

DBMS Concepts and Solutions



Each question carries 2 marks (2 x 5)

or

1. Define indexing in DBMS.

Define any two of the following terms: track, cylinder, sectors and blocks.

2. Consider the relation scheme R (A, B, C) with the following functional dependencies: ABC, CA Is R in 2NF?

or

Mention the criteria for a table to be in 3NF.

3. When is it preferable to use a dense index rather than a sparse index?

or

What is the need of multilevel indexing?

4. Given a relation R (A, B, C, D) and Functional Dependency set FD (ABCD, BC), 2 determine whether the given R is in 2NF?

or

Given a relation R (P, Q, R, S, T) and Functional Dependency set FD (PQR, S → T), convert it into 2 NF. 2

5. What is the difference between secondary indexing and clustered indexing?

45

or

What is the difference between B-Tree and B-Tree?

Part-B

Attempt 2 questions

Each question carries 5 marks (5 x 2)

6. Consider a database of fixed-length records, stored as an ordered file. The database has 50,000 records, with each record being 100 bytes, of which the primary key occupies 15 bytes. The data file

is block-aligned in that each data record is fully contained within a block. The database is indexed by a primary index file, which is also stored as a block-aligned ordered file. Suppose the block size of the file system is 1024 bytes, and a pointer

Page 1 of 3

2

2

2

2

2

2

2

2

5 solve these questions



Here are the solutions to the provided questions:

Part A: Each question carries 2 marks (2 x 5)

1. Define indexing in DBMS:

Indexing in DBMS is a technique for optimizing the performance of a database by creating an index, which is a data structure, to facilitate quick retrieval of records without scanning the entire table.

Define any two of the following terms:

- **Track:** A track is a circular path on the surface of a disk where data is magnetically recorded and read. Tracks are further divided into sectors.
- **Cylinder:** A cylinder is a set of corresponding tracks on all platters in a disk stack. It is formed when all the tracks are aligned vertically.

2. Is R in 2NF?

Given relation R(A, B, C) and functional dependencies: $ABC \rightarrow CA$.

- A relation is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependent on the primary key.
- The candidate key here is ABC. The only functional dependency, $ABC \rightarrow CA$, suggests no partial dependency.
- Hence, **R is in 2NF.**

Criteria for 3NF:

- A relation is in 3NF if it is in 2NF and no non-prime attribute is transitively dependent on the primary key.

3. When is it preferable to use a dense index rather than a sparse index?

- A dense index is preferred when the database requires fast and frequent searches and when there are frequent updates to the data because it provides an entry for every record.

Need for multilevel indexing:

- Multilevel indexing is needed to handle large databases where a single-level index is too large to fit into memory. It reduces the access time by organizing the index into multiple levels.

4. Is R in 2NF?

Given R(A, B, C, D) with FD: $AB \rightarrow CD$, $BC \rightarrow A$.

- Candidate key: AB, BC.
- Non-prime attributes (D, A) are fully dependent on the keys. No partial dependency exists.
- Hence, **R is in 2NF**.

Convert R(P, Q, R, S, T) to 2NF:

- Given FD: $PQR \rightarrow S$, $S \rightarrow T$.
- Step 1: Split into R1(PQR, S) and R2(S, T).
- Step 2: Verify no partial dependency.
- Final relations: R1(PQR, S) and R2(S, T).

5. Difference between secondary and clustered indexing:

- **Secondary Indexing:** Created on non-primary key columns to improve query performance; allows duplicates.
- **Clustered Indexing:** Reorganizes the physical order of the data to match the order of the key; only one clustered index per table.

Difference between B-Tree and B+ Tree:

- **B-Tree:** Stores keys and data in internal and leaf nodes.
- **B+ Tree:** Stores keys in internal nodes and data only in leaf nodes, which are linked sequentially for faster range queries.

Part B: Each question carries 5 marks (5 x 2)

6. Database Problem Analysis:

- **Database size:** 50,000 records \times 100 bytes = 5,000,000 bytes (5 MB).
- **Number of records per block:** Block size \div Record size = $1024 \div 100 = 10$.
- **Total blocks for data file:** Total records \div Records per block = $50,000 \div 10 = 5,000$.
- **Index entry size:** Key (15 bytes) + Pointer (4 bytes) = 19 bytes.
- **Entries per block:** Block size \div Index entry size = $1024 \div 19 \approx 53$.
- **Total blocks for index file:** Total records \div Entries per block = $50,000 \div 53 \approx 943$.

Answer:

- Data file: 5,000 blocks.
- Index file: 943 blocks.

Let me know if you need more details!



In 2 it is $A, B \rightarrow C, C \rightarrow A$



Thanks for the clarification. Here's the corrected solution for question 2:

2. Is R in 2NF?

Given relation $R(A, B, C)$ and functional dependencies:

- $A, B \rightarrow C$
- $C \rightarrow A$

Steps to check 2NF:

- A relation is in **2NF** if it is in **1NF** and has no **partial dependency** (i.e., no non-prime attribute depends on part of a candidate key).
- **Candidate key:** AB (Since AB determines all attributes).
- **Prime attributes:** A, B .
- C is a non-prime attribute.

Analysis:

- From $A, B \rightarrow C$, C is fully dependent on AB (no partial dependency).
- However, $C \rightarrow A$ introduces a **transitive dependency**:
 A (prime) depends on C (non-prime) via $AB \rightarrow C \rightarrow A$.

Conclusion:

The relation is **not in 2NF** because of the transitive dependency $C \rightarrow A$.

Let me know if further clarification is needed!



to a block occupies 5 bytes. The system uses binary search on the index file to search for a to a blokhe given key. You may assume that a binary search on an index file of b blocks takes $\lceil \log b \rceil$ block accesses in the worst case.

Given a key, find the number of block accesses required to identify the block in the data file that may contain a record with the key, in the worst case.

or

Find the difference between spanned records and unspanned records. Find the difference between fixed-length records and variable-length records.

7. $R(A, B, C, D)$ is a relation. Perform BCNF decomposition and check if the following does not have a lossless decomposition, dependency preserving BCNF decomposition - $AB \rightarrow C, CAD$

Justify your answer with clear working.

or

Consider the following set of functional dependency on the scheme (W, X, Y, Z)

X-W, WZXY, Y-WXZ

Find the canonical cover for the set.

Part-C

Attempt 1 question

Each question carries 10 marks (10 x 1)

8. Consider the following database schema for a library where primary keys are indicated by bold and italics fonts:

Member(**memb_no**, name, dob)

Books(**isbn**, title, authors, publisher) Borrowed(**memb_no**, **isbn**, date)

Write Tuple relational calculus for each of the following queries:

- Find the names of members who have borrowed any book published by "McGraw- Hill".
- Find the name and membership number of members who have borrowed more than five different books published by "McGraw-Hill".
- For each publisher, find the name and membership number of members who have borrowed more than five books of that publisher.
- Find the total number of books borrowed per member. Take into account that if a member does not borrow any books, then that member does not appear in the borrowed relation at all.

or

A table named 'EMPLOYEES' is having employee details including salary and there are already some tuples in the table.

Create a Stored procedure for the following: -

- There is 10% increase in the salary of the employee. Calculate the new salary.

Page 2 of 3

5

G

5



Here are the answers to the given questions:

Part A (Continued)

6. Find the number of block accesses required to identify the block in the data file in the worst case:

- **Index file analysis:**

- The index file has 943 blocks (as calculated earlier).
- Using binary search, the number of block accesses is $\lceil \log_2(943) \rceil$.
- $\log_2(943) \approx 9.88$, so $\lceil 9.88 \rceil = 10$.

- **Data file access:**

- Once the correct index block is located, a single access to the corresponding data block is needed.

Total block accesses: 10 (for index file) + 1 (for data file) = **11 block accesses** in the worst case.

Difference between spanned and unspanned records:

- **Spanned records:** Records that span across multiple blocks when their size exceeds the block size. These require additional pointers to link blocks.
- **Unspanned records:** Records that are entirely contained within a single block. They simplify record retrieval but may result in unused space within a block.

Difference between fixed-length and variable-length records:

- **Fixed-length records:** All records have the same size, simplifying storage and retrieval but wasting space if fields are underutilized.
 - **Variable-length records:** Records can have different sizes, allowing flexibility in storage but requiring extra metadata (e.g., length indicators or delimiters) to manage records.
-

7. Perform BCNF decomposition and check for lossless dependency-preserving decomposition:

Given Relation: $R(A, B, C, D)$

Functional Dependencies (FDs):

- $AB \rightarrow C$
- $CA \rightarrow D$

Step 1: Check for BCNF

- Candidate key: AB (since AB determines all attributes).
- $AB \rightarrow C$: This is BCNF compliant because AB is a superkey.
- $CA \rightarrow D$: CA is not a superkey (it does not determine all attributes), so the relation violates BCNF.

Step 2: Decompose to BCNF

- Decompose R into $R_1(CA, D)$ and $R_2(A, B, C)$.

Step 3: Check for lossless decomposition

- $R_1 \cap R_2 = C$, which is part of $CA \rightarrow D$. Hence, the decomposition is **lossless**.

Step 4: Check for dependency preservation

- $AB \rightarrow C$: Preserved in R_2 .

- $CA \rightarrow D$: Preserved in $R1$.

Conclusion: The decomposition is both **lossless** and **dependency-preserving**.

Canonical cover for functional dependencies on (W, X, Y, Z) :

Given FDs:

- $X \rightarrow W$
- $WZ \rightarrow XY$
- $Y \rightarrow WXZ$

Steps to find canonical cover:

1. Simplify RHS:

- $WZ \rightarrow XY$: Split into $WZ \rightarrow X, WZ \rightarrow Y$.
- $Y \rightarrow WXZ$: Split into $Y \rightarrow W, Y \rightarrow X, Y \rightarrow Z$.

2. Eliminate redundant FDs:

- $Y \rightarrow W, Y \rightarrow X$, and $Y \rightarrow Z$ imply $Y \rightarrow WXZ$, so retain only $Y \rightarrow Z$.

3. Final canonical cover:

- $X \rightarrow W$
- $WZ \rightarrow X$
- $WZ \rightarrow Y$
- $Y \rightarrow Z$.

Part C (Each question carries 10 marks)

8. Tuple Relational Calculus Queries:

Given Schema:

- $\text{Member}(\text{memb_no}, \text{name}, \text{dob})$
- $\text{Books}(\text{isbn}, \text{title}, \text{authors}, \text{publisher})$
- $\text{Borrowed}(\text{memb_no}, \text{isbn}, \text{date})$

(a) Find the names of members who have borrowed any book published by "McGraw-Hill":

$\{m.name \mid \exists b, br (br.memb_no = m.memb_no \wedge br.isbn = b.isbn \wedge b.publisher = "McGraw -$

(b) Find the name and membership number of members who have borrowed more than five different books published by "McGraw-Hill":

$\{(m.name, m.memb_no) \mid \exists b1, b2, \dots, b6, br1, \dots, br6 (br1.memb_no = m.memb_no \wedge br1.isbn =$

(c) For each publisher, find the name and membership number of members who have borrowed more than five books of that publisher:

$\{(p.publisher, m.name, m.memb_no) \mid \exists b1, \dots, b6, br1, \dots, br6 (br1.memb_no = m.memb_no \wedge b$

(d) Find the total number of books borrowed per member:

$$\{(m.memb_no, COUNT(b.isbn)) \mid br.memb_no = m.memb_no \wedge br.isbn = b.isbn\}$$

Stored Procedure:

(a) 10% salary increase calculation:

```
sql

DELIMITER $$
CREATE PROCEDURE IncreaseSalary()
BEGIN
    UPDATE EMPLOYEES
    SET salary = salary * 1.10;
END$$
DELIMITER ;
```



(b) Update salary in the table with the new salary.

(c) Display the new salary.

(d) Write appropriate code in the "EXCEPTION" clause to handle the situation when no tuple is found in the table.

(e) Call the above procedure with suitable arguments.

-End-



Here is the complete SQL code with all requested parts:

(b) Update salary in the table with the new salary:

```
sql

DELIMITER $$
CREATE PROCEDURE UpdateSalary()
BEGIN
    UPDATE EMPLOYEES
    SET salary = salary * 1.10;
END$$
DELIMITER ;
```

(c) Display the new salary:

```
sql

DELIMITER $$
CREATE PROCEDURE DisplayNewSalary()
BEGIN
```



```
SELECT employee_id, name, salary AS new_salary
FROM EMPLOYEES;
END$$
DELIMITER ;
```

(d) Handle the exception when no tuple is found:

To handle this situation in SQL, you can use a condition or a block to check if any rows are affected by the `UPDATE` operation.

```
sql

DELIMITER $$
CREATE PROCEDURE UpdateSalaryWithExceptionHandling()
BEGIN
    DECLARE row_count INT;

    -- Perform the salary update
    UPDATE EMPLOYEES
    SET salary = salary * 1.10;

    -- Check the number of rows affected
    SET row_count = ROW_COUNT();

    -- Handle the case where no rows are updated
    IF row_count = 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'No records found in the table to update salary.';
    END IF;
END$$
DELIMITER ;
```

(e) Call the procedure with suitable arguments:

Here, since the procedure does not require specific arguments, simply call it as follows:

```
sql

-- Call the procedure to update salary
CALL UpdateSalary();

-- Call the procedure to display the updated salaries
CALL DisplayNewSalary();

-- Call the procedure with exception handling
CALL UpdateSalaryWithExceptionHandling();
```

This solution ensures the salary is updated, displayed, and handled appropriately if no tuples are present in the table. Let me know if additional details are required!