

MODULE - I

Applications of Database System.

- 1) Banking 5) Finance
- 2) Airlines 6) Sales
- 3) Universities 7) Manufacturing
- 4) Credit card 8) Human Resource
- 5) Telecommunication

Database :- One or more tables which are interrelated data.

Data :- Known facts that can be recorded and have an implicit meaning.

Mini-world :- Some part of the real world about which data is stored in a database. For example, student grades and transcripts at a university.

Database Management Systems (DBMS) : A software package / system to facilitate the creation and maintenance of a computerized database.

Database Systems : The DBMS software together with the data itself. Sometimes, the applications are also included.

DBMS is a general purpose software

system that facilitates the process of

- 1) Defining — specify the data type; structure and constraints for the data to be stored in database. → in gmail (mail id)
- 2) Constructing — storing the data into storage medium. mainly 2^o mly
- 3) Manipulating the database. — Querying the database to retrieve specific data, update the data base to reflect the changes, generating reports from the database fields.

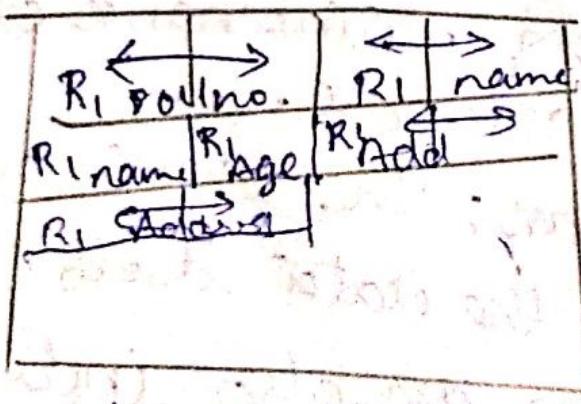
Record

	ROLLNO	Name	Age	Address
R ₁	1	XYZ	12	—
R ₂	2	AMN	20	—
R ₃	3	ABC	20	—
R ₄	4	DEF	29	—

32 bits

catalog / metadata

Fields	Byte
ROLL-no	2
Name	3
Age	1
Address	4



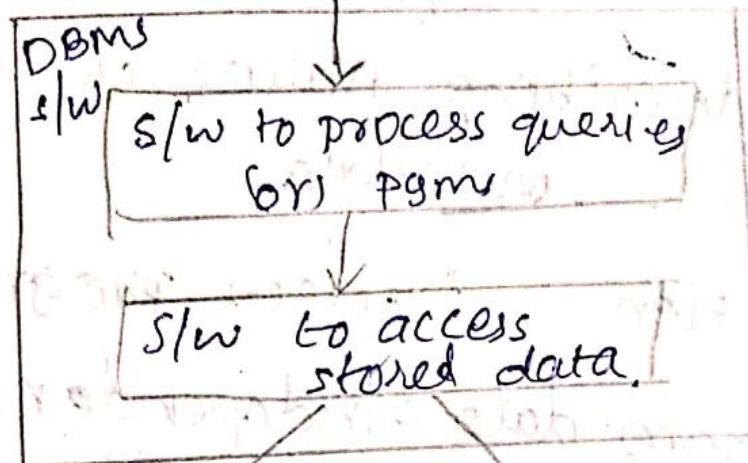
Vertical fields

Metadata → Data about database

User / programmer

Database
system

App. Pgms / queries



Data redundancy - same data at different places

Diff. btw file system & Data base.

Additional functionality of DBMS

- 4) Concurrent processing and sharing by a set of users and programs - yet, keeping all the data valid and consistent.
- 5) Protection or security measures to prevent unauthorized access
- 6) "Active" processing to take internal

actions on data.

7) Presentation and visualization of data.

Main characteristics of the Database

Approach

- Self-describing nature of a database system — metadata.
- Insulation between program and data
 - a) Program data-independence
 - b) Program operation independence.
- Data abstraction: A data model is used to hide storage details and present the users with a conceptual view of the database.
- Support of multiple views of the data: Each user may see a different view of the database, which describe only the data of interest to that user.
- Sharing of data and mult-user transaction processing.

A transaction is an executing program or process that includes one or more

database accesses, such as reading or updating or writing of database record.

Properties of Transaction

- 1) Atomicity - Either all the database operations in a transaction are done or none.
- 2) Consistency.
- 3) Isolation
- 4) Durability

Difference file systems & database systems.

File Management System	Database Management system
→ File system is a general, easy-to-use system to store general files which require less security and constraint.	→ Database management system is used when security constraints are high.
→ Data Redundancy is more in file management system.	→ Data Redundancy is less in database system.
→ Data inconsistency is more in file system.	→ Data consistency is less in database system.
→ Centralisation is hard to get when it comes to file management systems.	→ Centralisation is achieved in database management system.
→ User locates the physical address of the files to access data in file management system.	→ In database system, user is unaware of Physical address where data is stored.

- Security is low in file management system
- Database Management system stores structured data which have well defined constraints and interrelation.
- File management system stores unstructured data as isolated data files / entities

Database Users

1. Database administrators
2. Database designers
3. End-user
 - a) Casual user
 - b) Naive or parametric user
 - c) Sophisticated user
 - d) Stand-alone user
4. Systems analyst and application programmers

Determine requirements of end users.

Determine specification for canned transactions.

Determine standard types of queries and updates,

frequently used by native end user

Application programmers

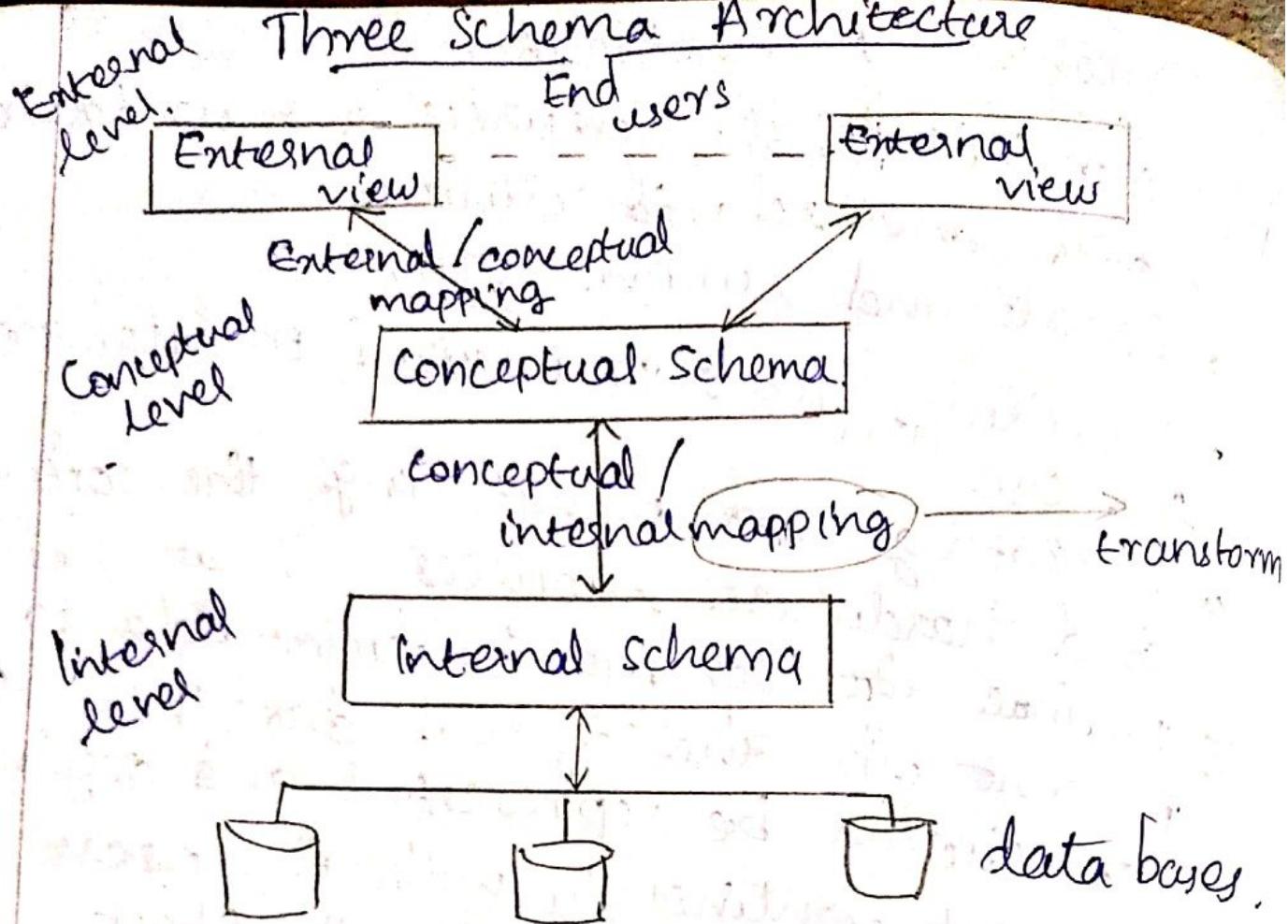
Implement specifications as program, test, debug, document and maintain the transaction.

Database Administrator

- 1) Administrate the resources such as database itself and related software.
- 2) Create and remove logins.
- 3) Providing access permissions to database for use.
- 4) Allocating and monitoring the software and hardware resources what need.
- 5) Define the schema (determine data to be present in the system and how this data should be represented and organised).
- 6) Interact continuously with the users.
- 7) Define security and integrity check (correctness and completeness of data).
- 8) Define back-up of recovery procedure.
- 9) Monitoring performance.

University Ques

1. List out any 3 salient features of database systems.
2. List any 3 categories of database users, highlighting any one important characteristic of each category.
3. List out any 3 responsibilities of database administrators.



Schema
student

R no	Name	Age	Addr
------	------	-----	------

BOOK

Bno	Iss Date	Author
-----	----------	--------

Three Schema Architecture is used to achieve

1. Self declaration
2. Insulation
- 3) Data abstraction etc

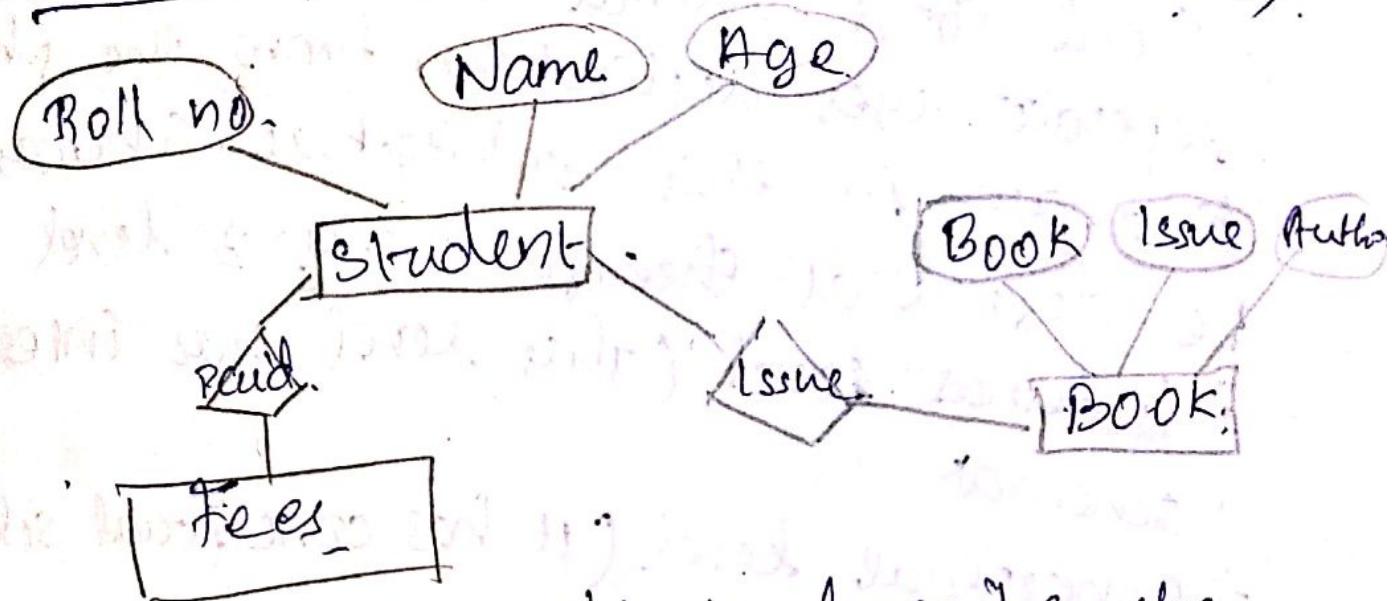
Data Independance

two type.

1 - Logical data Independence

2 . physical data Independence.

Data Model (ER model { Entity Relationship model }).



A set of concepts to describe the structure of a database and certain constraints that the database should obey.

Types of datamodel.

1. Relational model — like table.
2. Network model.
3. Hierarchical Data model — different levels like tree.
4. Object oriented data model.
5. Object relational data model.
6. Entity relationship model (ER model)

Three Schema Architecture

Three schema Architecture, it is an architecture for database system, to achieve the basic 3 characteristics of database approach. [1, 2, 3] {self descr., insulation }

Goal of 3 Schema Architecture is to separate user application from the physical database in this architecture, Schemas can be defined at the following 3 level

- 1) Internal level (This level has internal schema)
- 2) Conceptual level (It has conceptual schema)
- 3) External or View level (It has external schema or user view).

It is a tool through which user can visualize the schema levels in a database system. Three schema Architecture can be used, to further explain the concept of Data Independence, which is defined as the capacity to change the schema at one level, without changing schema at the next higher level.

DBMS Languages

- 1. Data Definition language (DDL)
 - 2. Storage definition language (SDL)
 - 3. View definition language (VDL)
 - 4. DML (Data Manipulation Language)
 - a) data sublanguage
 - b) host language
 - c) query language
- High level or Non Procedural language
- SQL, are set-oriented and specify what data to retrieve than how to retrieve. Also called declarative languages.
- Low level or Procedural languages:
- record-at-a-time, they specify how to retrieve data and include constructs such as looping.

DBMS Interfaces

- stand-alone query language interfaces.
- User-Friendly interfaces.
 - Menu based, popular for browsing on the web,
 - Forms-based, designed for naive users
 - graphics-based

NAVATHA
Fourth
edit.

- Natural language: requests in written English.
- Combinations of the above

Advantages of Using Database Approach

- controlling redundancy in data storage
- Data normalization ensure consistency of data and save storage space.
- Data normalization eliminate data redundancy
- sharing of data among multiple users.
- Restricting unauthorized access to data.
- Securing and authorization subsystem of DBMS create accounts and specify account restriction.
- Providing persistent storage for program objects.
- Providing storage structures for efficient Query processing (Indexing).
- Programming languages use data structures such as record type in Pascal or class definitions in C++ or JAVA.

Values of program variables or objects are discarded once a program terminates, unless programmer store them (variables or objects) in permanent files.

Auxiliary files called indexes are used for this purpose. Buffering or Caching module maintain part of database in main memory buffer. (fast retrieving). Query processing and optimization module choose an efficient query execution plan for each query.

- Provide backup and recovery services.
- Providing multiple interfaces to different classes of users.
- Representing complex relationships among data.
- Enforcing integrity constraints on the data base.
- Providing interfaces and actions using.

Types of Data Using Rule.

(406)

- Structured data

- Semistructured data

- Unstructured data

Structured data

Each record has same format as that of other records in the table. Ensures that all data follow the structures and constraints specified in schema.

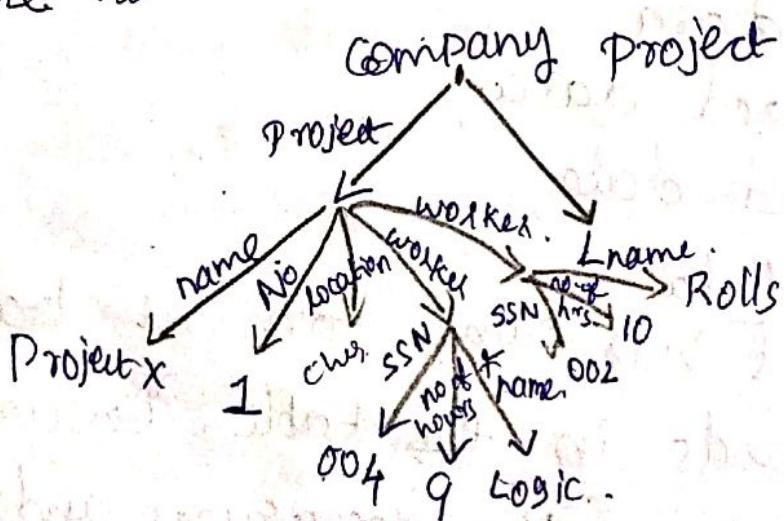
Unstructured data

Text documents with information embedded in web pages or in html, can be construed as unstructured data.

Semistructured data

Data may have a certain structure, but not all information collected have the identical structure. There is no predefined schema. Some attributes may be shared among various entities, but other attributes ~~only exist~~ ^{makes} only in certain entities.

Data models representing semi structured data are ~~is~~ ^{at} tree and graph. Each data object can have different attributes that are not known in advance.



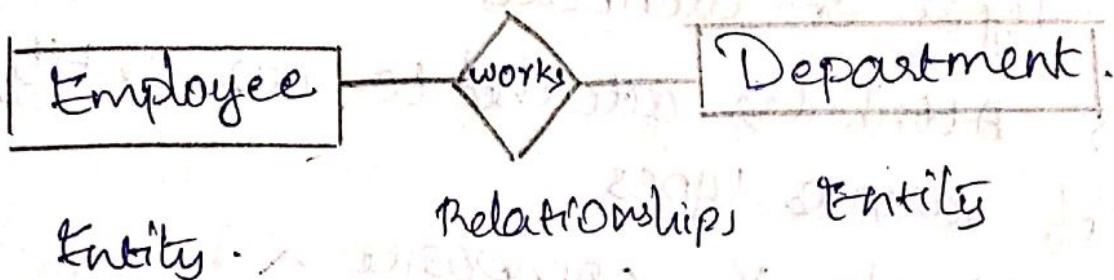
Component modules of the DBMS and

6/2/19

Database Modeling Using Entity Relationship Diagram (ER Diagram)

Eg:-

Company → Database



Employee is the entity type.

{e₁, e₂, e₃, e₄} → entity set.

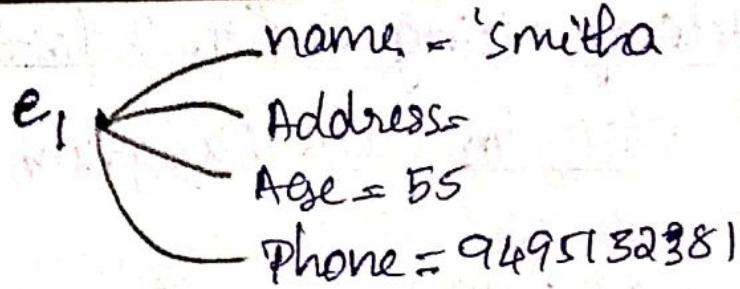
Set of values for particular attribute is called domain.

{001, 002, 003} → domain

SSN	Age, name
001	Person A
002	Person B
003	Person C

entity is a thing in the real world with an independent existence. An entity may be an object with physical existence or eg:- person, employee, student, car, home etc.

may be an object with a conceptual existence such as company, university, course etc. Attribute:- Properties that describe the entity can be described by employee name, address, salary etc. A particular entity will have a value for each of its attribute.



8/2/19

Types of attributes

Attributes attached to an entity can be of various types

→ simple Vs composite

1) Simple or atomic attribute

Attributes that are not divisible and represent the basic meaning is called simple or atomic attribute

e.g:- Age, sex etc

2. Composite attribute.

Attributes that can be further divided into smaller units and each individual unit contains a specific meaning

e.g:- Name, Name can be divided into Fname, Mname, Lname.

Address, it can be divided into house no, street etc.

→ Single Valued attribute Vs Multivalued attribute

1) Single Valued

Attributes that have a single value for a particular entity

e.g.: - Age,

2) Multivalued

Attributes that have more than one value for a particular entity is called multi-valued attribute.

e.g.: - Ph. number.

→ stored Vs Derived Attribute

In some cases 2 or more attribute values are related.

e.g.: - Age & date of birth of a person.

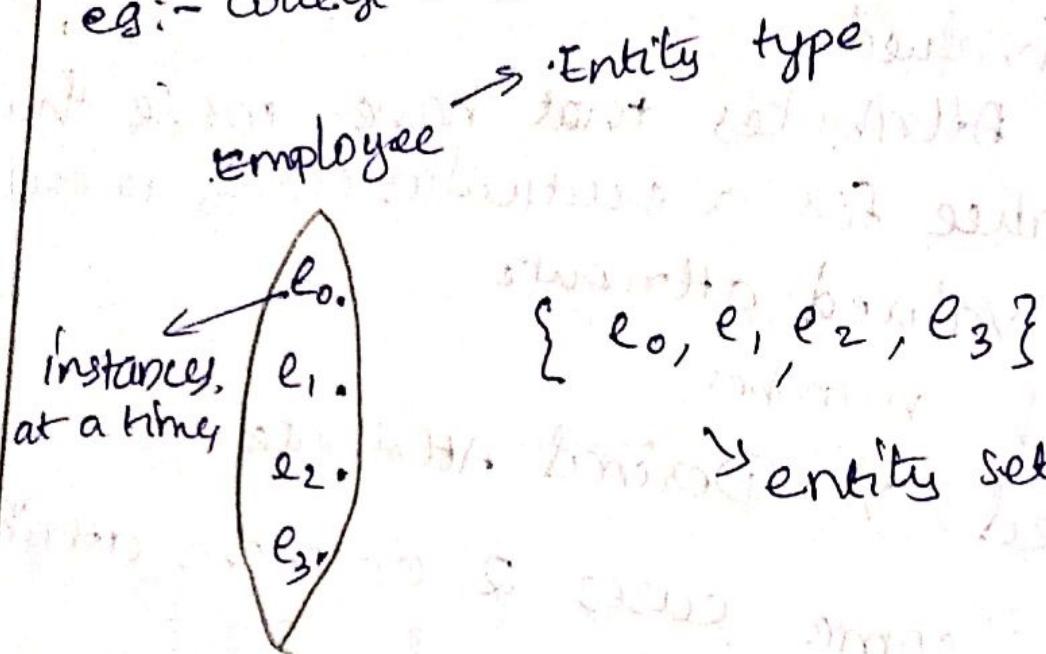
e.g.: - Age & date of birth of a person.
i.e., Value of age can be derived from dob. Hence Age attribute is called derived attribute. And dob is known as stored attribute.

→ complex attribute

Grouping of components of a composite attribute and multivalued attribute is called complex attribute.

e.g.: - { Previous degree (Name of college, yr, stream),
multi-valued }

NULL value: In some cases, a particular entity may not have an applicable value for an attribute.
e.g.: college degree.



SSN.	Name	Age	Prior degree
1	-	-	B Tech
2	Student	-	M Tech
3	-	-	NULL → no previous degree
4	-	-	BCA.

Domain

Domain of values (Value set) specifies the set of values that may be assigned to that attribute for each individual entity.

Relationship

A relationship relates two or more distinct entities with a specific meaning - i.e., relationship is an association among two or more entities.

Relationship type

→ Relationship set
Degree of relationship is the number of participating entities types.

Binary relationship means, no. of participating entities are two.
If it is of degree 3, then no. of participating entities are 3, and it is termed a Ternary relationship.

Role names

Relationship [Relationship]

Constraints on relationship type

Two types of relationships constraints

1) Cardinality ratio.

Cardinality ratio of a binary relationship specifies the number of relationship instances that an entity can participate. CR can be 1:1, 1:N, M:N, N:1

2) Participation constraint.

This concept specifies whether the existence of an entity depends on another entity via the relationship type.

Two types of participation constraint.

1) Total participation.

2) Partial participation.

Relationship types can also have attributes

Weak entity type.

Entity types that do not have a key attribute ~~as~~ ^{own} of their own are called weak entity type.

Entity type that have key attribute is called strong entity type.

Strong entity type

Identifying relationships } Identifying

Partial key

owner entity

type.

Company has the following scenario
there are a set of sales person, some of them manages other sales person, however a sales person cannot have more than 1 manager. A sales person can be an

12/2/19

Q

agent for many customers. A customer is managed by exactly one sales person. Customer can place any number of orders. An order can be placed by exactly one customer. Each order lists one or more items. An item may be listed in many orders. An item is assembled from different parts. And parts can be common for many ~~of~~ items. One or more employees assemble an item from parts. A supplier can supply different parts in certain quantities. A part may be supplied by different suppliers.

- 1) Identify and list entities, suitable attributes, primary keys, foreign keys & relationships to represent the scenario.
- 2) Draw an ER diagram to model the scenario using min-max notation.