PAGE NO.
DATE:

DATABASE MANAGEMENT SYSTEM
DBMS

INTRODUCTION TO DBMS :

A DBMS is a collection of Interrelated data and a set of programs to access those data.

PRIMARY GOAL :

Providing a way to store and retrieve information i.e. both in convenient and efficient way.
usually used for storing large amount of data.

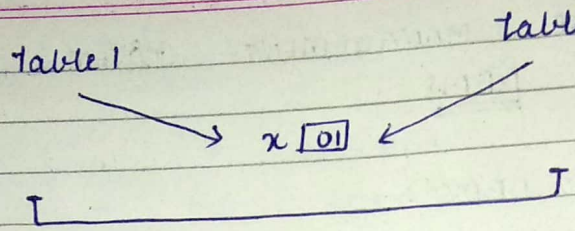
APPLICATIONS :

Banking	Telecommunication	Human Resource
Airlines	Finance	
Universities	Sales	
Online retailers	Manufacturing	

FILE SYSTEM VS DBMS :

I) Data Redundancy → duplicacy terminated
II) Difficulty in accessing data :
III) Atomicity Problem :
IV) Concurrency Access anomalities :

eg : x [0]



read and write
changing values

① read (A)	01	② read (A)	01
③ A = A+10	11	④ A = A+5	6
⑤ write (A)	11	⑥ write (A)	6

a 01
↓
a 6

v) Security :

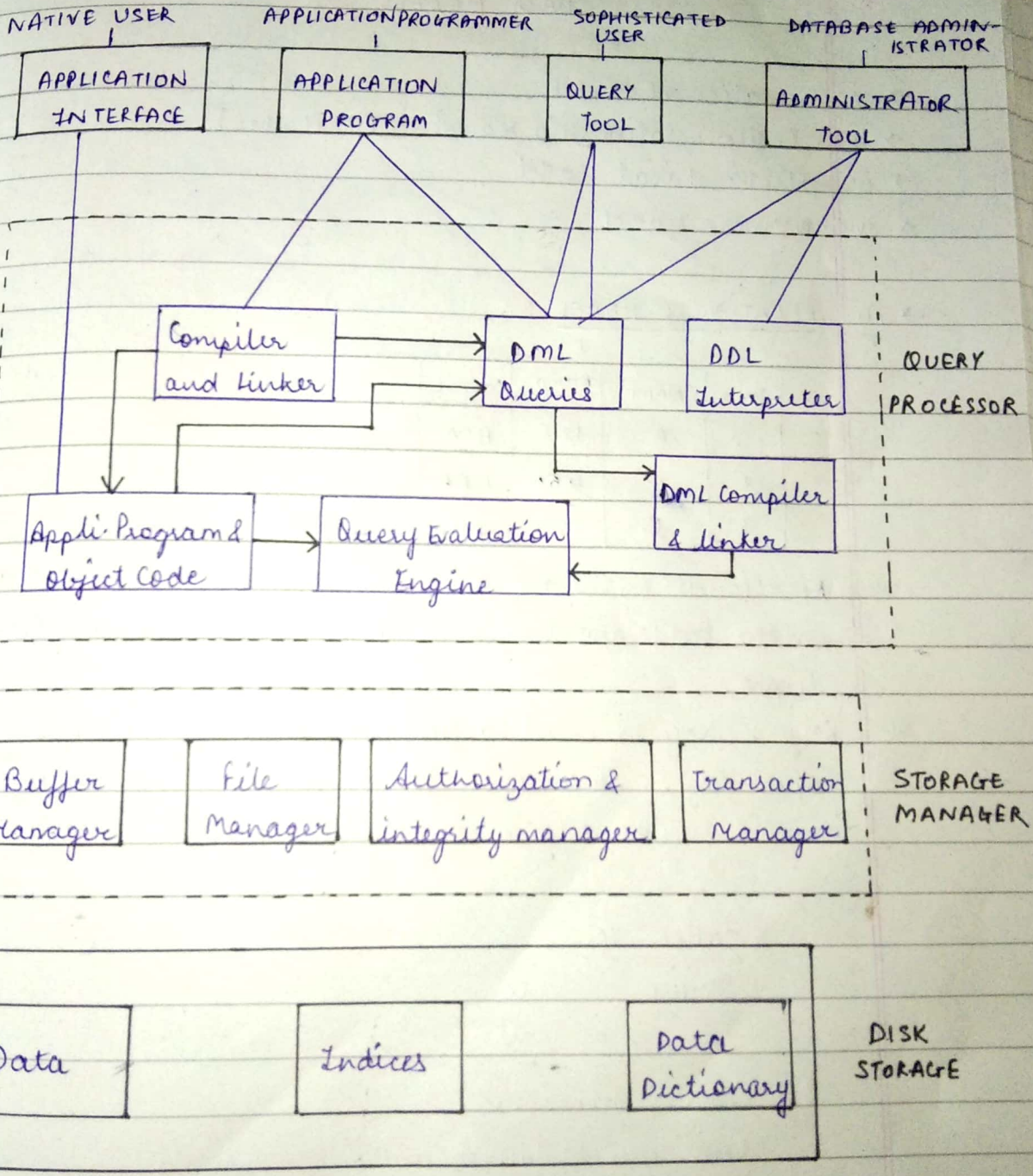
vi) view of data : DATA ABSTRACTION

ABSTRACTION : system hides certain details of how the data is stored and maintained.

a) Physical level → hiding data structures
b) logical level → hides relation between data
c) View level → controlling the access of data

* STRUCTURE OF DBMS

PAGE NO.
DATE:



PAGE NO.
DATE:

17.01.19

DATA MODELS

→ i Relational Model
→ ii Entity Relationship Model (ER Model)
→ iii Hierarchical Model
→ iv Network Model

→ (i) RELATIONAL MODEL :

S.NO	SNAME	PHNO	COURSE
1	A	123	BCA
2	B	456	BBA
3	C	789	MCA

* Relational Database consists of collection of tables. the columns in the table are termed as attributes & rows are termed as tuple.
* Key : Key is used to uniquely identify a tuple.
* Super Key : It is a set of one or more attributes, which allows us to identify a tuple in a table.

SUPER ID

SNO ✓
Name X
SNO, Name ✓
SNO, Name, Course ✓

* SET - Collection of elements.

PAGE NO.
DATE:

Prat

* Candidate Key : Super Key with no proper subset is super Key is called Candidate Key.
Candidate Key is a Super Key with no proper subset.

(Super Key subset = Super Key then it is not a Candidate Key.)

* Primary Key : is a Candidate key i.e. chosen by database designer as principles needs of identifying tuples uniquely.

Primary Key
↓ are
Candidate Key
↓ are
SUPER KEY

Continued on next page →

PAGE NO.
DATE:

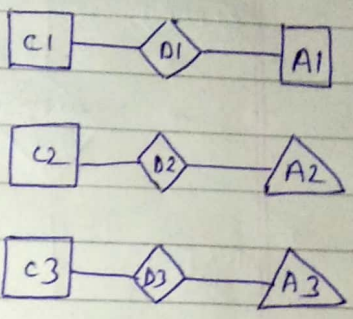
→ (ii) ENTITY RELATIONSHIP MODEL :

It is a Pictorial Representation

Entity — Real world Object

Relationship — association among entities

Entity Set

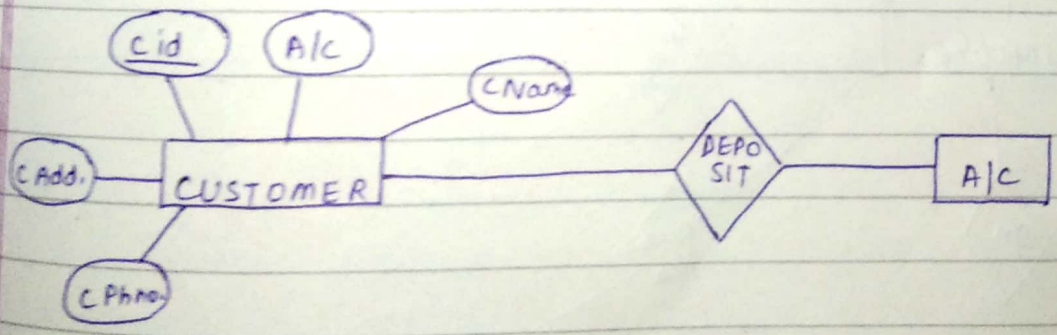


- Entity set is a set of similar entities.
- Relationship set is a set of similar type of associations.

□ = entity set , ⬭ = attributes , — = association

◇ = relationship set , ⬭ (underlined) = primary key , ▣ = weak entity

* EXAMPLE : 01

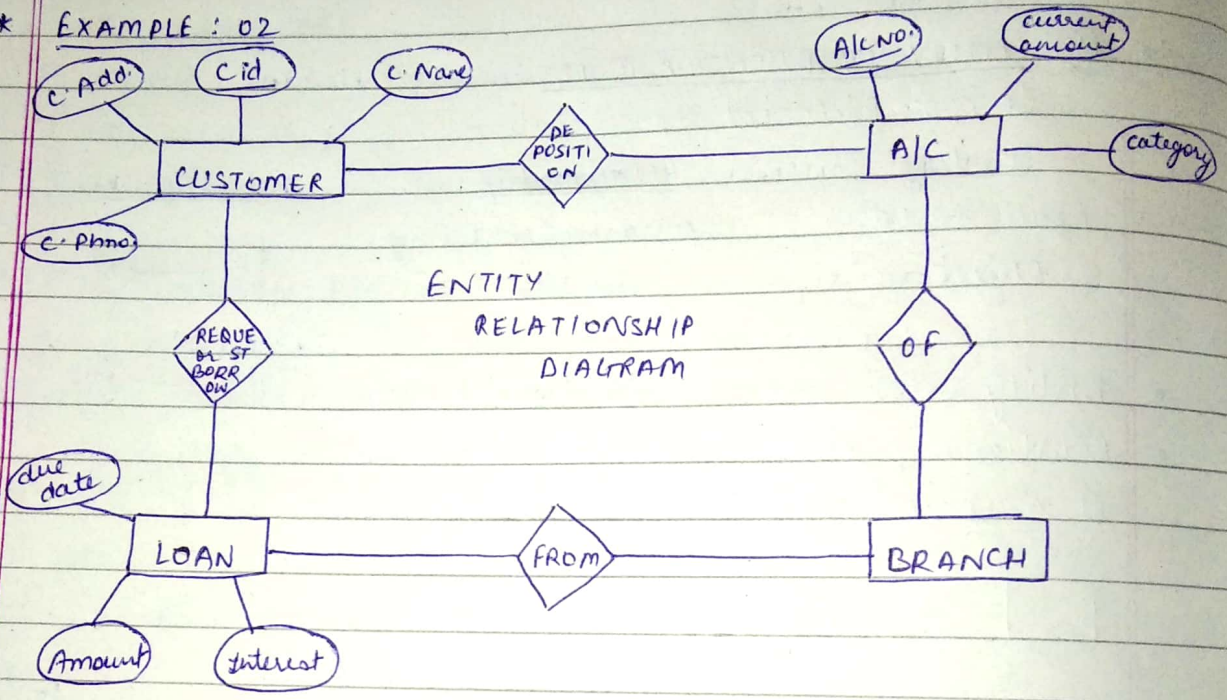


⬭ (double) = multivalued entity

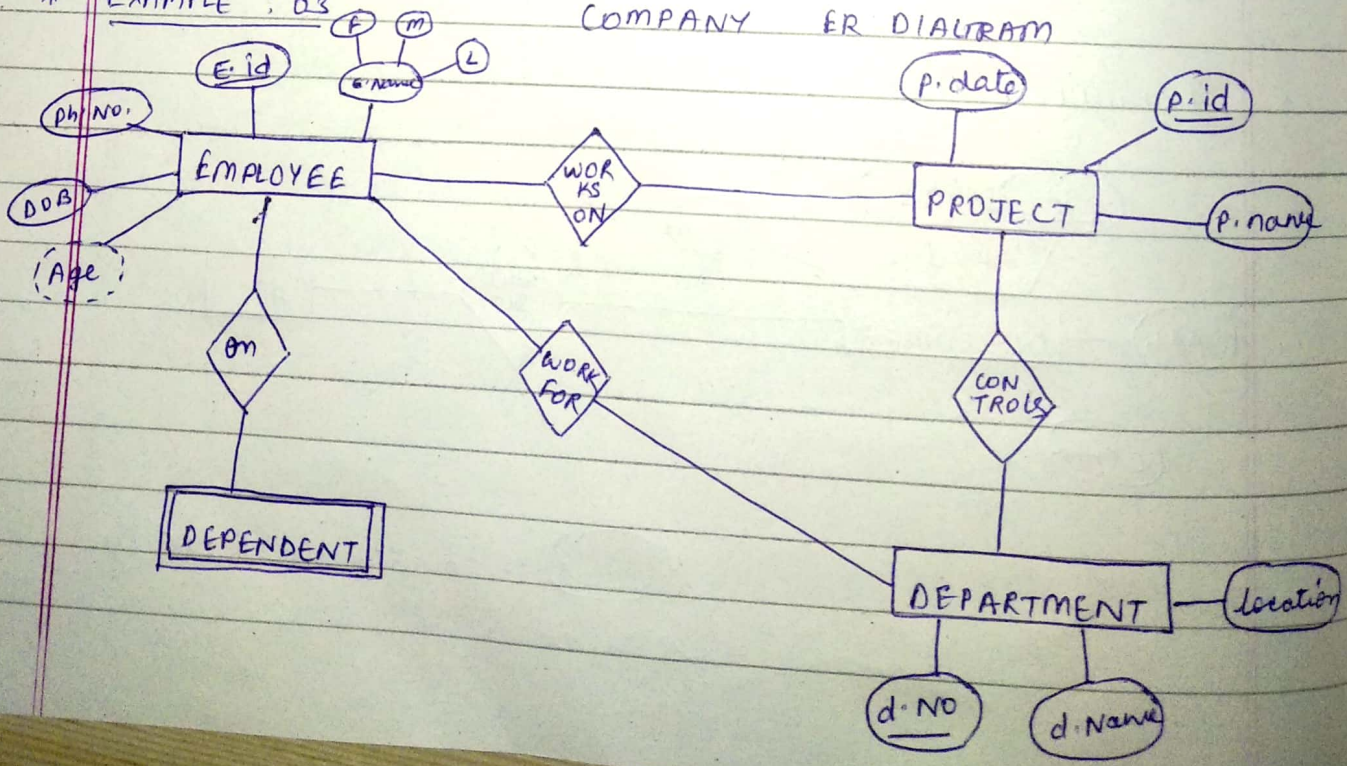
⬭ (dashed) = derived attribute

PAGE NO.
DATE:

* EXAMPLE : 02



* EXAMPLE : 03



PAGE NO.
DATE:

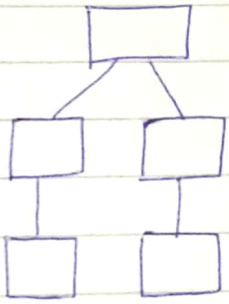
→ (iii) NETWORK MODEL:

- Network Model shows relationship between records in a database.
- Records consists of attributes of a single data value.

(iv) HIERARCHICAL MODEL:

- In the the records are related with each other in a form of a tree.

Example:



(v) OBJECT-ORIENTED MODEL:

- Unlike traditional database, an object model allows for data storage by storing object in the database.
- It is used for complex data structure such as 2D and 3D graphics.
- Object oriented database implements objects oriented concepts such as object identity, polymorphism, encapsulation, and Inheritance.

* * * * * *

PAGE NO.
DATE:

12.01.19

21.01.19

* **DATABASE USERS :**

i) NATIVE USER : Unsophisticated users, they invoke an application program to access DBMS like ATM User, or a person using Bank website.

ii) APPLICATION PROGRAMMER : writes application program.

iii) SOPHISTICATED USER : Interact with the system by forming their request in database Query language.

iv) SPECIALISED USER : Sophisticated users who can write specialised database application.

v) DATABASE ADMINISTRATOR : Jobs to do :

a) Structure definition or schema definition.
b) Storage structure and access control.
c) granting of information for data access.

- A person who has centraled control over the system is called Database Administrator.

PAGE NO.
DATE:

30/01/19

RELATIONAL MODEL	VS	OBJECT ORIENTED MODEL
• Class	——	Relation (Table)
• Object existance	——	Tuple (Rows)
• Attributes	——	Attributes (Columns)
• Method	——	Procedure

* Examples of Object Oriented Model :

a) dbYO
b) InterSystems
c) Versant
d) Objectivity Inc.

PAGE NO.
DATE:

STRUCTURED QUERY LANGUAGE

- SQL is a set based high level Computer language with which all programs and user can access database. programmers
- SQL is declarative and non-procedural language which describes what result we require and not how the user will get it.

QUERIES :

i) DDL = Data Definition Language
ii) DML = Data Manipulation Language

- example :

BANK DATABASE :

TABLES : Branch (B·NAME, B·CITY, ASSETS)
Customer (C-ID, C-NAME, C-SHEET, C-PHNO., C-CITY)
Loan (L-NO., B-NAME, AMOUNT)
Borrower (C-ID, L-NO)
Account (ACC-NO., BRANCH_NAME, BALANCE)
Depositor (ACC-NO., C_ID)

Basic DATATYPES :

i) CHAR (n)
ii) VARCHAR
iii) INT
iv) NUMERIC
v) DATE
vi) TIME
vii) TIME STAMP
etc.

PAGE NO.
DATE:

- CREATE TABLE statement :

```
CREATE TABLE Customer
-> (C_ID int as primary key,
->  C_NAME varchar(20),
->  C_SHEET varchar(30),
->  C_PHNO. int,
->  C_CITY varchar(10));
```

Q: Write a query in SQL to create table Branch and put a constraint that the value of asset should be > 0.

```
CREATE TABLE Branch
-> ( B_ID int,
->  NAME varchar(20),
->  B_CITY varchar(20),
->  ASSETS int,
->  Check (ASSETS > 0));
```

- SELECT statement :

Structure : SELECT * or Attributes
from Tablename
where CONDITION ;

- CRE INSERT statement :

Structure : INSERT into Tablename values
(details or data);

PAGE NO.
DATE:

Q: WAQ in SQL to find the name of all branches in the loan relation :

→ SELECT B_NAME
from Loan;

Q: select NAME of distinct branches from LOAN table :

→ select UNIQUE (B_NAME)
from loan;

Q: display complete LOAN table :

→ select * from LOAN;

Q: Return a relation that is same as loan Relation except that the attribute amount is multiplied by 100 :

→ select L_NO, B_NAME, AMOUNT × 100
from LOAN;

Q: Find all LOAN NO. for loan made at Jaipur branch and loan ₹ is greater the ₹ 50000 :

→ select L_NO from LOAN
where Branch_NAME = "Jaipur"
and AMOUNT > 50000;

Q: Find LOANNO. for the loans where ₹ is b/w 50000 to 80000 :

→ select L_NO from LOAN
where AMOUNT between 50000 and 80000;

PAGE NO.
DATE:

Q: Find L_NO for LOAN table where amount is not b/w 50000 to 80000:

→ select L_NO from LOAN
where AMOUNT not between 50000 and 80000;

31.01.19

Q: WAQ to find out NAME of employees who are working in HR department:

→ select ENAME from
employee023.e, department023 d
where e.DEPTNO = d.DEPTNO and
DNAME = 'Human Resource';

Q: Find EMPNO. and DEPTNO. for all employees whose location for department is 'JAIPUR':

→ select EMPNO, e.DEPTNO
from employee023 e, department023 d
where e.DEPTNO = d.DEPTNO and
Location = 'JAIPUR';

Q: Find EMPNAME who are working in a department whose no is 1:

→ select ENAME from employee023 e, department023 d
where e.deptNO = d.DEPTNO
and DEPTNO = '1';

_ = 1 bit
% = unlimited bit

PAGE NO.
DATE:

* STRING OPERATIONS :

Q: To display NAME of all employees for employee name starting with A:

-> select ENAME from Employee023
where ENAME like ('A%');

Q: To display NAME of employees where employee name is having minimum 5 characters:

-> select ENAME from EMPLOYEE023
where ENAME like ('_____%');

Q: Find EMPNAME where EMPNAME is having A in his name:

-> select ENAME from EMPLOYEE023
where ENAME like ('%A%');

Q: For all customers who have a loan from the BANK. find their CUSTID, LOANNO, LOANAMOUNT.

-> select CUSTID, LOANNO, LOANAMOUNT
from loan L, borrower b
where l.L-NO = b.L-NO;

Q: Find out Customer ID, loan-NO, Loan ₹ for all customers who have taken loan from JAIPUR branch.

-> select C-ID, LoanNO, amount
from loan l, borrower b
where l.L-NO = B.L-NO

and where B-NAME = 'Jaipur';

Q: Find the name of all CUSTOMERS whos street address include 'Main'.

-> select * from Customer
where C-STREET like ('% Main %');

Q: Find in alphabetic order all customers names (ID) who have taken loan from the JAIPUR Branch.

-> select CUST-ID, B-NAME
from loan l, borrower b
where l.LOANNO = b.LOANNO
and B-NAME = 'Jaipur'
order by C-ID;

* SET OPERATIONS :
Collection of set elements.

Union ∪
Intersect ∩
except → common values eliminate

PAGE NO.
DATE:

Q: Find all customers having a loan and (or) account or both at the bank

→ A {1, 2, 3} B {3, 4, 5}

- Union = {1, 2, 3, 4, 5}
- Intersect = {3}
- except = {1, 2, 3} − {3, 4, 5}
  ⇒ {1, 2}
  ⇒ {P, Q, R} − {P, Q}
  ⇒ {R}

→ select C-ID from DEPOSITOR
union
select C-ID from BORROWER;

Q: To find all customers who have both a loan and an account at the Bank.

→ select C-ID from DEPOSITOR
intersect
select C-ID from BORROWER;

Q: To find all customer who have an A/C but taken no loan:

→ select C-ID from DEPOSITOR
except
select C-ID from BORROWER;

PAGE NO
DATE

* AGGREGATE OPERATIONS & FUNCTIONS :
These are functions that take a collection of values as input and return a single value.
i) Average : avg
ii) Maximum : max
iii) minimum : min
iv) SumTotal : sum
v) Count : count

Q: Find the avg account balance at the Jaipur branch :
-> select avg (BALANCE)
from ACCOUNT
where BRANCH_NAME = 'Jaipur';
-> select max (BALANCE) = MAXIMUM
-> select Min (BALANCE) = MINIMUM
-> select Sum (BALANCE) = TOTAL
-> select Count (BALANCE) = COUNT