WINTER – 2019 EXAMINATION
MODEL ANSWER

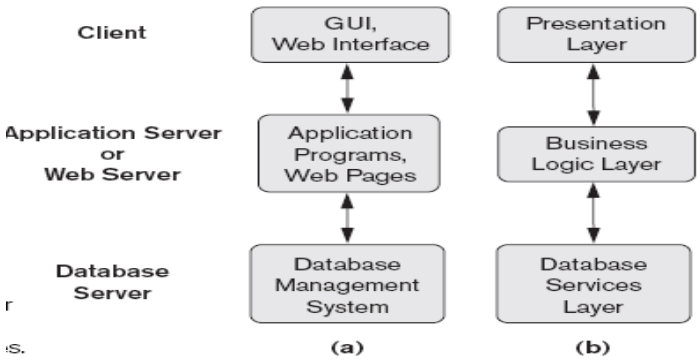
Subject: Principles of Database **Subject Code: 22321**

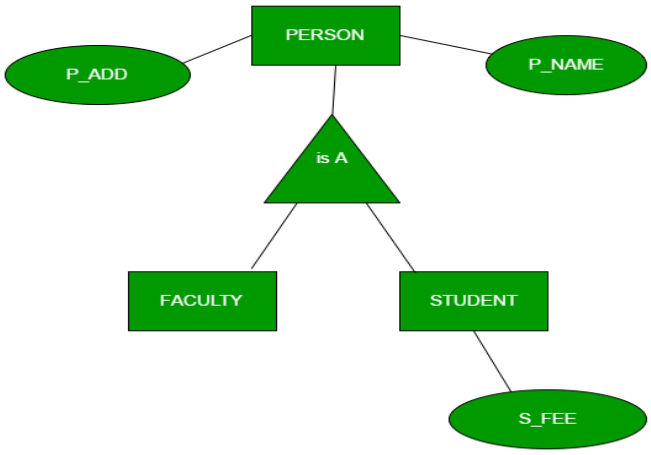
Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No.	Sub Q.N.	Answer	Marking Scheme
1.	**(a) Ans.**	**Attempt any FIVE:** **Define (i) Data Abstraction, (ii) Data Redundancy.** **(i) Data Abstraction:** Data Abstraction is hiding the details of data organization and storage and highlighting the essential features for an improved understanding of data. **(ii) Data Redundancy:** The Data redundancy is the storing of same data multiple times. This leads to duplication of effort. Second, storage space is wasted.	**10** **2M** ***Each definition 1M***
	(b) Ans.	**Define the term tuple and domain.** **tuple:** A row is called a **Tuple.** **domain:**A domain is a set of all possible (or permissible) values in an attribute. **OR** A Domain is defined as a kind of data represented by an attribute.	**2M** ***Each definition 1M***
	(c) Ans.	**Define primary key and candidate key.**	**2M**

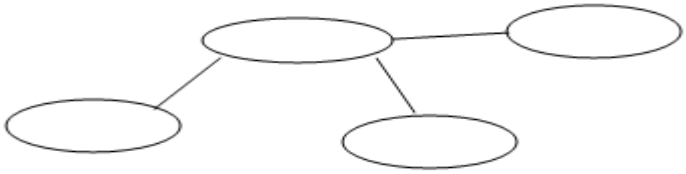
		Primary key: The PRIMARY KEY uniquely identifies each record in a database table. Primary keys must contain unique values. A primary key column cannot contain NULL values. Each table should have a primary key, and each table can have only one primary key. **Candidate key:** A minimal super key is called a candidate key. An entity set may have more than one candidate key. A candidate key is a column, or set of columns, in a table that can uniquely identify any database record without referring to any other data. Each table may have one or more candidate keys, but one candidate key is special, and it is called the primary key.	***Each definition 1M***
	(d) Ans.	**Define constraints, list types.** Constraints are used to limit the type of data that can go into a table. Constraints are used to ensure accuracy and consistency of data in a relational database. **Types of Constraints** : 1.NOT NULL Constraint 2.DEFAULT Constraint 3.UNIQUE Constraint 4.CHECK Constraint 5.Primary Key Constraint 6. Foreign Key Constraint	**2M** ***Definition 1M*** ***Types 1M***
	(e) Ans.	**Define Data and instance.** **Data:** Data can be defined as facts or information that can be recorded and have an implicit meaning. **Instance:** The collection of information stored in the database at a particular moment is called an instance of the database.	**2M** ***Each definition 1M***
	(f) Ans.	**Write Syntax for create table.** **Syntax of Create table:** CREATE TABLE table_name(column1 datatype (size), column2 datatype(size), column3 datatype(size), ... columnNdatatype(size));	**2M** ***Correct syntax 2M***
	(g) Ans.	**Define Normalization, list its types.**	**2M**

		Normalization is a systematic approach of decomposing tables to eliminate data redundancy(repetition) and undesirable characteristics like Insertion, Update and Deletion Anomalies. It is a multi-step process that puts data into tabular form, removing duplicated data from the relation tables. Types of Normalization are: 1NF,2NF,3NF,4NF,5NF	***Definition 1M*** ***Types 1M***
2.	**(a)** **Ans.**	**Attempt any THREE of the following:** **Explain three tier architecture of database with the help of diagram.**  **Application server or Web server** • Adds intermediate layer between client and the database server • Runs application programs and stores business rules Clients contain GUI interfaces and some additional application-specific business rules. The intermediate server accepts requests the clients, processes the requests and sends database commands to the database server and then acts as a conduit for passing (partially processed data from the database server to the clients, when it may be processed further and filtered to be presented to users in GUI format. Thus the user interfaces, application rules and the database acts as three tier.	**12** **4M** ***Diagram 2M*** ***Explanation 2M***
	(b) **Ans.**	**Describe client server system with example.** Client server system consists of two logical components. One is "Client" and the other one is "Server". Clients are those who send the request to perform a specific task to the server. Servers normally receive the command sent by the clients, perform the task and send	**4M** ***Description 2M***

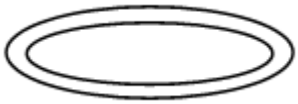
		the appropriate result back to the client. ***Example*** of client is PC where as the server is a large work station. The Client machine runs own copy of an operating system. It runs one or more applications through client's CPU and memory. But server runs a database management system which manages the whole database.	***Example 2M***
	(c) Ans.	**Explain Generalization with example.** **Generalization** uses bottom-up approach where two or more lower level entities combine together to form a higher level new entity if they have common attributes in common. The new generalized entity can further combine together with lower level entity to create a further higher level generalized entity. ***For Example***, STUDENT and FACULTY can be generalized to a higher level entity called PERSON  	**4M** ***Explanation 2M*** ***Example 2M***
	(d) Ans.	**Explain components of database in detail.** **Components of a DBMS:** **(i) Query processor**: The query processor transforms user queries into a series of low level instructions. It is used to interpret the online user's query and convert it into an efficient series of operations in a form capable of being sent to the run time data manager for execution. **(ii) Run time database manager**: Run time database manager is the	**4M**

		central software component of the DBMS, which interfaces with user-submitted application programs and queries. It handles database access at run time. It converts operations in user's queries coming. It accepts queries and examines the external and conceptual schemas to determine what conceptual records are required to satisfy the user's request. It enforces constraints to maintain the consistency and integrity of the data, as well as its security. It also performs backing and recovery operations. **(iii) Authorization control**: The authorization control module **checks the authorization of users** in terms of various privileges to users. **(iv) Command processor**: The command processor processes the queries passed by authorization control module. **(v) Integrity checker:** It .checks the integrity constraints so that only valid data can be entered into the database. **(vi) Query optimizer:** The query optimizers determine an optimal strategy for the query execution. **(vii) Transaction manager**: The transaction manager ensures that the transaction properties should be maintained by the system. **(viii) Scheduler:** It provides an environment in which multiple users can work on same piece of data at the same time in other words it supports concurrency. **(ix) Data Manager:** The data manager is responsible for the actual handling of data in the database. It provides recovery to the system which that system should be able to recover the data after some failure. It includes Recovery manager and Buffer manager. The buffer manager is responsible for the transfer of data between the main memory and secondary storage (such as disk or tape). It is also referred as the cache manger.	***Any four components 1M each***
3.	**(a) Ans.**	**Attempt any THREE of the following:** **Explain Domain constraints with Syntax and example.** Domain constraints are used to maintain value according to user specification Domain constraints are: 1. **Not null**-such constraints are applied to an attribute when we have to specify that the attribute cannot accept null value. Null is in the domain of all attributes unless not null is applied. *Example:* Consider the schema student.Student{rollno, name,sscper}. The name	**12** **4M** ***Explanation 2M***

		of the student should not be null. So we can apply the not null constraint to the name attribute. *General syntax* (While creating table) Create table tablename(attr1 datatype(size), attr2 datatype(size) not null,attr3 datatype(size)); After creating the table Alter table tablename modify attr not null; *Example:* Create table student(rollno number(5),name varchar(30) not null,sscper number(3)); Alter table student modify name not null; 2. **Check** – allows enforcing domain integrity by limiting the values accepted by an attribute. *Eg:* consider an attribute age of the entity employee. If age should be limited to 60, check constraint can be used *General syntax:* Create table tablename(attr1 datatype(size),attr2 datatype(size) constraint nameofconstraint check(attr<value)); or Alter table tablename add constraint nameofconstraint check(attr<value) *Eg:* Create table emp(empno number(4),name varchar(30),age number(3) constraint chk_emp check(age>60)); or Alter table emp add constraint chk_emo check(age>60)	***Syntax and example 2M***
	(b) Ans.	**Describe benefits and drawbook of denormalization.** **Benefits of denormalization:** • Reduce number of relations: It reduce the number of relations because it combines two relations into one new relation. • Reduce number of foreign keys: It reduce number of foreign keys because number of relations is reduced. • Minimizes need for joins: It minimizes need for joins because it combines many relations into one. • Increase Performance: It increase performance of database by adding redundant data or by grouping data. **Drawbacks of demoralization:** • Slow Data Updates: It may speed up the retrieval but can slow	**4M** ***Any 2 benefits and 2 drawback 1M each***

		down database updates • Increase size of relations: It can increase size of the relations due to combining multiple relations into one single relation. • Complex implementation: It may simplify implementation in some cases but may make it more complex in other. • Application Specific: It is always application-specific and needs to be re-evaluated if the application changes.	
	(c) **Ans.**	**Explain different types of attribute with example and their symbols used in ER diagram.** Different types of attributes are: 1. Simple attribute: A simple attributes are those which cannot be subdivided. *Eg:*Rollno– symbol  2. Composite attribute: a composite attribute is that which can be subdivided *Eg:* name – can be divided into first_name, middle_name and last_name Symbol  3. Single valued attribute- an attribute which can have only one value for an entity. *Eg:*ssc_per Symbol :  4. Multivalued attribute - an attribute that can take more than one value for an entity. *Eg:*phoneno	**4M** ***Any four attributes 1M each***

Symbol



5. Derived attribute - an attribute for which the value can be calculated or determined from another attribute
Eg: age from dateofbirth
Symbol

(d) Ans.

Differentiate between Hierarchical Database model and network database model. **4M**

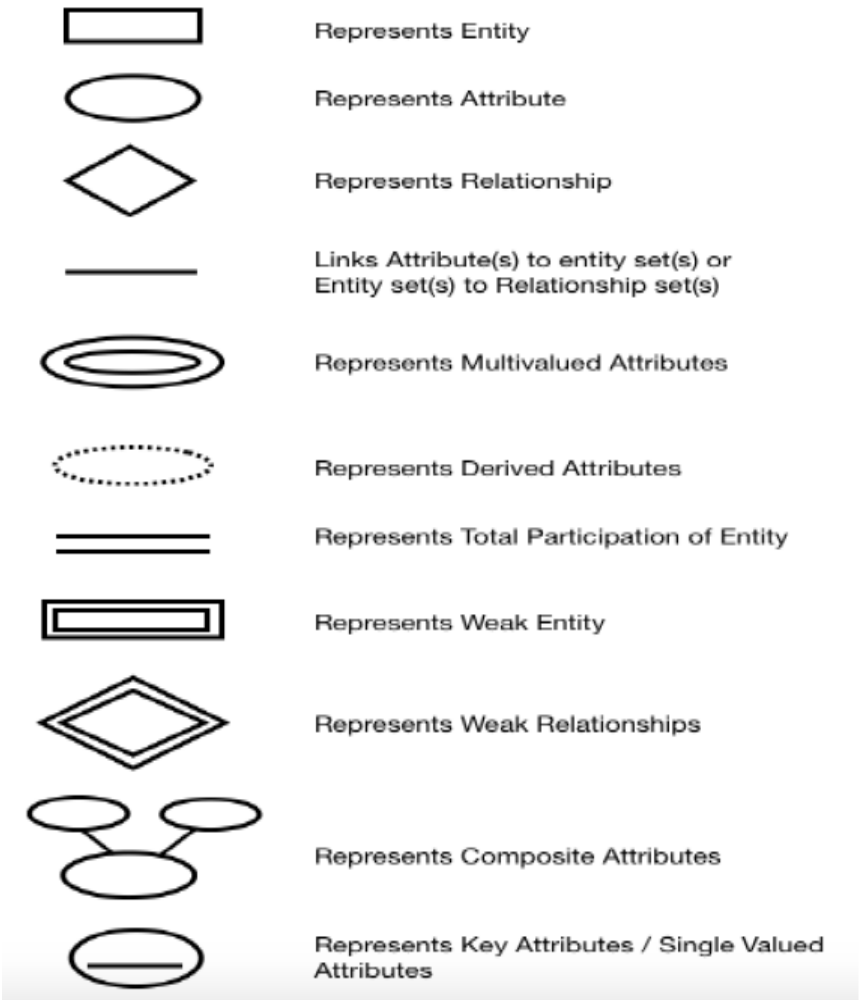
Sr. No.	Hierarchical data model	Network data model
1	Represents tree like structure with one root	Represents tree like structure with many roots
2	Reflects 1:N (one-to-many)relations	Reflects M:N(many to many) relations
3	There can be only one parent node	Allows a child to have more than one parent
4	Relationships between records is of parent-child type	Relationship is represented as pointers or links
5	There are multiple occurrence of child records and therefore inconsistency	This model is free from such inconsistency as there is only a single occurrence of a record set.
6	Searching a record is difficult as a child can be reached only through a parent	Searching a record is easy as there are multiple paths to a data element.

Any four points 1M each

WINTER – 2019 EXAMINATION
MODEL ANSWER

Subject: Principles of Database **Subject Code: 22321**

4.	**(a) Ans.**	**Attempt any THREE of the following:** **Explain functional dependency with example.** A functional dependency occurs when one attribute in a relation uniquely determines another attribute. **OR** Consider a relation say R(X,Y), where X and Y are one or more than one attribute, attribute X is functionally dependent on attribute Y if every value in X in the relation R has exactly one value of Y in the given relation. The functional dependency is represented as X→Y, which specifies Yis functionally dependent on X or X attribute functionally determine the attribute Y. Consider the schema, student(rollno, name, sscper). rollno→name, rollno→sscper are the functional dependencies. rollno uniquely identifies name and sscper. That is, given rollno of a student, the name and sscper can be determined or searched.	**12** **4M** ***Explanation 2M*** ***Example 2M***
	(b) Ans.	**Explain merits and demerits of Object Oriented Database model.** Object oriented models were introduced to overcome the shortcomings of conventional models like Relational, Hierarchical and network model. An object-oriented database is collection of objects whose behavior, state, and relationships are defined in accordance with object-oriented concepts (such as objects, class, class hierarchy etc). **Merits:** • Object oriented data model allows the real world to be modeled closely. The object encapsulates both state and behavior. The object can also store the relations with other objects. • It allows new data types to be built from existing types. Redundancy can be reduced as common factors of several classes can be grouped into a super class and can be shared by the sub classes. • It can be used to store a variety of data. • Data evolution is easier. **Demerits:** • There is a lack of universal data model. • Use of this type of modeling is still limited. • It lacks standards since there is no universal data model.	**4M** ***Any 2 merits & demerits 1M each***

		• Increased functionality provided by this modeling makes it complex. • There is no view mechanism • There is no adequate security mechanism.	
	(c) **Ans.**	**Draw the symbols used for entity relationship diagram and write their meaning.**  	**4M** ***Any eight ½M each***
	(d) **Ans.**	**Explain any 4 Codd's rules.** **Codd rules:** **Rule 1: The information rule** a has to be presented to the user should be in the form of table.	**4M**

		Rule 2: Guaranteed Access Rule Whole data should be available or accessible to the user without any ambiguity. The ambiguity can be avoided only through the perfect combination of the table name, primary key, and column name. **Rule 3: Systematic treatment of null values** The null values i.e. absence of the values in the table should be treated properly. The table should allow a field to remain empty. This is not applicable to primary keys. Key columns cannot have null values. **Rule 4: Active on-line catalog based on the relational model** Fourth rule specifies need of dynamic on-line catalog based on the relational model. There are certain system tables that stores the database definition should be present. The data accessing tools should be used to access the database structure information. **Rule 5: The comprehensive data sub language rule**: The system must support at least one relational language that Has a linear syntax Can be used both interactively and within application programs, Supports data definition operations (including view definitions), data manipulation operations (update as well as retrieval), security and integrity constraints, and transaction management operations (begin, commit, and rollback). **Rule 6: The view updating rule**: All views those can be updated theoretically, must be updated by the system. **Rule 7: High-level insert, update, and delete:** A database must support high-level insertion, updation, and deletion. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records **Rule 8: Physical data independence:** Changes to the physical level (how the data is stored, whether in arrays or linked lists etc.) must not require a change to an application based on the structure. **Rule 9: Logical data independence**: Changes to the logical level (tables, columns, rows, and so on) must not require a change to an application based on the structure. **Rule 10: Integrity independence**: Integrity constraints must be specified separately from application programs and stored in the catalog. It must be possible to change such constraints as and when appropriate without unnecessarily affecting existing applications. **Rule 11: Distribution independence**: The distribution of portions of the database to various locations should be invisible to users of the	***Any four rules 1M each***

		database. Existing applications should continue to operate successfully: when a distributed version of the DBMS is first introduced; and when existing distributed data are redistributed around the system. **Rule 12: The non subversion rule**: If the system provides a low level (record-at-a-time) interface, then that interface cannot be used to subvert the system, for example, bypassing a relational security or integrity constraint	
	(e) Ans.	**Explain distributed database system with example.** A distributed database is a database that consists of two or more files located in different sites either on the same network or on entirely different networks. Portions of the database are stored in multiple physical locations and processing is distributed among multiple database nodes. With distributed databases, data is physically stored across multiple sites and independently managed. The processors on each site are connected by a network, and they don't have any multiprocessing configuration. Distributed databases can be homogenous or heterogeneous. In a homogenous distributed database system, all the physical locations have the same underlying hardware and run the same operating systems and database applications. In a heterogeneous distributed database, the hardware, operating systems or database applications may be different at each location. **Advantage of Distributed databases:** **Better Response** − If data is distributed in an efficient manner, then user requests can be met from local data itself, thus providing faster response **More Reliable -** When the data and DBMS software are distributed over several sites one site may fail while other sites continue to operate ,which makes database more reliable **Easier Expansion** - : Expansion can be easily achieved by adding processing and storage power to the existing network. **Improved Performance** -These systems provide greater efficiency and better performance **Resource Sharing -**Since data is distributed, a group of users can easily share and use data of different sites Though there are many distributed databases to choose from, some	**4M** ***Explanation 3M*** ***Example 1M***

<table>
<tr><td></td><td></td><td>examples of distributed databases include Apache Ignite, Apache Cassandra, Apache HBase, Couchbase Server, Amazon SimpleDB, Clusterpoint, and FoundationDB</td><td></td></tr>
<tr><td>5.</td><td>(a)

Ans.</td><td>Attempt any TWO:
Consider following realtion
student (Roll_No, name, class, total_marks, percentage, Grade).
Find appropriate dependencies and normalize upto 3NF.
Functional Dependencies:
Roll_no→ name
Roll_no→ class
total_marks→ percentage
percentage → Grade

1NF: Student(Roll_no,name.class,total_marks,percentage,Grade)

2NF: To convert It into 2NF, We have to decompose the given table into two tables with fully functional dependencies and establishing a referential integrity constraint relationship among the two tables.

Student(Roll_No, name, class)

Marks(Roll_No, total_marks, Percentage, Grade)

3NF: To convert the above tables in 3NF ,We have to decompose them in three tables satisfying the transitive dependencies property

Student(Roll_No, name, class)

Marks(Roll_No, total_marks, percentage)

Grade (percentage, Grade)</td><td>12
6M

Functional dependency 2M

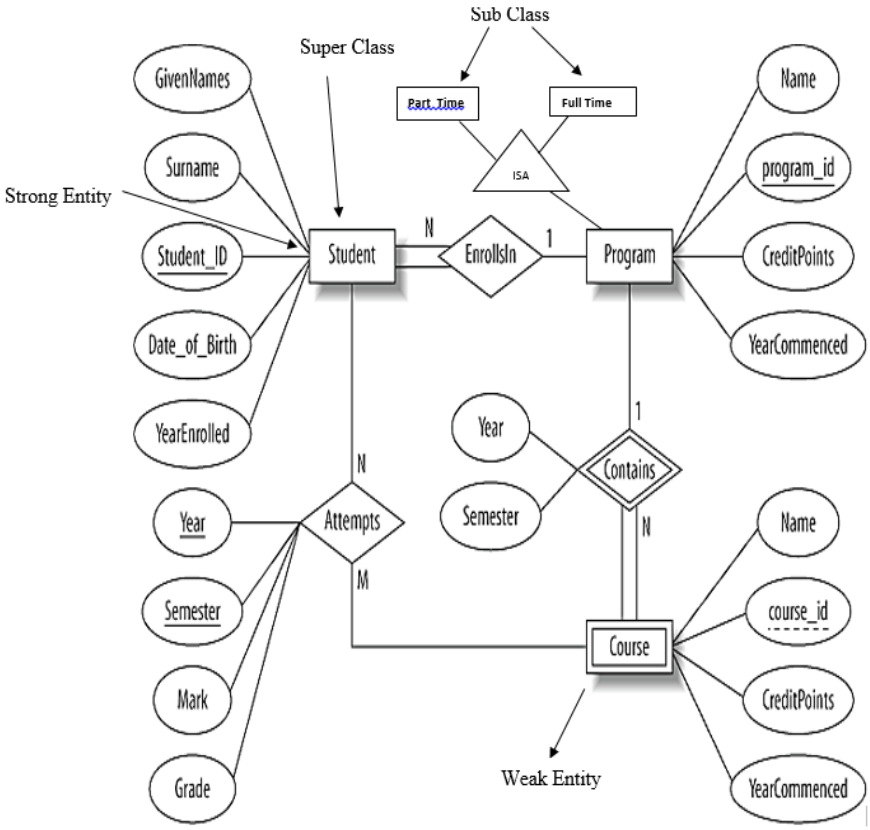
2NF 2M

3NF 2M</td></tr>
<tr><td></td><td>(b)

Ans.</td><td>Identify entities and their relationship in terms of tables for railway reservation system.
(Note: Any other entity or relationship shall be considered)
List of Entity Types:
<table><tr><th>Sr. No</th><th>Entity</th><th>Attributes</th></tr><tr><td>1</td><td>User</td><td>Email_Id,Password,Fullname,Gender,Age, Mobile,City,State</td></tr></table></td><td>6M</td></tr>
</table>

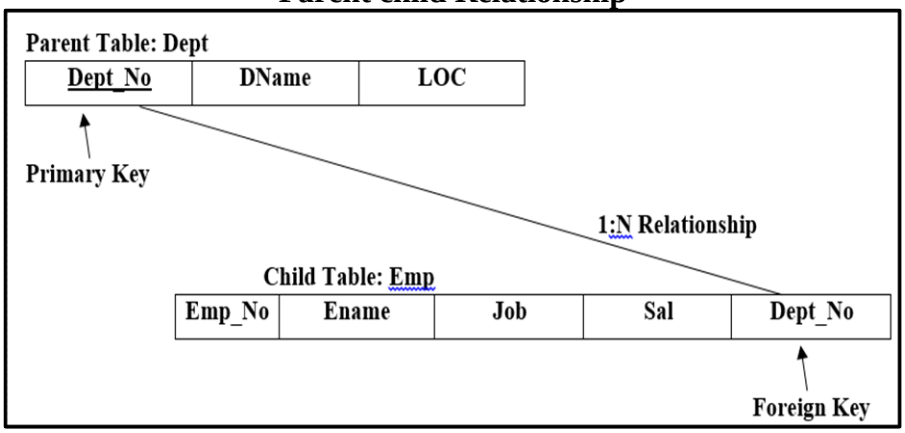
			2 | Passenger | PNR,Passenger_Name,Age,Gender,Reservation,Status,Booked_By 3 | Train | Train_Id,Train_Name,Train_Type,Avail_Days,Seat_Avail 4 | Route | Source_Dist,Stop_Number,Arrival_Time,Depart_Time 5 | Station | Station_Id,Station_Name 6 | Train_status | Avil_Date,Booked_Seat1,Waiting_Seat1, Avail_Seat1, Booked_Seat2,Waiting_Seat2, Avail_Seat2,Booked_Seat3,Waiting_Seat3, Avail_Seat3 **List of Relationship** **Sr. No** | **Relation Type** | **Entity Types Involved** 1 | Enquires | User, Train 2 | Consist_of | Station,Route 3 | Has | Train, Train_status 4 | checks | User, Train_status 5 | Has | Train,Route 6 | Starts_from/ends_on | Train, Station 7 | Assigns | User,Passenger	***Identify relevant entities 3M*** ***Identify relevant relationship3M***
	(c) **Ans.**	**Consider given relation R = (A, B, C, D, E) with the following functional dependencies {CE → D, D → B, C → A}.** **(i) List all key for R.** **(ii) Identify the best normal form that R satisfies.** **Step1:** Find attributes that are neither on the LHS nor on RHS --None **Step2:**Find the attributes that are only on RHS --A ,B **Step3:** Find the attributes that are only on LHS. --C, E **Step4:** Combine the attributes on step 1 and 3	**6M** ***Listing Key 3M***

		- C E The attributes C and E will belong to candidate key, but to find others we need to calculate closure of CE **Step5: Closure finding :** In our case, because with CE we get D and from D we get B and from C we get A **So we have only one candidate key that is CE** **The relation is in 1NF as it does not have any composite as well as multivalued attribute**. But it is not in 2NF as the statement says that 1) It should be in 1NF 2) All non-key attributes are fully functionally dependent on primary key In our case rule 2) is violated by C→A **Thus given relation is best suited for 1NF only.**	***Identification of normal form 3M***
6		**Attempt any TWO:**	**12**
	(a)	**Consider the following schema** **student (R_No, Name, DOB, Percentage, D_No).** **Write procedure to manipulate given database by adding, modifying and deleting records.**	**6M**
	Ans.	Consider given Schema **Student(R_No,Name,DOB,Percentage,D_No)** **<u>For adding records in table:</u>** We use Insert into command for adding/inserting data into Student table. ***Syntax for adding the values in the table is as follows:*** **SQL> Insert into <table name> values (value1, value2, value3…);** *Ex:* SQL>**insert into Student values(1,'Ram','12-Jan-1990',88,10)** **OR** *Ex:* SQL>**Insert into Student values(&R_No,'&Name','&DOB',&Percentage,&D_No);** **<u>For modifying records in table</u>** We use update command for modifying data of Employee table. ***The syntax of update command is:*** Update<table name>set	***Adding procedure 2M*** ***Modifying procedure 2M***

		<columnname>=<expression>,<columnname>=<expression>; *Ex:* SQL>**update Student set DOB='22-feb-1995' where R_No=3;** **<u>For deleting records from table:</u>** We use delete command for deleting data of Employee table. ***Syntax:-*** **Delete from <table name> where <condition>;** *Ex:* SQL>**delete from Student where R_No=2;**	***Deleting procedure 2M***
	(b) **Ans.**	**Draw the enhanced E-R diagram for College Management System and show strong entity set, weak entity set, super class and sub class.** ***(Note: Any relevant diagram shall be considered)***  	**6M** ***Correct Use Of symbols 2M*** ***Representation of strong entity 1M*** ***Representation of weak entity 1M*** ***Representation of super class 1M*** ***Representation of sub class 1M***

WINTER – 2019 EXAMINATION
MODEL ANSWER

Subject: Principles of Database **Subject Code: 22321**

	(c)	**Consider the following schemas:** **(i) Dept (Dept_No, DName, LOC)** **(ii) Emp (Emp_No, Ename, Job, Sal, Dept_No)** **Draw and explain parent child relationship for above schemas and apply referential integrity constraint.**	**6M**
	Ans.	**Parent child Relationship**  ***Diagram 2M***	
		Referential integrity constraint: • It is used to establish the parent child relation between two tables having common column. • Value of foreign key is derived from primary key. • We should define the column in the parent table as a primary key and same column in the child table as a foreign key referring to the corresponding parent key **Dept (Dept_No,** DName, LOC**)** **Emp(**Emp_No,Ename,Job,Sal,**Dept_No)** ***Explanation 1M***	
		In table D**ept, Dept_No is a primary key containing unique values for deptnos.** To set the relationship between these two tables , we can define Emp.Dept_No as a foreign key as 1. Create table Dept (Dept_No number(5) constraint Dept_Dept_No_pk primary key, DName varchar2(20), LOC char(10)); ***Primary key creation 1½M***	

		2. Create table Emp (Emp_No number(4), Ename varchar2(25), Job char(10), sal number(10,2) Dept_No number(5) constraint Emp_Dept_No_fk references Dept(Dept_No),);	***Foreign Key creation 1½M***