Mid Semester Examination – 2020

School of Computer Engineering

Kalinga Institute of Industrial Technology (KIIT) Deemed to be University

Subject: Database Management System

Time: 1.5 Hrs Full Marks: 20

(Answer any Four Questions including Question No. 1)

1. Answer the following questions briefly. [1 X 5]

a. Distinguish between Disjoint and Overlapping constraints?

Ans:

In a disjointness design constraint, an entity can belong to not more than one lower-level entity set. In overlapping generalizations, the same entity may belong to more than one lower-level entity sets.

Overlap Constraint -An overlap constraint determines whether or not two subclasses can contain the same entity. Disjoint constraints Disjoint Describes the relationship between members of the subclasses and constraint indicates whether it is possible for a member of a superclass to be a member of one, or more than one, subclass. The disjoint constraint only applies when a superclass has more than one subclass

b. Differentiate between Theta join and Equi join.

Ans:

Theta join combines tuples from different relations provided they satisfy the theta condition. The join condition is denoted by the symbol θ.

When Theta join uses only equality comparison operator, it is said to be equijoin.

c. Primary key is not same as unique key -Justify your answer.

Ans:

A primary key has the following functions:

- Each table must have one and only one primary key, not more than one.
- A primary key cannot contain NULL values.
- It may consist of one or more columns.
- All columns must be defined as NOT NULL.
- A primary key is clustered unique index by default.

A UNIQUE KEY has the following functions:

- A UNIQUE KEY constraint guarantees the uniqueness of the values.
- Multiple unique keys can be defined on a table.
- A column may contain a NULL value, but only one NULL value per column is allowed.
- A unique key may create a non-clustered index by default.

d. Consider the below two tables for the given question :

Write a SQL query to fetch employee names having a salary greater than or equal to 5000 and less than or equal 10000.

Table 1: EmployeeDetails

EmpId	FullName	ManagerId	DateOfJoining
121	John Snow	321	01/31/2014
321	Walter White	986	01/30/2015
421	Kuldeep Rana	876	27/11/2016

Table-2 EmployeeSalary

EmpId	Project	Salary
121	P1	8000
321	P2	1000
421	P1	12000

Ans:

```
SELECT FullName FROM EmployeeDetails NATURAL JOIN EmployeeSalary
WHERE Salary>=5000 AND Salary<=10000;
```

OR

```
SELECT FullName FROM EmployeeDetails, EmployeeSalary WHERE
EmployeeDetails.EmpId=EmployeeSalary.EmpId AND Salary>=5000 AND
Salary<=10000;
```

e. Given the relations

Employee (eid, name, salary, deptno) and
Department (deptno, deptname, address)
Which of the following queries cannot be expressed using the basic relational algebra operations (U, -, x, π, σ, p)?
(a) Department address of every employee
(b) Employees whose name is the same as their department name
(c) The sum of all employees' salaries
(d) All employees of a given department.

Ans:
(c) The sum of all employees' salaries

---

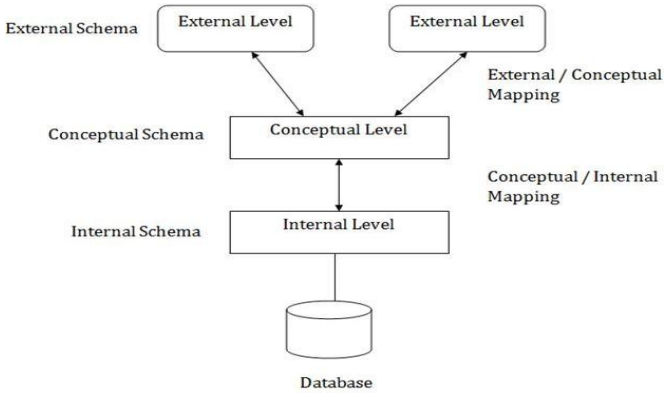
2. Describe the three-schema architecture. Why do we need mappings between schema levels? Also, differentiate between logical data independence and physical data independence. [5]

Ans

Scheme: *Three Schema Architecture- 2 marks, Mapping- 1 mark, Difference between logical and physical data independence: 2 marks*.

Three Schema Architecture: This framework is used to describe the structure of a specific database system. The three schema architecture is also used to separate the user applications and physical database. The three schema architecture contains three-levels. It breaks the database down into three different categories.

The three-schema architecture is as follows:



- Internal or physical: The internal level has an internal schema which describes the physical storage structure of the database. The internal schema is also known as a physical schema. It uses the physical data model. It is used to define that how the data will be stored in a block. The physical level is used to describe complex low-level data structures in detail.
- Conceptual or logical: The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level. The conceptual schema describes the structure of the whole database. The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data. In the conceptual level, internal details such as an implementation of the data structure are hidden. Programmers and database administrators work at this level.
- External or user view: At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database. An external schema is also known as view schema. Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group. The view schema describes the end user interaction with database systems.

Mapping is the process of transforming requests and results between levels. This is needed for visualization and schema matching. The mappings between schema levels help in the different types of transformation. Programs refer to an external schema, and are mapped by the DBMS to the Internal schema for execution.

PHYSICAL DATA INDEPENDENCE: Physical storage structure or devices can be changed without affecting the existing conceptual schema. Modifications are performed to improve performance. It is not difficult because, it is only to replace the data from one device to another device. No need to change the data only changes the location of data. It provides independence and immunity to conceptual and external schema.

LOGICAL DATA INDEPENDENCE: Conceptual schema can be changed without affecting the existing external schema. Conceptual schema can be changed/modified when structure of database is altered. It is very difficult because, it has to be seen that the relationship must not changing, data is not

getting lost and some other factors. It only provides immunity to external schema and application program.

3.(a) What are advantages of DBMS over traditional file based systems? [3]

Ans

Controlling of Redundancy: Data redundancy refers to the duplication of data (i.e. storing same data multiple times).In a database system, by having a centralized database and centralized control of data by the DBA the unnecessary duplication of data is avoided. It also eliminates the extra time for processing the large volume of data. It results in saving the storage space.

Improved Data Sharing: DBMS allows a user to share the data in any number of application programs.

Data Integrity: Integrity means that the data in the database is accurate. Centralized control of the data helps in permitting the administrator to define integrity constraints to the data in the database. For example: in customer database, we can enforce integrity that it must accept the customer only from Cuttack and Bhubaneswar city.

Security: Having complete authority over the operational data, enables the DBA in ensuring that the only mean of access to the database is through proper channels. The DBA can define authorization checks to be carried out whenever access to sensitive data is attempted.

Efficient Data Access: In a database system, the data is managed by the DBMS and all access to the data is through the DBMS providing a key to effective data processing.

Data Consistency: By eliminating data redundancy, we greatly reduce the opportunities for inconsistency. For example: a customer address is stored only once, we cannot have disagreement to the stored values. Also updating data values is greatly simplified when each value is stored in one place only. Finally, we avoid the wasted storage that results from redundant data storage.

Data Independence: In a database system, the database management system provides the interface between the application programs and the data. When changes are made to the data representation, the metadata obtained by the DBMS is changed but the DBMS is continues to provide the data to application program in the previously used way. The DBMS handles the task of transformation of data wherever necessary.

Enforcing Integrity Constraints

Providing Backup &Recovery

Providing Storage Structures for efficient query processing

(b) Discuss the roles of Data Analyst and Application Programmer. [2]

Ans

Data Analyst: A database analyst deals with database technologies that warehouse information in very specific ways. A database analyst is part of conventional corporate IT teams that maintain data assets through very specific research and activities. A Data Analysts deliver value to their companies by taking information about specific topics and then interprets, analyzes, and presents findings in comprehensive reports.

Application Programmer: They are computer professionals who write application programs to access data from the database. Application programmers can use different tools to develop user interfaces

4. Consider the following schema to write queries in relational Algebra: [5]

Sailor(sid, sname, age, rating)
Boats(bid, bname, bcolor)
Reserves(sid,bid,day)

a. Find id of the boats reserved by sailor with id 567.
b. Find the names of the sailors who reserved 'red' boats.
c. Find the boats which have at least two reservations by different sailors.
d. Find the sailors name who have ratings of 5 and their name started with 'P'.
e. Find the name of Sailors who have reserved boat id 100 on Monday.

Ans

a. $\prod_{bid}(\sigma_{sid=567}(\text{Reserves}))$

b. $\prod_{sname}(\sigma_{bcolor='red'}(\text{Sailor} \bowtie \text{Reserves} \bowtie \text{Boat}))$

c. $\prod_{\text{Reserves.bid}}(\sigma_{\text{Reserves.bid=Reserves2.bid} \wedge \text{Reserves.sid} \neq \text{Reserves2.sid}}(\text{Reserves} \times \rho_{\text{Reservse2}}(\text{Reserves})))$

d. $\prod_{sname}(\sigma_{rating=5 \wedge \text{ sname LIKE 'P\%'}}(\text{Sailor}))$

e. $\prod_{sname}(\sigma_{bid=100 \wedge day='Monday'}(\text{Sailor} \bowtie \text{Reserves}))$

5. A University wants to maintain a database to store the information about their students, courses, lecturers, cabins, and subjects. The university hired a leading consultancy firm for the project. After a detailed analysis, the development team came up with the following design:

I. For students, the database stores the details like name, roll, dob, age, hobby, and address. The address consists of door_no, street_name, city, state, and pin. A student can register for many courses where each course has a course_id, and course_name. Similarly, a course can be taken by many students.

II. Each lecturer, with a lect_id, lect_name, email, and contact_no, is assigned with a cabin, with cabin_no, floor. However, a cabin can be allotted to 2 lecturers on sharing basis.

III. A lecturer can teach only one subject, where the subject has a subject_id, subject_name, and duration. However, a subject can be taught by many lecturers.

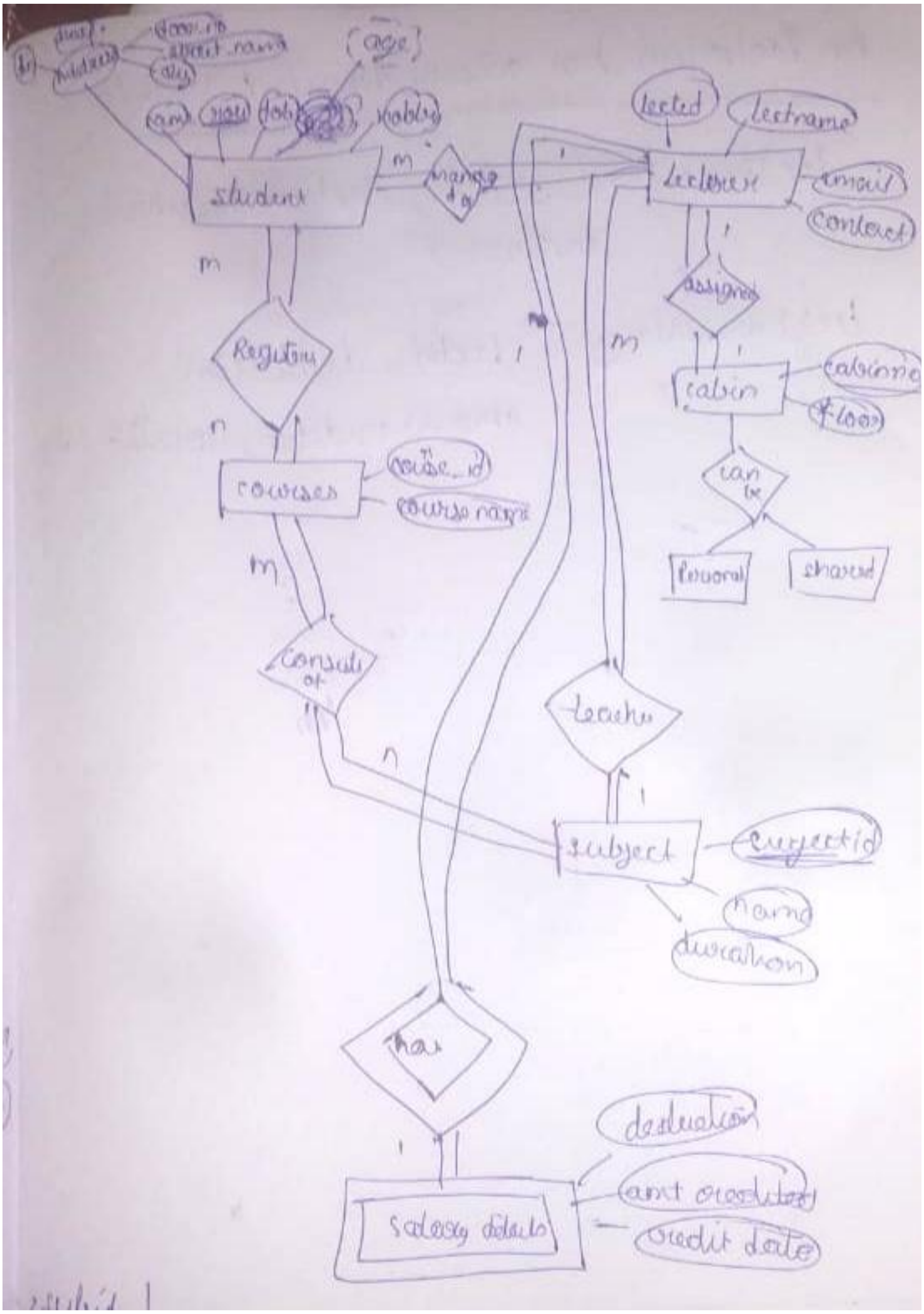
IV. Each lecturer has a salary history which contains amt_credited, credit_date, and deductions.

V. Each Course consists of many subjects , however one subject can belongs to multiple courses.

VI. The Lecturers are managing the students.

A. Draw ER diagram for the above scenario.
B. Convert the ER diagram into relational schema

Ans



student
lectid
lectname
email
contact
courses
course name
cabin
shared
subject
name
duration
salary details
credit date

For student Register course

student (roll, dob, name, age, hobby, street, door, state)

Register (roll, courseid)

courses (course-id, coursename)

For lecturers and cabin 1 table

lecturer_assigned cabin (lectid, lectname, email, contact, cabin no, floor)

For specialisation of cabin 2 tables

Personal (cabin no, floor)
shared (cabino, floor)

for lecturer teaches subject (1 table)

lecturer_teaches subjects (lecid, lectname, email, contact, subid, name, duration)

For courses consists of subject (2 tables)
(1 ignored as course is repeating)

Subject (subid, name, duration), consists(courseid, ...

For lecturer has salary details (2 tables)

lecturer has (lectid, lectname, email, contact)

lecthas_salary (lectid, deduction, amount credited, credit date

6. Consider a schema with two relations, R(A, B) and S(B, C), where all values are integers. Make no assumptions about keys. Consider the following three relational algebra expressions: [5]

a. $\pi_{A,C}(R \bowtie \sigma_{B=1} S)$

b. $\pi_{A}(\sigma_{B=1} R) \times \pi_{C}(\sigma_{B=1} S)$

c. $\pi_{A,C}(\pi_{A} R \times \sigma_{B=1} S)$

Two of the three expressions are equivalent (i.e., produce the same answer on all databases), while one of them can produce a different answer. Which query can produce a different answer? Give the simplest database instance you can think of where a different answer is produced.

Ans: a and b are same but c will produce different result, because,
suppose, R(A,B) = {(A1,1),(A2,2)} and S(B,C) = {(1,C1)}

a. $\prod_{A,C}\left(R \; NJ \; \sigma_{B=1} S\right)$

$$= \prod_{A,C}\left(\begin{Bmatrix} A1 & 1 \\ A2 & 2 \end{Bmatrix} NJ \; \{1 \;\; C1\}\right)$$

$$= \prod_{A,C}\left(A1 \;\; 1 \;\; C1\right)$$

$$= A1 \;\; C1$$

b. $\prod_{A}\left(\sigma_{B=1} R\right) \times \prod_{C}\left(\sigma_{B=1} S\right)$

$$= \{A1\} \times \{C1\}$$

$$= A1 \;\; C1$$

Same

c. $\prod_{A,C}\left(\prod_{A} R \times \sigma_{B=1} S\right)$

$$= \prod_{A,C}\left(\begin{Bmatrix} A1 \\ A2 \end{Bmatrix} \times \{1 \;\; C1\}\right)$$

$$= \prod_{A,C}\left(\begin{matrix} A1 & 1 & C1 \\ A2 & 1 & C1 \end{matrix}\right)$$

$$= \begin{Bmatrix} A1 & C1 \\ A2 & C1 \end{Bmatrix}$$