Sri Lanka Institute of Information Technology

B.Sc. Honours Degree in Information Technology

Specialized in Information Technology

Final Examination
Year 2, Semester 1 (2019)

IT2040 – Database Management Systems

Duration: 2 Hours

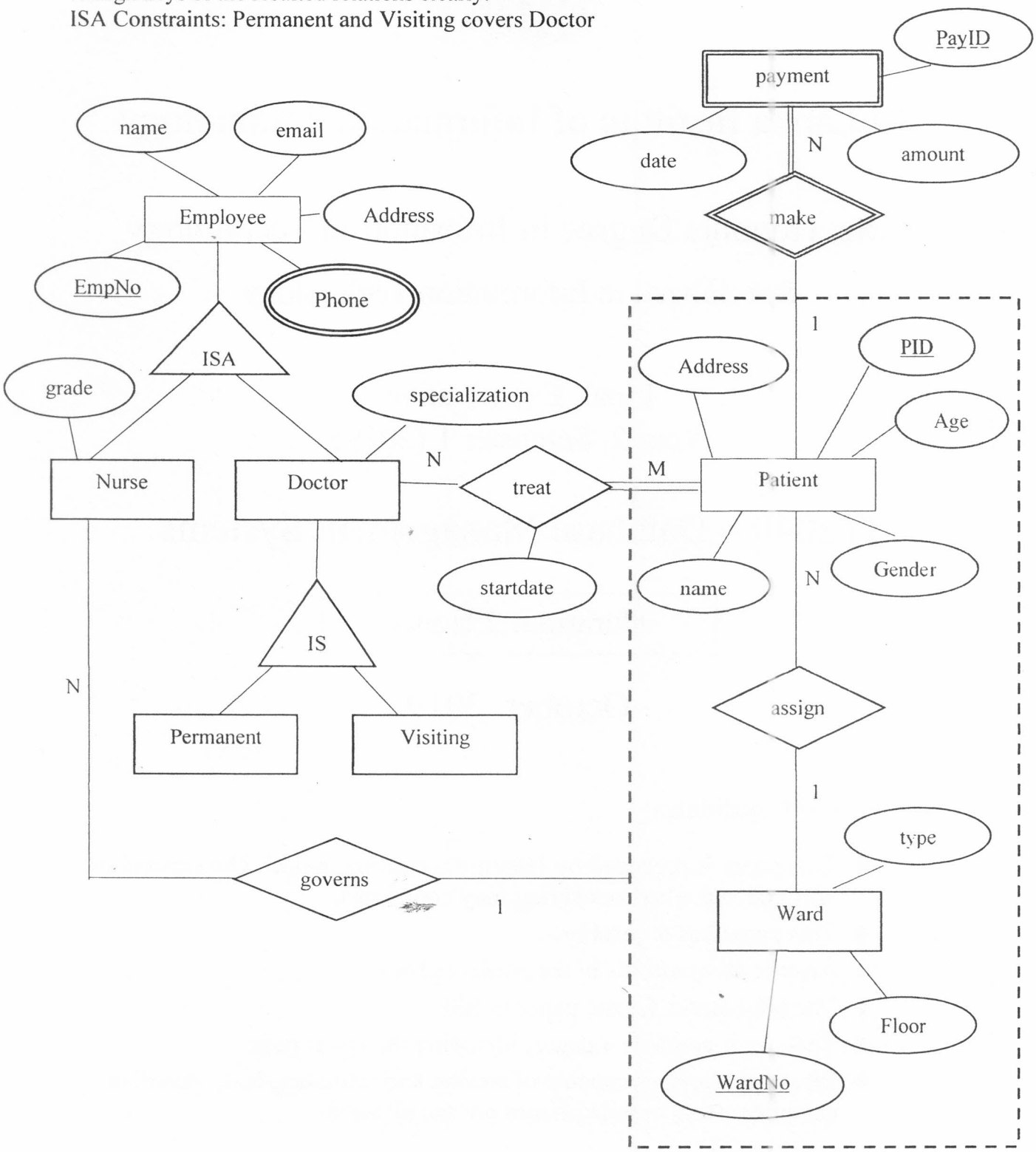
October 2019

Instructions to Candidates:

- This paper is preceded by 10minutes reading period. The supervisor will indicate when answering may commence.
- This paper has 4 questions.
- Answer all questions in the booklet given.
- The total marks for the paper is 100.
- This paper contains 4 pages, including the cover page.
- Electronic devices capable of storing and retrieving text, including calculators and mobile phones are not allowed.

Question 1 (20 marks)

Convert the following EER model in to the relational model.Indicate the primary keys and the foreign keys of the resulted relations clearly.
ISA Constraints: Permanent and Visiting covers Doctor



Question 2 (15 marks)

Consider a relation **R (A, B, C, D, E)** with the following set of functional dependencies over **R**:

$$F = \{B \rightarrow D, D \rightarrow E, AB \rightarrow C, B \rightarrow E\}$$

a) Find all the keys that follow from the given FDs using attribute closure. (6 marks)

b) Is **R** in 3NF? Give reasons for your conclusion. (3 marks)

c) Is **R** in BCNF? Give reasons for your conclusion. If R is not in BCNF, convert it to a set of BCNF relations. (6 marks)

Question 3 (25 marks)

Consider the following relations relate to patient management system database.

Doctor (SSN, FirstName, LastName, Specialty, YearsOfExperience, PhoneNum)

Patient (SSN, FirstName, LastName, Address, DOB, pDocSSN)

Prescription (Id, Date, DoctorSSN, PatientSSN)

Prescription_Medicine (PrescriptionId, TradeName, NumOfUnits)

Medicine (TradeName, UnitPrice, GenericFlag)

The **Doctor** relation has attributes Social Security Number (SSN), first and last names, specialty, the number of experience years, and the phone number. The **Patient** relation has attributes SSN, first and last names, address, date of birth (DOB), and the SSN of the patient's primary doctor (pDocSSN). **Prescription** relation has attributes id, the date in which the prescription is written, the SSN of the doctor who wrote the prescription, and the SSN of the patient to whom the prescription is written. The **Medicine** relation has attributes trade name, unit price, and whether or not the medicine is generic (True or False). The **Prescription_Medicine** relation stores the medicines written in each prescription along with their quantities (NumOfUnits).

Write the following queries in Relational Algebra.

a) List the trade names of generic medicine with unit price less than Rs.1000. (2 marks)

b) List the first and last name of patients whose primary doctor is 'John Smith'. (4 marks)

c) List the first and last name of doctors who are not primary doctors to any patient. (5 marks)

d) List the prescription IDs which contain both 'Vitamin C' and 'Aspirin' as trade name. (5 marks)

e) List the trade name of medicines that appears in the most of the prescriptions. (9 marks)

Question 4 (40 marks)

Consider the following relations in a Movie Database.

Movie (mID: int, title: varchar(30), year: date, rank: real)
Actor (aID: *int*, fname: varchar(30), lname: varchar(30), gender: char(10))
Director (dID: int, fname: varchar(30), lname: varchar(30), gender: char(10))
Casts (*actorID: int, movieID: int, role: varchar(10))*
Genre (genID: int, *type: varchar(10)*)
DirectsMovie (*directorID: int, movieID: int*)
MovieGenre (*movieID: int,* genID: int)
Reviewer (reviewID: int, reviewName: varchar(20))
Rating (*movieID: int,* reviewID: int, reviewStars: int)

The attributes of the **Movie** relation are id (*mID*), *title*, *year* it was made and the *rank* of the movies. The **Actor** relation has attributes to record the id *(aID)*, first name (*fname*), last name (*lname*) and *gender* ('M' or 'F') of actors. The **Director** relation has attributes to store the id (*dID*), first name (*fname*), last name(*lname*) and *gender* ('M' or 'F') of directors. The **Casts** relation associates the movie with actors who had acted in them and contains the *role* ('lead', 'support', or 'other') they played. The **Genre** relation stores information of movies types (such as 'Romantic', 'Horror', 'Action', 'Fiction', 'Documentary', 'Thriller', and 'Drama'). The **DirectsMovie** relation stores information on movies and their directors. The **MovieGenre** relation stores information related to movies respective genre. A Rating system is carried out based on the Reviewers' number of stars. The **Reviewer** relation has attributes to record the reviewer id, name. The **Rating** relation has attributes to store the movie id, reviewer id and reviewer's number of stars. Primary keys are underlined

a) Use SQL queries to answer following questions.
   i. Display names of all the Actors in the movie "The Princess Bride". (5 marks)

   ii. Display all directors who directed 500 movies or more, in descending order of the number of movies they directed. List the directors' names and the number of movies each of them directed. (6 marks)

b) Create a view named **movieDirection** that contains movie title, movie genre type, year, director's first name, last name and actors for those movies. (8 marks)

c) Create a function named as **getActorCount**, which takes movie title as the input and returns the number of actors that acted in that particular movie. (9 marks)

d) Assume that rank of the Movie are based on the number of stars the reviewers have rewarded. Write a trigger to update the rank when the reviewer places stars in a new movie. (12 marks)