Batch:E2 Roll No.: 16010123325
Experiment / assignment / tutorial No. 5
Grade: AA / AB / BB / BC / CC / CD /DD
Signature of the Staff In-charge with date

Title: Database application description and Design of Entity-Relationship diagram

---

Objective: To comprehend the data requirements of the application and design the Entity-Relationship (ER) diagram for the database

---

Expected Outcome of Experiment:

CO1: Comprehend the Characteristics of Relational Database Management Systems.
CO2: Create Relational Database Designs Based on Entity-Relationship Models.

Books/ Journals/ Websites referred:

1. G. K. Gupta :"Database Management Systems", McGraw – Hill
2. Korth, Slberchatz, Sudarshan : "Database Systems Concept", 6th Edition , McGraw Hill
3. Elmasri and Navathe, "Fundamentals of Database Systems", 5thEdition, PEARSON Education.

Pre Lab/ Prior Concepts:

ER Model:

The ER data model was developed to facilitate the database design by allowing specification of an enterprise schema that represents the overall logical structure of the database. The ER model is one of the several data models. The semantic aspect of the

model lies in its representation of the meaning of the data. The ER model is very useful many database design tools drawn on concepts from the ER model. The ER model employs 3 basic notations: entity set, relationship set and attributes**.**

Example Case Study: List the data requirements for the database of the company Case Study : A company which keeps track of employee, department and projects. The database designers provide the following description

1. The company is organized into departments. Each department has unique name, unique number, and particular employee to manage the department. We keep track of the start date and the employee begins managing the department. The department has several locations.
2. The department controls a number of projects each of which has a unique name, unique number and a single location.
3. We store each employee names social security number, address, salary, sex and dob. An employee is assigned one department but may work on several projects which are not necessarily controlled by the same department. We keep track of the department of each employee works on each project and for insurance purpose. We keep each dependents first name, sex, dob and relation.

Procedure for doing the ER diagram experiment

1. Identifying the Entities-Strong and weak entities (Nouns from the problem definition)

   Based on the requirements, we can identify four initial entity types in the COMPANY database:

   - DEPARTMENT
   - PROJECT
   - EMPLOYEE
   - DEPENDENT

2. Identify attributes of the Entity- keys, partial key, simple, composite,

multivalued, derived (characteristics of the entities)

The initial attributes shown are derived from the requirements description

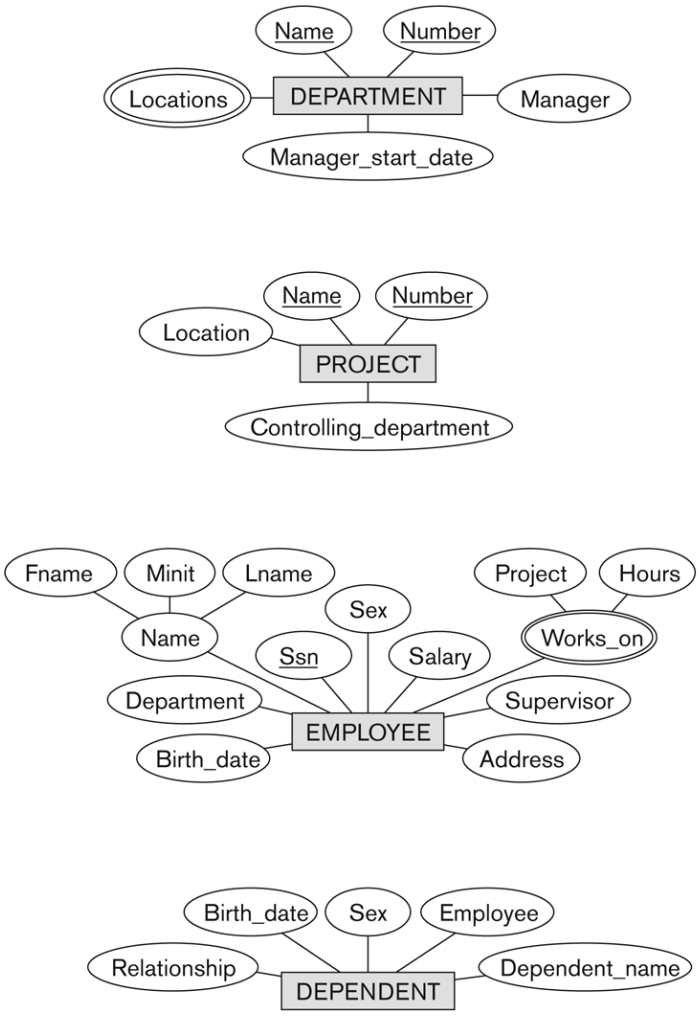


Figure 3.8
Preliminary design of entity types for the COMPANY database. Some of the shown attributes will be refined into relationships.

3. Identify relationships (verbs)

A **relationship** relates two or more distinct entities with a specific meaning.

- For example, EMPLOYEE John Smith *works on* the ProductX PROJECT, or EMPLOYEE Franklin Wong *manages* the Research DEPARTMENT.

Relationships of the same type are grouped or typed into a **relationship type**.

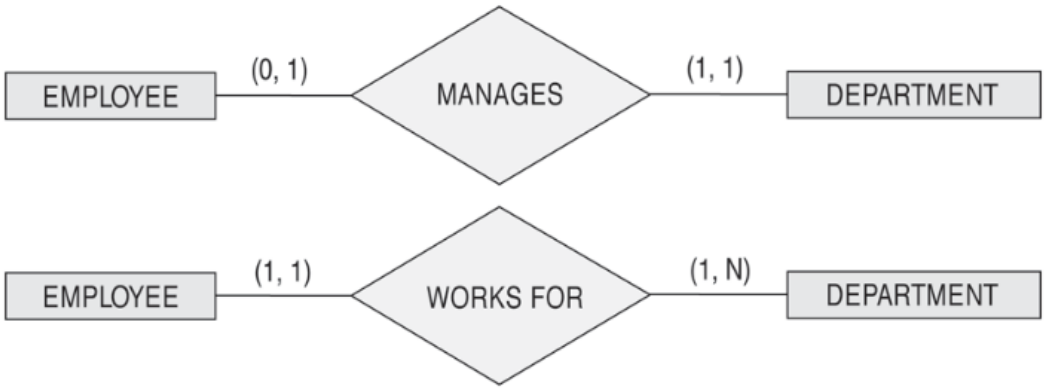
- For example, the WORKS_ON relationship type in which EMPLOYEEs and PROJECTs participate, or the MANAGES relationship type in which EMPLOYEEs and DEPARTMENTs participate.

The degree of a relationship type is the number of participating entity types.

- Both MANAGES and WORKS_ON are *binary* relationships.

In ER diagrams, we represent the *relationship type* as follows:

- Diamond-shaped box is used to display a relationship type
- Connected to the participating entity types via straight lines



Read the min,max numbers next to the entity type and looking **away from** the entity type

Listed below with their participating entity types:

- WORKS_FOR (between EMPLOYEE, DEPARTMENT)
- MANAGES (also between EMPLOYEE, DEPARTMENT)
- CONTROLS (between DEPARTMENT, PROJECT)
- WORKS_ON (between EMPLOYEE, PROJECT)
- SUPERVISION (between EMPLOYEE (as subordinate), EMPLOYEE (as supervisor))
- DEPENDENTS_OF (between EMPLOYEE, DEPENDENT)

ER- Diagram for company Case Study Database:

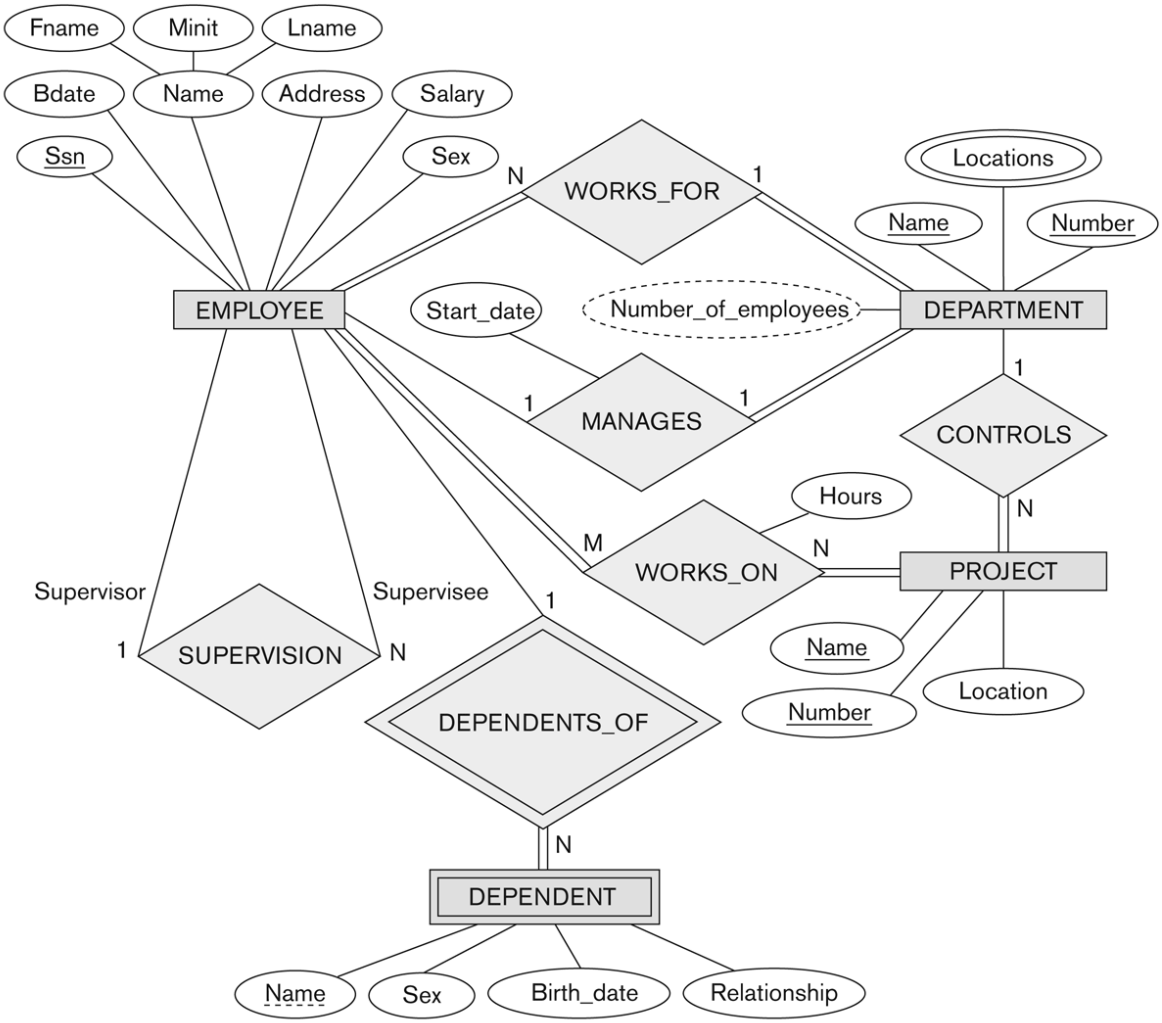
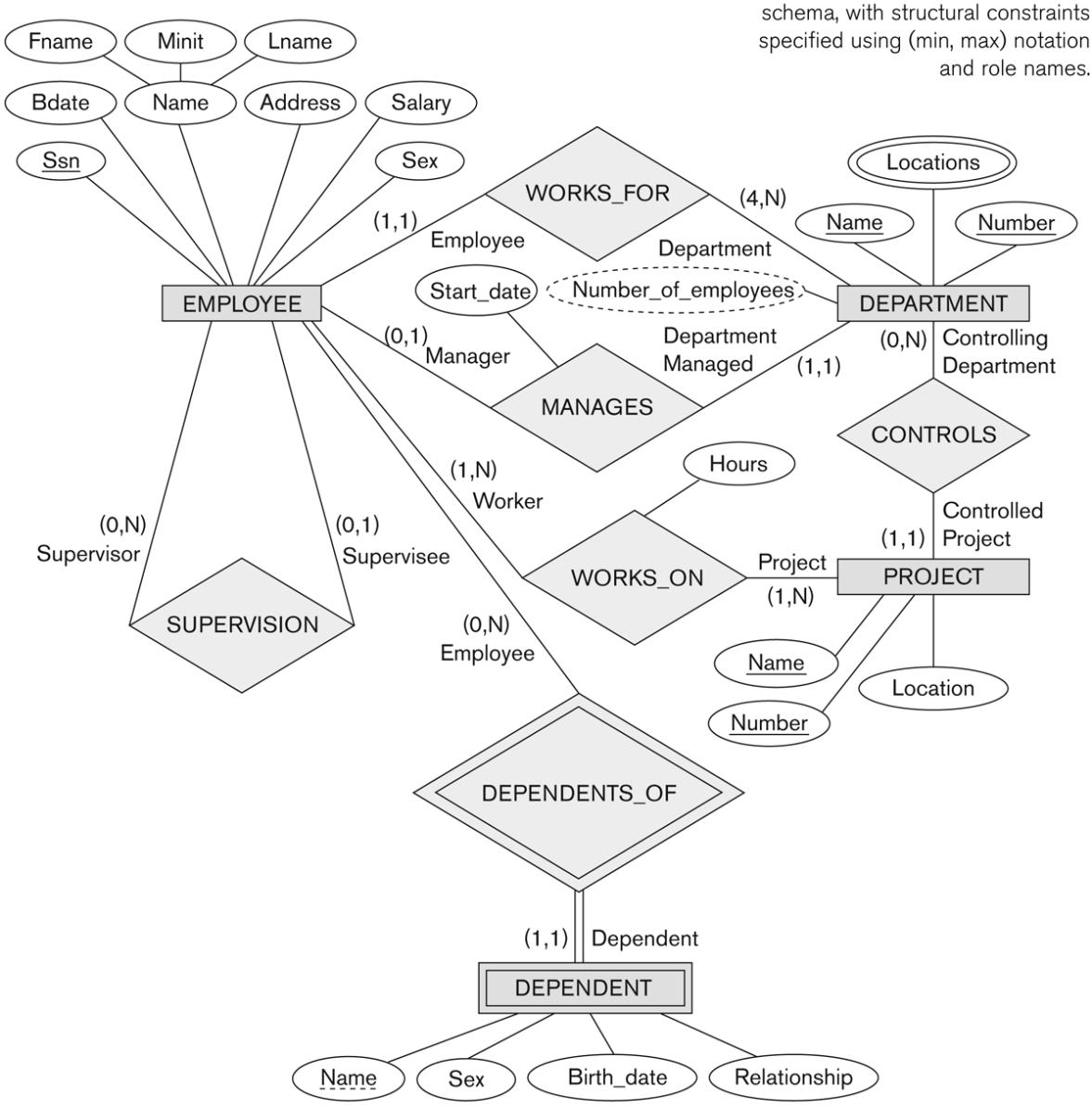


Figure 3.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

Identify the structural constraints of the relationship (cardinality ratio, participation constraints**)**

Figure 3.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.



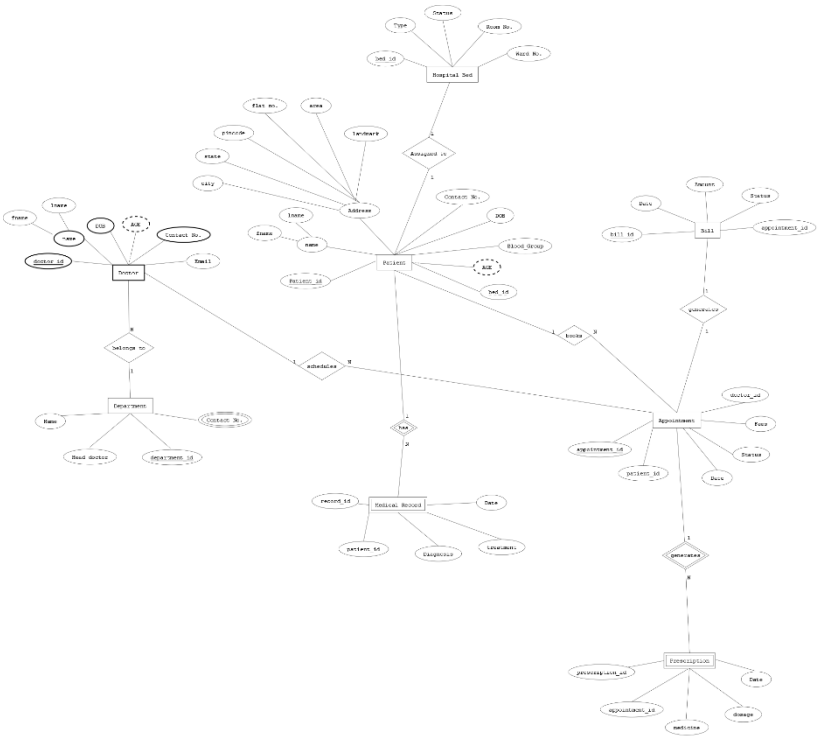
:

Problem Definition of student case study :

A hospital system that tracks doctors, patients, departments, hospital beds, medical records, prescriptions, appointments, and bills.
The database designers provide the following description:

1. The hospital is organized into departments. Each department has a unique name, unique number, and a senior doctor who manages the department. We track the start date when the doctor begins managing the department. Each department has multiple locations within the hospital.
2. Doctors belong to specific departments and have details like name, unique ID, specialization, contact information, and joining date.
3. Patients are assigned unique IDs and have attributes like name, address, contact information, blood group, date of birth, and assigned hospital bed. Patients book appointments with doctors, and we track the date, time, and status of these appointments.
4. Hospital beds are identified by unique numbers, bed types (e.g., ICU, general), and their current status. A bed can be assigned to a patient during their stay.
5. We maintain medical records for each patient, tracking details such as diagnosis, treatments, and the date of visit. Prescriptions are generated during these visits, containing details like medicines, dosages, and instructions.
6. Bills are generated for patient appointments and treatments, containing details like the amount, payment status, and the date of billing.

Entity Relationship Model w.r.t Problem definition



16010123325 - Shreyans Tatiya

1 ENTITIES

The Entities in the above given diagram are:-

- Doctor
- Department
- Patient
- Hospital Bed
- Medical Record
- Appointment
- Prescription
- Bill

2. ATTRIBUTES

Doctor:

- doctor_id
- DOB
- contact_no
- email
- name (composite):
  - fname
  - lname

Department:

- department_id
- name
- contact_no
- head_doctor

Patient:

- patient_id
- blood_group
- DOB
- age
- bed_id
- Address (composite attribute):
  - flat_no
  - area
  - pincode
  - state
  - city
  - landmark
- name (composite):
  - fname
  - lname

Hospital Bed:

- bed_id
- type
- status
- room_no
- ward_no

Medical Record:

- record_id

- patient_id
- date
- diagnosis
- treatment

Appointment:

- appointment_id
- doctor_id
- patient_id
- date
- status
- fees

Prescription:

- prescription_id
- appointment_id
- date
- dosage
- medicine

Bill:

- bill_id
- appointment_id
- amount
- status
- date

3. RELATIONSHIPS WITH PARTICIPATING ENTITIES

- Doctor belongs_to Department (1:1)
- Doctor schedules Appointment (1:N)
- Patient books Appointment (1:N)
- Appointment generates Bill (1:1)
- Appointment generates Prescription (1:1)
- Patient has Medical Record (1:N)
- Patient assigned_to Hospital Bed (1:1)

4. RELATIONSHIPS WITH CONSTRAINTS AND ROLES

1. Doctor - Department Relationship ("belongs_to")

- Constraint: (1,1) on Doctor's side, (1,N) on Department's side
- Role: Each doctor must belong to exactly one department
- Participation: Total participation from Doctor (mandatory)
- Meaning: A doctor cannot exist without being assigned to a department

2. Doctor - Appointment Relationship ("schedules")

- Constraint: (0,N) on Appointment side, (1,1) on Doctor side
- Role: Doctor acts as scheduler, Appointment as scheduled entity
- Participation: Partial participation from Doctor (optional)
- Meaning: A doctor can schedule multiple appointments, each appointment must have exactly one doctor

3. Patient - Appointment Relationship ("books")

- Constraint: (0,N) on Appointment side, (1,1) on Patient side
- Role: Patient acts as booker, Appointment as booked entity
- Participation: Partial participation from Patient (optional)
- Meaning: A patient can book multiple appointments, each appointment belongs to one patient

4. Patient - Hospital Bed Relationship ("assigned_to")

- Constraint: (0,1) for Patient, (0,1) for Hospital Bed
- Role: Patient as assignee, Hospital Bed as assigned resource
- Participation: Partial participation from both entities (optional)
- Meaning: A patient may or may not be assigned to a bed, a bed may or may not have a patient

5. Patient - Medical Record Relationship ("has")

- Constraint: (1,N) for Medical Record, (1,1) for Patient
- Role: Patient as record owner, Medical Record as owned entity
- Participation: Total participation from both entities (mandatory)
- Meaning: Each patient must have at least one medical record, each medical record belongs to exactly one patient

6. Appointment - Bill Relationship ("generates")

- Constraint: (1,1) for both entities
- Role: Appointment as generator, Bill as generated entity
- Participation: Total participation from both entities (mandatory)
- Meaning: Each appointment must generate exactly one bill, each bill must be associated with exactly one appointment

7. Appointment - Prescription Relationship ("generates")

- Constraint: (0,1) for Prescription, (1,1) for Appointment
- Role: Appointment as generator, Prescription as generated entity
- Participation: Partial participation from Appointment (optional)
- Meaning: An appointment may or may not generate a prescription, but each prescription must be associated with exactly one appointment

Conclusion: This ER diagram represents a comprehensive hospital management system that effectively captures the relationships between healthcare providers, patients, and administrative entities. The design supports key hospital operations including appointment scheduling, patient records management, billing, and bed management.

Post lab Descriptive Questions

1. Discuss the concept of aggregation. W.r.t your case study give example for aggregation

Solution:-

In this case study, we can see aggregation in the relationship between Appointment and its related entities (Bill and Prescription). The Appointment entity acts as a whole, while Bill and Prescription are treated as parts. This is because:

1. Bills and Prescriptions cannot exist without an Appointment
2. The Appointment is treated as a higher-level entity that encapsulates these related transactions
3. The appointment_id is propagated to both Bill and Prescription entities, showing their dependency on the Appointment entity

This aggregation helps in:

- Maintaining data consistency
- Establishing clear hierarchical relationships
- Simplifying the tracking of related healthcare transactions
- Ensuring proper record-keeping of patient visits and associated documents