SASURIE INSTITUTIONS

SASURIE
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

Department : CSE
Subject Code & Name : CS3492 & Database Management System
Class & Batch : II
Semester : IV

CONTENTS – COURSE FILE

S.NO	PARTICULARS	REMARKS
1	Time Table	✓
2	Student name list	✓
3	Student arrear list	–
4	Subject Information Record	✓
5	Syllabus	✓
6	Lesson Plan	✓
7	Test Plan for the Subject	✓
8	Result Analysis	✓
9	Corrective Action Report	✓
10	Quality objective monitoring record	✓
11	Internal test mark sheet(Consolidated)	✓
12	Internal test question paper with answer key	✓
13	Model question paper with answer key	✓
14	Slip test question paper with answer key	–
15	Sample Answer paper for all test(Min-3)	✓
16	Content beyond the syllabus	✓
17	Tutorial Class – schedule and content	–
18	Assignment – schedule and paper	✓
19	PPT - handout	✓
20	Question bank	✓
21	Sample university question papers(min 5 QP-recent exam)	✓
22	Personal Log book – Updated	–
23	Lecture Note	–
24	Special Class if any, Approval letter, Schedule, content covered.	

	Prepared By	Approved By
Sign:		
Name:		
	Faculty	HoD

SASURIE INSTITUTIONS

SASURIE
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

Department : CSE | **w.e.f : 08.02.2024**
Class : II YEAR | **Semester :IV**

HOUR	I	II	10.40 a.m TO 10.55 a.m	III	IV	12.35 p.m. TO 1.20 p.m.	V	VI	2.50 p.m. TO 3.05 p.m.	VII	VIII
DAY/ TIME	09.00a.m TO 09.50 a.m.	09.50a.m. TO 10.40a.m.		10.55 a.m. TO 11.45 a.m.	11.45 a.m. TO 12.35p.m.		1.20 p.m. TO 2.05p.m.	2.05 p.m. TO 2.50p.m.		3.05 p.m. TO 3.50 p.m.	3.50 p.m. TO 4.30 p.m.
MONDAY			BREAK	DBMS		LUNCH			BREAK		
TUESDAY	DBMS										
WEDNESDAY				DBMS							
THURSDAY							DBMS	DBMS LAB		DBMS LAB	
FRIDAY										DBMS	
SATURDAY							DBMS				

S.No	Subject Code	Name of the Subject	Name of the Staff	No of hours
1	CS3492	DataBase Management System		6
			TOTAL	6

	Prepared by	Verified by	Authorized by
Sign:			
Name:		Mr.S.PRABAKARAN	Dr.M.VIJAYAKUMAR
	Time Table Incharge	HOD	Principal

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tiruppur.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT : B.E - CSE

YEAR / SEM : II / IV

ACADEMIC YEAR : 2023-2024

STUDENT NAME LIST

S.No	Reg Number	Name of the Student	Dayscholar/ Hostel
1	732422104001	ABISHEK J	Dayscholar
2	732422104002	AKILESH KUMAR S	Dayscholar
3	732422104004	ARUNKUMAR A	Dayscholar
4	732422104005	ASWIN S	Dayscholar
5	732422104006	BASHARATH MAHMOOD S	Dayscholar
6	732422104007	BASKAR S	Dayscholar
7	732422104008	DEEPAK V	Hostel
8	732422104009	DEEPAKRAJ R	Dayscholar
9	732422104010	DHARSHINI R	Dayscholar
10	732422104011	DHARUN T	Dayscholar
11	732422104013	FARGATH A	Hostel
12	732422104014	GUHAN K R	Dayscholar
13	732422104015	GURUPRASAD R	Dayscholar
14	732422104016	HARIJEEVA M	Hostel
15	732422104017	HARIKRISHNAN B	Dayscholar
16	732422104018	HARIPRIYA V	Dayscholar
17	732422104019	IRUDHAYA VISHVA A	Dayscholar
18	732422104020	JEENA D	Dayscholar
19	732422104021	JEEVA S	Dayscholar
20	732422104022	JEEVA S	Dayscholar
21	732422104023	KALAISELVAN R	Dayscholar
22	732422104024	KARTHIKA K	Hostel
23	732422104025	KEERTHIKA S	Dayscholar
24	732422104027	MAHESWARI T	Dayscholar
25	732422104028	MATHAVAN C	Dayscholar
26	732422104030	MOHAMMED THAMEEMMUL ANSARI C J	Dayscholar
27	732422104031	NANDHINI M R	Dayscholar

28	732422104032	NAVANEETHA KRISHNAN M	Dayscholar
29	732422104033	NAVEENA M	Dayscholar
30	732422104034	NAVEEN KUMAR V	Dayscholar
31	732422104035	PANDI E	Dayscholar
32	732422104036	REVATHI P	Dayscholar
33	732422104037	SABARIYANANDHAN T	Dayscholar
34	732422104038	SARAN B	Dayscholar
35	732422104039	SARAVANAN R	Dayscholar
36	732422104040	SELVAPRIYA C	Dayscholar
37	732422104041	SHANMATHI C T	Dayscholar
38	732422104042	SIKKANTHAR BATHUSHA R	Dayscholar
39	732422104043	SRI RAJ S	Dayscholar
40	732422104044	SUBASH M	Dayscholar
41	732422104045	SWATHI R	Dayscholar
42	732422104046	THIRUPATHI P	Dayscholar
43	732422104047	VASANTH A	Dayscholar
44	732422104048	VASANTHKUMAR P	Dayscholar
45	732422104301	ARUN PRAKASH G	Dayscholar
46	732422104302	DHANUSH B	Dayscholar
47	732422104303	GUNASEKAR S	Dayscholar
48	732422104304	NITHISH KUMAR V	Dayscholar

	Prepared by	Verified by
Sign:		
Name:		
	Faculty	HOD

ॐ

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tiruppur

SUBJECT INFORMATION RECORD

Department : CSE

Subject : Database Management System

Year : II

Semester : IV

Last year handled by : —

Percentage of Result (last year) : —

Quality Objectives : To produce result more than 80 % in University exam

Reference Book : C.J. Date, A. Kannan, S. Swamynathan, "An Introduction to Database Systems", Eighth Edition, Pearson Education, 2006.

	Prepared By	Approved By
Sign:		
Name:		
	Faculty	HOD

SYLLABUS

CS3492 DATABASE MANAGEMENT SYSTEMS L T P C 3 0 0 3

COURSE OBJECTIVES:

- To learn the fundamentals of data models, relational algebra and SQL
- To represent a database system using ER diagrams and to learn normalization techniques
- To understand the fundamental concepts of transaction, concurrency and recovery
- processing To understand the internal storage structures using different file and indexing techniques
- which will help in physical DB design To have an introductory knowledge about the Distributed databases, NOSQL and database security

UNIT I RELATIONAL DATABASES 10

Purpose of Database System – Views of data – Data Models – Database System Architecture – Introduction to relational databases – Relational Model – Keys – Relational Algebra – SQL fundamentals – Advanced SQL features – Embedded SQL– Dynamic SQL

UNIT II DATABASE DESIGN 8

Entity-Relationship model – E-R Diagrams – Enhanced-ER Model – ER-to-Relational Mapping – Functional Dependencies – Non-loss Decomposition – First, Second, Third Normal Forms, Dependency Preservation – Boyce/Codd Normal Form – Multi-valued Dependencies and Fourth Normal Form – Join Dependencies and Fifth Normal Form

UNIT III TRANSACTIONS 9

Transaction Concepts – ACID Properties – Schedules – Serializability – Transaction support in SQL – Need for Concurrency – Concurrency control –Two Phase Locking- Timestamp – Multiversion – Validation and Snapshot isolation– Multiple Granularity locking – Deadlock Handling – Recovery Concepts – Recovery based on deferred and immediate update – Shadow paging – ARIES Algorithm

UNIT IV IMPLEMENTATION TECHNIQUES 9

RAID – File Organization – Organization of Records in Files – Data dictionary Storage – Column Oriented Storage– Indexing and Hashing –Ordered Indices – B+ tree Index Files – B tree Index Files – Static Hashing – Dynamic Hashing – Query Processing Overview – Algorithms for Selection, Sorting and join operations – Query optimization using Heuristics - Cost Estimation.

UNIT V ADVANCED TOPICS 9

Distributed Databases: Architecture, Data Storage, Transaction Processing, Query processing and optimization – NOSQL Databases: Introduction – CAP Theorem – Document Based systems – Key value Stores – Column Based Systems – Graph Databases. Database Security: Security issues – Access control based on privileges – Role Based access control – SQL Injection – Statistical Database security – Flow control – Encryption and Public Key infrastructures – Challenges

TOTAL:45 PERIODS

TEXT BOOKS:

1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, "Database System Concepts", Seventh Edition, McGraw Hill, 2020.

2. Ramez Elmasri, Shamkant B. Navathe, "Fundamentals of Database Systems", Seventh Edition, Pearson Education, 2017

REFERENCES:

1. C.J.Date, A.Kannan, S.Swamynathan, "An Introduction to Database Systems", Eighth Edition, Pearson Education, 2006.

SASURIE COLLEGE OF ENGINEERING
Vijayamangalam, Tirupur

(Accredited by NAAC, Under 2f and 12B status)

LESSON PLAN

Faculty Name :
Department : CSE
Subject / Code : DATABASE MANAGEMENT SYSTEM/CS3492
Academic Year : 2023-2024

Designation: Assistant Professor
Semester/ Year: IV/ II

S.No.	Proposed Date	Proposed Period	Details of Topic Covered	TA	Ref.	Actual Date	Actual Period	Remarks
			UNIT I RELATIONAL DATABASES					
1	4/3/24	3	Purpose of Database System, Views of data	1	1	4/3/24	3	
2	5/3/24	1	Data Models	1	1	5/3/24	1	
3	5/3/24	2	Database System Architecture	1	1	5/3/24	2	
4	6/3/24	3	Introduction to relational databases	1	1	6/3/24	3	
5	6/3/24	3	Relational Model	1	1	6/3/24	3	
6	7/3/24	5	Keys , Relational Algebra	1	1	7/3/24	5	
7	7/3/24	6	SQL fundamentals	1	1	7/3/24	6	
8	9/3/24	1	Advanced SQL features	1	1	9/3/24	1	
9	11/3/24	6	Embedded SQL	1	1	11/3/24	6	
10	14/3/24	1	Dynamic SQL	1	1	14/3/24	1	
			UNIT II DATABASE DESIGN					
1	15/3/24	8	Entity-Relationship model,E-R Diagrams	1	1	15/3/24	8	
2	18/3/24	3	Enhanced-ER Model,ER-to-Relational Mapping	1	1	18/3/24	3	
3	19/3/24	1	Functional Dependencies , Non-loss Decomposition	1	1	19/3/24	1	
4	22/3/24	7	First normal form, Second normal form	1	1	22/3/24	7	
5	25/3/24	3	Third Normal Forms,Dependency Preservation	1	1	25/3/24	3	
6	25/3/24	4	Boyce/Codd Normal Form	1	1	25/3/24	4	
7	28/3/24	5	Multi-valued Dependencies and Fourth Normal Form	1	1	28/3/24	5	
8	01/4/24	3	Join Dependencies and Fifth Normal Form	1	1	01/4/24	3	
			UNIT III TRANSACTIONS					
1	01/4/24	5	Transaction Concepts , ACID Properties	1	1	01/4/24	5	
2	05/4/24	7	Schedules , Serializability	1	1	05/4/24	7	
3	08/4/24	3	Transaction support in SQL., Need for Concurrency	1	1	08/4/24	3	
4	09/4/24	1	Concurrency control ,Two Phase Locking-	1	1	09/4/24	1	
5	13/4/24	1	Timestamp , Multiversion	1	1	13/4/24	1	
6	13/4/24	5	Validation and Snapshot isolation-,Multiple Granularity locking	1	1	13/4/24	5	
7	15/4/24	3	Deadlock Handling ,Recovery Concepts	1	1	15/4/24	3	

8	26/4/24	7	Recovery based on deferred and immediate update	1	1	26/4/24	7	
9	27/4/24	6	Shadow paging , ARIES Algorithm	1	1	27/4/24	6	
			UNIT IVIMPLEMENTATION TECHNIQUES					
1	29/4/24	3	RAID – File Organizatio	1	1	29/4/24	3	
2	30/4/24	1	Organization of Records in Files ,Data dictionary Storage	1	1	30/4/24	1	
3	06/5/24	3	Column Oriented Storage, Indexing and Hashing	1	1	06/5/24	3	
4	07/5/24	1	Ordered Indices , B+ tree Index Files	1	1	07/5/24	1	
5	09/5/24	5	B tree Index Files, Static Hashing	1	1	09/5/24	5	
6	09/5/24	6	Dynamic Hashing ,Query Processing Overview	1	1	09/5/24	6	
7	10/5/24	7	Algorithms for Selection	1	1	10/5/24	7	
8	11/5/24	5	Sorting and join operations	1	1	11/5/24	5	
9	13/5/24	3	Query optimization using Heuristics , Cost Estimation	1	1	13/5/24	3	
			UNIT V ADVANCED TOPICS					
1	17/5/24	7	Distributed Databases: Architecture, Data Storage	1	1	17/5/24	7	
2	20/5/24	3	Transaction Processing, Query processing and optimization	1	1	20/5/24	3	
3	21/5/24	1	NOSQL Databases,Introduction ,CAP Theorem ,Document Based systems	1	1	21/5/24	1	
4	22/5/24	3	Key value Stores , Column Based Systems	1	1	22/5/24	3	
5	23/5/24	5	Graph Databases	1	1	23/5/24	5	
6	27/5/24	3	Database Security, Security issues	1	1	27/5/24	3	
7	28/5/24	1	Access control based on privileges , Role Based access control	1	1	28/5/24	1	
8	29/5/24	3	SQL Injection , Statistical Database security ,Flow control	1	1	29/5/24	3	
9	04/6/24	1	Encryption and Public Key infrastructures ,Challenges	1	1	04/6/24	1	

Reference books (Ref):

1.. C.J.Date, A.Kannan, S.Swamynathan, "An Introduction to Database Systems", Eighth Edition, Pearson Education, 2006.

Teaching Aids (TA):

1. Black Board with Chalk
2. Overhead Projector
3. LCD Projector
4. Others (Field vists, Charts, Cutset Models)

	Prepared by	Verified by	Authorized by
Sign:			
Name:	Faculty	HOD	Principal

ॐ

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tiruppur

TEST PLAN FOR SUBJECT

Subject: Database Management System Faculty: S.Prabakaran

Semester: IV

Year: II

Department: CSE

S. No.	Description	Planned Date/Month	Actual Conducted Date / Month	Remarks
1.	Unit test - I	27/3/2024	27/3/2024	
2.	Unit test - II	17/4/2024	17/4/2024	
3.	Unit test - III	11/5/2024	11/5/2024	
4.	Unit test - IV	24/5/2024	24/5/2024	
5.	Model Exam - I	01/06/2024	01/06/2024	

	Prepared By	Approved By
Sign:		
Name:		
	Faculty	HOD

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

RESULT ANALYSIS OF TEST

Subject : Database Management System Date : 2.4.2024

Class : II Year Department : CSE

Semester : IV

Exam details & date : Unit Test-1 & 27/3/24

Faculty : S. Prabakaran

Number of students : 47

No. of students attended : 33

No. of students absent : 14

No. of students passed : 33

No. of students failed : -

Percentage of failures : 0.1.

RESULT DATA:

Marks	0-25	26-50	51-75	76-90	91-100
No. of Students	-	33			

	Prepared By	Approved By
Sign:		
Name:		
	Faculty	HoD

SASURIE INSTITUTIONS

ॐ SASURIE COLLEGE OF ENGINEERING

Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

RESULT ANALYSIS OF TEST

Subject : Database Management System Date : 19.4.24

Class : II Year Department : CSE

Semester : IV

Exam details & date : Unit Test - II & 17/4/24

Faculty : S. Prabakaran

Number of students : 47

No. of students attended : 46

No. of students absent : 1

No. of students passed : 38

No. of students failed : 8

Percentage of failures : 17.1.

RESULT DATA:

Marks	0-25	26-50	51-75	76-90	91-100
No. of Students	8	38			

	Prepared By	Approved By
Sign:		
Name:		
	Faculty	HoD

SASURIE INSTITUTIONS

SASURIE
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

RESULT ANALYSIS OF TEST

Subject : Database Management System Date : 14/5/24

Class : II Year Department : CSE

Semester : IV

Exam details & date : Unit Test - III & 11/5/24

Faculty : S. Prabakaran

Number of students : 47

No. of students attended : 45

No. of students absent : 2

No. of students passed : 42

No. of students failed : 3

Percentage of failures : 6.6%

RESULT DATA:

Marks	0-25	26-50	51-75	76-90	91-100
No. of Students	3	42			

	Prepared By	Approved By
Sign:		
Name:		
	Faculty	HoD

SASURIE INSTITUTIONS

ॐ
SASURIE
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

RESULT ANALYSIS OF TEST

Subject : Database Management System Date : 27/5/24

Class : II Year

Department : CSE

Semester : IV

Exam details & date : Unit Test - IV & 24/5/24

Faculty : S. Prabakaran

Number of students : 47

No. of students attended : 47

No. of students absent : –

No. of students passed : 46

No. of students failed : 1

Percentage of failures : 2.1.1.

RESULT DATA:

Marks	0-25	26-50	51-75	76-90	91-100
No. of Students	1	46			

	Prepared By	Approved By
Sign:		
Name:		
	Faculty	HoD

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

RESULT ANALYSIS OF TEST

Subject : Database Management System Date : 4/6/24

Class : II Year Department : CSE

Semester : IV

Exam details & date : Model Exam - I & 01/06/24

Faculty : S. Prabakaran

Number of students : 47

No. of students attended : 47

No. of students absent : -

No. of students passed : 43

No. of students failed : 4

Percentage of failures : 8.5 %

RESULT DATA:

Marks	0-25	26-50	51-75	76-90	91-100
No. of Students	-	4	25	16	2

	Prepared By	Approved By
Sign:		
Name:		
	Faculty	HoD

SASURIE
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

CORRECTIVE ACTION REPORT

Dept. : CSE Year : II

Subject : Database Management System Semester : IV

S.No	Unit Test	Percentage of marks	Root Cause (Metrics)	Corrective Action	Deadline date	Remarks
1	Unit test I	100 %	–	–	–	–
2	Unit test II	83 %	–	–	–	–
3	Unit test III	93 %	–	–	–	–
4	Unit test IV	97 %	–	–	–	–
5	Model - I	91 %	–	–	–	–

	Prepared By	Approved By
Sign:		
Name:		
	Faculty	HOD

SASURIE INSTITUTIONS

SASURIE
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

QUALITY OBJECTIVE MONITORING RECORD

Department : CSE

Year : II

Semester : IV

Subject : Database Management System

S.No	Quality Objective	Unit Test-I		Unit Test-II		Unit Test-III		Unit Test-IV		Model Exam - 1		Model Exam - 2	
		Expecting result	Obtained result	Expecting result	Obtained result	Expecting result	Obtained result	Expecting result	Obtained result	Expecting result	Obtained result	Expecting result	Obtained result
1.	To Produce result more than 80% in University Exam	80%	100%	90%	83%	90%	93%	90%	97%	90%	91%		

	Prepared By	Approved By
Sign:		
Name:		
	Faculty	HOD

SASURIE
COLLEGE OF ENGINEERING
Vijayamangalam, Tiruppur

DEPARTMENT : B.E - CSE
YEAR / SEM : II / IV
ACADEMIC YEAR : 2023-2024

STUDENT MARK LIST

S.No	Reg Number	Name of the Student	Unit Test - 1	Unit Test - 2	Unit Test - 3	Unit Test - 4	Model test - 1
1	732422104001	ABISHEK J	AB	36	AB	36	52
2	732422104002	AKILESH KUMAR S	AB	43	39	39	83
3	732422104004	ARUNKUMAR A	12	31	35	34	30
4	732422104005	ASWIN S	26	37	32	35	73
5	732422104006	BASHARATH MAHMOOD S	34	39	36	36	72
6	732422104007	BASKAR S	31	45	44	33	78
7	732422104008	DEEPAK V	30	32	AB	31	12
8	732422104009	DEEPAKRAJ R	32	29	22	34	71
9	732422104010	DHARSHINI R	47	47	45	39	91
10	732422104011	DHARUN T	33	42	35	31	62
11	732422104013	FARGATH A	40	44	35	36	74
12	732422104014	GUHAN K R	29	34	31	39	61
13	732422104015	GURUPRASAD R	AB	35	34	37	77
14	732422104016	HARIJEEVA M	35	39	34	37	56
15	732422104017	HARIKRISHNAN B	37	42	35	39	77
16	732422104018	HARIPRIYA V	46	41	38	45	77
17	732422104019	IRUDHAYA VISHVA A	39	41	42	39	84
18	732422104020	JEENA D	39	35	39	37	82
19	732422104021	JEEVA S	AB	AB	29	31	47
20	732422104022	JEEVA S	AB	39	29	34	59
21	732422104023	KALAISELVAN R	31	42	29	36	78
22	732422104024	KARTHIKA K	44	41	37	37	82
23	732422104027	MAHESWARI T	34	35	27	21	50
24	732422104028	MATHAVAN C	39	36	41	34	80
25	732422104030	MOHAMMED THAMEEMMUL ANSARI C J	AB	34	24	36	81
26	732422104031	NANDHINI M R	40	36	28	34	74
27	732422104032	NAVANEETHA KRISHNAN M	AB	29	23	30	69
28	732422104033	NAVEENA M	36	34	31	38	81
29	732422104034	NAVEEN KUMAR V	34	24	24	27	71
30	732422104035	PANDI E	31	23	4	36	70
31	732422104036	REVATHI P	45	46	44	37	88
32	732422104037	SABARIYANANDHAN T	35	32	33	29	72
33	732422104038	SARAN B	39	39	39	37	81
34	732422104039	SARAVANAN R	31	37	31	32	75
35	732422104040	SELVAPRIYA C	43	45	45	42	91

36	732422104041	SHANMATHI C T	AB	39	39	38	79
37	732422104042	SIKKANTHAR BATHUSHA R	32	35	34	26	72
38	732422104043	SRI RAJ S	36	27	42	35	75
39	732422104044	SUBASH M	AB	29	31	29	54
40	732422104045	SWATHI R	34	5	34	26	47
41	732422104046	THIRUPATHI P	AB	21	36	27	43
42	732422104047	VASANTH A	AB	29	34	25	49
43	732422104048	VASANTHKUMAR P	34	21	29	27	78
44	732422104301	ARUN PRAKASH G	AB	15	27	30	71
45	732422104302	DHANUSH B	28	22	29	32	59
46	732422104303	GUNASEKAR S	AB	24	31	26	62
47	732422104304	NITHISH KUMAR V	AB	27	37	35	71

SIGN			
NAME			
	FACULTY	HOD	PRINCIPAL

Register Number: | | | | | | | | | | | |

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tiruppur.

UNIT TEST I			Date/Session	27.03.2024	Marks	50
Course code	CS3492	Course Title	Database Management System			
Regulation	2021	Duration	1:30 Hours	Academic Year	2023-2024	
Year	II	Semester	IV	Department	CSE	

COURSE OUTCOMES

CO1:	Construct SQL Queries using relational algebra
CO2:	Design database using ER model and normalize the database
CO3:	Construct queries to handle transaction processing and maintain consistency of the database
CO4:	Compare and contrast various indexing strategies and apply the knowledge to tune the performance of the database
CO5:	Appraise how advanced databases differ from Relational Databases and find a suitable database for the given requirement.

Q.No.	Question	CO	BTS
	PART A (Answer all the Questions 10 x 2 = 20 Marks)		
1	Define data abstraction and different levels of data abstraction.	CO1	R
2	What is DBMS ?Why do we need a DBMS?	CO1	R
3	Differentiate between Dynamic SQL and static SQL?	CO1	A
4	What is different primary key and foreign key?	CO1	U
5	Name the categories of SQL commands.	CO1	R
6	What is the data definition language? give example	CO1	U
7	List the reason why null value might be introduced into the database.	CO1	R
8	Why key is essential? & write the different types of keys.	CO1	A
9	What is meant by instance and schema of the database?	CO1	R
10	Different between database system and conventional file system.	CO1	A
	PART B (Answer all the Questions 2 x 15 = 30 Marks)		
11a	Write about database system Architecture with neat diagram.	CO1	E
	OR		
11b	Briefly explain about Database system architecture	CO1	C
12a	Explain select project, Cartesian product and join operations in relational	CO1	A
	OR		
12b	Explain the codd's rules for relational design.	CO1	C

Course Faculty

HOD

Principal

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tiruppur

UNIT TEST I			Date/Session	27.03.2024	Marks	50
Course code	CS3492	Course Title	Database Management System			
Regulation	2021	Duration	1:30 Hours	Academic Year	2023-2024	
Year	II	Semester	IV	Department	CSE	

PART A

1. Define data abstraction and different levels of data abstraction.

- Data abstraction is a technique in database systems to simplify complex data by hiding low-level details and focusing on essential aspects of data for easier management and interaction.
- There are three levels of data abstraction:
  - **Physical level**: The lowest level, describing how data is stored on the physical storage media.
  - **Logical level**: Describes the structure of the database and relationships, without detailing how data is stored.
  - **View level**: The highest level, defining different views or ways data can be seen by different users based on access needs.

2. What is DBMS?

·DBMS stands for **Database Management System**. It is a software that enables the creation, management, and use of databases, providing an interface for users to interact with stored data.

·It helps in organizing data, ensuring data integrity, security, and supporting transactions, making it easier for users to store, retrieve, and manage data efficiently.

Why do we need a DBMS?

- A Database Management System (DBMS) is essential for managing large volumes of data in an organized and efficient manner, allowing for storage, retrieval, and manipulation of data.
- DBMS ensures data security, integrity, and ease of access, along with providing functionalities like backup, recovery, and concurrency control.

3. Differentiate between Dynamic SQL and Static SQL.

- **Dynamic SQL**: SQL statements that are constructed and executed at runtime, allowing for more flexibility and handling of various scenarios, as the SQL code can adapt to changing inputs.
- **Static SQL**: SQL statements written and embedded in the application code, known at compile-time, offering better performance and security as the SQL commands are fixed.

4. What is the difference between primary key and foreign key?

- **Primary Key**: A unique identifier for each record in a table, ensuring that no two rows have the same value for this field.
- **Foreign Key**: A field in one table that is a primary key in another table, used to establish a relationship between the two tables.

5. **Name the categories of SQL commands.**

- **Data Definition Language (DDL)**: Commands that define and structure database objects, such as CREATE, ALTER, and DROP.
- **Data Manipulation Language (DML)**: Commands that handle data operations, such as INSERT, UPDATE, and DELETE.
- **Data Control Language (DCL)**: Commands that manage access permissions, such as GRANT and REVOKE.
- **Transaction Control Language (TCL)**: Commands that manage transaction operations, such as COMMIT and ROLLBACK.

6. **What is the data definition language? Give an example.**

- Data Definition Language (DDL) is a set of SQL commands used to define and modify the structure of database objects, like tables and indexes.
- **Example**: CREATE TABLE students (id INT PRIMARY KEY, name VARCHAR(50));

7. **List the reasons why a null value might be introduced into the database.**

- The value is unknown or missing for a certain attribute in a record.
- The attribute does not apply to the specific record or entity, hence leaving it empty (e.g., "middle name" for someone without one).

8. **Why is a key essential in a database?**

- ·Keys are essential for uniquely identifying each record in a table, ensuring data integrity and enabling efficient retrieval of records.
- They also establish relationships between tables, allowing for the linking of related data across multiple tables.

What are the different types of keys?

- **Primary Key**: Unique identifier for each row in a table.
- **Foreign Key**: Establishes a relationship between two tables.
- **Candidate Key**: Any attribute or set of attributes that can uniquely identify a row.
- **Super Key**: A combination of attributes that uniquely identifies a row.
- **Composite Key**: A key that consists of more than one attribute to uniquely identify a row.
- **Alternate Key**: A candidate key that was not chosen as the primary key.

9. **What is meant by instance and schema of the database?**

- **Instance**: Refers to the actual data stored in a database at a specific point in time; it is dynamic and changes as data is added or modified.
- **Schema**: The overall structure or design of the database, defining tables, fields, and relationships; it is relatively static and only changes if the database structure is altered.

10. **Differentiate between database systems and conventional file systems.**

- **Database Systems**: Organized collections of data managed by DBMS, offering features like data integrity, concurrency, and secure access.
- **Conventional File Systems**: Simple storage systems without DBMS features, lacking centralized data management, often resulting in data redundancy and inconsistency

PART B

11a. Write about Database System Architecture with a Neat Diagram.

Database system architecture defines how different components of a Database Management System (DBMS) interact to manage, store, process, and retrieve data. It plays a crucial role in designing efficient database systems by organizing the interaction between users, applications, and data storage in a structured way.

The **three-tier architecture** is a commonly used structure that divides the database system into three main layers:

Presentation Layer (User Interface Layer):

1. This is the topmost layer, where end-users interact with the system through a graphical user interface (GUI) or an application.
2. This layer provides interfaces that allow users to perform operations such as querying, updating, and retrieving data.
3. Typically, the presentation layer doesn't interact directly with the database; instead, it forwards requests to the application layer.
4. By keeping the user interface separate, this layer allows easy modifications in the appearance and user interactions without impacting the database.
2.

Application Layer (Business Logic Layer):

1. The application layer, or middle layer, serves as the intermediary between the user interface and the database.
2. It contains the core logic for processing user requests and ensuring data accuracy, often implementing validation, data processing, and error handling.
3. This layer transforms user inputs into database operations and enforces business rules. For example, it may check if an item is in stock before processing a purchase order.
4. This separation allows enhanced security since direct access to data is restricted, and only authorized requests can modify or access the database.
5. Additionally, it improves scalability, as the application layer can be adjusted to handle growing numbers of users or increased data processing needs.
3.

Database Layer (Data Storage Layer):

1. The database layer is where the data resides, organized and managed by the DBMS.
2. This layer is responsible for storing data physically on the storage media and managing access through indexing, query processing, and transaction handling.
3. Features of the DBMS, such as data consistency, concurrency control, and recovery mechanisms, operate here to maintain data integrity.
4. The database layer can also provide optimizations like indexing and caching to improve query performance, particularly in large-scale systems.
5. By keeping the data management isolated, this layer allows the application and presentation layers to function independently of the physical data storage details

Diagram:

Presentation Layer
(User Interface)
|
Application Layer
(Business Logic)

Database Layer
(Data Storage)

11b. Briefly Explain Database System Architecture

Database system architecture organizes the interaction between different components in a DBMS, creating a structured approach for data management. This structure ensures efficient data processing, secure access, and system reliability by separating user interfaces, application logic, and data storage.

Types of Database System Architectures:

Two-tier Architecture:

1. In this simpler setup, the client application communicates directly with the database server.
2. **Client Layer**: Users interact with the application on this layer, sending requests and receiving data directly from the database.
3. **Database Layer**: The DBMS manages data storage, retrieval, and integrity, directly responding to client requests.
4. This type is suitable for small-scale applications where direct access provides faster performance with limited users.
5. However, it lacks the ability to scale well for large user bases and may present security risks since each client has direct access to the database.
2.

Three-tier Architecture:

1. Three-tier architecture introduces an **application layer** between the user interface and database, making it a preferred choice for larger applications.
2. **Presentation Layer (User Layer)**: The top layer where users interact with the system, often through web browsers, mobile apps, or GUIs.
3. **Application Layer (Business Logic Layer)**: Acts as an intermediary that processes user requests, enforces business rules, and validates inputs before passing them to the database. This layer also handles responses back to the user, adding security by managing data access.
4. **Database Layer (Data Layer)**: The lowest layer responsible for storing, retrieving, and organizing data. The DBMS in this layer manages transactions, indexing, and backup, among other functions.
5. This architecture is more secure and scalable. Since the database is isolated, it's protected from unauthorized access, and the middle layer allows for consistent data handling and validation.

Benefits of Three-Tier Architecture:

- **Enhanced Security**: Direct access to the database is restricted by the application layer, preventing unauthorized access.
- **Scalability**: Additional users and resources can be managed more effectively since the application layer can be scaled independently.
- **Maintainability**: Changes in business rules or database structure can be managed in the application layer without impacting other components.
- **Flexibility**: Modifications in the user interface or data structure do not affect each other directly, allowing for better data independence.

12a. Explain Select, Project, Cartesian Product, and Join Operations in Relational Algebra.

In relational algebra, these fundamental operations help in retrieving and manipulating data from relational databases by performing various transformations on tables (relations). Each operation has a specific role in filtering, combining, or restructuring data.

Operations in Relational Algebra:

Select (σ):

1. The Select operation is used to filter rows from a relation based on a given condition. Only rows that satisfy the condition are included in the result.
2. **Notation**: σ condition (Relation)
3. **Example**: If we have an Employees relation and want to retrieve employees older than 25, we write σ age > 25 (Employees). This will return only rows where the age attribute is greater than 25.
4. **Usage**: Select helps narrow down data based on specific criteria, making it easier to work with relevant subsets of a larger dataset.

Project (π):

1. The Project operation is used to select specific columns (attributes) from a relation, essentially reducing the number of columns in the output.
2. **Notation**: π attributes (Relation)
3. **Example**: π name, age (Employees) retrieves only the name and age columns from the Employees table, omitting other columns.
4. **Usage**: Project is useful when only certain attributes are needed for analysis or reporting, reducing the volume of data retrieved.

Cartesian Product

1. The Cartesian Product operation combines two relations by pairing every row from the first relation with every row from the second. It is a foundational operation for other complex operations, like join.
2. **Notation**: Relation1 × Relation2
3. **Example**: If we have Employees and Departments, Employees × Departments results in a new relation with all possible combinations of employees and departments.
4. **Usage**: The Cartesian Product is used primarily as an intermediate step for joins but can be computationally intensive due to the large number of combinations generated.

Join

1. The Join operation combines rows from two relations based on a related attribute or condition, matching rows with common values.
2. **Types of Joins**:
    1. **Theta Join**: Matches rows based on a condition (condition).
    2. **Equi-Join**: A join with equality condition (=).
    3. **Natural Join**: Matches rows with the same value in common attributes, eliminating duplicate columns.
3. **Notation**: Relation1 condition Relation2
4. **Example**: Employees Employees.department_id = Departments.department_id Departments combines employee and department data where department_id matches.
5. **Usage**: Joins are essential for combining related data from multiple tables, allowing complex queries across related entities in a database.

12b. Explain Codd's Rules for Relational Design.

E.F. Codd, the pioneer of the relational database model, formulated 12 rules (commonly called "Codd's Rules") to define what constitutes a relational database and how it should be designed.

These rules ensure that a database fully utilizes relational principles for efficient, consistent, and flexible data management.

Codd's 12 Rules for Relational Database Systems:

Rule 0 (Foundation Rule):

1. A relational database management system (RDBMS) must manage data entirely through its relational capabilities, adhering to all other rules.
2. This rule emphasizes that an RDBMS must fundamentally be relational.

Rule 1 (Information Rule):

1. All information in a database should be stored in tables (relations), with rows and columns.
2. This structure allows for a consistent, organized way of storing data.

Rule 2 (Guaranteed Access):

1. Each value in a table must be accessible by specifying the table name, primary key, and column name.
2. This rule ensures that data can be retrieved using a straightforward method, regardless of table complexity.

Rule 3 (Systematic Treatment of Nulls):

1. Null values must be supported to represent missing or inapplicable data, allowing for three states: true, false, and unknown.
2. This flexibility accommodates situations where data is incomplete or unknown.

Rule 4 (Dynamic Online Catalog):

1. The database's metadata (catalog) should also be stored in tables and be accessible via SQL.
2. Users can query database structure information in the same way as data, supporting transparency.

Rule 5 (Comprehensive Data Sub-language):

1. The RDBMS should support a powerful data sub-language, such as SQL, for defining, manipulating, and controlling data and transactions.
2. This ensures a complete command set for interacting with the database.

Rule 6 (View Updating Rule):

1. Users should be able to modify data through views, as long as such modifications do not affect the underlying integrity.
2. This feature allows flexibility in data presentation while preserving data accuracy.

Rule 7 (High-Level Insert, Update, and Delete):

1. The system must support set-based operations, allowing rows to be inserted, updated, or deleted in bulk rather than one at a time.
2. This capability improves efficiency in data manipulation, especially for large datasets.

Rule 8 (Physical Data Independence):

1. The application should remain unaffected by changes in how data is physically stored.
2. This rule supports ease of optimization and hardware changes without affecting applications.

Rule 9 (Logical Data Independence):

1.The application should remain unaffected by changes to the logical structure of the database.

2.Logical data independence protects applications from schema modifications like adding new fields.

Rule 10 (Integrity Independence):

1.This ensures that data rules are defined at the database level, maintaining consistency and security.

2.Logical data independence protects application from schema modifications like adding new fields

Rule 11 (Distribution Independence):

1.The RDBMS should function correctly regardless of data being distributed across multiple locations.

2.This rule supports flexibility for distributed databases, enabling data to be accessed or modifieds seamlessly

Rule 12 (Non-Subversion Rule):

1. Low-level access methods should not bypass or undermine the RDBMS's integrity constraints or rules.
2. This rule prevents unauthorized modifications, ensuring data integrity and consistency.

Register Number:

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tiruppur

UNIT TEST II			Date/Session	17.04.2024	Marks	50
Course code	CS3492	Course Title	Database Management System			
Regulation	2021	Duration	1:30 Hours	Academic Year	2023-2024	
Year	II	Semester	IV	Department	CSE	

COURSE OUTCOMES

CO1:	Construct SQL Queries using relational algebra
CO2:	Design database using ER model and normalize the database
CO3:	Construct queries to handle transaction processing and maintain consistency of the database
CO4:	Compare and contrast various indexing strategies and apply the knowledge to tune the performance of the database
CO5:	Appraise how advanced databases differ from Relational Databases and find a suitable database for the given requirement.

Q.No.	Question	CO	BTS
	PART A (Answer all the Questions 10 x 2 = 20 Marks)		
1	Explain Entity relationship model.	CO2	R
2	Give the limitation of E-R model? How do you overcome this?	CO2	A
3	What is an entity?	CO2	R
4	Define functional dependency?	CO2	R
5	Why certain functional dependencies are called trival functional dependency?	CO2	R
6	Define normalization.	CO2	U
7	What is multivalued dependency?.	CO2	R
8	Describe BCNF and describe a relation which is in BCNF?	CO2	A
9	Why it is necessary to decompose a relation?	CO2	A
10	Explain atleast two desirable properties of decomposition .	CO2	R
	PART B (Answer all the Questions 2 x 15 = 30 Marks)		
11a	Explain in detail about normal forms.	CO2	E
	OR		
11b	Explain the various components of ER diagram with example Briefly.	CO2	E
12a	Explain in detail about multivalued dependencies and forth normal form.	CO2	C
	OR		
12b	Discuss about integrity constraints in SQL?	CO2	U

Course Faculty

HOD

Principal

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tiruppur.

UNIT TEST II			Date/Session	17.04.2024	Marks	50
Course code	CS3492	Course Title	Database Management System			
Regulation	2021	Duration	1:30 Hours	Academic Year	2023-2024	
Year	II	Semester	IV	Department	CSE	

PART A

1. Explain Entity relationship model.

The **Entity-Relationship Model (ER Model)** is a conceptual framework used in database design to visually represent the structure and relationships of data within a system.

2. Give the limitation of E-R model? How do you overcome this?

- Limited Representation of Data Constraints
- Difficulty Handling Complex Relationships
- Lack of Support for Temporal Data
- Limited Scalability for Complex Systems
- Inflexibility in Representing Advanced Data Types

3. What is an entity?

An **entity** in the context of database design and the Entity-Relationship (ER) model is a distinct object, concept, or "thing" that can be identified and represented within a system. Entities represent real-world items or abstractions that need to be stored and managed in a database.

4.Define functional dependency.

Functional dependency is a relationship in which one attribute uniquely determines another attribute within a relation. For example, if attribute A determines attribute B, knowing A's value means we can uniquely identify B's value. Functional dependencies help establish rules for data integrity and are critical in database normalization.

5.Why are certain functional dependencies called trivial functional dependencies?

Trivial functional dependencies occur when an attribute functionally determines itself, such as $\{A\} \rightarrow A$, or when one subset determines another subset within the same attributes. These dependencies are inherently satisfied, as the determinant set contains or equals the determined attribute. Trivial dependencies are always present but do not affect normalization.

6.Define normalization.

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. It involves dividing tables into smaller relations and establishing dependencies to minimize anomalies during data manipulation. The goal of normalization is to create a structure that supports efficient storage and accurate retrieval of data.

7.What is a multivalued dependency?

A multivalued dependency exists when an attribute in a relation uniquely determines multiple independent values of another attribute, creating a one-to-many relationship. This situation leads to redundancy

if not managed properly. Multivalued dependencies are common in scenarios with repeating data, such as lists of skills for a person.

8.Describe BCNF and describe a relation which is in BCNF.

Boyce-Codd Normal Form (BCNF) is an advanced version of the Third Normal Form. A relation is in BCNF if every determinant is a candidate key, ensuring no anomalies related to functional dependencies. For example, a relation where each attribute depends only on a candidate key is considered in BCNF, eliminating redundancy more effectively than 3NF.

9.Why is it necessary to decompose a relation?

Decomposition is necessary to eliminate redundancy and prevent data anomalies (insert, update, delete) in a database. By breaking down a relation, it ensures data integrity, maintains a logical structure, and simplifies queries. Proper decomposition helps optimize storage and minimizes data inconsistency in relational databases.

10.Explain at least two desirable properties of decomposition.

Two desirable properties of decomposition are **Lossless Join** and **Dependency Preservation**. Lossless join ensures that decomposed tables can be joined back together without losing information, maintaining data completeness. Dependency preservation means that functional dependencies are preserved in the new tables, making constraint enforcement simpler.

PART B

11a. Explain in detail about Normal Forms.

First Normal Form (1NF):

- Ensures that each attribute contains atomic (indivisible) values.
- Removes repeating groups, making sure each column contains unique values.
- Example: If a table has multiple phone numbers in one cell, split each into separate rows.

Second Normal Form (2NF):

- Builds on 1NF and ensures that all non-key attributes depend on the entire primary key.
- Applies to tables with composite keys, eliminating partial dependencies.
- Example: In an "Orders" table with {OrderID, ProductID}, remove attributes like "CustomerName" that depend on only one part of the key.

Third Normal Form (3NF):

- Ensures no transitive dependencies, where non-key attributes depend on other non-key attributes.
- All attributes must depend directly on the primary key.
- Example: In a "Students" table, move "Department" details to a separate "Department" table if they depend on "DepartmentID."

Boyce-Codd Normal Form (BCNF):

- A stricter form of 3NF where every determinant is a candidate key.
- Resolves cases where 3NF is insufficient due to dependencies that violate candidate keys.
- Example: If a professor teaches only one subject, split the "Professor" and "Subject" tables if dependencies are unclear.

Fourth Normal Form (4NF):

- Removes multivalued dependencies, ensuring that attributes are independent.
- Deals with situations where one attribute can determine multiple values of another attribute independently.
- Example: If "StudentID" determines both "Course" and "Hobby" independently, separate them into distinct tables.

11b. Explain the various components of ER Diagram with examples briefly.

Entities:

- Objects or concepts that have a distinct existence in the database.
- Represented by rectangles, such as "Employee" or "Department."

Attributes:

- Characteristics or properties of entities.
- Shown as ovals, for example, "EmployeeName" for an "Employee" entity.

Relationships:

- Connections between two or more entities.
- Represented by diamonds, such as a "WorksFor" relationship between "Employee" and "Department."

Primary Key:

- A unique attribute identifying each entity instance.
- Example: "EmployeeID" is the primary key for "Employee."

Cardinality:

- Specifies the number of instances in one entity that can relate to instances in another.
- Example: 1-to-many between "Department" and "Employee."

12a. Explain in detail about Multivalued Dependencies and Fourth Normal Form.

Multivalued Dependency (MVD):

- Occurs when one attribute uniquely determines multiple independent values of another attribute.
- Causes redundancy if not resolved, as one attribute controls multiple values.
- Example: In a "Student" table with "StudentID" determining both "Course" and "Hobby" independently, "StudentID → Course" and "StudentID → Hobby" are MVDs.

Fourth Normal Form (4NF):

- Builds on BCNF and eliminates multivalued dependencies.
- Ensures no attribute has multiple independent values dependent on the same primary key.
- Example: Split "Student" into two tables, one for "StudentID" and "Course" and another for "StudentID" and "Hobby" to eliminate MVDs.

2b. Discuss about integrity constraints in SQL?

Integrity constraints in SQL ensure the accuracy and consistency of data within a relational database. These constraints define rules for permissible values in tables to maintain the correctness and reliability of the data. Here's an overview of common integrity constraints in SQL:

1. Primary Key Constraint

A **Primary Key** uniquely identifies each row in a table. It must contain unique values, and it cannot contain NULL values.

- **Purpose**: Ensure each record is unique and identifiable.
- **Example**

```
CREATE TABLE Employee (
  EmpID INT PRIMARY KEY,
  Name VARCHAR(50),
  DeptID INT
);
```

Here, EmpID is the primary key, meaning each EmpID in the Employee table must be unique and not null.

Foreign Key Constraint

A **Foreign Key** establishes a relationship between two tables by linking a column in one table to the primary key in another table.

- **Purpose**: Maintain referential integrity between tables.

Example

```
CREATE TABLE Department (
  DeptID INT PRIMARY KEY,
  DeptName VARCHAR(50)
);

CREATE TABLE Employee (
  EmpID INT PRIMARY KEY,
  Name VARCHAR(50),
  DeptID INT,
  FOREIGN KEY (DeptID) REFERENCES Department(DeptID)
);
```

In this example, DeptID in the Employee table is a foreign key that references the primary key DeptID in the Department table, ensuring that each employee belongs to a valid department.

Unique Constraint

The **Unique** constraint ensures that all values in a column or a set of columns are distinct across the table. Unlike the primary key, a table can have multiple unique constraints, and unique columns can contain a single NULL value (in most SQL implementations).

- **Purpose**: Prevent duplicate values in specific columns.

Example

```
CREATE TABLE Employee (
    EmpID INT PRIMARY KEY,
    Name VARCHAR(50) UNIQUE,
    DeptID INT
);
```

Not Null Constraint

The **Not Null** constraint ensures that a column cannot contain NULL values. It is often used to ensure required fields are filled in.

- **Purpose**: Ensure essential data is not missing.
- **Example**

```
CREATE TABLE Employee (
    EmpID INT PRIMARY KEY,
    Name VARCHAR(50) NOT NULL,
    DeptID INT NOT NULL
);
```

Register Number:

SASURIE INSTITUTIONS

ॐ

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tiruppur.

UNIT TEST III			Date/Session	11.05.2024	Marks	50
Course code	CS3492	Course Title	Database Management System			
Regulation	2021	Duration	1:30 Hours	Academic Year	2023-2024	
Year	II	Semester	IV	Department	CSE	

COURSE OUTCOMES

CO1:	Construct SQL Queries using relational algebra
CO2:	Design database using ER model and normalize the database
CO3:	Construct queries to handle transaction processing and maintain consistency of the database
CO4:	Compare and contrast various indexing strategies and apply the knowledge to tune the performance of the database
CO5:	Appraise how advanced databases differ from Relational Databases and find a suitable database for the given requirement.

Q.No.	Question	CO	BTS
	PART A (Answer all the Questions 10 x 2 = 20 Marks)		
1	What is transaction.	CO3	R
2	List out ACID property?	CO3	R
3	Why is it necessary to have control of concurrent execution of transaction ? how it is possible ?	CO3	R
4	What are the different modes of lock?	CO3	U
5	Define deadlock.	CO	R
6	List the four conditions for dead lock	CO3	U
7	What is meant by concurrency control .	CO3	R
8	Define two phase locking protocol.	CO3	A
9	What is meant by serialization? How it is tested?	CO3	R
10	Different strict two phase locking and rigourous two phase locking protocol.	CO3	R
	PART B (Answer all the Questions 2 x 15 = 30 Marks)		
11a	Discuss the violation caused by each of the following: lost update problem, dirty read,non repeatable read and phantoms with suitable example .	CO3	E
	OR		
11b	Explain Deadlock in detail with an example.	CO3	E
12a	(i)Write short notes on transition concepts and ACID property (ii) Explain conflict serializability and view serializability.	CO3	A
	OR		
12b	Define functional dependencies. How are primary keys related to FD's?	CO3	C

Course Faculty

HOD

Principal

Register Number: | | | | | | | | | | | |

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tiruppur.

UNIT TEST III			Date/Session	11.05.2024	Marks	50
Course code	CS3492	Course Title	Database Management System			
Regulation	2021	Duration	1:30 Hours	Academic Year	2023-2024	
Year	II	Semester	IV	Department	CSE	

PART A

1. What is transaction.

A transaction in the context of databases is a sequence of operations performed as a single logical unit of work. Transactions are fundamental to maintaining data consistency and integrity within a database, especially in multi-user and concurrent environments.

2. List out ACID property?

- Atomicity
- Consistency
- Isolation
- Durability

3. Why is it necessary to have control of concurrent execution of transaction ? how it is possible ?

Controlling the concurrent execution of transactions is necessary to maintain data integrity, consistency, and isolation in a database, especially when multiple users or applications access and modify data simultaneously.

4. What are the different modes of lock?

- Shared Lock (S-Lock)
- Exclusive Lock (X-Lock)

5.Define deadlock.

Deadlock is a situation in a database where two or more transactions are waiting for each other's locked resources, causing each transaction to be indefinitely blocked and unable to proceed.

6.List the four conditions for deadlock.

The four conditions for deadlock are mutual exclusion, hold and wait, no preemption, and circular wait. All must be present for a deadlock to occur, where each process holds resources needed by the others.

7.What is meant by concurrency control?

Concurrency control is the management of concurrent transaction execution to ensure data integrity and consistency. It uses techniques like locking and timestamping to handle multiple transactions simultaneously.

8.Define two-phase locking protocol.

The two-phase locking protocol is a concurrency control method with two phases: the growing phase (acquiring all locks) and the shrinking phase (releasing locks). It ensures conflict-serializability and prevents data conflicts.

9.What is meant by serialization? How is it tested?

Serialization is the process of ensuring that transactions execute in a sequence that produces the same result as some sequential order. Testing involves conflict-serializability and view-serializability to validate correctness.

10.Differentiate strict two-phase locking and rigorous two-phase locking protocol.

In strict two-phase locking, locks are released only after a transaction completes. In rigorous two-phase locking, all locks are held until the transaction commits or aborts, providing stricter control and preventing cascading rollbacks.

PART B

11a. Discuss the violation caused by each of the following: lost update problem, dirty read, non-repeatable read, and phantoms with suitable examples.

Lost Update Problem:

- Occurs when two transactions read and update the same data, but one update is overwritten by the other.
- Example: Two users modify the same bank balance, and one update is lost due to concurrent access.

Dirty Read:

- Happens when a transaction reads uncommitted data from another transaction.
- Example: Transaction A reads data updated by Transaction B, but B later rolls back, leading to invalid data in A.

Non-Repeatable Read:

- Occurs when a transaction reads the same data twice and gets different values due to another transaction's updates in between.
- Example: Transaction A reads a value, then Transaction B modifies it, causing A's second read to differ from the first.

Phantoms:

- Arises when a transaction re-executes a query and finds additional rows due to another transaction's insertions.
- Example: Transaction A retrieves a set of rows based on a condition, while Transaction B inserts new rows matching the condition.

11b. Explain Deadlock in detail with an example.

Deadlock Definition:

- A deadlock is a situation where two or more transactions are waiting indefinitely for resources locked by each other, causing a standstill.

Conditions for Deadlock:

- ***Mutual Exclusion***: Only one transaction can hold a resource at a time.
- ***Hold and Wait***: Transactions hold resources while waiting for others.
- ***No Preemption***: Resources cannot be forcibly taken from transactions.

- **Circular Wait**: A circular chain of transactions exists, each waiting for the next.

Example:

- Transaction A locks Resource X and waits for Resource Y, while Transaction B locks Resource Y and waits for Resource X, causing both to be blocked.

12a.i) Write short notes on transaction concepts and ACID properties.

Transaction Concepts:

- Transactions are a unit of work in a database, treated as a single, indivisible process.
- Ensures data consistency and integrity through sequential and complete processing.
- Supports operations like Commit (saving changes) and Rollback (reverting changes).

ACID Properties:

- **Atomicity**: Ensures all parts of a transaction are completed, or none are.
- **Consistency**: Maintains data integrity before and after a transaction.
- **Isolation**: Ensures transactions execute independently without interference.
- **Durability**: Guarantees that committed changes are permanent, even in case of failures.

12a.ii) Explain conflict serialization and view serialization.

Conflict Serialization:

- Refers to a schedule where transactions execute in a conflict-free order, resulting in a serial order.
- Ensures that conflicting operations (e.g., read/write on the same data) are scheduled to prevent data inconsistencies.
- Conflict-serializable schedules are tested using techniques like precedence graphs.

View Serialization:

- Ensures that a schedule has the same final output as some serial execution of transactions.
- View-equivalent schedules may have different orders but lead to the same database state.
- Ensures data consistency without requiring transactions to execute in strict order.

12b) Define functional dependencies. How are primary keys related to FD's?

In relational database design, a **functional dependency** (FD) is a relationship between two attributes or sets of attributes within a relation (table). A functional dependency describes how one attribute's value is determined by another attribute or combination of attributes.

Notation: $X \rightarrow Y$, meaning "X determines Y" or "Y is functionally dependent on X."

Types of Functional Dependencies

- Trivial Functional Dependency
- Non-Trivial Functional Dependency
- Full Functional Dependency

- Partial Functional Dependency

Primary Keys and Functional Dependencies

A **primary key** in a relation is an attribute (or a set of attributes) that uniquely identifies each row in the table. Primary keys are closely related to functional dependencies:

- **Primary Key as Determinant**: In a table, the primary key should determine all other attributes, meaning the primary key has a full functional dependency on every other attribute in the table.
- **Minimal Superkey**: A primary key is the minimal superkey. A superkey is any set of attributes that can uniquely identify a row. The primary key is the smallest subset of a superkey that maintains this unique identification.
- **No Partial Dependencies**: In a well-normalized table (at least in 2NF), there should be no partial dependencies on the primary key. This means that no attribute should be functionally dependent on only part of a composite primary key.

Register Number:

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tiruppur.

UNIT TEST IV			Date/Session	04.07.2024	Marks	50
Course code	CS3492	Course Title	Database Management System			
Regulation	2021	Duration	1:30 Hours	Academic Year	2023-2024	
Year	II	Semester	IV	Department	CSE	

COURSE OUTCOMES

CO1:	Construct SQL Queries using relational algebra
CO2:	Design database using ER model and normalize the database
CO3:	Construct queries to handle transaction processing and maintain consistency of the database
CO4:	Compare and contrast various indexing strategies and apply the knowledge to tune the performance of the database
CO5:	Appraise how advanced databases differ from Relational Databases and find a suitable database for the given requirement.

Q.No.	Question	CO	BTS
	PART A (Answer all the Questions 10 x 2 = 20 Marks)		
1	What is B-Tree?	CO4	R
2	Define rotational latency time?	CO4	R
3	What is an index?	CO4	U
4	What are the types of storage devices?	CO4	U
5	What is hashing file organization.	CO4	R
6	Differentiate static and dynamic hashing	CO4	A
7	What is called query processing?	CO4	R
8	Explain "Query optimization"?	CO4	A
9	Define Primary and Secondary Indices?	CO4	R
10	What are the Types of Ordered Indices.	CO4	R
	PART B (Answer all the Questions 2 x 15 = 30 Marks)		
11a	Define RAID and Briefly Explain RAID	CO4	E
	OR		
11b	Discuss two phase locking protocol and strict two phase locking protocols?	CO4	E
12a	Explain about Query optimization with neat diagram	CO4	C
	OR		
12b	Give detail Explanation about Hashing & Types of Hashing.	CO4	E

Course Faculty

HOD

Principal

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tirupur.

UNIT TEST IV			Date/Session	24.05.2024	Marks	50
Course code	CS3492	Course Title	Database Management System			
Regulation	2021	Duration	1:30 Hours	Academic Year	2023-2024	
Year	II	Semester	IV	Department	CSE	

PART A

1. What is B-Tree?

A B-Tree (Balanced Tree) is a self-balancing tree data structure that maintains sorted data and allows for efficient insertion, deletion, and search operations. B-Trees are widely used in databases and file systems because they can handle large volumes of data and are optimized for systems that read and write large blocks of data (e.g., disks or other secondary storage).

2. Define rotational latency time?

Rotational Latency Time (also known as rotational delay) is the time taken for the desired sector of a spinning hard disk or other rotating storage media to rotate under the read/write head so that data can be accessed. This delay is one of the key components that impact the overall time to access data on a traditional hard drive.

3. What is an index?

An index in the context of databases is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional space and maintenance overhead. Indexes are used to quickly locate rows in a table based on the values of one or more columns, improving the performance of SELECT queries.

4.What are the types of storage devices?

Storage devices include primary storage (RAM), secondary storage (HDDs, SSDs), and tertiary storage (tape drives). They vary in speed, cost, and durability, serving different data storage needs.

5.What is hashing file organization?

Hashing file organization uses hash functions to assign data to specific locations in storage, enabling fast retrieval. It's efficient for direct access but may require collision handling mechanisms.

6.Differentiate static and dynamic hashing.

Static hashing uses a fixed number of storage buckets, which may cause overflow as data grows. Dynamic hashing adjusts the number of buckets based on data volume, allowing for flexible storage expansion.

7.What is called query processing?

Query processing is the sequence of steps the database system uses to interpret and execute SQL queries. It includes parsing, optimization, and execution to retrieve results efficiently.

8.Explain "Query optimization."

Query optimization is the process of selecting the most efficient execution plan for a query, reducing response time and resource usage. Optimizers consider factors like indexes, join methods, and execution costs.

9.Define Primary and Secondary Indices.

A primary index is based on a table's primary key, ensuring unique and ordered entries. Secondary indices are additional indexes on non-key columns, used to improve query performance on frequently searche fields.

10.What are the Types of Ordered Indices?

Ordered indices include dense and sparse indices. Dense indices have entries for every record, while sparse indices only index certain records, reducing storage space at the cost of slightly slower lookups

PART B

11a. Define RAID and Briefly Explain RAID

Definition of RAID:

- RAID (Redundant Array of Independent Disks) is a data storage technology that combines multiple physical drives into one unit to improve performance and reliability.

Types of RAID:

- **RAID 0**: Data is striped across multiple disks, increasing speed but without redundancy.
- **RAID 1**: Mirrors data across disks, providing redundancy but no speed improvement.
- **RAID 5**: Stripes data with parity, balancing speed and redundancy; requires at least three disks.
- **RAID 6**: Similar to RAID 5 but with extra parity for increased fault tolerance; requires at least four disks.

11b. Discuss two phase locking protocol and strict two phase locking protocols?

The Two-Phase Locking Protocol (2PL) is a concurrency control protocol used in database systems to ensure serializability of transactions (i.e., the execution of transactions is equivalent to some serial execution of those transactions). The protocol ensures that the database system remains consistent and avoids issues like lost updates, temporary inconsistency, or deadlocks during concurrent transaction processing.

The **Two-Phase Locking** protocol operates in two distinct phases:

1. **Growing Phase**:
   - A transaction can acquire locks but cannot release any locks.
   - The transaction can acquire any number of locks, whether shared or exclusive, as long as it does not release any locks.
2. **Shrinking Phase**:
   - Once a transaction releases its first lock, it enters the shrinking phase.
   - During this phase, the transaction can only release locks and cannot acquire any more locks.

roperties of Two-Phase Locking:

- **Serializable**: The protocol guarantees serializability, ensuring that the execution of transactions is equivalent to some serial execution, which preserves the database's consistency.
- **Deadlocks**: Two-phase locking does not inherently prevent deadlocks. Deadlock detection and resolution techniques are often used to manage this.
- **Concurrency**: 2PL ensures correct transaction execution at the cost of some concurrency because transactions cannot hold locks indefinitely (they must eventually release them).

12a. Explain about Query Optimization with Neat Diagram

Definition:

- Query optimization is the process of choosing the most efficient way to execute a SQL query to reduce response time and resource usage.

Steps in Query Optimization:

- **Parsing:** Analyzes the syntax and checks for errors in the query.
- **Plan Generation:** Creates various execution plans to retrieve the data.
- **Cost Estimation:** Evaluates the cost of each execution plan based on factors like I/O and CPU usage.
- **Plan Selection:** Chooses the lowest-cost execution plan for query execution.

12b. Give Detailed Explanation about Hashing & Types of Hashing

Hashing:

- A file organization technique that provides fast data retrieval by converting keys into hash codes, which map to data storage locations.

Types of Hashing:

- **Static Hashing:** Uses a fixed number of buckets; collisions are handled using overflow buckets, but performance may degrade over time.
- **Dynamic Hashing:** Adjusts the number of buckets dynamically based on data size; helps in handling growing data sets efficiently.
- **Extendible Hashing:** Uses a directory that grows with data, providing a flexible structure to minimize overflow.
- **Linear Hashing:** Gradually adds buckets as data grows, balancing between storage utilization and access speed.

Register Number:

ॐ

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tiruppur.

MODEL EXAM I			Date/Session	01.06.2024	Marks	100
Course code	CS3492	Course Title	Database Management System			
Regulation	2021	Duration	3:00 Hours	Academic Year	2023-2024	
Year	II	Semester	IV	Department	CSE	

COURSE OUTCOMES

CO1:	Construct SQL Queries using relational algebra
CO2:	Design database using ER model and normalize the database
CO3:	Construct queries to handle transaction processing and maintain consistency of the database
CO4:	Compare and contrast various indexing strategies and apply the knowledge to tune the performance of the database
CO5:	Appraise how advanced databases differ from Relational Databases and find a suitable database for the given requirement.

Q.No.	Question	CO	BTS
	PART A		
	(Answer all the Questions 10 x 2 = 20 Marks)		
1	Define data abstraction.	CO1	R
2	What is full outer join.	CO1	R
3	Short note on second normal form.	CO2	A
4	List the feature of EER model.	CO2	A
5	Define deadlock.	CO3	R
6	List the four condition for dead lock.	CO3	U
7	What are the states of transaction?	CO6	R
8	Define two phase locking protocol.	CO6	A
9	When is a transaction rolled back.	CO4	R
10	Define upgrade and downgrade.	CO5	R
	PART B		
	(Answer all the Questions 5 x 13 = 65 Marks)		
11a	Explain DBMS system architecture	CO1	E
	OR		
11b	Explain about relational model constraints	CO1	E
12a	Draw E-R diagram for hospital managment.	CO2	C
	OR		
12b	Explain briefly about normalization	CO2	E
13a	Explain in detail about ACID property.	CO3	E
	OR		

13b	Explain log based recovery in detail i) Immediate database modification ii) Deferred modification	**CO3**	**C**
14a	Explain all types of data models.	CO4	**A**
	OR		
14b	Explain various DML command with neat syntax.	**CO4**	**E**
15a	What is concurrency control? How is it implementation in DBMS?	**CO5**	**A**
	OR		
15b	Explain E-R model concept and extended E-R model	**CO5**	**C**
	PART C **(Answer all the Questions 1 x 15 = 15 Marks)**		
16a	Consider the employee database, where the primary keys are underlined Employee (empname, street, city) Works(empname, companyname, salary) Company(companyname, city) Manages(empname, management) Give an expression in the relational algebra for each request 1. Find the names of all employees who work for first bank corporation 2. Find the names, street address and cities of residence of all employees who work for first bank corporation and earn more than 200000 per annum 3. Find the names of all employees in this database who lives in the same city as the company for which they work. 4. Find the name of all employees who earn more then every employees of smallbank corporation	CO6	C
	OR		
16b	Describe briefly about embedded SQL	**CO6**	**C**

Course Faculty

HOD

Principal

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Vijayamangalam, Tiruppur

MODEL EXAM I			Date/Session	01.06.2024	Marks	100
Course code	CS3492	Course Title	Database Management System			
Regulation	2021	Duration	3:00 Hours	Academic Year	2023-2024	
Year	II	Semester	IV	Department	CSE	

PART A

1.Define Data Abstraction.

Data abstraction in DBMS hides lower-level details to simplify data access for users. It provides three levels: physical (storage details), logical (structure and relationships), and view (user-specific perspectives). This helps in efficient data management.

2.What is Full Outer Join?

A full outer join returns all rows from both tables, with NULLs in columns where there is no match. It ensures that unmatched rows from both sides are included in the result.

3.Short note on Second Normal Form.

A table is in Second Normal Form (2NF) if it is in First Normal Form (1NF) and all non-key attributes depend fully on the primary key. This eliminates partial dependency and ensures better data organization.

4.List the features of the EER Model.

The Extended Entity-Relationship (EER) model enhances ER models by adding subclasses, superclasses, specialization, and generalization, helping to create more complex and detailed database designs.

5.Define Deadlock.

A deadlock is a situation where two or more transactions are waiting indefinitely for each other to release resources. This results in a standstill, where no transaction can proceed.**List the four conditions for deadlock.** Deadlock occurs under four conditions: mutual exclusion (one resource per transaction), hold and wait (holding resources while waiting for others), no preemption, and circular wait.

6.List the four conditions for deadlock.

Deadlock occurs under four conditions: mutual exclusion (one resource per transaction), hold and wait (holding resources while waiting for others), no preemption, and circular wait.

7.What are the states of a transaction?

A transaction has multiple states: active (executing), partially committed, committed (successfully completed), failed, and aborted (rolled back due to failure).

8.Define Two-Phase Locking Protocol.

The two-phase locking protocol ensures transaction serializability by dividing the process into two phases: a growing phase (acquires locks) and a shrinking phase (releases locks). This helps prevent conflicts.

9.When is a transaction rolled back?

A transaction is rolled back when it cannot complete successfully, such as during errors or deadlocks. Rolling back restores the database to a consistent state.

10.Define Upgrade and Downgrade.

In DBMS locking, upgrading changes a read lock to a write lock for higher access, while downgrading reduces a write lock to a read lock, adjusting access levels based on transaction needs.

PART B

11a. Explain DBMS System Architecture

- DBMS architecture includes three core levels: internal, conceptual, and external, which work together for effective data management.
- **Internal level**: Manages physical storage details, including data compression, indexing, and data structure. It deals with efficient storage and retrieval mechanisms.
- **Conceptual level**: Provides an abstract structure for the entire database, independent of storage and focused on logical organization.
- **External level**: Presents specific views to users, showing only the relevant information, enhancing user experience and data security.
- This layered structure supports data independence, efficient access, and data abstraction in complex systems.

11b. Explain Relational Model Constraints

- Relational model constraints maintain database integrity and accuracy, enforcing rules on data.
- **Key Constraints**: Primary and foreign keys ensure unique identification of records and maintain relationships between tables.
- **Domain Constraints**: Restrict data types for each attribute to ensure correct data entry, like integer-only for age fields.
- **Referential Integrity**: Maintains consistent relationships by ensuring foreign key values exist in referenced primary keys.
- **Entity Integrity**: Requires primary keys to be unique and not null, preventing duplicates and ensuring every record is identifiable.

12a. Draw E-R Diagram for Hospital Management

- The E-R diagram for a hospital system includes entities like **Patient**, **Doctor**, **Nurse**, **Appointment**, **Treatment**, and **Billing**.
- **Attributes**: Each entity has unique attributes, such as Patient (ID, name, address), Doctor (ID, specialization), etc., to capture detailed information.
- **Relationships**: Define associations, such as Patient and Doctor through Appointment, or Doctor with Treatment, to reflect interactions within the hospital.

- **Cardinality**: Specifies one-to-many or many-to-many relationships, like a Patient having multiple Appointments, helping manage hospital operations effectively.

12b. Explain Normalization Briefly

- **Normalization** is the process of organizing data to reduce redundancy and enhance integrity by structuring tables and dependencies.
- **1NF (First Normal Form)**: Ensures atomic values in each field and removes repeating groups or duplicate columns.
- **2NF (Second Normal Form)**: Builds on 1NF by eliminating partial dependencies, ensuring non-key attributes depend fully on the primary key.
- **3NF (Third Normal Form)**: Removes transitive dependencies, meaning non-key attributes are independent of each other, leading to better data integrity.
- **BCNF (Boyce-Codd Normal Form)**: A higher level where every determinant must be a candidate key, further refining the database structure.

13a. Explain ACID Properties in Detail

- **ACID properties** are critical for reliable transaction processing in databases.
- **Atomicity**: Ensures a transaction is completed fully or not at all, with rollbacks on failure to maintain data consistency.
- **Consistency**: Guarantees that database rules are enforced at all stages, so the database transitions from one valid state to another.
- **Isolation**: Prevents transactions from affecting each other's execution, which is essential for accurate concurrent transactions.
- **Durability**: Ensures that once a transaction is committed, it is permanently recorded, even if the system crashes.

13b. Explain Log-Based Recovery in Detail

(i) Immediate database modification

Immediate Database Modification

Immediate Database Modification is a recovery technique where changes made by a transaction to the database are written immediately to the database **before** the transaction commits. These changes are then recorded in the log, and the log entries are used to determine the recovery actions in the event of a failure.

Key Concepts in Immediate Database Modification:

1. **Write-Ahead Logging (WAL)**: In this approach, the log is written to disk **before** the actual data is written to the database. This ensures that if a crash occurs, the system can use the log to recover and roll

back any uncommitted changes. Specifically, any changes to the database must be preceded by a log entry.

2. **Transaction Commit**: When a transaction commits, the database has already been updated (because changes are written immediately). The system writes a **commit record** to the log to indicate that the transaction has completed successfully. This record tells the recovery system that the transaction should be considered permanent.
3. **Crash Recovery**: After a system crash, the recovery process will use the log to determine which transactions were committed, which were rolled back, and which ones were in-progress at the time of the crash. It uses the following steps:
    - **Redo**: Reapplies the changes from committed transactions to ensure that all committed updates are reflected in the database.
    - **Undo**: Reverts the changes made by transactions that did not commit before the crash.

ii) Deferred modification

Deferred Database Modification is a log-based recovery technique where changes made by a transaction are not written to the database until the transaction has committed. Instead, the transaction's operations are logged in the transaction log, but the actual changes are deferred (i.e., not applied to the database) until the transaction is committed.

This method relies on write-ahead logging and ensures that only committed transactions can modify the database. If a system crash occurs before a transaction commits, the changes made by that transaction are discarded because they have not yet been written to the database.

14a. Explain All Types of Data Models

- Various data models structure databases according to different needs and relationships:
    - **Hierarchical Model**: Organizes data in a tree format with parent-child relationships, suitable for hierarchical data.
    - **Network Model**: Extends hierarchical relationships to many-to-many, forming a graph structure for complex associations.
    - **Relational Model**: Uses tables (relations) to represent data, focusing on relationships between tables through keys.
    - **Object-Oriented Model**: Stores data as objects, like in programming, allowing for complex data types and relationships.
    - **ER Model**: Visualizes data with entities and relationships, often used as a design framework before building the database.

14b. Explain Various DML Commands with Syntax

- **Data Manipulation Language (DML)** commands allow manipulation of data within tables.
    - **SELECT**: Retrieves specific data from tables, with options for filtering and joining (SELECT * FROM table_name WHERE condition).

- **INSERT**: Adds new records to a table with specified values (INSERT INTO table_name VALUES(...)).
- **UPDATE**: Modifies existing records, typically with a condition to target specific rows (UPDATE table_name SET column=value WHERE condition).
- **DELETE**: Removes records from a table based on specified conditions (DELETE FROM table_name WHERE condition).
- **MERGE**: Combines INSERT and UPDATE operations based on matching conditions, providing a flexible data modification option.

15a. What is Concurrency Control? How is it Implemented in DBMS?

- **Concurrency control** manages the simultaneous execution of multiple transactions, preserving database consistency.
- **Locking Mechanisms**: Use shared and exclusive locks to prevent conflicting access to the same data by multiple transactions.
- **Timestamp Ordering**: Assigns timestamps to transactions, enforcing an order to avoid conflicts in execution.
- **Two-Phase Locking Protocol**: Divides transactions into growing (acquiring locks) and shrinking (releasing locks) phases to avoid deadlocks.
- **Optimistic Concurrency Control**: Assumes minimal conflict, checking for issues only at commit time to improve transaction speed.

15b. Explain E-R Model Concept and Extended E-R Model

- The **E-R (Entity-Relationship) Model** represents data with entities, attributes, and relationships, providing a framework for data organization.
    - **Entities**: Represent real-world objects, like Student, Course, or Instructor, with unique attributes such as name, ID, or course title.
    - **Attributes**: Define properties of entities, such as name, age, or address, to capture detailed information about each entity.
    - **Relationships**: Represent associations, like a Student "enrolls in" a Course, specifying how entities interact with each other.
- **Extended E-R (EER) Model**: Adds complexity with subclasses, superclasses, specialization, and generalization, allowing for inheritance and hierarchy in data relationships.

PART C

16a. Consider the employee database and give relational algebra expressions for each request

- **Employee Database Structure:**
  1. Tables include:
     - **Employee**: (empname, street, city)
     - **Works**: (empname, companyname, salary)
     - **Company**: (companyname, city)
     - **Manages**: (empname, management)
- **Relational Algebra Expressions:**

1. **Find names of all employees working for First Bank Corporation:**
   - Use **selection** and **projection** to identify employees associated with "First Bank Corporation."
   - Expression: $\pi_{empname}(\sigma_{companyname='First\ Bank\ Corporation'}(Works))$
2. **Find names, street address, and cities of employees at First Bank Corporation with salary > 200,000:**
   - Use **selection** and **join** on Works and Employee to filter for company and salary conditions.
3. Expression: $\pi_{empname, street, city}(\sigma_{companyname='First\ Bank\ Corporation' \wedge salary > 200000}$
4. **Find names of employees living in the same city as their company:**
   - Use **selection** and **join** on Employee and Company to match city fields.
   - Expression: $\pi_{empname}(\sigma_{Employee.city = Company.city}$
5. **Find names of employees earning more than every employee at SmallBank Corporation:**
   - Use **division** or **nested query** approach to compare salaries across companies.
   - Expression: $\pi_{empname}(Works) - \pi_{empname}(\sigma_{salary < ALL(\pi_{salary}(\sigma_{companyname = 'SmallBank\ Corporation'}(Works))))}$

16b. Describe Embedded SQL Briefly

- **Embedded SQL:**
  - **Definition**: Embedded SQL is SQL code embedded within a programming language like C, Java, or Python. It allows applications to interact with the database directly within the code, simplifying database operations.
- **Components of Embedded SQL:**
  - **SQL Statements**: Standard SQL statements, like SELECT, INSERT, UPDATE, and DELETE, are used to query or modify data directly within application code.
  - **Host Variables**: Variables in the host programming language can be used to pass values into SQL queries, making them dynamic and adaptable.
  - **SQL Control Statements**: Use constructs like cursors and error-handling statements to manage database interactions efficiently.
- **Process:**
  - **Pre-compilation**: The embedded SQL code is pre-compiled to ensure compatibility with the database, creating efficient access paths.
  - **Execution**: When the application runs, SQL statements are executed as part of the program, allowing real-time data processing within the application.
- **Advantages:**

- **Efficiency**: Reduces the overhead of separate database calls by embedding SQL, improving performance.
- **Data Integrity**: Enforces data checks directly within the application, helping to maintain data consistency.
- **Seamless Integration**: Allows developers to write SQL queries in the application's programming language, bridging database and application code efficiently.

SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

Answer Book

Name	V. Haripriya			Year/ Semester/Section	II / IV
Register Number	732422104018	Date/Session	27.3.24	Department	CSE
Course code	CS3492	Course Title	Database Management System		
Internal Assessment Test	IAT 1 ☑	IAT 2 ☐	IAT 3 ☐	Model ☐	
Name and Signature of the Invigilator with date					

Instruction to the Student: Put tick mark to the question attended in the column against question.

Part A			Part B/ Part C					Total Marks
Q. No.	✓	Marks	Q. NO.	✓	a Marks	✓	b Marks	
1	✓	2	11	✓	13			13
2	✓	2	12	✓	13			13
3	✓	2	13					
4	✓	2	14					
5	✓	2	15					
6	✓	2	16					
7	✓	2					Grand Total	26
8	✓	2						
9	✓	2	46/50					
10	✓	2						
Total		20	Grand Total				Name and Signature of the Examiner with date	

To be filled by the examiner

Course Outcomes	1	2	3	4	5	6	Total
Marks allotted	50	-	-	-	-	-	50
Marks Obtained	46	-	-	-	-	-	46

IQAC Audit - Remarks

Verified

Name and Signature of the IQAC member

27/3/24

TNPL
DATE / /

Unit - Test - I

Reg NO : 732422104018
Name : V. Hasipriya
Dept : BE. CSE
Subject : Data Base Management System
Subcode : CS3492
Date : 27.03.2024

16
50

PART - A

1. Data abstraction:

*. Data abstraction is defines as only provide relevant data / information for the system without background details.

Data abstraction level:

*. Physical level
*. Logical level
*. View level

---

2. DataBase Management system:

*. DataBase Management System is a collection of interrelated data and various program used to handle the data
*. DBMS provides a way to retrieve and store data from system in convenient and efficient manner.

---

5. Categories of SQL commands:

*. Data Definition Language (DDL)
*. Data Manipulation Language (DML)
*. Data control language (DCL)

03.

Static SQL	Dynamic SQL
* The SQL statements are compiled at compiler time	* The SQL statements are compiled at run time
* It is less efficient	* It is more efficient
* It is more flexible	* It is less flexible
* Used in situation data are distributed uniformly	* Used in situation of data is distributed non uniformly

04.

Primary key	Foreign key
* Primary key is a column or set of column used to identify a rows uniquely	* Foreign key is a column or set of column used to refer the primary key (or) candidate key for other table
* The table have only one primary key	* The table can have multiple Foreign key

6. Data Definition language:

* Data Definition language is a special language used to simplify and facilitates to access the data.
* It is a language which is provide creating and modifying the table, view, indexs so on.

Common command for DDL are

* CREATE
* ALTER
* DROP

---

7. NULL value:

* In Database NULL value provides special value to Database. They are in two cases.

(i) When field name of some tuples are unknown (i.e) city name is not given
(ii) Inapplicable (i.e) middle name is not present.

---

8. Key:

* key is used to specify the tuples distinctly in the given relation.

The various types of keys are

* super key
* Foreign key
* Primary key
* Candidate key.

---

9. Instance:

* The declared variable have values so the value assigned for the variable is called instance.

Ex: int a=10;

Schema:

* The declaration of variable is called schema.

Ex: int a;

10. DataBase system	Conventional File system
* Data reductancy is less	* Data reductancy is more
* security is high	* Security is low
* Database system are used in high security constraints.	* Conventional file system used in less security constraint
* There is ability to access concurrently data	* There is no ability to access concurrently the data

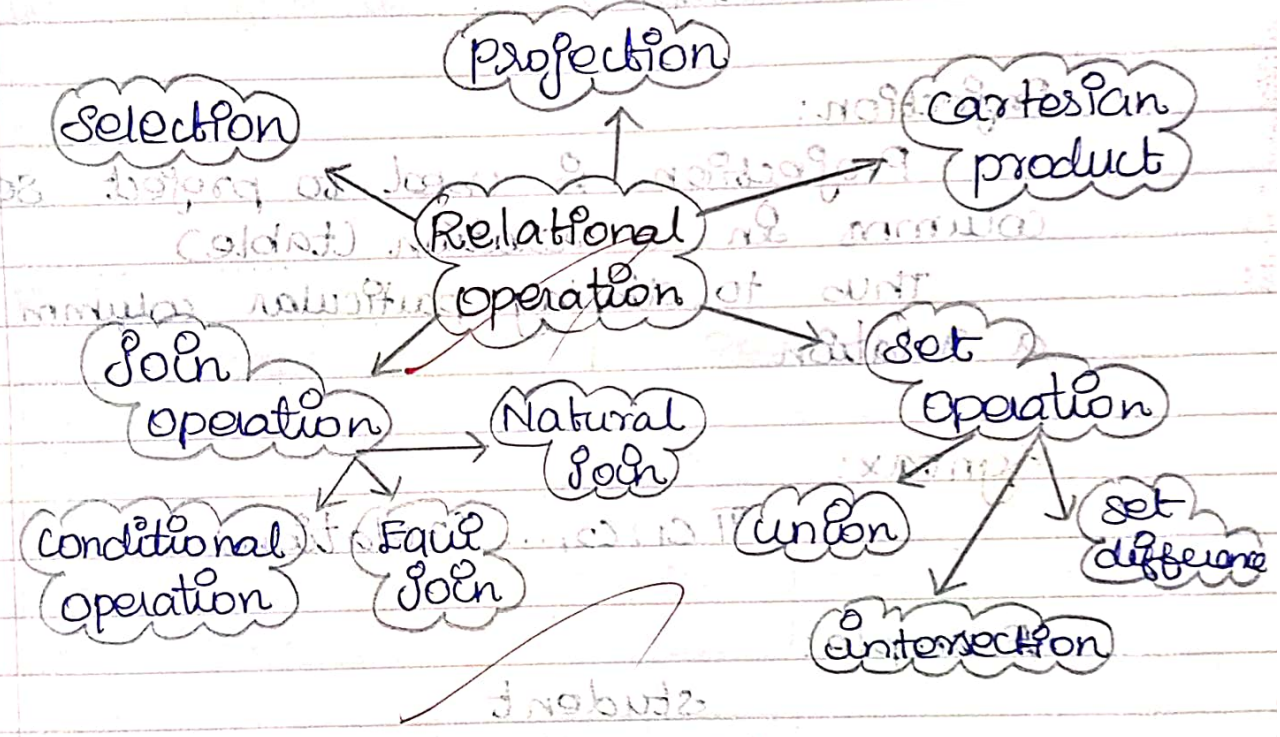
27/3/24.

PART-B.

18. **Relational Algebra:**

Relational algebra is a query language used to specify the relation to read the table in different ways.

Relational operation:



Selection:

Selection operation is used to fetch tuples in a relation (table)

Syntax:

$\sigma_{predicate}(relation)$

Example:

STD	SName	Age
001	AAAA	18
002	BBBB	23
003	CCCC	16
004	DDDD	19

Student (relation)

Query: To fetch the students have more than age 18.

$\sigma_{age \geq 18}(student)$

Output:

sName
AAAA
BBBB
DDDD

Projection:

Projection is used to project some columns in a relation (table)

Thus to display particular column in a relation.

Syntax:

$\Pi_{C1, C2, \ldots}(relation)$

Example:

student

SID	SNAME	AGE
01	AAA	18
02	BBB	23
03	CCC	19
04	DDD	28

Query: We want to display the columns of student name and age only.

$\Pi_{(SNAME, AGE)}(student)$

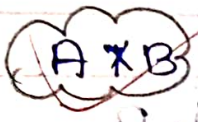
Output:

SNAME	AGE
AAA	18
BBB	19
CCC	20
DDD	21

Cartesian Product:

Cartesian product is used to combine two different relations (tables).

Syntax:



Example:

student

SID	SNAME	AGE
01	AAA	18
02	BBB	19
03	CCC	20
04	DDD	21

Reserve

SID	ISBN	DAY
01	005	8/8/19
02	005	7/8/19
03	006	9/8/19
04	007	10/8/20

Query:

We want to combine who are reserve ISBN = 005

σ (Student.Id = Reserve.SID) $\wedge$ (Reserve.isbn = 005) (Student × Reserve)

Output:

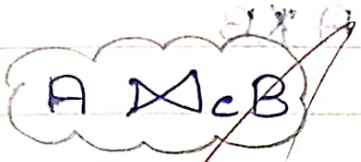
SID	Name	AGE	ISBN	DAY
01	AAA	18	005	8/8/19
02	BBB	19	005	7/8/19

Join operation:

Join operation is used to combine two or more relation (tables) then the combined relation we crave to fetch data.

(1) Conditional join: This operation is used to combine two relations (tables) under some conditions.

Syntax:



Example:

Student

SID	name	age
01	AAA	18
02	BBB	19
03	CCC	20
04	DDD	21

Reserve

SID	Isbn	Day
01	005	9/9/20
02	005	9/10/11
03	006	10/11/19
04	007	13/4/20

Query: we want to combine the student and Reave relation under the condition of Isbn = 005.

Student ⋈c student.SID = Reserve.SID) ^ (Reserve.Isbn = 005) (Reserve)

27/3/24

Output:

SID	NAme	age	Isbn	Day
01	AAA	18	005	8/4/10
02	BBB	19	005	9/10/11

(ii) Equijoin:

*. It is a kind of join equality condition between relation (table) we equalate the relation is a equijoin.

(iii) Natural join:

*. There are common column in a relation when we want to combine the common column by using natural join

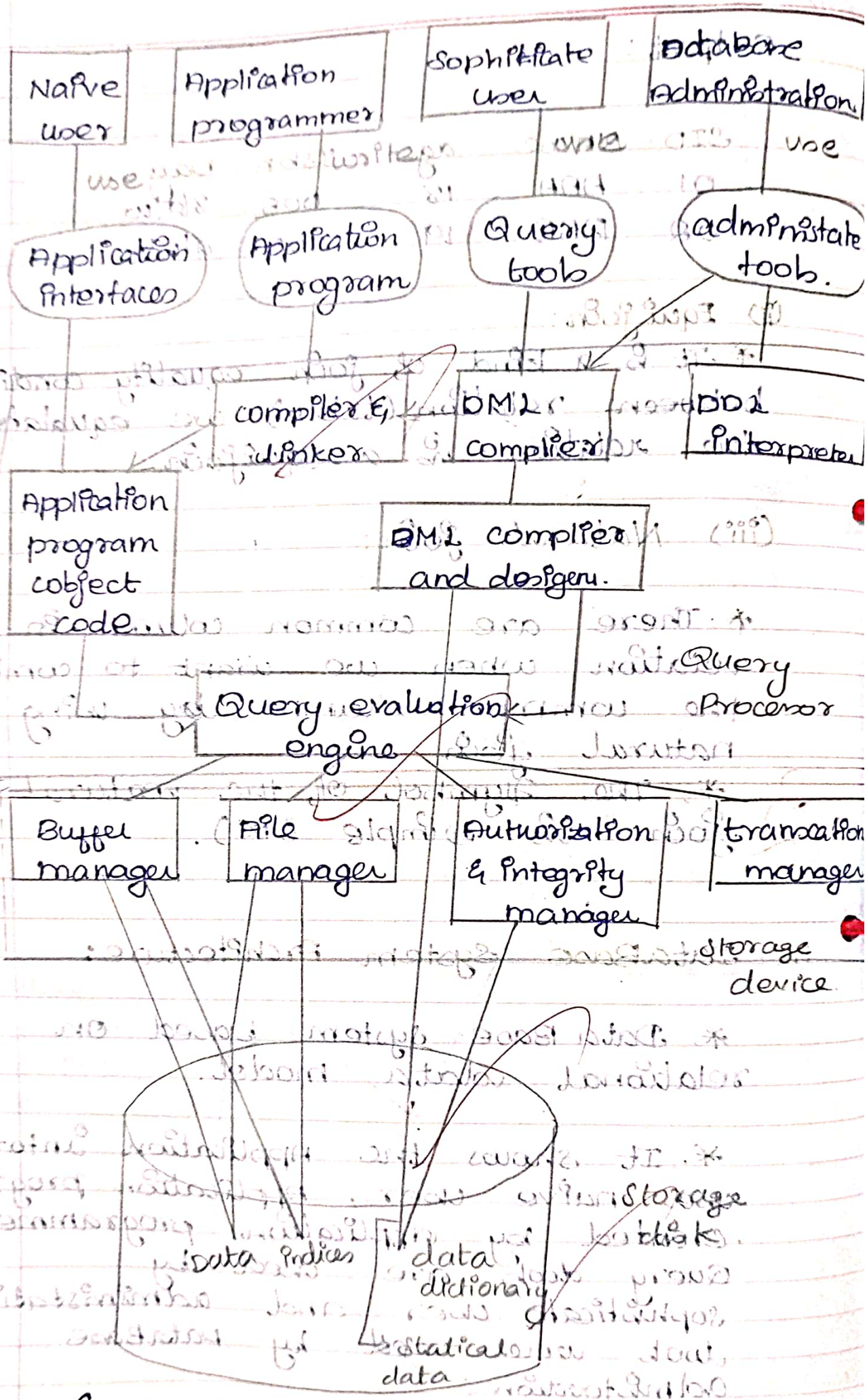
*. The symbol of the natural join is simple (⋈).

---

11. DataBase System Architecture:

*. Data Base system based on relational data model.

*. It shows the Application interface user naive user, Application program created by application programmer, Query tools are used by sophisticate user and administration tools were used by DataBase Administration.



(Data Base System Architecture)

*. The lower level architecture is storage disk.

*. There two main component of database system
    *. Query processor
    *. storage device.

Query processor:

* The query processor interactive one used to specify the simple and facilitate to access the data.
*. It includes a DDL interpreter, DML compiler and query evalution engine.

DDL interpreter:
*. It is a translator in which it transmits query in data dictionary

DML compiler:
*. It is a translator which transmits the data into query engine plan.

Query evalution engine:
*. It executes the low level instructions from the DML compiler.

Storage device:

* storage device is component component in data base It provides a memory store and retrieve data easy.

* storage device are easy to create, retaive, update the data.

It includes the
* Authorization integrity
* Buffer manage
* File manager
* Trancation manger
* Data
* Data indices
* Data dictionary.

SASURIE
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

SASURIE INSTITUTIONS

Answer Book

Name	P. Revathi			Year/ Semester/Section	II / IV
Register Number	732422104036	Date/Session	17.4.24	Department	CSE
Course code	CS3492	Course Title	Database Management System		
Internal Assessment Test	IAT 1 ☐	IAT 2 ☑	IAT 3 ☐	Model ☐	
Name and Signature of the Invigilator with date					

Instruction to the Student: Put tick mark to the question attended in the column against question.

Part A			Part B/ Part C					Total Marks
Q. No.	✓	Marks	Q. NO.	✓	a Marks	✓	b Marks	
1	✓	2	11	✓	13			13
2	✓	2	12	✓	13			13
3	✓	2	13					
4	✓	2	14					
5	✓	2	15					
6	✓	2	16					
7	✓	2					Grand Total	26
8	✓	2						
9	✓	2						
10	✓	2						
Total		20	46/50 Grand Total				Name and Signature of the Examiner with date	

To be filled by the examiner

Course Outcomes	1	2	3	4	5	6	Total
Marks allotted	50	50	-	-	-	-	50
Marks Obtained	46	46	-	-	-	-	46

IQAC Audit - Remarks

Verified

Name and Signature of the IQAC member

17/04/24

Unit Test - II.

Name : P Revathi
Reg no : 732422104036
Dept : BE - CSE / II year.
Subject : Data Base Management System.
Subject Code : CS3492.
Date :- 17/04/2024.

46/50

verified by,
P. R

Part - A.

1. Explain Entity Relationship Model :-

* the Entity Relationship is denoted by ER.

* the ER Model is specify the Data to represent the Enterprises of Schema in the Data of Structure of logical and database.

* the ER Model is Represent the Schema of Entity Relationship.

2) Give the limitation of E-R Model? How do you Over Come this?

(1) Loss of Information Content - the loss of the entity from the ER Model

(2) Less time of representation - they are representation of ER Model.

(3). No representation of Manipulating Data - they are Manipulated from the ER Model.

2) High Level design :- They can be high level of Data warehouse design into ER-Model.

3) What is an entity :-

* The Entity is the object of the DataBase is with distinguished to some other objects there is called Entity.

For Example : They student name of "poonam" the entity of the person is the Name. There is Related from the table to Student Name can be entity.

4) Define functional Dependancy :-

* The two column of set in P and Q is the table of functional Dependancy in P is the Determine functional is written of the $P \rightarrow Q$ then they have two row of the Dependant of entity are equal on the P equal on the Mode.

$T_1.P = T_2.P$ and $T_1.Q = T_2.Q$.

They are functional Dependant with Tuple of project is (T) and attribute of the (P)

Then P is the functional Dependancy are equal.

TNPL
DATE / /

5) trival functional Dependancy:-

* They are table of A->B in the left of A in the value of Dependancy B in the MVD of the functional Dependancy in the A,B in the trival of B. then the B in the equalent values.

For example:- the table two column names and A and B the multiple there is a partial order of Dependancy in the name, course id is calculed

---

b) Define Normalization:-

* the Normalization is reorganization of the Data to need from the Data base.

* They are conditions of Data in Normalization.

* No dependancy of the table

* Data redundancy of the table

* It is import of Data base take up last loss of Data.

* It also help to need to Improved Performense.

TNPL

7) Multivalued Dependancy?

* the table is should have Multivalued Dependancy, it satisfy the following condition. true

* It should set of column is A → B is the A is the functionally Dependancy, B is the Multivalued Dependancy there table is called Multivalued dependancy.

* It should table has at-least 3 columns in the Multivalued Dependancy.

* they have Relation of R(A B C) is the A and B then B and C is the Independent of the table is the Multivalued Dependancy.

---

8) BCNF and the relation which is BCNF?

* BCNF is denoted by the Boyce-Codd Normal For.

* they BCNF is the right value of the represented with the certain types of the relation.

* they are non relation of Dependancy is the BCNF.

* It can be third Normal form of BCNF. It should satisfy the form.

* They are anamal is not is 3NF is BCNF.

* there is called BCNF.

EX:

Sid	Cource	place.
Reva	BE	Dubai.
Dharsh.	BE	Chennai.

a) Necessary to Decompose a relation:-

* Hey are Nessary to the Decompose a relation is One Single table value break down by the Multi value is Decompose.

* Here is Decompose of the relation is eliminated into the relation

b) Two Desirable properties of Decomposition:-

(i) Loss-Loss or nonloss :- It is have the Decompose of Strored to the original Data storage of the entity from the Data Base.

2) Dependant retention:- It should have whether the all Information can handle the Property of the client with the retention.

Part-B.

11) Explain in Detail about Normal For

Normal Form :-

* the Normalization is the process of reorganization of Data to need for the Data Base.

* they are Normal form to Satisfy the two Basic Components:

- It has no Dependancy of the tab
- redundancy of the table

* It should have the table in Normal form is important of Data Base in take up less of the time

* It also need to help for the Improved performance.

(i) First Normal form :-

* they are Denoted by 1NF in the Data Base.

* It should have table of only single attributa / columns.

* It should table is stored in the Data Base.

* It has table row, Colume name has unique name.

* the table of Data Base not a matter.

Example :-

Student

Sname	Sid	
Sid	S.Name	Phone no.
Deva	001	1144
Dharsh		2244
Para	002	3422
Sri.		2243.

Sname	Sid	Phone no.
Deva	001	1144
Dharsh	001	2244
Para	002	3422
Sri.	002	2243.

* It should have Sid can be Independent they have one by one has split to the table where the 4 row of three columns in the table

(ii) Second Normal Form:-

* It should have First Normal form.

* It should denoted as a 2NF.

* It shold not have partially functionaly Dependancy.

Example:

Student and Course:

Sid	Sname	Cid	Coursename.
001	Reva.	101	BE.EBB
002	Dharsh	102	ECE.
003	Paru	103	CSE
004.	Sri	104.	Mech.

Step 1: It is the 1NF

Step 2: Hey have Sid Snamed and associated with Cid → course name to beegwat. So Hey have Sid=2 Can be deleted to the table to Course of ECE Can be deleted.

Step 3: Sid → Sname and Sid-Course name It hot Partial functionally Dependancy.

13/4/24

Sid	Sname	Cid.
001	Reva	101
002	Dharsh	102
003	Fary.	103
004.	Sri.	104.

Cid	Cname.
104	BE. EEE
102	ECE
103	CSE
104.	Mech.

(iii) Third Normal form:-

* It has Denoted from the 3NF.
* It should have 2 Normal form.
* It is the not transitive Dependancy.

They have 3NF is the 2NF and 3NF is similarly functional Dependancy.

$X \rightarrow Y$

X - Super key of the table
Y - Prime attribute of the table

Sid	Sname	zipcode	cityname State
1	Reva	4444	Mayilai tamil nadu
2	Dharsh	2222	Erode tamil
3	Fary	3333	thirupur odisa
4	Sri	1111	Selar. Satfisa

12) Multivalued Dependancies and Fourth Normal form :-

Multivalued Dependency :-

* the table of the Multivalued Dependancy is the satisfy the following Condition is true

* It should have two set of column in A, B is the A is the single value Dependancy and B is the Multivalued Dependancy they are the table is Multivalued Dependency.

* It should have at-least 3 Columns in the table of Multivalued Dependancy.

* It Relation of R(ABC) is the forms in A and B then B and C is Independance of the for is MVD.

they are satisfy the three Conditions is true.

* It Can be for example A|BC
A is the Multivalue Dependancy
B is the Multivalued Dependency of the they three Columns are Independent

* It should have Multivalued Dependancy is denoted by $\twoheadrightarrow$

* They have A and B is the not have MVD is the table in the fourth normal form.

Example.

Sid	Course	Place language
001	BE-CSE BE-ECE	English French.
002.	Mech EEE	English german.

*They have It can be Independent of the Coloumn in the table the they are Inpendent of the table for the represented

Sid	Course	Place language.
001	BE-CSE	English
001	BE-CSE	French.
001	BE-ECE	English
001	BE-ECE	French.
002	Mech	English
002	Mech	german
002	EEE	English
002	EEE	german.

* the table should have dependent of the coloumn for splits of the Date Base Normal forms.

Fourth Normal Form:-

* The forthNormal is the 1NF is the following it satisfy the two Condition

* It should have Boyce-codd Normal form.

* It should NO Multivalued Dependency.

Example.

Sid	Sname	Course	Skills
001		BE-CSE.	Playgames.
002		BE-ECE	foodball
003		BE-EEE	athlete
004.		Mech.	coreame

It should have devoted frome Independent to Dependent.

Sid_Course - Form

Sid	Course.
001	BE-CSE
002	BE-ECE
003	BE-EEE
004.	Mech

Spl_Skills : Spill of the forth Norma form.

Spl	Skills
001	playgames.
002	footbaal.
003	athlola
004.	Corouna.

3.

SASURIE
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

SASURIE INSTITUTIONS

Answer Book

Name	V. Haripriya			Year/ Semester/Section	II / IV
Register Number	732422104018	Date/Session	24.5.24	Department	CSE
Course code	CS3492	Course Title	Database Management System		
Internal Assessment Test	IAT 1 ☐	IAT 2 ☐	IAT 4 ☑	Model ☐	
Name and Signature of the Invigilator with date					

Instruction to the Student: Put tick mark to the question attended in the column against question.

Part A			Part B/ Part C						Total Marks
Q. No.	✓	Marks	Q. NO.	✓	a Marks	✓	b Marks		
1	✓	2	11	✓	12				12
2	✓	2	12	✓	13				13
3	✓	2	13						
4	✓	2	14						
5	✓	2	15						
6	✓	2	16						
7	✓	2	Grand Total						25
8	✓	2	45/50						
9	✓	2							
10	✓	2							
Total		20	Grand Total						Name and Signature of the Examiner with date

To be filled by the examiner

Course Outcomes	1	2	3	4	5	6	Total
Marks allotted	-	-	-	50	-	-	50
Marks Obtained	-	-	-	45	-	-	45

IQAC Audit - Remarks

Verified

Name and Signature of the IQAC member

24/5/24 V. Haripriya 24/05/24
732422104018
BE CSE
DBMS Unit Test-IA
PART-A

1. B-tree:

* B-tree is a self-balancing search tree in which data is stored in sorted order and allows searches, sequential access, insertion and deletion in logarithmic time.
* It is used in database and file systems to ensure that the tree remains balanced with its height kept low for efficient operation.

2. Rotational latency time:

* It refers to the delay caused by the time it takes for the desired sector of a disk to rotate under the read/write head.

3. Index:

* An Index is a data structure that improves the speed of data retrieval operation on a database table.
* It provides quick access to rows in a database table by mapping key.

4. Types of storage device:

* Primary storage
* Secondary storage
* Tertiary storage
* Offline storage.

5. Hashing file organization:

* Hashing file organization is a technique where a hash function is applied to a key field to determine the address of the data record

6. Static Hashing:

* The hash table is fixed, and collision are managed using techniques like changing or open addressing.

Dynamic Hashing:

* The hash table size can grow or shrink dynamically based on data. Technique like extendible hashing, a linear hashing

7. Query processing:

* Query processing involves translating a high-level query written in a query language into low-level instruction that the database can execute.

8. Query optimization:

* It is the process of selecting the most efficient way to execute a given query.
* The database management system evaluates different execution plans and chooses the one with the lowest cost.

9. Primary Index:

* An Index on a primary key, which ensures uniqueness and is directly tied to the physical ordering of data.

Secondary Index:

* An index on non-primary keys, allowing efficient access to data based on fields other than the primary key.

10. Types of ordered Indices:

* Dense Index
* Sparse Index
* Clustered Index
* Non-clustered Index.

PART-B

11(a) RAID:

* Redundant array of independent disk (RAID)
* originally redundant array of inexpensive disks is a way of storing the same data in different places on multiple hard disk to protect data in the case of a drive failure.

* Disk organization techniques that manage a large number of disk, providing a view of a single disk of high capacity and high speed by using multiple disk in parallel and high reliability by storing data redundantly.

Benefits of RAID

* Data loss can be very dangerous for an organization
* RAID technology prevents data loss due to disk failure.

RAID 0:

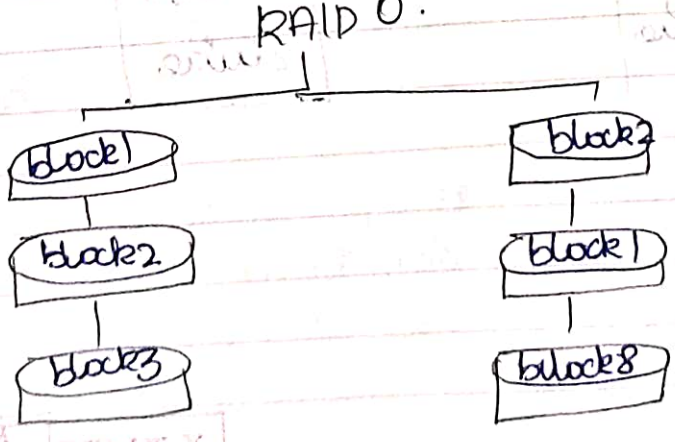
* RAID level 0 divides block units and writes them across a number of disks
* As data is placed across multiple disk. It is also called data striping
* There is no parity checking of data, then so if data in one drive get corrupted then all the data would be lost

Advantages:

* I/O performance is greatly improved by spreading the I/O load across many channels & drives.
* Best performance is achieved when data is striped across multiple controller with only one drive per controller.

Disadvantages:

* It is not fault-tolerant, failure of one drive will result in all data in an array being lost.

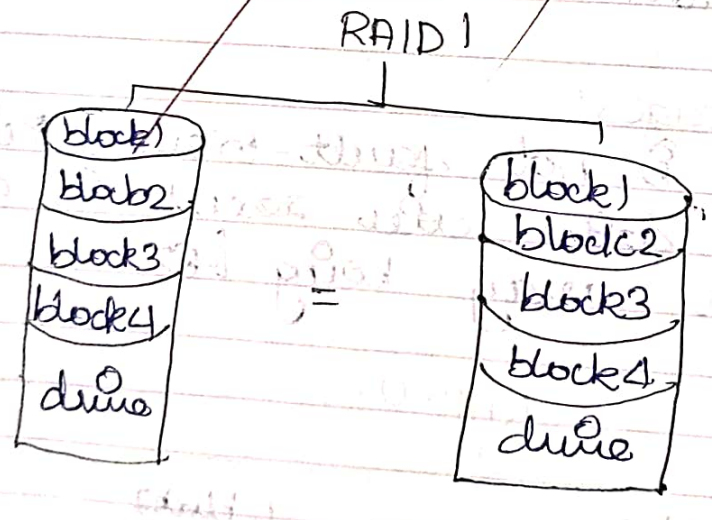


RAID level 1 :

* Also known as disk mirroring this configuration consists of at least two drives that duplicate the storage of data. There is no striping

* Read performance is improved since either disk can be read at the same time. Write performance is the same as for single disk storage.

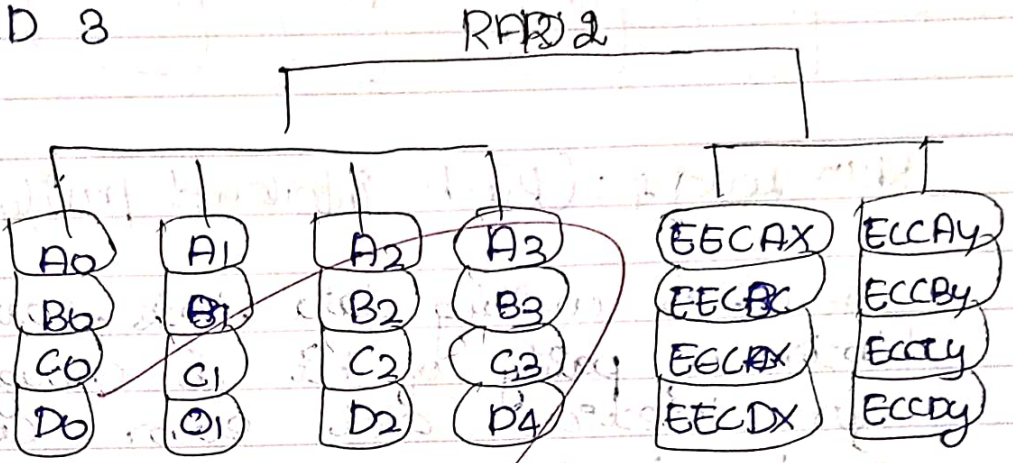
* Data loss would occur only if a disk fails, and its mirror disk also fails before the system is repaired. probability of combined event is very small.



RAID level 2:

* This configuration uses striping across disks, with some disks storing error checking and correcting (ECC) information.

It has no advantages over RAID 3



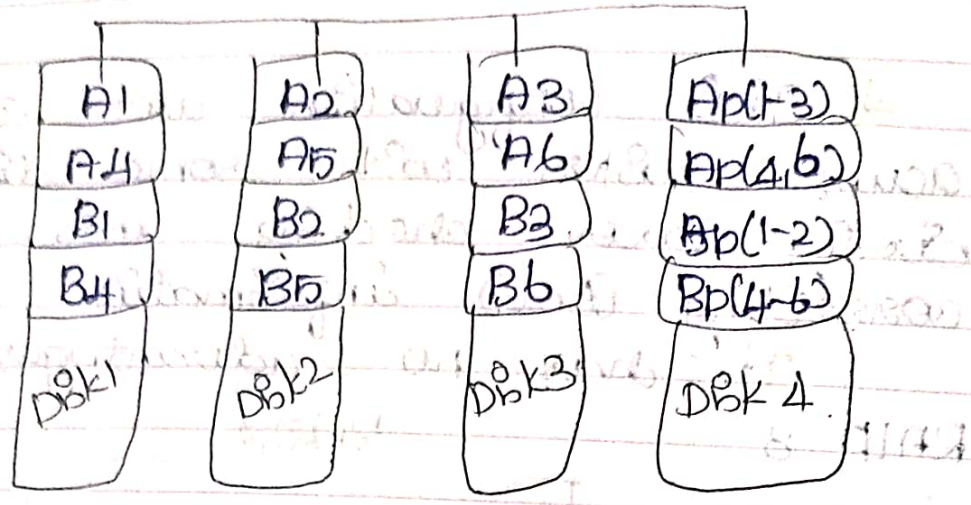
RAID Level 3:

* Bit-interleaved parity is level 3

* A single parity bit is enough for error correction, not just detection since we know which disk has failed.

* When writing data, corresponding parity bit must also be computed and written to a parity bit disk

RAID 3

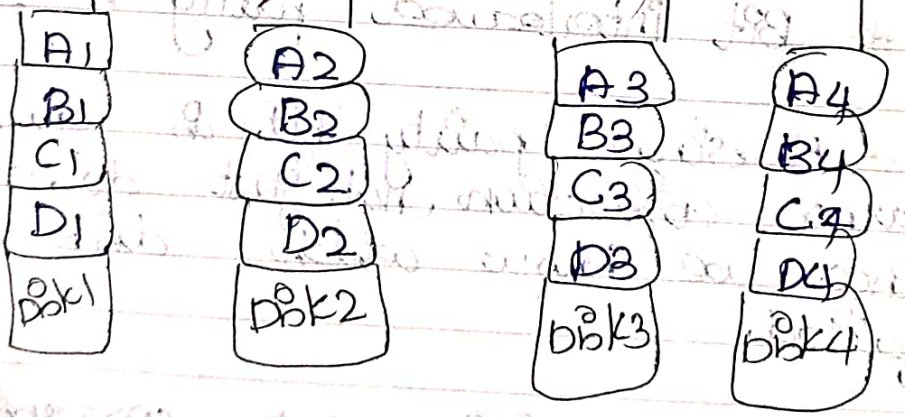


RAID Level 4 : (Block - Interleved parity)

* When writing data block, corresponding block of parity of bit is enough for detection and error correction must abo be computed and written to parity disk.

* Each entire block written onto a data disk.

RAID 4



(b) Hashing :

* It is a technique used in database management system to efficiently retrieve data used based on a specific search key.

* Instead of searching the entire database sequentially, hashing uses a hash function to map a search key to a specific location in the memory or storage.

* This reduces the search time to a constant time complexity O(1)

Types of Hasing:

* Hashing can be broadly classified the following types.

Static Hashing:

* In static hashing, the hash function and number of buckets are fixed at the time of database creation. This means the structure does not grow or shrink dynamically

Advantages:

* Simple to implement
* Suitable for systems where the database size is stable.

Disadvantage:

* Inefficient for dynamic database as it cannot handle significant growth.
* collision may increase as the database grows.

collision Resolution Techniques:

* Chaining
* Open addressing
  1) Linear probing
  2) Quadratic probing
  3) Double hashing

Dynamic Hashing:

* Dynamic hashing adjusts the size of the hash table dynamically as the database grows or shrinks.
* This is commonly used in database where the number of records changes frequently.

Advantages:

* Efficient for growing or shrinking database
* Reduce overflow issues by dynamically adjusting the structures.

Methods in Dynamic Hashing:

1) Extendible hashing
2) Linear hashing

Application of Hashing:

1) Indexing: Hashing is widely used for creating hash-based indexes for faster searches.

2) Caching: Hash function are used in caching mechanisms to map data to memory location

3) Password storage: Hashing ensures security by storing hashed passwords

4) Data deduplication.

CONTENT BEYOND THE SYLLABUS

1. Advanced SQL Techniques

- Window Functions: Used for complex data analysis within partitions of data.
- Recursive Queries: Useful for hierarchical data, such as organization charts or folder structures.
- Common Table Expressions (CTEs): For creating temporary result sets to use within a SELECT, INSERT, UPDATE, or DELETE statement.
- JSON and XML Data Handling: Storing and querying semi-structured data within SQL databases.

2. Indexing and Optimization Techniques

- Advanced Indexing Types: Such as partial, clustered, and composite indexes, and when to use each.
- Query Optimization Techniques: Deep dive into the query optimizer, execution plans, and using hints to influence the optimizer.
- Database Statistics: Understanding how databases gather statistics and how it impacts query performance.
- Materialized Views: Creating precomputed views that improve performance for frequently accessed data.

3. Concurrency Control and Transaction Management

- Isolation Levels and Consistency Models: Including Serializable, Read Committed, Read Uncommitted, and Repeatable Read.
- Optimistic vs. Pessimistic Locking: Different strategies for handling simultaneous transactions.
- Multiversion Concurrency Control (MVCC): Used in databases like PostgreSQL and Oracle to handle concurrent transactions.
- Deadlock Detection and Resolution: Methods for managing deadlocks in database transactions.

4. Distributed Databases and Sharding

- CAP Theorem: Understanding the trade-offs between consistency, availability, and partition tolerance.
- Data Partitioning: Horizontal vs. vertical partitioning, and hash-based, range-based, and composite sharding.
- Replication: Types of replication (master-slave, master-master) and their trade-offs.
- Eventual Consistency: Used in distributed systems to provide availability and partition tolerance.

5. **NoSQL and NewSQL Databases**

- Types of NoSQL Databases: Key-value, document-based, column-family, and graph databases, with examples like MongoDB, Cassandra, and Neo4j.
- Event Sourcing and CQRS (Command Query Responsibility Segregation): Techniques used in high-volun e applications.
- NewSQL Databases: A newer class that provides SQL capabilities with the scalability of NoSQL, like CockroachDB and Google Spanner.

6. **Data Warehousing and ETL (Extract, Transform, Load)**

- Star Schema and Snowflake Schema: Data modeling for analytical databases.
- ETL Pipelines: Tools and processes for extracting, transforming, and loading data, including Apache NiFi, Talend, and Informatica.
- Columnar Storage: Used in databases like Amazon Redshift and Google BigQuery for analytical processing.
- Data Lake vs. Data Warehouse: Understanding the differences and when to use each.

7. **Database Security and Compliance**

- Encryption Techniques: Data-at-rest and data-in-transit encryption.
- Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC): Advanced security models.
- SQL Injection and Prevention: How to prevent injection attacks.
- Compliance Standards: GDPR, HIPAA, and their implications on data storage and processing.

8. **Database Backup, Recovery, and High Availability**

- Point-In-Time Recovery (PITR): Techniques for restoring data to a specific time.
- High Availability Strategies: Failover clustering, active-active vs. active-passive configurations.
- Database Snapshots: Using snapshots for backup and quick recovery.
- Log Shipping and Replication: Techniques to ensure data is mirrored in real-time or near-real-time.

SASURIE INSTITUTIONS

SASURIE
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Assignment Question Paper

Assignment – 1			Date of Issue: 15/03/2024	Marks	10
Course code	CS3492	Course Title	DataBase management system		
Year	II	Semester/Section	IV	Date of Submission:	28/03/2024

Q.No	Questions	CO
1.	Explain in detail about relational algebra with example	CO1

Name and Signature of the Faculty Incharge

HoD/CSE

SASURIE INSTITUTIONS

SASURIE
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Assignment Answer Sheet

Name of the Student: S. Sri Raj

AU Register Number: 732422104043

Assignment – 1			Date of Issue: 15/03/2024	Marks	10
Course code	CS3492	Course Title	DataBase Management System		
Year	II	Semester/Section	IV	Date of Submission:	28/03/2024

Q.No	Questions	CO
1	Explain in detail about Relational Algebra with example	CO1
2		
3		

Mark Allocation

Rubrics	Marks Allocated	Marks obtained
Content Quality	6	5
Presentation Quality	2	1
Timely submission	2	1
Total marks	10	7

Name and Signature of the Faculty Incharge

HoD/CSE

Assignment - I

Name : A. Shri Raj

Subject : Data Base Management Systems

Reg No : 732422104043

Date : 22.03.2024

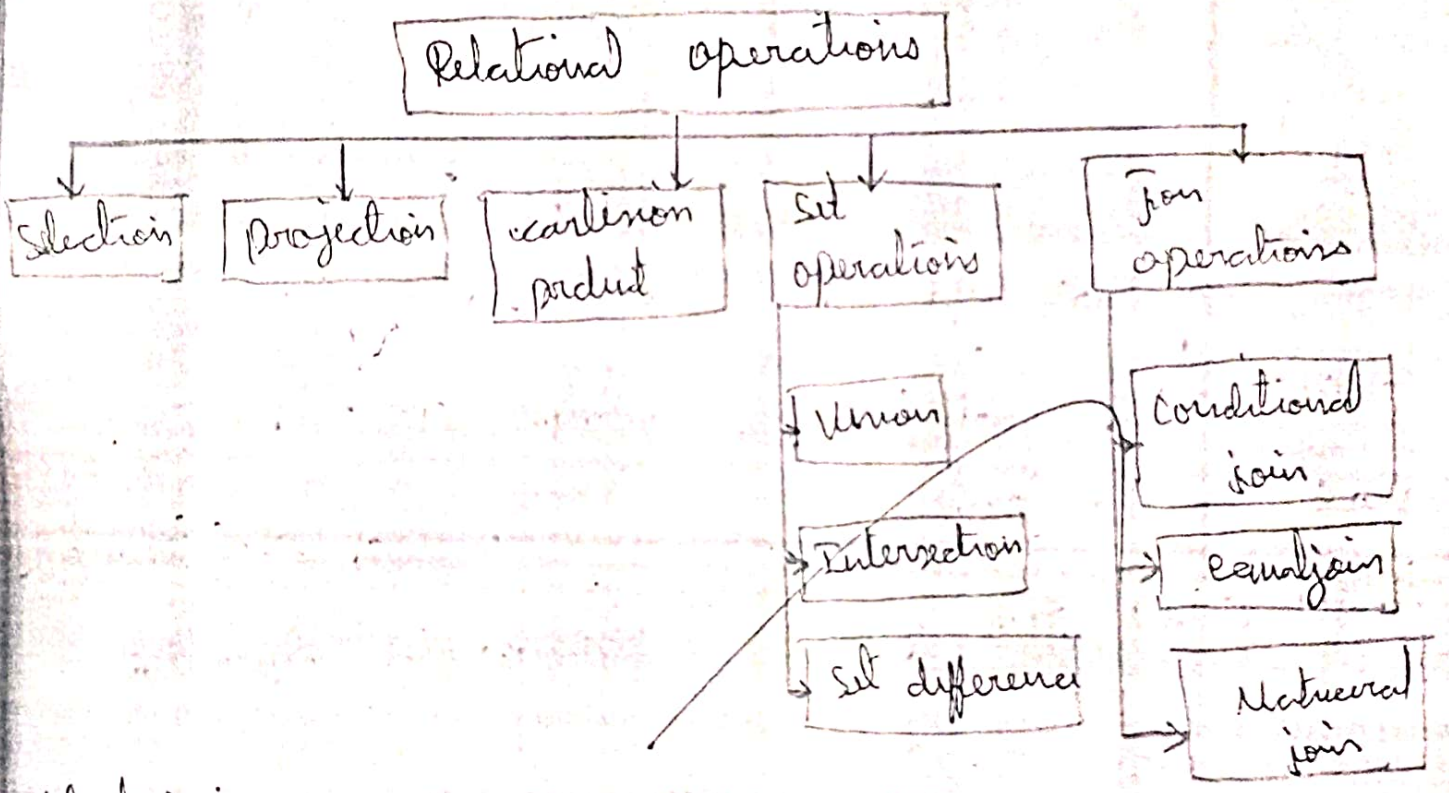
22/3/24.

Unit - I

Relational Algebra:

* Relational algebra is a procedural query language which is used to access the database tables to read data in different ways.

* The queries present in the relational algebra are denoted using operators.



Selection:

This operation is used to fetch rows or tuples.

Syntax: $\sigma_{Predicate}$ (table name or relation)

Sid	Sname	Age	Gender
1	Ram	21	M
2	Rasa	32	M

Case :

11 students with age more than 18

σ age > 18 (Students)

S name
Ram
Rasa

Projection :

projection operation is used to project only a certain set of attributes of relation. That means if you want to see only the name coll of the students in the table then you can use project operation.

Syntax : Π $C_1, C_2 \ldots$ (Relation)

Query : Display the name and age of all the st

Π Sname, age (Students)

Student name	age
Ram	21
Shyam	18
Seeta	16
Geeta	23

3) Cartisian product :

This is used to compile data from two different relation (tables) into one and a fetch data from the compined relation.

Syntax : A × B

For Example : Suppose we have two tables students and result as.

Students		
SID	Sname	age
1	Ram	21
2	Shyam	18
3	Seeta	16
4	Geeta	23

Result		
SID	ISBN	class
1	005	07.07.18
2.	007	29.03.91
3.	007	17.07.91

Query: find the names roll the students who have
ISBN = 007 to satisfy the Query.

Student.sid : Reserve.sid ^ Reserve.ISBN : 007

Out put: (Student.id x Reserve)

Sid	Sname	rating	Sid	ISBN	date
2	Clyan	19	2	007	29.03.23
3	Peter	16	2	007	17.09.24

Set operation:

Various set of operations are union inter
and set difference.

Union: This operation is used to fetch data from
relation (tables).

Syntax: A ∪ B

Example:

	Student		Book		
Sid	Sname	age	isbn	bname	Author
1	aaa	18	005	DBMS	XXX
2	bbb	18	006	OS	YYY

Query:

We want to display both the structure and
name from both the tables.

Π_{sname} (student) $\cup$ Π_{bname} (book)

output:

Sname	bname
aaa	DBMS
bbbb	OS

Intersection:

This operation is used to fetch data from both the tables which is common in the both tables.

Syntax: $A \cap B$

Example:

student	
Name	Branch
aaa	CSE
bbb	IT

worker	
Name	Salary
aaa	50000
bbb	60000

Query:

If we want to find out the names of the students who are working in a company.

Π_{name} (student) $\cap$ Π_{name} (worker)

Out put:

Name
aaa
bbb

Set - difference:

* The result of set difference operation on tuples, which are present in one relation but are not in second relation (tables).

Syntax: A - B

Example:- Consider two relation full time and part time employee, if we want to out working for full time then find the set - difference.

Π empname (Full_time_employee) -

Π empname (Part_time_employee).

Join operation:

* A join operation is used to common information from two or more relation.

* Join operation is used as $\bowtie$.

Condutional Join:

This is an operation information from two to condition same condition and this condition specified during join operation.

$$A \bowtie_{\theta} B = \sigma_{\theta}(A \times B)$$

Equal join:

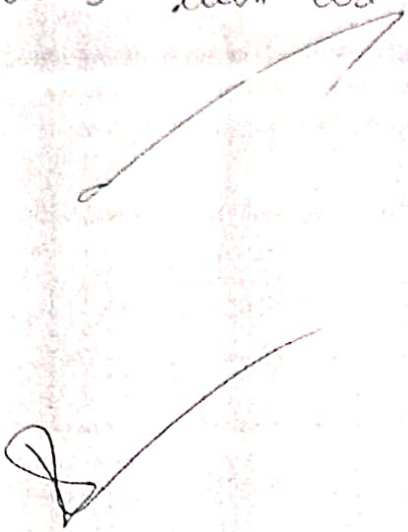
This is a kind of join operation in which

is equality condition between two attributes of relation

Natural join:

when there are common colums and we have to equate its common colums then we use natural join.

Symbol : A ⋈ B.



SASURIE INSTITUTIONS

SASURIE COLLEGE OF ENGINEERING

Approved by AICTE, New Delhi
Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Assignment Answer Sheet

Name of the Student: B. Saran

AU Register Number: 732422104038

Assignment – 2			Date of Issue: 15/04/2024	Marks	10
Course code	CS3492	Course Title	DataBase Management System		
Year	II	Semester/Section	IV	Date of Submission:	25/04/2024

Q.No	Questions	CO
1	Explain in detail about Normal Forms with Suitable examples.	CO3
2		
3		

Mark Allocation

Rubrics	Marks Allocated	Marks obtained
Content Quality	6	4
Presentation Quality	2	2
Timely submission	2	1
Total marks	10	7

Name and Signature of the Faculty Incharge

HoD/CSE

Assignment - 2

Data Base Management System

Name : B. Saran

Dept : BE CSE

Sub Code : CS 8492

Reg no : 732422104038

2/5/24

Normal Forms:

* Normalization is the process of reorganizing data in a database so that it means two basic requirements:

1) There is no redundancy of data
2) Data dependencies are logical.

* The normalization is important because it allows database to take less space.

* It also helps in increasing the performance

First Normal form:

The table is said to be in 1NF if it follows rules:

1) It should only have single (atomic) valued attributes.
2) Values stored in a table should be of the same domain
3) All the column should be have unique names
4) And the order in which data is stored, does not matter.

Student

Sid	Sname	Phone
1	AAA	1111
2	BBB	2222
3	CCC	3333
		4444
4	DDD	5555

As, there are multiple values of phone number for Sid 1 and 3, the above table is not in INF We make it in 1NF. The conversion is a follow

Sid	Sname	Phone
1	AAA	1111
2	BBB	2222
3	CCC	3333
3	CCC	4444
4	DDD	5555

Second Normal Form:

Before understanding the second normal form let us first discuss the concept of partial functional dependency and prime and non prime attribute

Concept of Partial functional Dependency

Partial dependency means that a non prime

attributes is functionally dependent on part of a candidate key.

For example : Consider a relation R(A,B,C,D) with functionally dependent [AB -> CD->, A->C]

Here (AB) is an candidate key because

(AB)+ = {ABCD} = {R}

Prime and Non Prime Attributes

* Prime Attribute : An attribute, which is a part of the candidate key, is known as a prime attribute
* Non Prime Attribute : An attribute, which is not a part of the primary key, is said to be non prime attribute.

Example : Consider a Relation R = {A,B,C,D} candidate key as AB

The Second Normal form :

For a table to be in the second normal form, following condition must be followed.

i) It should be in first normal form
ii) It should not have partial functional dependency
iii) For example : Consider a table with every information about a student

Student _ Course

Sid	Sname	rid	Cname
1	AAA	101	C
2	BBB	102	C++
3	CCC	101	C
4	DDD	103	Java

This table is not in 2NF. For converting above table to 2NF we must follow the following steps:

Step 1: The above table is in 1NF

Step 2: Here Sname and Sid are associated similarly cid and Cname are associated with each other. Now if we delete a record with sid=3 then Automatically the course C++ will also get deleted.

Student

Sid	Sname	Cid
1	AAA	101
2	BBB	102
3	CCC	103
4	DDD	104

Course

Cid	Cname
101	C
102	C++
101	C
103	Java

Third Normal Form:

Concept of Super key and candidate key

Super key: A Super is a set of one of more columns to uniquely identify rows in a table

Candidate key: The minimal set of attribute which can uniquely identify a tuple is known as candidate key.

RegID	Rollno	Sname
101	1	AAA
102	2	BBB
103	3	CCC
104	4	DDD

Superkeys

* [RegID]
* [RegID, Rollno]
* [RegID, Sname]

* [RollNo, Sname]
* [RegID, Rollno, Sname]

Candidate Key:
* [RegID]
* [Rollno]

Third Normal Form:

A table is said to be in the Third Normal Form when,

i) It is in the Second Normal form
ii) It doesnot have transitive dependency

For example: Consider following table Student_details as follows.

Sid	Sname	Zipcode	Cityname
1	AAA	1111	Pune
2	BBB	2222	Surat
3	CCC	3333	Tamilnadu
4	DDDD	4444	Rajastan

Superkey: [Sid], [Sid Sname], [Sid, Sname, Zipcode] [Sid, Zipcode, Cityname]. and So,

Candidate key: [Sid]

Student

Sid	Sname	Zip code
1	AAA	1111
2	BBB	2222
3.	CCC	3333
4	DDD	4444

Zip

Zip Code	City name
1111	Pune
2222	Surat
3333	Chennai
4444	Jaipur

Agenda

- What is Data?
- Types of Data
- Structured Data and Unstructured Data
- How do we store Data into Computers ?
- File System Vs Database System
- File System Issues
- What is Management?
- What is DBMS?

What is Data?

- From Dictionary :

  Factual information (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation

- From Wikipedia :

  Data are characteristics , usually numerical, that are collected through observation. In a more technical sense, data is a set of values of qualitative or quantitative variables about one or more persons or objects, while a **datum** (singular of data) is a single value of a single variable.

- Example: Name, Address, Zip, SSN are characteristics or information about a person.

Types of Data

- **Unstructured-Data**
- **Structured – Information (interpreted data – data supplied with semantics)**

Structured and Unstructured Data

How do we store Data into Computers?.

- Data could be stored computers in File System and or Database Management and or content management systems.
- Let us focus on File System and Database approaches . Each one has its own advantages and disadvantages.

File System Vs Database System

File System **Database System**

File System Vs Database System Cont…

File System **Database System**

File System Issues

- **Redundancy of data**: Data is said to be redundant if same data is copied at many places. If a student wants to change Phone number, he has to get it updated at various sections. Similarly, old records must be deleted from all sections representing that student.
- **Inconsistency of Data**: Data is said to be inconsistent if multiple copies of same data does not match with each other. If Phone number is different in Accounts Section and Academics Section, it will be inconsistent. Inconsistency may be because of typing errors or not updating all copies of same data.
- **Difficult Data Access**: A user should know the exact location of file to access data, so the process is very cumbersome and tedious. If user wants to search student hostel allotment number of a student from 10000 unsorted students' records, how difficult it can be.
- **Unauthorized Access**: File System may lead to unauthorized access to data. If a student gets access to file having his marks, he can change it in unauthorized way.
- **No Concurrent Access**: The access of same data by multiple users at same time is known as concurrency. File system does not allow concurrency as data can be accessed by only one user at a time.
- **No Backup and Recovery**: File system does not incorporate any backup and recovery of data if a file is lost or corrupted.

What is Management?

- Generally Management refers create, Retrieve, update and delete.
- For example: Money Management refers earning the money, querying money, distribute the money and spend the money.

What is Database?

- **Database:** Database is a collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc. For Example, university database organizes the data about students, faculty, and admin staff etc. which helps in efficient retrieval, insertion and deletion of data from it.
- Now Database Management talks about
  - Create Database
  - Retrieve(View) database
  - Update database
  - Delete Database

What is DBMS?

- **Database Management System:** The software which is used to manage database is called Database Management System (DBMS). For Example, MySQL, Oracle etc. are popular commercial DBMS used in different applications. DBMS allows users the following tasks:
- **Data Definition:** It helps in creation, modification and removal of definitions that define the organization of data in database.
- **Data Updation:** It helps in insertion, modification and deletion of the actual data in the database.
- **Data Retrieval:** It helps in retrieval of data from the database which can be used by applications for various purposes.
- **User Administration:** It helps in registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control and recovering information corrupted by unexpected failure.

Database Languages

- A database system provides **a data definition language** to specify the database schema and a **data manipulation language** to express database queries and updates. In practice, the data definition and data manipulation languages are not two separate languages; instead they simply form parts of a single database language, such as the widely used SQL language

Data Definition Language(DDL)

We specify a database schema by a set of definitions expressed by a special language called a **data definition language (DDL)**.

For instance, the following statement in the SQL language defines the *account* table:

```
create table account (account-number char (10), balance integer)
```

Execution of the above DDL statement creates the *account* table. In addition, it updates a special set of tables called the data dictionary or data directory.

A data dictionary contains metadata—that is, data about data. The schema of a table is an example of metadata. A database system consults the data dictionary before reading or modifying actual data.

We specify the storage structure and access methods used by the database system by a set of statements in a special type of DDL called a data storage and definition language. These statements define the implementation details of the database schemas, which are usually hidden from the users.

The data values stored in the database must satisfy certain **consistency constraints**. For example, suppose the balance on an account should not fall below $100. The DDL provides facilities to specify such constraints. The database systems check these constraints every time the database is updated.

Data Manipulation Language(DML)

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.

DML Continuation…

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 retrieve information from more than one table. For instance, the following query finds the balance of all accounts owned by the customer with customer-id 192-83-7465:

DML Continuation…

```
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.

Database Management Systems
UNIT-I
Short answer Questions

1. Define the terms data and information?
2. Define (i) Database (ii)DBMS
3. List the advantages and applications of DBMS?
4. What are the disadvantages of file processing system?
5. Define instances and schemas of database?
6. What is data model? List the types of data models?
7. Discuss about Data Definition language?
8. Discuss about Data Manipulation language?
9. What is data Abstraction? Give the levels of data abstraction?
10. Who is DBA? What are the responsibilities of DBA?
11. Discuss Data Independence?
12. What is an entity relationship model?
13. Define (i) Entity (ii) Attribute
14. Define Relationship and Relationship set?
15. What are key constraint and participating constraints?
16. Define weak entity and strong entity sets?
17. Define relation, relation instance and relation schema.
18. Define i) super key ii)candidate key iii) primary key
19. Explain the use of foreign key constraint?
20. Define the terms arity and cardinality of relation?
21. What are domain constraints
22. Explain about querying relational data?
23. Define views?
24. Discuss how can you change the data in the table?
25. List various types of attributes?
26. Discuss how can you alter and destroy tables?
27. Explain the use of null values?

Long answer Questions

1. Compare and Contrast file Systems with database system?
2. Define Data Abstraction and discuss levels of Abstraction?
3. Discuss about different types of Data models?
4. Describe the architecture of DBMS?
5. Discuss additional features of the ER-Models?
6. Discuss about the Conceptual Design with the ER-Model?
7. Write about views and updates on views?
8. Explain different types of database users and write the functions of DBA?
9. Explain about different types of integrity constraints?
10. Discuss about the logical database Design?
11. Distinguish strong entity set with weak entity set? Draw an ER diagram to illustrate Weak entity set?
12. Explain how the integrity constraints are specified and enforces?
13. Explain in detail about views?

UNIT-II

Short answer Questions

1 Define relational database query?
2 Explain different types of query languages?
3 Explain about relational algebra?
4 State about SELECT operation in Relational algebra?
5 State about PROJECT operation in Relational algebra?
6 Explain about set operations?
7 Discuss the use of rename operation?
8 Define join ? Explain different join operations?
9 Illustrate division operation?
10 Explain about tuple relational calculus?
11 Explain about Domain relational calculus?
12 Discuss about the expressive power of relational algebra and calculus?

13 Discuss the basic form of SQL query?

14 Explain the working of union, intersection and except operations?

15 Define nested queries?

16 Define correlated nested queries?

17 Explain Aggregate Functions?

18 What is the use of groupby and having clauses?

19 Define Null Values?

20 Define tuple variable with its syntax?

21 Define outer join? Explain its types?

22 Explain how to create new domain?

23 Define Assertions?

24 Discuss about trigger?

25 Demonstrate how to add a NOT NULL column to a table?

26 Write a TRC query to find the names of sailors who have reserved boat103?

27 Write a DRC query to find the names of sailors who have reserved red boat?

Long answer Questions

1. Illustrate different operations in Relational algebra with an example?
2. Define Join? Explain different types of joins?
3. Discuss about Relational calculus in detail?
4. Define trigger and explain its three parts? Differentiate row level and statement level triggers?
5. Illustrate Group by and having clauses with examples?
6. Discuss about Complex integrity constraints in SQL?
7. Define null value? Describe the effect of null values in database?
8. Discuss different types of aggregate operators with examples in SQL?
9. a Define a nested query?

   b Write a nested query to find the names of sailors who have reserved both a red and green boat?

   c. Write a nested query to find the names of sailors who have reserved all boats?

9 a. Discuss correlated nested queries?

b. Write a query to find the names of sailors who have reserved a red boat?

c. Write a query to find the names of sailors who have not reserved a red boat?

UNIT-III

Short answer Questions

1. Define redundancy?
2. Define functional dependency?
3. Explain the problems with Redundancy?
4. What is decomposition? Explain the properties of Decomposition?
5. Discuss normalization?
6. Illustrate functional dependency with example?
7. Illustrate fully functional dependency with example?
8. Demonstrate transitive dependency? Give an example?
9. Define First Normal Form?
10. Define Second Normal Form?
11. Define Third Normal Form?
12. Explain about Loss Less Join Decomposition?
13. Describe Dependency Preserving Decomposition?
14. What is multi valued Dependency?
15. Define Fourth Normal Form?
16. Define Join Dependency?
17. Define BCNF?
18. Explain Fifth Normal Form?
19. Explain about Inclusion Dependency?

Long answer Questions

1. Illustrate redundancy and the problems that it can cause

2. Define decomposition and how does it address redundancy? Discuss the problems that may be caused by the use of decompositions?
3. Define functional dependencies. How are primary keys related to FD's?
4. Define normalization? Explain 1NF,2NF,3NF normal forms
5. Compare and contrast BCNF with 3NF?
6. Describe properties of decompositions

UNIT-IV

Short answer Questions

1. Define a Transaction? List the properties of transaction
2. Discuss different phases(states) of transaction?
3. What is shadow copy technique?
4. List the advantages of concurrent execution?
5. Define Schedule? What is a serial schedule?
6. Discuss the Procedure to test Serializability?
7. Demonstrate Conflict Serializability?
8. Discuss View Serializability?
9. Discuss recoverable schedules?
10. Discuss cascade less schedules?
11. Explain the procedure to test for serializability?
12. Explain about different types of locks?
13. Define Deadlock?
14. Explain about locking protocols?
15. Define Two Phase locking protocol?
16. Demonstrate the implementation of Isolation?
17. Explain how the locks are implemented?
18. Explain the rules of tree protocol?
19. What is timestamp? Explain different timestamps used by a transaction?
20. Explain Thomas write rule?
21. What are the phases of validation based protocol?
22. Explain different timestamps used by validation protocol?
23. Define granularity?

24. Discuss about Failure Classification?
25. What are different storage types?
26. What are the fields of update record?
27. Discuss log based recovery?
28. Differentiate deferred and immediate database modifications?
29. Define a checkpoint?
30. Discuss the failures that can occur with loss of Non-volatile storage
31. Explain about ARIES?

Long answer Questions

1. Explain ACID properties and illustrate them through examples?
2. Discuss How do you implement Atomicity and Durability
3. Illustrate Concurrent execution of transaction with examples
4. Discuss Serializability in detail?
5. Discuss two phase locking protocol and strict two phase locking protocols?
6. Describe Times tamp based locking protocols?
7. Describe Validation-based locking protocols?
8. Discuss in detail Multiple Granularity?
9. Explain in detail storage structure
10. Discuss how do you recover from failure?
11. Explain Buffer Management?
12. Explain different types of advanced recovery techniques
13. Write in detail about Remote Backup systems?

UNIT-V

Short answer Questions

1. Discuss about data on External storage?
2. What is indexing and what are the different kinds of indexing?
3. Explain Clustered Indexes?
4. Discuss the Primary and Secondary indexes?
5. Define Tree Indexing?
6. Explain Hash based Indexing?
7. Compare different file organizations?
8. Discuss the intuition for Tree Indexes?
9. Define Indexed Sequential Access Method?
10. Discuss about Overflow pages and Locking considerations of ISAM?
11. Discuss the Cost model of Heap files, Sorted files and Clustered files?
12. Explain the structure of B+ tree?
13. Describe how the insert and delete operations are performed in B+ tree?
14. Explain how search is performed in B+ tree?
15. Define static Hashing?
16. Explain extendible hashing?
17. Define linear hashing?
18. Differentiate between linear and extensible hashing?

Long answer Questions

1. Write in detail about hash based indexing and Tree based indexing
2. Compare I/O costs for all file organizations
3. Explain in detail about ISAM
4. Explain about B+ tree index file?
5. Demonstrate searching a given element in B+ trees? Explain with example?
6. Illustrate insertion of an element in B+ Tree with example
7. Illustrate deletion of an element in B+ Tree with example
8. Write in detail about Static Hashing
9. Explain in detail about Extendible hashing
10. Explain in detail about Linear hashing

11. Compare and contrast Extendible hashing With Linear hashing

Reg. No. :

Question Paper Code : 80099

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2019.

Third/Fourth Semester

Information Technology

CS 8492 — DATABASE MANAGEMENT SYSTEMS

(Common to Computer Science and Engineering/Computer and Communication Engineering)

(Regulation 2017)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is a data model? List the types of data model used.
2. List any eight applications of DBMS.
3. Give the properties of decomposition.
4. Define the terms Entity set and Relationship set.
5. What are the states of transaction?
6. What is meant by log-based recovery?
7. Define dense index.
8. Mention all the operations of files.
9. Mention two features of Multimedia databases
10. Compare sequential access devices versus random access devices with an example.

PART B — (5 × 13 = 65 marks)

11. (a) Explain the three different groups of data models with suitable examples.

Or

(b) Describe about the static and dynamic SQL in detail.

12. (a) What is normalization? Explain in detail about all Normal forms.

Or

(b) Briefly discuss about the functional dependency concepts.

13. (a) Discuss in detail about the testing of serializability.

Or

(b) Explain deferred and immediate modification versions of the log based recovery scheme.

14. (a) What is RAID? Briefly discuss about RAID.

Or

(b) Describe the structure of B+ tree and give the algorithm for search in the B+ tree with example.

15. (a) Discuss in detail about the distributed databases.

Or

(b) Explain in detail about the Deductive DB and Spatial DB.

PART C — (1 × 15 = 15 marks)

(Application / Design / Analysis / Evaluation /Creativity/Case Study questions)

16. (a) Discuss in detail about the ACID properties of a transaction.

Or

(b) What is concurrency control? How it is implemented in DBMS? Briefly elaborate with suitable diagrams and examples.

20/11/19/FN

Reg. No. : | | | | | | | | | | | | | |

Question Paper Code : 90156

B.E./B.Tech. DEGREE EXAMINATIONS, NOVEMBER/DECEMBER 2019
Third/Fourth/Fifth Semester
Computer Science and Engineering
CS 8492 – DATABASE MANAGEMENT SYSTEMS
(Common to Computer and Communication Engineering /Mechanical and Automation Engineering/Information Technology)
(Regulations 2017)

Time : Three Hours | Maximum : 100 Marks

Answer ALL questions

PART – A (10×2=20 Marks)

1. What are the four main characteristics that differentiate the database approach from the file-processing approach ?
2. Express in relational algebra, the division operation(/) using the project, cartesian product and minus operations. Give a simple example.
3. 'Boyce-Codd normal form is found to be stricter than third normal form'. Justify the statement.
4. What is the significance of "participation role name" in the description of relationship types ?
5. List the responsibilities of a DBMS has whenever a transaction is submitted to the system for execution ?
6. Brief any two violations that may occur if a transaction executes a lower isolation level than Serializable.
7. How do you represent leaf node of a B^+ tree of order p ?
8. Which cost components contribute to query execution ?
9. List information types of documents necessary for relevance ranking of documents in IR.
10. What one could understand from allocation schema ?

x) Departments have a professor (known as the chairman) who runs the department.

xi) Professors work in one or more departments, and for each department that they work in, a time percentage is associated with their job.

xii) Graduate students have one major department in which they are working on their degree.

xiii) Each graduate student has another, more senior graduate student (known as a student advisor) who advises him or her on what courses to take.

Design and draw an ER diagram that captures the information about the university.

Use only the basic ER model here; that is, entities, relationships and attributes. Be sure to indicate any key and participation constraints. (5+10)

(OR)

b) i) For the following relation schema R and set of functional dependencies F : R(A, B, C, D, E), F = { AC → E, B → D, E → A}. List all candidate keys. (6)

ii) Consider the Table-16 and answer to queries given below. (9)

Table-16 User_personal

Userid	U_Email	Fname	Lname	City	State	Zip
MA12	mani@ymail.com	Manish	Jain	Bilaspur	Chatisgarh	458991
PO45	punag@gmail.com	Pooja	Magg	Kacch	Gujrat	832212
LA33	lavle98@jj.com	Lavleen	Dhalla	Raipur	Chatisgarh	853578
CH99	chekn9j@ih.com	Chimal	Bedi	Trichy	Tamil Nadu	632011
DA74	danu58@g.com	Dany	James	Trichy	Tamil Nadu	645018

1) Is this table in First Normal Form-1NF ? Justify and normalize to 1NF if needed.

2) Is this table in Second Normal Form-2NF ? Justify and normalize to 2NF if needed.

3) Is User_personal in Third Normal Form-3NF ? Justify and normalize to 3NF if needed.

PART – B (5×13=65 Marks)

11. a) i) Consider the following schema :

Suppliers (sid : integer, sname : string, address : string)

Parts (pid : integer, pname : string, color : string)

Catalog (sid : integer, pid : integer, cost : real)

The key fields are underlined and the domain of each field is listed after the field name. Therefore sid is the key for Suppliers, pid is the key for Parts and sid and pid together form the key for Catalog. The Catalog relation lists the prices charged for parts by suppliers. (6)

Write the following queries in relational algebra.

1) Find the sids of suppliers who supply some red or green part.

2) Find the sids of suppliers who supply some red part or are at 221 Packer Street.

3) Find the pids of parts supplied by atleast two different suppliers.

ii) Sketch the typical component modules of DBMS. Indicate and explain the interactions between those modules of the system. (7)

(OR)

b) i) Consider the schema given in question no. 11.a) i) and write the following queries in SQL. (8)

1) Find the names of suppliers who supply some red part.

2) Find the sids of suppliers who supply some red part and some green part.

3) Find the sids of suppliers who supply every red part.

4) Find the pids of parts supplied by atleast two different suppliers.

ii) Explain the three schema architecture with a neat diagram. (5)

12. a) i) Discuss in detail the steps involved in the ER-to-Relational mapping in the process of relational database design. (7)

ii) Exemplify the multi-value dependency and the fourth normal form-4NF. (6)

(OR)

b) i) Explain with suitable example, the constraints of specialisation and generalization in ER data modeling. (7)

ii) Exemplify the join dependency and the fifth normal form-5NF. (6)

13. a) i) Discuss elaborately the two-phase locking protocol that ensures serializability. (9)

ii) Brief the states of a transaction with a neat diagram. (4)

(OR)

b) i) Narrate the actions that are considered for deadlock detection and the recovery from deadlock. (9)

ii) Discuss the properties of a transaction that ensure integrity of data in the database system. (4)

14. a) i) Explain the various levels of RAID systems. (10)

ii) Why data dictionary storage is important ? (3)

(OR)

b) i) With simple algorithms explain the computing of Nested-loop join and Block Nested-loop join. (10)

ii) Sketch and concise the basic steps in Query Processing. (3)

15. a) i) Illustrate the usage of OQL, the DMG's query language. (9)

ii) Brief on the methods to store XML documents. (4)

(OR)

b) i) Illustrate the approaches to store relations in distributed database. (9)

ii) How effectiveness of retrieval is measured ? Discuss. (4)

PART – C (1×15=15 Marks)

16. a) Consider the following information about a university database :

i) Professors have an SSN, a name, an age, a rank and a research specialty.

ii) Projects have a project number, a sponsor name (e.g., NSF), a starting date, an ending date and a budget.

iii) Graduate students have an SSN, a name, an age and a degree program (e.g., M.S. or Ph.D.).

iv) Each project is managed by one professor (known as the project's principal investigator).

v) Each project is worked on by one or more professors (known as the project's co-investigators).

vi) Professors can manage and/or work on multiple projects.

vii) Each project is worked on by one or more graduate students (known as the project's research assistants).

viii) When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one.

ix) Departments have a department number, a department name and a main office.