**DBMS Theory (all in one)**

**s m sadman sakib**

**Question Format**

**MCQ**

1. **functional dependency, armstrongs axiom and one proof**
2. **relational algebra**
3. **random question**
4. **Normalization**
5. **EER to relational conversion**
6. **EER diagram**

**1. which type of entity cannont exist in the database unless another type of entity also exist in the database but does not exist require that**

**the identifier of that other entity be included as part of its own identifier(2024,2022)**

**a. weak entity.**

**b. strong entity.**

**c. ID-dependent entity.**

**d. ID-independent entity.**

**answer: a**

**2. The different classes of relations created by the technique for preventing modification anomalies are called:(2024,2022,2018)**

**a. Normal forms.**

**b. Referential integrity constraints.**

**c. functional dependencies.**

**d. None of above is correct.**

**answer: a**

**3. A keys:(2024)**

**a. Must always be composed of two or more columns.**

**b. can only be one column.**

**c. identifies a row.**

**d. identifies a column.**

**answer: c**

**4. If a table has been normalized so that all determinants are candidate keys, then the table is in:(2024)**

**a. 1NF.**

**b. 2NF.**

**c. 3NF.**

**d. BCNF.**

**answer: d**

**5. When the values is one or more attributes being used as foreign key must exist in another set of one or**

**more attributes in another tables, we have created a(n):(2024,2022)**

**a. transitive dependency.**

**b. insertion anomaly.**

**c. referential integrity constraint.**

**d. normal form.**

**answer: c**

**6. A functional dependency is a relationship between or among:(2024)**

**a. tables.**

**b. rows.**

**c. relations.**

**d. attributes.**

**answer:d**

**7.\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is the minimal superkey.(2022)**

**a.primary key**

**b.candidate key**

**c.foreign key**

**d.alternate key**

**e.unique key**

**answer:b**

**8.Metadata about structure of database is stored in:(2022)**

**a.data files**

**b.indices**

**c.data dictionary**

**d.metadata files**

**answer:c**

**9.In the relational model ,relationship between relations or tables are created by:(2022,2018)**

**a.composite key**

**b.determinants**

**c.candidate keys**

**d.foreign keys**

**answer:d**

**10.A combination of Cartesian Product followed by selection process:(2022)**

**a.association**

**b.product formalization**

**c.joins**

**d.protocol**

**answer:c**

**11.Functional dependencies are the types of constraints that are based on:(2018)**

**a.keys**

**b.key revisited**

**c.superset keys**

**d.none of these**

**answer:a**

**12.Database ------- is the logical design of the data and the database ----------**

**which is snapshot of data at a given instance of time.(2018,2023)**

**a.instance, schema**

**b.relation, schema**

**c.relation, domain**

**d.schema, instance**

**answer:d**

**13.The tuples of the relations can be of \_\_\_\_\_\_\_\_\_ order.(2018,2023)**

**a.any**

**b.same**

**c.sorted**

**d.constant**

**answer:a**

**14.Which one of the following is a set of one or more attributes taken collectively to uniquely**

**identify a record:(2018,2023)**

**a.candidate key**

**b.sub key**

**c.super key**

**d.foreign key**

**answer:c**

**15.Tables in Second Normal Form:(2NF) (2018,2023)**

**a.eliminates all hidden dependencies**

**b.eliminates all possibilities of insertion anamolies**

**c.have a composite key**

**d.have all non-key fields depend on the whole primary key**

**answer:d**

**16.For each attribute of a relation, there is a set permitted values, called the\_\_\_\_\_\_ of that attribute. (2023)**

**a.Domain**

**b.Relation**

**c.Set**

**d.Schema**

**answer:a**

**17. An attribute in an relation is a foreign key if the \_\_\_\_\_\_ key from one relation is used as an attribute in that relation.(2023)**

**a.Candidate**

**b.Super**

**c.Primary**

**d.Sub**

**answer:c**

**18.Which forms simplifies and ensures that there is minimal data aggregates and repetitive groups:(2023)**

**a.1NF**

**b.2NF**

**c.3NF**

**d.All of the mentioned**

**answer:c**

**19.Several entities are associated through (2023)**

**a.connection**

**b.relationship**

**c.agreement**

**d.integration**

**answer:b**

**20.A collection of tables to represent data and relationship among data is represented through model----------(2023)**

**a.ER model**

**b.Relational model**

**c.semistructered model**

**d.object based model**

**answer:b**

**1.Define function dependency!**

**List the Armstrong axioms and state the below statement**

**{WX->Y, X->Z , Z->WY} |= {X->Y}**

**ANSWER:**

**A functional dependency x->y, between two sets of attributes specifies a constraint that any two tuples t1 and t2 that have t1[x]= t2[x] ,they must also have t1[y] = t2[y]**

**Armstrong's Axioms:**

**1.The Reflexive Rule:If y is a proper subset of x ,then x->y**

**2.The Augmentation Rule:{x ->y} |= xz -> yz**

**3.The Transitivity Rule:{x->y,y->z} |= x->z**

**demonstration:**

**1.X->Z**

**2.Z->WY**

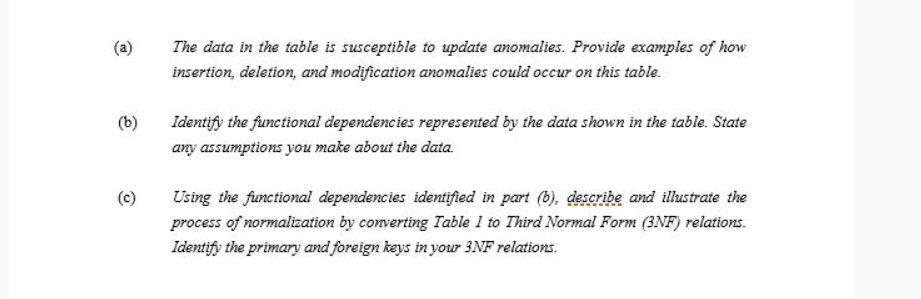
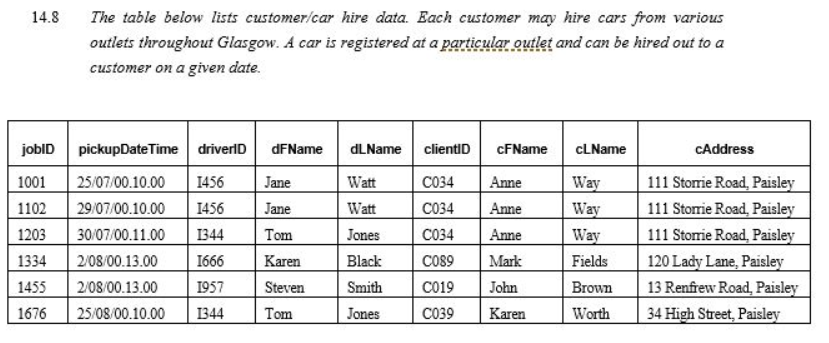
**3.x->wy (transitivity)**

**4. y is a subset of w . wy -> y (reflexivity)**

**5.{x->wy, wy->y} |= x->y (transivity)**

Demonstration:  
1. {X→Z} |= WX→WZ (augmentation)  
2. {Z ⃀ WZ} |= WZ → Z (reflexivity)  
3. {WX→WZ, WZ→Z} |= WX→Z (transitivity)

**2.solve the question**



**Solution (a):**

Insertion Anomalies: If a new driver joins the outlet, then driver cannot be added to database if no rides are done by him

Update Anomalies: If client address is updates then that needs to be chaged in all the tuples.

Deletion Anomalies: If some ride detail is deleted and that is the only ride taken by client or delivered by driver, then driver and client details also get deleted.

**Solution (b):**

driverID-> dFName, dLName

clientID-> cLName, cFName, cAddress

jobID -> pickupDateTime, driverID, clientID

**Solution (c):**

Primary Key: jobID

Thus driverID-> dFName, dLName and clientID-> cLName, cFName, cAddress are partial dependencies. Thus tables is not in 2NF.

Table T-1( jobID, pickupDateTime, driverID, clientID)

Table T-2( driverID, dFName, dLName)

Table T-3(clientID, cLName, cFName, cAddress)

Since no transitive dependencies exist, tables T-1, T-2 and T-2 are in 3NF.

**3.Given the relational schema: 4 points**

**Jedi-Tamas (master, apprentice)**

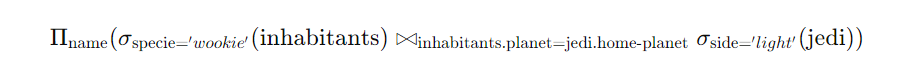
**jedi(name,side,home-planet)**

**government(leader , planet, position)**

**inhabitants(specie, planet)**

**write and algebraic expression which will query(by listing their names) to find all “wookie” (which is a specie) jedi on the light side of the force!**

**[for half marks write an sql query]**

****

**4. From an SQL user’s perspective , does relational model provide logical and physical independence?why? 2 points**

**yes.**

the relational model provides both logical and physical independence.

**Logical Independence**: Users can change the logical schema (e.g., add new fields or tables) without affecting the application programs.

**Physical Independence**: Users can change the physical storage (e.g., how data is stored or indexed) without affecting the logical schema or application programs.

**5.Normalization - Solved exercise**

**Question:**  
  
Consider a relation Movies\_Screened with attributes Theatre, Movie, Day, Time, and Certificate. Sample tuples are as follows:

***Sathyam, 'Slumdog Millionaire', Wed, 18:00, 15***

***Sathyam, 'Slumdog Millionaire', Wed, 20:00, 15***

***PVR, 'Slumdog Millionaire', Wed, 20:30, 15***

***PVR, 'Vicky Christina Barcelona', Wed, 20:30, 12A***

Each movie is assigned a certificate by the Indian Board of Film Certification; the certificate value 15 means that nobody younger than 15 years of age can see this movie in a cinema. The same theatre can show a movie on multiple times during a day, and may show different movies at the same time (on different screens).

1. Does this relation violate the second normal form requirements? Explain.

(b) Decompose this relation into BCNF, and explain why the resulting relations are in BCNF.  
  
**Answer (a):**

To check for 2NF, first we need to find the candidate keys for MOVIES\_SCREENED.

Let us find the functional dependencies (FDs) of MOVIES\_SCREENED.

* THEATRE cannot determine any attributes as a theatre screens more than one movie, it screens on different days, different timings, and different certification movies.

* MOVIE can determine the CERTIFICATE value as a movie will be given only one certificate. Hence, we can include MOVIE **→** CERTIFICATE.

* Likewise, DAY, TIME and CERTIFICATE cannot determine the other attributes uniquely.

We get the set of FDs for this relation as follows;

F = { MOVIE **→** CERTIFICATE, (THEATRE, MOVIE, DAY, TIME) **→** CERTIFICATE }

To find the candidate key, we need to find the closure of left hand side attributes of the FDs.

(THEATRE, MOVIE, DAY, TIME)+ = THEATRE, MOVIE, DAY, TIME, CERTIFICATE.

Hence, the composite key (THEATRE, MOVIE, DAY, TIME) is the candidate key for the relation MOVIES\_SCREENED.

**To be in 2NF, a relation should not have partial functional dependency**.

In our relation, a non-key attribute CERTIFICATE is determined by MOVIE, which is part of a candidate key (THEATRE, MOVIE, DAY, TIME). So the given relation is not in 2NF.

**The relation MOVIES\_SCREENED violates second normal form**.   
  
**Answer (b):**

As discussed, the relation violates 2NF. To normalize to 2NF, we decompose the the relation using the violating functional dependency MOVIE **→** CERTIFICATE.

It results in the following relations;

Movie\_Screens (THEATRE, MOVIE, DAY, TIME)

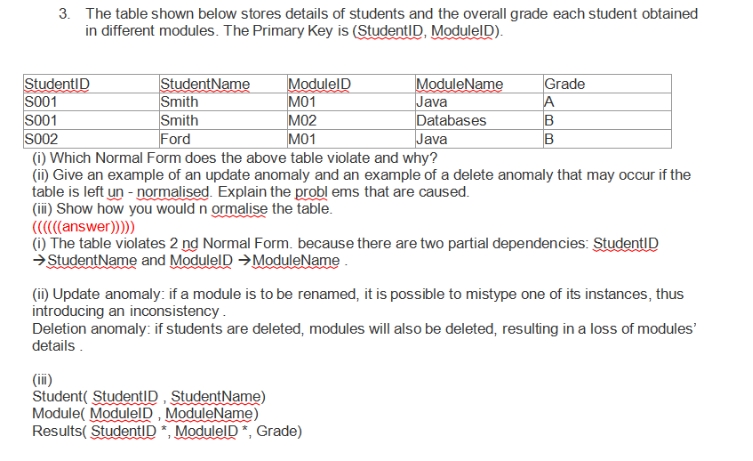
Movies (MOVIE, CERTIFICATE).

Both relations are in 2NF because no partial dependency exists [see the keys underlined].

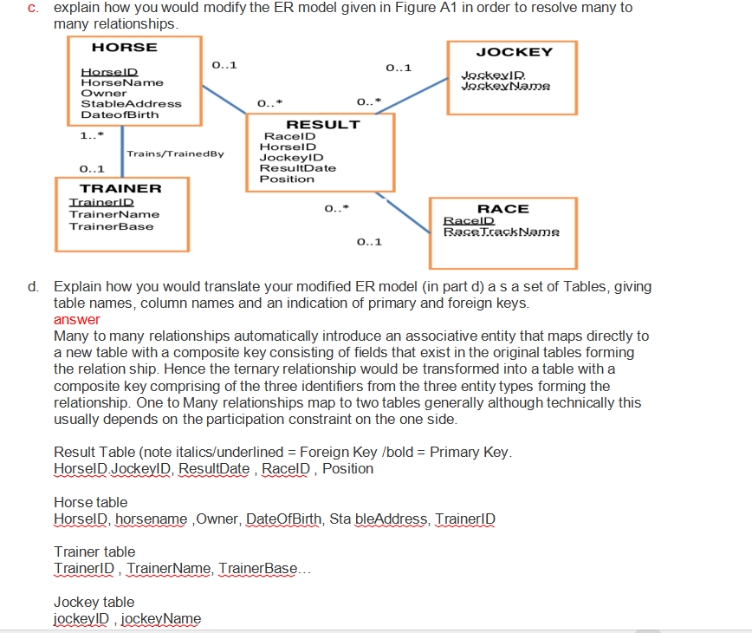
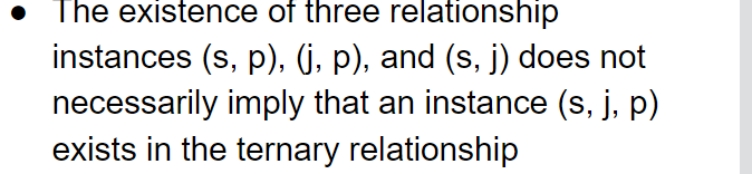
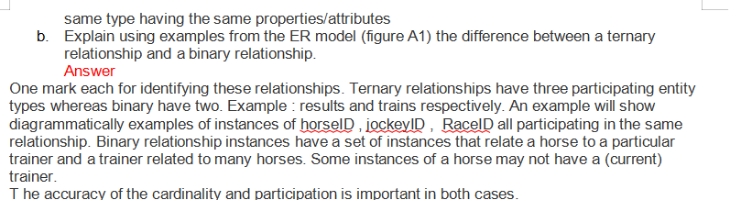
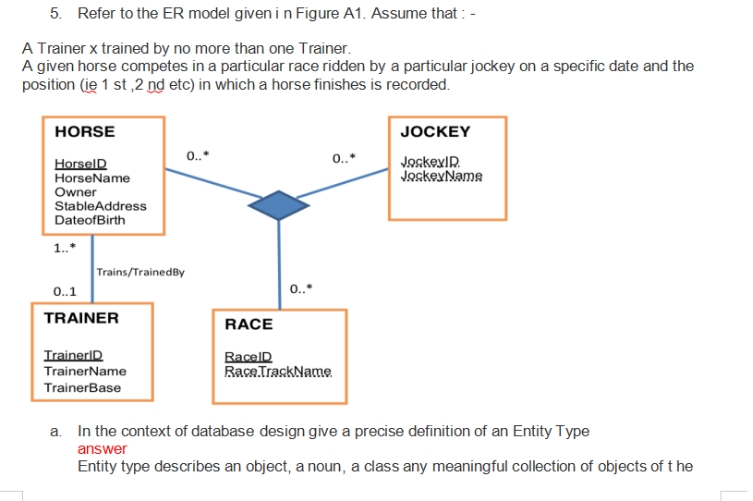
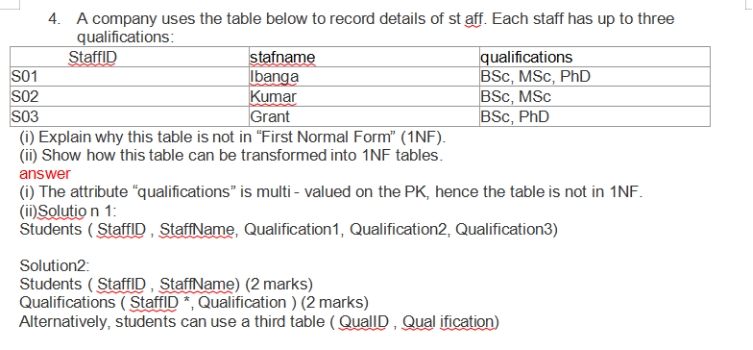
Both relations are in 3NF too because no transitive dependencies found.

Also, both are in BCNF because in the Movie\_Screens relation, no subset of the attributes determines any other attribute, and the only non-trivial dependency in MOVIES is from MOVIES to CERTIFICATE.

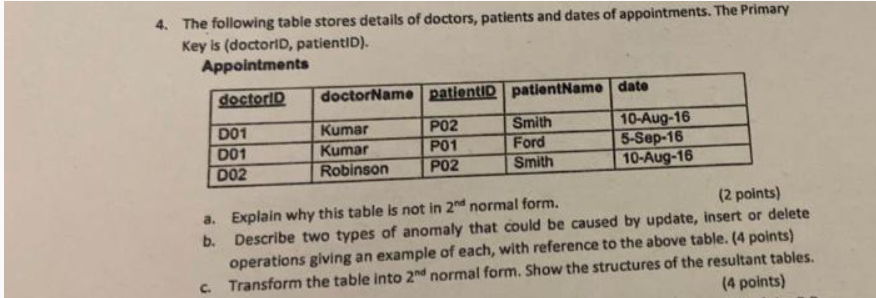
6.solve the normalization question.



**7.solve the problem**



8.solve the normalization question:



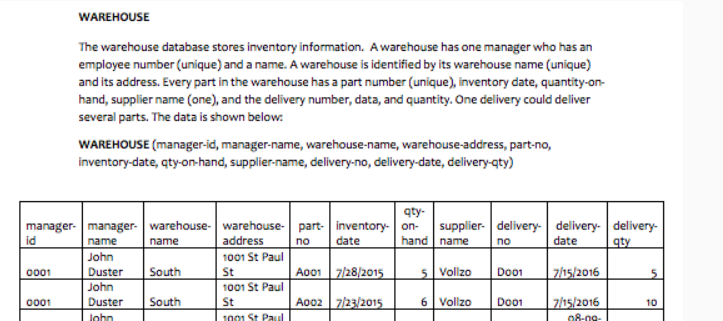
1. there is partial dependencies like doctorID -> doctorName
2. deletion anamoly:If we delete the appointment of Smith , we will lose the detail of a doctor

update anamoly:if we update patient name , we need to update in two rows otherwise the data will be inconsistent

c.doctor(doctorID,doctorName)

patient(patientId, patientName)

Appintment(doctorID\*, patientID\*, date)

**3. WAREHOUSE Database:**

**.**

**Given details:**

**.**

WAREHOUSE database has below details in it:

a. Warehouse manager: manager-id, manager-name

b. warehouse details: warehouse-name, warehouse-address

c. part details: part\_no, inventory-date, quantity-on-hand

d. supplier details: supplier-name

e. part-supply details: delivery-no, delivery-date, many parts(delivery-qty, part-no)

.

**Functional Dependencies:**

**.**

Below is the list of functional dependencies in the given details:

a. manager\_id -> manager\_name

b. warehose\_name-> warehouse\_address, manager\_id

c. part\_no -> supplier\_name

d. warehose\_name, part\_no -> inventory-date, quantity-on-hand

e. delivery\_no, part\_no->delivery-qty, delivery-date

.

**Normalized tables:**

**.**

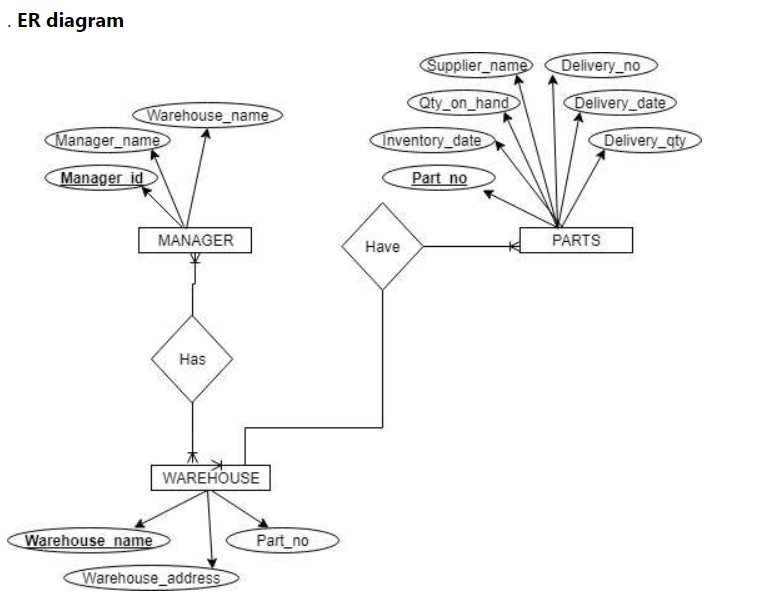
a. WarehouseManager(manager\_id(fk), manager\_name)

b. Warehouse(warehose\_name(pk), warehouse\_address, manager\_id(fk))

c. Part(warehose\_name(pk+fk),part\_no(pk), inventory\_date, quantity\_on\_hand, supplier\_name)

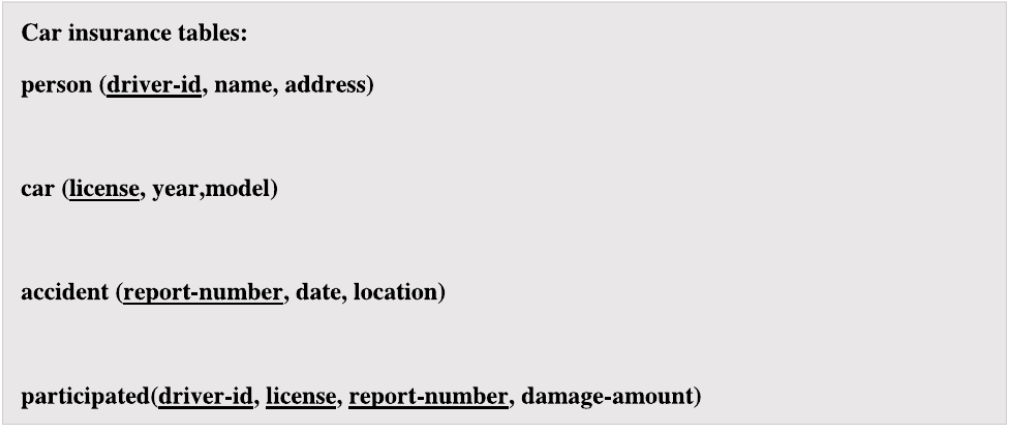
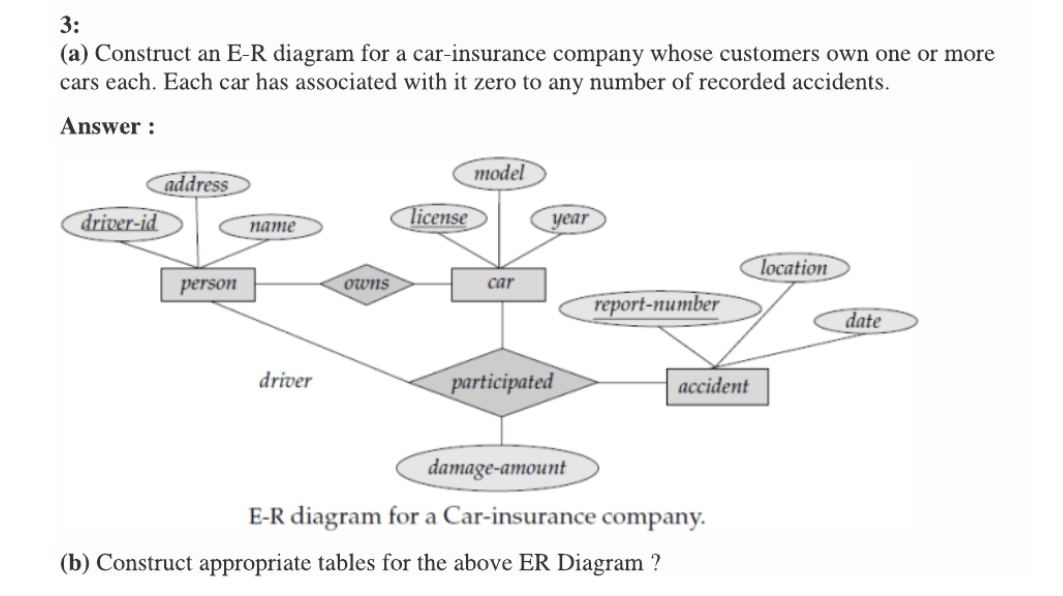
d. Delivery(delivery\_no(pk), part\_no(pk)(fk), delivery-qty)

.



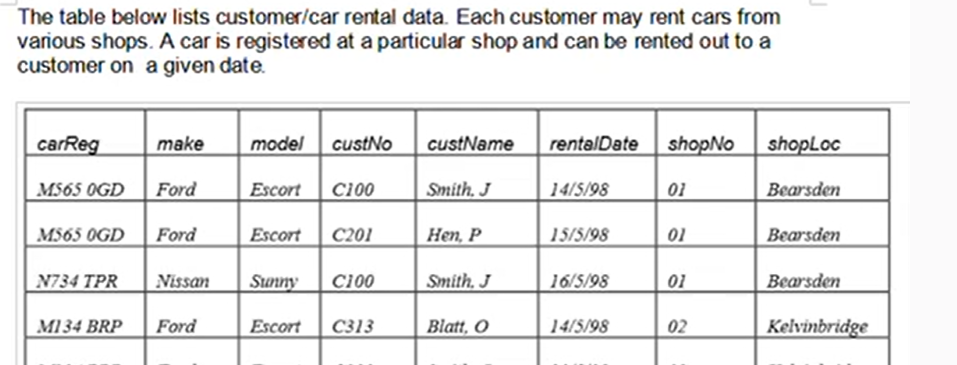
This is asked in 2024 22nd may.

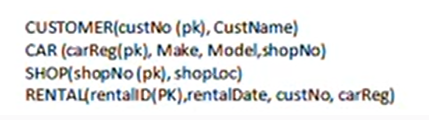
**9.solve the ER diagram**



this was asked several times

**10.solve the normalization problem**





**11. CD Database:**

.

CD database has below details:

CD Details: including ID, Title, Year\_Published, Year\_purchased

Artist Details: Artist name where the name can repeat for Many CD’s, thus need to have Identifier for an artist

Genre Details: Genre details where same genre can have many CD’s thus need to have an identifier for Genre

Track Details: Track details including title, length and adding an identifier as it is assumed that various CD’s can have same track name(title). track table will be separated.

.

**Functional Dependencies:**

**.**

Below is the list of functional dependencies in the given details:

a. ArtistID-> Name

b. GenreID ->Name

c. ID -> Title, Year\_Published, Year\_purchased, Artist\_id, GenreID

d. TrackID-> title, length, CDID

.

**Normalized tables:**

**.**

The normalized tables are as below:

a. . Artist(ArtistID(pk), Name)

b.  Genre(GenreID(pk), Name)

c. CD(ID(pk), Title, Year\_Published, Year\_purchased, Artist\_id(fk), GenreID(fk))

d. Track(TrackID(pk), title, length, CDID)

.

.

12.solve the normalization problem

An embassy records details of interviews of visa applicants in the table below. Interviews are

       conducted by members of staff in some of the embassy rooms. In any given day, a member of

       staff  tends to use the same room throughout that day. An applicant cannot have two interviews in

       the same day :

R:
applicantNo interview Date interview Time staffNo
AP1
13.5.2016 10:30
S5
AP2
13.5.2016 12:00
S5
AP3
13.5.2016 12:00
S9
AP2

**-** List all possible candidate keys for the above relation :

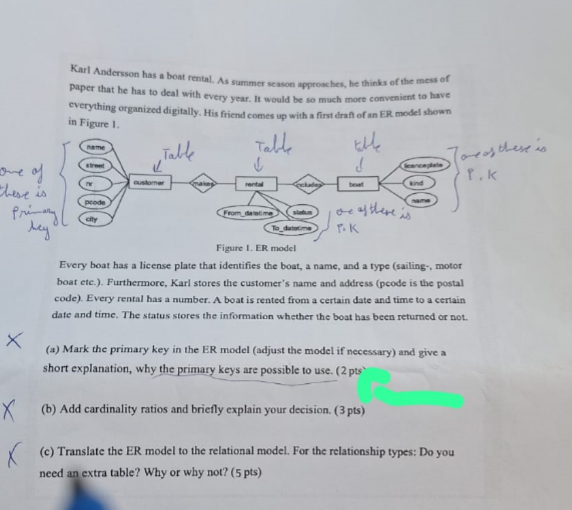
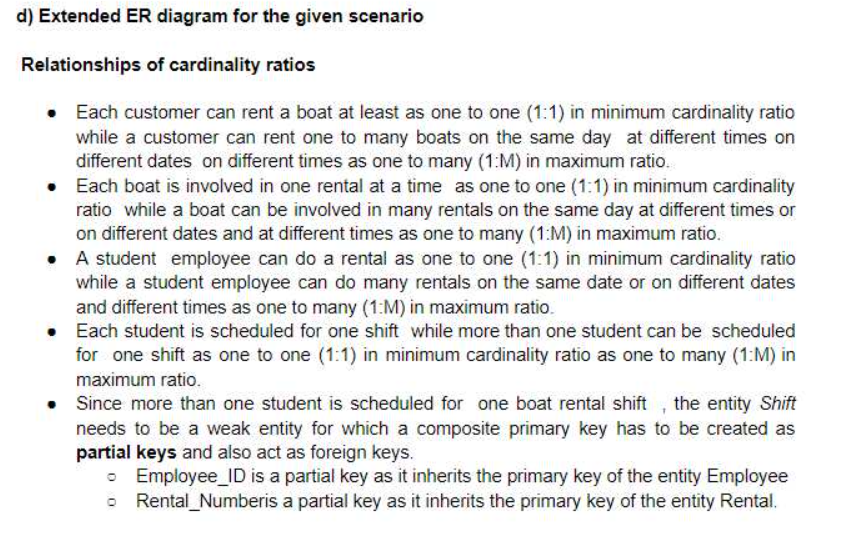
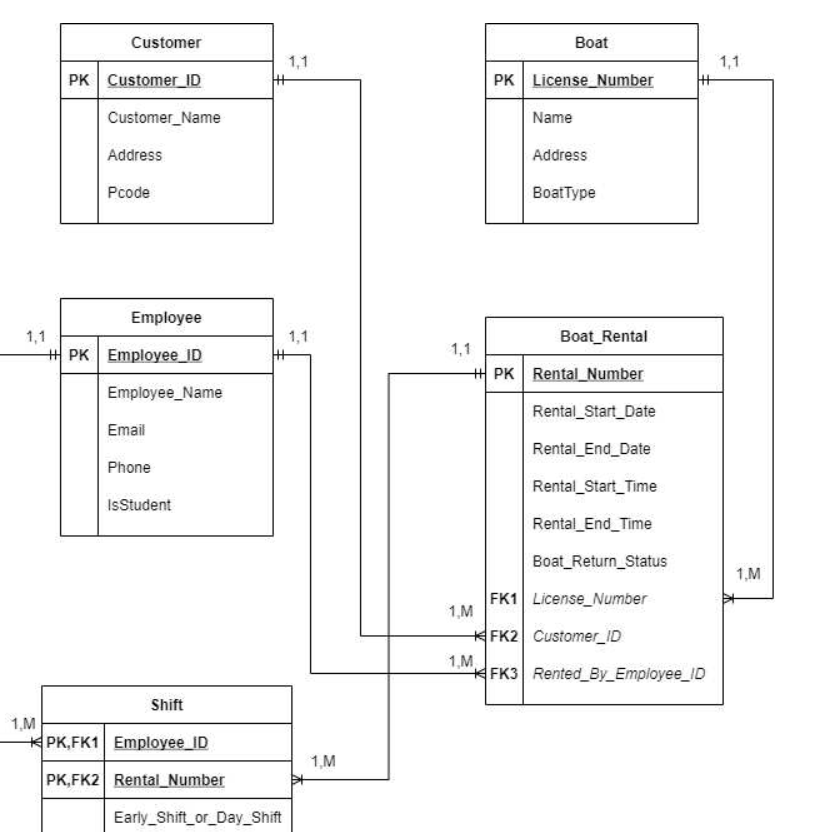
{applicantNo,interviewDate,interviewTime}

{staffNo,interviewDate,interviewTime}

{roomNo,interviewDate,interviewTime}

# 13.Design an E-R Diagram for a marketplace company (such as e-Bay)where individuals can sell and buy products. The ER must be able to keep track of customers who sell items and customers who bought them; some customers may do both. Customer info must include street address, city, state, and zip code. Product info must include name, description, color, and size. Each product sold or for sale must be identifiable, so the E-R must be able to store the sale date if sold, and the current status of the sales process (available, ordered, shipped, delivered). Develop an ER diagram for each of these questions based on Ebay. 1. Define entity sets that contain all data objects that you need. Define attributes and a primary key for each entity set 2. Define relationships between entities that reflect the requirements completely 3. Define total and partial participation in the relationships

14.solve the question

15.Given the relational schema.

Driver(Name,LicenseNumber,Address)

Car(LicensePlate,Type,Manufacturer)

Owner(Licenseplate,LicenseNumber,Year\_of\_purchase)

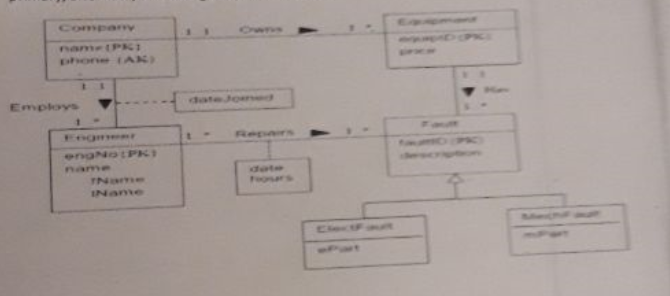
Write a relational algebra expression that gives all the drivers living in Debrecen and have a Lexus car purchased in 2013 [For half mark you can write just an SQL expression.)

solution:

π Name(σ Address='Debrecen' &Type='Lexus' &Year\_of\_purchase=2013 (Driver ⨝ Owner ⨝ Car))

16.Provide a set of relational tables for the high level data model shown below .Identify primary , alternate and foreign key

in the table.



a)

Below is the relational schema of the given high level data model.

It consist of 8 relations. Each relation contains a primary key (mentioned as underlined)  and possible foreign key (as bold).

Relation fault has specilization relations ElectFault and MechFault.

Company (Name, phone)

Equipment (equipID, price)

CompanyEquip (**Name fk**, **equipID**, dateOfPurchase)

Engineer (engNo, fNName, lName,**Name fk**, dateJoined)

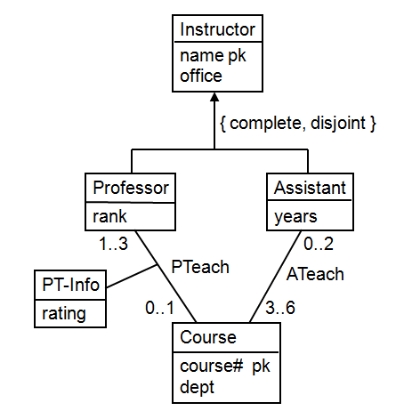
Fault (faultID, description, **equipID fk**)

Repair (**engNo**, **faultID**, date, hours)

ElectFault (**faultID**, ePart)

MechFault (**faultID**, mPart)

17. **Provide a set of relational tables for the high-level data model shown below. Identify primary and foreign keys in the tables. There are several possible automatic translations; use the translation for subclassing most appropriate for the specified properties. (8 points)**



Professor(name,office,rank,course#,rating)  
     Assistant(name,office,years)  
     Course(course#,dept)  
    ATeach(name,course#)

18. **a) In the database context, what do we mean by redundant data? (1 point)  
b) Why might it be a good idea to have redundant data in a database? (2 points)  
c) Why might it be a bad idea to have redundant data in a database? (2 points)**

a) Redundant data in a database refers to the repetition of the same data in multiple places. This duplication can occur within a single table or across multiple tables, leading to multiple copies of the same piece of information.  
b) Data Availability: Redundancy can increase data availability and reliability. If one instance of the data becomes corrupted or unavailable, other copies can still be used, enhancing fault tolerance.  
c) Data Inconsistency: Redundant data can lead to data inconsistency. If one copy of the data is updated while others are not, it can result in conflicting information within the database. This makes maintaining data integrity more challenging.

19. **Consider the following relation for published books:  
BOOK (Book\_title, Author\_name, Book\_type, List\_price, Author\_affil, Publisher)  
Author\_affil refers to the affiliation of author. Suppose the following dependencies exist:**

**Book\_title → Publisher, Book\_type  
Book\_type → List\_price  
Author\_name → Author\_affil**

**a)What normal form is the relation in? Explain your answer. (2 points)  
b) Apply normalization until you cannot decompose the relations further. State the reasons behind each decomposition. (4 points)**

Given relation is

BOOK(Book\_title, Author\_name, Book\_type, List\_price, Author\_affil, Publisher)

Given functional dependencies are:

Book\_title -> Publisher, Book\_type

Book\_type -> List\_price

Author\_name -> Author\_affil

a.

- The given relation is in 1NF (1st Normal Form)

- The keys here are Book\_title and Author\_name

- As no fully functional dependent attributes exists on the key, it is not in 2NF (2nd Normal Form)

- It is not in 3NF (3rd Normal Form) because the keys are partially functional dependent.

b.

- Decompose to 2NF:

- Book\_0 (Book\_title, Author\_name) => Book\_title, Author\_name is key

- Book\_1 (Book\_title, Publisher, Book\_type, List\_price) => Book\_title is key

- Book\_2 (Author\_name, Author\_affil) => Author\_name is key

- Now the relation is in 2NF as it eliminates the partial dependencies.

- Decompose to 3NF:

- Reasons for the relationship not in 3NF:

- Book\_title -> Book\_type -> List\_price

- Book\_type is not a key as well not a subset of a key

- List\_price is not a prime attributes

- Decomposition:

- Book\_0 (Book\_title, Author\_name)

- Book\_11 (Book\_title, Publisher, Book\_type)

- Book\_12 (Book\_type, List\_price)

- Book\_2 (Author\_name, Author\_affil)

- Now the transitive dependencies on List\_price is eliminated

**20. Consider the relation REFRIG (Model#, Year, Price, Manuf\_planet, Color), which is abbreviated as REFRIG (M, Y, P, Mp, C), and the following set F of functional dependencies:**

**F = { M→ Mp , { M , Y } → P , Mp → C }**

**a) Evaluate each of the following as a candidate key for REFRIG, giving reasons why it can or cannot be a key: {M}, {M, Y}, {M, C}.   
b) Based on the above key determination, state whether the relation REFRIG is in 3NF and in BCNF, giving proper reasons.   
c) Consider the decomposition of REFRIG into:  
D = {R1 (M, Y, P); R2 (M, MP, C)}.  
Is this decomposition lossless? Show why!**

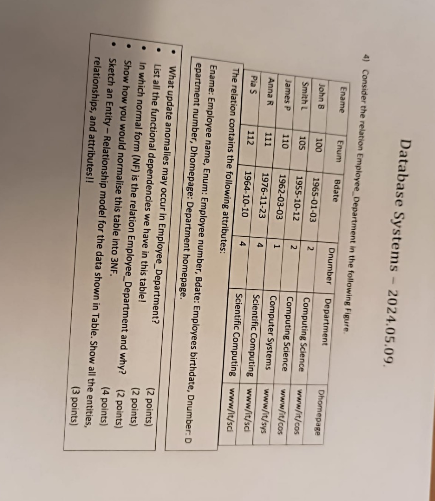
**Answer:-------------**

**a.  
i) {M} is not a candidate key** since it does not functionally determine attributes Y or P.  
**ii) {M, Y} is a candidate key** since it functionally determines the remaining attributes P, MP, and C. From F it is clear that MP and P are functionally determined by M andY. By transitivity (M -> MP and MP -> C) we also know that M -> C.  
**iii) {M, C}** **is not a candidate key** since it does not functionally determine attributes Y or P.

**b.**  
BCNF can be directly tested by using all of the given dependencies and finding out if the left hand side of each is a super key. Take the two FD’s M -> MP and MP -> C, since neither M nor MP is a super key, we can conclude that **REFRIG is is neither in 3NF nor in BCNF.**  
Alternatively, REFRIG is not in 2NF, because {M,Y} is a key. So, we can see here, Partial Functional dependency exists i.e ( M -> MP ). So it is not also  in 2NF,  
Therefore, **Given relation is in 1NF.**

c).   
**Given the decomposition of REFRIG into D:  
R1 (M, Y, P)  
R2(M, MP, C)**  
Using the test for Binary Decomposition, we calculate:  
(R1∩ R2) = {M} ;(R1 – R2) = {Y, P};(R2 – R1) = { MP, C}  
Then, we have: {M} -> {MP} (given) {M} -> {C}   
{M} -> {MP} and {MP} {C}) {M}-> {MP, C}  
**So,(R1∩ R2)** -> **((R2 – R1) or the decomposition D is lossless.**

**21.solve the question**



a)If we update the Dhomepage of computer science dept. we need to update in two tuples ,otherwise the data will be inconsistant.

b){Enum} -> {Ename,Bdate}

{Dnumber} -> {Department,Dhomepage}

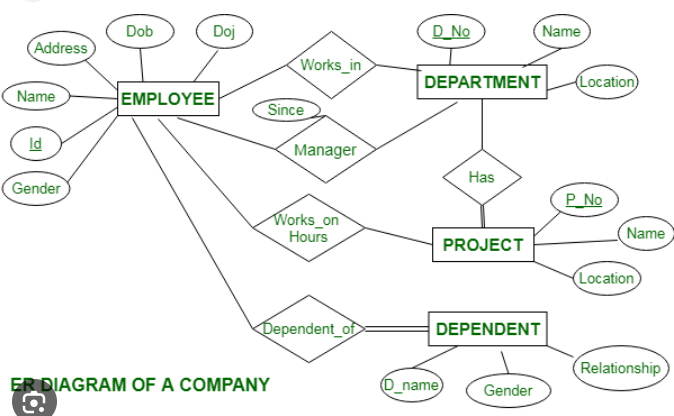
c)It is in 1NF because all attributes are simple and atomic

not in 2nf because PK(Enum,Dnumber) where Enum -> Ename  
so partial dependency

d)Employee(Enum,Ename,Bdate,Dnumber\*)

Department(Dnumber,Department,Dhomepage)

e)



22. **The following table stores details of doctors, patients and dates of appointments. The Primary Key is (doctorID, patientID).**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| doctorID | doctorName | patientID | patientName | date |
| D01 | Kumar | P02 | Smith | 10-August |
| D01 | Kumar | P01 | Ford | 15-September |
| D02 | Robinson | P02 | Smith | 10-August |

**a.Explain why this table is not in 2NF. (2 points)  
b. Describe two types of anomalies that could be caused by update, insert or delete operations giving an example of each, with reference to the table above. (4 points)  
c. Transform table into 2NF. Show the structures of the resultant tables. (4 points)**

**a.The table is not in 2NF because here primary key is {doctorID,patientID} where doctorID**

**-> doctorName. So here is partial dependency**

**b.Deletion Anamoly: If we want to delete the appointment date of Smith the doctors detail will also be lost**

**insert anamoly:If we want to insert a new doctor it will not be possible unless he has an appointment**

**c.Doctor(doctorID pk,doctorName)**

**Patient(patientID pk, patientName)**

**Appointment(doctorID,patientID,date)**

**23.Some Extra MCQs**

**QUESTION 3..Entity A is a\_\_\_\_\_as it can not exist in the database unless another type of entity also exist in the database**

 A. Weak Entity.  
B. Strong Entity.  
C. Entity.  
D. None of the given.

**Ans:-- Option A.  (Weak Entity).**

**Explanation:--Weak entity** cannot exist in the database unless another type of entity also exists in the database, but does not require that the identifier of that other entity be included as part of its own identifier.The identifier of the weak entity must be a composite key.

**QUESTION 4...A recursive relationship is a relationship between an entity and\_\_\_\_\_.**  
 A. Strong Entity.  
B. Weak Entity.  
C. Composite Entity.  
D. Itself.

**Ans:---Option D. (Itself).**

**Explanation:--**A recursive relationship is a relationship between an entity and **itself**. In relational databases, **when a table is involved in a relationship with itself**, it is called a recursive relationship. For example, in a supervisor-supervisee relationship, a table storing employee records is involved in a relationship with itself.

**A strong entity** is an entity type whose existence doesn't depend on any other entity.

**A weak entity** is an entity that cannot be uniquely identified by its attributes alone; therefore, it must use a foreign key in conjunction with its attributes to create a primary key.

**Composite Entity.**:-This entity lies between the two entities that are of interest and this composite entity shares the primary keys from both the connecting tables.

**QUESTION 5:--Which of the following is Not True regarding Weak Entity?**  
A. None of the given.  
B. Identifying relationship.  
C. Primary key derived from parent entity.  
D. Has an Existence dependency.

**Ans:--Option:--A..( None of the given).**

**Explanation:---The weak entities have total participation constraint (existence dependency)** in its identifying relationship with owner identity.

**A weak entity has a primary key that contains only one foreign key**. The entity DEPENDENT, for example, is a weak entity because it uses the primary key of the EMPLOYEE entity as part of its own primary key.

In general (though not necessarily) a weak entity does not have any items in its primary key other than its **inherited primary key and a sequence number.**

**The Relationship t**hat connects the weak entity to its owner identity is called Total Participation inthe identifying relationship.

**QUESTION 6...Which of the following is True regarding Associative entity ?**  
A. Called bridge entity  
B. Primary key is a composite attribute  
C. Used to resolve complexity in many to many relationship  
D. All of the given

**Ans:--Option D. (All of the Given).**

**Explanation:---**

Associative entities are implemented in a database structure using associative tables, which are tables that can contain references to columns from the same or different database tables within the same database.

Associative tables are colloquially known under many names, including **association table**, **bridge table**,

. A relational database requires the implementation of a base relation (or base table) to resolve many-to-many relationships. A base relation representing this kind of entity is called, informally, an **associative table**.

**Associative entities don't have a primary key based on their own attributes.An associative entity has the same primary key as its supertype associative entity**, and this primary key is also a foreign key to the supertype associative entity.An associative (or junction) table maps two or more tables together by referencing the primary keys (PK) of each data table. In effect, it contains a number of foreign keys (FK), each in a many-to-one relationship from the junction table to the individual data tables. The PK of the associative table is typically composed of the FK columns themselves.

**QUESTION 7....The attribute AGE is calculated from DATE OF BIRTH. The attribute AGE is:**  
 A. Derived Attribute  
 B. Stored Attribute  
 C. Multivalued Attribute  
 D. Composite Attribute

**Ans:---Option A..  (Derived Attribute).**

**Explanation:--**T**he value for this type of attribute can be derived from the values of other related attributes or entities.**

**A stored attribute** is an attribute that cannot be derived from other attributes.

**A multivalued attribute** of an entity is an attribute that can have more than one value associated with the key of the entity.

The attributes which can be divided into sub-parts are called **composite attributes**

**QUESTION 8...An attribute that identify an entity is called**  
 A. Entity.  
 B. Relationship.  
 C. Composite Key.  
 D. Identifier.

**Ans:--Option D.. (Identifier).**

**Explanation:--**Entity identifiers are **attributes, specifically, key attributes that uniquely identify each entity**.

An **entity**in DBMS (Database management System) is a real-world thing or a real-world object which is distinguishable from other objects in the real world.

**A composite key** is a candidate key that consists of two or more attributes (table columns) that together uniquely identify an entity occurrence (table row).

**Relationships**allow relational databases to split and store data in different tables, while linking disparate data items.

**QUESTION 9..Which of the following indicate the maximum number of records that can be involved in a relationship?**  
 A. Maximum Cardinality.  
B. Maximum Connectivity.  
C. Minimum Connectivity.  
D. Minimum Cardinality.

**Ans:--Option A...(Maximum Cardinality.).**

**Explanation:--Maximum cardinality**is defined as the maximum number of entity classes which are taking part in a relationship. There are three types of Maximum Cardinality: one-to-one, one-to-many, many-to-many. In 1:n , 1 is the minimum cardinality, n is the maximum cardinality.

**QUESTION 10,,In which of the following can one entity instance of one type be related to many entity instances of another type?**  
A. One to Many relationship.  
 B. Many to Many relationship.  
C.Zero to Many relationship.  
 D.One to One relationship.

**Ans:--Option A...( One to Many relationship.)**

**Explanation:--**A one-to-many relationship is a type of cardinality that refers to the relationship between two entities A and B in which an element of A may be linked to many elements of B, but a member of B is linked to only one element of A..

**A many-to-many r**elationship is a type of cardinality that refers to the relationship between two entities, say, A and B, where A may contain a parent instance for which there are many children in B and vice versa.

**A zero to many** optional relationship indicates that a person may have no phone, one phone, or many phones, and that the phone may not be "owned," but can only be owned by a maximum of one person.

**A one-to-one**relationship is a type of cardinality that refers to the relationship between two entities A and B in which one element of A may only be linked to one element of B, and vice versa.

24.