Database Management Systems

1. Introduction

Introduction; An example; Characteristics of Database approach; Actors on the screen; Workers behind the scene; Advantages of using DBMS approach; A brief history of database applications; when not to use a DBMS.

Data models, schemas and instances; Three-schema architecture and data independence; Database languages and interfaces; The database system environment; Centralized and client-server architectures; Classification of Database Management systems.

2. Entity-Relationship Model

Using High-Level Conceptual Data Models for Database Design; An Example Database Application; Entity Types, Entity Sets, Attributes and Keys; Relationship types, Relationship Sets, Roles and Structural Constraints; Weak Entity Types; Refining the ER Design; ER Diagrams, Naming Conventions and Design Issues; Relationship types of degree higher than two.

3. Relational Model and Relational Algebra

Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations, Transactions and dealing with constraint violations; Unary Relational Operations: SELECT and PROJECT; Relational Algebra Operations from Set Theory; Binary Relational Operations: JOIN and DIVISION; Additional Relational Operations; Examples of Queries in Relational Algebra; Relational Database Design Using ER- to-Relational Mapping.

4. SQL

SQL Data Definition and Data Types; Specifying basic constraints in SQL; Schema change statements in SQL; Basic queries in SQL; More complex SQL Queries. Insert, Delete and Update statements in SQL; Specifying constraints as Assertion and Trigger; Views (Virtual Tables) in SQL;

Additional features of SQL; Database programming issues and techniques; Embedded SQL, Dynamic SQL; Database stored procedures and SQL/PSM.

5. Database Design

Informal Design Guidelines for Relation Schemas; Functional Dependencies; Normal Forms Based on Primary Keys; General Definitions of Second and Third Normal Forms; Boyce-Codd Normal Form.

6. Transaction Management

The ACID Properties; Transactions and Schedules; Concurrent Execution of Transactions; Lock- Based Concurrency Control; Performance of locking; Transaction support in SQL; Introduction to crash recovery; 2PL, Serializability and Recoverability; Lock Management; Introduction to ARIES; The log; Other recovery-related structures; The write-ahead log protocol; Checkpointing; Recovering from a System Crash; Media Recovery; Other approaches and interaction with concurrency control.

Text Books:

1. Elmasri and Navathe: Fundamentals of Database Systems, 5th Edition, Addison-Wesley, 2007
2. Raghu Ramakrishnan and Johannes Gehrke: Database Management

Systems, 3rd Edition, McGraw-Hill, 2003.

Reference Book:

1. Silberschatz, Korth and Sudharshan: Data base System Concepts, 5th Edition, Mc-GrawHill, 2006.

CHAPTER 1: INTRODUCTION

DATABASE MANAGEMENT SYSTEM

Data may be used to represent thing like name, telephone number, address, people name. The data can be stored using Microsoft ACCESS, EXCEL, etc. The data base is collections of data are group together to make large information. The database system is collection of programs that enables the users to create and manipulate a dbase. For example, to creates a new account in a bank and creates a new account in yahoo. The database management system is a general purpose software system, in which specifies defining, constructing and manipulating of the database.

DATABASE(Dbase) MANAGEMENT SYSTEM

1. It is collection of programs that enables users to create and manipulate a dbase
a. Ex: creates a new account in yahoo.com
b. Ex: creates a new account in a bank
2. It is a general purpose software system
3. It specifies defining, constructing and manipulating

An Example

1. In University database system, it has four files namely STUDENT, COURSE, SECTION, GRADE_REPORT schema
2. Ex: Student dbase
3. STUDENT file has four attributes

Table 1. STUDENT file

Student name	Register no	Semester	Major
Varaprasad	001	1	CSE
Prasad	004	2	ECE

4. Ex: COURSE file has four attributes

Table 2. COURSE file

Course name	Course no	Credit	Dept
OS	08SCS11	3	CSE
DS&A	08SCS12	4	CSE
DBMS	08SCS13	4	CSE
CSPA	08SCS14	4	CSE
TFCS	08SCS151	3	CSE

5. Ex: SECTION file has five attributes

Table 3. SECTION file

Section no	Course no	Semester	Year	Instructor
1	08SCS11	1	2008	Dr.G.V.P
2	08SCS12	1	2008	Dr.G.V.P
3	08SCS13	1	2008	KNS
4	08SCS14	1	2008	Dr.S.R.K
5	08SCS151	1	2008	GRP
6	08SCS27	3	2008	Dr.G.V.P

6. Ex: GRADE_REPORT file has three attributes

Table 4. GRADE_REPORT

Register no	Section no	Grade
001	1	A
004	2	B
001	2	C

CHARACTERISTICS OF DATABASE APPROACH

1. Single Repository of Data(SRD)

a. A place where the data is stored
b. A place where multiple databases or files are located for distribution

2. Self-describing
a. Catalog which contains information such as structure of the file,
b. Storage format and constraints
3. Multiple views of same data
a. It allows multiple users to access the data database at same time.
4. Multi-user transaction processing
a. It allows concurrency control s/w to ensure that several users trying to update the same data
b. Ex: A seat reservation on a flight.
ACTORS ON THE SCREEN
1. **Maintenance of Dbase** (single person)
a. We need multiple people to design, development, and maintenance of a dbase

2. Database Administrator (BDA)
a. Creates a database
b. Managing the resources
c. Creation of users accounts
d. Providing security and authentication
e. System recovery if system fails
f. Manages the system poor response time

3. Database designer
a. Indentifying the data to be stored in the database
b. Planning the database design that meets all users requirements

4. End users
a. Casual users(C users)
i. There are people who use the Dbase occasionally
b. Naïve users (N users)
i. There are people who constantly querying and updating the dbase without having much knowledge abt dbase
ii. Ex: In railway, reservation clerks.
c. Sophisticated users(SO Users)
i. There are people who use the dbase for complex requirements
ii. Ex: Engineers, Scientists, business analysts
d. Standalone users(ST users)
i. There are people who maintenance the dbase for personal use
ii. Ex: http://dr.g.varaprasad.googlepages.com

System analyst
1. Determine user's requirements (N users)
2. Application programmers (S/w engineer)
3. Implement specifications as programs then they will test, debug, document, maintain

Workers behind the Scene
1. There are many people who are involved in D&D&O of Dbase
2. DBMS system designers and implementers
a. There are persons who design and implement DBMS modules
3. Tool developers
a. There are persons who work for S/W tools to enhance the performance of a DBMS package
4. Operator and maintenance personnel
a. There are people who are responsible for the actual running and maintenance of h/w and s/w environments

Advantages of using DBMS
1. Controlling redundancy
a. Duplication effort
b. Wastage of storage space

c. Inconsistency in data

2. Restricting unauthorized access
a. Some users will not be authorized to access all information in Dbase (Read the data but not update the dbase)
b. Ex: Financial data is often considered confidential

3. Good UI
a. DBMS should provide a menu driven S/W so that user can access the data without remembering(commands)

4. Providing multiple user interfaces
a. Provides a variety of users interfaces

5. System should support various types of users with varying knowledge
a. Query language interface for casual users
b. Programming language interface for application programmers
c. Formal and command interfaces for particular users
d. Menu-driven interface and natural language interface for standalone users

6. Representing complex relationship among data
a. Ex: Student name is Prasad in student table

7. Enforcing Integrity Constraints
a. Each record in a table should have a proper semantic relationship with a record of another table.
b. If employee works in dept no.5, then there must be dept table in which has dept no.5.

8. **Providing backup and recovery**
a. It should provide facilities for recovering data from s/w and h/w failures

A Brief History of Database Applications
1. We can use Dbase at different places as follows
a. Railway/Flight reservation systems
b. Universities
c. Banking systems (Credit card systems)
d. Online purchasing systems

NOT TO USE A DBMS
1. There are few disadvantages of using DBMS as follows
a. High level investment in h/w and s/w and training
b. Too complex to use and maintain as this is a large piece of s/w
c. Overhead for providing security, concurrency control, recovery and integrity functions
2. Don’t use DBMS
a. The application is simple
b. Not going to change often
c. Multiple-user access the data is not required
d. Stringent Real time requirements for some programs

Instance
1. Collection of information in dbase at a particular movement is called instance of dbase
2. Dbase changes over time as information is inserted or deleted
3. Ex: STUDENT relation

Table 5. STUDENT relation

Student name	Register no	Semester	Major
Varaprasad	001	1	CSE
Prasad	004	2	ECE

Schema

1. Overall design of dbase is called schema
2. Ex: Student database
3. STUDENT relation

Table 6. STUDENT relation

Student name	Register no	Semester	Major
Varaprasad	001	1	CSE
Prasad	004	2	ECE

Ex: COURSE relation

Table 7. COURSE relation

Course name	Course no	Credit	Dept
OS	08SCS11	3	CSE
DS&A	08SCS12	4	CSE
DBMS	08SCS13	4	CSE
CSPA	08SCS14	4	CSE
TFCS	08SCS151	3	CSE

Ex: SECTION file

Table 8. SECTION file

Section no	Course no	Semester	Year	Instructor
1	08SCS11	1	2008	Dr.G.V.P
2	08SCS12	1	2008	Dr.G.V.P
3	08SCS13	1	2008	KNS
4	08SCS14	1	2008	Dr.S.R.K
5	08SCS151	1	2008	GRP
6	08SCS27	3	2008	Dr.G.V.P

Ex: GRADE_REPORT file

Table 9. SECTION file

Register no	Section no	Grade
001	1	A
004	2	B
001	2	C

Data base state:

1. Empty state
2. Initial state
3. current state
4. Final state

Database System Utilities

It helps DBA to mange the dbase

1. Loading utility
1. is used to load the existing data file into a dbase
2. Ex: Text file
2. Backup utility
1. Creates a backup copy of dbase
2. if failure occurs, we can use backup
3. Make more reliability
3. File reorganization utility
1. We use file reorganization to store the data
4. Performance utility
1. Monitors the dbase usage & provides static information to DBA

Classification of Dbase Management System

1. First, we differentiate DBMS based on the data models
2. Presently, there are two types of database models used in commercial dbase

a. Relational data model
b. Object data model

3. Secondly, classification of DBMS is based on number of users supported by the system
a. Single user system supports for single user
b. Multi-user system supports for multiple users
4. Thirdly, number of sites over which dbase in distributed
5. Fourth, cost of DBMS

CHAPTER 2: ENTITY-RELATIONSHIP MODEL

ENTITIES AND ATTRIBUTES

Attributes

1. Represents the properties of an entity
2. Ex: SSN, DOB, Name, Sal, etc.

Entity

1. It is anything that exists in real world
2. Have multiple attributes
3. EX: Employee relation

Figure 1. Employee relation.

Entity set

1. Collection of similar entities with same properties
2. Ex: COMPANY, EMPLOYEE, TREE

Attributes types

Single attribute or atomic attribute

1. Can't be subdivided further
2. Ex: in employee relation, SSN, Age, sex, sal

Composite attribute(C attribute)

1. Can be subdivided further
2. Ex: in Employee relation, address

Figure 2. Address attribute

Single value attribute

1. Has single value for a particular entity set
2. Ex: In Employees relation, Age, Dept no

Multi-value attribute

1. Can be divisible
2. Ex: Degree(B.E, B.Tech, M.E, M.Tech, PhD, DSc)

Address
H.NO Cros Main Street Area
City State County Pin code
EMPLOYEE
SSN DOB Name Sal
FN Min LN
Add

3. Ex: Color(Red, Green, Yellow)

Stored attribute

1. The value of certain attributes can not be obtained/derived from other attributes
2. Ex: In Employees relation, SSN is stored attribute
3. Ex: In student relation, Reg no is stored attribute

Derived attribute

1. The value of attribute can be derived from other attributes
2. Ex: In Employees relation, age

Key

1. A key is a minimal set of attributes of an entity set, which would uniquely identify an entity in an entity set.
2. Ex: In Employee relation, SSN is primary key

Figure 3. Employee relation

3. Ex: Student

Figure 4. Student relation.

Domain of attributes

1. Each attribute is associated with set of values
2. Ex: Name : - the domain is [a-z], [A-Z]
3. Ex: number: Domain is [1-17] used in register number for M.Tech(CSE)

E-R diagram

1. Strong entity set
1. Ex: Employee
2.

Figure 5. Strong entity.

2. Weak entity set
1. Ex: Dependents
2.

Figure 6. Weak entity.

EMPLOYEE
SSN DOB Name Sal
FN MIN LN
Add
STUDENT
Name Sem Major RegN
Employee
Dependents
Attribute
1. Ex: RegN
2.
4. Multi-valued attribute
1. Ex: Degree
2.
5. Derived attribute
1. Ex: Age
2.
Primary key
1. Ex: SSN
2.
Relationship
1. Ex: Deposit
2.
Participation
1. Partial Participation
2. _________
3. Total portion
4.

Relationship

1. It is an association among several entities
2. A relationship set is a mathematical relational among entities n=2
1. WorksFor
2. Ex:

WORKSFOR
Age
RegN
SSN
EMPLOYEE
SSN Name
Add
DEPARTM
ENT
Dno DNam
DAdd Dadd
3. Manages

4. Ex:
MANAGES
5. If the no.of participating entity sets is two, then we called as binary relationship
6. Ex: Binary relationship
MANAGES
7. If the no.of participating entity sets is three, then we called as ternary relationship(TR)
8. Ex:
SUPPLY
9. Ex:
USES
10. Ex:
MANAGES
PROJECTS
PNam
EMPLOYEE
SSN Name
Add
Date Name
DEPARTMENT
Dno
DAdd Dadd
PROJECT
PN PNam
EMPLOYEE
SSN Name
Add
MACHINE
S
Mno MName
Mcost
SUPPLIER
SSN Name
Add
PARTS
PN PNam
Pno
Plocat
EMPLOYEE DEPARTMENT
EMPLOYEE DEPARTMENT
11. ER-Diagram for employees database
Supervision
WorkFor
Supervisor Supervisee
Manages
Control
Dependence of Works_on

RELATIONSHIP

1:1 Relation
Ex: one person has one passport
1:N Relation
Ex: one student registers many subjects
N:1 Relation

Ex: Many students have joined with a College
N:N Relation
Ex: Many employees works for many projects

TOTAL PARTICIPATION

Total Participation of Dept in Manges(Relation)
MANAGES
EMPLOYE
E
Dept
FN MM
LN
SSN Name
Add
Project
Pno PNam
Ploct
Hours
Startdate
Dno DNam
DAdd Dadd
Name Sex Relationship
DOB
Dependent
EMPLOYEE DEPARTMENT

RECURSIVE RELATIONSHIP SET (RR SET)

1. Where there is a relationship set which connects the same entity set is called RR set

supervisor Supervision
supervisee
HoD supervisees many employees

WEAK ENTITY SET

1. Entity set
2. Doesn't have a primary
3. Ex:

LOAN PAYMENT

REFINING THE ER-DRAM FOR COMPANY DBASE

1. Identify the entities
2. Identify the strong entities and weak entities

a. Employees Strong entity
b. Department Weak entity
c. Project Strong entity
d. Dependent Weak entity

3. Identify the relevant attributes

a. Employee, SSN Name, add, sex, sal
b. Department, Dno, dname, Dlocation
c. Project , Pno, Pname, Plocation
d. Dependent, Dename, DoB, Sex, Relationship

4. Identify the relationship sets

a. Employee , Department, WorkFor
b. Employee , Department, manages
c. Department, Project, Controls
d. Project, Employee, WorkOn
e. Employee , Department, Dependent_Off
f. Employee, Employee, supervision

5. Identify the cardinality ratio & participation constraints

a. Works_For N 1

b. Works_On N N
c. Manages 1 N
d. Dependent_Off 1 N
e. Controls 1 N
f. Supervision
EMPLOYEE
Pno
LOAN
Lno Type
Amou
Pdate
P-amount
PAYMENT
6. ER-Diagram for employees database
Supervision
WORKFOR
Supervisor Supervisee
MANAGES
CONTROL
DEPENDENCE_OF WORKS_ON
7. Write ER diagram of a company that keeps track of company and employee
phone
WORKS_IN
Contains
Has
EMPLOYEE
FN MM
LN
DEPARTMENT
Dno DNam
SSN Name
Add
PROJECT
Pno PNam
Ploct
Name Sex
Hours
Startdate
DAdd Dadd
Relationship
DOB
DEPENDENT
SSN Name
Add
EMPLOYEE
DNam
DEPARTMENT
Dno
DAdd Dadd
PN Model
Add
PHONE

8. Write ER diagram for a database that keeps track of text books used in course
TEACHES
REFEREES USES
SSN Name
Add
INSTRUCTOR
CN Name
COURSE
BN Name
Author
TEXT BOOK
Edition

CHAPER 3. RELATIONAL MODEL AND RELATIONAL ALGEBRA

RELATIONAL MODEL

1. Relational model represents the dbase as collection of records
2. Ex: University dbase
3. Student table
Student
name
Register
no
addr Phone DoB Sem Perc
Varaprasad 001 N.S.Halli 5567895 22/9/79 1 70
Prasad 004 DS. Halli 5567897 10/6/65 2 60
4. Student table is called Student relation
5. In student relation, row is called a tuple
6. In student relation, column header is called an attribute

DOMAIN, ATTRIBUTES, TUPLES, AND RELATIONS

1. A domain ‘D’ is a set of atomic values
a. Ex: in student relation
b. Reg no: 10 alphanumeric characters
i. Ex: 1BM99CS015
c. Name : characters
i. Ex: Varaprasad
d. Address
i. alphanumeric characters
ii. Ex: 456, N.S. Halli, Bangalore-19
e. Phone
i. 7 digits
ii. 5567895
f. DOB
i. Date
ii. Ex: 22/9/79
g. Sem
i. 1 digit
ii. Ex: 4
h. Perc
i. Float{0-100}
ii. Ex: 70
2. A Relation schema R, denoted by R(A1, A2,….An) is made up of a relation name R and a list of attributes (A1, A2,….An).
3. Degree of relation is number of attributes in a R
a. Ex: in student relation(R), degree is 7

CHARACTERISTIC OF REMATISTICS OF RELATIONS

1. Ordering of tuples in a Relation(R)

a. R is defined as a set of tuples
b. Tuples in R do not have any particular order
c. In file, records are physically stored on disk so there is an order among the records
d. Many logical models can bed used to sort records in R using SSN, name, age
2. Ordering of values within a tuple
a. At logical level, the order of attributes, their values are not important
b. Ex: A tuple can be considered as set of (<attribute>, <value>) pair
3. Values in the tuples
a. Each value in a tuple is an atomic value.
b. Multi-value attributes must be represented by separated relation
c. Composite attributes are represented only their simple component attributes
d. Null
i. In employee relation, some employees do not have phone

RELATIONAL CONSTRAINTS AND RELATIONAL DBASE SCHEMAS

1. Domain constraints
a. Specifies the value of each attribute
b. That must be atomic value from domain in R
c. Ex: in student relation, name
i. Name domain: domain is [a-z], [A-Z]
ii. Reg no domain: domain is [1-17]
2. Key constraints and constraints on null
a. A relation R is defined as a collection of tuples
i. Ex: in employee relation, Varaprasad, SRK, Indramma, Prasad
b. All records are distinct
c. All records must have a key to identify the records uniquely
i. Ex: in employee relation, SSN
d. Super key
i. Specifies uniqueness that no two distinct tuples in a state r of R can
have same value
ii. Has redundant attributes
iii. In student relation, {Reg no, name, age} is a super key but it is not key of student
e. Candidate key
i. In R, it can't have same two tuples with same value
ii. In some case, subset of set of attributes of relation has uniqueness property
iii. Ex: in VEHICLE relation, Lice No and Engine No are candidate keys
Lice No Engine
No
Make Model Year
Ka-04-
EM7244
A68967 Hond Splendor + 2007
Ka-03-
M7244
B67856 Tata Nono 2008
iv. Lice no is considered as a primary key
f. Primary key
i. Specifies uniqueness that no two distinct tuples in a state r of R
ii. EX: Student relation, Register no
Student
name
Register no
addr Phone DoB Sem
Perc
Varaprasad
001 N.S.Halli
5567895
22/9/79
1 70

Prasad 004 DS.
Halli
5567897
10/6/65
2 60
iii. It should not have null value
1. Ex: Reg no : null

RELATIONAL DATABASES AND RELATIONAL DATABASE SCHEMAS

1. So far we have discussed single relation and single schema
2. A relation dbase schema ‘S’ has many relations with tuples
3. Represents as S={R1, R2, R3,…….Rn}
a. Ex: company database
b. EMPLOYEE relation
FN MN LN SSN DoB Add Sex Sal SuperSSN DN
c. DEPARTMENT relation
DN DName MGRSSN MGRSTDATE
d. DEPT-LOCATION relation
DN D-Location
e. PROJECT relation
PN PName P-Location DN
f. WORKS_ON
SSN PN Hours
g. DEPENDENT
ESSN Dependent_Name Sex BOD Relation
4. Foreign key
a. It is a set of attributes of a relation, say, R2 whose values are required to mach values of some primary key of some other relation R1.
b. Need to maintain contain the data consistency
c. In Employee relation, SuperSSN, DN
FN MN LN SSN DoB Add Sex Sal SuperSSN DN
5. Referential integrity
a. A tuple in one relation should reference to an existing tuple in another relation
b. Is used to maintain consistency among tuples of two relations
c. Ex:Company database with pk and fk
Relation Primary key Foreign key Referencing relation
SSN SuperSSN Employee(SSN) EMPLOYEE
DNo Department (DN)
DEPARTMENT DN MGRSSN Employee(SSN)
DN DN Department (DN) DEPTLOCATION
D-Location
PROJECT PN DN Department (DN)
ESSN ESSN Employee(SSN) WORKS_ON
PN PN Project (PN)
DEPENDENT ESSN ESSN Employee(SSN)
Dependent_Name
d. EMPLOYEE relation
FN MN LN SSN DoB Add Sex Sal SuperSSN DN
e. DEPARTMENT relation
DN DName MGRSSN MGRSTDATE
f. DEPT-LOCATION relation
DN D-Location
g. PROJECT relation
PN PName P-Location DN
h. WORKS_ON
SSN PN Hours
i. DEPENDENT
ESSN Dependent_Name Sex BOD Relation

UPDATE OPERATIONS AND DEALING WITH CONSTRAINTS VIOLATIONS

1. Operations of the relational model can be categorized into retrievals and updates.

2. Three basic operations in RM

a. Insert

i. Is used to insert new record into a relation

ii. We want insert new employee in a relation.

FN MN LN SSN DoB Add Sex Sal SuperSSN DN

iii. Ex: We want insert tuple whose name is Jerge W. Buss in employee relation

iv. insert<’Jerge’ , ‘W’, ‘Buss’, ‘766669999’, ‘1960-05-06’, ’53 DS Lane, NS’, ‘M’, ‘20000’, ‘766669969’, ‘5’> ----- is possible

FN	MN	LN	SSN	DoB	Add	Sex	Sal	SuperSSN	DN
Jerge	**W**	**Buss**	**766669999**	**1960-05-06**	**’53 DS Lane, NS**	**M**	**20000**	**766669969**	**5**

v. Ex: We want insert tuple whose name is Krishma R. Murthy in employee relation

vi. insert<’Krishma’ , ‘R’, ‘Murthy’, ‘766669998’, ‘1940-04-03’, ’23 DS Lane, LS’, ‘M’, ‘40000’, ‘766669969’, ‘5’>----- is possible

FN	MN	LN	SSN	DoB	Add	Sex	Sal	SuperSSN	DN
Jerge	W	Buss	766669999	1960-05-06	’53 DS Lane, NS	M	20000	766669969	5
Krishma	**R**	**Murthy**	**766669998**	**1940-04-03**	**’23 DS Lane, LS**	**M**	**40000**	**766669969**	**5**

vii. Ex: We want insert tuple whose name is Jerge J. Buss with SSN is null in employee relation

viii. Ex: insert<’Jerge’ , ‘J’, ‘Buss’, Null, ‘1980-07-06’, ’53 DS Lane, NS’, ‘M’, ‘20000’, ‘766669969’, ‘5’>---------- is not possible SSN is null

ix. Ex: We want insert tuple whose name is Jerge J. Buss in DN 100000 in employee relation

x. insert<’Jerge’ , ‘J’, ‘Buss’, 6767669998, ‘1980-07-06’, ’53 DS Lane, NS’, ‘M’, ‘20000’, ‘766669969’, ‘100000’>---------- not possible DN 100000’ is not there in dept relation

xi. Ex: insert<’Senior’, ‘W’, ‘Buss’, ‘766669999’, ‘1944-04-05’, ’43 NS Lane, LS’, ‘M’, ‘54000’, ‘766669969’, ‘5’>----- is not possible(previous record had same SSN)

xii. Insert new record in WORKS_ON relation, insert <’766669999’, ‘5’, ‘6’>---- is possible

SSN	PN	Hours
766669999	**5**	**6**

xiii. Insert new record in WORKS_ON relation, insert <’766669998’, ‘5’, ‘7’>---- is possible

SSN	PN	Hours
766669999	5	6
766669998	**5**	**7**

b. Delete

i. Is used to delete record a record from relation

ii. Ex: Delete the WORKS_ON tuple with SSN=’ 766669999’, and PN=’5’ (possible since this record is available in dbase)

SSN PN Hours
766669998
iii. Ex: Delete the EMPLOYEE tuple with SSN=’ 766669998’ is not
possible (since WORKS_ON refer to this tuple)
FN MN LN SSN DoB Add Sex Sal SuperSSN DN
766669999
c. Update
i. Is used to modify the values of some attributes in a relation
ii. Ex: Update the sal of the EMPLOYEE tuple with SSN=‘766669999’
to 32000 ------(Accepted)
FN M
N
LN SSN DoB Add Se
x
Sal SuperSSN D
N
Jerge W Buss 7666699
99
196
0-
05-
06
’53
DS
Lan
e,
NS
M 3200
0
7666699
69
5
Krish
ma
R Murt
hy
76666999
8
1940
-04-
03
’23
DS
Lane
, LS
M 4000
0
76666996
9
5
iii. Update the DN of the EMPLOYEE tuple with SSN=‘766669999’ to 1
------(Accepted)
FN M
N
LN SSN DoB Add Se
x
Sal SuperSSN D
N
Jerg

e
W Bus
s
7666699
99
1960
-05-
06
’53
DS
Lan
e,
NS
M 3200
0
7666699
69
1

iv. Update the SSN of the EMPLOYEE tuple with SSN=‘766669998’ to 766669999 ------(not accepted, since 766669999 already there in dbase)

BASIC RELATIONAL ALGEBRA OPERATIONS

1. Basic set of relational model operations constitute the relational algebra
2. Enables the user to retrieval requests
3. Set operations are to group

a. First
i. Select
ii. Project
iii. Join
b. Second
i. Union
ii. Intersection
iii. Set Difference
iv. Cartesian product

4. Select operation

a. Used to select a subset of the tuples from a relation that satisfy a condition
b. Syntax is Ó
c. Ex: Select the tuples(employees) who work in dept no 4, or those whose salary grater than Rs.30,000 P.M

i. Ódno=4(EMPLOYEE)

ii. Ósal>30000(EMPLOYEE)

d. Ó(dno=4 and sal>30000)(EMPLOYEE)

e. Ex: Select the tuples who either work in dept no 4 or make over Rs.25000 P.Mor, those who’s salary greater than 30,000 P.M, or work in dept no 5.

i. Ódno=4(EMPLOYEE)

ii. Ósal>25000(EMPLOYEE)

iii. Ódno=4(EMPLOYEE)

iv. Ósal>30000(EMPLOYEE)

v. ó(dno=4 and sal>25000) OR (dno=5 and sal>30000)(EMPLOYEE)

f. Boolean conditions

i. AND

1. (cond 1 AND cond 2) is true if both cond 1 AND cond 2 are true, otherwise, it is false

ii. OR

1. (cond 1 OR cond 2) is true if either cond 1 or cond 2 or both are true, otherwise, it is false

iii. NOT

1. (NOT cond) is true if cond is false, otherwise, it is true

5. Project operation

a. Used to select a subset of the attributes of a relation by specifying the names of the required attributes.

b. Syntax is ð<Attribute list>(R)

c. For example to get a list of all employees first name and last name and sal from EMPLOYEE relation

d. ðLname, Fname, sal(EMPLOYEE)

6. Sequence of Operations and the RENAME Operation

a. We can use SELECT and PROJECT operations

b. Find those customer who live in Bangalore

c. ð Customer-name(óCustomer-city =Bangalore) (Customer)

d. Ex: Retrieve the first name, last name and sal of employees who work in department 5,

e. ðLname, Fname, sal(ó(dno=5(EMPLOYEE)))

f. We can write this query in another way

i. DEP5-EMP ó(dno=5(EMPLOYEE)

ii. RESULTS ðLname, Fname, sal(DEP5-EMP)

g. We can write this query in another way

i. TEMP ódno=5(EMPLOYEE)

ii. F(FIRSTNAME, LASTNAME, SAL) ðLname, Fname, sal(TEMP)

7. Set Theoretical Operations

a. Consider two relations R and S.

b. UNION of R and S

i. The union of two relations is a relation that includes all the tuples that are either in R or in S or in both R and S.

ii. Duplicate tuples are eliminated.

iii. Ex: Retrieve the SSNs of all employees who either work in dept 5 or directly supervise an employee who works in dept 5

iv. In this section, we can query using UNION operation

v. DEP5-EMP ó(dno=5(EMPLOYEE)

vi. RESULT1 ðSSN(DEP5-EMP)

vii. RESULT2 ðSUPERSSN(DEP5-EMP)

viii. RESULT RESULT1 U RESULT2

c. Student relation

Fist name	**Last name**
GV	P
RV	L
MV	S
HT	L

Instructor

Fist name	**Last name**
LD	R
RV	**L**

HV T

HT L

Student U Instructor

Fist name	**Last name**
GV	P
RV	L
MV	S
HT	L
LD	R
HV	T

d. INTERSECTION of R and S

i. The intersection of R and S is a relation that includes all tuples that are both in R and S.

ii. Student Ð Instructor

Fist name	**Last name**
RV	L
HT	L

iii. Find all customers who have both loan and an accont

iv. ðCustomer-name(BORROWER)Ð ðCustomer-name(DEPOSIT)

e. DIFFERENCE of R and S

i. The difference of R and S is the relation that contains all the tuples that are in R but that are not in S.

ii. Ex: R-S or S-R

iii. Student – Instructor

Fist name	**Last name**
GV	P
MV	S

iv. Instructor - Student

Fist name	**Last name**
LD	R
HV	T

8. Cartesian Product

a. Is binary operator which combines both relations

b. Denoted by X

c. Works on two sets

d. It is sometimes called CROSS JOIN.

e. S(A1, A2, …….Am) X R(B1, B2, ……Bn)= m+n attributes

DEPARTMENT Relation

DNo	Dame	MGRSSN
1	CSE	40
2	ISE	56
3	MCA	44

PROJECT Relation

PName	Plocation
DS	Bangalore
DBMS	Chennai

f. PROJECT Relation X DEPARTMENT Relation

DNo	Dame	MGRSSN	PName	Plocation
1	CSE	40	DS	Bangalore
1	CSE	40	DBMS	Chennai
2	ISE	56	DS	Bangalore
2	ISE	56	DBMS	Chennai
3	MCA	44	DS	Bangalore
3	MCA	44	DBMS	Chennai

g. We want to retrieve for each female employee a list of the names of her dependents

i. FEMALE_EMP Ósex=’F’(EMPLOYEE)

ii. EMPNAMES ðLname, Fname, sal(FEMALE_EMP)

iii. EMP_DEPENDENTS EMPNAMES X DEPENDENTS

iv. ACTUAL_DEPENDENTS ÓSSN=ESSN(EMP_DEPENDENTS)

v. RESULT ðLname, Fname, Dependent_name(ACTUAL_DEPENDENTS)

9. Join Operation
a. Is used to combine the related tuples from two relations into single tuples
b. Do cross product of the two relations(improved version of CP).
c. Becomes more complex, tuples are removed within the cross product to make the result of the join more meaningful.
d. Syntax is X
e. Join operations are
i. Natural Join
ii. Theta join
iii. Equijoin join
iv. Outer Join
f. Natural Join
i. When we omit the condition during joining we call it is natural joining
ii. Let S and R are two relation, they have attribute like S(A1, A2, …….Am, C1, C2……….Ck) and R(C1, C2……….Cl, B1, B2, ……Bn)
iii. C1, C2……….Cl are common attributes
iv. We will use simple of *
v. Ex: Department

DNo	Dame
1	CSE
2	ISE
3	MCA

PROJECT Relation

PNo	PName	DNo
10	DS	2
20	DBMS	2
30	TFCS	1
40	OS	3

vi. Natural join of Department and PROJECT is

PNo	PName	DNo	Dame
10	DS	2	ISE
20	DBMS	2	ISE
30	TFCS	1	CSE
40	OS	3	MCA

vii. Here, joining is done over the attribute of DNo of Department and DNo of PROJECT
viii. In PROJECT Relation, DNo is FK
g. Theta join
i. Used joining of two relations
ii. is similar like Cartesian Product
iii. Here, we will use condition
iv. Let R and S are relations
v. If x is attribute of R and y is attribute of S, then RXS where xèy
vi. Where, è is .,=,>,=, etc..
h. Equijoin join
i. Let R and S are relations
ii. Syntax is R X <joint_condtion> S
iii. We want to retrieve the name of manager of each department.
iv. DEPT_MGR DEPARTMENT X MGRSSN=SSN(EMPLOYEE)

v. RESULT ðDname, Fname, Lname(DEPT_MGR)
i. Outer Join
i. Is similar like CP and Equijoin join
ii. Tuples of a relation don’t participate in join is called dangling tuples
iii. Outer join operations are

1. Left outer join
2. Right outer join
3. Full outer join

iv. Left outer join
1. Similar to natural join
2. but keep all dangling tuples of R
3. Syntax is R X S

STUDENT Relation

Regno	Name	%	Bracnh
1BMCS05	Ravi	89	CSE
1BMME02	Prasad	92	MEC
1BMIS15	Shankar	86	IS
IBMCV24	Siva	98	CV

AWARD Relation

Regno	Amount	Year
1BMCS05	1000	2007
1BMME02	2000	92
BMCV24	1500	98

RESULT STUDENT X AWARD

Regno	Name	%	Bracnh	Amount	Year
1BMCS05	Ravi	89	CSE	1000	2007
1BMME02	Prasad	92	MEC	2000	92
1BMIS15	Shankar	86	IS	Null	Null
IBMCV24	Siva	98	CV	1500	98

v. Right outer join
1. Similar to natural join
2. but keep all dangling tuples of S
3. Syntax is R X S

vi. Full outer join
1. Similar to natural join
2. but keep all dangling tuples of both R and S
3. Syntax is R X S

10. Division Operation
a. Is kind of query operation
b. Ex: Retrieve the names of employees who work on all projects that john smith works_on

c. SMITH óFanme ='John' and Lname='Smith'$_{\text{(EMPLOYEE)}}$

d. SMITH_PNO ðPNo(WORKS_ON X ESSN=SSN$_{\text{SMITH}}$)

e. SSN_PNOS ðESSN, PNo($_{\text{WORKS\_ON}}$)
f. SSNS SSN_PNOS ÷ SMITH_PNO

g. RESULT ðFname, Lname$_{\text{(SSNS*EMPLOYEE)}}$

11. Grouping and Aggregation function
a. All relational algebra operations are used to represent as relation consisting of rows and columns
b. Aggregation functions are
i. Sum, Age, Count, Maximum, Minimum
ii. Group_attributes æl Aggregate_functions(Relation)
c. Retrieve dept no, no.of employees and their average salary
i. RESULT(Dno, N, Avgsal) Dno ælcount(SSN), Avg(sal)$_{\text{(EMPLOYEE)}}$
d. Participation the relation into groups
e. Apply aggregate function to each group
f. Retrieve dept no, no.of employees and their average age
i. RESULT(Dno, N, Avgsal) Dno ælcount(SSN), Avg(age)$_{\text{(EMPLOYEE)}}$

Examples of queries in Relational Algebra

g. Retrieve the name and address of all employee who work for CSE dept

CSE_DEPT ÓDname=’CSE’(Dept)
CSE_EMP (CSE_DEPT X Dno=Dno(EMPLOYEE)

RESULT ðFname, Lname(CSE_EMP)

12. For every project located in Stanford, list the projects numbers, the controlling, and dept managers, last name, address, birth of date.

STANFFORD_PROJ ÓPlocation=’ Stanford’

CONTROL_DEPT STANFFORD_PROJ X Dno=Dno(Dept)

PROJ_DEPT_MGR CONTROL_DEPT X MGRSSN=SSN(EMPLOYEE)

RESULT ðPno, Dno, Fname, Lname, add, bod(PROJ_DEPT_MGR)

13. Find the name of employees who work on all projects controlled by dept 5.

a. DEPT5_PROJ(Pno) ðPno(ÓDno=’5’(PROJECT)

b. EMP_PROJ(SSN, Pno) ðEssn, Pno(Works_On)

c. RESULT_EMP_SSNS EMP_PROJ ÷ DEPT5_PROJ

d. RESULT ðFname, Lname(RESULT_EMP_SSNS* EMPLOYEE)

14. Make list of project numbers that involves an employee who last name is Smith, either as a worker or a manager of dept that controls the department.

SMITH(ESSN) ðssn(ÓLname=’Smith’ (EMPLOYEE)

SMITH_WORKER_PROJ ðPno(WORKS_ON * SMITH)

15. List the name of managers who have atleat one dependent.

MGRS ðMGRSSN(Department) //Find MGRSSN’s

EMP_With_Deps(ssn), ðESSN(Dependent)

MGRS_With_Deps (MGRS Ï EMP_With_Deps)

Result ðFname, Lname(MGRS_With_Deps *Employee)

ER To Relational Mapping Rules (ER to Table)

1. ER diagram it doesn’t mean that logical dbase design is over all(constraints and proper set of relations).
2. Strong entity to table

Rule 1:

a. For every strong entity E, creates a table R that includes all attributes like primary key, etc.

b. Employee table

SSN Name Sal Add DoB

3. Rule 2: Weak entity to table

a. How to identify weak entity

i. Doesn’t have primary key

ii. Has partial key

iii. Participates as one-to-many relations

iv. If the participating owner entity is deleted, then all its references must be deleted.

b. Convert weak entity to table

EMPLOYEE

SSN DOB Name Sal Add

c. Rule 3: Mapping rule for ER relationship sets.

ii. Entity sets are related with binary relationship

iii. Relationship sets are WorksFor, Manages, Dependents, WorksOn, Control, Supervision.

1. Employee Workfor Department
2. Employee Manages Department
3. Employee Dependentsof Dependents
4. Employee WorksOn Project

5. Department Control Project
6. Employee Supervision Employee
iv. One-to-One cardinality
1. r is attribute in Employee full participation
v. One-to-Many
vi. Many-to-Many cardinality
1. Translating M-to-M relationship set to table
2. Creates a new table with relationship set name
3. A&B are relations, which contain primary keys of t and s to be composite P.K for new relation.
a. Rule 4: Handling Multi-valued attribute
4. Each multi-valued attribute should be translated into table.
a. Rule 5: Ternary relationship set to table
5. You want create a table with P.K's of all three participating entities

Correspondence between ER & Relation model

ER model Relational model
1. Entity type 1. Entity relation
2. 1:1 or 1:N type 2. Foreign key
Dependents
dname Sex Address
3. M:N relationship type 3. "Relationship" relation and two F.K's.
4. N-array relationship type 4. Relationship relation and *n* F.K's.
5. Composite attribute 5. Set of simple component attributes
6. Value set 6. Domain
7. Key attribute 7. P.K or S.K

Introduction to client/Server Architecture

1. No. of applications are executed in client/server environment
2. Server has more powerful than clients to keep the data
3. Client sends request to server using its IP address and port number
4. Based on request, the server response to the client
5. Advantages
a. Centralized resource
i. To avoid problem caused by redundant and inconsistent data
b. Improve security
i. It provides more security
c. Server level administration
i. As client doesn't play major role
d. Scale network
i. We can add new user or delete clients
6. Disadvantage
a. Increased cost due to the technical complexity of the server
b. Weak link(if server fails, network will be disconnected
7. Centralize Architecture
a. Network management platform resides on a single computer
i. Ex: E-mail, banking
b. Used in commercial applications, network alerts and events
c. Pros:
i. Single location to view events and alerts, more security
d. Cons:
i. Single system is not redundant or fault tolerant
ii. More queue if system has more number of users.
iii. More accessing time if system has more number of users.

CHAPER 4. STRUCTURE QUERY LANGUAGE (SQL)

Introduction about SQL

1. It is used to represent a relational database,
2. Table is used to represent for relation.
a. Rows for tuples.
i. Ex: Records

b. Columns for attributes.
i. Ex: SSN, name, dept-no
3. SQL2 commands for data definition such as Create, Alter, and Drop.
4. CREATE command is used to create/specify a new relation.
a. Ex: CREATE TABLE COMPANY EMMPLOYEE.
5. DROP command is used to remove schema in a Dbase.
a. The options are CASCADE and RESTRICT.
6. The CASCADE operation is used to remove dbase schema and tables, domains, other elements.
a. DROP SCHEMA COMPANY CASCADE.
7. The RESTRICT operation is used if it has no element in it.
8. ALTER table Company is used to change dbase table.
9. ALTER TABLE COMPANY DEPARTMENT ALTER MGRSSN SET DEFAULT.

Data types and Domains

1. Numeric
Ex: SSN.
2. Character
Ex: String.
3. Date
Ex: 2008/12/20
4. Time
Ex: 10:30 AM

Basic Queries in SQL

Select<Attribute list>-> is also called mapping
From<Table list>
<Condition>

Question 1: Retrieve the birth date and address of employee whose name is John B. Smith
Select BoD, ADD
From EMPLOYEE
Where FN='JOHN' and MN='B' and LN='Smith'

Question 2: Retrieve the name and address of all the employees who work for the Research' dept
Select FN, LN, ADD,
Form EMPLOYEE, DEPARTMENT
Where DNAME =RESEARCH' and DNUMBER= DNO

Question 3: For ever product located in Stanford, list the project number, construing dept number, department manager, last name, address, DOB.
Select PNO, DNO, LN, ADD, DoB
From PROJECT, DEPARTMENT, EMPLOYEE
Where DNUM=DNO add MGRSSN=ESSN and PLOCATION='STANDFORD'

Question 4: Retrieve the all employees who work in Dept 5.
Select *
From EMPLOYEE
Where DNO=5.

Question 5: Retrieve the all employees who work in Research Dept.

```
Select *
From EMPLOYEE, DEPARTMENT
Where DNAME=’RESEARCH’ and DNO=DNUM
```

Union : It is used to combine all records

Question 6:

```
Select FN, LN
From STUDENT
Union
Select FN, LN
From TEACHER.
```

Intersect

Question 7: Let us find out names of students as well as teachers

```
Select FN, LN
From STUDENT
Intersect
Select FN, LN
From TEACHER.
```

Minus

Question 8: Let us find out the names of student who are not teacher

```
Select FN, LN
From STUDENT
Minus
Select FN, LS
From TEACHER
```

Question 9: Retrieve all employees in Dep 5 whose salary is between $ 30,000 and $40,000.

```
Select *
From EMPLOYEE
Where(Salary between 30000 and 40000) and DNO=’5’.
```

Question 10: Retrieve all the employees whose address is in HOUSTON, TEXAS

```
Select FN, LN
From EMPLOYEE
Where address like 1% HOUSTON, TEXAS%
```

Question 11: Find all the employees who were born during the 1950’s

```
Select FN, LN
From EMPLOYEE
Where BOD like --5-,--,--
```

Question 12: Retrieve the name of each employee who works in all projects Controlled by Dept
‘5’

```
Select FN, LN
From EMPLOYEE
Where ((Select PNO
From WORKS_ON
Where SSN=ESSN)
Contains(Select Pno from PROJECT where Pnum='5'))
```

Question 13: Retrieve the names of employees who name no dependents

```
Select FN, LN
From EMPLOYEE
Where not exists(Select *
From DEPENDENT
Where SSN=ESSN);
```

Aggregate Functions

1. Count, Sum, Max, Min, Avg

Question 14: Find the sum of the salaries of all the employees, the maximum salary, the minimum salary, the average salary

```
Select Sum(Salary), Max(Salary), Min(Salary), Avg(Salary)
from EMPLOYEE
```

Question 15: Find the sum of the salaries of all employee of research department, as well as maximum salary, the minimum salary, the average salary in dept

```
Select Sum(Salary), Max(Salary), Min(Salary), Avg(Salary)
From EMPLOYEE, DEPARTMENT
Where Dno=Dnum and Dname= 'Research'
```

Grouping and Aggregate Function

1. All relation algebra operations are used to represent a relation consisting of rows, columns.
2. It has Sum, Avg, Count, Maximum, Minimum functions
3. Group attributes 3 Aggregate functions Relation
4. Prediction the relation into groups
5. Apply aggregate function to each group

Question 16: Retrieve Dept no, No of employees and their average salary

Result (Dno, N, Avg Sal)   Dno 3 Count(SSN), Avg(Sal) (EMPLOYEE)

Question 17: Retrieve Depno, No of employee and their average age

Result (Dno, N, Average) Dno 3 Count(SSN), Avg(Age) (EMPLOYEE)

Examples of Queries in Relational Algebra.

Question 18: Retrieve the name and address of all the employees who work for CSE dept

CSE-Dept (Dname='CSE'(DEPARTMENT))

CSE-Emp (CSE-Dept Dno=Dno EMPLOYEE)

Result FN, Add(CSE-Emp)

Question 19: For every project located in 'Stanford', list the project number, the controlling and dept managers last name address, birth date

Standford_Project (Plocation='Stafford'(PROJECT)

Control_Dept (Stanford_Project Dno=Dno DEPARTMENT)
Project_Dept_MGR (Control_Dept MGRSSM=SSN EMPLOYEE)
Result (Pno, Dno, FN, LN, Add(Project_Dept_MGR)

Question 20: Find the name of the employees who work on all projects controlled By dept no5
Dep5-Proj Pno(Dno=5(PROJECT)) //project number
Emp-Proj ESSN, Pno(WORKS_ON)
Result_Emp_SSNS Emp_Proj /Dept5_Proj
Result FN, LN(Resul_Emp_SSNS)

Question 21: Make a list of project numbers that involve an employee whose last name is smith, either as a worker or a manager of dept that controls the project:
Smith (ESSN) SSN(LN='Smith') (EMPLOYEE)

Insert command in SQL
1. Used to add single tuple into the dbase
a. Ex: INSERT INTO EMPLOYEE
VALUES ('Richard', 'K', 'Kotti', '63332986', 1962-12-30', 98 Oak, Tack, TX', 'M', '37000', '986756999', '4');
2. We can also use to assign values for certain attributes for new tuple.
a. INSERT INTO EMPLOYEE(Fname, Lname, Dno, SSN)
VALUES ('Varaprasad', 'Golla', '4', '98653257');

Delete command in SQL
1. Used to remove tuple from a relation

Question 22: Remove employee whose name is Brown
Delete from EMPLOYEE
Where LN='Brown'

Question 23: Remove employee whose SSN is '123456789'
Delete from EMPLOYEE
Where SSN='123456789'

Question 23: Remove all employees who work in research department
Delete from EMPLOYEE
Where Dno in (Select Dnumber
From Department
Where Dname='Research')

Update Command in SQL
1. Used to modify value of attribute
2. Ex:To change the location and controlling department number of project 10 to Bangalore and 5.
a. Update Project
Set Plocation='Bangalore' and Dno='5';
Where pnumber='10'

View (Virtual table) in SQL

1. It is a single table that is derived from other tables
2. Doesn't necessarily exist in physical form(virtual table)
3. Ex: Create View Works_On1
As Select Fn, Ln, Pname, Hours
From Employee, Project, Works_On
Where SSN=ESSN AND Pno=Pnumber
4. Create View Dept_Infor(Dept_name, N0_of_emps, Total_sal)
As Select Danme, Count(*), Sum(Salary)
From Department, Employee
Where Dnumber=Dno
Group by Dname;
5. Retrieve the last name, first name, of all employees who work on 'Project X'
Select FN, LN
From Works_On1
Where Pname ='Project'

View implementation and view update

1. Query modification, modifies view query into a query.
2. View materialization, involves physical creating a temporary view table
3. Ex: Update the Pname attribute of Jhon B Smith from Product 'X' to Product 'Y'.
a. Update Works_On1
Set Pname='Product Y'
Where LN=' Smith' AND FN='Jhon' AND Pname='Product Y';

Specifying General Constraints as Ascertain

1. To specify the constraints that salary of an employee must not be grater than salary of the manager that employee works for
Create Assertion Salary_Constraint
Check (Not exists(Select * from Employee E,
Employee M, Department D
Where E.Sal>M.Sal AND
E.Dno=D.Dnumber AND
D.MGRSSN=M.SSN)

Additional Features of SQL

1. Granting and revoking privileges to the users
2. Supports insert, delete, update, and select operations
3. SQL statements support general purpose languages(C, C++, Pascal)
4. has transaction control commands for concurrency control, recovery purpose

Database program issues

1. Interface between application program and dbase
2. Records in dbase corresponds to records in a programming language should be distinct type.

CHAPTER 5. DATABASE DESIGN

1 Need of normal formal forms

1. It is a logical design method, which minimizes data redundancy and reduces design flaws.
2. Consists of applying various normal forms to do database.

3. The normal forms break down large tables into smaller subsets.

First Normal formal form (INF)

Def: A relation(R) is said to be in 1NF if every attribute of R takes only single value(atomic value).

1. It contains no repeating attributes or groups of attributes.
2. No repeating columns within a row.
3. No multi-valued columns.
4. Ex: In employee relation,

ESSN Name Age Sex Dependent
41 Dr. Varaprasad 28 M {Dr. S.R. Kirhna, Dr. Babu}
42 Dr. Ravi 32 M {Dr. Summa, Shankar}

5. It is not in 1NF. Since it don’t have single value for attribute Dependent

ESSN Name Age Sex Dependent
41 Dr.Varaprasad 28 M Dr. S.R. Kirhna,
41 Dr.Varaprasad 28 Dr. Babu
42 Dr.Ravi 32 M Dr. Summa
42 Dr.Ravi 32 M Shankar

6. Ex: In department relation

Department no Department name MGRSSN Dept location
1 CSE 41 {B’lore, Mysore}
2 ISE 42 {B’lore, Pune}

7. It is not in 1NF. Since it don’t have single value for attribute Dept location

Department no Department name MGRSSN Dept location
1 CSE 41 B’lore
1 CSE 41 Mysore
2 ISE 42 B’lore
2 ISE 42 Pune

Functional Dependency

Def: In relation, X Y is full functional dependency(FD), if removal of attribute X, then it does not hold any more relation.

ESSN Name // if removal of attribute ESSN doesn’t hold a relation
{ESSN, PNo} Hours// if removal of attribute ESSN, PNo doesn’t hold a relation

Partial Functional Dependency

Def: In relation, if removal of any attribute, then it does hold still a relation.

Ex:{ESSN, Name} Name, if you remove of attribute, Name, it still holds a relation

Second Normal formal form (2NF)

Def: A relation(R) is said to be in 2NF if ever non-prime attributes in R must be functionally dependent on the primary key(candidate key).

EMPPROJECT relation

SSN PNo Hours Ename Pname Plocation

1. FD-1: SSN Ename.
2. FD-2: PNo {Pname, Plocation}
3. FD-3: {SSN, PNo} Hours

4. It satisfies 1NF but not in 2NF
5. if we remove PNo, it still hold a relation FD-1;
6. if we remove SSN, it still hold a relation FD-2;
7. Decompose EMPPROJECT relation into 3 parts
a. S-E table, PPP table and SPH table
8. S-E table

SSN EName

----

9. PPP table

Pno Pname Plocation

----

10. SPH table

SSN Pno Hours

Third Normal Form(3NF)
Def: A relation schema ‘R’ is in 3NF, if it is in 2NF and no non-prime attribute of ‘R’ is
EMPDEPT relation

-

SSN Name DoB Addr Dno Dname MGRSSN

--

1. FD-1: SSN {Name, DoB, Addr, Dno}
2. FD-2: Dno {Dname, MGRSSN}
3. Here, Dno is non-prime attribute(violate 3NF)
4. .EMPDEPT relation is decomposed into employee and department.
5. Employee relation

SSN Name DoB Addr Dno

----

6. Department relation

Dno Dname MGRSSN

Boyce-Codd Normal Form(BCNF)
Def: A Relation is in 3NF, if and only if every deterministic is a table is candidate key.
Relation table

A B C D

----

1. FD has four attributes
2. Since FD-2 is violating BCNF
3. Spilt FD relation into R1 and R2
a. R1(A,C, D) with FD AC D; R2(C, B) with C B
4. R1 table

A C D

----

5. R2 table

C D

CHAPTER 6. TRANSACTION MANAGEMENT

1. Transaction is a logical unit of dbase

2. Read operation : It read the data item in data base but cant not update the value in dbase

T1 T2
Read_Item(X);
X:=X-N;
Read_Item(X);
X:=X+N;

3. Write operation: It read the data item as well as update the item in data base.

T1 T2
Read_Item(X);
X:=X-N;
Write_Item(X);
Read_Item(X);
X:=X+N;
Write_Item(X);

4. Single user: Only one user can access the Dbase at time

a. Ex: DoS is Single user OS.

5. Multi-user: Many users can access the Dbase currently

a. Ex: Window XP is multi-user OS.

6. Multiprogramming : which allows computer to execute multiple programs

7. Interleaved:

a. It allows system to execute one task for period of time then it suspends the task and executes other tasks.

b. Reduces the cost

c. Increases the delay

d. Ex: We have one process and have to do three tasks

Process A will do all three tasks

T1 T2 T3
Read_Item(X);
X:=X+N;
Read_Item(X);
X:=X+2N;
Read_Itme(X)
X:=X+3N
Write_Item(X);
Write_Item(X);
Write_Item(X);

8. Parallel Processing:

a. It has multiple processes to execute multiple tasks.

b. Reduces the delay

T
i
m
e
T
i
m
e

T
i
m
e
c. Increases cost
d. Ex: We have one process and have to three tasks
e. Here, we have multiple processes (for example 3)
Process A Process B Process C
T1 T2 T3
Read_Item(X); Read_Item(X); Read_Itme(X)
X:=X+N; X:=X+2N; X:=X+2N;
Write_Item(X); Read_Item(X); Read_Itme(X)
Why Concurrent Control Required
6. Reduces delay
7. Keeps CPU always busy with some jobs without waiting in idle.
8. Problems
a. Last update problems
b. A=25
T1 T2 Database
Read_Item(A); // A=25
A:=A+100
Read_Item(A); // A=25
A:=AX2;
Write_Item(A); //A=125
Read_item(B)
Write_Item(A); Here, A has incorrect value
because its update by T1
9. Temporary update problem
a. A=25;
b. B=50;
T1 T2 Database
Read_Item(A); // A=25
A:=A+100
Write_Item(A); // A=125
Read_Item(A); // A=125
A:=AX2;
Write_Item(A); //A=250
Read_Item(B); //B=50
Fail
//System has failed due to some problems and Dbase contains
inconsistent information(In T1, some values are updated and some values are not
updated(B))

Types of failures
1. Computer failures
i. Due to H/W, S/W, N/W during a transaction
2. Transaction failures
i. Due to integer over flow, division by zero
3. Local errors
T
i
m
e
T
i
m

e
T
i
m
e
i. You want to withdraw amount from the bank but you don't have sufficient
amount in ur account.
4. Concurrent control problems
i. Due to dead lock
5. Physical problems
i. Power, Fire, Cyclone, Earthquake, Theft, etc…

Why Recovery is needed

1. Data should be recovered when computer fails due to transaction fails, Local errors, physical problems, concurrent control problems.

Transaction states and additional operations

1. Begin state: We are about to start transaction
2. Read state: We are reading an item in a dbase
Ex: Read_item(X)
3. Write state: We are writing item in database
Ex: Write_item(X)
4. End_ Transaction state: We have finished all operations and are going to write all updates in Dbase.
5. Commit_ Transaction state: Successfully completed transaction
6. Rollback state: Transaction has failed due to some problems.

Read, write
Begin Commit
End transaction
Abort
Abort

Desirable properties of transactions

1. Transaction posses several properties
2. **Atomicity**:
A transaction is an atomic unit of processing. It is either performed a task or not.
3. **Consistency Preservation**:
A transaction is consistency preserving if it's complete of dbase from one state to another.
4. **Isolation**:
A transaction is being executed in isolation from other transactions. T1 should not interface with T2 while execution of T1.
Active Partial
committed
Committed
Failed Terminated
5. **Durability**:
The changes applied to the dbase by using committed operation must persist in a dbase.

Schedules of transactions:

1. Schedule has n transactions T1, T2, T3…..Tn.
2. Schedule has ordering operations subject to constraint.
T1 T2

Read_Item(X);
X:=X+100
Read_Item(X);
X:=X*2;
Write_Item(X);
Read_item(Y)
Write_Item(X);

3. **Sa:** r1(x), r2(x), w1(x), r1(y), w2(x);

T1 T2
Read_Item(X);
X:=X+100
Write_Item(X);
Read_Item(X);
X:=X*2;
Write_Item(X);
Read_item(Y)
Fail

4. **Sb :** r1(x), w1(x), r2(x), w2(x), r1(y, a1 ;

Serial Schedule :

1. Execute all operations of transaction T1 followed by all operations of transaction T2.

T1 T2
Read_Item(X);
X:=X-N
Write_Item(X);
Read_Item(Y);
Y:=Y+N;
Write_Item(Y);
Read_item(X)
X:=X+M;
Write_Item(X);

2. Execute all operations of transaction T2followed by all operations of transaction T1.

T1 T2
R Read_item(X)
X:=X+M;
Write_Item(X);
Read_Item(X);
X:=X-N
Write_Item(X);
Read_Item(Y);
Y:=Y+N;
Write_Item(Y);
T
i
m
e
T
i
m
e
T
i
m
e
T

i
m
e
3. We are executing transactions without interleaved operations

Non Serial Schedule

4. We are executing transactions with interleaved operations

Read_Item(X);
X:=X-N
Read_item(X)
X:=X+M;
Write_Item(X);
Read_Item(Y);
Write_Item(X);
Y:=Y+N;
Write_Item(Y);
Read_Item(X);
X:=X-N
Write_Item(X);
Read_item(X)
X:=X+M;
Write_Item(X);
Read_Item(Y);
Y:=Y+N;
Write_Item(Y);

Serializable

A Schedule ‘S’ of *n* transactions is serializable, if it is equivalent to some serial schedule of the same transaction.

X=100
S1 S2
Read_Item(X); Read_Item(X);
X:=X+10 X:=X*1.1
Write_Item(X); Write_Item(X);
X=110; X=110;