# 1. Vision, Mission, PEO's & PO's

## VISION

To be renowned department imparting both technical and non-technical skills to the students through implementing new engineering pedagogy and research to produce competent new age electrical engineers

## MISSION

- To transform the students into motivated and knowledgeable new age electrical engineers.
- To advance the quality of education to produce world class technocrats with an ability to adapt to the academically challenging environment.
- To provide a progressive environment for learning through organized teaching methodologies, contemporary curriculum and research in the thrust areas of electrical engineering.

## PROGRAM EDUCATIONAL OBJECTIVES

| | |
|---|---|
| **PEO 1** | Apply knowledge and skills to provide solutions to Electrical and Electronics Engineering problems in industry and governmental organizations or to enhance student learning in educational institutions |
| **PEO 2** | Work as a team with a sense of ethics and professionalism, and communicate effectively to manage cross-cultural and multidisciplinary teams |
| **PEO 3** | Update their knowledge continuously through lifelong learning that contributes to personal, global and organizational growth |

## PROGRAM OUTCOMES

| Pos | DESRCRIPTION |
|---|---|
| **PO1** | An ability to apply knowledge of computing, mathematical foundations, algorithmic principles, and computer science and engineering theory in the modeling and design of computer-based systems to real-world problems. |
| **PO2** | An ability to design and conduct experiments, as well as to analyze and interpret data. |
| **PO3** | An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs, within realistic constraints such as economic, environmental, social, political, health and safety, manufacturability, and sustainability. |
| **PO4** | An ability to function effectively on multidisciplinary teams. |
| **PO5** | An ability to analyze a problem, and identify, formulate and use the appropriate computing and engineering requirements for obtaining its solution. |
| **PO6** | An understanding of professional, ethical, legal, security and social issues and responsibilities. |
| **PO7** | Communicate effectively with the engineers and society at large through the ability to comprehend and write effective reports, make effective presentations, give and receive clear instructions. |
| **PO8** | 0Ability to understand the local and global impact of computing and engineering solutions on individuals, organizations, and society. |
| **PO9** | Recognition of the need for, and an ability to engage in continuing professional development and lifelong learning. |
| **PO10** | Have an open mind and have an understanding of the impact of engineering on society and demonstrate awareness of contemporary issues. |

| | |
|---|---|
| **PO11** | An ability to use current techniques, skills, and tools necessary for computing and engineering practice. |
| **PO12** | Ability to recognize the importance of professional development by pursuing |
| | postgraduate studies or face competitive examinations that offer challenging and rewarding careers in Computer Science Engineering. |

# 2. Syllabus (University Copy)

**UNIT I: Introduction** : Data base System Applications, Purpose of Database Systems, View of Data Database Languages–DDL, DML, Relational Database, Database Design, Data Storage and Querying, Transaction Management, Database Architecture, Data Mining and information Retrieval, Specialty Database ,Database Users and Administrator, History of Data base Systems.

**Introduction to Data base design**: Database Design and ER diagrams, Entities, Attributes and Entity sets, Relationships and Relationship sets, Additional features of ER Model, Conceptual Design with the ER Model, Conceptual Design for Large enterprises.

**Relational Model** – Introduction to the Relational Model, Integrity Constraint Over relations, Enforcing Integrity constraints, Querying relational data, Logical data base Design: ER to Relational, Introduction to Views, Destroying /altering Tables and Views.

**UNIT II: Relational Algebra and Calculus**: Preliminaries, Relational Algebra, Relational calculus-Tuple relational Calculus, Domain relational calculus – Expressive Power of Algebra and calculus.
**SQL:** Queries, Constraints, Triggers :Form of Basic SQL Query, UNION, INTERSECT, and EXCEPT Nested Queries, Aggregate Operators, NULL values Complex Integrity Constraints in SQL Triggers and Active Data bases, Triggers and Active Databases ,Designing Active Databases.

**UNIT III: Schema Refinement and Normal Forms:** Introduction to Schema refinement, Function

dependencies, reasoning about FDS, Normal Forms, properties of Decompositions, Normalization, Schema Refinement in Database Design, Other Kinds of Dependencies.

**UNIT IV: Transaction Management**- Transaction, Transaction Concept, A Simple transaction Model, Storage Structure, Transaction Atomicity and Durability, Transaction Isolation Serializability, Transaction Isolation and Atomicity Transaction Isolation levels, Implementation of Isolation Levels.

**Concurrency Control**: Lock –Based Protocols, Multiple Granularity, Timestamp Based Protocols, Validation- Based Protocols, Multisession Schemes.

**Recovery system** : Failure Classification, Storage Structure, Recovery and Atomicity, Recovery Algorithm, Buffer Management ,Failure with loss of nonvolatile storage ,Early Lock Release and Logical Undo Operations, Remote Backup systems.

**UNIT V: Storage and Indexing**: Overview of Storage and Indexing: Data on External Storage, File Organization and Indexing, Index data Structures, Comparison of File Organizations.

**Tree Structured Indexing:** Intuitions for tree Indexes – Indexed Sequential Access Methods (ISAM) – B+ Trees: A Dynamic Index Structure, Search, Insert,Delete.

**Hash Based indexing**: Static Hashing, Extendable Hashing, Linear Hashing, Extendible vs. Linear Hashing

**TEXT BOOKS**:

Data base Management Systems, Raghurama Krishnan, Johannes Gehrke, TATA McGrawHill 3rd Edition

Data base System Concepts, Silberschatz, Korth, McGraw hill, VI edition.

**REFERENCES**:

Data base Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.

Fundamentals of Database Systems, Elmasri Navrate Pearson Education Introduction to Database

Systems, C.J.Date Pearson Education

# 3. Course Objectives, Course Outcomes and Topic Outcomes

## COURSE OBJECTIVES

At the end of the course, the students will be able to:

1. To understand the basic concepts and the applications of database systems.
2. To master the basics of SQL and construct queries using SQL.
3. To understand the relational database design principles.
4. To become familiar with the basic issues of transaction processing and concurrency control.
5. To become familiar with database storage structures and access techniques.

## COURSE OUTCOMES

After completing this course the student must demonstrate the knowledge and ability to:

1. **Explain** the basic concept and the application of database system queries using SQL.
2. **Apply** commercial relational database system by writing Queries using SQL and relational Database Theory to write relational algebra Expressions.
3. **Explain** design principles for logical design of database including the ER model and normalization approach.
4. **Demonstrate** the basics of query evaluation and apply query optimization techniques of transaction processing and concurrency control.
5. **Explain** the basics of storage structures, indexing and page organization methods including B-Tree and Hashing.

TOPIC OUTCOMES

| SNO | TOPIC | TOPIC OUTCOMES |
|---|---|---|
| | **Unit-I** | |
| 1 | Introduction to DBMS | Understand about the database management system |
| 2 | Introduction, Data base System Applications Purpose of Database System | Evaluate the system application and also the purpose of DBMS |
| 3 | View of Data ,Database Languages – DDL, DML | Analyze the type of data languages |
| 4 | Relational Databases | To know and understand a relational database |
| 5 | Database Design, Data storage and querying | To create a design and querying |
| 6 | Transaction Management | Evaluate about the transaction management |
| 7 | Database Architecture | Understand about the data base architecture |
| 8 | Data Mining and information Retrieval, Specialty Data bases | To analyze the datamining and speciality |
| 9 | Database Users and Administrator, History of Database System | To understand about the admine |
| 10 | Database Design and ER diagrams | To create the entity relational diagrams |
| 11 | Entities, Attributes and Entity sets, Relationships and Relationship sets | To analyze all the relationships and sets and attributes |
| 12 | Additional features of ER Model | Remember the entity relational model |

| | | |
|---|---|---|
| 13 | Concept Design with the ER Model Conceptual Design for Large enterprises. Case study(additional) | Evaluate the entity relational model and large enterprises |
| 14 | Introduction to the Relational Model Integrity Constraint Over relations Enforcing Integrity constraints | Understand the relational model and integrity constraints |
| 15 | Querying relational data – Logical data base Design | Create the logical database design |
| 16 | Introduction to Views – Destroying /altering Tables and Views | Create the destroying and altering of tables |
| **UNIT-II** | | |
| 17 | Preliminaries, Relational Algebra | To evaluate the preliminaries and relational algebra |
| 18 | Relational calculus – Tuple relational Calculus , Domain relational calculus | To analyze trc,drc |
| 19 | Expressive Power of Algebra and calculus | To understand the power of algebra and calculus |
| 20 | Queries, Constraints, Triggers: Form of Basic SQL Query – Examples of Basic SQL Queries | To create the triggers and constraints |
| 21 | Introduction to Nested Queries | To understand the nested queries |
| 22 | Aggregative Operators – NULL values | To understand the null values |
| 23 | Complex Integrity Constraints in SQL Triggers and Active Data bases, Designing Active Databases | Evaluate about the triggers and active databases |
| **UNIT –III** | | |
| 24 | Schema refinement | Understand the schema rrefinement |

| | | |
|---|---|---|
| 25 | Functional dependencies – reasoning about FDS | To analyze the functional dependenncies |
| 26 | FIRST, SECOND, THIRD Normal forms – BCNF | Evaluate the normal forms |
| 27 | FORTH Normal Form | Evaluate the normal forms |
| 28 | Fifth Normal Form Dependencies | Evaluate the normal forms |
| 29 | Properties of Decompositions, Normalization | To understand the properties od decoposition |
| 30 | Schema Refinement in Database | To remember all the schema |
| | Design, Other kinds of Dependencies | refinement |
| | **UNIT –IV** | |
| 31 | Transaction ,Transaction Concept | Understand the transaction concepts |
| 32 | A simple Transaction Model, Storage Structure | Remember a simple transaction model |
| 33 | Transaction of Atomicity and Durability | To evaluate the atomicity and durability |
| 34 | Transaction Isolation, Serializability | Analyze the isolation and serializability |
| 35 | Transaction Isolation and Atomicity Transaction Isolation Levels,Implementation of isolation levels | To apply the atomicity transaction |
| 36 | Lock –Based Protocols,Multiple Grnularity | Know and understand the protocal |
| 37 | Timestamp Based Protocols | Evaluate the time stamp based protocal |
| 38 | Validation- Based Protocols | Analyze the validation based protocal |
| 39 | Multiversion Schemes | Create the multi version schemes |
| 40 | Recovery System-Failure Cllassification | Remember the recovery system failure |
| 41 | Recovery and Atomicity,Recovery Algorithm,Buffer Management | Buffer management will be apply |

| | | |
|---|---|---|
| 42 | Failure with loss of non volatile storage | Understand the loss of non volatile storage |
| 43 | Early Lock Release and Logical undo Operations, Remote backup systems | To evaluate the remote backup system |
| **UNIT –V** | | |
| 44 | Overview of Storage and Indexing: Data on External Storage – File Organization and Indexing | To understand the data on external storage and indexing |
| 45 | Index data Structures | To evaluate the index data structure |
| 46 | Comparison of the File Organization | To analyze the comparision of file organization |
| 47 | Tree-Structured Indexing Intuition for tree indexes, Indexed Sequential Access Methods (ISAM) | To apply the indexing and sequential methods |
| 48 | B+ Trees: A Dynamic Index Structure | Create a B+ Trees with index structure |
| 49 | Search, Delete, Insert | To remember the search delete and insert |
| 50 | Hash based indexing Static Hashing, | To evaluate the indexing and hashing |
| 51 | Extendable hashing | To analyze the extendable hashing |
| 52 | Linear Hashing | To evaluate the linear hashing |
| 53 | Extendible vs. linear Hashing | To apply the extendible and linear hashing |

# 4. Course Prerequisites

Basic concepts of files, data structures and design of database systems

## 5. Course Objectives, TO's, CO's, PO's & PEO's Mapping

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 3 | 2 | | l | | | | | | | | |
| CO2 | | 3 | | 2 | | | | | | | | |
| CO3 | | 2 | | 3 | | | | | | | | |
| CO4 | | 3 | | 2 | | | | | | | | 1 |
| CO5 | | 3 | | 2 | | | | | | | | |

1- Low
2- Medium
3- high

# 6. Course Information Sheet (CIS)

### i. Course Description

| | |
|---|---|
| PROGRAMME: B. Tech. (Electrical and Electronic Engineering.) | DEGREE: BTECH |
| COURSE: **DATABASE MANAGEMENT SYSTEMS** | YEAR: III SEM: I<br>CREDITS: 3 |
| COURSE CODE: CS5120E<br>REGULATION: R16 | COURSE TYPE: Open Elective |
| COURSE AREA/DOMAIN: Design | CONTACT HOURS: 4 (L))<br>hours/Week. |
| CORRESPONDING LAB COURSE CODE (IF ANY): No | LAB COURSE NAME: **DATABASE MANAGEMENT SYSTEMS** |

## ii. Syllabus

| Unit | Details | Hours |
|---|---|---|
| I | **Introduction** : Data base System Applications, Purpose of Database Systems, View of Data , Database Languages–DDL, DML, Relational Database, Database Design, Data Storage and Querying, Transaction Management, Database Architecture, Data Mining and information Retrieval, Specialty Database ,Database Users and Administrator, History of Data base Systems.<br>**Introduction to Data base design**: Database Design and ER diagrams, Entities, Attributes and Entity sets, Relationships and Relationship sets, Additional features of ER Model, Conceptual Design with the ER Model, Conceptual Design for Large enterprises. **Relational Model** – Introduction to the Relational Model, Integrity Constraint Over relations, Enforcing Integrity constraints, Querying relational data, Logical data base Design: ER to Relational, Introduction to Views, Destroying /altering Tables and Views. | **16** |
| II | **Relational Algebra and Calculus**: Preliminaries, Relational Algebra, Relational calculus-Tuple relational Calculus, Domain relational calculus – Expressive Power of Algebra and calculus.<br>**SQL:** Queries, Constraints, Triggers :Form of Basic SQL Query, UNION, INTERSECT, and EXCEPT Nested Queries, Aggregate Operators, NULL values Complex Integrity Constraints in SQL Triggers and Active Data bases, Triggers and Active Databases<br>,Designing Active Databases. | **12** |
| III | **Schema Refinement and Normal Forms:** Introduction to Schema refinement, Function dependencies, reasoning about FDS, Normal Forms, properties of Decompositions, Normalization, Schema Refinement in | |

| | | |
|---|---|---|
| | Database Design, Other Kinds of Dependencies. | **07** |
| **IV** | **Transaction Management**- Transaction ,Transaction Concept, A Simple transaction Model, Storage Structure, Transaction Atomicity and Durability, Transaction Isolation , Serializability , Transaction Isolation and Atomicity Transaction Isolation levels, Implementation of Isolation Levels.<br>**Concurrency Control**:Lock –Based Protocols, Multiple Granularity, Timestamp Based Protocols, Validation- Based Protocols, Multiversion Schemes.<br>**Recovery system** : Failure Classification, Storage Structure, Recovery and Atomicity, Recovery Algorithm, Buffer Management ,Failure with loss of nonvolatile storage ,Early Lock Release and Logical Undo Operations, Remote Backup systems. | 13 |
| **V** | **Storage and Indexing**: Overview of Storage and Indexing: Data on External Storage, File Organization and Indexing , Index data Structures , Comparison of File Organizations.<br>**Tree Structured Indexing :** Intuitions for tree Indexes – Indexed Sequential Access Methods (ISAM) – B+ Trees: A Dynamic Index Structure, Search, Insert, Delete.<br>**Hash Based indexing**: Static Hashing, Extendable Hashing, Linear Hashing, Extendible vs. Linear Hashing. | 10 |
| | **Contact classes for syllabus coverage** | **58** |
| **Lectures beyond syllabus** | | **04** |
| **Tutorial classes** | | **00** |
| | **Classes for gaps& Add-on classes** | **04** |
| **Total No. of classes** | | **66** |

## iii. GAPS IN SYLLABUS

| S.NO. | DESCRIPTION | PROPOSED ACTIONS |
|---|---|---|
| 1 | Basic concepts of files, data structures and design of database systems | 4 period (Class Room) |

## iv. TOPICS BEYOND SYLLABUS

| | | |
|---|---|---|
| 1 | PLSQL CONCEPTS (PROCEDURES,TRIGGERS,CURSORS) | 4 classes by guest faculty |
| 2 | Data modeling and schema design | NPTEL |

## V. WEB SOURCE REFERENCE

| Sl. No. | Name of book/ website |
|---|---|
| a. | www.Tutorialpoint.com |
| b. | www.nptelvideos.in |
| c. | ebooks.library.cornell.edu |

## vi. Delivery/Instructional Methodologies

| | | |
|---|---|---|
| □ CHALK & TALK | □ STUD. ASSIGNMENT | □ WEB RESOURCES |
| □ LCD/SMART BOARDS | □ STUD. SEMINARS | ☐ ADD-ON COURSES |

## vii. Assessment Methodologies-Direct

| | | | |
|---|---|---|---|
| □ ASSIGNMENTS | □ STUD. SEMINARS | □ TESTS/MODEL EXAMS | □ UNIV. EXAMINATION |
| ☐ STUD. LAB PRACTICES | ☐ STUD. VIVA | ☐ MINI/MAJOR PROJECTS | ☐ CERTIFICATIONS |
| ☐ ADD-ON COURSES | ☐ OTHERS | | |

## vii. Assessment Methodologies-Indirect

| | |
|---|---|
| □ ASSESSMENT OF COURSE OUTCOMES (BY FEEDBACK, ONCE) | □ STUDENT FEEDBACK ON FACULTY (TWICE) |
| ☐ ASSESSMENT OF MINI/MAJOR PROJECTS BY EXT. EXPERTS | ☐ OTHERS |

## ix. Text books & Reference books

| T/R | BOOK TITLE/AUTHORS/PUBLICATION |
|---|---|
| Text Book | Data base Management Systems, Raghurama Krishnan, Johannes Gehrke, TATA McGrawHill 3rd Edition |
| Text Book | Data base System Concepts, Silberschatz, Korth, McGraw hill, VI edition. |
| Reference Book | Data base Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition. |
| Reference Book | Fundamentals of Database Systems, Elmasri Navrate Pearson Education |
| Reference Book | Introduction to Database Systems, C.J.Date Pearson Education |

# 7. Micro Lesson Plan

| S.N. | Topic | Schedule data | Actual Date |
|---|---|---|---|
| | **UNIT-I** | | |
| 1 | Introduction to DBMS | | |
| 2 | Introduction, Data base System Applications Purpose of Database System | | |
| 3 | View of Data ,Database Languages – DDL, DML | | |
| 4 | Relational Databases | | |
| 5 | Database Design, Data storage and querying | | |
| 6 | Transaction Management | | |
| 7 | Database Architecture | | |
| 8 | Data Mining and information Retrieval, Specialty Data bases | | |
| 9 | Database Users and Administrator, History of Database System | | |
| 10 | Database Design and ER diagrams | | |
| 11 | Entities, Attributes and Entity sets, Relationships and Relationship sets | | |
| 12 | Additional features of ER Model | | |
| 13 | Concept Design with the ER Model Conceptual Design for Large enterprises. Case study(additional) | | |

| | | | |
|---|---|---|---|
| 14 | Introduction to the Relational Model Integrity Constraint Over relations Enforcing Integrity constraints | | |
| 15 | Querying relational data – Logical data base Design | | |
| 16 | Introduction to Views – Destroying /altering Tables and Views | | |
| | **UNIT-II** | | |
| 17 | Preliminaries, Relational Algebra | | |
| 18 | Relational calculus – Tuple relational Calculus , Domain relational calculus | | |
| 19 | Expressive Power of Algebra and calculus | | |
| 20 | Queries, Constraints, Triggers: Form of Basic SQL Query – Examples of Basic SQL Queries | | |
| 21 | Introduction to Nested Queries | | |
| 22 | Aggregative Operators – NULL values | | |
| 23 | Complex Integrity Constraints in SQL Triggers and Active Data bases, Designing Active Databases | | |
| | **UNIT –III** | | |
| 24 | Schema refinement | | |
| 25 | Functional dependencies – reasoning about FDS | | |
| 26 | FIRST, SECOND, THIRD Normal forms – BCNF | | |
| 27 | FORTH Normal Form | | |

| | | | |
|---|---|---|---|
| 28 | Fifth Normal Form Dependencies | | |
| 29 | Properties of Decompositions, Normalization | | |
| 30 | Schema Refinement in Database Design, Other kinds of Dependencies | | |
| | **UNIT –IV** | | |
| 31 | Transaction ,Transaction Concept | | |
| 32 | A simple Transaction Model, Storage Structure | | |
| 33 | Transaction of Atomicity and Durability | | |
| 34 | Transaction Isolation, Serializability | | |
| 35 | Transaction Isolation and Atomicity Transaction Isolation Levels,Implementation of isolation levels | | |
| 36 | Lock –Based Protocols,Multiple Grnularity | | |
| 37 | Timestamp Based Protocols | | |
| 38 | Validation- Based Protocols | | |
| 39 | Multiversion Schemes | | |
| 40 | Recovery System-Failure Cllassification | | |
| 41 | Recovery and Atomicity,Recovery Algorithm,Buffer Management | | |
| 42 | Failure with loss of non volatile storage | | |
| 43 | Early Lock Release and Logical undo Operations, Remote backup systems | | |
| | **UNIT –V** | | |

| | | | |
|---|---|---|---|
| 44 | Overview of Storage and Indexing: Data on External Storage – File Organization and Indexing | | |
| 45 | Index data Structures | | |
| 46 | Comparison of the File Organization | | |
| 47 | Tree-Structured Indexing:Intuition for tree indexes, Indexed Sequential Access Methods (ISAM) | | |
| 48 | B+ Trees: A Dynamic Index Structure | | |
| 49 | Search, Delete, Insert | | |
| 50 | Hash based indexing<br>Static Hashing, | | |
| 51 | Extendable hashing | | |
| 52 | Linear Hashing | | |
| 53 | Extendible vs. linear Hashing | | |

# 8. Teaching Schedule

| | |
|---|---|
| **Subject** | **DATABASE MANAGEMENT SYSTEMS(DBMS)** |
| **Text Books (to be purchased by the Students)** | |
| **Book 1** | Data base Management Systems, Raghurama Krishnan, Johannes Gehrke, TATA McGrawHill 3rd Edition |
| **Book 2** | Data base System Concepts, Silberschatz, Korth, McGraw hill, VI edition. |
| **Reference Books** | |
| **Book 3** | Data base Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition. |

| **Book 4** | Fundamentals of Database Systems, Elmasri Navrate Pearson Education | | | | |
|---|---|---|---|---|---|
| **Unit** | **Topic** | **Chapters Nos** | | | **No of classes** |
| | | **Book 2** | **Book 3** | **Book 4** | |
| **I** | Introduction to data ware house and data mining, Data ware house, Differences between operational databases and data warehouses<br>data ware house characteristics<br>data ware house architecture, data ware house components, Extraction-Transformation- Loading | 1 | | | 3 |
| | ,Logical(Multi-Dimensional), Data Modeling, Schema Design, Star and Snow-Flake Schema, Fact Consultation, Fact Table, Fully Addictive, Semi-Addictive, Non Addictive Measures: Fact-Less-Facts, Dimension Table Characteristics | 2 | | | 5 |
| | OLAP Cube, OLAP Operations, OLAP Server architecture-ROLAP, MOLAP and HOLAP. | 3 | | | 2 |
| **II** | Introduction to Data mining, KDD, challenges | 3 | | | 2 |
| | Data mining tasks, Data Preprocessing, data cleaning | 5 | | | 3 |

| | | | | | |
|---|---|---|---|---|---|
| | Missing data<br><br>Dimensionality Reduction | 4 | | | 2 |
| | Feature subset selection, Discretization and Binaryzation | 6 | | | 2 |
| | Data Transformation, Measures of Similarity and Sissimilarity-Basics | 7 | | | 1 |
| **III** | Association rules: problem definition, Frequent itemset generation | 6 | | 4 | 2 |
| | The APRIORI Principle, Support and Confidence Measures, Association Rule Generation; APRIORI Algorithm, The<br>Partition Algorithms, FP-Growth Algorithms | | | | 4 |
| | Compact representation of Frequent itemset- Maximal frequent item set, closed frequent<br>item set | 6 | | 10 | 5 |
| **IV** | **Classification:** Problem Definition, General Approaches to solving a classification problem, Evaluation of Classifiers,<br>Classification techniques | 7 | | 9 | 3 |
| | Decision tree construction, Methods for expressing attribute test conditions | 7 | | 9 | 2 |
| | Measures for Selecting the Best Split, Algorithm for Decision tree Induction; Naïve- Bayes Classifier, Bayesian Belief Networks; K-Nearest neighbor | 11 | | | 5 |

| | | | | | |
|---|---|---|---|---|---|
| | classification-Algorithm and Characteristics. | | | | |
| V | **Cluster:** Problem Definition, Clustering. Overview | 11 | | 10 | 1 |
| | Evaluation of Clustering Algorithms, Partitioning Clustering –, PAM Algorithm, Hierarchical Clustering K-Means Algorithm,<br>K-Means Additional issues | 11 | | | 4 |
| | -Agglomerative Methods and divisive methods, Basic Agglomerative Hierarchical Clustering, Strngths and Weakness; Outlier Detection | 14 | | 11 | 4 |
| **Contact classes for syllabus coverage** | | | | **58** | |
| **Tutorial classes** | | | | **08** | |
| **Total No. of classes** | | | | **66** | |

## 9. Hand Written Notes unit wise Completed and Submitted

## 10. OHD/LCD SHEETS/CDS/DVDS/PPT(Soft/Hard Copies)

## 11. University Previous Question Papers

**R15**

**Code No: 124CQ**

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**B.Tech II Year II Semester Examinations, December - 2017**

**DATABASE MANAGEMENT SYSTEMS**

**(Common to CSE, IT)**

**Time: 3 Hours** **Max. Marks: 75**

**Note:** This question paper contains two parts A and B.

Part A is compulsory which carries 25 marks. Answer all questions in Part A.
Part B consists of 5 Units. Answer any one full question from each unit.
Each question carries 10 marks and may have a, b, c as sub questions.

**PART – A**

**(25 Marks)**

1.a) What are five main functions of a database administrator? [2]

b) List and explain the database system applications. [3]

c) Define a trigger. What are the differences between row level and statement level triggers? [2]

d) How are queries expressed in SQL? [3]

e) List the benefits of BCNF and 3NF. [2]

f) Write the Properties of Decompositions. [3]

g) Why is recoverability of schedules desirable? [2]

h) Suppose that there is a database system that never fails. Is a recovery manager required for this system?

i) How is data organized in a hash based index? [2]

j) Give a brief note on Static Hashing. [3]

## PART – B

**(50 Marks)**

2.a) What is a partial key? How is it represented in ER diagram? Give an example.

b) Define query. Explain the data manipulation language in detail. [5+5]

**OR**

3.a) Explain how to build ER model for university with entities department, instructor, student, and class. Instructors and students belong to one department only. Instructors and students related to a class with many to many relations. Assume suitable attributes. Explain how the ER model can be translated to relations.

b) List and explain the design issues of entity relationship. [5+5]

4. Consider the following schema

instructor (ID, name, dept_name),

teaches (ID, course_id, sec_id, semester, year),

section (course_id, sec_id, semester, year),

student (ID, name, dept_name),

takes (ID, course_id, sec_id, semester, year, grade)

Write the following queries in SQL

a) Find the names of the students not registered in any section

b) Find the names of the instructors not teaching any course

c) Find the total number of courses taught department wise

[10]

d) Find the total number of courses registered department wise.

**OR**

5.a) Make a comparison between the tuple relational calculus and domain relational calculus.

b) What are nested queries? What is correlation in nested queries? Explain. [5+5]

6. Discuss how schema refinement through dependency analysis and normalization can improve schemas obtained through ER design. [10]

**OR**

7. Why is a table whose primary key consists of a single attribute automatically in 2NF when it is in 1NF? Explain. [10]

8. Discuss about log based recovery with immediate update and deferred update with suitable examples. [10]

**OR**

9. When a transaction is rolled back under timestamp ordering, it is assigned a new timestamp. Why can it not simply keep its old timestamp? [10]

10.a) Give a brief note on Indexed Sequential Access Methods.

b) Make a comparison between the primary index and a secondary index. [5+5]

**OR**

11. Where does a DBMS store persistent data? How does it bring data into main memory for processing? What DBMS component reads and writes data from main memory, and what is the unit of I/O? [10]

**R15**

**Code No: 124CQ**

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**B.Tech II Year II Semester Examinations, May – 2017**

**DATABASE MANAGEMENT SYSTEMS**

**(Common to CSE, IT)**

**Time: 3 Hours** **Max. Marks: 75**

**Note:** This question paper contains two parts A and B.

Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

## PART - A

**(25 Marks)**

| | | |
|---|---|---|
| 1.a) | What is DBMS? What are the goals of DBMS? | [2] |
| b) | Explain about DDL and DML languages. | [3] |
| c) | Explain views in SQL language. | [2] |
| d) | Explain domain relational calculus. | [3] |
| e) | Define loss less join decomposition with example. | [2] |
| f) | What is the difference between 3NF and BCNF? | [3] |
| g) | What is locking Protocol? | [2] |

d) When are two schedules conflict equivalent? What is conflict serializable schedule? [3]

e) Why are tree-structure indexes are good for searches, especially range selections. [2]

j)What is the main difference between ISAM and B+ tree indexes? [3]

**PART-B**

**(50 Marks)**

2.a) What are the main components in a DBMS and briefly explain what they do.

i) Explain the following:

i) View of Data ii) Data Abstraction iii) Instances and Schemas. [5+5]

**OR**

3.a) Develop ER-Diagram for a hospital with a set of patients and a set of medical doctors. Associated with each patient a log of the various tests and examinations conducted.

2 What is relation? Differentiate between a relation schema and relation instance define the term arity and degree of a relation? What are domain constraints? [5+5]

4.a) Explain the fundamental operations in relational algebra with examples.

7. Explain the following Operators in SQL with examples:

i) SOME ii) IN iii) EXCEPT iv) EXISTS [5+5]

**OR**

5.a) Let R=(ABC) and S=(DEF) let r(R) and s(S) both relations on schema R and S. Give an expression in the Tuple relational calculus that is equivalent to each of the following.

i) $\sigma_{B=19}(r)$ ii) $\prod_{A,F,}(\sigma_{C=D}(r \times s))$ iii) $r \cap s$

8. What are integrity constraints? Define the terms primary key constrains and foreign key constraints. How are these expressed in SQL? [5+5]

6.a) What is normalization? What are the conditions are required for a relation to be in

2NF, 3NF and BCNF explain with examples.

9. Compute the closer of the following set of functional dependencies for a relation scheme.
R(A,B,C,D,E)F={A $\rightarrow$ BC, CD $\rightarrow$ E, B $\rightarrow$ D,E A $\rightarrow$ A}

**OR**

7.a) What are the conditions are required for a relation to be in 4NF and 3NF explain with examples.

10. Compute the closer of the following set of functional dependencies for a relation scheme.
R(A,B,C,D,E,F,G,H), F={ AB $\rightarrow$ C, BD $\rightarrow$ E F, AD $\rightarrow$ G,A $\rightarrow$ H}

List the candidate keys of R. [5+5]

8.a) What is transaction? Explain the ACID Properties of transactions.

12. Explain the Check point log based recovery scheme for recovering the database. [5+5]

**OR**

9.a) Describe the steps in crash recovery in ARIES.

b) Explain the *Time Stamp - Based Concurrency* Control protocol. [5+5]

10.a) Explain Deletion and insertion operations in ISAM with examples.

b) How does *Extendable hashing* use a directory of buckets? How does it handles insert and delete operations. [5+5]

**OR**

11.a) Explain how insert and delete operations are handled in a static hash index.

b) Explain deletion and insertion operation in *B+ trees*. [5+5]

**R13**

**Code No: 114CQ**

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**B.Tech II Year II Semester Examinations, October/November – 2016**

**DATABASE MANAGEMENT SYSTEMS**

**(Common to CSE, IT)**

**Time: 3 Hours** **Max. Marks: 75**

**Note:** This question paper contains two parts A and B.

Part A is compulsory which carries 25 marks. Answer all questions in Part A.

Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

**PART - A** **(25 Marks)**

1.a) List the properties of ER diagram. [2]

b) Explain the three levels of abstraction. [3]

c) Explain integrity constraints over relations. [2]

f) Create a table with employee details like eno, ename, bdate, address, dno, age, phone number. List the name. eno, dname and phone number of the employee who are also the managers of the respective departments. [3]

e) What is functional dependency? [2]

f) How can we identify that the relation is in 2NF? [3]

g) Write about transaction states. [2]

h) What are ACID properties? Explain. [3]

i) What is an index? Give an example. [2]

j) What are the advantages of using tree structured indexes? [3]

**PART - B** **(50 Marks)**

2.a) What is a data model? What are the different data models? Explain E-R model and relation model briefly.

b) Explain database users, user interfaces, DBA and functions of a DBA. [5+5]

**OR**

3.a) What are the application programs? Explain database access for application programs.

j) What is null attribute? With suitable diagram explain weak and strong entity set. [5+5]

4.a) Discuss in detail about the properties of relation algebra.

4.b) How we can convert relationship sets with key constraints into tables? Explain. [5+5]

**OR**

5.a) Write short notes on difference, union, rename and Cartesian product operations in relational algebra.

b) How we can translate E-R diagram with aggregation? Explain. [5+5]

6.a) Explain different normal forms based on functional dependencies.

b) Explain about dependency preserving decomposition. [5+5]

**OR**

7.a) Explain BCNF. Give an example.

8. What are the steps to be followed to convert a relation in 3NF to BCNF? [5+5]

8.a) Explain ARIES in detail.

b) How the lock manager implements lock and unlock requests? Explain. [5+5]

**OR**

9.a) How the concurrency control is done in B+ trees? Explain.

b) What is schedule? Explain about serial and non serial schedule. [5+5]

10.a) What is a composite search key? What are the pros and cons of composite search keys?

b) What are the performance implications of disk structure? Explain. [5+5]

**OR**

11.a) What are the different RAID levels? Explain.

b) Compare linear hashing and extendable hashing. [5+5]

**R13**

**Code No: 114CQ**

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**B.Tech II Year II Semester Examinations, October/November - 2016**

**DATABASE MANAGEMENT SYSTEMS**

**(Common to CSE, IT)**

**Time: 3 Hours** **Max. Marks: 75**

**Note:** This question paper contains two parts A and B.

Part A is compulsory which carries 25 marks. Answer all questions in Part A.

Part B consists of 5 Units. Answer any one full question from each unit.
Each question carries 10 marks and may have a, b, c as sub questions.

**PART - A** **(25 Marks)**

| | | |
|---|---|---|
| 1.a) | List the properties of ER diagram. | [2] |
| b) | Explain the three levels of abstraction. | [3] |
| c) | Explain integrity constraints over relations. | [2] |
| g) | Create a table with employee details like eno, ename, bdate, address, dno, age, phone number. List the name. eno, dname and phone number of the employee who are also the managers of the respective departments. | [3] |
| e) | What is functional dependency? | [2] |

f) How can we identify that the relation is in 2NF? [3]

g) Write about transaction states. [2]

h) What are ACID properties? Explain. [3]

i) What is an index? Give an example. [2]

j) What are the advantages of using tree structured indexes? [3]

**PART - B** **(50 Marks)**

2.a) What is a data model? What are the different data models? Explain E-R model and relation model briefly.

b) Explain database users, user interfaces, DBA and functions of a DBA. [5+5]

**OR**

3.a) What are the application programs? Explain database access for application programs.

b) What is null attribute? With suitable diagram explain weak and strong entity set. [5+5]

4.a) Discuss in detail about the properties of relation algebra.

4.b) How we can convert relationship sets with key constraints into tables? Explain. [5+5]

**OR**

5.a) Write short notes on difference, union, rename and Cartesian product operations in relational algebra.

b) How we can translate E-R diagram with aggregation? Explain. [5+5]

6.a) Explain different normal forms based on functional dependencies.

b) Explain about dependency preserving decomposition. [5+5]

**OR**

7.a) Explain BCNF. Give an example.

9. What are the steps to be followed to convert a relation in 3NF to BCNF? [5+5]

8.a) Explain ARIES in detail.

b) How the lock manager implements lock and unlock requests? Explain. [5+5]

**OR**

9.a) How the concurrency control is done in B+ trees? Explain.

b) What is schedule? Explain about serial and non serial schedule. [5+5]

10.a) What is a composite search key? What are the pros and cons of composite search keys?

b) What are the performance implications of disk structure? Explain. [5+5]

**OR**

11.a) What are the different RAID levels? Explain.

b) Compare linear hashing and extendable hashing. [5+5]

**Code No: 114CQ** **R13**

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD**

**B.Tech II Year II Semester Examinations, May-2015**

**DATABASE MANAGEMENT SYSTEMS**

**(Common to CSE, IT)**

**Time: 3 Hours** **Max. Marks: 75**

**Note:** This question paper contains two parts A and B.

Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

**PART- A** **(25 Marks)**

| | | |
|---|---|---|
| 1.a) | Differentiate between schema and data model. | [2M] |
| b) | Give an example for total participation and partial participation. | [3M] |
| c) | List the primitive operators in Relational Algebra. | [2M] |
| d) | What is an active database? | [3M] |
| e) | Define SECOND Normal form. | [2M] |
| f) | Write about join dependencies. | [3M] |
| g) | What methods are used to assign timestamps to transactions? | [2M] |
| h) | What is the significance remote backup system? | [3M] |

i) What is meant by secondary index? [2M]

j) How to compute the disk access time? [3M]

**PART - B** **(50 Marks)**

2.a) List various categories of database users and discuss their interfaces to DBMS.

b) Discuss the functionality of query evaluation engine. [5+5]

**OR**

h) Construct an Entity-Relationship diagram for a online shopping systems such as Jabong/Flipcart. Quote your assumptions and list the requirements considered by you for conceptual database design for the above system. [10]

4.a) With a suitable example explain division operation in relational algebra.

b) What is the usage of 'group by' and 'having' clauses in SQL? [5+5]

**OR**

l) Consider the following schema to write queries in Domain relational calculus: Sailor(sid, sname, age, rating)
Boats(bid, bname, bcolor)
Reserves(sid,bid,day)

Find the boats reserved by sailor with id 567.
Find the names of the sailors who reserved 'red' boats.

5 What is meant by closure of F? Where F is the set of functional dependencies.

c) Find the boats which have at least two reservations by different sailors. [10]

Explain computing F+ with suitable examples. [10]

**OR**

7.a) Differentiate bet een FD and MVD.

b) Explain the problems related to decomposition. [5+5]

a) Explain transaction states and desirable properties.

b) How to test serializability of a schedule? Explain with an example. [5+5]

**OR**

9.a) Explain Failure classification.

b) What is log? What is log tail? Explain the concept of checkpoint log record. [5+5]

10. Explain extendable hashing techniques for indexing data records. Consider your class students data records and roll number as index attribute and show the hash directory.

[10]

**OR**

11.a) Is disk cylinder a logical concept? Justify your answer.

b) Compare heap file organization with hash file organization. [5+5]

**---ooOoo--**

## 12. MID Exam Descriptive Question Papers

## 13. MID Exam Objective

## 14. Assignment topics with material

### DBMS Assignment Questions

### DBMS Assignment Questions

**UNIT-1**

**1. DATABASE SYSTEMAPPLICATIONS:** The following are the various kinds of applications/organizations uses databases for their business processing activities in their day-to-day life. They are:

1. Banking: For customer information, accounts, and loans, and banking transactions.
2. Airlines: For reservations and schedule information. Airlines were among the first to use databases in a geographically distributed manner—terminals situated around the world accessed the central database system through phone lines and other data networks.
3. Universities: For student information, course registrations, and grades.
4. Credit Card Transactions: For purchases on credit cards and generation of monthly statements.
5. Telecommunication: For keeping records of calls made, generating monthly bills, maintaining balances on prepaid calling cards, and storing information about the communication networks.
6. Finance: For storing information about holdings, sales, and purchases of financial instruments such as stocks and bonds.
7. Sales: For customer, product, and purchase information.
8. Manufacturing: For management of supply chain and for tracking production of items in factories, inventories of items in warehouses/stores, and orders for items.
9. Human resources: For information about employees, salaries, payroll taxes and benefits, and for generation of paychecks.
10. Railway Reservation Systems: For reservations and schedule information
11. Web: For access the Back accounts and to get the balance amount.

12. E –Commerce: For Buying a book or music CD and browse for things like watches, mobiles from internet

## 2. PURPOSE OF DATABASE SYSTEMS:

**Structured and Described Data:**

Fundamental feature of the database approach is that the database system does not only contain the data but also the complete definition and description of these data. These descriptions are basically details about the extent, the structure, the type and the format of all data and, additionally, the relationship between the data. This kind of stored data is called metadata ("data about data").

**Separation of Data and Applications:**

Application software does not need any knowledge about the physical data storage like encoding, format, storage place, etc. It only communicates with the management system of a database (DBMS) via a standardized interface with the help of a standardized language like SQL. The access to the data and the metadata is entirely done by the DBMS. In this way all the applications can be totally separated from the data.

**Data Integrity:**

Data integrity is a byword for the quality and the reliability of the data of a database system. In a broader sense data integrity includes also the protection of the database from unauthorized access (confidentiality) and unauthorized changes. Data reflect facts of the real world.

**Transactions:**

A transaction is a bundle of actions which are done within a database to bring it from one consistent state to a new consistent state. In between the data are inevitable inconsistent. *A* transaction is atomic what means that it cannot be divided up any further. Within a transaction all or none of the actions need to be carried out. Doing only a part of the actions would lead to an inconsistent database state.

## 3. VIEW OF DATA:

Abstraction is one of the main features of database systems. Hiding irrelevant details from user and providing abstract view of data to users, helps in easy and efficient **user-database** interaction.

## 4. DATABASE LANGUAGES:

Database languages are used for read, update and store data in a database. There are several such languages that can be used for this purpose; one of them is SQL (Structured Query Language).

**Data Definition Language (DDL)**: DDL is used for specifying the database schema. Let's take SQL for instance to categorize the statements that comes under DDL.

To create the database instance – CREATE

To alter the structure of database – ALTER

To drop database instances – DROP

To delete tables in a database instance – TRUNCATE

To rename database instances – RENAME

All these commands specify or update the database schema that's why they come under Data Definition language.

**Data Manipulation Language (DML)**: DML is used for accessing and manipulating data in a database.

To read records from table(s) – SELECT

To insert record(s) into the table(s) – INSERT

Update the data in table(s) – UPDATE

Delete all the records from the table – DELETE

## 5. RELATIONAL DATA BASE:

A relational database management system (RDBMS) is a collection of programs and capabilities that enable IT teams and others to create, update, administer and otherwise interact with a relational database. Most commercial RDBM use Structured Query Language (SQL) to access the database, although SQL was invented after the initial development of the relational model and is not necessary for its use.

## UNIT-2

### 1. PRELIMINARIES:

A query language is a language in which user requests to retrieve some information from the database. The query languages are considered as higher level languages than programming languages. Query languages are of two types,

Procedural Language

Non-Procedural Language

1. In procedural language, the user has to describe the specific procedure to retrieve the information from the database.

*Example:* The Relational Algebra is a procedural language.

2. In non-procedural language, the user retrieves the information from the database without describing the specific procedure to retrieve it.

*Example:* The Tuple Relational Calculus and the Domain Relational Calculus are non-procedural languages.

### 2. RELATIONAL ALGEBRA:

The relational algebra is a procedural query language. It consists of a set of operations that take one or two relations (tables) as input and produce a new relation, on the request of the user to retrieve the specific information, as the output.

### 3. RELATIONAL CALCULUS:

Relational calculus is an alternative to relational algebra. In contrast to the algebra, which is procedural, the relational calculus is non-procedural or declarative.

It allows user to describe the set of answers without showing procedure about how they should be computed. Relational calculus has a big influence on the design of commercial query languages such as SQL and QBE (Query-by Example).

Relational calculus are of two types,

Tuple Relational Calculus (TRC)

Domain Relational Calculus (DRC)

Variables in TRC takes tuples (rows) as values and TRC had strong influence on SQL.

Variables in DRC takes fields (attributes) as values and DRC had strong influence on QBE.

i) Tuple Relational Calculus (TRC):

The tuple relational calculus, is a non-procedural query language because it gives the desired information without showing procedure about how they should be computed.

A query in Tuple Relational Calculus (TRC) is expressed as { T | p(T) }

Where, T - tuple variable,

P(T) - 'p' is a condition or formula that is true for 't'.

In addition to that we use,

T[A] - to denote the value of tuple t on attribute A and

T □□r - to denote that tuple t is in relation r.

ii) Domain Relational Calculus (DRC):

A Duple Relational Calculus (DRC) is a variable that comes in the range of the values of domain (data types) of some columns (attributes).

A Domain Relational Calculus query has the form, { < x1, x2, …., xn > | p( < x1, x2, …., xn > ) }

Where, each xi is either a domain variable or a constant and p(< x1, x2, …., xn >) denotes a DRC formula.

A DRC formula is defined in a manner that is very similar to the definition of a TRC formula. The main difference is that the variables are domain variables.

## 4. EXPRESSIVE POWER OF ALGEBRA AND CALCUS:

The tuple relational calculus restricts to safe expressions and is equal in expressive power to relational algebra. Thus, for every relational algebra expression, there is an equivalent expression in the tuple relational calculus and for tuple relational calculus expression, there is an equivalent relational algebra expression.

A safe TRC formula Q is a formula such that,

For any given I, the set of answers for Q contains only values that are in dom(Q, I).

For each sub expression of the form □R(p(R)) in Q, if a tuple r makes the formula true, then r contains

When the domain relational calculus is restricted to safe expression, it is equivalent in expressive power to the tuple relational calculus restricted to safe expressions. All three of the following are equivalent,

The relational algebra

The tuple relational calculus restricted to safe expression

The domain relational calculus restricted to safe expression

**UNIT-3**

## 1. INTRODUCTION TO SCHEMA REFINEMENT:

The fourth step in database design is to analyze the collection of relations in our relational database schema to identify potential problems, and to refine it. In contrast to the requirements analysis and conceptual design steps, which are essentially subjective, schema refinement can be guided by some elegant and powerful theory.

Data redundancy means duplication of data. It causes duplicate data at different locations which destroys the integrity of the database and wastage of storage space.

The problems of redundancy are:

1.Wasted Storage Space. 2. More difficult Database Updates. 3. A Possibility of Inconsistent data.

## 2. FUNCTIONAL DEPENDENCY:

The normalization theory based on the fundamental notion of FUNCTIONAL DEPENDENCY. Given a relation R, attribute A is functionally dependent on attribute B if each value of value of A in R is associated with precisely one value of B.

(OR)

In other words, attribute A is functionally dependent on B if and only if, for each value of B, there is exactly one value of A.

(OR)

Functional dependency is a relationship that exists when one attribute uniquely determines another attribute. If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as X->Y, which specifies Y is functionally dependent on X.

Here X is a determinant set and Y is a dependent attribute. Each value of X is associated precisely with one Y value.

3. **Rules about Functional Dependencies**

The set of all FD's implied by a given set F of FD's is called the closure of F, denoted as F+. The following three rules, called Armstrong's axioms, can be applied repeatedly to compute all FDs implied by a set of FDs. Here, we use X, Y and Z to denote sets of attribute over a relation schema R,

Rule 1: Reflexivity: if X □Y, then X □□Y.

Rule 2: Augmentation: if X □□Y, then XZ □□YZ for any Z.

Rule 3: Transitivity: if X □□Y and Y □□Z, then X □□Z.

It is convenient to use some additional rules while reasoning about F+.

Rule 4: Union: If x □□Y and X □□z, then X □□YZ.

Rule 5: Decomposition: If X □□YZ, then X □□Y and X □□Z.

Rule 6: Pseudotranstivity: If X □□Y and Y □□P, then XZ □□P.

4. **INTRODUCTION TO NORMALIZATION:**

Redundancy:

- Redundancy means repetition of data.
- Redundancy increases the time involved in updating, adding, and deleting data.
- It also increases the utilization of disk space and hence, disk I/O increases.

DEFINITION OF NORMALIZATION:

- Normalization is scientific method of breaking down complex stable structures into simple table structures by using certain rules.
- Using this method, you can, reduce redundancy in a table and elimates the problems of inconsistency and disk space usage.
- We can also ensure that there is no loss of information.

Benefits of Normalization:

- Normalization has several benefits.

- It enables faster sorting and index creation, more clustered indexes, few indexes per table, few NULLs, and makes the database compact.
- Normalization helps to simplify the structure of table. The performance of an application is directly linked to database design.
- A poor design hinders the performance of the system.
- The logical design of the database plays the foundation of an optical database.

The following are the some rules that should be followed to achieve a good data base design .They are:

Each table should have an identifier.

Each table should store data for single type entity.

Columns that accept NULLs should be avoided.

The repetition of values or columns should be avoided.

**5. NORMAL FORMS:**

The normalization results in the formation of that satisfy certain specified rules and represent certain normal forms.

The normal forms are used to ensure that several of anomalies and inconsistencies are not introduced in the database.

A table structure is always in a certain normal form. Several normal forms have been identified.

The following are the most important and widely used normal forms are:

First normal form(1NF)

Second normal form(2NF)

Third normal form(3NF)

Boyce-Codd Normal Form (BCNF)

Fourth Normal Form (4NF)

The first, second and third normal forms are originally defined by Dr. E. F. Codd.

Later, Boyce and Codd introduced another form called the Boyce-Codd Normal form.

## Unit4

**1. TRANSACTIONS:**

A transaction is a unit of program execution that accesses and possibly updates various data items.

(or)

A transaction is an execution of a user program and is seen by the DBMS as a series or list of actions i.e., the actions that can be executed by a transaction includes the reading and writing of database.

Transaction Operations:

Access to the database is accomplished in a transaction by the following two operations,

1) read(X) : Performs the reading operation of data item X from the database.

2) write(X) : Performs the writing operation of data item X to the database.

Example:

Let T1 be a transaction that transfers $50 from account A to account B. This transaction can be illustrated as follows,

T1 : read(A);

A := A – 50;

write(A);

read(B);

B := B + 50;

write(B);

**2. Transaction Concept:**

- The concept of transaction is the foundation for concurrent execution of transaction in a DBMS and recovery from system failure in a DBMS.
- A user writes data access/updates programs in terms of the high-level query language supported by the DBMS.
- To understand how the DBMS handles such requests, with respect to concurrency control and recovery, it is convenient to regard an execution of a user program or transaction, as a series of reads and writes of database objects.

- To read a database object, it is first brought in to main memory from disk and then its value is copied into a program. This is done by read operation.
- To write a database object, in-memory, copy of the object is first modified and then written to disk. This is done by the write operation.

**3. Serializability:**

Schedule − A chronological execution sequence of a transaction is called a schedule. A schedule can have many transactions in it, When multiple transactions are being executed by the operating system in a multiprogramming environment, there are possibilities that instructions of one transaction are interleaved with some other transaction.

each comprising of a number of instructions/tasks.

Serial Schedule − It is a schedule in which transactions are aligned in such a way that one transaction is executed first. When the first transaction completes its cycle, then the next transaction is executed. Transactions are ordered one after the other. This type of schedule is called a serial schedule, as transactions are executed in a serial manner.

**4 .CONCURRENCY CONTROL**

In a multiprogramming environment where multiple transactions can be executed simultaneously, it is highly important to control the concurrency of transactions.

We have concurrency control protocols to ensure atomicity, isolation, and serializability of concurrent transactions.

Why DBMS needs a concurrency control?

In general, concurrency control is an essential part of TM. It is a mechanism for correctness when two or more database transactions that access the same data or data set are executed concurrently with time overlap. According to Wikipedia.org, if multiple transactions are executed serially or sequentially, data is consistent in a database. However, if concurrent transactions with interleaving operations are executed, some unexpected data and inconsistent result may occur. Data interference is usually caused by a write operation among transactions on the same set of data in DBMS. For example, the lost update problem may

occur when a second transaction writes a second value of data content on top of the first value written by a first concurrent transaction. Other problems such as the dirty read problem, the incorrect summary problem

Concurrency Control Techniques:

The following techniques are the various concurrency control techniques. They are:

concurrency control by Locks

Concurrency Control by Timestamps

Concurrency Control by Validation

## 5. TIMESTAMP ORDERING PROTOCOL:

This protocol guarantees that the execution of read and write operations that are conflicting is done in timestamp order.

Working of Timestamp Ordering Protocol:

The Time stamp ordering protocol ensures that any conflicting read and write operations are executed in time stamp order. This protocol operates as follows:

1) If TA executes read(x) instruction, then the following two cases must be considered,

i) TS(TA) < WTS(x)

ii) TS(TA) WTS(x)

Case 1 : If a transaction TA wants to read the initial value of some data item x that had been overwritten by some younger transaction then, the transaction TA cannot perform the read operation and therefore the transaction must be rejected. Then the transaction TA must be rolled back and restarted with a new timestamp.

Case 2 : If a transaction TA wants to read the initial value of some data item x that had not been updated then the transaction can execute the read operation. Once the value has read, changes occur in the read timestamp value (RTS(x)) which is set to the largest value of RTS(x) and TS

2) If TA executes write(x) instruction, then the following two cases must be considered,

i) TS(TA) < RTS(x)

ii) TS(TA) < WTS(x)

iii) TS(TA) >WTS(x)

Case 1 : If a transaction TA wants to write the value of some data item x on which the read operation has been performed by some younger transaction, then the transaction cannot execute the write operation. This is because the value of data item x that is being generated by TA was required previously and therefore, the system assumes that the value will never be generated. The write operation is thereby rejected and the transaction TA must be rolled back and should be restarted with new timestamp value.

Case 2 : If a transaction TA wants to write a new value to some data item x, that was overwritten by some younger transaction, then the transaction cannot execute the write operation as it may lead to inconsistency of data item. Therefore, the write operation is rejected and the transaction should be rolled back with a new timestamp value.

Case 3 : If a transaction TA wants to write a new value on some data item x that was not updated by a younger transaction, then the transaction can executed the write operation. Once the value has been written, changes occur on WTS(x) value which is set to the value of TS(TA).

**Unit5**

## 1. COMPARISION FILE ORGANIZATIONS:

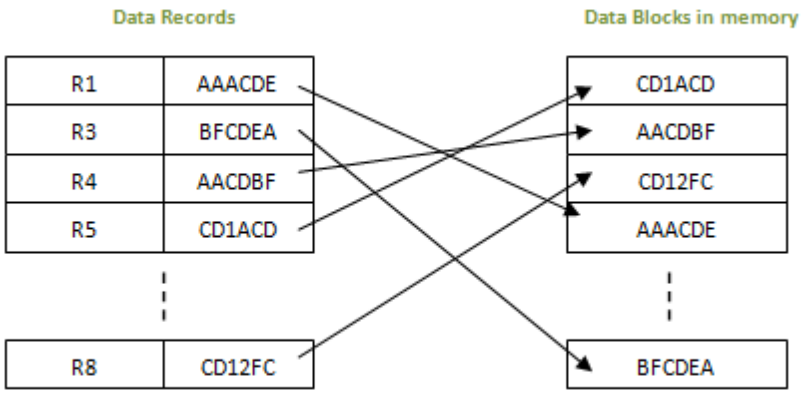
As we know already, database consists of tables, views, index, procedures, functions etc. The tables and views are logical form of viewing the data. But the actual data are stored in the physical memory. Database is a very huge storage mechanism and it will have lots of data and hence it will be in physical storage devices. In the physical memory devices, these datas cannot be stored as it is. They are converted to binary format. Each memory devices will have many data blocks, each of which will be capable of storing certain amount of data. The data and these blocks will be mapped to store the data in the memory.

Any user who wants to view these data or modify these data, simply fires SQL query and gets the result on the screen. But any of these queries should give results as fast as possible. But how these data are fetched from the physical memory? Do you think simply storing the data in memory devices give us the better results when we fire queries? Certainly not. How is it stored in the memory, Accessing method, query type etc makes great affect on getting the results. Hence organizing the data in the database and hence in the memory is one of important think about.

## 2. ISAM(INDEX SEQUENTIAL ACCESS METHOD):

This is an advanced sequential file organization method. Here records are stored in order of primary key in the file. Using the primary key, the records are sorted. For each primary key, an index value is generated and mapped with the record. This index is nothing but the address of record in the file.

In this

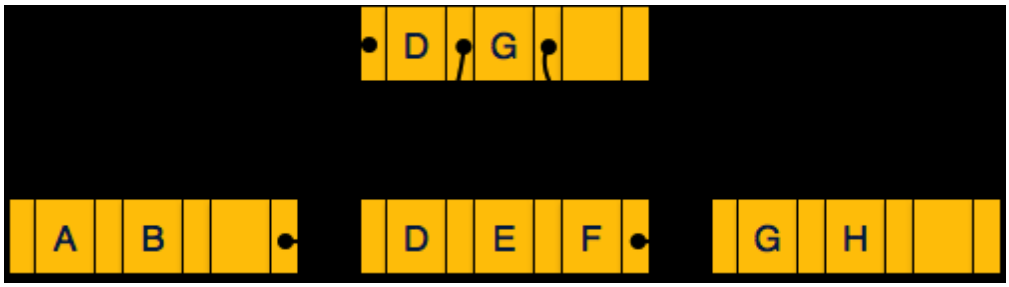



**3. B+ Tree:**

A B+ tree is a balanced binary search tree that follows a multi-level index format. The leaf nodes of a B+ tree denote actual data pointers. B+ tree ensures that all leaf nodes remain at the same height, thus balanced. Additionally, the leaf nodes are linked using a link list; therefore, a B+ tree can support random access as well as sequential access.

Structure of B+ Tree

Every leaf node is at equal distance from the root node. A B+ tree is of the ordern where n is fixed for every B+ tree.



**4. HASHING:**

 Bucket − A hash file stores data in bucket format. Bucket is considered a unit of storage. A bucket typically stores one complete disk block, which in turn can store one or more records.

 Hash Function − A hash function, h, is a mapping function that maps all the set of search-keys K to the address where actual records are placed. It is a function from search keys to bucket addresses.

Dept. of CSE, CREC Page 111

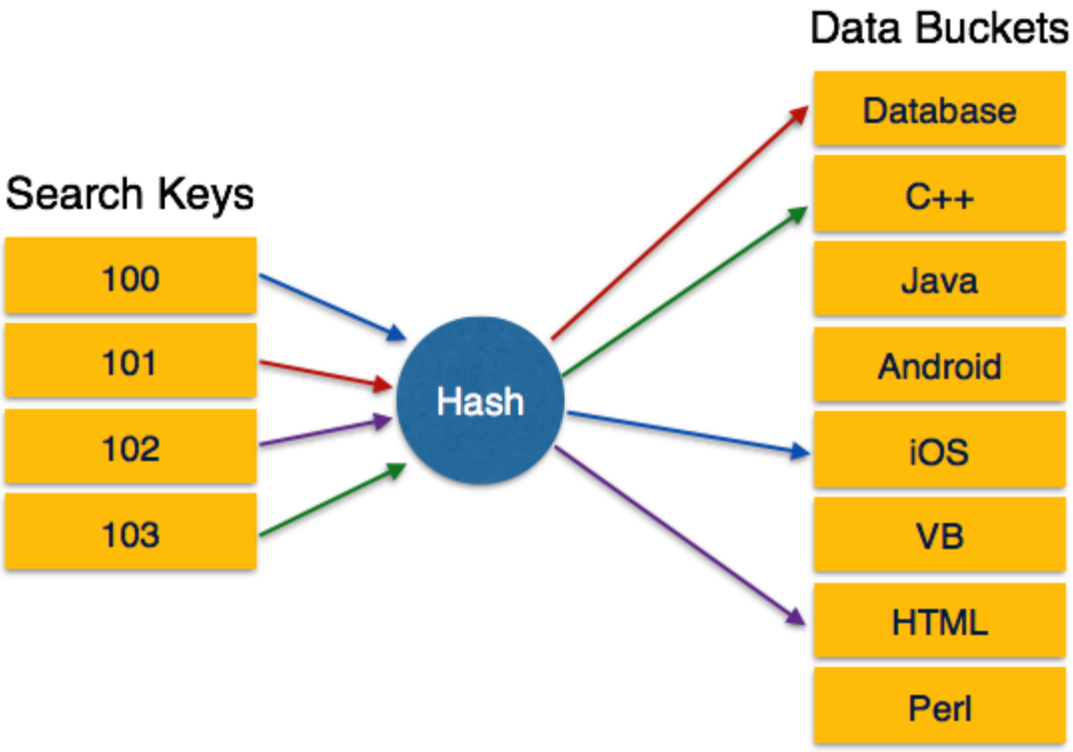
HASHING:


 Bucket − A hash file stores data in bucket format. Bucket is considered a unit of storage. A bucket typically stores one complete disk block, which in turn can store one or more records.

 Hash Function − A hash function, h, is a mapping function that maps all the set of search-keys K to the address where actual records are placed. It is a function from search keys to bucket addresses.


4.Static Hashing

In static hashing, when a search-key value is provided, the hash function always computes the same address. For example, if mod-4 hash function is used, then it shall generate only 5 values. The output address shall always be same for that function. The number of buckets provided remains unchanged at all times.




**5. LINEAR HASHING:**

Hash table is a data structure that associates keys with values. To know more about liner hashing refers Wikipedia. Here are main points that summarize linear hashing.

Full buckets are not necessarily split

Buckets split are not necessarily full

Every bucket will be split sooner or later and so all Overflows will be reclaimed and rehashed.

Split pointer s decides which bucket to split s is independent to overflowing bucket

## 15. Tutorial topics with Questions

## 16. Unit Wise-Question Bank

### i. Two Marks questions with Answers-5questions

### ii. Three marks questions with answers-5questions

iii. **Five marks questions with answers-5questions**

iv. **Objective questions with answers-10questions**

v. **Fill in the blanks with answers-10 questions**

## UNITWISE-QUESTION BANK

### UNIT-1

**TWO MARKS QUESTIONS WITH ANSWER**

1. What are five main functions of a database administrator?

Schema definition

Storage structure and access methods

Schema and physical organization modification

Granting of authorization for data access

Integrity constraint specification

2. What is DBMS? What are the goals of DBMS?

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified -- and the database schema, which defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

3. Explain the transaction management in database

**Transaction Management**. **A transaction** is one or more SQL statements that make up a unit of work performed against the database, and either all the statements in **a transaction** are committed as a unit or all the statements are rolled back as a unit.

4. Explain the relation between weak entity and strong enity?

The **Strong Entity** is the one whose existence does not depend on the existence of any other entity in a schema. It is denoted by a **single rectangle**. A strong entity always has the **primary key** in the set of attributes that describes the strong entity. It indicates that each entity in a strong entity set can be uniquely identified.

A **Weak entity** is the one that depends on its owner entity i.e. a strong entity for its existence. A weak entity is denoted by the **double rectangle**. Weak entity do **not** have the **primary key** instead it has a **partial key** that uniquely discriminates the weak entities. The **primary key of a weak entity** is a composite key formed from the **primary key of the strong entity** and **partial key of the weak entity**.

5. Explain the relationship and relationship set

**A relationship** is an association between several entities. **A relationship set** is a**set** of **relationships** of the same type. Formally it is a mathematical **relation** on (possibly non-distinct) **sets**.

THREE MARKS QUESTIONS WITH ANSWERS

1. List and explain the database system applications.

A **database**-management **system** (DBMS) is a computer-software **application** that interacts with end-users, other **applications**, and the **database**itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of **databases**.

2. Explain about DDL and DML languages.

**DML** stands for Data Manipulation **Language**. The schema (Table) created by **DDL** (Data Definition**Language**) is populated or filled using Data Manipulation **Language**. **DDL** fill the rows of the table, and each row is called Tuple. Using **DML**, you can insert, modify, delete and retrieve the information

3. What is the relational database query

SQL (Structured **Query** Language) is a programming language used to communicate with data stored in a **relational database management system**. For example, SQLite is a **relational database management system**. SQL it e contains a minimal set of SQL commands

4. Explain the E-R diagram component

Entities,which are represented by rectangles

Actions, which are represented by diamond shapes, show how two entities share information in the database

Attributes, which are represented by ovals.

What is a relation?

In relational **database** theory, a **relation**, as originally defined by E. F. Codd, is a set of tuples ($d_1$, $d_2$, ..., $d_n$), where each element $d_j$ is a member of $D_j$, a data domain. ... A set of tuples having the same heading is called a body.

FIVE MARKS QUESTIONS WITH ANSWERS

What is a partial key? How is it represented in ER diagram? Give an example.

A **partial key** means just part of a **key** - some proper subset of the **key** attributes. In your example if the primary **key** of a Child was (Empid, ChildName) with Empid as a foreign **key** referencing the Employee then Child is a weak entity.

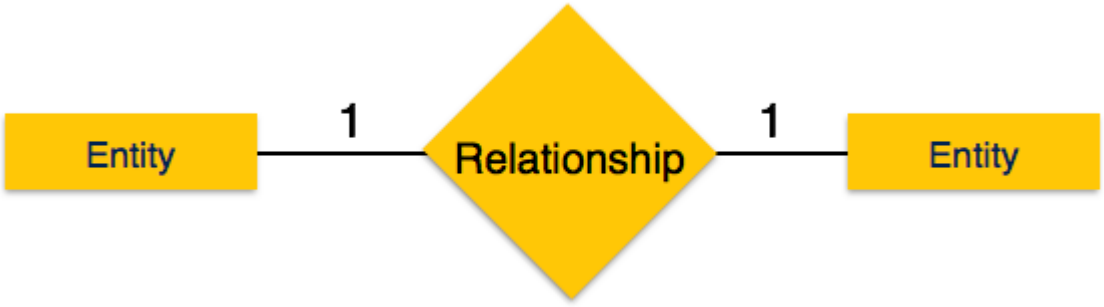
Entities are represented by means of rectangles. Rectangles are named with the entity set they represent. Relationship

Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box. All the entities (rectangles) participating in a relationship, are connected to it by a line.

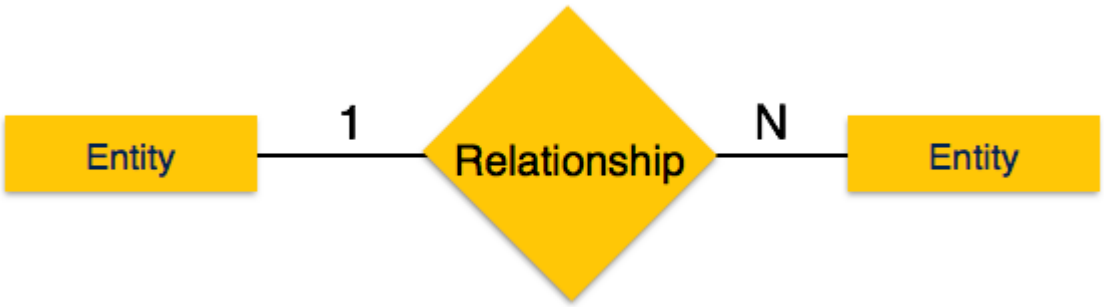
Binary Relationship and Cardinality

A relationship where two entities are participating is called a **binary relationship**. Cardinality is the number of instance of an entity from a relation that can be associated with the relation.

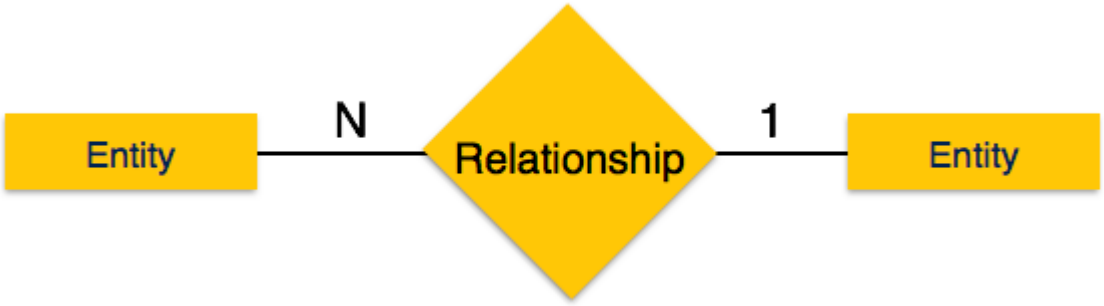
**One-to-one** − When only one instance of an entity is associated with the relationship, it is marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts one-to-one relationship.



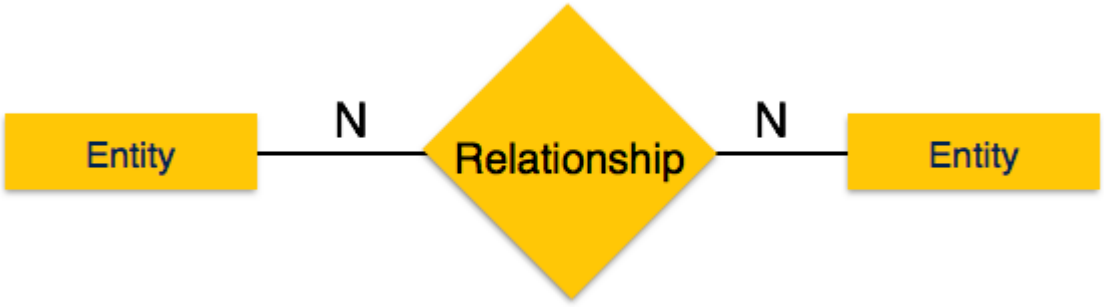
**One-to-many** − When more than one instance of an entity is associated with a relationship, it is marked as '1:N'. The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts one-to-many relationship.




**Many-to-one** − When more than one instance of entity is associated with the relationship, it is marked as 'N:1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts many-to-one relationship.




**Many-to-many** − The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts many-to-many relationship.




Define query. Explain the data manipulation language in detail.

**Data-Manipulation Language**

Data manipulation is

The retrieval of information stored in the database

The insertion of new information into the database

The deletion of information from the database

The modification of information stored in the database

A **data-manipulation language (DML)** is a language that enables users to access or manipulate data as organized by the appropriate data model. There are basically two types:

**Procedural DMLs** require a user to specify *what* data are needed and *how* to get those data.

**Declarative DMLs** (also referred to as **nonprocedural** DMLs) require a user to specify *what* data are needed *without* specifying how to get those data.

Declarative DMLs are usually easier to learn and use than are procedural DMLs. However, since a user does not have to specify how to get the data, the database system has to figure out an efficient means of accessing data. The DML component of the SQL language is nonprocedural.

A **query** is a statement requesting the retrieval of information. The portion of a DML that involves information retrieval is called a **query language**. Although technically incorrect, it is common practice to use the terms *query language* and *data manipulation language* synonymously.

This query in the SQL language finds the name of the customer whose customer-id is 192-83-7465:

**select** *customer.customer-name*

**from** *customer*

**where** *customer.customer-id* = 192-83-7465 The query specifies that those rows *from* the table *customer where* the *customer-id* is 192-83-7465 must be retrieved, and the *customer-name* attribute of these rows must be displayed.

Queries may involve information from more than one table. For instance, the following query finds the balance of all accounts owned by the customer with customerid 192-83-7465.

**select** *account.balance*

**from** *depositor, account*

**where** *depositor.customer-id* = 192-83-7465 **and**

*depositor.account-number = account.account-number*

There are a number of database query languages in use, either commercially or experimentally.

The levels of abstraction apply not only to defining or structuring data, but also to manipulating data. At the physical level, we must define algorithms that allow efficient access to data. At higher levels of abstraction, we emphasize ease of use. The goal is to allow humans to interact efficiently with the system. The query processor component of the database system translates DML queries into sequences of actions at the physical level of the database system.

What are the main components in a DBMS and briefly explain what they do.

The two main types of data dictionary exist, integrated and stand alone. An integrated data dictionary is included with the DBMS. For example, all relational DBMSs include a built in data dictionary or system catalog that is frequently accessed and updated by the RDBMS. Other DBMSs especially older types, do not have a built in data dictionary instead the DBA may use third party stand alone data dictionary systems.

Data dictionaries can also be classified as active or passive. An active data dictionary is automatically updated by the DBMS with every database access, thereby keeping its

Explain the following: i) View of Data ii) Data Abstraction iii) Instances and Schemas.

**Data independence:** Application programs should be as independent as possible from details of data representation and storage. The DBMS can provide an abstract view of the data to insulate application code from such details.

**Efficient data access:** A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important if the data is stored on external storage devices.

**Data integrity and security:** If data is always accessed through the DBMS, the DBMS can enforce integrity constraints on the data. For example, before inserting salary information for an employee, the DBMS can check that the department budget is not exceeded. Also, the DBMS can enforce *access controls* that govern what data is visible to different classes of users.

8 Smartworld.asia Specworld.in Smartzworld.

**UNIT-2**

TWO MARKS QUESTIONS WITH ANSWERS

1. Define a trigger. What are the differences between row level and statement level triggers?

A query language is a language in which user requests to retrieve some information from the database. The query languages are considered as higher level languages than programming languages. Query languages are of two types,

Procedural Language

Non-Procedural Language

1. In procedural language, the user has to describe the specific procedure to retrieve the information from the database.

*Example:* The Relational Algebra is a procedural language.

2. In non-procedural language, the user retrieves the information from the database without describing the specific procedure to retrieve it.

*Example:* The Tuple Relational Calculus and the Domain Relational Calculus are non-procedural languages.

2. Explain views in SQL language.

The relational algebra is a procedural query language. It consists of a set of operations that take one or two relations (tables) as input and produce a new relation, on the request of the user to retrieve the specific information, as the output.

3. What SQL construct enables the definition of a relation?

Relational calculus is an alternative to relational algebra. In contrast to the algebra, which is procedural, the relational calculus is non-procedural or declarative.

It allows user to describe the set of answers without showing procedure about how they should be computed. Relational calculus has a big influence on the design of commercial query languages such as SQL and QBE (Query-by Example).

Relational calculus are of two types,

Tuple Relational Calculus (TRC)

Domain Relational Calculus (DRC)

Variables in TRC takes tuples (rows) as values and TRC had strong influence on SQL.

Variables in DRC takes fields (attributes) as values and DRC had strong influence on QBE.

i) Tuple Relational Calculus (TRC):

The tuple relational calculus, is a non-procedural query language because it gives the desired information without showing procedure about how they should be computed.

A query in Tuple Relational Calculus (TRC) is expressed as { T | p(T) }

Where, T - tuple variable,

P(T) - 'p' is a condition or formula that is true for 't'.

In addition to that we use,

T[A] - to denote the value of tuple t on attribute A and

T □□r - to denote that tuple t is in relation r.

ii) Domain Relational Calculus (DRC):

A Duple Relational Calculus (DRC) is a variable that comes in the range of the values of domain (data types) of some columns (attributes).

A Domain Relational Calculus query has the form, { < x1, x2, …., xn > | p( < x1, x2, …., xn > ) }

Where, each xi is either a domain variable or a constant and p(< x1, x2, …., xn >) denotes a DRC formula.

A DRC formula is defined in a manner that is very similar to the definition of a TRC formula. The main difference is that the variables are domain variables.

4. What is a relational database query?

The tuple relational calculus restricts to safe expressions and is equal in expressive power to relational algebra. Thus, for every relational algebra expression, there is an equivalent expression in the tuple relational calculus and for tuple relational calculus expression, there is an equivalent relational algebra expression.

A safe TRC formula Q is a formula such that,

For any given I, the set of answers for Q contains only values that are in dom(Q, I).

For each sub expression of the form □R(p(R)) in Q, if a tuple r makes the formula true, then r contains

5. What is nested query?

When the domain relational calculus is restricted to safe expression, it is equivalent in expressive power to the tuple relational calculus restricted to safe expressions. All three of the following are equivalent,

The relational algebra

The tuple relational calculus restricted to safe expression

The domain relational calculus restricted to safe expression

THREE MARKS QUESTIONS WITH ANSWERS

1. How are queries expressed in SQL?

A query language is a language in which user requests to retrieve some information from the database. The query languages are considered as higher level languages than programming languages. Query languages are of two types,

Procedural Language

Non-Procedural Language

1. In procedural language, the user has to describe the specific procedure to retrieve the information from the database.

*Example:* The Relational Algebra is a procedural language.

2. In non-procedural language, the user retrieves the information from the database without describing the specific procedure to retrieve it.

*Example:* The Tuple Relational Calculus and the Domain Relational Calculus are non-procedural languages.

2. What are preliminaries

The relational algebra is a procedural query language. It consists of a set of operations that take one or two relations (tables) as input and produce a new relation, on the request of the user to retrieve the specific information, as the output.

3. Explain domain relational calculus.

Relational calculus is an alternative to relational algebra. In contrast to the algebra, which is procedural, the relational calculus is non-procedural or declarative.

It allows user to describe the set of answers without showing procedure about how they should be computed. Relational calculus has a big influence on the design of commercial query languages such as SQL and QBE (Query-by Example).

Relational calculus are of two types,

Tuple Relational Calculus (TRC)

Domain Relational Calculus (DRC)

Variables in TRC takes tuples (rows) as values and TRC had strong influence on SQL.

Variables in DRC takes fields (attributes) as values and DRC had strong influence on QBE.

i) Tuple Relational Calculus (TRC):

The tuple relational calculus, is a non-procedural query language because it gives the desired

information without showing procedure about how they should be computed.

A query in Tuple Relational Calculus (TRC) is expressed as { T | p(T) }

Where, T - tuple variable,

P(T) - 'p' is a condition or formula that is true for 't'.

In addition to that we use,

T[A] - to denote the value of tuple t on attribute A and

T □□r - to denote that tuple t is in relation r.

ii) Domain Relational Calculus (DRC):

A Duple Relational Calculus (DRC) is a variable that comes in the range of the values of domain (data types) of some columns (attributes).

A Domain Relational Calculus query has the form, { < x1, x2, …., xn > | p( < x1, x2, …., xn > ) }

Where, each xi is either a domain variable or a constant and p(< x1, x2, …., xn >) denotes a DRC formula.

A DRC formula is defined in a manner that is very similar to the definition of a TRC formula. The main difference is that the variables are domain variables.

4. What is aggregate operators

The tuple relational calculus restricts to safe expressions and is equal in expressive power to relational algebra. Thus, for every relational algebra expression, there is an equivalent expression in the tuple relational calculus and for tuple relational calculus expression, there is an equivalent relational algebra expression.

A safe TRC formula Q is a formula such that,

For any given I, the set of answers for Q contains only values that are in dom(Q, I).

For each sub expression of the form □R(p(R)) in Q, if a tuple r makes the formula true, then r contains

When the domain relational calculus is restricted to safe expression, it is equivalent in expressive power to the tuple relational calculus restricted to safe expressions. All three of the following are equivalent,

The relational algebra

The tuple relational calculus restricted to safe expression

The domain relational calculus restricted to safe expression

5. What is constraints?

A Duple Relational Calculus (DRC) is a variable that comes in the range of the values of domain (data types) of some columns (attributes).

A Domain Relational Calculus query has the form, { < x1, x2, …., xn > | p( < x1, x2, …., xn > ) }

Where, each xi is either a domain variable or a constant and p(< x1, x2, …., xn >) denotes a DRC formula.

A DRC formula is defined in a manner that is very similar to the definition of a TRC formula. The main difference is that the variables are domain variables.

FIVE MARKS QUESTIONS WITH ANSWERS

1. Make a comparison between the tuple relational calculus and domain relational calculus.

The tuple relational calculus restricts to safe expressions and is equal in expressive power to relational algebra. Thus, for every relational algebra expression, there is an equivalent expression in the tuple relational calculus and for tuple relational calculus expression, there is an equivalent relational algebra expression.

A safe TRC formula Q is a formula such that,

For any given I, the set of answers for Q contains only values that are in dom(Q, I).

For each sub expression of the form □R(p(R)) in Q, if a tuple r makes the formula true, then r contains

When the domain relational calculus is restricted to safe expression, it is equivalent in expressive power to the tuple relational calculus restricted to safe expressions. All three of the following are equivalent,

The relational algebra

The tuple relational calculus restricted to safe expression

The domain relational calculus restricted to safe expression

2. What are nested queries? What is correlation in nested queries? Explain.

A query language is a language in which user requests to retrieve some information from the database. The query languages are considered as higher level languages than programming languages. Query languages are of two types,

Procedural Language

Non-Procedural Language

1. In procedural language, the user has to describe the specific procedure to retrieve the information from the database.

*Example:* The Relational Algebra is a procedural language.

2. In non-procedural language, the user retrieves the information from the database without describing the specific procedure to retrieve it.

*Example:* The Tuple Relational Calculus and the Domain Relational Calculus are non-procedural languages.

3. Explain queries,constraints,triggers?

The relational algebra is a procedural query language. It consists of a set of operations that take one or two relations (tables) as input and produce a new relation, on the request of the user to retrieve the specific information, as the output.

4. What is triggers and active data bases?

Relational calculus is an alternative to relational algebra. In contrast to the algebra, which is procedural, the relational calculus is non-procedural or declarative.

It allows user to describe the set of answers without showing procedure about how they should be computed. Relational calculus has a big influence on the design of commercial query languages such as SQL and QBE (Query-by Example).

Relational calculus are of two types,

Tuple Relational Calculus (TRC)

Domain Relational Calculus (DRC)

Variables in TRC takes tuples (rows) as values and TRC had strong influence on SQL.

5. What is designing active databases?

The tuple relational calculus restricts to safe expressions and is equal in expressive power to relational algebra. Thus, for every relational algebra expression, there is an equivalent expression in the tuple relational calculus and for tuple relational calculus expression, there is an equivalent relational algebra expression.

**UNIT-3**

TWO MARKS QUESTIONS WITH ANSWERS

1. List the benefits of BCNF and 3NF.

The fourth step in database design is to analyze the collection of relations in our relational database schema to identify potential problems, and to refine it. In contrast to the requirements analysis and conceptual design steps, which are essentially subjective, schema refinement can be guided by some elegant and powerful theory.

Data redundancy means duplication of data. It causes duplicate data at different locations which destroys the integrity of the database and wastage of storage space.

The problems of redundancy are:

1.Wasted Storage Space. 2. More difficult Database Updates. 3. A Possibility of Inconsistent data.

2. Define loss less join decomposition with example.

The normalization theory based on the fundamental notion of FUNCTIONAL DEPENDENCY. Given a relation R, attribute A is functionally dependent on attribute B if each value of value of A in R is associated with precisely one value of B.

(OR)

In other words, attribute A is functionally dependent on B if and only if, for each value of B, there is exactly one value of A.

(OR)

Functional dependency is a relationship that exists when one attribute uniquely determines another attribute. If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as X->Y, which specifies Y is functionally dependent on X.

Here X is a determinant set and Y is a dependent attribute. Each value of X is associated precisely with one Y value.

3. What is locking Protocol?

The set of all FD's implied by a given set F of FD's is called the closure of F, denoted as F+. The following three rules, called Armstrong's axioms, can be applied repeatedly to compute all FDs implied by a set of FDs. Here, we use X, Y and Z to denote sets of attribute over a relation schema R,

Rule 1: Reflexivity: if X □Y, then X □□Y.

Rule 2: Augmentation: if X □□Y, then XZ □□YZ for any Z.

Rule 3: Transitivity: if X □□Y and Y □□Z, then X □□Z.

It is convenient to use some additional rules while reasoning about F+.

Rule 4: Union: If x □□Y and X □□z, then X □□YZ.

Rule 5: Decomposition: If X □□YZ, then X □□Y and X □□Z.

Rule 6: Pseudotranstivity: If X □□Y and Y □□P, then XZ □□P.

4. Explain normal form

Redundancy:

- Redundancy means repetition of data.
- Redundancy increases the time involved in updating, adding, and deleting data.
- It also increases the utilization of disk space and hence, disk I/O increases.

DEFINITION OF NORMALIZATION:

- Normalization is scientific method of breaking down complex stable structures into simple table structures by using certain rules.
- Using this method, you can, reduce redundancy in a table and elimates the problems of inconsistency and disk space usage.
- We can also ensure that there is no loss of information.

Benefits of Normalization:

- Normalization has several benefits.
- It enables faster sorting and index creation, more clustered indexes, few indexes per table, few NULLs, and makes the database compact.

5. What is normalization

- Normalization helps to simplify the structure of table. The performance of an application is directly linked to database design.
- A poor design hinders the performance of the system.
- The logical design of the database plays the foundation of an optical database.

The following are the some rules that should be followed to achieve a good data base design .They are:

Each table should have an identifier.

Each table should store data for single type entity.

Columns that accept NULLs should be avoided.

The repetition of values or columns should be avoided.

THREE MARKS QUESTIONS WITH ANSWERS

1. Write the Properties of Decompositions.

The normalization results in the formation of that satisfy certain specified rules and represent certain normal forms.

The normal forms are used to ensure that several of anomalies and inconsistencies are not introduced in the database.

A table structure is always in a certain normal form. Several normal forms have been identified.

The following are the most important and widely used normal forms are:

First normal form(1NF)

Second normal form(2NF)

Third normal form(3NF)

Boyce-Codd Normal Form (BCNF)

Fourth Normal Form (4NF)

The first, second and third normal forms are originally defined by Dr. E. F. Codd.

Later, Boyce and Codd introduced another form called the Boyce-Codd Normal form.

2. What is the difference between 3NF and BCNF?

The fourth step in database design is to analyze the collection of relations in our relational database schema to identify potential problems, and to refine it. In contrast to the requirements analysis and conceptual design steps, which are essentially subjective, schema refinement can be guided by some elegant and powerful theory.

Data redundancy means duplication of data. It causes duplicate data at different locations which destroys the integrity of the database and wastage of storage space.

The problems of redundancy are:

1.Wasted Storage Space. 2. More difficult Database Updates. 3. A Possibility of Inconsistent data.

3. When are two schedules conflict equivalent? What is conflict serializable schedule?

The normalization theory based on the fundamental notion of FUNCTIONAL DEPENDENCY. Given a relation R, attribute A is functionally dependent on attribute B if each value of value of A in R is associated with precisely one value of B.

(OR)

In other words, attribute A is functionally dependent on B if and only if, for each value of B, there is exactly one value of A.

(OR)

Functional dependency is a relationship that exists when one attribute uniquely determines another attribute.

If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as X->Y, which specifies Y is functionally dependent on X.

Here X is a determinant set and Y is a dependent attribute. Each value of X is associated precisely with one Y value.

4. Discuss the reason about FD'S

**Rules about Functional Dependencies**

The set of all FD's implied by a given set F of FD's is called the closure of F, denoted as F+. The following three rules, called Armstrong's axioms, can be applied repeatedly to compute all FDs implied by a set of FDs. Here, we use X, Y and Z to denote sets of attribute over a relation schema R,

Rule 1: Reflexivity: if X □Y, then X □□Y.

Rule 2: Augmentation: if X □□Y, then XZ □□YZ for any Z.

Rule 3: Transitivity: if X □□Y and Y □□Z, then X □□Z.

It is convenient to use some additional rules while reasoning about F+.

Rule 4: Union: If x □□Y and X □□z, then X □□YZ.

Rule 5: Decomposition: If X □□YZ, then X □□Y and X □□Z.

Rule 6: Pseudotranstivity: If X □□Y and Y □□P, then XZ □□P.

5. What is schema refinement in database design?

Redundancy:

- Redundancy means repetition of data.
- Redundancy increases the time involved in updating, adding, and deleting data.
- It also increases the utilization of disk space and hence, disk I/O increases.

DEFINITION OF NORMALIZATION:

- Normalization is scientific method of breaking down complex stable structures into simple table structures by using certain rules.
- Using this method, you can, reduce redundancy in a table and elimates the problems of inconsistency and disk space usage.
- We can also ensure that there is no loss of information.

Benefits of Normalization:

- Normalization has several benefits.
- It enables faster sorting and index creation, more clustered indexes, few indexes per table, few NULLs, and makes the database compact.

FIVE MARKS QUESTIONS WITH ANSWERS

1. Discuss about schema refinement

The fourth step in database design is to analyze the collection of relations in our relational database schema to identify potential problems, and to refine it. In contrast to the requirements analysis and conceptual design steps, which are essentially subjective, schema refinement can be guided by some elegant and powerful theory.

Data redundancy means duplication of data. It causes duplicate data at different locations which destroys the integrity of the database and wastage of storage space.

The problems of redundancy are:

1.Wasted Storage Space. 2. More difficult Database Updates. 3. A Possibility of Inconsistent data.

2. What is functional dependency

**6. FUNCTIONAL DEPENDENCY:**

The normalization theory based on the fundamental notion of FUNCTIONAL DEPENDENCY. Given a relation R, attribute A is functionally dependent on attribute B if each value of value of A in R is associated with precisely one value of B.

(OR)

In other words, attribute A is functionally dependent on B if and only if, for each value of B, there is exactly one value of A.

(OR)

Functional dependency is a relationship that exists when one attribute uniquely determines another attribute.

If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as X->Y, which specifies Y is functionally dependent on X.

Here X is a determinant set and Y is a dependent attribute. Each value of X is associated precisely with one Y value.

3. What is normal forms and properties of decompositions

The set of all FD's implied by a given set F of FD's is called the closure of F, denoted as F+. The following three rules, called Armstrong's axioms, can be applied repeatedly to compute all FDs implied by a set of FDs. Here, we use X, Y and Z to denote sets of attribute over a relation schema R,

Rule 1: Reflexivity: if X □Y, then X □□Y.

Rule 2: Augmentation: if X □□Y, then XZ □□YZ for any Z.

Rule 3: Transitivity: if X □□Y and Y □□Z, then X □□Z.

It is convenient to use some additional rules while reasoning about F+.

Rule 4: Union: If x □□Y and X □□z, then X □□YZ.

Rule 5: Decomposition: If X □□YZ, then X □□Y and X □□Z.

Rule 6: Pseudotranstivity: If X □□Y and Y □□P, then XZ □□P.

4. Discuss about normalization

Redundancy:

- Redundancy means repetition of data.
- Redundancy increases the time involved in updating, adding, and deleting data.
- It also increases the utilization of disk space and hence, disk I/O increases.

DEFINITION OF NORMALIZATION:

- Normalization is scientific method of breaking down complex stable structures into simple table structures by using certain rules.
- Using this method, you can, reduce redundancy in a table and elimates the problems of inconsistency and disk space usage.
- We can also ensure that there is no loss of information.

Benefits of Normalization:

- Normalization has several benefits.
- It enables faster sorting and index creation, more clustered indexes, few indexes per table, few NULLs, and makes the database compact.

5. What are the reasoning about FD'S

**6. NORMAL FORMS:**

The normalization results in the formation of that satisfy certain specified rules and represent certain normal forms.

The normal forms are used to ensure that several of anomalies and inconsistencies are not introduced in the database.

A table structure is always in a certain normal form. Several normal forms have been identified.

The following are the most important and widely used normal forms are:

First normal form(1NF)

Second normal form(2NF)

Third normal form(3NF)

Boyce-Codd Normal Form (BCNF)

Fourth Normal Form (4NF)

The first, second and third normal forms are originally defined by Dr. E. F. Codd.

Later, Boyce and Codd introduced another form called the Boyce-Codd Normal form.

**UNIT-4**

TWO MARKS QUESTIONS WITH ANSWERS

1. Why is recoverability of schedules desirable?

A transaction is a unit of program execution that accesses and possibly updates various data items.

(or)

A transaction is an execution of a user program and is seen by the DBMS as a series or list of actions i.e., the actions that can be executed by a transaction includes the reading and writing of database.

Transaction Operations:

Access to the database is accomplished in a transaction by the following two operations,

1) read(X) : Performs the reading operation of data item X from the database.

2) write(X) : Performs the writing operation of data item X to the database.

Example:

Let T1 be a transaction that transfers $50 from account A to account B. This transaction can be illustrated as follows,

T1 : read(A);

A := A – 50;

write(A);

read(B);

B := B + 50;

write(B);

2. What is locking Protocol?

**Transaction Concept:**

- The concept of transaction is the foundation for concurrent execution of transaction in a DBMS and recovery from system failure in a DBMS.

A user writes data access/updates programs in terms of the high-level query language supported by the DBMS

- To understand how the DBMS handles such requests, with respect to concurrency control and recovery, it is convenient to regard an execution of a user program or transaction, as a series of reads and writes of database objects.
- To read a database object, it is first brought in to main memory from disk and then its value is copied into a program. This is done by read operation.
- To write a database object, in-memory, copy of the object is first modified and then written to disk. This is done by the write operation.

3. What is storage structure?

When multiple transactions are being executed by the operating system in a multiprogramming environment, there are possibilities that instructions of one transaction are interleaved with some other transaction.

Schedule − A chronological execution sequence of a transaction is called a schedule. A schedule can have many transactions in it, each comprising of a number of instructions/tasks.

Serial Schedule − It is a schedule in which transactions are aligned in such a way that one transaction is executed first. When the first transaction completes its cycle, then the next transaction is executed.

Transactions are ordered one after the other. This type of schedule is called a serial schedule, as transactions are executed in a serial manner.

4. What is lock based protocol

In a multiprogramming environment where multiple transactions can be executed simultaneously, it is highly important to control the concurrency of transactions.

We have concurrency control protocols to ensure atomicity, isolation, and serializability of concurrent transactions.

Why DBMS needs a concurrency control?

In general, concurrency control is an essential part of TM. It is a mechanism for correctness when two or more database transactions that access the same data or data set are executed concurrently with time overlap. According to Wikipedia.org, if multiple transactions are executed serially or sequentially, data is consistent in a database. However, if concurrent transactions with interleaving operations are executed, some unexpected data and inconsistent result may occur. Data interference is usually caused by a write operation among transactions on

5. What is buffer management

This protocol guarantees that the execution of read and write operations that are conflicting is done in timestamp order.

Working of Timestamp Ordering Protocol:

The Time stamp ordering protocol ensures that any conflicting read and write operations are executed in time stamp order. This protocol operates as follows:

1) If TA executes read(x) instruction, then the following two cases must be considered,

i) TS(TA) < WTS(x)

ii) TS(TA) WTS(x)

Case 1 : If a transaction TA wants to read the initial value of some data item x that had been overwritten by some younger transaction then, the transaction TA cannot perform the read operation and therefore the transaction must be rejected. Then the transaction TA must be rolled back and restarted with a new timestamp.

Case 2 : If a transaction TA wants to read the initial value of some data item x that had not been updated then the transaction can execute the read operation. Once the value has read, changes occur in the read timestamp value (RTS(x)) which is set to the largest value of RTS(x) and TS

2) If TA executes write(x) instruction, then the following two cases must be considered,

i) TS(TA) < RTS(x)

ii) TS(TA) < WTS(x)

iii) TS(TA) >WTS(x)

THREE MARKS QUESTIONS WITH ANSWERS

1. Suppose that there is a database system that never fails. Is a recovery manager required for this system?

Case 1 : If a transaction TA wants to write the value of some data item x on which the read operation has been performed by some younger transaction, then the transaction cannot execute the write operation. This is because the value of data item x that is being generated by TA was required previously and therefore, the system assumes that the value will never be generated. The write operation is thereby rejected and the transaction TA must be rolled back and should be restarted with new timestamp value.

Case 2 : If a transaction TA wants to write a new value to some data item x, that was overwritten by some younger transaction, then the transaction cannot execute the write operation as it may lead to inconsistency of data item. Therefore, the write operation is rejected and the transaction should be rolled back with a new timestamp value.

Case 3 : If a transaction TA wants to write a new value on some data item x that was not updated by a younger transaction, then the transaction can executed the write operation. Once the value has been written, changes occur on WTS(x) value which is set to the value of TS(TA).

2. When are two schedules conflict equivalent? What is conflict serializable schedule?

This protocol guarantees that the execution of read and write operations that are conflicting is done in timestamp order.

Working of Timestamp Ordering Protocol:

The Time stamp ordering protocol ensures that any conflicting read and write operations are executed in time stamp order. This protocol operates as follows:

1) If TA executes read(x) instruction, then the following two cases must be considered,

i) TS(TA) < WTS(x)

ii) TS(TA) WTS(x)

Case 1 : If a transaction TA wants to read the initial value of some data item x that had been overwritten by some younger transaction then, the transaction TA cannot perform the read operation and therefore the transaction must be rejected. Then the transaction TA must be rolled back and restarted with a new timestamp.

Case 2 : If a transaction TA wants to read the initial value of some data item x that had not been updated then the transaction can execute the read operation. Once the value has read, changes occur in the read timestamp value (RTS(x)) which is set to the largest value of RTS(x) and TS

2) If TA executes write(x) instruction, then the following two cases must be considered,

i) TS(TA) < RTS(x)

ii) TS(TA) < WTS(x)

iii) TS(TA) >WTS(x)

3. What is transaction atomicity and durability

In a multiprogramming environment where multiple transactions can be executed simultaneously, it is highly important to control the concurrency of transactions.

We have concurrency control protocols to ensure atomicity, isolation, and serializability of concurrent transactions.

Why DBMS needs a concurrency control?

In general, concurrency control is an essential part of TM. It is a mechanism for correctness when two or more database transactions that access the same data or data set are executed concurrently with time overlap. According to Wikipedia.org, if multiple transactions are executed serially or sequentially, data is consistent in a database. However, if concurrent transactions with interleaving operations are executed, some unexpected data and inconsistent result may occur. Data interference is usually caused by a write operation among transactions on the same set of data in DBMS. For example, the lost update problem may occur when a second transaction writes a second value of data content on top of the first value written by a first concurrent transaction. Other problems such as the dirty read problem, the incorrect summary problem

Concurrency Control Techniques:

The following techniques are the various concurrency control techniques. They are:

concurrency control by Locks

Concurrency Control by Timestamps

Concurrency Control by Validation

4. What is serializability

the same set of data in DBMS. For example, the lost update problem may occur when a second transaction writes a second value of data content on top of the first value written by a first concurrent transaction. Other problems such as the dirty read problem, the incorrect summary problem

Concurrency Control Techniques:

The following techniques are the various concurrency control techniques. They are:

concurrency control by Locks

Concurrency Control by Timestamps

Concurrency Control by Validation

5. What is multi version schemes

A transaction is an execution of a user program and is seen by the DBMS as a series or list of actions i.e., the actions that can be executed by a transaction includes the reading and writing of database.

Transaction Operations:

Access to the database is accomplished in a transaction by the following two operations,

1) read(X) : Performs the reading operation of data item X from the database.

2) write(X) : Performs the writing operation of data item X to the database.

Example:

Let T1 be a transaction that transfers $50 from account A to account B. This transaction can be illustrated as follows,

T1 : read(A);

A := A – 50;

write(A);

read(B);

B := B + 50;

write(B);

FIVE MARKS QUESTIONS WITH ANSWERS

1. What is time stamp based protocol

**Transaction Concept:**

- The concept of transaction is the foundation for concurrent execution of transaction in a DBMS and recovery from system failure in a DBMS.
- A user writes data access/updates programs in terms of the high-level query language supported by the DBMS.
- To understand how the DBMS handles such requests, with respect to concurrency control and recovery, it is convenient to regard an execution of a user program or transaction, as a series of reads and writes of database objects.
- To read a database object, it is first brought in to main memory from disk and then its value is copied into a program. This is done by read operation.
- To write a database object, in-memory, copy of the object is first modified and then written to disk. This is done by the write operation.

2. Describe validation based protocol

Schedule − A chronological execution sequence of a transaction is called a schedule. A schedule can have many transactions in it, When multiple transactions are being executed by the operating system in a multiprogramming environment, there are possibilities that instructions of one transaction are interleaved with some other transaction.

each comprising of a number of instructions/tasks.

Serial Schedule − It is a schedule in which transactions are aligned in such a way that one transaction is executed first. When the first transaction completes its cycle, then the next transaction is executed. Transactions are ordered one after the other. This type of schedule is called a serial schedule, as transactions are executed in a serial manner.

3. Discuss recovery and atomicity

In a multiprogramming environment where multiple transactions can be executed simultaneously, it is highly important to control the concurrency of transactions.

We have concurrency control protocols to ensure atomicity, isolation, and serializability of concurrent transactions.

Why DBMS needs a concurrency control?

In general, concurrency control is an essential part of TM. It is a mechanism for correctness when two or more database transactions that access the same data or data set are executed concurrently with time overlap. According to Wikipedia.org, if multiple transactions are executed serially or sequentially, data is consistent in a database. However, if concurrent transactions with interleaving operations are executed, some unexpected data and inconsistent result may occur. Data interference is usually caused by a write operation among transactions on the same set of data in DBMS. For example, the lost update problem may occur when a second transaction writes a second value of data content on top of the first value written by a first concurrent transaction. Other problems such as the dirty read problem, the incorrect summary problem

Concurrency Control Techniques:

The following techniques are the various concurrency control techniques. They are:

concurrency control by Locks

Concurrency Control by Timestamps

Concurrency Control by Validation

4. What is recovery algorithm

This protocol guarantees that the execution of read and write operations that are conflicting is done in timestamp order.

Working of Timestamp Ordering Protocol:

The Time stamp ordering protocol ensures that any conflicting read and write operations are executed in time stamp order. This protocol operates as follows:

1) If TA executes read(x) instruction, then the following two cases must be considered,

i) TS(TA) < WTS(x)

ii) TS(TA) WTS(x)

Case 1 : If a transaction TA wants to read the initial value of some data item x that had been overwritten by some younger transaction then, the transaction TA cannot perform the read operation and therefore the transaction must be rejected. Then the transaction TA must be rolled back and restarted with a new timestamp.

Case 2 : If a transaction TA wants to read the initial value of some data item x that had not been updated then the transaction can execute the read operation. Once the value has read, changes occur in the read timestamp value (RTS(x)) which is set to the largest value of RTS(x) and TS

2) If TA executes write(x) instruction, then the following two cases must be considered,

i) TS(TA) < RTS(x)

ii) TS(TA) < WTS(x)

iii) TS(TA) >WTS(x)

5. Discuss remote backup system

Case 1 : If a transaction TA wants to write the value of some data item x on which the read operation has been performed by some younger transaction, then the transaction cannot execute the write operation. This is because the value of data item x that is being generated by TA was required previously and therefore, the system assumes that the value will never be generated. The write operation is thereby rejected and the transaction TA must be rolled back and should be restarted with new timestamp value.

Case 2 : If a transaction TA wants to write a new value to some data item x, that was overwritten by some younger transaction, then the transaction cannot execute the write operation as it may lead to inconsistency of data item. Therefore, the write operation is rejected and the transaction should be rolled back with a new timestamp value.

Case 3 : If a transaction TA wants to write a new value on some data item x that was not updated by a younger transaction, then the transaction can executed the write operation. Once the value has been written, changes occur on WTS(x) value which is set to the value of TS(TA).

UNIT-5

TWO MARKS QUESTIONS WITH ANSWERS

1. How is data organized in a hash based index?

As we know already, database consists of tables, views, index, procedures, functions etc. The tables and views are logical form of viewing the data. But the actual data are stored in the physical memory. Database is a very huge storage mechanism and it will have lots of data and hence it will be in physical storage devices. In the physical memory devices, these datas cannot be stored as it is. They are converted to binary

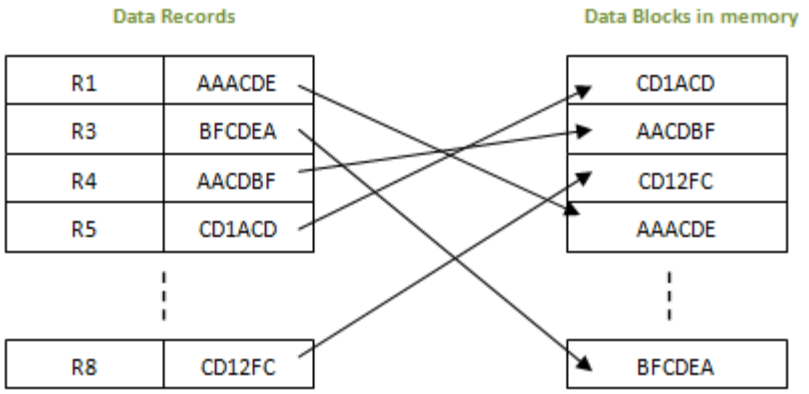
format. Each memory devices will have many data blocks, each of which will be capable of storing certain amount of data. The data and these blocks will be mapped to store the data in the memory.

Any user who wants to view these data or modify these data, simply fires SQL query and gets the result on the screen. But any of these queries should give results as fast as possible. But how these data are fetched from the physical memory? Do you think simply storing the data in memory devices give us the better results when we fire queries? Certainly not. How is it stored in the memory, Accessing method, query type etc makes great affect on getting the results. Hence organizing the data in the database and hence in the memory is one of important think about.

2. Why are tree-structure indexes are good for searches, especially range selections.

This is an advanced sequential file organization method. Here records are stored in order of primary key in the file. Using the primary key, the records are sorted. For each primary key, an index value is generated and mapped with the record. This index is nothing but the address of record in the file.

In this

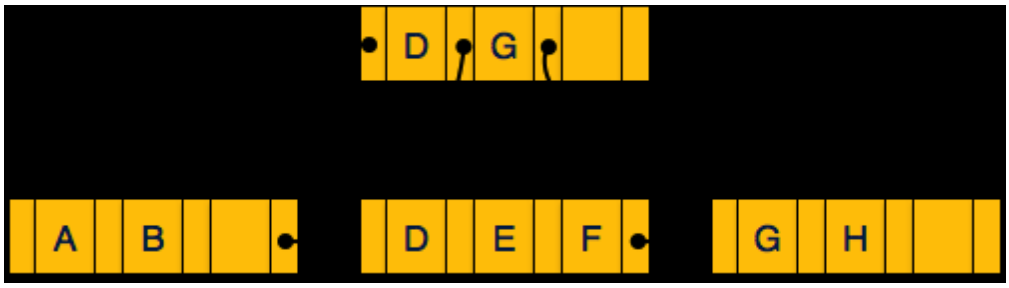



3. Discuss the overview of storage and indexing

A B+ tree is a balanced binary search tree that follows a multi-level index format. The leaf nodes of a B+ tree denote actual data pointers. B+ tree ensures that all leaf nodes remain at the same height, thus balanced. Additionally, the leaf nodes are linked using a link list; therefore, a B+ tree can support random access as well as sequential access.

Structure of B+ Tree

Every leaf node is at equal distance from the root node. A B+ tree is of the ordern where n is fixed for every B+ tree.




4. What is index data structures

 Bucket − A hash file stores data in bucket format. Bucket is considered a unit of storage. A bucket typically stores one complete disk block, which in turn can store one or more records.

 Hash Function − A hash function, h, is a mapping function that maps all the set of search-keys K to the address where actual records are placed. It is a function from search keys to bucket addresses.

Dept. of CSE, CREC Page 111

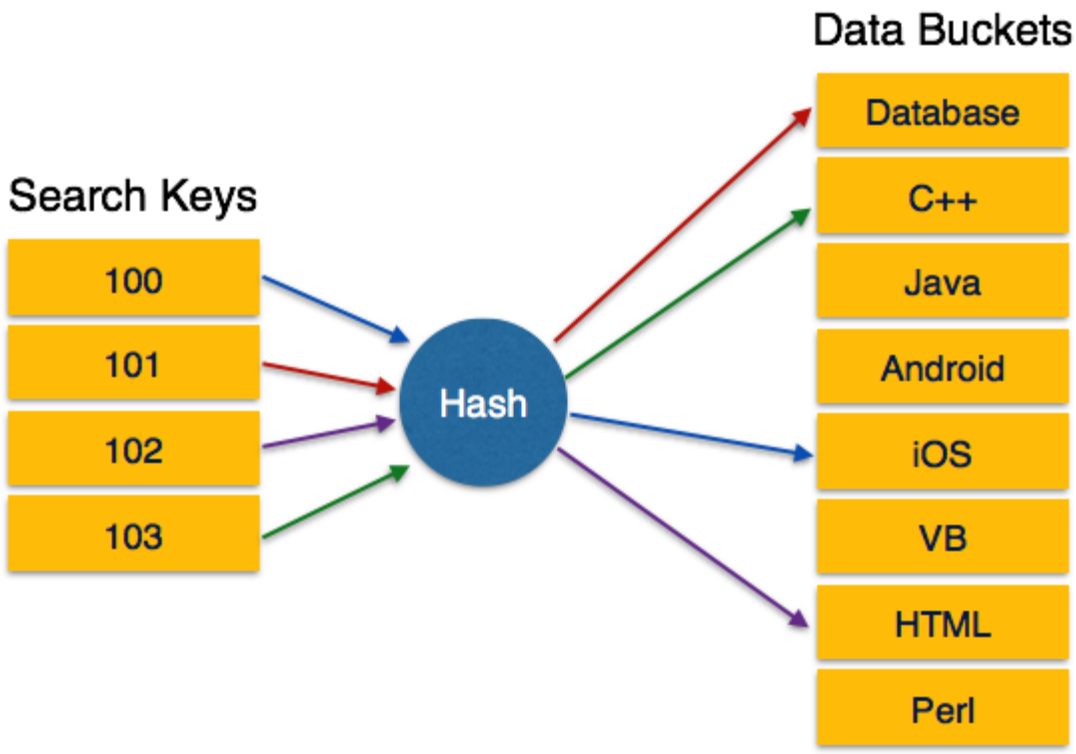
HASHING:

 Bucket − A hash file stores data in bucket format. Bucket is considered a unit of storage. A bucket typically stores one complete disk block, which in turn can store one or more records.

 Hash Function − A hash function, h, is a mapping function that maps all the set of search-keys K to the address where actual records are placed. It is a function from search keys to bucket addresses.

5. Discuss the comparision of file organization

In static hashing, when a search-key value is provided, the hash function always computes the same address. For example, if mod-4 hash function is used, then it shall generate only 5 values. The output address shall always be same for that function. The number of buckets provided remains unchanged at all times.




THREE MARKS QUESTIONS WITH ANSWERS

1. What is the main difference between ISAM and B+ tree indexes?

Hash table is a data structure that associates keys with values. To know more about liner hashing refers Wikipedia. Here are main points that summarize linear hashing.

Full buckets are not necessarily split

Buckets split are not necessarily full

Every bucket will be split sooner or later and so all Overflows will be reclaimed and rehashed.

Split pointer s decides which bucket to split s is independent to overflowing bucket

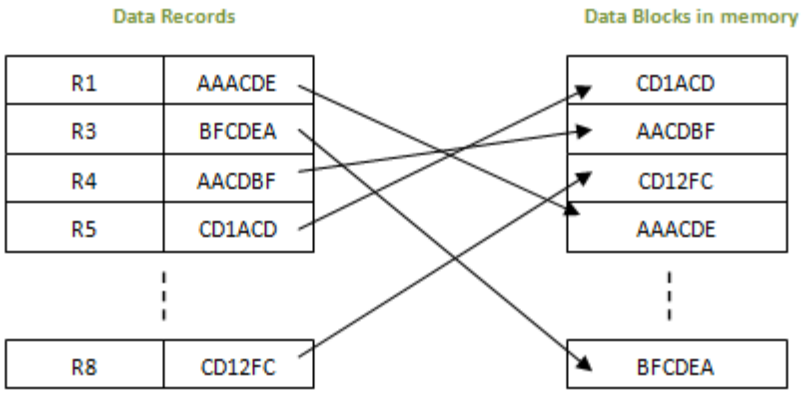
As we know already, database consists of tables, views, index, procedures, functions etc. The tables and views are logical form of viewing the data. But the actual data are stored in the physical memory. Database is a very huge storage mechanism and it will have lots of data and hence it will be in physical storage devices. In the physical memory devices, these datas cannot be stored as it is. They are converted to binary format. Each memory devices will have many data blocks, each of which will be capable of storing certain amount of data. The data and these blocks will be mapped to store the data in the memory.

Any user who wants to view these data or modify these data, simply fires SQL query and gets the result on the screen. But any of these queries should give results as fast as possible. But how these data are fetched from the physical memory? Do you think simply storing the data in memory devices give us the better results when we fire queries? Certainly not. How is it stored in the memory, Accessing method, query type etc makes great affect on getting the results. Hence organizing the data in the database and hence in the memory is one of important think about.

2. Explainindex sequentialaccess method

This is an advanced sequential file organization method. Here records are stored in order of primary key in the file. Using the primary key, the records are sorted. For each primary key, an index value is generated and mapped with the record. This index is nothing but the address of record in the file.

In this

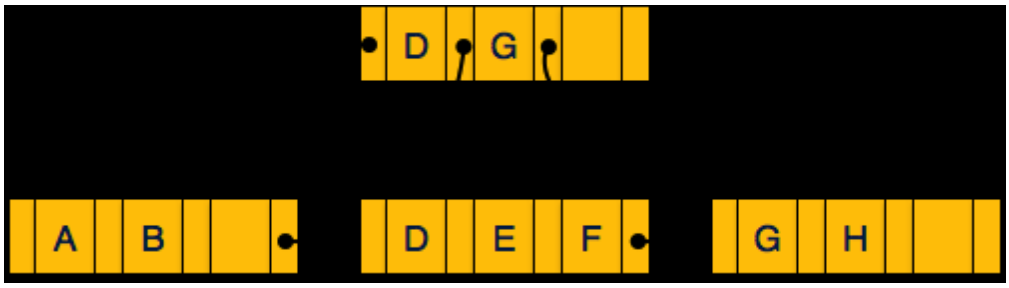



3. Explain a dynamic index structure

A B+ tree is a balanced binary search tree that follows a multi-level index format. The leaf nodes of a B+ tree denote actual data pointers. B+ tree ensures that all leaf nodes remain at the same height, thus balanced. Additionally, the leaf nodes are linked using a link list; therefore, a B+ tree can support random access as well as sequential access.

Structure of B+ Tree

Every leaf node is at equal distance from the root node. A B+ tree is of the ordern where n is fixed for every B+ tree.




4. Explain static hashing

 Bucket − A hash file stores data in bucket format. Bucket is considered a unit of storage. A bucket typically stores one complete disk block, which in turn can store one or more records.

 Hash Function − A hash function, h, is a mapping function that maps all the set of search-keys K to the address where actual records are placed. It is a function from search keys to bucket addresses.

Dept. of CSE, CREC Page 111

HASHING:

 Bucket − A hash file stores data in bucket format. Bucket is considered a unit of storage. A bucket typically stores one complete disk block, which in turn can store one or more records.

 Hash Function − A hash function, h, is a mapping function that maps all the set of search-keys K to the address where actual records are placed. It is a function from search keys to bucket addresses.

5. Explin dynamic hashing

In static hashing, when a search-key value is provided, the hash function always computes the same address. For example, if mod-4 hash function is used, then it shall generate only 5 values. The output address shall always be same for that function. The number of buckets provided remains unchanged at all times.

FIVE MARKS QUESTIONS WITH ANSWERS

1. Give a brief note on Static Hashing.

Hash table is a data structure that associates keys with values. To know more about liner hashing refers Wikipedia. Here are main points that summarize linear hashing.

Full buckets are not necessarily split

Buckets split are not necessarily full

Every bucket will be split sooner or later and so all Overflows will be reclaimed and rehashed.

Split pointer s decides which bucket to split s is independent to overflowing bucket

2. Discuss extendible hashing

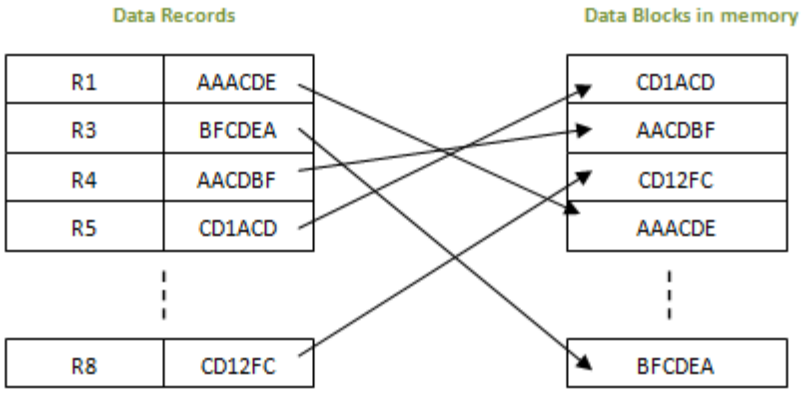
As we know already, database consists of tables, views, index, procedures, functions etc. The tables and views are logical form of viewing the data. But the actual data are stored in the physical memory. Database is a very huge storage mechanism and it will have lots of data and hence it will be in physical storage devices. In the physical memory devices, these datas cannot be stored as it is. They are converted to binary format. Each memory devices will have many data blocks, each of which will be capable of storing certain amount of data. The data and these blocks will be mapped to store the data in the memory.

Any user who wants to view these data or modify these data, simply fires SQL query and gets the result on the screen. But any of these queries should give results as fast as possible. But how these data are fetched from the physical memory? Do you think simply storing the data in memory devices give us the better results when we fire queries? Certainly not. How is it stored in the memory, Accessing method, query type etc makes great affect on getting the results. Hence organizing the data in the database and hence in the memory is one of important think about.

3. Explain linear hashing

This is an advanced sequential file organization method. Here records are stored in order of primary key in the file. Using the primary key, the records are sorted. For each primary key, an index value is generated and mapped with the record. This index is nothing but the address of record in the file.

In this

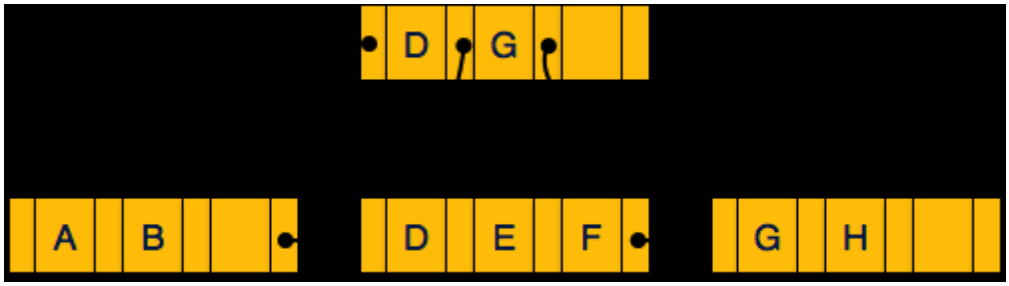



4. Discuss tree structured indexing

A B+ tree is a balanced binary search tree that follows a multi-level index format. The leaf nodes of a B+ tree denote actual data pointers. B+ tree ensures that all leaf nodes remain at the same height, thus balanced. Additionally, the leaf nodes are linked using a link list; therefore, a B+ tree can support random access as well as sequential access.

Structure of B+ Tree

Every leaf node is at equal distance from the root node. A B+ tree is of the ordern where n is fixed for every B+ tree.




5. What is data on external storage

 Bucket − A hash file stores data in bucket format. Bucket is considered a unit of storage. A bucket typically stores one complete disk block, which in turn can store one or more records.

 Hash Function − A hash function, h, is a mapping function that maps all the set of search-keys K to the address where actual records are placed. It is a function from search keys to bucket addresses.

HASHING:

 Bucket − A hash file stores data in bucket format. Bucket is considered a unit of storage. A bucket typically stores one complete disk block, which in turn can store one or more records.

 Hash Function − A hash function, h, is a mapping function that maps all the set of search-keys K to the address where actual records are placed. It is a function from search keys to bucket addresses.

## DBMS Unit wise Quiz Questions

### UNIT-1

Q.6 In the relational modes, cardinality is termed as:

(A) Number of tuples.

(B) Number of attributes.

(C) Number of tables.

(D) Number of constraints.

Ans: A

Q.7 Relational calculus is a

(E) Procedural language.

(F) Non- Procedural language.

(G) Data definition language.

**(H)** High level language.

Ans: **B**

Q.8 The view of total database content is

(E) Conceptual view.

(F) Internal view.

(G) External view.

(H) Physical View.

Ans: **A**

Q.9 Cartesian product in relational algebra is

(E) A Unary operator.

(F) A Binary operator.

(G) A Ternary operator.

(H) Not defined.

Ans: **B** Cartesian product in relational algebra is a binary

Operator. (It requires two operands. e.g., P X Q)

Q.10DML is provided for

(E) Description of logical structure of database.

(F) Addition of new structures in the database system.

(G) Manipulation & processing of database.

(H) Definition of physical structure of database system.

Ans: **C** DML is provided for manipulation & processing of database.

(Data stored in the database is processed or manipulated using data manipulation language commands as its name)

Q. 1‘AS’ clause is used in SQL for

(A) Selection operation.

(B) Rename operation.

(C) Join operation.

(D) Projection operation.

Ans: B ‘AS’ clause is used in SQL for rename operation. (e.g., SELECT ENO AS EMPLOYEE_NO FROM EMP)

Q. 2ODBC stands for

(A) Object Database Connectivity.

(B) Oral Database Connectivity.

(C) Oracle Database Connectivity.

(D) Open Database Connectivity.

Ans: D

Q. 3Architecture of the database can be viewed as

(A) Two levels.

(B) Four levels.

(C) Three levels.

(D) One level.

Ans: C

Q. 4In a relational model, relations are termed as

(A) Tuples.

(B) Attributes

(C) Tables.

(D) Rows.

Ans:

Q. 5The database schema is written in

(A) HLL

(B) DML

(C) DDL

(D) DCL

Ans: C

Q. 6In the architecture of a database system external level is the

(A) Physical level.

(B) Logical level.

(C) conceptual level

(D) View level.

Ans: D

**Unit-II**

Q.11An entity set that does not have sufficient attributes to form a primary key is a

(E) Strong entity set.

(F) Weak entity set.

(G) Simple entity set.

(H) Primary entity set.

Ans: B

Q.12In a Hierarchical model records are organized as

(E) Graph.

(F) List.

(G) Links.

(H) Tree.

Ans: D

Q.13In an E-R diagram attributes are represented by

(E) Rectangle.

(F) Square.

(G) Ellipse.

(H) Triangle.

Ans: C

Q.14In case of entity integrity, the primary key may be

(E) Not Null

(F) Null

(G) Both Null & not Null.

(H) Any value.

Ans: A

Q.15In tuple relational calculus 2 1 P P →

is equivalent to

(E) 2 1 P P ∨ ¬

(F) 2 1 P P∨

(G) 2 1 P P∧

(H) 2 1 P P ¬ ∧

Ans: A In tuple relational calculus P1 P2 is equivalent to ¬P1 ∨ P2.

(The logical implication expression A B, meaning if A then B,is equivalent to ¬A ∨ B)

Q.16 The language used in application programs to request data from the DBMS is referred to as the

(A) DML

(B) DDL

(C) VDL

(D) SDL

Ans: A

Q.17 A logical schema

(A) Is the entire database.

(B) Is a standard way of organizing information into accessible parts.

(C) Describes how data is actually stored on disk.

(D) both (A) and (C)

Ans: A

Q. 8Related fields in a database are grouped to form a

(A) Data file.

(B) Data record.

(C) Menu.

(D) Bank.

Ans: B Related data fields in a database are grouped to form a data record. (A record is a collection of related fields)

Q. 9The database environment has all of the following components except:

(A) users.

(B) separate files.

(C) database.

(D) database administrator.

Ans: A

Q.20 The language which has recently become the defacto standard for interfacing Application programs with relational database system is

(A) Oracle.

(B) SQL.

(C) DBase.

(D) 4GL.

Ans: B

Q.21 The way a particular application views the data from the database that the application uses is a

(A) Module.

(B) Relational model.

(C) Schema.

(D) Sub schema.

Ans: D

Q. 12 In an E-R diagram an entity set is represent by a

(A) Rectangle.

(B) Ellipse.

(C) Diamond box.

(D) Circle.

Ans: A

**Unit-III**

Q.2 A report generator is used to

(E) Update files.

(F) Print files on paper.

(G) Data entry.

(H) Delete files.

Ans: B

Q.3 The property / properties of a database is / are :

(E) It is an integrated collection of logically related records.

(F) It consolidates separate files into a common pool of data records.

(G) Data stored in a database is independent of the application programs using it.

(H) All of the above.

Ans: D

Q.4 The DBMS language component which can be embedded in a program is

(E) The data definition language (DDL).

(F) The data manipulation language (DML).

(G) The database administrator (DBA).

(H) A query language.

Ans: B

Q.6 A relational database developer refers to a record as

(E) A Criteria.

(F) A relation.

(G) A tuple.

(H) An attribute.

Ans: C

Q.7 The relational model feature is that there

(E) Is no need for primary key data

(F) Is much more data independence than some other database models.

(G) Are explicit relationships among records

(H) Are tables with many dimensions

Ans: B

Q.8 Conceptual design

(A) Is a documentation technique.

(B) Needs data volume and processing frequencies to determine the size of the database.

(C) Involves modelling independent of the DBMS.

(D) Is designing the relational model.

Ans:C

Q.9 The method in which records are physically stored in a specified order according to a key field in each recordis

(A) Hash.

(B) Direct.

(C) Sequential.

(D) All of the above.

Ans: A

Q. 0A subschema expresses

(A) The logical view.

(B) The physical view.

(C) The external view

(D) All of the above.

Ans: C A subschema expresses the external view. (External schemas are called also called as subschemas)

Q. 1Count function in SQL returns the number of

(A) values.

(B) distinct values.

(C) groups.

(D) columns.

Ans: A Count function in SQL returns the number of values. (Count function counts all the not null values in the specific column. If we want to count only distinct values than the

DISTINCT keyword is also to be used)

Q.12 Which one of the following statements is false?

(A) The data dictionary is normally maintained by the database administrator.

(B) Data elements in the database can be modified by changing the data dictionary.

(C) The data dictionary contains the name and description of each data element.

(D) The data dictionary is a tool used exclusively by the database administrator.

Ans: B

**Unit-IV**

Q.11An advantage of the database management approach is

(E) Data is dependent on programs.

(F) Data redundancy increases.

(G) Data is integrated and can be accessed by multiple programs.

(H) None of the above.

Ans: C

Q.12A DBMS query language is designed to

(E) Support end users who use English-like commands.

(F) Support in the development of complex applications software.

(G) Specify the structure of a database.

(H) All of the above.

Ans: D

Q.13Transaction processing is associated with everything below except

(E) Producing detail, summary, or exception reports.

(F) Recording a business activity.

(G) Confirming an action or triggering a response.

(H) Maintaining data.

Ans: C

Q.14It is possible to define a schema completely using

(E) VDL and DDL.

(F) DDL and DML.

(G) SDL and DDL.

(H) VDL and DML.

Ans: B

Q.15The method of access which uses key transformation is known as

(E) Direct.

(F) Hash.

(G) Random.

(H) Sequential.

Ans: B

Q. 6Data independence means

(A) Data is defined separately and not included in programs.

(B) Programs are not dependent on the physical attributes of data.

(C) Programs are not dependent on the logical attributes of data.

(D) Both (B) and (C).

Ans: D both (B) and (C)

Q. 7The statement in SQL which allows to change the definition of a table is

(A) Alter.

(B) Update.

(C) Create.

(D) Select.

Ans: A

Q. 8E-R model uses this symbol to represent weak entity set ?

(A) Dotted rectangle.

(B) Diamond

(C) Doubly outlined rectangle

(D) None of these

Ans: C

Q. 9SET concept is used in :

(A) Network Model

(B) Hierarchical Model

(C) Relational Model

(D) None of these

Ans: A

Q.20 Relational Algebra is

(A) Data Definition Language.

(B) Meta Language

(C) Procedural query Language

(D) None of the above

Ans: C

Q.21 Key to represent relationship between tables is called

(A) Primary key

(B) Secondary Key

(C) Foreign Key

(D) None of these

Ans: C

**Unit-V**

Q.1 produces the relation that has attributes of R1 and R2

(A) Cartesian product

(B) Difference

(C) Intersection

(D) Product

Ans: A

Q.2 The file organization that provides very fast access to any arbitrary record of a file is

(A) Ordered file

(B) Unordered file

(C) Hashed file

(D) B-tree

Ans: C

Q.3 DBMS helps achieve

(A) Data independence

(B) Centralized control of data

(C) Neither (A) nor (B)

(D) both (A) and (B)

Ans: D

Q.4 Which of the following are the properties of entities?

(A) Groups

(B) Table

(C) Attributes

(D) Switchboard

Ans: C

Q.5 In a relation

(A) Ordering of rows is immaterial

(B) No two rows are identical

(C) (A)and (B) both are true

(D) None of these.

Ans: C

Q.6 Which of the following is correct:

(A) A SQL query automatically eliminates duplicates.

(B) SQL permits attribute names to be repeated in the same relation.

(C) A SQL query will not work if there are no indexes on the relations

(D) None of these

(E) Ans: D

Q.7 It is better to use files than a DBMS when there are

(A) Stringent real-time requirements.

(B) Multiple users wish to access the data.

(C) Complex relationships among data.

(D) All of the above

Ans: B

Q.8 The conceptual model is

(A) Dependent on hardware.

(B) Dependent on software.

(C) Dependent on both hardware and software .

(D) Independent of both hardware and software.

Ans: D

Q.9 What is a relationship called when it is maintained between two entities?

(A) Unary

(B) Binary

(C) Ternary

(D) Quaternary

Ans:B

Q.10 Which of the following operation is used if we are interested in only certain columns of a table?

(A) PROJECTION

(B) SELECTION

(C) UNION

(D) JOIN

Ans: A

## BEYOND SYLLABUS

1. Which popular DBMS products are based on the relational data model? DBMS products are based on a logical model other than the relational data model? The relative strengths and weaknesses of the different types (relational versus other logical models) of DBMSs

A. No single answer exists with this case; indeed, solutions will vary depending upon student ingenuity and creativity. Reports should be graded in terms of how well each issue was addressed and in terms of writing quality. Students should be able to find the following information:

Relational DBMSs include DB2, Oracle, SQL Server and Access.

Many newer products are based on the object-oriented data model, or are a hybrid of the relational and Object-oriented approaches. Older mainframe DBMS are based on hierarchical or network logical models. Hierarchical and network DBMSs often provide performance advantages--especially in terms of processing

speed. Those advantages, however, usually come at the cost of making it much more difficult for end users to do ad-hoc queries of the database. Relational databases support easy to use, yet powerful query languages like SQL and graphical query-by-example languages such as that provided by Microsoft Access. Object-oriented databases are especially effective for including multimedia, whereas hierarchical, network, and relational databases are better suited for alphanumeric data (although the relational model can be extended to include multimedia data). Pure object-oriented databases are more often designed for special purpose scientific use when graphical images and sound need to be stored in the database. Relational and hybrid object-relational DBMSs are commonly used in newer transaction processing systems, although older systems are based on the hierarchical or network data models.

2. Contrast the logical and the physical view of data and discuss why separate views are necessary in database applications. Describe which perspective is most useful for each of the following employees: a programmer, a manager, and an internal auditor. How will understanding logical data structures assist you when designing and using database systems?

A. Databases are possible because of their database management system (DBMS). As shown in Figure 4.2, the DBMS is a software program that sits between the actual data stored in the system and the application programs that use the data. As shown in Figure 4.4, this allows users to separate the way they view the data (called the logical view) from the way the data is actually stored (the physical view). The DBMS interprets the users' requests and retrieves, manipulates, or stores the data as needed. The two distinct views separate the applications from the physical information, providing increased flexibility in applications, improved data security, and ease of use. In a database system, the manager will rarely need to understand or be familiar with the physical view of the data. Nor, in most instances, will the internal auditor and the programmer as most everything they do involves the logical view of the data.

If accountants understand logical data structures and the logical view of the data, they are better able to manage, use, and audit a database and its data.

3. The relational data model represents data as being stored in tables. Spreadsheets are another tool that accountants use to employ a tabular representation of data. What are some similarities and differences in the

way these tools use tables? How might an accountant's familiarity with the tabular representation of spreadsheets facilitate or hinder learning how to use a relational DBMS?

A. Major difference between spreadsheets and databases is that spreadsheets are designed primarily to handle numeric data, whereas databases can handle both text and numbers. Consequently, the query and sorting capabilities of spreadsheets are much more limited than what can be accomplished with a DBMS that has a good query language. Accountants' familiarity with spreadsheets might hinder their ability to design and use relational DBMS because many links in spreadsheets are preprogrammed and designed in, whereas a well-designed relational database is designed to facilitate ad-hoc queries. Accountants' familiarity with spreadsheets sometimes leads them to use a spreadsheet for a task that a database could handle much better. Over the years, the Journal of Accountancy has published a number of very good articles on how to use databases and when to use databases and when to use spreadsheets. These articles can be found on the Journal's website: http://www.journalofaccountancy.com/

4 Some people believe database technology may eliminate the need for double-entry accounting. This creates three possibilities: (1) the double-entry model will be abandoned; (2) the double-entry model will not be used directly, but an external-level schema based on the double-entry model will be defined for accountants' use; or (3) the double-entry model will be retained in database systems. Which alternative do you think is most likely to occur?

A. There is no correct answer to this question because it is asking the student to express his opinion on what will happen in the future. Therefore, the quality of his answer depends on the justifications provided. Good answers should address the following:

Database technology does permit abandonment of double entry, but there will likely be great resistance to such a radical change. Thus, students choosing this option need to present reasons why they think such a radical change would succeed.

• The use of a schema for accountants seems quite plausible. It does eliminate the redundancy of double entry from the database system, yet it still provides a framework familiar and useful to accountants and financial analysts.

• There is a good possibility that double entry will remain, even in databases, due to inertia. Indeed, many modern AIS, such as ERP systems, use databases but also retain the principles of double entry.

5. Relational DBMS query languages provide easy access to information about the organization's activities. Does this mean that online, real-time processing should be used for all transactions? Does an organization need real-time financial reports? Why or why not?

A. On-line real-time processing is not necessary for every business transaction. For example, batch processing is adequate for payroll: there is little need for the data to be current except on payday. Real-time financial statements are useful for planning and provide management with better ability to react to changes in the environment. Nevertheless, real-time financial statements may present distorted pictures of reality if accruals have been ignored or not properly recognized.

6. Why is it so important to have good data?

A. Bad data costs businesses over $600 billion a year. Some people estimate that over 25% of business data is inaccurate or incomplete. In addition, incorrect database data can lead to bad decisions, embarrassment, and angry users. The text illustrated this with the following examples:

For quite some time, a company sent half its catalogs to incorrect addresses. A manager finally investigated the large volume of returns and customer complaints and corrected the customer addresses in the database. He saved the company $12 million a year.

Valparaiso, Indiana used the county database to develop its tax rates. After mailing the tax notices, it was discovered that a $121,900 home was valued at $400 million. Due to the $3.1 million property tax revenue shortfall, the city, the school district, and governmental agencies had to make severe budget cuts.

Managing data is not going to get any easier as the quantity of data generated and stored doubles every 18 months.

7. What is a data dictionary, what does it contain, and how is it used?

A data dictionary contains information about the structure of the database. Table 4-1 shows that there is a record in the dictionary describing each data element. The DBMS maintains the data dictionary, whose inputs include new or deleted data elements and changes in data element names, descriptions, or uses. Outputs include reports for programmers, designers, and users. These reports are used for system documentation, database design and implementation, and as part of the audit trail.

8. Compare and contrast the file-oriented approach and the database approach. Explain the main advantages of database systems.

Information about the attributes of a customer, such as name and address, are stored in fields. Fields contain data about one entity (e.g., one customer). Multiple fields form a record. A set of related records, such as all customer records, forms a file (e.g., the customer file). A set of interrelated, centrally coordinated files forms a database.

Figure 4-2 illustrates the differences between file-oriented and database systems. In the database approach, data is an organizational resource that is used by and managed for the entire organization, not just the originating department. A database management system (DBMS) is the interface between the database and the various application programs. The database, the DBMS, and the application programs that access the database through the DBMS are referred to as the database system. Database systems were developed to address the proliferation of master files. This proliferation created problems such as the same data stored in two or more master files. This made it difficult to integrate and update data and to obtain an organization-wide view of data. It also created problems because the data in the different files was inconsistent.

Databases provide organizations with the following benefits:

Data integration. Master files are combined into large “pools” of data that many application programs access. An example is an employee database that consolidates payroll, personnel, and job skills master files.

Data sharing. Integrated data is more easily shared with authorized users. Databases are easily browsed to research a problem or obtain detailed information underlying a report. The FBI, which does a good job of collecting data but a poor job of sharing it, is spending eight years and $400 million to integrate data from their different systems.

Minimal data redundancy and data inconsistencies. Because data items are usually stored only once, data redundancy and data inconsistencies are minimized.

Data independence. Because data and the programs that use them are independent of each other, each can be changed without changing the other. This facilitates programming and simplifies data management.

Cross-functional analysis. In a database system, relationships, such as the association between selling costs and promotional campaigns, can be explicitly defined and used in the preparation of management reports.