Department of Data Science, Bishop Heber College Tiruchirappalli

Modern Database Systems Lab

NAME: PAVITHIRAN .V ROLL NO : 235229122

Lab1. Data Modeling using ER Diagrams Objectives

In this lab, you will create Entity-Relationship diagramsfor the given business scenarios and understand the terminologies.

Question1: For the ER diagram shown below, answer the following questions

1. How many entities are here in the Employee database?

   Ans: 4

2. How many relationships are there?

   Ans: 6

3. What is the primary key for the entity EMPLOYEE?

   Ans: ssn

4. Is there any unary relationship in this ERD?. What is it?

   Ans : YES, SUPERVISION

5. Which one is the weak entity here?

   Ans : DEPENDENT

6. Which one is the multivalued attribute?

   Ans: locations

7. Is there any derived attribute?

   Ans: yes

8. Which one is the weak entity?. Why?

   Ans :DEPENDENT, Enclosed with double rectangle.

9. How many departments can an employee work for?

Ans: many to 1

10. How many employees can work for a Department?

Ans: 1 to many

11. What are the total participations here?

Ans: employee-work_for-department ,

manages-department,

employee-work_on,

employee-work_on-project and dependent

Question2: Create schema for all entities and relationships from ER diagram of Employee Database

```
CREATE TABLE EMPLOYEE(

      SSN INT(10) PRIMARY KEY,

      BDATE DATE,

      NAME VARCHAR2 (15),

      FNAME VARCHAR2 (15),

      MINIT VARCHAR2 (15),

      LNAME VARCHAR2(15),

      ADDRESS VARCHAR2(30),

      SALARY    FLOAT(7,2),

      SEX VARCHAR2(2));


CREATE TABLE DEPENDENT(

      NAME VARCHAR2(10) FOREIGN KEY,

      SEX VARCHAR2(2),

      BIRTH_DATE DATE,

      RELATIONSHIP VARCHAR2(10));


CREATE TABLE DEPARTMENT(
      NAME VARCHAR2(10) ALTERNATE KEY,

      NUMBER INT(12) PRIMARY KEY,

      ADDRESS_LINE1 VARCHAR2(10),

      ADDRESS_LINE2 VARCHAR2(10),

      STATE VARCHAR2(10),

      COUNTRY VARCHAR2(10));
```

```
CREATE TABLE PROJECT(

    NAME VARCHAR2(10) ALTERNATE KEY,

    NUMBER VARCHAR(10) PRIMARY KEY,

    LOCATION VARCHAR(30));
```

Question3: Create ER diagram for the following University application

The university database stores details about university students, courses, the semester a student

took a particular course (and his mark and grade if he completed it), and what degree program each

student is enrolled in. The database is a long way from one that'd be suitable for a large tertiary institution,

but it does illustrate relationships that are interesting to query, and it's easy to relate to when you're

learning SQL. We explain the requirements next and discuss their shortcomings at the end of this section.

Consider the following requirements list:

- The university offers one or more programs.
- A program is made up of one or more courses.
- A student must enroll in a program.
- A student takes the courses that are part of her program.

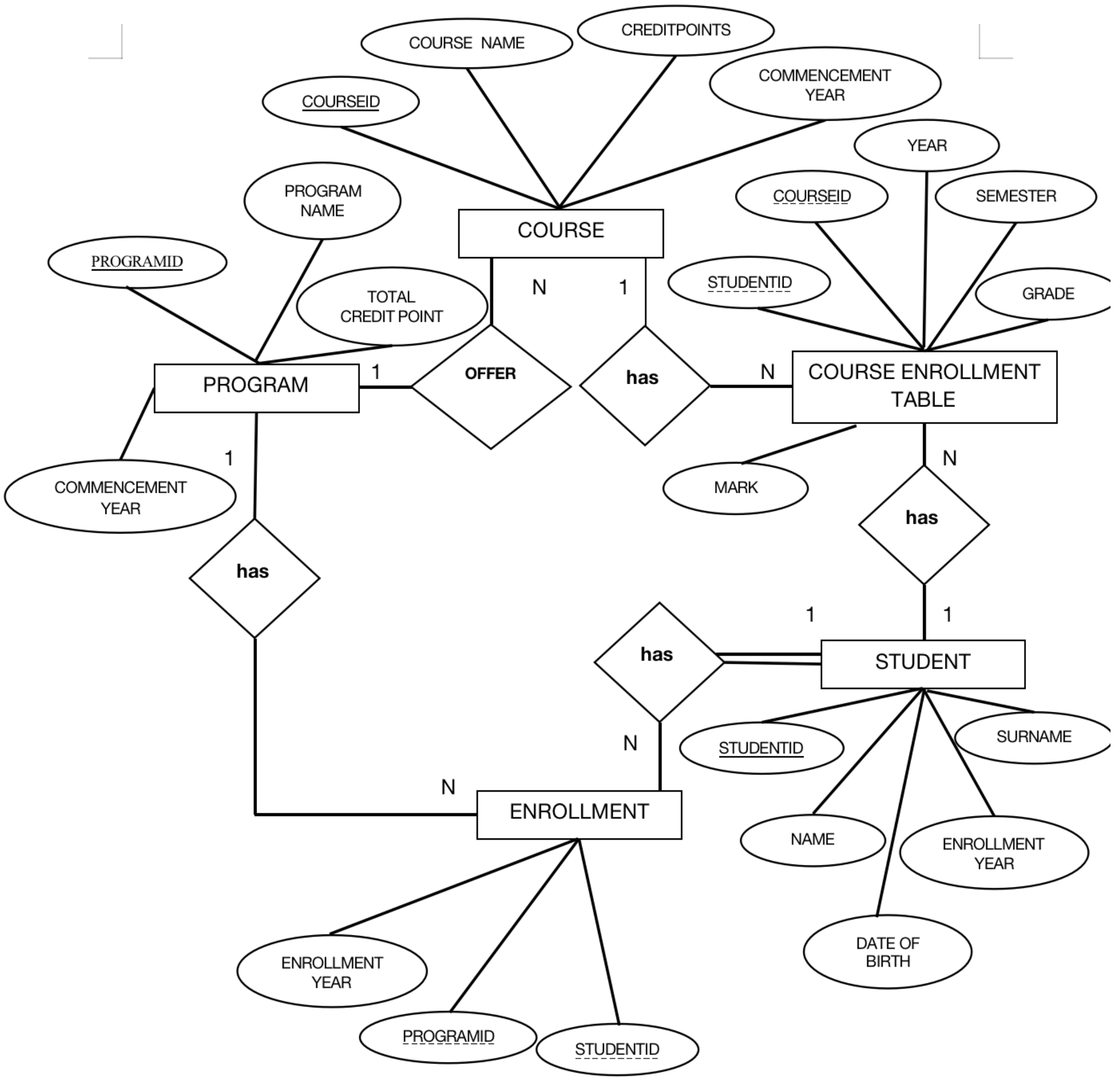
- A program has a name, a program identifier, the total credit points required to graduate, and the year it commenced.

- A course has a name, a course identifier, a credit point value, and the year it commenced.

- Students have one or more given names, a surname, a student identifier, a date of birth, and the year they first enrolled. We can treat all given names as a single object—for example, “John Paul.”

- When a student takes a course, the year and semester he attempted it are recorded. When he finishes the course, a grade (such as A or B) and a mark (such as 60 percent) are recorded.

- Each course in a program is sequenced into a year (for example, year 1) and a semester (for example, semester 1).



Question 3B: Create schema for all entities and relationships from the university ER diagram

```
CREATE TABLE PROGRAM (
    PROGRAMID INT PRIMARY KEY,
    PROGRAMNAME VARCHAR(100),
    TOTALCREDITPOINTS INT,
    COMMENCEMENTYEAR INT
);

CREATE TABLE COURSE (
    COURSEID INT PRIMARY KEY,
    COURSENAME VARCHAR(100),
    CREDITPOINTS INT,
    COMMENCEMENTYEAR INT,
    PROGRAMID INT,
    FOREIGN KEY (PROGRAMID) REFERENCES PROGRAM(PROGRAMID)
);

CREATE TABLE STUDENT (
    STUDENTID INT PRIMARY KEY,
    NAMES VARCHAR(100),
    SURNAME VARCHAR(100),
    DATEOFBIRTH DATE,
    ENROLLMENTYEAR INT
);

CREATE TABLE COURSEENROLLMENT (
    ENROLLMENTID INT PRIMARY KEY,
    STUDENTID INT,
    COURSEID INT,
    YEAR INT,
    SEMESTER INT,
    GRADE VARCHAR(2),
    MARK DECIMAL(5,2),
    FOREIGN KEY (STUDENTID) REFERENCES STUDENT(STUDENTID),
    FOREIGN KEY (COURSEID) REFERENCES COURSE(COURSEID)
);

CREATE TABLE ENROLLMENT (
    ENROLLMENTID INT PRIMARY KEY,
    STUDENTID INT,
    PROGRAMID INT,
    ENROLLMENTYEAR INT,
    FOREIGN KEY (STUDENTID) REFERENCES STUDENT(STUDENTID),
    FOREIGN KEY (PROGRAMID) REFERENCES PROGRAM(PROGRAMID)
);
```

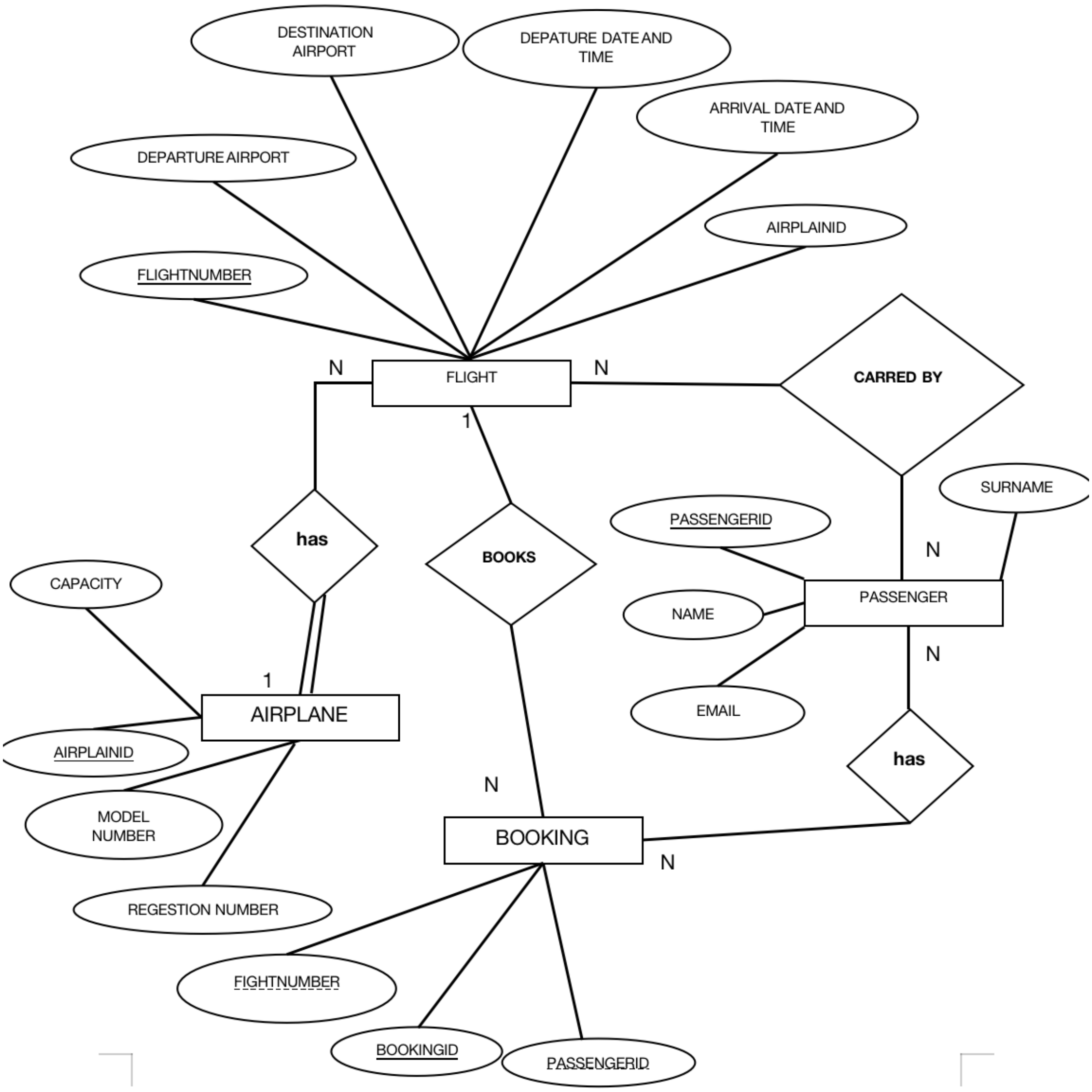
Question4: Create ER diagram for the Flight database application

The flight database stores details about an airline’s fleet, flights, and seat bookings. Again, it’s a hugely

simplified version of what a real airline would use, but the principles are the same.

Consider the following requirements list:

- The airline has one or more airplanes.
- An airplane has a model number, a unique registration number, and the capacity to take one or more passengers.
- An airplane flight has a unique flight number, a departure airport, a destination airport, a departure date and time, and an arrival date and time.
- Each flight is carried out by a single airplane.
- A passenger has given names, a surname, and a unique email address.
- A passenger can book a seat on a flight



Question 4B: Create schema for all entities and relationships from the flight ER diagram

```
CREATE TABLE AIRPLANE (

    AIRPLANEID INT PRIMARY KEY,

    MODELNUMBER VARCHAR(100),

    REGISTRATIONNUMBER VARCHAR(100) UNIQUE,

    CAPACITY INT

);


CREATE TABLE FLIGHT (

    FLIGHTNUMBER INT PRIMARY KEY,

    DEPARTUREAIRPORT VARCHAR(100),

    DESTINATIONAIRPORT VARCHAR(100),

    DEPARTUREDATETIME TIMESTAMP,

    ARRIVALDATETIME TIMESTAMP,

    AIRPLANEID INT,

    FOREIGN KEY (AIRPLANEID) REFERENCES AIRPLANE(AIRPLANEID)

);


CREATE TABLE PASSENGER (

    PASSENGERID INT PRIMARY KEY,

    NAME VARCHAR(100),

    SURNAME VARCHAR(100),

    EMAIL VARCHAR(100) UNIQUE

);

CREATE TABLE BOOKING (

    BOOKINGID INT PRIMARY KEY,
    PASSENGERID INT,

    FLIGHTNUMBER INT,

    FOREIGN KEY (PASSENGERID) REFERENCES PASSENGER(PASSENGERID),

    FOREIGN KEY (FLIGHTNUMBER) REFERENCES FLIGHT(FLIGHTNUMBER)

);
```