College of IT && Computer Engineering

Database Management System

Employee Training project

DR:Mohamed AL-Saheb

Student:
sajeda swad (227087) ,
nourhan osama(201161)

Database Design for Employee Training Management

Aims:

1. Conceptual model (ER-Model using UML)
2. Logical Model (using mapping)
3. Check the Logical model using Normalization
4. Give the data dictionary for each table and the needed SQL statement to create the table using LIveSQL.

—---

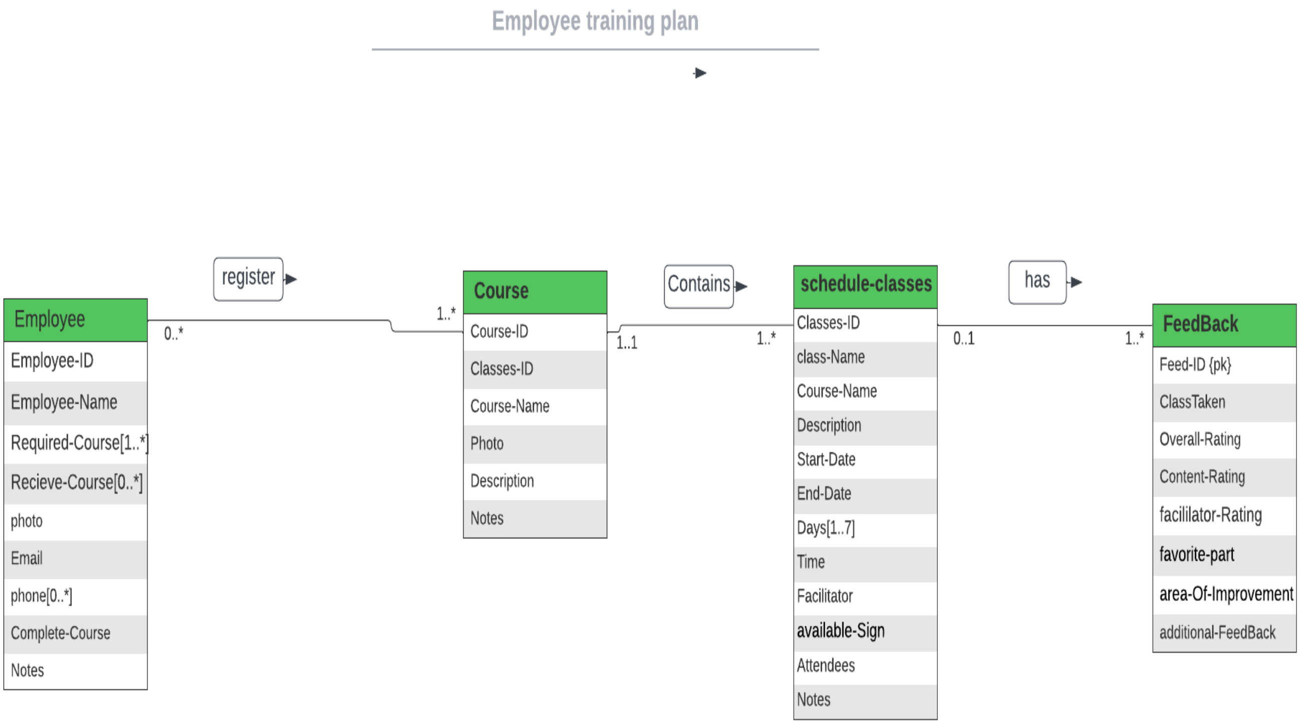
ER MODEL:
first step determine Entity && attribute
Entity?
1-Employee
2- course
3-classes
4-Feedback

Attribute for each entity is :
EMP
{FName,LName},Pic,ReqCourses,RecCourses,SignUpCourses,phone,Email,complete,Note

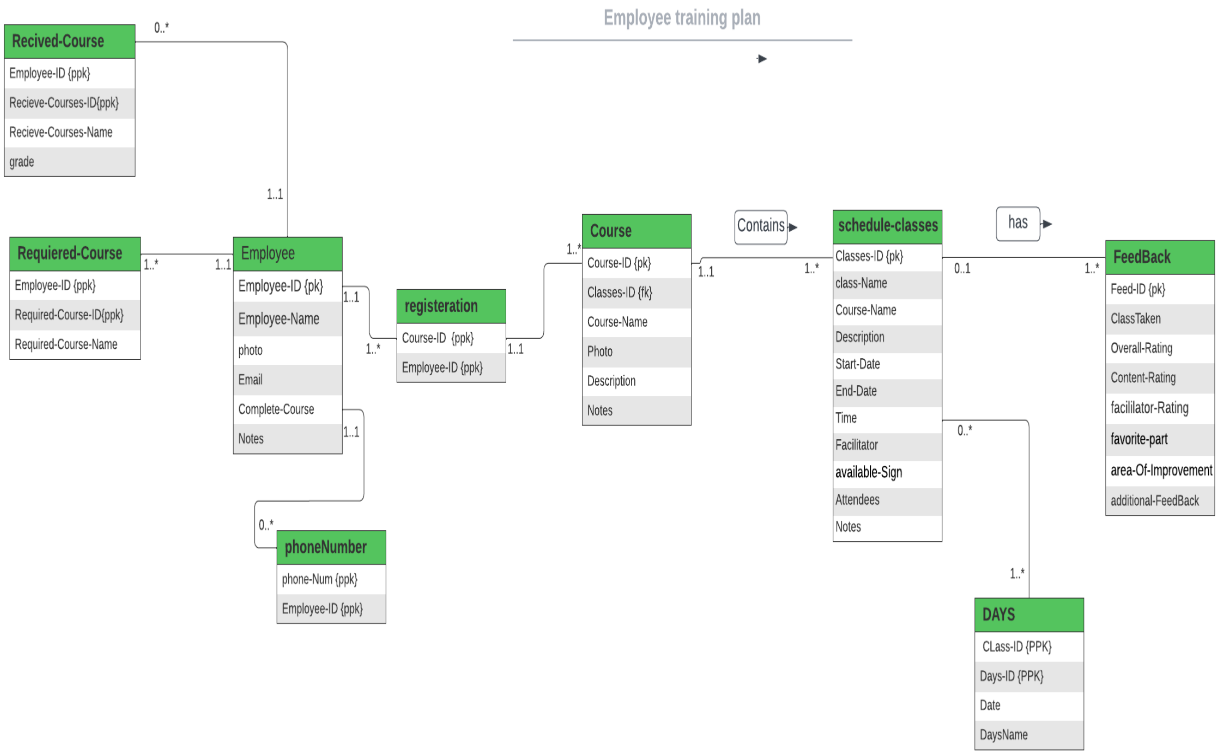
Course
CourseName ,photo ,Dis,ClassName ,notes

classes
ClassName , courseName ,Dis , date(S-date,L-date),days,time,facilitator ,available(boolean) ,link,attendees

CLASS FEEDBACK
-id ,classeTaken ,rating(overall-rating ,content-rating , facilitator-rating),favorite part,area-of-improvement ,add-FeedBack



THATS LINK FOR OUR WORK ON BLANK DIGRAM (if the picture is not clear)
https://lucid.app/lucidchart/5a9d0ede-3530-4eb0-b5a9-f596732ea383/edit?view_items=CK0WTb3r~w2Q%2CCK0Wnb94yI19%2CCK0WQi-9P_kd%2CJM0WimZHi~4o%2CCK0Wi5K5uOA_%2CCK0Wb9J3AIkk%2CCK0W5UnVLmH9%2CCK0W8eq~Inxr%2CCK0W1L6.2Myv%2CCK0WAUkD5_NC%2CCK0WqMRUM9FP%2CJM0WcpMCfJd2%2CCK0WYpYLIKH6%2CCK0WmftPJ6vQ%2CpM0W0f6O_Oha&invitationId=inv_5c664748-fd36-45bb-a9a4-de6fce9a3c32



THATS LINK FOR OUR WORK ON BLANK DIGRAM (if the picture is not clear) https://lucid.app/lucidchart/65dca2fd-91ee-4e66-a723-0f2eefe22ecf/edit?viewport_loc=-141%2C-16%2C1480%2C692%2Cv~oVnpmMbZy9&invitationId=inv_057ac079-7125-4684-8de4-3729d8370738

For Normalization
Note: we do a checking in mapping 😊

Employee(
Employee-ID,Name,Picture,SignUp-Courses,phone,Email,complete-Courses,Note)
Registration(Employee-ID,Course-ID)
Course(Course-ID,Class-ID,Course-Name ,photo ,Description ,notes)
Received-Course (Employee-ID,Received-Courses-Name,Recieve-Courses-ID)
Required-Course(Employee-ID, Required-Course-Name ,Required-Course-ID)
Phone (Employee-ID,Phone-Num)
schedule-Classes
 (Class-ID, Class-Name,course-Name,Description,Start-Date,Last-Date,days,time,facilitator ,available-Sign ,link,attendees)

Class-Feedback(Feed-ID ,classe-Taken ,overall-Rating ,content-Rating , facilitator-Rating ,favorite-part,area-Of-Improvement ,additional-FeedBack)
Days (DaysName,Class-ID, Date)

1st NF: must no repeated group or compound
Employee
Registration ✔
Course✔
Received-Course✔
Required-Course✔
Phone ✔
Classes✔
Class-Feedback✔
Days✔

2nd NF: must no have partial dependency
Employee
Registration ✔
Course✔
Received-Course✔
Required-Course✔
Phone ✔
Classes✔
Class-Feedback✔
Days✔

3rd NF : must have no transitive dependency

Employee
Registration ✔
Course✔
Received-Course✔
Required-Course✔
Phone ✔
Classes✔
Class-Feedback✔
Days✔

FOR Data dictionary

1-Employee Dictionary

NAME	TYPE	LENGTH	PK	DESCRIPTION
Employee-ID	NUMERIC		✅	Unique identifier for each employee
Employee-Name	VARCHAR	255		Full name of the employee
Photo	BLOB			Binary large object storing an image of the employee
Email	VARCHAR	255		Email address of the employee
Complete-Course	BOOLEAN			Flag indicating whether the employee has completed a course
Notes	TEXT			Additional notes or comments about the employee or their record

2- Received-Course

Name	TYPE	LENGTH	PPK	DESCRIPTION
Employee-ID	NUMERIC		✅	Unique identifier for each employee
Receive-Courses-ID	NUMERIC		✅	Unique identifier for the received course

Receive-Courses-Name	VARCHAR	255		Name of the received course
Grade	VARCHAR	255		Grade or evaluation received for the completed course

3- Required-Course

Name	**TYPE**	LENGTH	PPK	DESCRIPTION
Required-Course-ID	NUMERIC		✅	Unique identifier for the required course
Employee-ID	NUMERIC		✅	Unique key referencing Employee table
Required-Course-Name	VARCHAR	255	✅	Name of the required course

4-PhoneNumber

NAME	TYPE	LENGTH	PPK	DESCRIPTION
phone-Num	VARCHAR	0-9	✅	Phone number of the employee(Unique Key)
Employee-ID	INT		✅	Unique key referencing Employee table

5-Registration

NAME	TYPE	LENGTH	PPK	DESCRIPTION
Course-ID	INTr		✅	Unique identifier for the required course
Employee-ID	INT		✅	Unique key referencing Employee table

6-Course

NAME	TYPE	LENGTH	PK	FK	DESCRIPTION
Course-ID	INT		✅		Unique identifier for the course
Classes-ID	INT			✅	Foreign key referencing Class table
Course-Name	VARCHAR	255			Name of the course
Photo	BLOB				Binary data of the course photo
Description	VARCHAR	255			Description or details of the course
Notes	VARCHAR	255			Additional notes or comments about the course

7- Schedule-Classes

NAME	TYPE	PK	FK	DESCRIPTION
Classes-ID	NUMBER	✅		Unique identifier for the class
class-Name	VARCHAR2			Name of the class
Course-Name	VARCHAR2			Name of the course associated with the class
Description	VARCHAR2			Description or details of the class
Start-Date	DATE			Start date of the class
End-Date	DATE			Last date or end date of the class

Days	VARCHAR2			Days of the week when the class occurs
Time	INTERVAL DAY TO SECOND			Time of day when the class takes place
Facilitator	VARCHAR2			Facilitator or instructor of the class
available-Sign	VARCHAR2			Availability status or sign-up information for the class
Attendees	NUMBER			Number of attendees or participants registered for class
Notes	VARCHAR			

8-Feedback

Name	Type	LENGTH	PK	DESCRIPTION
Feed-ID	INT		✅	Unique identifier for the feedback
Class Taken	VARCHAR	255		Referencing Class table
Overall-Rating	INT			Rating for the overall experience
Content-Rating	INT			Rating for the content of the class
facilitator-Rating	INT			Rating for the facilitator's performance
favorite-part	VARCHAR	255		Favorite part or aspect of the class
area-Of-Improvement	VARCHAR	255		Area of improvement suggested for the class
additional-FeedBack	VARCHAR	255		Additional feedback or comments on the class

9-Days

Name	Type	LENGTH	PPK	Description
CLass-ID	INT		✅	Unique key referencing Class
Days-ID	INT		✅	Unique key for the table
Date	DATE			Date of the class
Days Name	VARCHAR	255		Name of the day

SQL Statements:

1.Create Employee table

```
CREATE TABLE Employee (
  EmployeeID INT PRIMARY KEY,
  FirstName VARCHAR(255),
  LastName VARCHAR(255),
  Email VARCHAR(255),
  PhoneNum VARCHAR(20),
  HireDate DATE,
  Department VARCHAR(255),
  Salary DECIMAL(10, 2)
);
```

2.Create Registration table

```
CREATE TABLE Registration (
  EMPLOYEE_ID INT,
  COURSE_ID INT,
  PRIMARY KEY (EMPLOYEE_ID, COURSE_ID),
  CONSTRAINT FK_Registration_EmployeeID FOREIGN KEY (EMPLOYEE_ID) REFERENCES Employee(EMPLOYEE_ID),
  CONSTRAINT FK_Registration_CourseID FOREIGN KEY (COURSE_ID) REFERENCES Course(COURSE_ID)
);
```

3.Create Course table

```
CREATE TABLE Course (
  "Course-ID" INT,
  "Class-ID" INT,
  "Course-Name" VARCHAR2(255),
  "photo" BLOB,
  "Description" VARCHAR2(1000),
  "notes" VARCHAR2(1000),
  PRIMARY KEY ("Course-ID")
);
```

4.Create Received-Course`table

```
CREATE TABLE Received_Course (
  EMPLOYEE_ID INT,
  RECEIVED_COURSES_NAME VARCHAR2(255),
  RECEIVE_COURSES_ID INT,
  PRIMARY KEY (EMPLOYEE_ID, RECEIVE_COURSES_ID),
  CONSTRAINT FK_Received_Course_EmployeeID FOREIGN KEY (EMPLOYEE_ID)
REFERENCES Employee(EMPLOYEE_ID)
)
```

5.Create Required-Course`table

```
CREATE TABLE Required_Course (
  Employee_ID INT,
  Required_Course_Name VARCHAR2(255),
  Required_Course_ID INT,
  PRIMARY KEY (Employee_ID, Required_Course_ID),
  CONSTRAINT FK_Required_Course_EmployeeID FOREIGN KEY (Employee_ID)
REFERENCES Employee(Employee_ID)
);
```

6.Create Phone `table

```
CREATE TABLE Phone (
  Employee_ID INT,
  Phone_Num VARCHAR2(20),
  PRIMARY KEY (Employee_ID, Phone_Num),
  CONSTRAINT FK_Phone_EmployeeID FOREIGN KEY (Employee_ID) REFERENCES
Employee(Employee_ID)
);
```

7.Create schedule-Classes`table

```
CREATE TABLE schedule_Classes (
  Class_ID INT,
  Class_Name VARCHAR2(255),
  Course_Name VARCHAR2(255),
  Description VARCHAR2(1000),
  Start_Date DATE,
  Last Date DATE,
  Days VARCHAR2(255),
  Time VARCHAR2(100),
  Facilitator VARCHAR2(255),
  Available_Sign VARCHAR2(10),
  Link VARCHAR2(255),
  Attendees INT,
  PRIMARY KEY (Class_ID)
);
```

8.Create Class-Feedback table

```
CREATE TABLE Class_Feedback (
  Feed ID INT,
  Classe Taken VARCHAR2(255),
  Overall_Rating INT,
  Content_Rating INT,
  Facilitator_Rating INT,
  Favorite_Part VARCHAR2(1000),
  Area_Of_Improvement VARCHAR2(1000),
  Additional_Feedback VARCHAR2(1000),
  PRIMARY KEY (Feed_ID)
);
```

9.Create Days`table

```
CREATE TABLE Days (
  DaysName VARCHAR2(255),
  Class_ID INT,
  Date DATE,
  PRIMARY KEY (DaysName, Class_ID, Date),
  CONSTRAINT FK_Days_ClassID FOREIGN KEY (Class_ID) REFERENCES
schedule_Classes(Class_ID)
);
```

Insert random data

-- Insert data into the Employee table

```
INSERT INTO Employee (Employee-ID, Name, Picture, SignUp-Courses, phone, Email,
complete-Courses, Note)
 VALUES
  (1, 'John Doe', 'https://example.com/johndoe.png', 'Intro to Programming', '555-1234',
'johndoe@example.com', 'Database Design', 'Excellent worker'),
  (2, 'Jane Smith', 'https://example.com/janesmith.png', 'Database Design, Intro to
Programming', '555-5678', 'janesmith@example.com', 'Web Development', 'Punctual and
reliable');

-- Insert data into the Registration table
INSERT INTO Registration (Employee-ID, Course-ID)
VALUES
  (1, 1),
  (1, 2),
  (2, 2),
  (2, 3);

-- Insert data into the Course table
INSERT INTO Course (Course-ID, Class-ID, Course-Name, photo, Description, notes)
VALUES
  (1, 1, 'Intro to Programming', 'https://example.com/intro.png', 'Learn the basics of
programming', 'No prerequisites required'),
  (2, 1, 'Database Design', 'https://example.com/database.png', 'Learn how to design and
create databases', 'Some programming experience recommended'),
  (3, 2, 'Web Development', 'https://example.com/webdev.png', 'Learn how to create web
applications', 'Experience with HTML, CSS, and JavaScript required');

-- Insert data into the Received-Course table
INSERT INTO `Received-Course` (Employee-ID, Received-Courses-Name,
Recieve-Courses-ID)
VALUES
  (1, 'Intro to Programming', 1),
  (1, 'Database Design', 2),
  (2, 'Database Design', 2),
  (2, 'Web Development', 3);

-- Insert data into the Required-Course table
INSERT INTO `Required-Course` (Employee-ID, Required-Course-Name,
Required-Course-ID)
 VALUES
  (1, 'Database Design', 2),
  (2, 'Web Development', 3);

-- Insert data into the Phone table
INSERT INTO Phone (Employee-ID, Phone-Num)
VALUES
  (1, '555-1234'),
  (2, '555-5678');

-- Insert data into the schedule-Classes table
INSERT INTO `schedule-Classes` (Class-ID, Class-Name, course-Name, Description,
Start-Date, Last-Date, days, time, facilitator, available-Sign, link, attendees)
VALUES
```

```
  (1, 'Programming 101', 'Intro to Programming', 'Learn the basics of programming',
'2023-06-01', '2023-07-15', 'Monday, Wednesday, Friday', '10:00am - 12:00pm', 'John Smith',
'Open', 'https://example.com/programming101', 20),
  (2, 'Web Development Fundamentals', 'Web Development', 'Learn the fundamentals of web
development', '2023-07-01', '2023-08-15', 'Tuesday, Thursday', '1:00pm - 3:00pm', 'Jane
Doe', 'Open', 'https://example.com/webdevfundamentals', 15);

-- Insert data into the Class-Feedback table
INSERT INTO `Class-Feedback` (Feed-ID, classe-Taken, overall-Rating, content-Rating,
facilitator-Rating, favorite-part, area-Of-Improvement, additional-FeedBack)
VALUES
  (1, 'Programming 101', 4, 3, 5, 'I enjoyed learning about loops', 'More hands-on activities
would be helpful', 'Great course overall'),
  (2, 'Web Development Fundamentals', 3, 4, 3, 'I liked learning about responsive design',
'The pace was a bit too fast', 'Overall a good experience');
```