HOMEWORK - 2

MIS686 - Enterprise Database Management

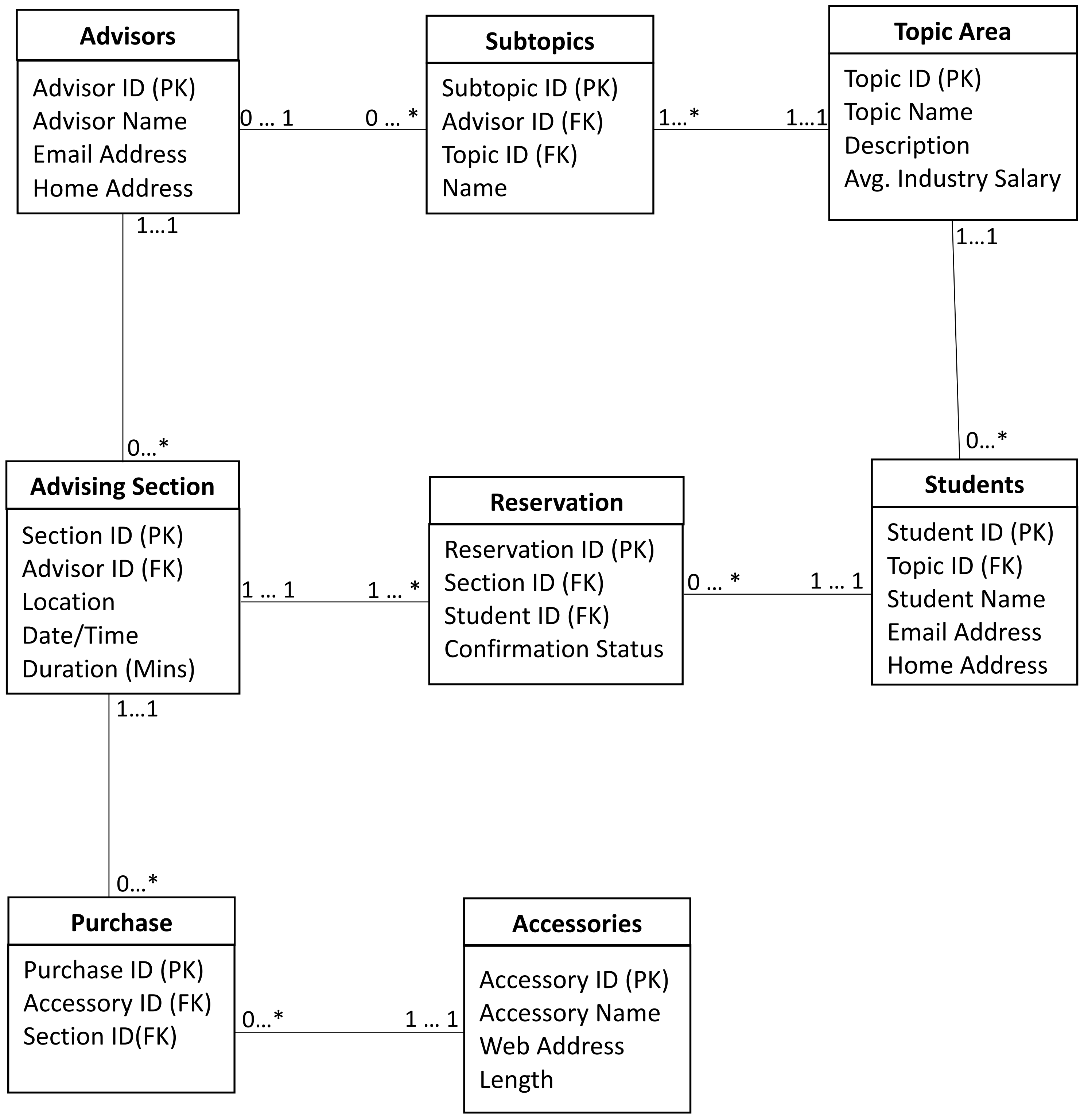
Name: Umadevi Betageri

Red ID: 827194401

Q.No 1. A University is interested in recording data on its advising processes. The University wants to record data on advisors; each advisor has an ID number, a name, an email address, and a home address. The University also wants to record data on students, each of whom similarly have an ID number, a name, an email address, and a home address. Advisors each specialize in at least one and potentially several topic areas, and it is also possible that multiple advisors will have a shared topic area of expertise. Students are also interested in these topic areas; however, you may assume that each individual student is interested in only one topic area. For each topic area, the University wants to store both the name as well as a short description and the average industry salary for graduates working in that area. Students and advisors interact through advising sessions, for which the University would like to record a location, date/time, and duration in minutes. Finally, each advising session may involve some advising materials, such as pamphlets with career information. For each advising material, the University would like to store its name, web address, and length. Each advising session can utilize multiple advising materials.

Ans:

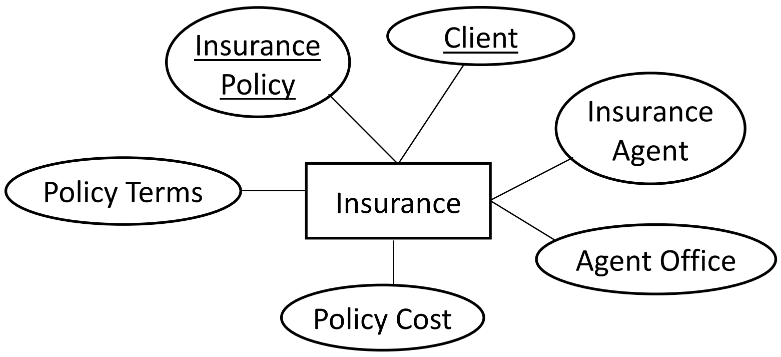
- **Entities: Advisor, Student, Topic Area, Advising Section and Advising Materials.**
- **Associative Entities: Subtopics, Reservations.**
- **Diagram Type: URL Diagram**
- **Assumptions:**
  - Since an advisor is associated with multiple topic areas and a topic area is associated with multiple advisors, I have created an associative entity "Subtopics" between these two with subtopic ID as primary key, assuming each subtopic is associated with only one advisor and one topic area.
  - For advising section I choose section ID as a primary key, since date/time, location can repeat.
  - Since a student can attend multiple advising section and an advising section can have more than one student, I have created an associative entity "Reservation", assuming each reservation is made by one student for one advising section.
  - As mentioned in question, I assumed that a student is interested in only one topic area and his records are stored when he takes at least one topic.
  - Since an advising section can have multiple accessories and an accessory is used in many advising sections, I have created an associative entity "Purchase" with Purchase ID as primary key, assuming purchase ID is associated with one advising section and an accessory.
  - I have assumed that the each advising section is associated with one advisor and an advisor can have zero to multiple advising sections.
  - If an advisor is new and is not assigned any topics, I have assumed he can have minimum of zero subtopic and maximum of many. And, subtopic is with minimum of zero advisor and maximum to one.


Advisors
Advisor ID (PK)
Advisor Name
Email Address
Home Address
0 … 1
0 … *
Subtopics
Subtopic ID (PK)
Advisor ID (FK)
Topic ID (FK)
Name
1…*
1…1
Topic Area
Topic ID (PK)
Topic Name
Description
Avg. Industry Salary
1…1
0…*
1…1
0…*
Advising Section
Section ID (PK)
Advisor ID (FK)
Location
Date/Time
Duration (Mins)
1 … 1
1 … *
Reservation
Reservation ID (PK)
Section ID (FK)
Student ID (FK)
Confirmation Status
0 … *
1 … 1
Students
Student ID (PK)
Topic ID (FK)
Student Name
Email Address
Home Address
1…1
0…*
Purchase
Purchase ID (PK)
Accessory ID (FK)
Section ID(FK)
0…*
1 … 1
Accessories
Accessory ID (PK)
Accessory Name
Web Address
Length

Q.No. 2 Three table designs are presented below. For each table design, check whether the design meets 1st normal form, 2nd normal form, 3rd normal form, and Boyce-Codd normal form. Perform this process in sequence. That is, first check if each design meets the requirements for 1st normal form. If the design does not meet the requirements, then explain why not, and document the necessary modifications to meet 1st normal form. Repeat this process for 2nd normal form, 3rd normal form, and Boyce-Codd normal form.

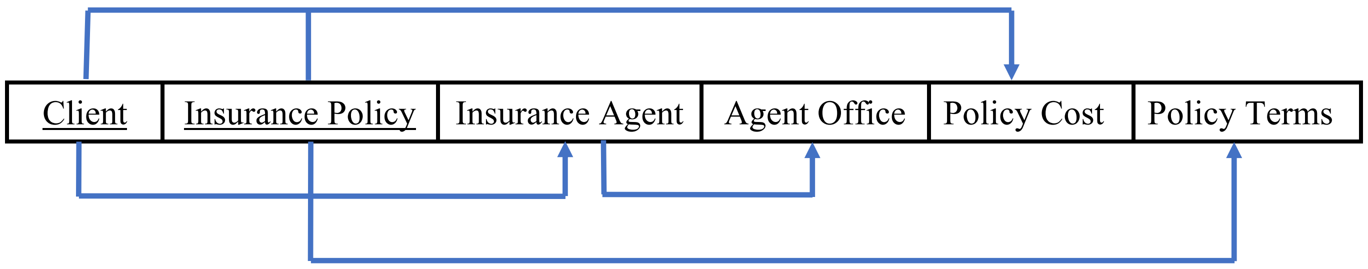
a) **Fields:** client, insurance policy, insurance agent, agent office, policy cost, policy terms
Primary key: client, insurance policy
Dependencies: Client, insurance policy -> policy cost
Insurance policy -> policy terms
Client -> insurance agent
Insurance agent -> agent office

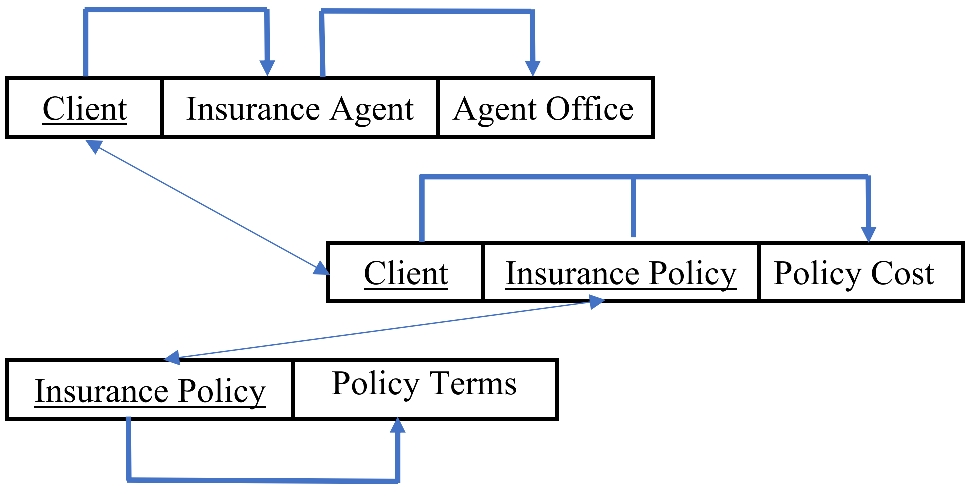
Ans: ER Diagram for unnormalized table



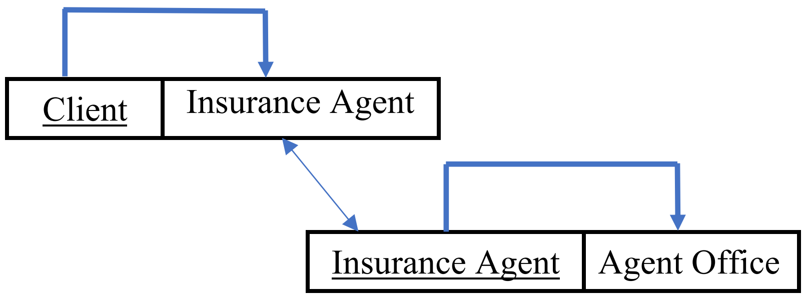
i. Since there are no multiple valued attributes in the table, it is in **1NF**.
ii. There are some attributes (Insurance Agent & Policy Terms) in the table that do not fully depend on composite primary keys (Depends on only one primary key). Hence this table is not in **2NF**. To achieve 2NF, I have created three separate tables.

Assumption: Policy terms is considered as a single document for a particular insurance policy.



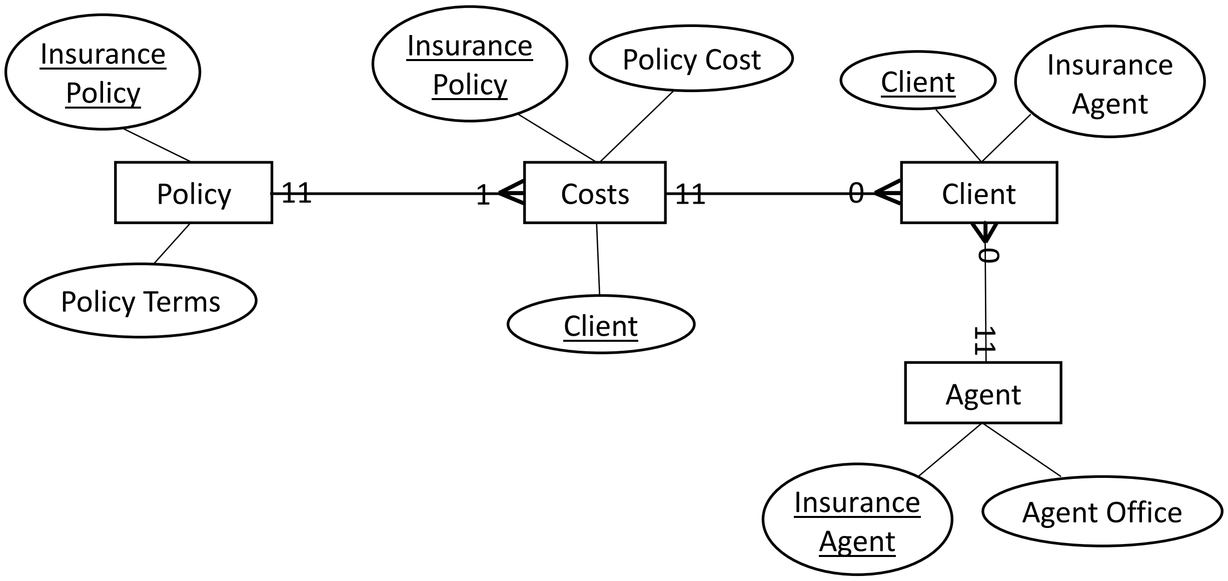


iii. Since Agent office is dependent on non-primary key (Insurance Agent), this table is not in 3NF. Hence, I have created separate table for agent details assuming a clint is associated with one agent and an agent can have many clients.

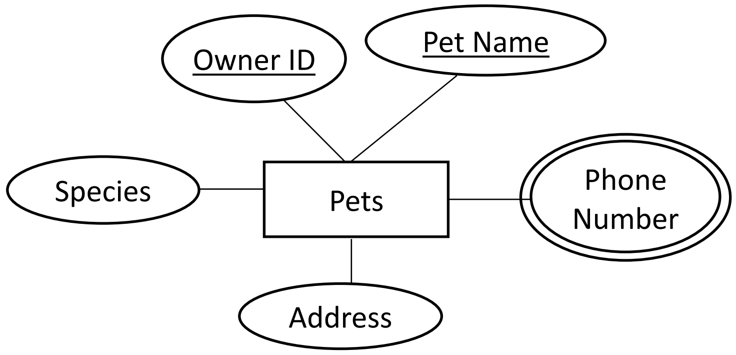


iv. Since none of the primary keys dependent on non-key attributes, the tables are in BCNF

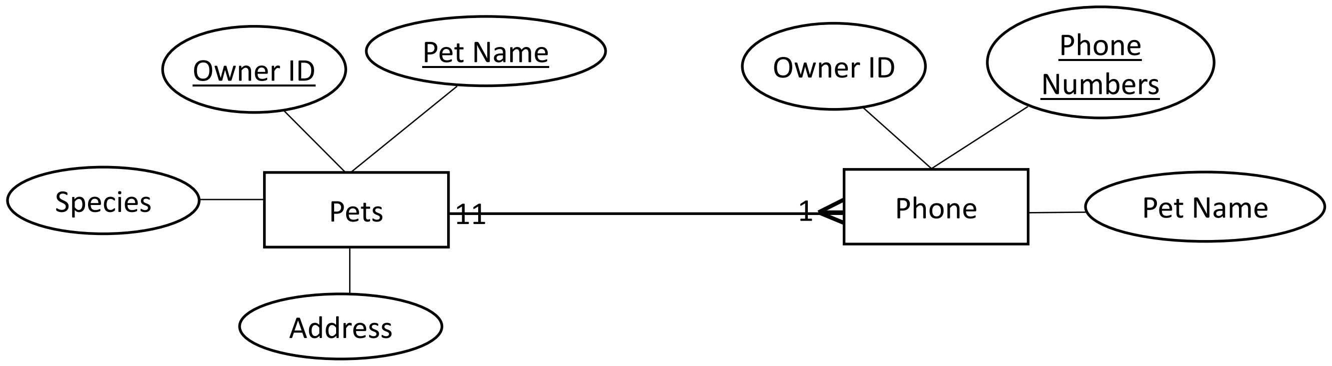
ER Diagram after achieving 2NF and 3NF.



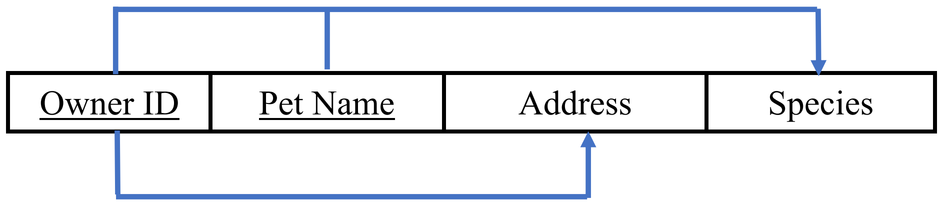
b) Fields: owner ID, pet name, address, phone number (potentially multiple), species
Primary key: owner ID, pet name
Dependencies: Owner ID -> address, phone
Owner ID, pet name -> species
Ans: ER Diagram for unnormalized table

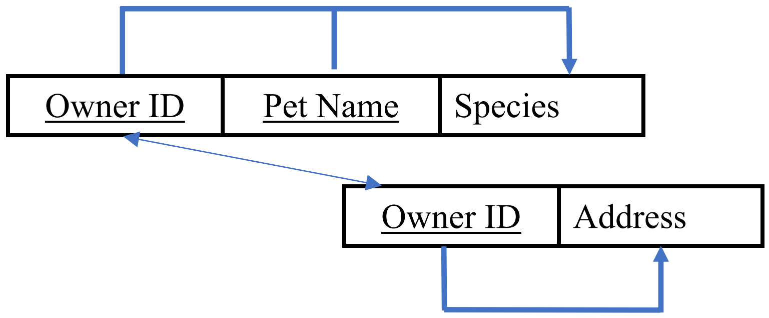


I. In the above unnormalized table the Phone Number attribute is multivalued. Hence it is not in 1NF. Hence, I have created a separate table for Phone Numbers.



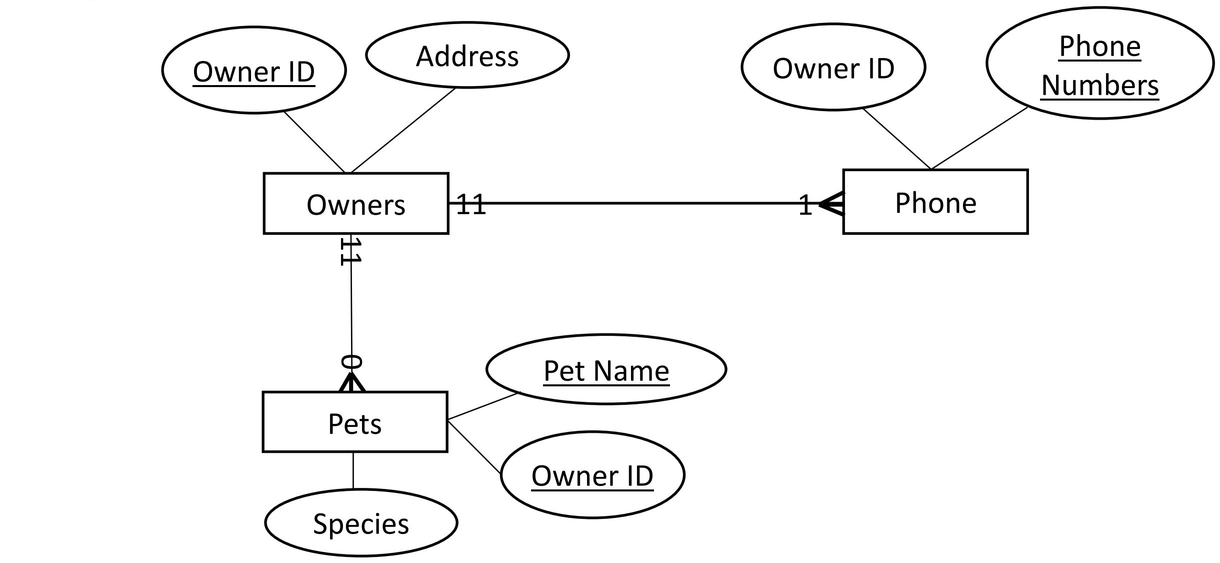
II. Since there is an attribute (Address) in the table that do not fully depend on composite primary keys (Depends only one Owner ID), this table is not in **2NF**. To achieve 2NF, I have created separate table.





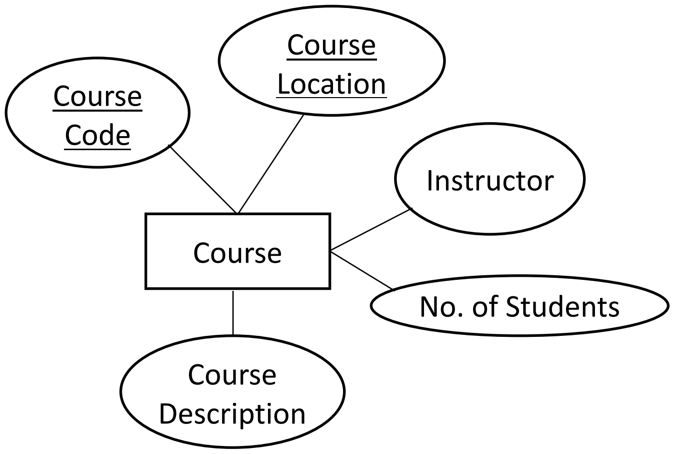
III. Since none of the attributes depends on non-primary attribute, the tables are in **3NF**

IV. Since none of the primary attributes are dependent on non-primary attributes, the tables are in **BCNF.**

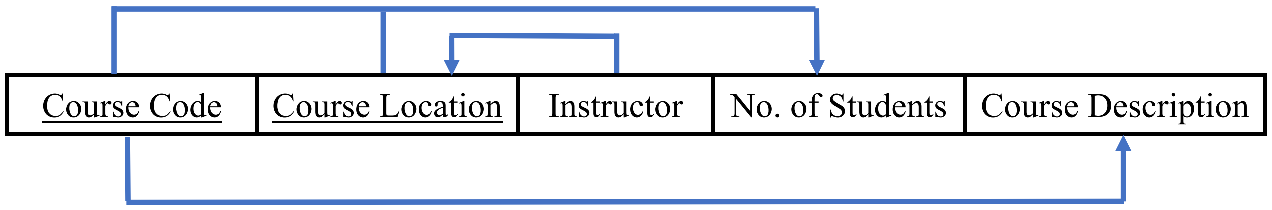


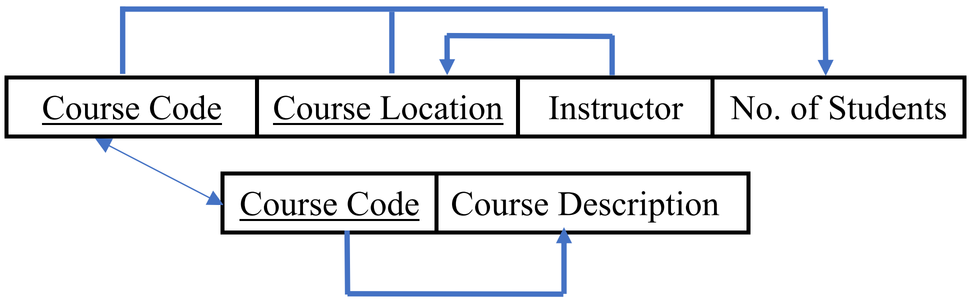
c) Fields: course code, course location, instructor, number of students, course description
Primary key: course code, course location
Dependencies: Course code, course location -> number of students
Course code -> course description
Instructor -> course location

ER Diagram for unnormalized table



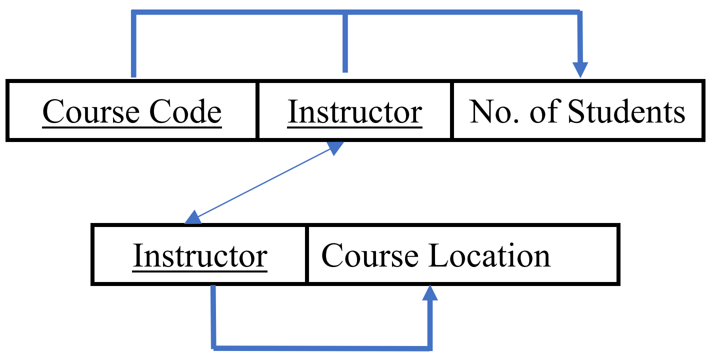
i. Since there are no multiple valued attributes in the table, it is in 1NF.
ii. Since there is an attribute (Course Description) in the table which do not fully depend on composite primary keys (depends on only Course Code), this table is not in **2NF**. To achieve 2NF, I have created separate table.



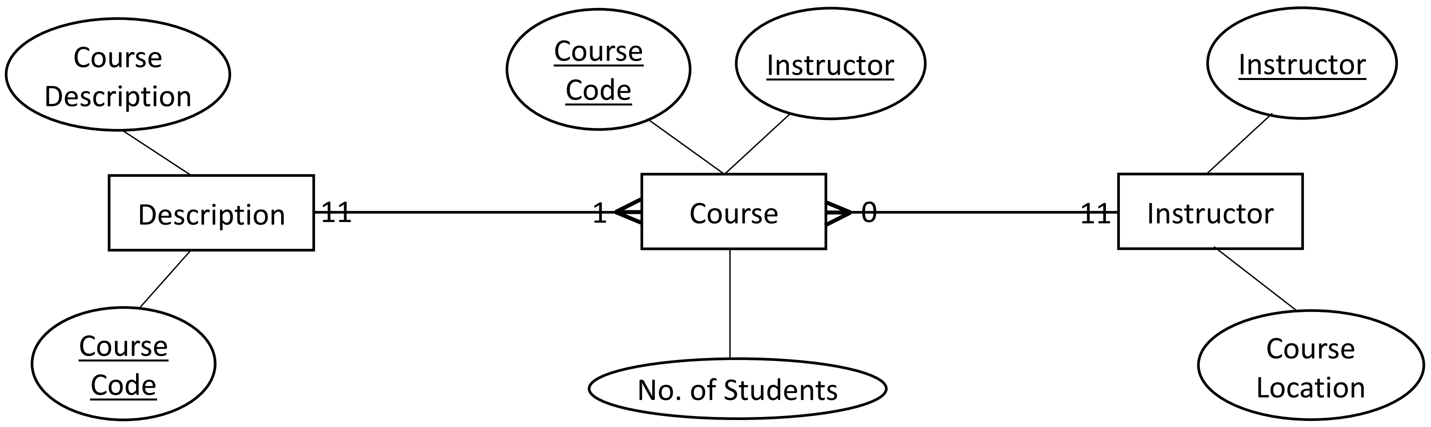


iii. Since none of the attributes depends on non-primary attribute, the tables are in **3NF**
iv. In the above table, there is a primary key (Course Location) which is dependent on non-key attribute. Hence it is not in **BCNF.** To achieve BCNF, I have created separate table for Course

Location and Instructor with Instructor as a primary key to maintain connection with other tables.



ER Diagram after achieving 2NF and BCNF.



Q.No 3. Below is a UML class diagram completed in class (Week 3 Example 1). Create database tables reflecting this UML class diagram. You may either a) submit SQL code that would be used to create the database tables or b) submit screenshots showing the structure of tables you have created. While you should record foreign keys in your tables where indicated, you do not need to link the tables via SQL.

Ans: SQL code for creating database table for Wikipedia:

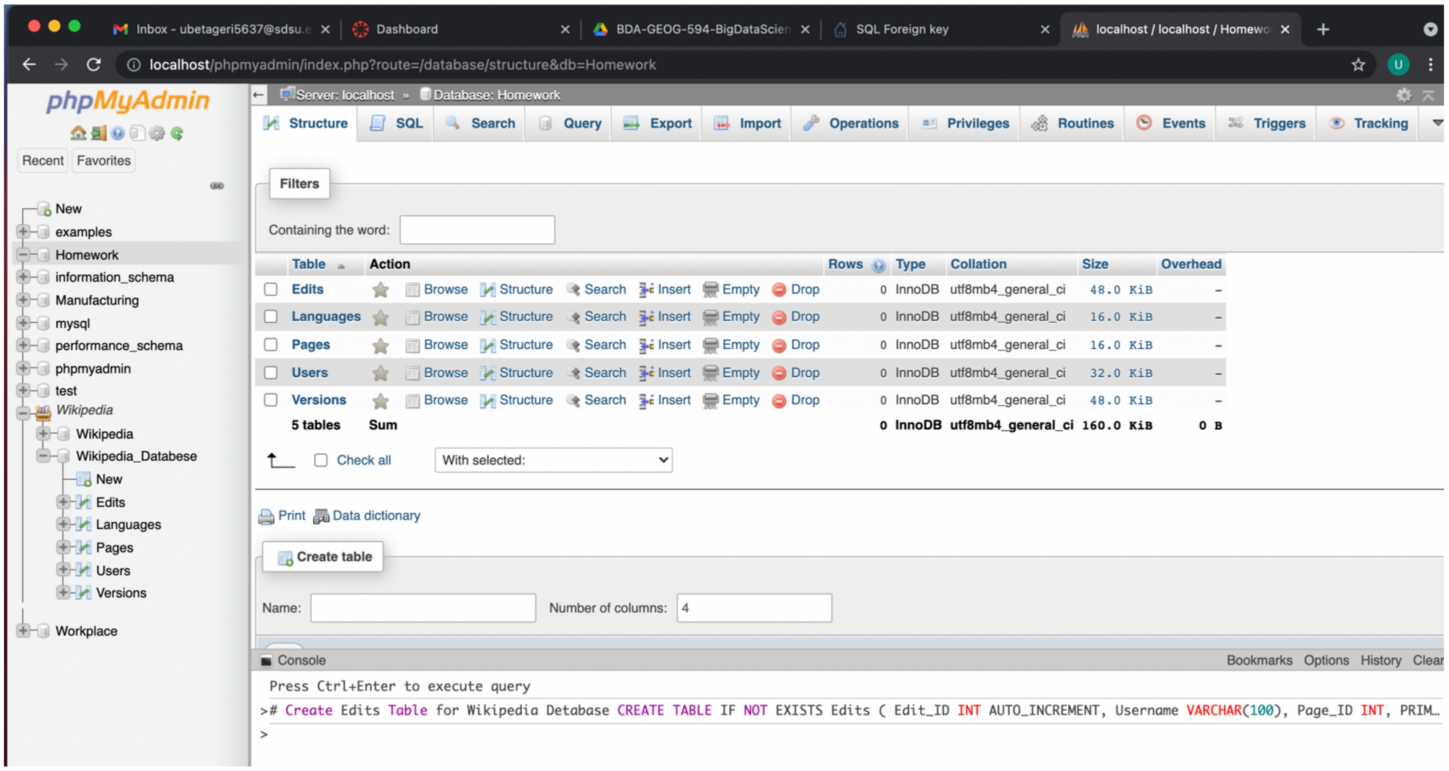
```
# Create Languages Table for Wikipedia Database
CREATE TABLE IF NOT EXISTS Languages (
   Language_Code INT AUTO_INCREMENT,
   Language_Name VARCHAR(100),
   PRIMARY KEY (Language_Code)
);
# Create Users Table for Wikipedia Detabase
CREATE TABLE IF NOT EXISTS Users (
   Username VARCHAR(100),
   Legal_Name VARCHAR(200),
   Email_Address VARCHAR(100),
   Language_Code INT,
   PRIMARY KEY (Username),
   FOREIGN KEY (Language_Code) REFERENCES Languages (Language_Code)
);

# Create Pages Table for Wikipedia Detabase
CREATE TABLE IF NOT EXISTS Pages (
   Page_ID INT AUTO_INCREMENT,
   Page_Name VARCHAR(100),
   Texts VARCHAR(5000),
   Date_Created Date,
   PRIMARY KEY (Page_ID)
);

# Create Versions Table for Wikipedia Detabase
CREATE TABLE IF NOT EXISTS Versions (
   Version_ID INT AUTO_INCREMENT,
   Page_ID INT,
   Language_Code INT,
   PRIMARY KEY (Version_ID),
   FOREIGN KEY (Languages_Code) REFERENCES Languages (Language_Code),
   FOREIGN KEY (Page_ID) REFERENCES Pages (Page_ID)
);

# Create Edits Table for Wikipedia Detabase
CREATE TABLE IF NOT EXISTS Edits (
   Edit_ID INT AUTO_INCREMENT,
   Username VARCHAR(100),
   Page_ID INT,
   PRIMARY KEY (Edit_ID),
   FOREIGN KEY (Page_ID) REFERENCES Pages (Page_ID),
   FOREIGN KEY (Username) REFERENCES Users (Username)
);
```

Attached Screen Shots

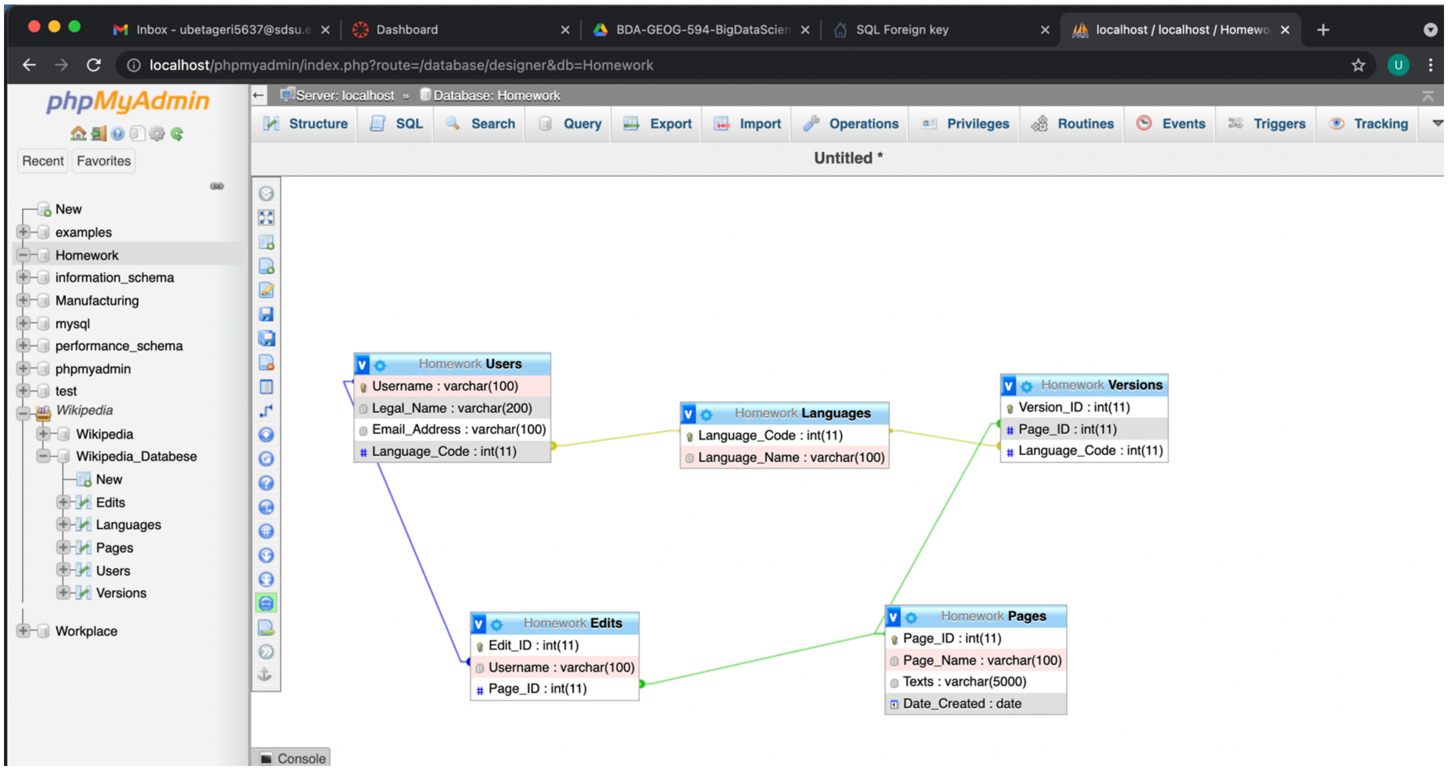


localhost/phpmyadmin/index.php?route=/database/structure&db=Homework

Server: localhost » Database: Homework

Table	Action	Rows	Type	Collation	Size	Overhead
Edits	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	–
Languages	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	–
Pages	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	–
Users	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	–
Versions	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	48.0 KiB	–
5 tables	Sum	0	InnoDB	utf8mb4_general_ci	160.0 KiB	0 B

```
Press Ctrl+Enter to execute query
># Create Edits Table for Wikipedia Detabase CREATE TABLE IF NOT EXISTS Edits ( Edit_ID INT AUTO_INCREMENT, Username VARCHAR(100), Page_ID INT, PRIM…
>
```



localhost/phpmyadmin/index.php?route=/database/designer&db=Homework

Untitled *