SUMMARY OF MODULE 8

1. A conceptual model of data used in an application is obtained by using an entity relationship model (E-R model).
2. **Entities** specify distinct real world items in an application, **relationships** represent dependencies between entities, and **attributes** specify properties of entities and relationships.
3. Entity set describes items which are similar. Students, courses, teachers are examples of entity sets.
4. Examples of relationship are students **attend** courses, Teachers **teach** courses. "Attend" and "teach" are relationships.
5. E-R models have the facility to describe 1 to 1, 1 to many, many to many, conditional and optional relationships.
6. E-R models assist in designing relational databases.
7. A relation is a table which depicts an entity set. Each column in the relation corresponds to an attribute of the entity. Each row contains a member of the entity set.
8. A relational database consists of a collection of relations relevant for a specified application.
9. Relations in a database are normalized. Normalization is a procedure used to transform a set of relations into another set which has some desirable properties.
10. Normalization ensures that data in the database are not unnecessarily duplicated. It also ensures that addition and deletion of entity rows (or tuples) or change of

individual attribute values do not lead to accidental loss of data or errors in database.

11. Normalization is performed by first converting a relation so that all entities are simple. This results in a relation said to be in First Normal Form (INF).
12. Successive normalization steps are carried out to meet the objectives stated in point 10.
13. The successive normal forms are called 2NF, 3NF, BCNF, 4NF and 5NF.
14. A summary of how normalization steps are carried out is shown in the table below.

Summary of Normalization Steps

Input Relation	*Transformation*	*Output relation*
All relations	Eliminate variable length records. Remove multiattribute lines in table	INF
INF relation	Remove dependency of non-key attributes on part of a multiattribute key	2NF
2NF	Remove dependency of non-key attributes on other non-key attributes	3NF
3NF	Remove dependency of an attribute of a multiattribute key on an attribute of another (overlapping) multiattribute key	BCNF
BCNF	Remove more than one independent multivalued dependency from relation by splitting relation	4NF
4NF	Add one relation relating attributes with multivalued dependency to the two relations with multivalued dependency	5NF

15. Traditionally, application programs were written after designing data files appropriate to the application.
16. As organizations require many application programs and these applications need some common data, it is found that the same data is stored in many files, thereby wasting storage space.
17. Waste of storage space is not the only problem. The same data (such as an address of an individual) may be in different forms in different files, leading to confusion.
18. As data is scattered in many files, it is difficult to correlate them and use them for various applications.
19. When requirements change, the application programs and the entire data organization may have to be changed. Such a change is time consuming and expensive.
20. Instead of designing independent files for different applications, it is better to design a common database which can be used for all current applications and applications expected to be implemented in the future.
21. A database constitutes the data resource of an organization. It incorporates not only data but also relationships between data items. It is modelled not for specific current applications, but all current and potential applications.
22. A Database Management System (DBMS) is a set of procedures that manage a database and provide application programs access to the database in a form requested by the applications programs.

23. A good DBMS allows sharing of data among applications. Sharing of data is possible if application programs can be written without having a knowledge of data representation, location or access. In other words, application programs should be free to have their own view of data by DBMS.

24. Other major objectives of DBMS are to minimize storage of redundant data, maintain consistency of data values, ensure validity of stored data, protect data, prevent unauthorized access, control insertion, deletion and alteration of data and ensure fast access to data required by application programs.

25. To use a DBMS, a conceptual model of data needed for current and future applications is designed. E-R modelling is useful for this. The conceptual model is next converted to a logical model appropriate for the DBMS. For RDBMS, normalised relations constitute the logical model.

26. The physical data model is then designed to ensure good performance. User's view of data for application is primarily the logical view.

QUESTION BANK - MODULE 8

8.1 What is an entity? Give examples of entities.

8.2 What is a relationship? In what way is it different from an entity?

8.3 In what way is an attribute different from an entity?

8.4 What do you understand by a 1 to 1, 1 to many, many to many relationships? Give one example for each.

8.5 What is the difference between a relation and a relationship?

8.6 What is normalization of a relation?

8.7 Why should relations be normalized?

8.8 What is a functional dependency?

8.9 What is a key attribute in a relation?

8.10 What is the difference between a 2NF and a 3NF relation?

8.11 When is BCNF required?

8.12 When are 4NF and 5NF required?

8.13 What is the difference between 4NF and 5NF?

8.14 Develop E-R diagram for the following:

Customer withdraws money from his account

Students write examinations.

Students attend classes

Professors write books

Driver drives a car

8.15 Draw an E-R diagram showing the cardinality for the following:

(i) A bill is sent to a customer. A customer can receive many bills.

(ii) A clerk works in a bank. The bank has many clerks

(iii) A part is used in many products and a product uses many parts.

(iv) Students apply for seats in colleges. Each student can atmost get one seat. A college has many seats. A student can send many applications.

(v) A car is owned by a person. The person can own many cars.

8.16 For Exercise 8.14, obtain relations for each entity. Normalize the relations.

8.17 For the following word statement, obtain E-R diagram and relations. Use any reasonable assumptions. "A machine shop produces many parts which it takes on contract. It employs many machinists who operate any of the machines. A part needs working on only one machine. A record is kept on the quantity of material needed for producing each part. The production of each part is tracked by giving a job number, start time and end time and machinist identifications".

8.18 For the problem on library, periodical management stated in Exercise 4.23 (Module 4), obtain E-R diagrams and relations. Make any reasonable assumptions and state the assumptions.

8.19 For the problem statement of Exercise 5.15 (Module 5) obtain the E-R diagram and relations for the problem.

8.20 For the problem statement of Exercise 5.16 (Module 5), obtain the E-R diagram and a set of relations by using the diagram

8.21 For the problem statement of Exercise 5.18(Module 5), obtain the E-R diagrams and relations .

8.22 What are the advantages and disadvantages of systems using separate data files?

8.23 What do you understand by the term *data integrity?*

8.24 If redundant data is stored can it lead to data integrity problem?

8.25 Student's records in a University are kept by various sections: Hostel, Health Centre, Academic Office, major departments, Accounts Section and Library. If each of these sections maintains its own file-based system for processing, what problems do you foresee? Give examples.

8.26 Define a database of an organization.

8.27 What is the difference between a database and a Database Management System (DBMS)?

8.28 What are the basic objectives in evolving a database for an organization?

8.29 What do you understand by the term *data independence?*

8.30 What advantages are available to a programmer from data independence?

8.31 Draw an analogue between the advantages of high level language programming and data independence of application programs.

8.32 How can data integrity be maintained in a database?

8.33 Distinguish between issues of privacy and security in a database.

8.34 What is the role of E-R diagrams in database design?

8.35 What is the difference between a conceptual model and a logical model of a database?

8.36 What is an internal model of a DBMS?

8.37 What data models are used by application programs in a database oriented system?

8.38 Why is a DBMS divided into three layers, namely, conceptual model, logical model, and internal model?

8.39 How is data independence of application programs ensured in a DBMS?

8.40 What is RDBMS?

8.41 What are the responsibilities of a Database Administrator (DBA) in an organization?

8.42 What are the ideal features of a database system?

8.43 Are DBMS relevant to Personal Computers (PCs)?

Module 8

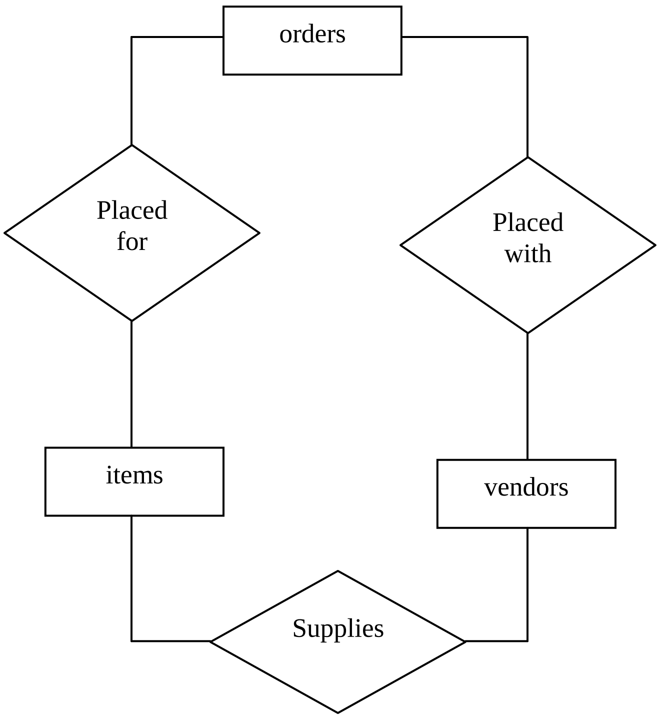
Mini Case Example 1

We will illustrate the method used to obtain an E-R diagram with an example. Usually an imprecise statement of the needs of an organization is given by the management. We now give a sample of such a statement. This is to be used to obtain an E-R diagram.

User's narrative of requirements. "Our company receives a number of items from many vendors and they are received at the receiving office. As we receive over 1000 items a day it is now virtually impossible for the receiving office to check whether the vendor has supplied items against an order, or sent a wrong item and inform the purchase office. We are also not able to find out if there are excesses or deficiencies in delivery and whether the vendor adhered to the delivery schedule as per the order. The items received at the receiving office are sent for physical inspection. The physical inspection consists of checking whether the quantities stated in the delivery note agree with the physical count, whether the item is the correct one ordered, and a check on the quality of item. We would like to keep a record of rejections due to bad quality, incorrect items, excess/deficient supply etc., determined during inspection. This will enable us to assess vendors' reliability to guide us in placing orders in the future, besides keeping track of supplies. Items cleared by the inspection office are taken into the inventory by the stores office which keeps a ledger of items stocked and quantity available of each item. Customers send requisitions to the stores. The stores fulfill the requests based on availability and update the ledger. Currently we are not able to meet some of our customers' requests. We would like to incorporate automatic reordering by our purchase office if the inventory level of an ite

We would also like to keep track of unfulfilled requests and meet them when items reach the store. Currently we are not able to pay our vendors promptly due to delays in payment order reaching our accounts office. We would like to rectify this. We would also like to bill our customers promptly and keep track of customers' payments".

The E-R diagram for the problem is:



The relations are:

ORDER (**order no.**., order date)

ORDER PLACED FOR (**order no**., **item code**, qty. ordered, delivery time allowed)

ORDER PLACEDWITH (**order no.**. **vendor code**, **item code**)

VENDOR (**vendor code**, vendor name, vendor address)

ITEM (**item code**, item name, price/unit)

SUPPLIES (**vendor code, item code, order no**., qty.supplied, date of supply).

The key attribute(s) are in bold letter(s) in each relation.

Let us examine whether the relations are in normal form. ORDERS and ORDER PLACED FOR are simple relations. In the relation ORDER PLACED WITH, the key is the composite attributes *order no., vendor code* and *item code.* However, the entity *item code* is not needed in the ORDER PLACED WITH relation. Given an **order no**., all the items supplied against this order can be found from ORDER PLACED FOR relation. The vendor with whom the order has been placed can be found from the ORDER PLACED WITH relation given below as each order no., has only one vendor.

ORDER PLACED WITH (**order no**., **vendor code**)

The two relations ORDER PLACED WITH and ORDER PLACED FOR have composite keys. The non-key fields are not related to one another. In a key more than one attribute, the individual attributes are not functionally dependent. Thus these two relations are in normalized form and do not need any further change. The relations VENDOR and ITEM are simple and are in normalized form. The relation SUPPLIES is, however, not normalized. **vendor code** and **order no**., are functionally dependent. There is a multivalued dependency between vendor code and item code as a vendor can supply many items. We thus split the relations into two relations.

ACTUAL VENDOR SUPPLY (**order no**., **item code**, qty.supplied, date of supply)

VENDOR SUPPLY CAPABILITY (**vendor code, item code**)

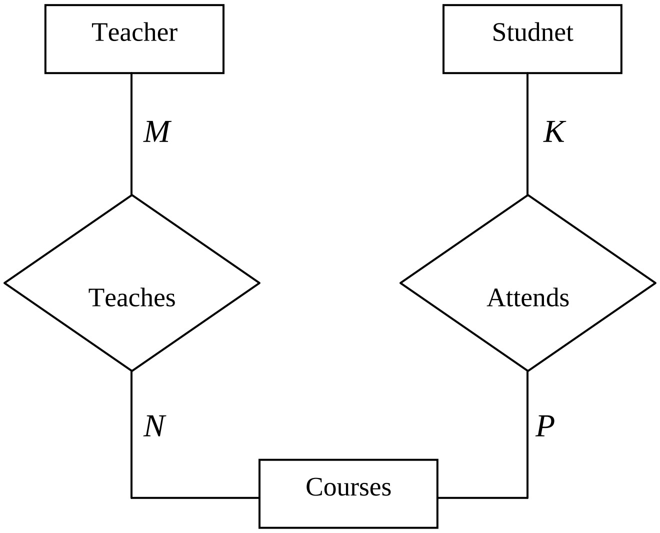
Observe that the relation VENDOR SUPPLY CAPABILITY will have a (**vendor code**, **item code**) table without a vendor having supplied any item. The relation ORDER PLACED WITH will have a tuple only when a vendor actually supplies an item.

Minicase Example 2

Let a database contain the following: Teacher code, Teacher's name, Teacher's address, rank, department, courses taught by the teacher, course name, credits for course, no.of students in the class, course taught in semester no., student no., name, dept., year, and courses taken in semesters no. The following information is given on dependencies.

- A teacher may teach more than one course in a semester
- A teacher is affiliated to only one department
- A student may take many courses in a semester
- The same course may have more than one section and different sections will be taught by different teachers.
- A course may be taught in more than one semester

An entity relationship diagram for this problem is given below. The relations corresponding to the E-R diagram are:



An E-R diagram for teacher database

TEACHER (**Teacher code**, teacher's name, teacher's address, rank, dept.)

TEACHES COURSES (**Teacher code, course no**., **semester no.,** no.of students, section no)

COURSE (**course no., semester no.,** course name, credits)

STUDENT (**student no**., student's name, dept., year)

STUDENT –COURSES (**student's no**., **course no**., **semester no**.)

TEACHER relation has only one key. All non-key attributes are functionally dependent only on the key. There is no functional dependency among non-key attributes. Thus the relation is normalized in 3NF (No higher NFs are applicable).

STUDENT relation is also, similarly, in 3NF. In the COURSE relation, course name could also be a key. The relation is in 3NF and no further normalization is required. The relations TEACHES COURSES and STUDENT-COURSES have multiattribute keys, but the relations themselves are in normal form. The only point which is not clear, from these relations, is the relation between teacher and student. This has been missed in the E-R diagram. The relationship is between the teacher, courses taught and students. In other words, we should be able to answer the question “which teacher is teaching course no.X to student no. Y in Semester 2?”. Let us add a relation.

TEACHER-STUDENT (**Teacher code, student no**., **course no**.)

In this relation **Teacher code** and **course no.** have a multivalued dependency. Similarly, **Teacher code** and **student no.** as well as **student no.** and **course no.** have multivalued dependency. However, TEACHES COURSES (**Teacher code, course no**.,**semester no.**., no.of students, section no.) and STUDENT COURSES (**student no**., **course no**., **semester no**) relations are already in the database. Thus the relation TEACHER-STUDENT as it is specified above is sufficient to give the idea that student Y takes course X from Teacher Z.

References

1. Most of the material in this module has been adapted from the book “Analysis and Design of Information Systems”, 2nd Edition, by V.Rajaraman. Chapter 10 Logical Data Base Design, pp.130 to 153. Chapter 12 , Data Base Management Systems, pp. 171 –1719.
2. Good treatment of E-R Modelling and Normalization may be found in the book “Introduction to Systems analysis and Design”, by I.T.Hawryszkiewycz, Prentice Hall of India, 1989.
3. For those interested in detailed study of Data Base Design and DBMS, the following books are standard texts:

   1, R.Elmasri and S.B.Navathe, Fundamentals of Data Base Systems, 4th Edition, Pearson Education Asia, New Delhi, 2004.

   2. T.Connolly and C.Begg, Dta Base Systems, 3rd Edition, Pearson Education Asia, New Delhi, 2003.