COMP 5/6120

Database Systems I
Midterm Exam

Name: **Sample**
Student ID: _____

	Points	Received
Problem 1	20	
Problem 2	20	
Problem 3	20	
Problem 4	20	
Problem 5	20	
Total	100	

Exam Rules:

1. Close book and notes, 75 minutes.
2. Please write down your name and student ID number.
3. Please wait until being told to start reading and working on the exam.

Problem 1 Database Concepts (20 points)

(1) Unlike assertions which must be, in principle, checked any time when data is modified, triggers allow users to specify when checking should occur.
True **False**

(2) For any SQL query, there exists a unique translation into relational algebra.
True **False**

(3) In SQL, the value NULL is ignored in any aggregation.
True **False**

(4) In SQL, a view can be used like a stored relation in any operations.
True **False**

(5) The relational data model was proposed by Dr. Edgar Codd in early 1970's.
True **False**

(6) Schema normalization not only reduces potential data redundancy but also enhances query efficiency.
True **False**

(7) SQL is a declarative query language, in which we simply declare what we want, but not how to compute, in formulating a query.
True **False**

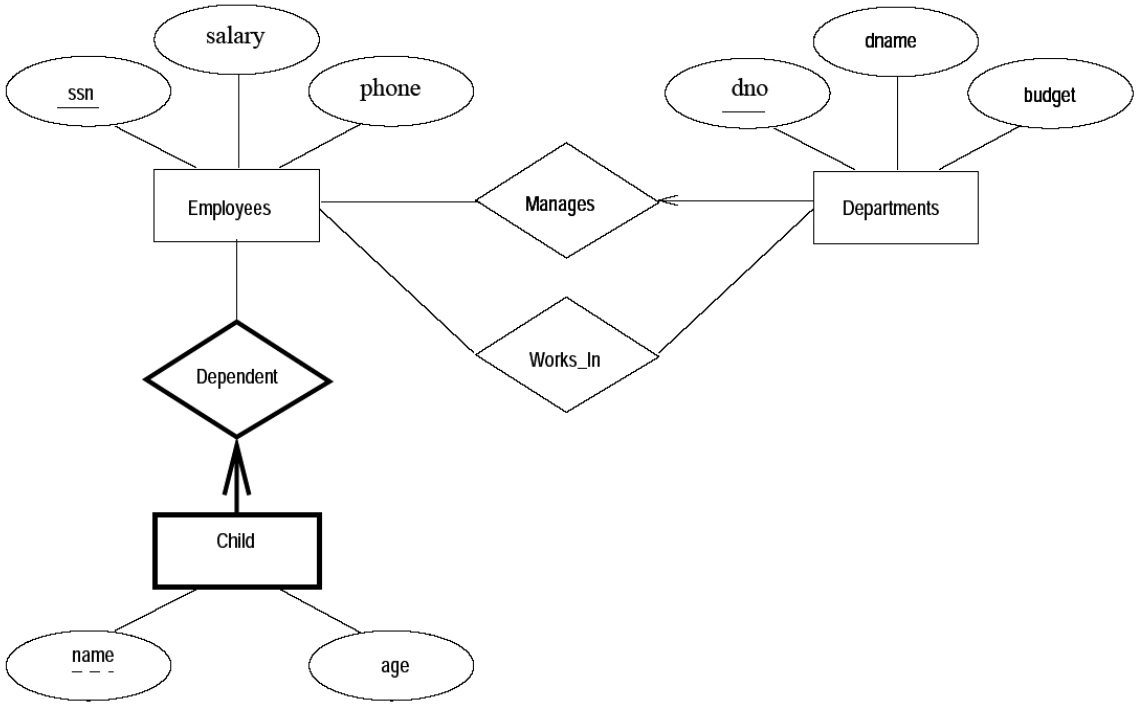
(8) With the five basic operators (selection, projection, union, set-difference, and cross-product), relational algebra can compose most queries. Other operators are just syntactic sugar and can be derived from the basic operations.
True **False**

(9) A natural join is a special case of an equijoin.
True **False**

(10) An aggregation function, e.g., SUM and AVG, returns a value computed from a set of values. Thus, MIN and MAX are not aggregate, since they only return a single value.
True **False**

Problem 2 ER and Translation to the Relational Model (20 points)

Translate the following ER diagram into a relational schema. For each relation in your schema, specify the key of that relation.



```
CREATE TABLE Employees ( ssn      CHAR(10),
                         sal      INTEGER,
                         phone    CHAR(13),
                         PRIMARY KEY (ssn) )


CREATE TABLE Departments ( dno      INTEGER,
                           budget   INTEGER,
                           dname    CHAR(20),
                           PRIMARY KEY (dno) )


CREATE TABLE Works_in (  ssn      CHAR(10),
                         dno      INTEGER,
                         PRIMARY KEY (ssn, dno),
                         FOREIGN KEY (ssn) REFERENCES Employees,
                         FOREIGN KEY (dno) REFERENCES Departments)


CREATE TABLE Manages (   ssn      CHAR(10),
                         dno      INTEGER,
                         PRIMARY KEY (dno),
                         FOREIGN KEY (ssn) REFERENCES Employees,
                         FOREIGN KEY (dno) REFERENCES Departments)


CREATE TABLE Dependents (ssn      CHAR(10),
                         name     CHAR(10),
                         age      INTEGER,
                         PRIMARY KEY (ssn, name),
                         FOREIGN KEY (ssn) REFERENCES Employees,
                                 ON DELETE CASCADE )
```

Problem 3 Relational Algebra (20 points)

Consider the following schema:

Suppliers (*sid:* integer, *sname:* string, *address:* string)
Parts (*pid:* integer, *pname:* string, *color:* string)
Catalog (*sid:* integer, *pid:* integer, *cost:* real)

The key fields are underlined, and the domain of each field is listed after the field name. Write the following queries in ***relational algebra***.

(1) Find the Supplier names of the suppliers who supply a red part that costs less than 100 dollors and a green part that costs less than 100 dolloars. (5 pts)

$$(\pi_{sname}((\sigma_{color='red'}Parts) \bowtie (\sigma_{cost<100}Catalog) \bowtie Suppliers)) \cap (\pi_{sname}((\sigma_{color='green'}Parts) \bowtie (\sigma_{cost<100}Catalog) \bowtie Suppliers))$$

(2) Find the *pids* of the most expensive parts supplied by suppliers named Andy. (5 pts)

$$\rho(R1, \pi_{sid}\sigma_{sname='\text{Andy}'}Suppliers)$$
$$\rho(R2, R1 \bowtie Catalog)$$
$$\rho(R3, R2)$$
$$\rho(R4(1 \to sid, 2 \to pid, 3 \to cost), \sigma_{R3.cost<R2.cost}(R3 \times R2))$$
$$\pi_{pid}(R2 - \pi_{sid,pid,cost}R4)$$

(3) Find the *pids* of parts supplied by at least two different suppliers. (5 pts)

$$\rho(R1, Catalog)$$
$$\rho(R2, Catalog)$$
$$\pi_{R1.pid}\sigma_{R1.pid=R2.pid \wedge R1.sid \neq R2.sid}(R1 \times R2)$$

(4) Find pairs of *sids* such that the supplier with the first *sid* charges more for some part than the supplier with the second *sid*. (5 pts)

$$\rho(R1, Catalog)$$
$$\rho(R2, Catalog)$$
$$\pi_{R1.sid, R2.sid}(\sigma_{R1.pid=R2.pid \wedge R1.sid \neq R2.sid \wedge R1.cost > R2.cost}(R1 \times R2))$$

Problem 4 SQL (20 points)

Consider the relational conceptual database schema used by company Goldfish, Inc. to store information about fish, species and fish tanks (sno is the species number and tno is the tank number):

Fish (fname varchar, fcolor varchar, fweight int, sno int, tno int);
Species (sno int, sname varchar, sfood varchar);
Tank (tno int, tname varchar, tcolor varchar, tvolume int);

Write SQL statements for the following queries:

(1) Find the names of tanks containing an orange fish heavier than 10 pounds. (5 pts)

```
Select T.tname
From Tank T, Fish F
Where T.tno = F.tno and F.fcolor = 'orange' and F.fweight > 10;
```

(2) Find the names of blue tanks that contain a species that also appears in a green tank. (5 pts)

```
Select T.tname
From Tank T, Fish F, Fish G, Tank Q
Where T.tno = F.tno and F.sno = G.sno and G.tno = Q.tno and Q.tcolor = 'green' and
T.tcolor = 'blue';
```

(3) Find the average fish weight for each species. (5 pts)

```
Select S.sno, avg(F.fweight) as avgWeight
From Species S, Fish F
Where S.sno = F.sno
Group by S.sno;
```

(4) Find the average tank volume by species, for those species that appear in two or more tanks. (5 pts)

```
Select S.sno, avg(T.tvolume) as avgVol
From Species S, Tank T
Where exists
(Select *
 From Fish F
 Where S.sno = F.sno and F.tno = T.tno)
Group by S.sno
Having count(*) >= 2;
```

Problem 5 SQL (20 points)

Consider the following schema

People (*ssn:* integer, *name:* string, *phone:* string, *city:* string)
Purchase (*buyer-ssn:* integer, *seller-ssn:* integer, *store:* string, *pid:* integer)
Product (*pid:* integer, *name:* string, *price:* real, *category:* string, *cid:* integer)
Company (*cid:* integer, *name:* string, *stock-price:* real, *country:* string)

Write the following queries in SQL.

(1) Find the names, stock prices, and countries of companies that have sold products to people living in Auburn but not to people living in Opelika. (10 pts)

```
Select Company.name, stock-price, country
From Company, Product, Purchase, Person
Where Company.cid = Product.cid AND Product.pid = Purchase.pid AND buyer-ssn
= ssn AND city = "Auburn"
EXCEPT
Select Company.name, stock-price, country
From Company, Product, Purchase, Person
Where Company.cid = Product.cid AND Product.pid = Purchase.pid AND buyer-ssn
= ssn AND city = "Opelika"
```

(2) Lists the names of all companies that have sold at least two different products (that is, products with different *pids*). (10 pts)

```
Select Company.name
From Company, Product, Purchase
Where Company.cid = Product.cid AND Product.pid = Purchase.pid
Group By Company.name
Having Count (Distinct Product.pid) >= 2
```