

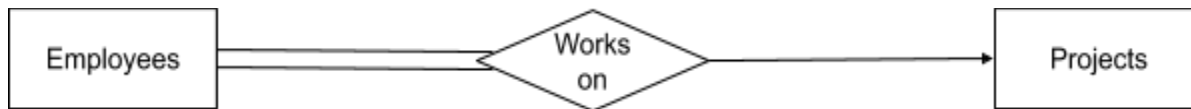
CS 122A : Introduction to Data Management-Assignment 2

Fall 2019

In this assignment, your task is mapping the ER design (given to you at the end of this document) into the relational model. Please note that you should use the given ER design on that page, not the one that you submitted for the Assignment 1. Please make sure to stick to the design and the description provided to you in this document for this assignment. You are required to work on this assignment **in teams of three**.

Deadline : Friday, October 25th (23:45 pm)

Problem 1



In the ER diagram above, even though we do not show it, the Employee entity set has attributes id, Name, Gender, Salary. Assume that id is the key for the Employee entity set.

Furthermore, Project has attributes P-id, P-Name, and Department. Assume that P-id is a key for the Project entity set.

We translate the above ER diagram to the corresponding two relations an instance of which is shown below. Answer the following questions.

ID	Name	Gender	Salary	ProjectID
22	John	Male	50,000	1
34	Amelia	Female	70,000	2
54	John	Male	75,000	4
67	Mary	Female	60,000	1

Employees Table

P-ID	P-Name	Department
1	Talent recruiting	HR
2	Server maintenance	IT

Projects

1. Is the above mapping of ER diagrams into two relations Employee(id, name, gender, salary, Projectid) and Projects(P-id, P-name, Department) with the corresponding primary keys being id, and pid respectively and the Projectid being a foreign key in Employee table a correct mapping of the above ER diagram?

Yes

2. Is above instance of the relation valid based on the corresponding mapping?. If not , which tuple of tuples violate which constraint? Explain.

Yes, the above instance of the relation is valid

3. List all **candidate** key(s) for Employees table given that ID is the primary key.

{ID}, {Salary}, {Name, Salary}, {Name, ProjectID}, {Gender, Salary}, {Gender, ProjectID}, {Salary, ProjectID}

4. List all **superkey**(s) for Employee table given that ID is the primary key.

{ID}, {Salary}, {ID, Name}, {ID, Gender}, {ID, Salary}, {ID, ProjectID}, {Name, Salary}, {Name, ProjectID}, {Gender, Salary}, {Gender ProjectID}, {Salary, ProjectID}, {ID, Name, Gender}, {ID, Name, Salary}, {ID, Name, ProjectID}, {ID, Gender, Salary}, {ID, Gender, ProjectID}, {ID, Salary, ProjectID}, {Name, Gender, Salary}, {Name, Gender, ProjectID}, {Name, Salary, ProjectID}, {Gender, Salary, ProjectID}, {ID, Name, Gender, Salary}, {ID, Gender, Salary, ProjectID}, {Name, Gender, Salary, ProjectID}, {ID, Name, Gender, Salary, ProjectID}

5. We have 3 integrity maintenance policies, abort, cascade and SET NULL. Explain what will happen in these 3 scenarios if we delete the row with id number 2 in Projects table.

Abort - no data is deleted

Cascade - the data corresponding to Amelia is deleted with the project

SET NULL - Amelia's project ID is set to null, and her project is deleted

Consider a relation R with five attributes ABCDEFG. You are given the following dependencies: $A \rightarrow B$, $BC \rightarrow E$ and $DE \rightarrow AF$.

6. Please list all the candidate keys.

{CDEG}, {ACDG}

7. Which one of the following is not in the closure of the given FDs. Show the steps if it is in the closure.

a. $FEG \rightarrow AB$

Not in the closure

b. $ACD \rightarrow F$

Not in the closure. (The only letter not given is G, though, is this a mistake?)

Problem 2

1 Goal

In this problem, you are required to translate the given entity relationship model, that is provided to you in the following pages, into relations for the relational model and produce MySQL DDL statements to create corresponding tables.

2 Description

A. Use the methods presented in class to translate the ER diagram into relations. (60%)

- Translate the ER design into a set of relations. (30%)
- Describe the **primary key**, **domain**, **not NULL** and **inclusion constraints** needed for the relational schema to capture and enforce the semantics of the ER design. (30%)

Example:

Relation:

student(id, depId, name, level, awards)

Domains:

Level= {undergraduate, graduate}

Not Null Attributes : All non key attributes are not NULL.

Inclusion constraints:

student(depId) \subseteq department(id)

B. Write SQL DDL statements for creating the tables corresponding to the relations you developed. Pick suitable data types for each attribute. Also include the appropriate referential integrity constraints and “NOT NULL” constraint while creating the tables. Execute your DDL statements on MySQL and record the system’s response (40%).

Here is an example of a DDL statement for a table:

```
CREATE TABLE Student(  
id          INTEGER NOT NULL,  
name        VARCHAR(40) NOT NULL,  
depId       VARCHAR(10) NOT NULL,  
level       ENUM('undergraduate','graduate') NOT NULL,  
awards      VARCHAR(40),  
PRIMARY KEY (id),  
FOREIGN KEY (depId) REFERENCES Department(depID)  
);
```

Deliverables

Your assignment has to be submitted in two parts to **gradescope** and **EEE**.

1. **Only one team member** has to submit to **gradescope** a **PDF file** named **last_name1_last_name2_last_name3_assignment1.pdf** containing the answers problem 1 and question 2.A.
2. **Only one team member** has to submit to **EEE** a **zip file** named **last_name1_last_name2_last_name3_assignment2.zip** having :
 1. A **SQL script** containing the SQL DDL statements - it should be able to be executed on MySQL command line processor: **"script.sql"**
 2. An **output file** containing the SQL DDL statements and the responses obtained when executing them on MySQL: **"script_output.txt"**

Please note that your script should be runnable and produce the same result as script_output.txt

Instructions

Creating the result of a SQL file: You need to create a TXT file to include your queries and its results. Points may be deducted if you don't follow the instructions. Here are the instructions. **Please do not copy text and type each command. When copying text from a PDF file in the terminal, it does not work sometimes.**

1. Open a blank text file in a text editor and copy the following template into the file. Then, paste your DDL statements after the "-- Paste .." comment.

```
DROP DATABASE IF EXISTS `cs122a`;  
CREATE DATABASE `cs122a`;  
USE `cs122a`;
```

-- Paste your DDL statements in the below

2. Save it as "script.sql".

3. You are going to use **mysql** command-line tool (<https://dev.mysql.com/doc/refman/8.0/en/mysql.html>) to execute your script and generate an output. Execute the following command to load your SQL script and generate an output. Here, we assume that "script.sql" is located in your home directory. The result file name should be script_output.txt. Do not convert it to other formats such as DOC or PDF. In the command prompt (or terminal), execute the following command (not after executing mysql).

```
mysql --force --comments -v -v -u root < yourHomePath/script.sql >
yourHomePath/script_output.txt
```

Do not omit any options. Especially the option **-v** is repeated twice on purpose. If the root account has a password, use the following command.

```
mysql --force --comments -v -v -u root -pYOURPASSWORD < yourHomePath/script.sql >
yourHomePath/script_output.txt
```

* OS Specific instructions -

Windows

1. Put your script in a folder that your account can access. (e.g., d:\)
2. Open a command prompt (cmd) and go to the MySQL folder.
cd C:\Program Files (x86)\MySQL\MySQL Server 8.0\bin
3. Execute the following command.
mysql --force --comments -v -v -u root < d:\script.sql > d:\script_output.txt

OS X

1. Put your script in your home directory. (e.g., /Users/youraccount)
2. Open a terminal and execute the following command.

```
/usr/local/mysql/bin/mysql --force --comments -v -v -u root < /Users/youraccount/script.sql >
/Users/youraccount/script_output.txt
```

If you have a problem to execute mysql tool, here is the default location that you can find it. It might be a good idea to move script.sql to a folder that your account has an access like the above.

OS X: /usr/local/mysql/bin/mysql

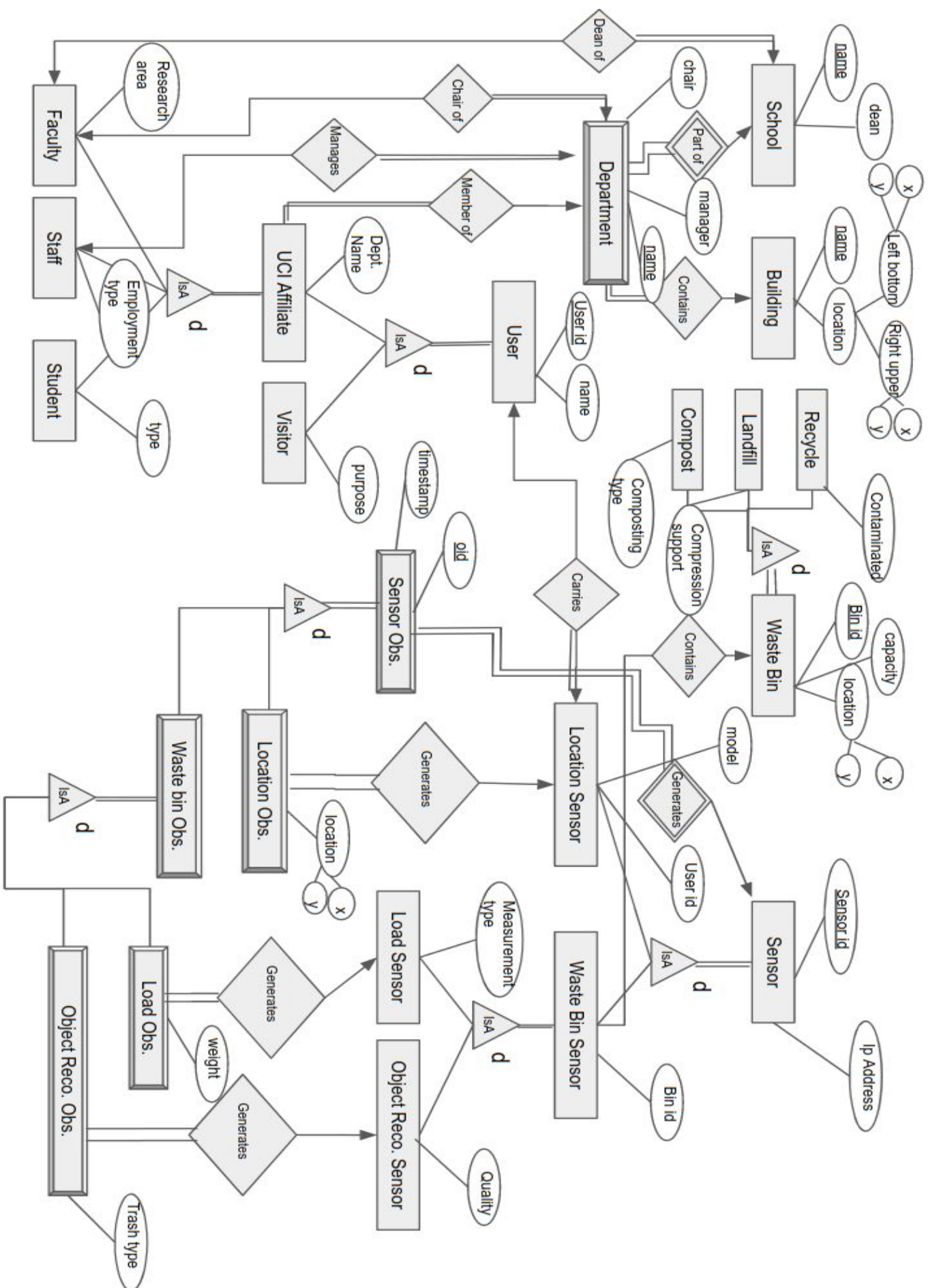
Windows: C:\Program Files (x86)\MySQL\MySQL Server 8.0\bin

Refer to the following guideline for the mysql command-line tool.

[For OS X](#) | [For Windows](#)

ER Design

The following is our ER model for the Smart Waste Management project. You have to develop the rest of our database solution (e.g. mapping to relations) based on the following ER model.



Strong Entities

Relation:

School(name, budget, deanUID)

Domain:

none

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

School(deanUID) \subseteq Faculty(UserID)

Relation:

Building(name, xLeftBottom, yLeftBottom, xRightUpper, yRightUpper)

Domain:

none

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

none

Relation:

Visitor(UserID, name, purpose)

Domain:

none

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

none

Relation:

Student(UserID, name, UCIEmailaddress, DeptName, type)

Domain:

type = {undergrad, grad}

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

Student(DeptName) \subseteq Department(name)

Relation:

Staff(UserID, name, UCIEmailaddress, DeptName, Employmenttype)

Domain:

none

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

Staff(DeptName) \subseteq Department(name)

Relation:

Faculty(UserID, name, UCIEmailaddress, DeptName, Researcharea)

Domain:

none

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

Faculty(DeptName) \subseteq Department(name)

Relation:

Recycle(BinID, capacity, xLocation, yLocation, Contaminated)

Domain:

Contaminated = {yes, no}

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

none

Relation:

Landfill(BinID, capacity, xLocation, yLocation, Compressionsupport)

Domain:

CompressionSupport = {yes, no}

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

none

Relation:

Compost(BinID, capacity, xLocation, yLocation, compostingtype)

Domain:

none

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

none

Relation:

LocationSensor(SensorID, Ip Address, UserID, mode, maxRange)

Domain:

none

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

LocationSensor(UserID) \subseteq Visitor(UserID)

LocationSensor(UserID) \subseteq Faculty(UserID)

LocationSensor(UserID) \subseteq Staff(UserID)

LocationSensor(UserID) \subseteq Student(UserID)

Relation:

LoadSensor(SensorID, Ip Address, Batterypower, BinID, MeasurementType)

Domain:

none

Not Null Attributes: all non-key attributes besides BinID are not NULL

Inclusion Dependencies:

LoadSensor(BinID) \subseteq Recycle(BinID)

LoadSensor(BinID) \subseteq Landfill(BinID)

LoadSensor(BinID) \subseteq Compost(BinID)

Relation:

ObjectRecoSensor(SensorID, Ip Address, Batterypower, BinID, Quantity)

Domain:

none

Not Null Attributes: all non-key attributes besides BinID are not NULL

Inclusion Dependencies:

ObjectRecoSensor(BinID) \subseteq Recycle(BinID)

ObjectRecoSensor(BinID) \subseteq Landfill(BinID)

ObjectRecoSensor(BinID) \subseteq Compost(BinID)

Weak Entities

Relation:

Department(deptName, schoolName, Startdate, Phoneno., buildingName, managerUID, chairUID)

Domain:

none

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

Department(schoolName) \subseteq School(name)

Department(buildingName) \subseteq Building(name)

Department(managerUID) \subseteq Staff(UserID)

Department(chairUID) \subseteq Faculty(UserID)

Relation:

ObjectRecoObs(oid, SensorID, timestamp, Trashtype)

Domain:

Trashtype = {Recycle, Landfill, Compost}

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

ObjectRecoObs(SensorID) \subseteq ObjectRecoSensor(SensorID)

Relation:

LoadObs(oid, SensorID, timestamp, weight)

Domain:

none

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

LoadObs(SensorID) \subseteq LoadSensor(SensorID)

Relation:

LocationObs(oid, SensorID, timestamp, xLocation, yLocation)

Domain:

none

Not Null Attributes: all non-key attributes are not NULL

Inclusion Dependencies:

LocationObs(SensorID) \subseteq LocationSensor(SensorID)