Aum Sri Sai Ram

Office Management System

Problem Statement:

- The office management system manages various aspects of office functions, including employee records, meeting schedules, task assignments, managing the documents, reports and resource allocation. The goal is to enhance overall productivity, communication, and coordination within the office environment.
- Each aspect has many fields. Employee management stores the employee details such as name, designation, contact information and employee history. A unique employee ID is given for each member.
- Meeting management allows the employees to schedule meetings with date and time and location. It also takes care of the attendance and their responses.
- The office also provides many resources to their employees. The resources and their availability are kept tracked. It also enables employees to request for the resources with specific time and date. The resources attributes consist of resources ID, name of the resources, employee ID asked for the resources, dates of the resources provided.
- **Manager** provides the resources.
- The office also gives tasks to each employee. So, there is a task assignment and tracking. It allows supervisors to assign tasks to specific employees. And also includes a deadline for it. The attributes for the tasks are task provided by, provided to, date of assigning, date of deadline.
- Office also allows employees to upload, share, and organise documents related to projects and tasks. There is a service provided for employees to manage these documents also.
- Office allows employees to give reports of their project such as tasks completed and tasks pending. And the supervisor can see the employee's performance through these activities.

Entities for My Problem:

- **Person** -Employee
- **Event** -Task, Meeting, Meeting_Attendees
- **Object** - Resource, Resource_Request
- **Concept** - Document, Report

Attributes for my entities:

Entity_Type	Attributes	Type
Employee (Strong)	Emp_ID(Primary Key) Emp_Name Designation Contact_Info Emp_History Salary Address Manager_ID	Simple Composite Simple Multivalued Simple Simple Simple Simple
Task (Strong)	Task_id(Primary Key) Task_Name Description Deadline Priority Status Emp_ID	Simple Single Required Simple Simple Single Simple
Meeting (Strong)	Meeting_ID(Primary Key) Meeting_Date Meeting_Time Location	Simple Simple Simple Composite
Meeting_Attendees	Meeting_ID Emp_ID Response	Simple Simple Required
Resource (Strong)	Resource_ID(Primary Key) Resource_Name Availability_Status	Simple Simple Required
Resource_Request (Strong)	Request_ID(Primary Key) Resource_ID Emp_ID Request_Date Request_Time	Simple Simple Simple Simple Simple
Document (Strong)	Document_ID(Primary Key) Document_Name Document_Path Upload_Date Emp_ID	Simple Composite Optional Simple Simple
Report (Strong)	Report_ID(Primary Key) Report_Name Description Report_Date Emp_ID	Simple Required Required Required Required

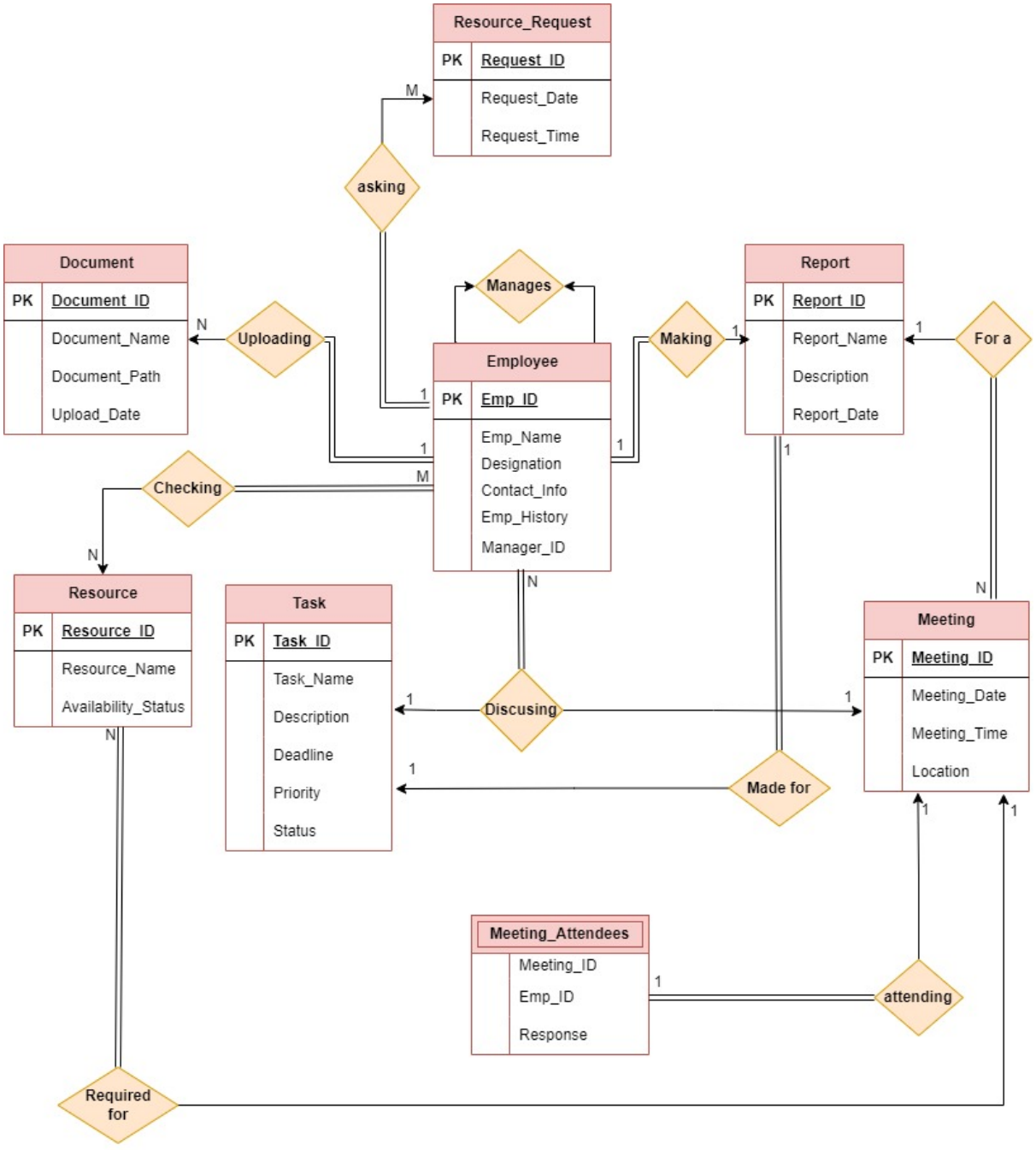
DEGREE OF RELATIONSHIPS:

- **Unary Relation:**
  - Employee
- **Binary Relation:**
  - Employee to Document
  - Employee to Resource_Request
  - Employee to Report
  - Employee to Resource
  - Meeting to Report
  - Meeting_Attendees to Meeting
  - Resource to Meeting
- **Ternary Relation:**
  - Employee,Task and Meeting

THE RELATIONSHIP SETS IN MY DESIGN ARE LISTED BELOW:

- **manages**: relating to employees.
- **discussing**: ternary relationship for employee,task and meeting.
- **asking:** employee asking for a resource_req.
- **uploading:** employee uploading document.
- **making:** employee making report.
- **checking:** employee checking available resources.
- **required for:** resource required for meeting.
- **for an:** report for meeting.
- **made for:** report made for task.
- **attending:** meeting attendees attending the meeting and response given for the meeting.

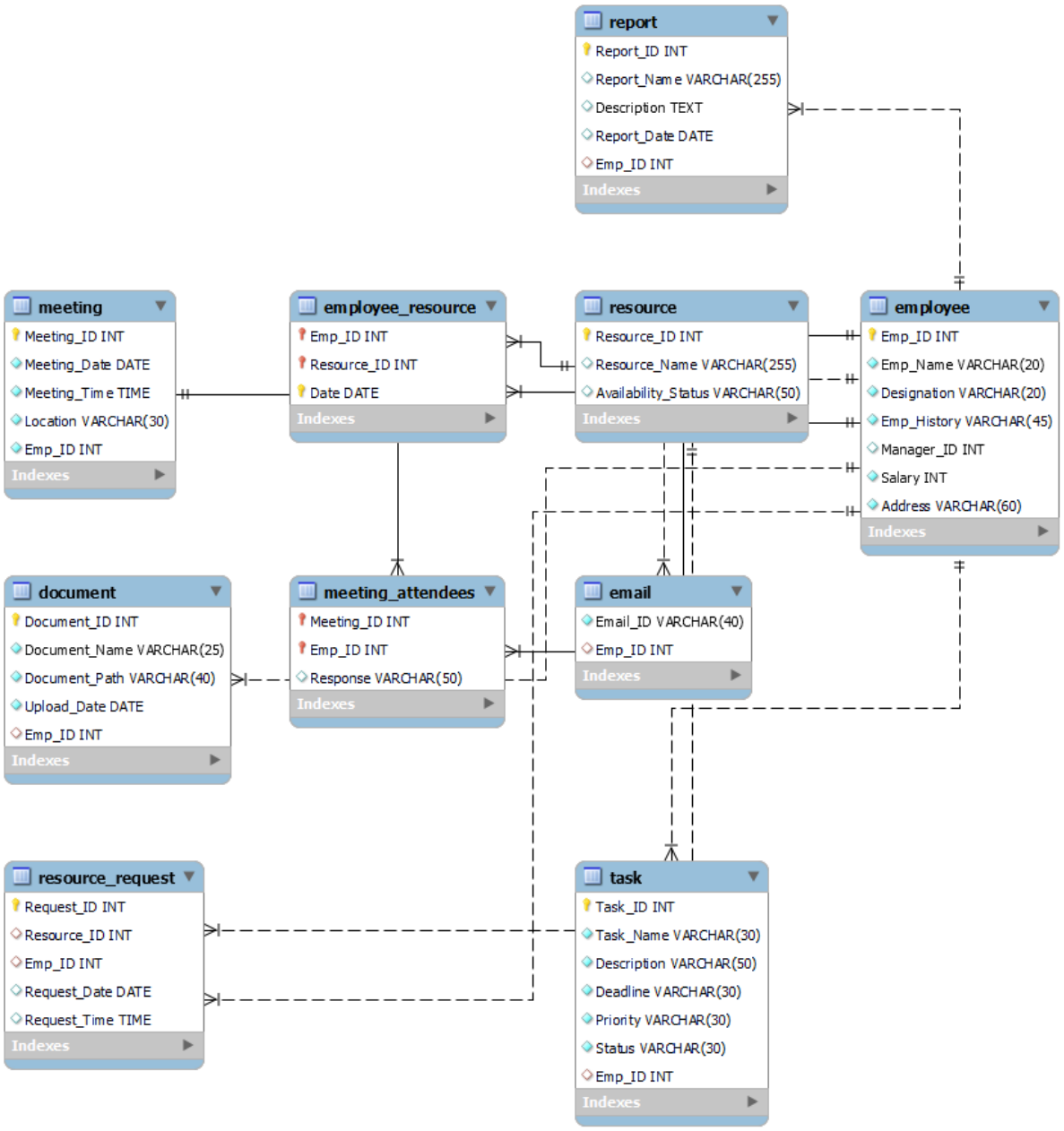
ER diagram for Office Management System



RELATIONS/TABLES:

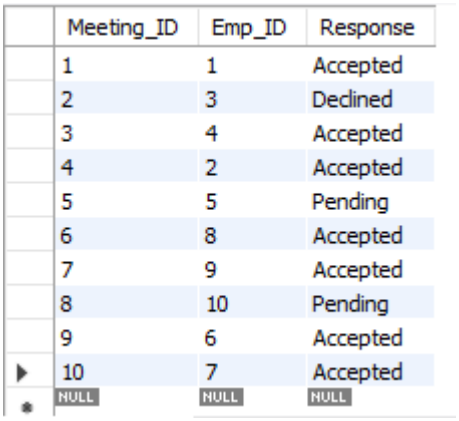
- **Employee(**Emp_ID,Emp_Name,Designation,Emp_History,Manager_ID,Salary,Address**)**
- **Task(**Task_id,Task_Name,Description,Deadline,Priority,Status,Emp_ID**)**
- **Meeting(**Meeting_ID,Meeting_Date,Meeting_Time,Location,Emp_ID**)**
- **Meeting_Attendees(**Meeting_ID,Emp_ID,Response**)**
- **Resource(**Resource_ID,Resource_Name,Availability_Status**)**
- **Resource_Request(**Request_ID,Resource_ID,Emp_ID,Request_Date,Request_Time**)**
- **Document(**Document_ID,Document_Name,Document_Path,Upload_Date,Emp_ID**)**
- **Report(**Report_ID,Report_Name,Description,Report_Date,Emp_ID**)**
- **Employee_Resource(**Emp_ID,Resource_ID,Date**)**
- **Email(**Email_ID,Emp_ID**)**

ER-diagram provided by SQL Workbench



Functional Dependency And Normalisations:

The table selected to find normalisation is the **meeting_attendees** relation.



Meeting_ID	Emp_ID	Response
1	1	Accepted
2	3	Declined
3	4	Accepted
4	2	Accepted
5	5	Pending
6	8	Accepted
7	9	Accepted
8	10	Pending
9	6	Accepted
10	7	Accepted
NULL	NULL	NULL

Checking for 1NF:

- R={Meeting_ID,Emp_ID,Response}
- This relation is already in 1-NF form because there is no multivalued attribute(i.e., they are atomic in nature).

Checking for 2NF:

- Meeting_ID=A,Emp_ID=B,Response=C
- The FDs are F={AB->C,B->C}
- Identify the Prime and Non-Prime Attributes
  - prime={A,B}
  - non-prime={C}
- Check if the Candidate key is a Composite Key.
  - Yes,AB is a Composite key
- Check if the Non-Prime attributes are fully dependent on the Prime Attributes.
  - True
- Check if the Non-Prime attribute is Partially dependent on the Prime Attributes.

  $B \subseteq C$

  B->C

  ∴B->C is partially dependent

∴ The relation R is not in 2NF. Because B->C is partially dependent.

Converting to 2NF:

- **To convert to 2NF create a new relation** containing the determinant of a partial dependency which is the primary key in the new relation.
- Move the non-key attributes that are dependent on this primary key attribute(or attributes) from the old relation to the new relation.
- So new relation Meeting_Response ={B,C}
- Old Relation Meeting_Attendees={A,B}

OLD Relation i.e, Meeting_Attendees:

Meeting_ID	Emp_ID
1	1
4	2
2	3
3	4
5	5
9	6
10	7
6	8
7	9
8	10
NULL	NULL

NEW Relation i.e, Meeting_Response :

Emp_ID	Response
1	Accepted
2	Accepted
3	Declined
4	Accepted
5	Pending
6	Accepted
7	Accepted
8	Accepted
9	Accepted
10	Pending

<u>Checking for 3NF:</u>

- The above table is already in 2NF.
- Check if there is any transitive dependency
    - There is no transitive dependency for the above.

∴ The table is in 3NF. So no need to convert it to 3NF.