CSE 4/560
Databases and Query Languages
Homework 1 Solution
Total Marks 60

Your Name:
Your email ID:
Your UB Person ID:

1. [20] Explain the following terms briefly: attribute, domain, entity, relation, relational database schema, entity set, relationship set.

Attribute - a property or description of an entity. A toy department employee entity could have attributes describing the employee's name, salary, and years of service.

Domain - a set of possible values for an attribute.

Entity - an object in the real world that is distinguishable from other objects such as the green dragon toy.

Relationship - an association among two or more entities.

Entity set - a collection of similar entities such as all of the toys in the toy department.

Relationship set - a collection of similar relationships

A relation schema can be thought of as the basic information describing a table or relation. This includes a set of column names, the data types associated with each column, and the name associated with the entire table. For example, a relation schema for the relation called Students could be expressed using the following representation:

Students(sid: string, name: string, login: string, age: integer, gpa: real)

There are five fields or columns, with names and types as shown above. A relational database schema is a collection of relation schemas, describing one or more relations.

2. [5] Given the following relational schema, give two examples of foreign key constraints **that involves** the Dept relation.

Emp(*eid:* `integer`, *ename:* `string`, *age:* `integer`, *salary:* `real`)
Works(*eid:* `integer`, *did:* `integer`, *pcttime:* `integer`)
Dept(*did:* `integer`, *dname:* `string`, *budget:* `real`, *managerid:* `integer`)

```
CREATE TABLE Works(
eid INTEGER NOT NULL,
did INTEGER NOT NULL,
pettime INTEGER,
PRIMARY KEY(eid, did),
UNIQUE (eid),
FOREIGN KEY (did) REFERENCES Dept
);
CREATE TABLE Dept(
did INTEGER NOT NULL,
budget REAL,
managerid INTEGER,
PRIMARY KEY(did),
FOREIGN KEY (managerid) REFERENCES Emp
);
```

3. [6] Explain the statement that relational algebra operators can be composed. Why is the ability to compose operators important?

Every operator in relational algebra accepts one or more relation instances as arguments and the result is always an relation instance. So the argument of one operator could be the result of another operator. This is important because, this makes it easy to write complex queries by simply composing the relational algebra operators.

4. [12] Briefly explain the significance of the followings in representing information in the real world: a) data definition langauge; b) data manipulation language; c) data model.

a) DDL commands are used for defining relation schemas, deleting relations, and modifying relation schemas.

b)DML commands provides the ability to query information from the database and to insert tuples into, delete tuples from, and modify tuples in the database.

c)A data model is a notation for describing data or information. The description generally consists of three parts: (1) Structure of the data: in the database world, data models are at a somewhat higher level than data structures, and are sometimes referred to as a conceptual model to emphasize the difference in level. (2) Operations on the data: In database data models, there is usually a limited set of operations that can be performed. We are generally allowed to perform a limited set of queries (operations that retrieve information) and modifications (operations that change the database). This limitation is not a weakness, but a strength. By limiting operations, it is possible for programmers to describe database operations at a very high level, yet have the database management system implement the operations efficiently. In comparison, it is generally impossible to optimize programs in

conventional languages like C, to the extent that an inefficient algorithm (e.g., bubblesort) is replaced by a more efficient one (e.g., quicksort). (3) Constraints on the data. Database data models usually have a way to describe limitations on what the data can be. These constraints can range from the simple (e.g., "a day of the week is an integer between 1 and 7" or "a movie has at most one title") to some very complex limitations.

5. [5] What is the difference between the natural join $R \times S$ and the theta-join $R \bowtie S$ where the condition C is that R.A = S.A for each attribute A appearing in the schemas of both R and S?

Each will have the same number of tuples, but$R \bowtie S$ conditioned on C, will have two copies of the attributes the relations have in common. The natural join $R \times S$ will consist of all the joined tuples of R and S, with common attributes collapsed.

6.[12] Consider the database schema consists of four relations,whose schemas are:

Product(maker, model, type)
PC(model, speed, ram, hd, price)
Laptop(model, speed, ram, hd, screen, price)
Printer(model, color, type, price)

The Product relation gives the manufacturer, model number and type (PC, laptop, or printer) of various products. We assume for convenience that model numbers are unique over all manufacturers and product types; that assumption is not realistic, and a real database would include a code for the manufacturer as part of the model number. The PC relation gives for each model number that is a PC the speed (of the processor, in gigahertz), the amount of RAM (in megabytes), the size of the hard disk (in gigabytes), and the price. The Laptop relation is similar, except that the screen size (in inches) is also included. The Printer relation records for each printer model whether the printer produces color output (true, if so), the process type (laser or ink-jet, typically), and the price.

Write the following declarations:

a) A suitable schema for relation Product.

```
CREATE TABLE Product (
maker CHAR(30),
model CHAR(10) PRIMARY KEY,
type CHAR(15)
);
```

b) A suitable schema for relation PC.

```
CREATE TABLE PC (
```

model CHAR(30),
speed DECIMAL(4,2),
ram INTEGER,
hd INTEGER,
price DECIMAL(7,2)
);

c) A suitable schema for relation Laptop.

CREATE TABLE Laptop (
model CHAR(30),
speed DECIMAL(4,2),
ram INTEGER,
hd INTEGER,
screen DECIMAL(3,1),
price DECIMAL(7,2)
);

d) A suitable schema for relation Printer.

CREATE TABLE Printer (
model CHAR(30),
color BOOLEAN,
type CHAR (10),
price DECIMAL(7,2)
);

e) An alteration to your Printer schema from (d) to delete the attribute color.

ALTER TABLE Printer DROP color;
f) An alteration to your Laptop schema from (c) to add the attribute od (optical-disk type, e.g., cd or dvd). Let the default value for this attribute be 'none' if the laptop does not have an optical disk.
ALTER TABLE Laptop ADD od CHAR (10) DEFAULT 'none';