Mid Semester Examination – 2019
School of Computer Engineering
Kalinga Institute of Industrial Technology (KIIT) Deemed to be University
Subject: Database Management System (CS-2004)

Time: 1½ Hrs Full Marks: 20

(Answer any Four Questions including Question No. 1)

1. 1×5

a) Difference between Primary key, super key, candidate key.
Super Key: An attribute or set of attributes that uniquely identifies a tuple within a relation.
Candidate key: Super keys for which no proper subset is a super key within the relation. In other words candidate keys are minimal super keys.
Primary key: The candidate key that is selected to identify tuples uniquely within the relation, the candidate keys which are not selected as PKs are called "Alternate keys".

b) Explain the importance of logical designing before the physical designing of database.
Logical database design is the process of deciding how to arrange the attributes of the entities in a given business environment into database structures, such as the tables of a relational database.
Physical database design translates the logical data model into a set of SQL statements that define the database. For relational database systems, it is relatively easy to translate from a logical data model into a physical database. Rules for translation: Entities become tables in the physical database.

A logical data model should be used as the blueprint for designing and creating a physical database. But the physical database cannot be created properly with a simple logical to physical mapping. Many physical design decisions need to be made by the DBA before implementing physical database structures.

c) Write a SQL statement to add a composite primary key to an existing table.
ALTER table TABLE_NAME
ALTER TABLE tablename ADD CONSTRAINT constraintname PRIMARY KEY (col1, col2…);

d) Differentiate between Theta join and Equi join.
Theta join allows for arbitrary comparison relationships (such as ≥).
Equijoin is a theta join using the equality operator.
Example:

e) Differentiate between Cartesian product and join operations.
Cartesian product combines each tuple of one relation with all the tuples of the other relation. Tf there are 5 tuples in first relations and 5 rows in second relation, then there will be 5×5=25 tuples in Cartesian product of these two relations. Denoted by R×S, where R and S are two relations.

JOIN is used to combine related tuples from two relations depending on join condition. JOIN allows to evaluate a join condition between the attributes of

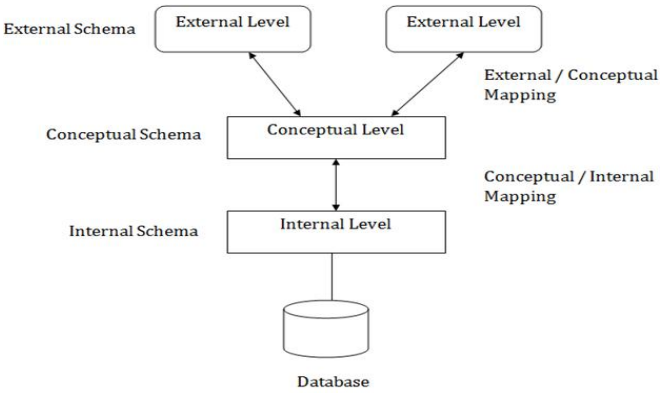
two relations on which the join is undertaken. The notation used is R $JOIN_{join\ condition}$ S, which is a subset of its cross product.

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

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 conceptual schema. Modification 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 change the location of data. It provides independence and immunity to conceptual and external schema.

LOGICAL DATA INDEPENDENCE: Conceptual schema can be changed without affecting 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 provide immunity to external schema and application program.

3. Write short notes on unary and binary operations in relational algebra with suitable. 5

Scheme: *Any five operations with example 1 mark each*.

Relational Algebra: A query language is a language in which user requests information from the database. it can be categorized as either procedural or nonprocedural. In a procedural language the user instructs the system to do a sequence of operations on database to compute the desired result. In nonprocedural language the user describes the desired information without giving a specific procedure for obtaining that information. The relational algebra is a procedural query language. It consists of a set of operations that take one or two relations as input and produces a new relation as output.

Fundamental Operations

- SELECT
- PROJECT
- UNION
- SET DIFFERENCE
- CARTESIAN PRODUCT
- RENAME

Select and project operations are unary operation as they operate on a single relation.Union, set difference, Cartesian product and rename operations are binary operations as they operate on pairs of relations.

Other Operations

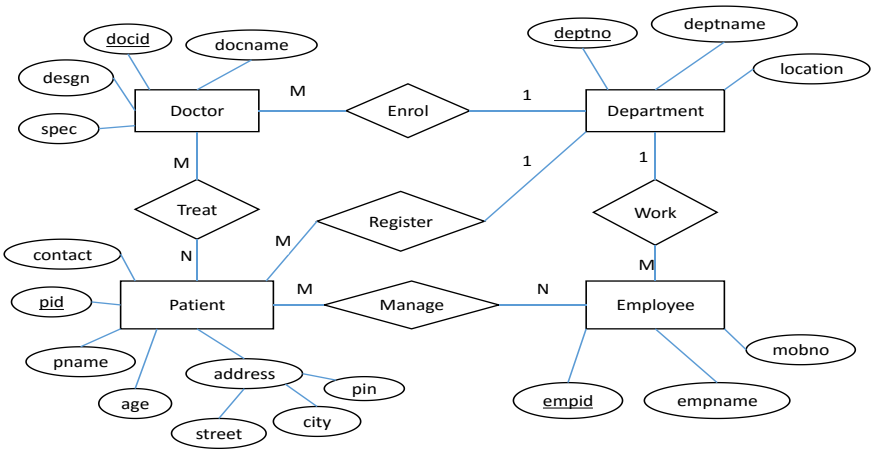
- SET INTERSECTION
- NATURAL JOIN
- DIVISION
- ASSIGNMENT

Examples

4. Draw the entity-relationship diagram for the hospital as given below: 5

The database maintains the details of doctors (identified by unique docid along with docname, desgn and specialization) who are enrolled to departments; one doctor can enrol to one department only. Employees (identified by unique empid along with empname and mobno) are working in the departments; one employee can work in a single department only. Each department is identified through unique deptno along with deptname and location. There is a registration process required for all patients to a department before they treated by any doctor. Patient details must contain

unique pid, pname, address (can be decomposed to street, city and pin), age and contactno. Employees are managing the patients. One patient can be treated by multiple doctors; also one doctor can treat multiple patients.
Make necessary assumptions. Also, convert the E/R diagram into relational schema.
Scheme: *ER diagram 3 marks, relational Schema 2 marks.*



Relational Schema

Doctor (docid, docname, desgn, spec, enroll_deptno)
(fk)

Department (deptno, deptname, location)

Employee (empid, empname, mobno, working_deptno)
(fk)

Patient (pid, pname, street, city, pin, age, contact, deptno)
(fk)

Treat (docid, pid)
(fk) (fk)

Manage (empid, pid)
(fk) (fk)

5. Employee (empno, ename, dept, doj, mob) 5
Customer (cno, cname, city, mobno, interest)
Deal (empno, cno, date, category)
Solve the following queries by using relational algebra / sql.
a) Display the customers' names staying in 'BBSR'.
b) Find the employees' names dealing in 'Fashion' category.
c) Find the customers' names, who are dealing with all the customers.
d) Find the customers, who are dealing the customers with 'Designer Handbags' interests.
e) List the customers' names dealing to the 'BBSR' customers in the year 2018.
Scheme: *1 mark to each correct answer (either in algebra or in SQL)*

ALGEBRA

$\pi_{cname}(\sigma_{city='BBSR'}(Customer))$

$\pi_{ename}(\sigma_{category='Fashion'}(Employee \bowtie Deal))$

$\pi_{ename,cno}(Employee \bowtie Deal) \div \pi_{cno}(Customer)$

$\pi_{Employee.empno,ename,dept,doj,mob}(\sigma_{interest='DesignerHandbags'}(Employee \bowtie Deal \bowtie$

$\pi_{ename}(\sigma_{city='BBSR' \wedge date\ LIKE\ '\%2018'}(Employee \bowtie Deal \bowtie Customer))$

SQL

a. SELECT cname FROM Customer WHERE city='BBSR';
b. SELECT ename FROM Employee, Deal WHERE Employee.empno=Deal.empno AND category='Fashion';
c. SELECT s1.ename, s1.cno FROM Employee NATURAL JOIN Deal AS s1 WHERE NOT EXISTS ((SELECT s2.cno FROM Customer AS s2) EXCEPT (SELECT s3.cno FROM Employee NATURAL JOIN Deal AS s3 WHERE s3.ename=s1.ename));

d. SELECT Employee.empno, ename, dept, doj, mob FROM Employee, Deal, Customer WHERE Employee.empno=Deal.empno AND Deal.cno=Customer.cno AND interest='Designer Handbags';
e. SELECT ename FROM Employee, Deal, Customer WHERE Employee.empno=Deal.empno AND Deal.cno=Customer.cno AND city='BBSR' AND date LIKE '%2018';

If the faculty members find any other way of finding solution, also can award marks.