# R-22

Sub code:- R22IT403PC

(1)

# DBMS Key

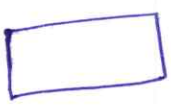
## PART-A

1 @) Define DBMS? What are the goals of DBMS? (2M)

Defn:- Data base management system is a s/w that allows users to define, create, store, retrieve, and manage data in the database efficiently.

Goals:-
* Data Independence
* Data security
* Data integrity
* Data consistency.
* Efficient Data Access.
* Concurrency Control.

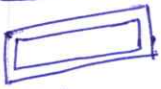
(b) List various symbols used in E-R Diagrams. (1M)

Ans:- 1) Rectangle  Represents Entities

2) Ellipse  Represents Attribute

3) Diamond  Represents relationships among Entities.

4) Double Ellipse  - represents multivalued Attribute

5) Double Rectangle  Represents weak Entity.

6) Double Diamond  Represents relation b/w Strong Entity and Weak Entity.

(c) Write a note on Domain Relational Calculus.

DRC in DBMS is a non-Procedural, declarative Query language that defines what data to retrieve not how. (1M)

d) Distinguish b/w Super key and Candidate Key. 1(m)

| Super key | Candidate key |
|---|---|
| 1) set of one & more attributes which can uniquely identify a row in a table | subset of super key.   |
| 2) All super keys can't be candidate keys | But all candidate keys are super keys. |
| 3) super key attributes can contain null values | It also contains null values |

e) What is schema refinement. 1(m)

Process of organizing database tables, through decompositions to eliminate data redundancy Anomalies.

f) Discuss about the problems caused by redundancy.

Problems caused by Redundency are called Anomalies. 1(m)

* Insertion Anomaly
* Deletion Anomaly
* Updation Anomaly.

g) What is transaction? Explain its states? (1m)

A transaction refers to a sequence of one & more actions performed on the database.

States :- Active, Partially committed, Committed, Failed, Aborted Terminated.

h) Explain about Durability of transaction

It ensures that once a transaction successfully commits, its changes are permanent and survive system failure. 1(m)

i) Primary and secondary indexing?

Primary indexing sorts data by unique primary key.

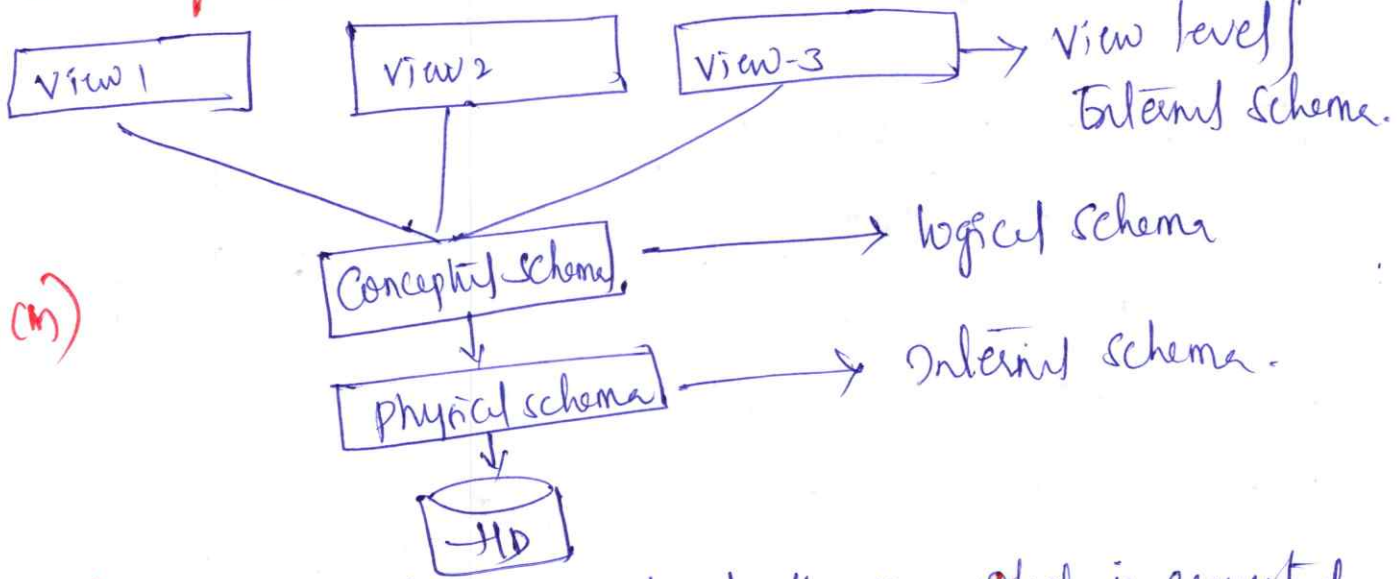
Secondary indexing creates additional indexes on non primary key columns. 1(m)

i) D/f b/w ISAM and B+ trees indexes. 1(m)

ISAM is a static index structure. While B+ trees is dynamic, self balancing structure.

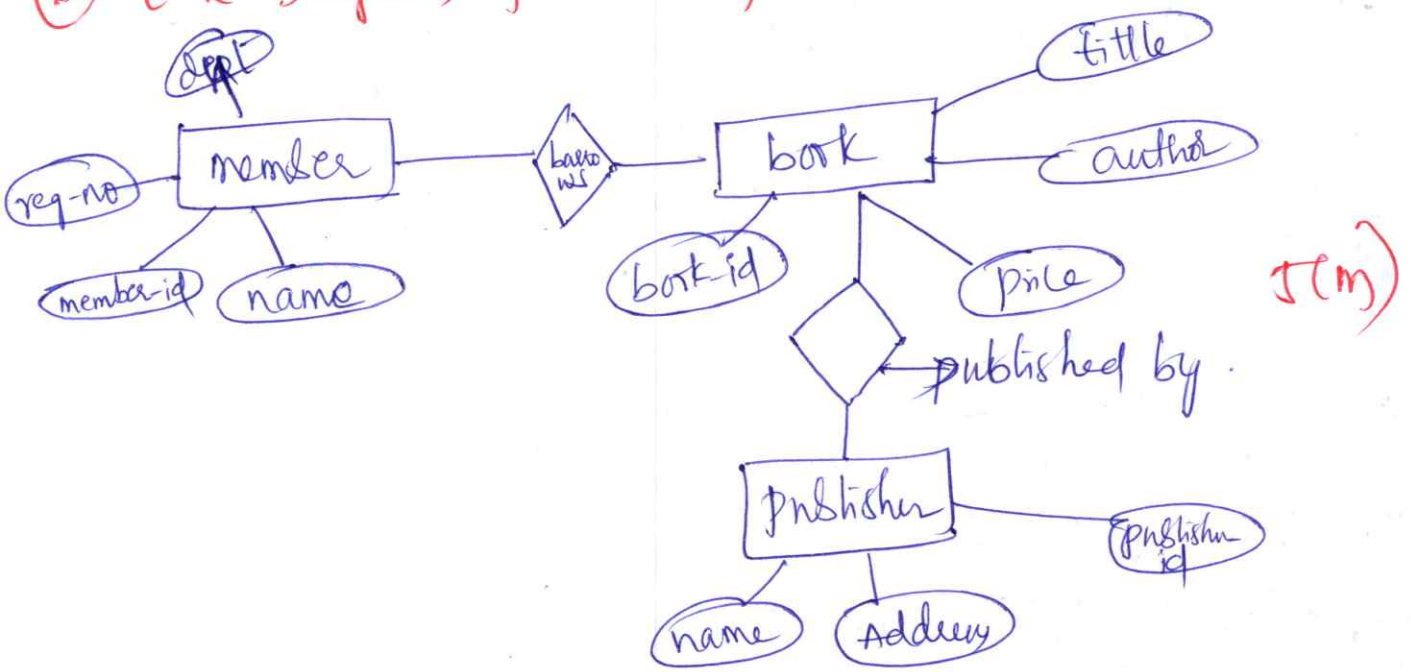
## PART-B

2(a) levels of Abstraction in DBMS. 3(m)




View schema :- Information visible to the user which is requested.
logical schema :- Entire information is displayed. 2(m)
Internal schema :- Explains how the data is stored Actually.

2(b) E-R Diagram for library management system. 5(m)



3 (a) Database languages DDL and DML

3(M)

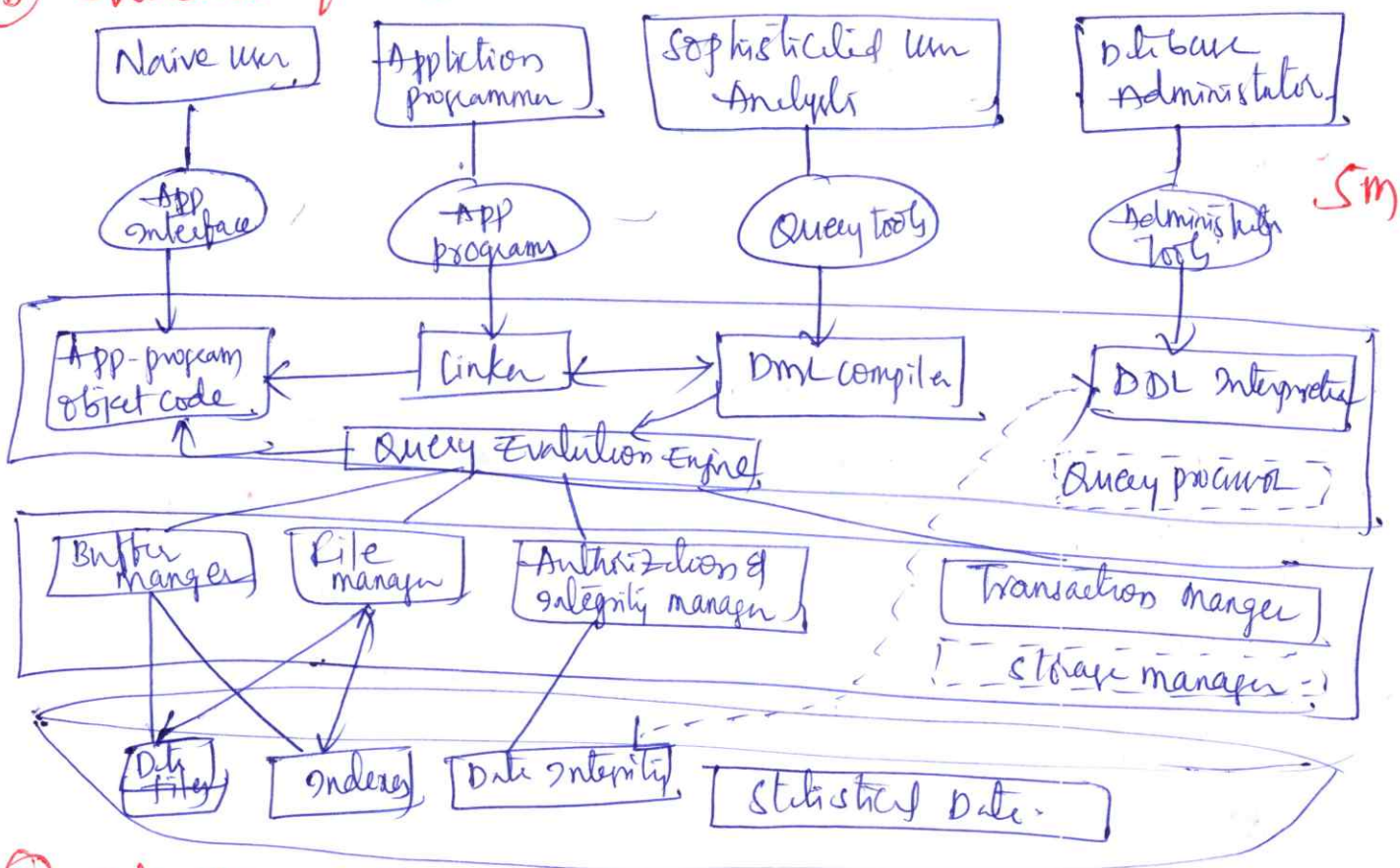
DDL (Data Definition language) — 2(M)

1) Create (Create its object)
2) Alter (modify existing)
3) Drop (permanently Delete)
4) Truncate (Remove All records)
5) Rename (change name)

DML (Data manipulation language)

1) Insert (Add new Records)
2) Update (modify existing)
3) Delete (remove existing record)
4) Select (retrieve data from one & more tables).

3 (b) Structure of DBMS.

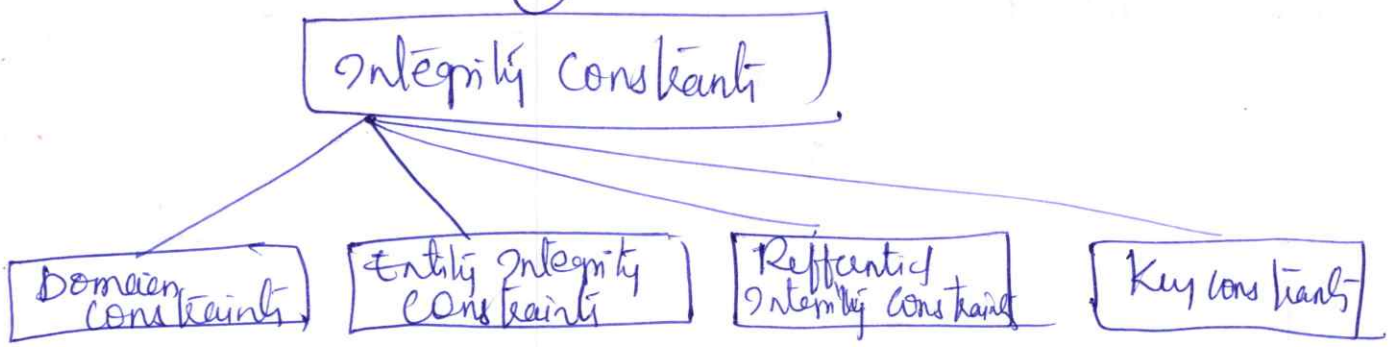



4 (a) Integrity constraints over relations.

Integrity constraints are set of rules helps to maintain accuracy and consistency of data in Database. Types of Integrity constraints are — 2M

* Domain constraints :- Valid set of values for an attribute. The value of the attribute must be available in the corresponding domain.
* Entity Integrity Constraints :- primary key value cannot be null.
* Referential Integrity Constraints :- Specified relationship b/w two tables.
* Key constraints :- used to identify an entity within its entity set uniquely. — 3(M)

(3)




4

(b) Example Queries for Alter, Destroy, Tables and views.

Alter TABLE table_name ADD column_name data type 2m

Eg: ALTER TABLE Students ADD Email varchar(25)

ALTER TABLE table_name MODIFY COLUMN column_name data type

Eg: ALTER TABLE students MODIFY COLUMN Address varchar(250).

ALTER TABLE Table_name DROP Column Column_name;

Eg: ALTER TABLE Students DROP COLUMN Grade;

DELETE FROM EMPLOYEES WHERE emp_id = 5;

Delete from table_name where some condition;

DROP VIEW view_name 3 (m)

DROP VIEW markview.

UPDATE view_name
SET column1 = value1
Where [condition].

ALTER VIEW view_name AS
Select column1
FROM table name
Where condition;

DROP VIEW view_name

DROP VIEW Employees_View.

5(a) Fundamental operations in Relational Algebra with examples?

Relational Algebra is procedural Query language that takes relations as input and returns relations as output. 2M

Operations :-

1) Select (σ) :- filters rows
2) project (π) :- select columns
3) Union (∪) :- Combine rows
4) set Difference (−) :- rows in one but not the other
5) Cartesian product (×) :- Combines all Rows / columns
6) Rename (ρ) :- renames relation / Attribute.
7) Joins (⋈) 3 (M)
   - Inner Join (*) < natural Join (*), Theta Join, Equi Join
   - Outer Join < left Outer Join, Right Outer Join, Full Outer Join.

5(b) Tuple Relational Calculus with Example:

2M

TRC is a non-procedural Query language used to retrieve data from relational database by describing the properties of required data.

Syntax: $\{t \mid P(t)\}$ t → tuple variable P(t) → predicate condition.

logical operators in TRC - 3M

1) ∧ : AND (2) ∨ : OR (3) ¬ : NOT.

Quantifiers -

1) $\exists t \in r\,(Q(t))$ → There exists a tuple t in relation r satisfy Q(t)
2) $\forall t \in r\,(Q(t))$ → for all tuples t in relation r, predicate Q(t) holds.

6(a) Normalization? Various Normal forms with Examples:-

It is also called as schema Refinement. It is the process of decomposing a larger relation into smaller efficient relations.

Types :

(2M)

1) 1NF (2) 2NF (3) 3NF (4) BCNF
(5) 4NF (6) 5NF.

(4)

(3m)

1NF :- It contains atomic values. There should be no repeated values in a particular column.

2NF :- It should be in 1NF and not contain any Partial Dependency. i.e NO partial Dependency i.e primary key should not contain Duplicate values

3NF :- It should be in 2NF and there is no transitive dependency for Non-prime attributes.

BCNF :- It should be in 3NF And for Any dependency A → B, A should be a Super key.

4NF :- It should be in BCNF and no multivalued dependency.

5NF :- It should be 4NF and does not contain any Join dependency.

7(a) trigger? How to implement triggers in SQL. (2m)

trigger is a specialized type of stored procedure that automatically runs when specific events occurs. like (Insert, update, delete).

Implementation :-

```
CREATE TRIGGER Trigger_AuditSalaryChange
AFTER Update ON Employees                    3(m)
FOR EACH ROW
BEGIN
    IF OLD.salary <> NEW.salary THEN
        INSERT INTO SalaryAudit (empID, Oldsalary, Newsalary,
        VALUES(OLD.EmpID, OLD.Salary, New.Salary, CURRENT_TIMESTAMP);
                                  ChangeDate).
    END IF;
END;
```

7(b) Decomposition? problems related to Decomposition.

Decomposition is process of breaking down single table into multiple tables in order to eliminate redundancy, reduce data Anomalies. (2m)

Problems related to decomposition :— 3 (m)

1) loss of information
2) loss of functional Dependency
3) Increased complexity
4) Redundancy
5) performance overhead.

8(a) locking protocol? Strict Two phase locking protocol: (2 m)

locking protocols are concurrency control methods that use locks to manage simultaneous Data access, ensuring Data integrity, Consistency.

Strict Two phase locking protocol :- A common variant that holds all exclusive locks until the transaction commits or aborts. 3 (m)

How it works:

1) Request lock. 2) Grant / wait 3) perform operation
4) Phase Transition.

8(b) Multiple Granularities (2m)

Multiple Granularities introduces a hierarchical structure where locks can be applied at various levels. to balance efficiency and concurrency

lock modes in multiple granularity:- 3(m)

1) Intention Shared (IS)
2) Intention Exclusive (IX)
3) Shared Intention Exclusive (SIX)

9(a) Time-stamp based protocols 2m

It is a Concurrency Control protocol that ensures serializability by assigning unique timestamps to transactions, dictating to their execution order and prevents conflicts without locks.

(b) Time stamp based rules Associated with.

(5)

1) Timestamp Assignment
2) Reading / writing rules
3) Handling violations.

3 (m)

9(b) Concurrent execution of transaction 2m

When multiple transaction in DBMS run concurrently, Problems such as Inconsistencies, uncommitted updates, lost Data can occur. These issues known as concurrency problems.

Types of Concurrency problems:- 3m

1) Dirty Read problem:- occurs when transaction reads uncommitted data
2) unrepeatable Read Problem / Inconsistent Retrievals:- occurs when transaction reads same data multiple times, but during its execution, another transaction modifies the data.
3) lost Read problem:- occurs when two or more transactions attempt to update the same data item concurrently, leading to one or more updates being overwritten.

10(a) Dynamic Index structure: (2m)

A Dynamic Index structure is an index that automatically Adjusts its structure as the data grows, shrinks. Dynamic Index structures reorganize themselves in response to changes in the data they index. This makes them more flexible and scalable. Capable of handling varying workloads and data distribution efficiently.

Ex:- 1) B-Trees and B+-Trees.

⇒ B-Trees and B+-Trees automatically split or merge nodes to maintain a balanced tree ensuring that search, insert and delete operations take logarithmic time.

→ B- Tree :- is a balanced search tree where records are stored within its nodes. It works on

* Nodes and keys
* Balanced structure
* Search process.

(3 m)

→ B+ Trees :- modified version of B-Trees. Specifically designed for indexing. More efficient than B-Trees

It works on

* leaf nodes
* Internal Nodes
* Linked leaf nodes

10 (b) Diff b/w Hash based indexing and Tree based indexing.

→ A hash index takes the key of the value that you are indexing and hashes it into buckets.

Key components

(2 m)

* Hash function
* Buckets
* Collision
* Hash Table.

Types of Hashing

* Dynamic Hashing &
* static Hashing

(6).

Tree Based Indexing :- It uses tree data structure to facilitate efficient data retrieval, insertion and deletion operations. Types of Indexing 3(m)

* B-Tree Indexing
* B+ Tree Indexing

Tree Based Indexing flexible because

* Faster Data Retrieval
* Efficient Range Queries
* Dynamic Updates.

11) What are B+ Trees. How to insert, delete and search Explain with Example.

B+ Trees is a self balancing tree based data structure widely used for database Indexing and file System. 2m

* Key Properties :-
  1) Data Storage at leaves Only
  2) Internal Nodes for Indexing
  3) Linked leaves
  4) Balanced Structure.

* Example :- Assume B+ Tree of order 4. meaning each node can have max 4 children. and max 3 keys. 3(m).

10, 20, 5, 6, 12, 30, 7, 17

→ Insert 10, 20, 5, 6
  * Initial node :- 5, 6, 10, 20.

→ To Insert the next item, the node is Split.

2) Split :- median key.
   * Internal Node 10.
     leaf nodes 5,6 and 10,20

3) Insert 12
   10, 12, 20

4) Insert 30
   10, 12, 20, 30 full.

5) Insert 7, it go into leaf node. 5, 6, 7
   * 5, 6, 7 first leaf
   * 10, 12, 20, 30 second leaf
   * 10, 20 internal node.
   * 5, 6, 7 linked to 10, 12 linked to 20, 30.

6) Insert 17.
   * greater than 10 less than 20.
   * The middle leaf node [10, 12] has room so
     17 inserted 10, 12, 17.