14 Feb 2019

Indian Institute of Information Technology, Sri City, Chittoor

Name of the Exam: Database Management Systems

Duration: 1.5 hr Max. Marks: 20

Roll No.:____________________________________Room No.: _______________Seat No.: __________

Name: _____________________________________Invigilator's Signature: _______________________

Instructions: 1. All questions have to be answered in the box space provided only.
2. You have to do rough work in the question paper if required in the last

Q1. Multiple Choice Questions. Write the answer for the following questions in the space provided. Only one answer to be selected. (1x5=5 marks)

i) Database __________ which is the logical design of the database, and the database _______ which is a snapshot of the data in the database at a given instant in time.

a) Instance, Schema
b) Relation, Schema
c) Relation, Domain
d) Schema, Instance

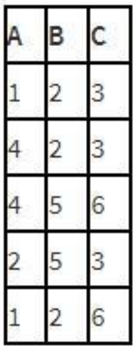
Ans:d

ii) The result which operation contains all pairs of tuples from the two relations, regardless of whether their attribute values match.

a) Join
b) Cartesian product
c) Intersection
d) Set difference

Ans:

iii) Suppose relation R(A,B,C) has the following tuples. (See Fig 1). Compute the projection $\pi_{C,B}(R)$. Which of the following tuples is in the result?



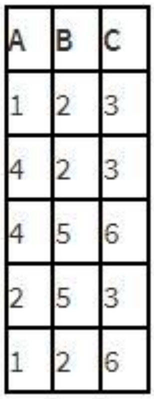
A	B	C
1	2	3
4	2	3
4	5	6
2	5	3
1	2	6

Figure 1

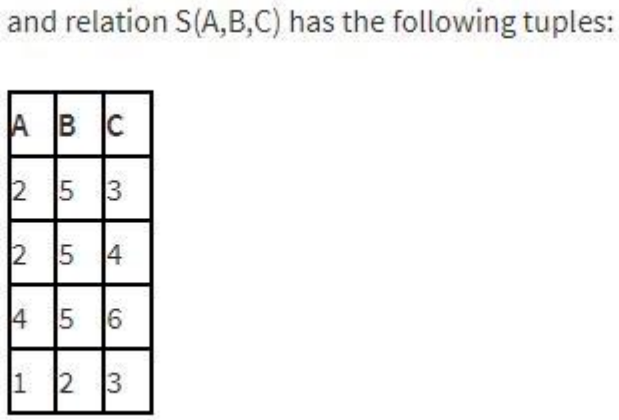
a. (2,5)
b. (3,2)
c. (5,6)
d. (2,6)

b
Ans:

iv) Suppose relation R(A,B,C) has the following tuples. (See Fig 2).



A	B	C
1	2	3
4	2	3
4	5	6
2	5	3
1	2	6



and relation S(A,B,C) has the following tuples:

A	B	C
2	5	3
2	5	4
4	5	6
1	2	3

Figure 2

Compute the intersection of the relations R and S. Which of the following tuples is in the result?

a. (4,2,3)
b. (1,2,3)
c. (1,2,6)
d. (2,4,3)

Ans:b

v) Consider the following schema for a courses database:

- department(did, dname, location)
- student(sid, sname, did, age)
- course(cid, cname, time, room)
- enrolled(sid, cid)

Which of the following SQL queries will count the number of departments with no students taking the course 'Databases'.

A. SELECT COUNT(d.did) FROM department d WHERE d.did IN (SELECT s.did FROM student s WHERE s.sid IN (SELECT e.sid FROM enrolled e, course c WHERE e.cid = c.cid AND c.cname = 'Databases'));

B. SELECT COUNT(DISTINCT s.did) FROM student s WHERE s.sid NOT IN (SELECT e.sid FROM enrolled e, course c WHERE e.cid = c.cid AND c.cname = 'Databases');

C. SELECT COUNT(DISTINCT d.did) FROM department d WHERE d.did NOT IN (SELECT s.did FROM enrolled e, course c, student s WHERE e.cid = c.cid
AND c.cname = 'Databases' AND e.sid = s.sid);

D. SELECT COUNT(d.did) FROM department d, student s, course c, enrolled e WHERE e.cid = c.cid AND c.cname='Database' AND e.sid != s.sid AND d.did = s.did;

Ans: □

B is incorrect because it counts the number of departments with students not taking the course 'Databases'

Q2. Subjective Questions. Answer the following questions in the space provided only. (3 marks)

i) How Facebook or other popular social networking platform stores the data records. Explain the entire workflow/pipeline in using them.

(For the databases, 0.5 marks and for the pipeline diagram , 0.5 marks)

•The king of scalability, Google, uses BigTable (developed by Google).
•Facebook uses Hive (Data warehouse for Hadoop, supports

tables and a variant of SQL called hiveQL) and Cassandra (Multi-dimensional, distributed key-value store) for Facebook's private

messaging.
•Yahoo uses modified PostgreSQL.
•YouTube uses MySQL but they are moving to Google's BigTable.
•Myspace uses SQL Server.
•Twitter and Wikipedia uses MySQL.
•Microsoft uses SQL Server, which is very obvious.
•Flickr uses MySQL.

ii) Define the 3 types of explicit constraints with examples in detail in DBMS.

(for the 3 explicit constraints , 0.5 marks and for the examples, 0.5 marks)

Constraints determine which values are permissible and which are not in the database.
They are of three main types:
1. Inherent or Implicit Constraints: These are based on the data model itself. (E.g., relational model does not allow a list as a value for any attribute)
2. Schema-based or Explicit Constraints: They are expressed in the schema by using the facilities provided by the model. (E.g., max. cardinality ratio constraint in the ER model)
3. Application based or semantic constraints: These are beyond the expressive power of the model and must be specified and enforced by the application programs.

There are three main types of (explicit schema-based) constraints that can be expressed in the

relational model:
□ Key constraints
□ Entity integrity constraints
□ Referential integrity constraints

(ALONG WITH EXAMPLES)

iii) Give 2 advantages of DBMS systems over file systems. Also name the 2 types of users in a typical DBMS environment and explain their roles.

(For the advantages, 0.5 marks and for the users, 0.5 marks)

□ Controlling redundancy in data storage and in development and maintenance efforts.
□ Sharing of data among multiple users.
□ Restricting unauthorized access to data. Only the DBA staff uses privileged commands and facilities.
□ Providing persistent storage for program Objects
□ E.g., Object-oriented DBMSs make program objects persistent
□ Providing storage structures (e.g. indexes) for efficient query processing Providing optimization of queries for efficient processing
□ Providing backup and recovery services
□ Providing multiple interfaces to different classes of users
□ Representing complex relationships among data
□ Enforcing integrity constraints on the database
□ Drawing inferences and actions from the stored data using deductive and active rules and triggers

Users may be divided into
□ Those who actually use and control the database content, and those who design, develop and maintain database applications (called "Actors on the Scene"), and
□ Those who design and develop the DBMS software and related tools, and the computer systems operators (called "Workers Behind the Scene").

Q.3: ER Diagram (3 marks)

Draw a complete ER diagram and its extensions-generalization and specialization cases (EER) in the same diagram for Hospital Management System. Show disjoint/overlapping cases clearly. Draw the notations very carefully in the diagram.

>For not mentioning primary keys, cardinality, disjoint/overlapping conditions -0.5 each
>for, inadequate number of entities or attributes negative marks are awarded subjectively.
>If the relations among the entities are not clear, -0.5 to -1 mark.
>usage of proper symbols and clarity of the diagram are also taken into consideration.

Q. 4: Use Case Diagram (3 marks)

Draw a neat use case diagram with all the relevant notations and symbols showing the workflow of an E-commerce (e.g. Amazon, Flipkart etc.) website. Also show the include, exclude relationships very clearly.

Basis of mark cutting:
□ If the representation symbols are wrongly drawn or not drawn the required symbols cut 0.5 to 1 marks.
□ If include, exclude are not given cut 0.5 marks.
□ If other fundamental mistakes are there cut 0.5 to 1 marks based on the mistake.
Inadequate number of usecases.

Q.5: SQL Queries (3+3=6 marks)

i) Consider the following employee database, where the primary keys are underlined. Give an expression in SQL for each of the following queries. (0.5x6=3mark)

Employee (employee_name, street, city)

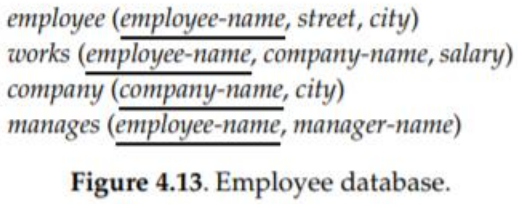
Works(employee_name, company_name, salary)

Company(company_name, city)

Manages(employee_name, manager_name)

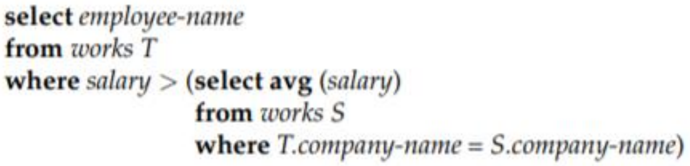
(Each query is awarded 0.5 marks if correct)

i)a) The following solution assumes that all people work for at most one company.



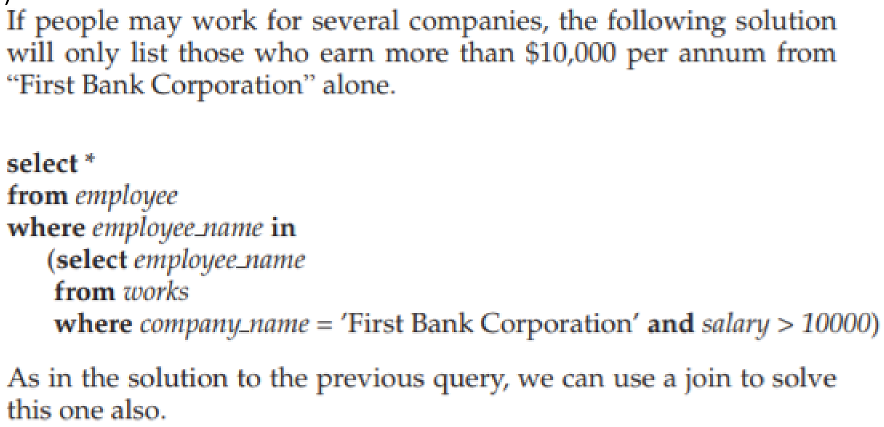
employee (employee-name, street, city)
works (employee-name, company-name, salary)
company (company-name, city)
manages (employee-name, manager-name)

Figure 4.13. Employee database.



```
select employee-name
from works T
where salary > (select avg (salary)
                from works S
                where T.company-name = S.company-name)
```

b)

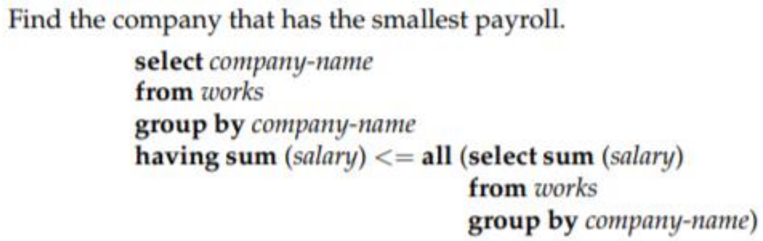


If people may work for several companies, the following solution will only list those who earn more than $10,000 per annum from "First Bank Corporation" alone.

```
select *
from employee
where employee_name in
    (select employee_name
     from works
     where company_name = 'First Bank Corporation' and salary > 10000)
```

As in the solution to the previous query, we can use a join to solve this one also.

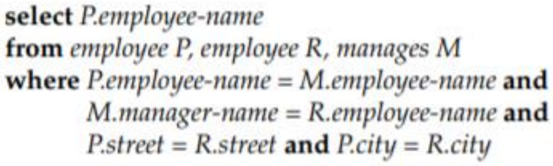
c)



Find the company that has the smallest payroll.

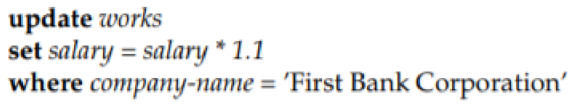
```
select company-name
from works
group by company-name
having sum (salary) <= all (select sum (salary)
                            from works
                            group by company-name)
```

d) Find all employees in the database who live in the same cities and on the same streets as do their managers.



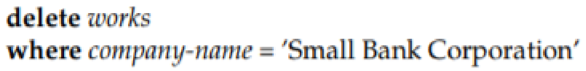
```
select P.employee-name
from employee P, employee R, manages M
where P.employee-name = M.employee-name and
      M.manager-name = R.employee-name and
      P.street = R.street and P.city = R.city
```

e) Give all employees of First Bank Corporation a 10-percent raise.



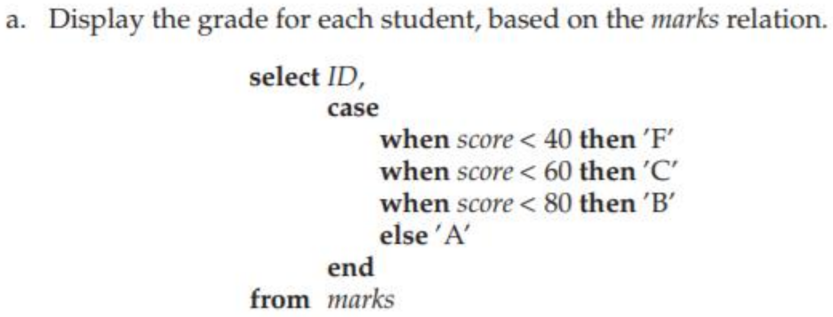
```
update works
set salary = salary * 1.1
where company-name = ’First Bank Corporation’
```

f) Delete all tuples in the works relation for employees of Small Bank Corporation.



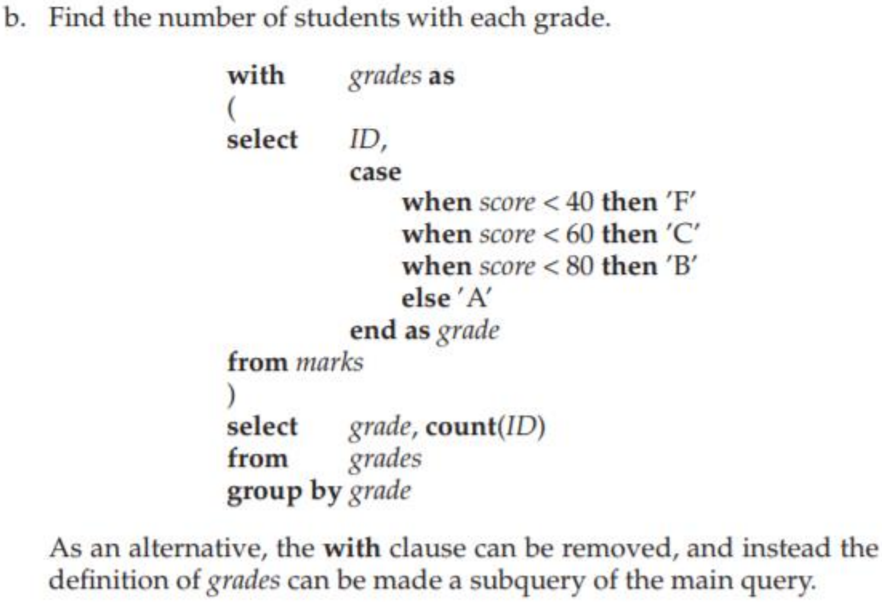
```
delete works
where company-name = ‘Small Bank Corporation’
```

ii) **(1x3=3mark)**



a. Display the grade for each student, based on the *marks* relation.

```
select ID,
       case
           when score < 40 then ’F’
           when score < 60 then ’C’
           when score < 80 then ’B’
           else ’A’
       end
from  marks
```

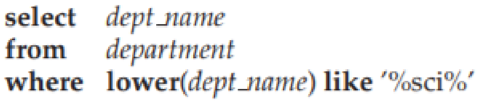


b. Find the number of students with each grade.

```
with      grades as
(
select    ID,
          case
              when score < 40 then ’F’
              when score < 60 then ’C’
              when score < 80 then ’B’
              else ’A’
          end as grade
from marks
)
select    grade, count(ID)
from      grades
group by grade
```

As an alternative, the **with** clause can be removed, and instead the definition of *grades* can be made a subquery of the main query.

c)



```
select   dept_name
from     department
where    lower(dept_name) like ’%sci%’
```