combined with each patient they are taking care of.

SELECT doctor.docName, doctor.rate, takeCare.patient FROM doctor JOIN takeCare ON (doctor.docName = takeCare.doctor) WHERE doctor.docName. insurance = 'HMO';

Part 2.3 SQL Updates

Question 2.3.2 (1.5 Points)

Increase the rate of all doctors for HMO insurances by 1,000.

UPDATE doctor SET rate = rate + 1000 WHERE insurance = 'HMO';

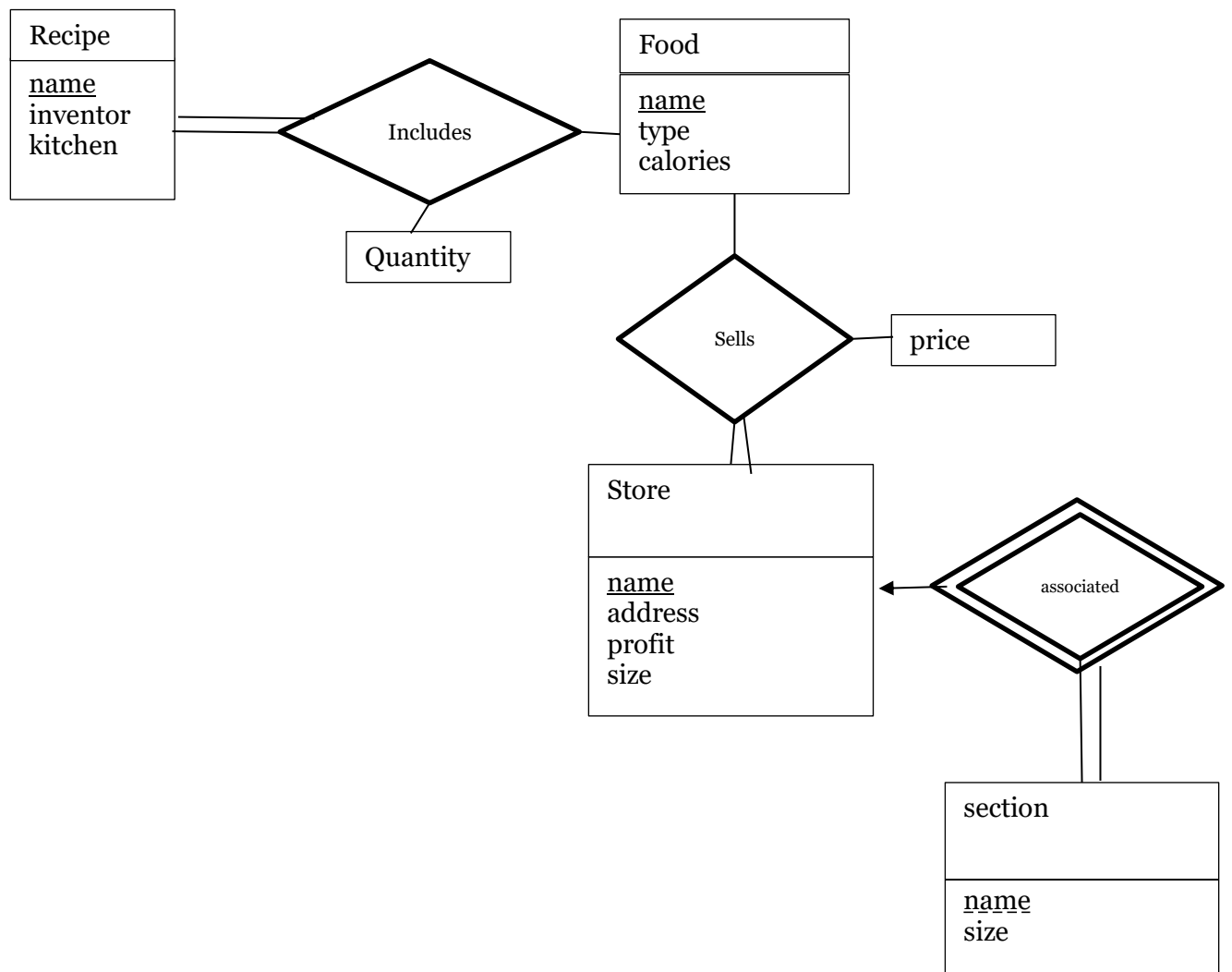
Part 3 ER diagram food managment

Recipe information is identified by recipe name, it includes inventor and kitchen A food item is identified by its name and has type and number of calories as attribute. A store is identified by its name and has other attributes: address, it has yearly profit and size. A section is associated with a store and has discriminator name, and size

The foodItem is sold one or multiple stores and has a listed price in each store.

Each recipe has list of ingredient name (food item) and with associated quantity.

Question 3.1 create an ER diagram the food management relation (3.5 points)



Question 3.2 Create the relational schema described in the ER above (3 points)

recipe(name, inventor, kitchen) – Rule 1
 food(name, type, calories) – Rule 1
 store(name, address, profit, size) – Rule 1
 section(storeName, sectionName, size) – Rule 2
 includes(recipeName, foodName, quantity) – Rule 5
 sells(storeName, foodName, price) – Rule 5
