VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

BCSE302L – Database Systems

(Common question paper)

CAT-I Examination, May, 2023

Time: 90 minutes **Max. Marks: 50**

(10 X 5 = 50)

Answer ALL Questions

1. A) Distinguish the main characteristics of database approach from the much older approach of managing data with traditional file processing system. **(CO1, 5 marks) (BL: 4)**

Comparison should be done between DBMS and traditional file handling system by considering the main characteristics of database approach (Self describing nature, Insulation between programs and data, supporting multiple views of data and sharing of data and multiuser transaction processing) with the traditional file processing difficulties (sharing data with multiple users, data redundancy and inconsistency, difficulty in accessing data on time and atomicity of data).

b) Assume that you are developing a web application which has to hold a huge volume of data and should be accessed in a secured and fast manner to perform several operations concurrently without any conflicts. Determine the suitable architecture that is applicable for the given scenario and discuss in detail.

(CO1, 5 marks) (BL: 4)

Three-Tier is suitable for the given scenario since it improves data security and gets the required data on time as per the request and present it in good GUI format.

Many Web applications use an architecture called the **three-tier architecture**, which adds an intermediate layer between the client and the database server, as illustrated in Figure 2.7(a).

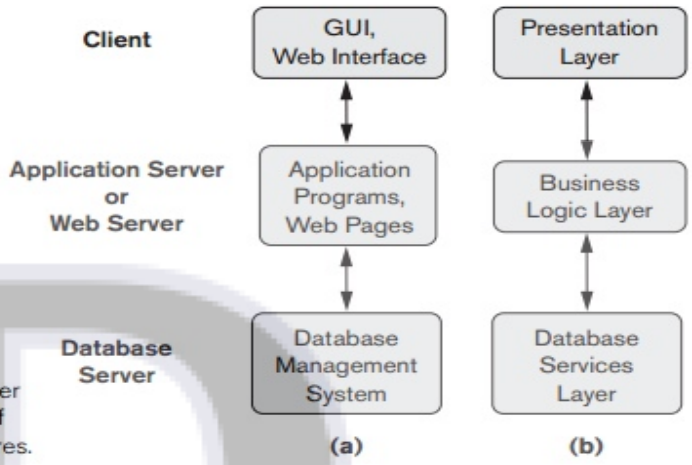


Figure 2.7
Logical three-tier client/server architecture, with a couple of commonly used nomenclatures.

This intermediate layer or **middle tier** is called the **application server or the Web server**, depending on the application. This server plays an intermediate role by running application programs and storing business rules (procedures or constraints) that are used to access data from the database server. It can also improve database security by checking a client's credentials before forwarding a request to the data-base server. Clients contain GUI interfaces and some additional application-specific business rules.

The intermediate server accepts requests from the client, processes the request and sends database queries and commands to the database server, and then acts as a conduit for passing (partially) processed data from the database server to the clients, where it may be processed further and filtered to be presented to users in GUI format. Thus, the *user interface, application rules,* and *data access* act as the three tiers. Figure 2.7(b) shows another architecture used by database and other application package vendors.

The presentation layer displays information to the user and allows data entry. The business logic layer handles intermediate rules and constraints before data is passed up to the user or down to the DBMS. The bottom layer includes all data management services. The middle layer can also act as a Web server, which retrieves query results from the database server and formats them into dynamic Web pages that are viewed by the Web browser at the client side.

Other architectures have also been proposed. It is possible to divide the layers between the user and the stored data further into finer components, thereby giving rise to *n*-tier architectures, where *n* may be four or five tiers. Typically, the business logic layer is divided into multiple layers.

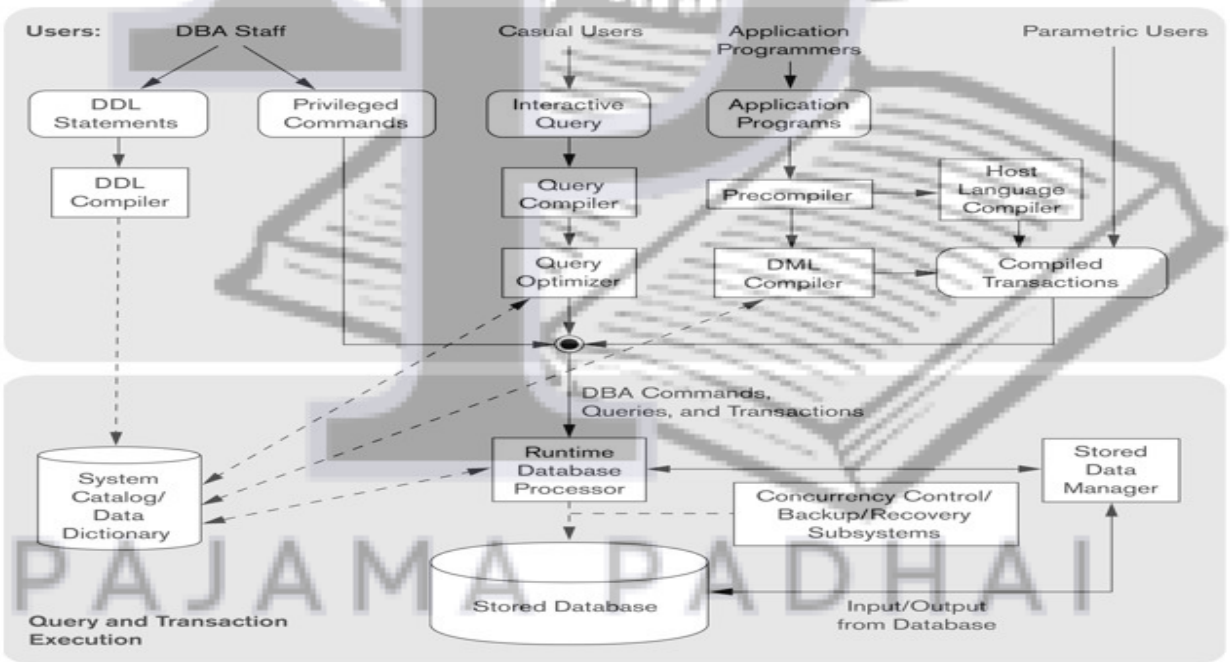
Besides distributing programming and data throughout a network, *n*-tier applications afford the advantage that any one tier can run on an appropriate processor or operating system platform and can be handled independently.

Vendors of ERP (enterprise resource planning) and CRM (customer relationship management) packages often use a *middleware layer,* which accounts for the front-end modules (clients) communicating with a number of back-end databases (servers).

Advances in encryption and decryption technology make it safer to transfer sensitive data from server to client in encrypted form, where it will be decrypted. The latter can be done by the hardware or by advanced software. This technology gives higher levels of data security, but the network security issues remain a major concern. Various technologies for data compression also help to transfer large amounts of data from servers to clients over wired and wireless networks.

2. Discuss various component modules of a DBMS and their interactions with a neat diagram.

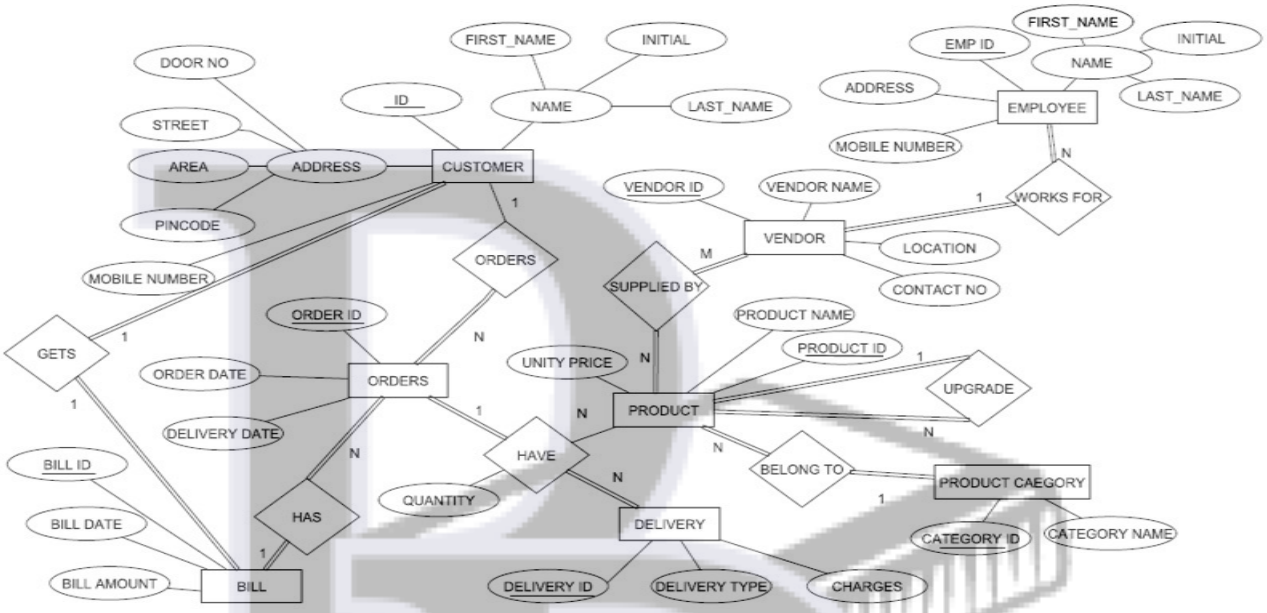
(CO1, 10 marks) (BL: 2)



A database system being a complex software system is partitioned into several software components that handle various tasks such as data definition and manipulation, data security and integrity, data recovery and concurrency control, and performance optimization. All these components should be explained properly.

3. Depict the given ER model to relational schema by following appropriate mapping steps.

(CO2, 10 marks) (BL: 6)



A) Strong Entity Sets - Mapping
1) Customers (CID, FN, LN, Ini, Door_id, Stno, Area, PC, mob)
2) Employee (EmpID, FN, LN, Ini, Add, mob)
3) Vendor (Ven_ID, Vname, Vloc, V_contact no)
4) Orders (O_id, or_date, Del_Date)
5) Products (P_id, Pname, unit_Price)
6) Product_category (Cat_id, Cat_Name)
7) Delivery (D_id, Del_Type, charges)
8) Bill (B_id, B_date, B_amt)

B) There is no weak Entity Set in this ER diagram

C) 1:1 Relationship mapping
Bill (B_id, C_id, B_date, B_amt)

D) 1:N Relationship mapping (5 nos of 1:N Relation)
1) Orders (O_id, cid, or_date, del_date)
{ between customers & orders}
2) Employee (Emp_id, V_id, FN, LN, Ini, Add, mob)
3) Orders (or_id, Cid, B_id, or_date, del_date)
{ between Orders & Bill}
4) Product (P_id, Cat_id, P_Name, Unit Price)
{ Between Products & Product Category}
5) Product (P_id, Cat_id, Super P_id, P_Name, Unit Price)
{ self relationship (upgrade)}

E) M:N relationship mapping
Supplied by (V_id, P_id)

F) There is no multi valued attribute mapping

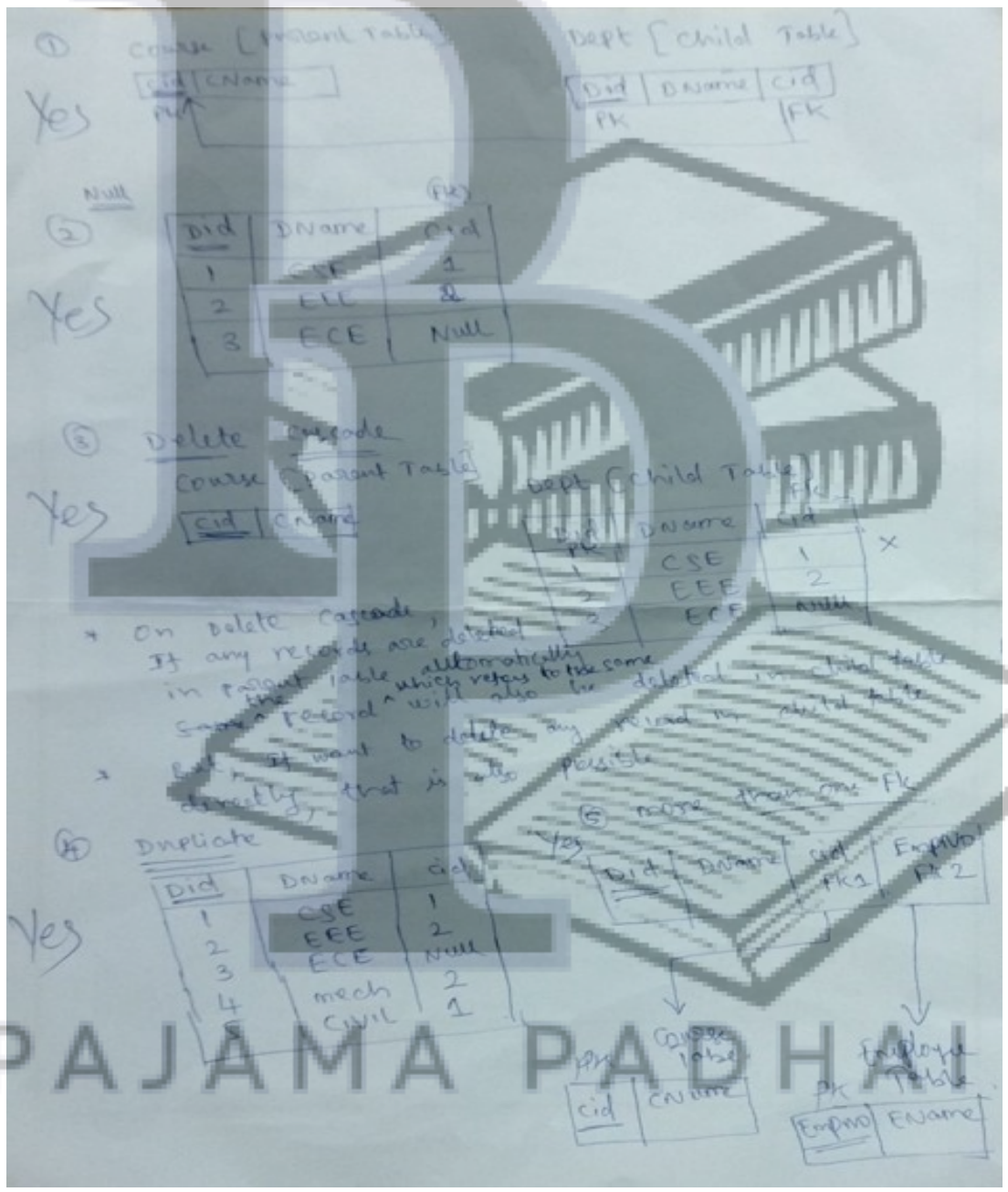
G) N ary relationship mapping (Ternary)
Have (P_id, D_id, O_id, Quantity)
{ between Orders, Product & Delivery}
{ N Side - Product, delivery, So, P_id, d_id are Primary keys}

Final mapped Relations from given ER diagram
1) Customers (C_id, FN, LN, Ini, D_id, Stno, Area, PC, mob)
2) Employee (Emp_id, Ven_id, FN, LN, Ini, Add, mob)
3) Vendor (V_id, Vname, V_locations, V_Cont no)
4) Orders (O_id, C_id, B_id, Ord_date, Del_date)
5) Products (P_id, Cat_id, Super_P_id, P_Name, Unit Price)
6) Product_Cat (Cat_id, Cat_Name)
7) Delivery (D_id, Del_type, Charges)
8) Bill (B_id, cid, B_date, B_amt)
9) Supplied by (Vid, P_id)
10) Have (Pid, Did, O_id, Quantity)

4. Justify your answer for the following questions with appropriate examples. **(CO2, 10 marks) (BL: 4)**

i) Is it mandatory for a foreign key to be a primary key in another table? No
ii) Can foreign key values be null? Yes
iii) After enforcing foreign key constraint using on delete cascade rule, is it possible to delete a record in the child relation whose foreign key value is NOT NULL before deleting the corresponding primary key record in the parent relation? Yes
iv) Can foreign keys have duplicate values? Yes
v) Can there exist more than one foreign key in a relation? Yes

Examples for all the five sections are furnished below:



5. Consider the pet health history report table given below and normalize it to 3NF**.**

(CO2, 10 marks) (BL: 6)

PET ID	PET NAME	PET TYPE	PET AGE	OWNER	VISIT DATE	PROCEDURE
246	ROVER	DOG	12	SAM COOK	JAN 13/2002	01 - RABIES VACCINATION
					MAR 27/2002	10 - EXAMINE and TREAT WOUND
					APR 02/2002	05 - HEART WORM TEST
298	SPOT	DOG	2	TERRY KIM	JAN 21/2002	08 - TETANUS VACCINATION
					MAR 10/2002	05 - HEART WORM TEST
341	MORRIS	CAT	4	SAM COOK	JAN 23/2001	01 - RABIES VACCINATION
					JAN 13/2002	01 - RABIES VACCINATION
519	TWEEDY	BIRD	2	TERRY KIM	APR 30/2002	20 - ANNUAL CHECK UP
					APR 30/2002	12 - EYE WASH

UNF:

Pet [pet_id, pet_name, pet_type, pet_age, owner, (visitdate, procedure_no, procedure_name)]

1NF:

Pet [pet_id, pet_name, pet_type, pet_age, owner]

Pet_Visit [pet_id, visitdate, procedure_no, procedure_name]

note: a procedure may occur on multiple dates, therefore visitdate is included as part of the key

2NF:

Pet [pet_id, pet_name, pet_type, pet_age, owner]

Pet_Visit [pet_id, visitdate, procedure_no]

Procedure [procedure_no, procedure_name]

3NF:

same as 2NF