Final

Q1. What is the difference between a DBMS and a file system?

Q2. What are the levels of abstractions for a DBMS.

Q3. What are the SQL languages for creating, modifying and querying a database?

Q4. What is the ACID property for transactions?

Q5. A company database needs to store information about employees (identified by *ssn*, with *name*, *salary*, *age*, and *phone* as attributes), departments (identified by *dno*, with *dname* and *budget* as attributes), and children of employees (with *name* and *age* as attributes). Employees work in departments; each department is managed by an employee; a child must be identified uniquely by name when the parent (who is an employee; assume that only one parent works for the company) is known. We are not interested in information about a child once the parent leaves the company. Show the SQL to create the database. Identify primary keys and foreign keys.

Q7. Write example statements to put a tuple in each table.

Q8. Write relational algebra and SQL queries to answer the following questions.

a) Who is the employee with the most children?

b) What is the phone number of the oldest employee?

c) Who is the employee who has the biggest difference between their age and one of their kids age.

d) What is the department with the most employees?

e) Who is the employee that manages the most departments?

f) Print the employees sorted by salary, starting with the employee with the highest salary.

g) What is the employee that manages the department with the highest budget?

h) What are the names of parents that have a kid named Joe in the company?

Q9 Is the following schedule conflict-serializable: r1(X), w1(Y), w2(Y), r2(Y), w1(X), r1(Y)? Draw the precedence graph to show.

Q10 Explain how a scheduler that follows the two-phase locking protocol will process the schedule from Question 9.

create table if not exists Employee (

ssn integer primary key,

name varchar(30),

salary real,

age integer,

phone varchar(20),

dno integer);

create table if not exists Department(

dno integer primary key,

dname varchar(30),

budget real,

managerSSN integer,

foreign key (managerSSN) references Employee(ssn));

create table if not exists Children(

name varchar(30),

parentSSN integer,

age integer,

foreign key (parentSSN) references Employee(ssn) on DELETE CASCADE,

primary key (name,parentSSN)

);

alter table Employee add

foreign key (dno) references Department(dno);

insert into Employee values (1,'Val',30000,44,'234322',null);

insert into Department values (1,'CS',555444,1);

update Employee

set dno =1

where ssn = 1;

insert into Children values ('Suzan',1,3);

a)

select \*

from Employee e

where (select count(\*) from children c where c.parentSSN=e.SSN) >= ALL (select count(\*) from Children group by parentSSN);

b)

select phone

from Employee

where age = (select max(age) from Employee);

c)

select name

from Employee e

where (select max(e2.age-c2.age) from Employee e2, Children c2 where c2.parentSSN=e2.ssn) in

(select e.age -c.age as maxDif from Children c where c.parentSSN = e.ssn);

d)

select d.dname

from Department d

where (select count(\*) from Employee e where e.dno = d.dno) >= all

(select count(\*) from Employee e2,Department d2 where e2.dno = d2.dno group by d2.dno);

e)

select e.name

from Employee e

where (select count(\*) from Department d where d.managerSSN=e.SSN) >= all

(select count(\*) from Employee e2,Department d2 where d2.managerSSN=e2.SSN group by e2.ssn);

f)

select \*

from Employee

order by salary desc;

g)

select name

from Employee e

where (select max(budget) from Department d where d.managerSSn = e.ssn) =

(select max(budget) from department);

h)

select name

from Employee e

where exists (select \* from Children c where c.parentSSn = e.ssn and c.name='Joe');