The University of Sussex

SCHOOL OF ENGINEERING & INFORMATICS

Coursework declaration – to be included with coursework when not personally submitting

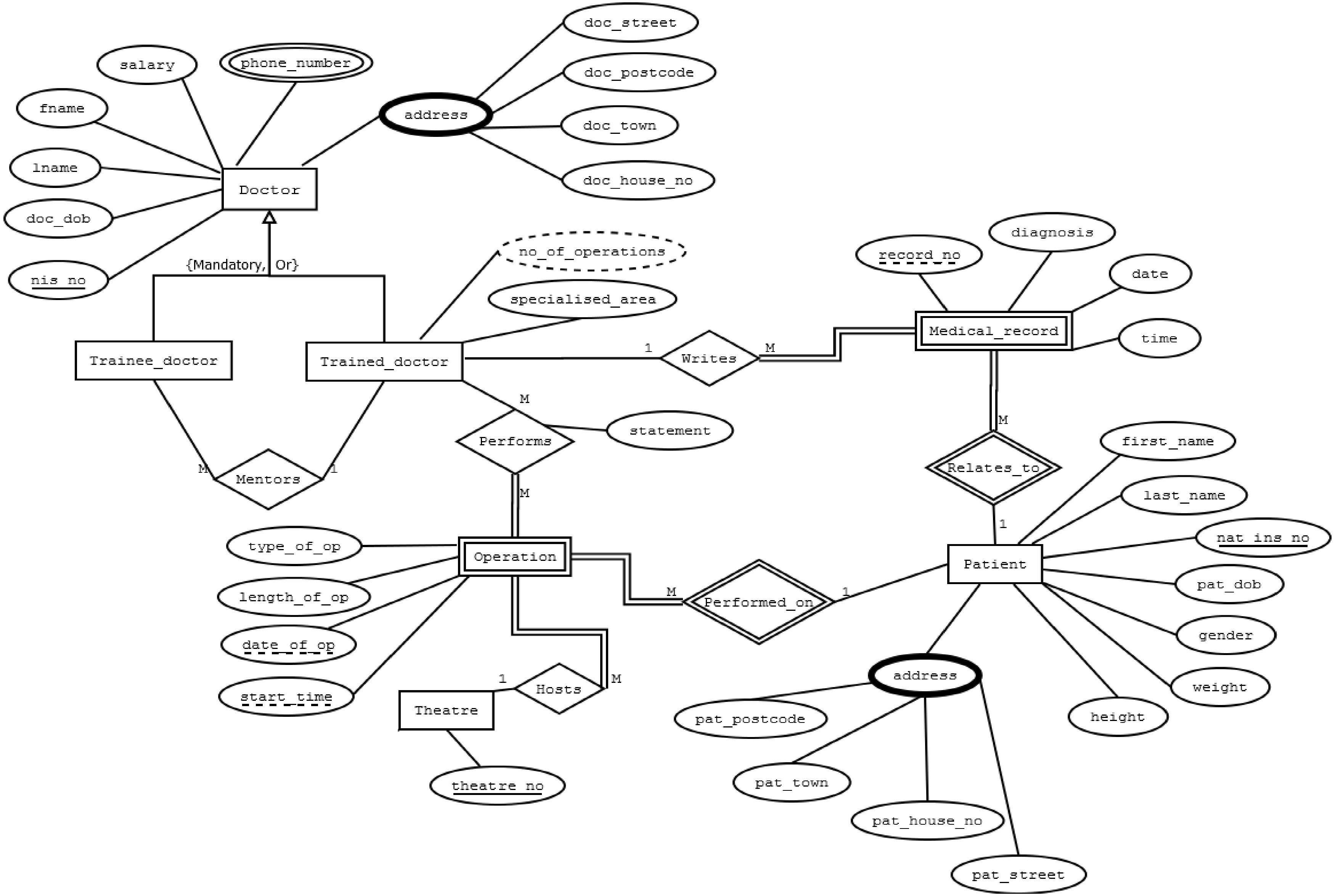
In making this submission I declare that my work contains no examples of misconduct, such as plagiarism, collusion or fabrication of results.

Title of Module: Databases and Application Development

Title of Assignment: Coursework 1

Date: 5/11/2019

Question 1


doc_street
salary
phone_number
doc_postcode
fname
address
doc_town
lname
Doctor
doc_house_no
doc_dob
no_of_operations
diagnosis
record_no
{Mandatory, Or}
nis_no
date
specialised_area
Medical_record
time
Trainee_doctor
Trained_doctor
1
Writes
M
M
M
statement
Performs
first_name
Relates_to
M
Mentors
1
last_name
M
1
type_of_op
nat_ins_no
Operation
Patient
length_of_op
M
Performed_on
1
pat_dob
date_of_op
gender
1
Hosts
M
address
start_time
weight
Theatre
pat_postcode
height
theatre_no
pat_town
pat_house_no
pat_street

Assumptions:

- In an operating theatre, only one operation can be taken place at one time i.e. two operations cannot take place in the same theatre at the same time.
- Trainee doctors cannot perform operations, or write up medical records and do not have a specialised area (yet)
- Both the doctors and patients only have one home address.
- Trained doctors do not have to perform any operations, or write any medical records, but can do this.
- The date_of_op attribute stores the year along with the day and month. (This is needed to ensure each operation can be identified uniquely).
- Trainee doctors can be mentored by non-doctors, as they '**may**' have a mentor who is also a doctor.
- If there are no patients / doctors operations cannot be performed.
- If there are no patients / doctors medical records cannot be written

b.)

Schema:

Patient (nat_ins_no, first_name, last_name, pat_dob, gender, height, weight, pat_postcode, pat_street, pat_town, pat_house_no)
primary key nat_ins_no

Doctor (nis_no, fname, lname, salary, doc_dob, doc_street, doc_postocde, doc_town, doc_house_no)
primary key nis_no

Trainee (traineeID, mentor)
primary key traineeID
foreign key traineeID references Doctor (nis_no)
foreign key mentor references Trained_doctor (docID)

mentor can be null, if and only if, the mentor is not a doctor

Trained_doctor (docID, specialised_area)
primary key docID
foreign key docID references Doctor (nis_no)

no_of_operations is a derived field and can be calculated through operations entity

PhoneNumber (phone_number, doctorID)
primary key (phone_number, doctorID)
foreign key nis_no references Doctor (nis_no)

Theatre (theatre_no)
primary key theatre_no

Operation (start_time, date_of_op, patientID, theatreID, length_of_op, type_of_op)
primary key (start_time, date_of_op, patientID)
foreign key theatreID references Theatre (theatre_no)
foreign key patientID references Patient (nat_ins_no)
theatreID NOT NULL

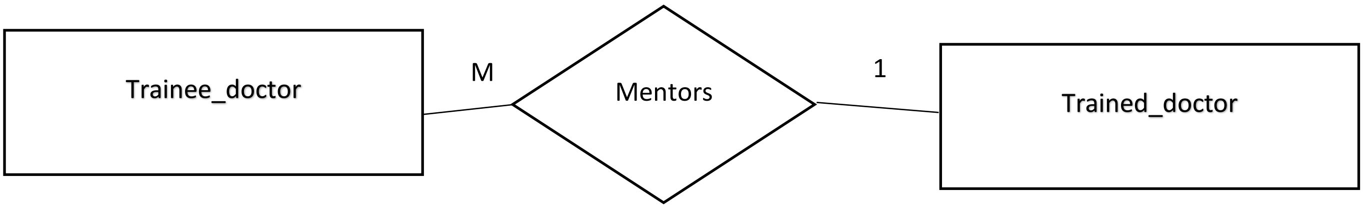
Key does not need to include theatreID, as the start_time, date_of_op and patientId would make for a unique identifier, as a patient cannot have two operations that start at the same time, on the same day. Theatre_no cannot be null in order to retain referential integrity, and since it is not park of the primary key, NOT NULL must be explicitly stated.

Medical_record (record_no, patientID, doctorID, diagnosis, date, time)
primary key (record_no, patientID)
foreign key doctorID references Trained_doctor (docID)
foreign key patientID references Patient (nat_ins_no)
doctorID NOT NULL

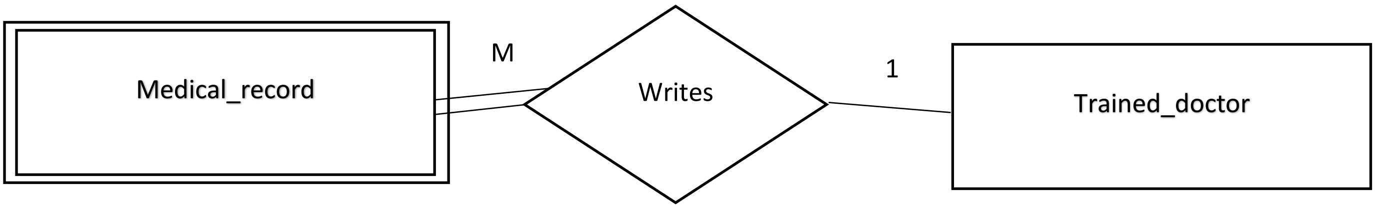
Key does not need to include doctorID, as it is already unique with record_no and patientID. This is because a patient can have a number of records, i.e. patientID of 'abc' records stored as 'abc1, abc2, abc3' etc. doctorID cannot be null in order to retain referential integrity, and since it is not park of the primary key, NOT NULL must be explicitly stated.

Performs (doctorID, op_time, op_date, patientID, statement)
primary key (doctorID, op_time, op_date, patientID)
foreign key doctorID references Trained_doctor (docID)
foreign key op_time, op_date, patientID references Operation (start_time, date_of_op, patientID)

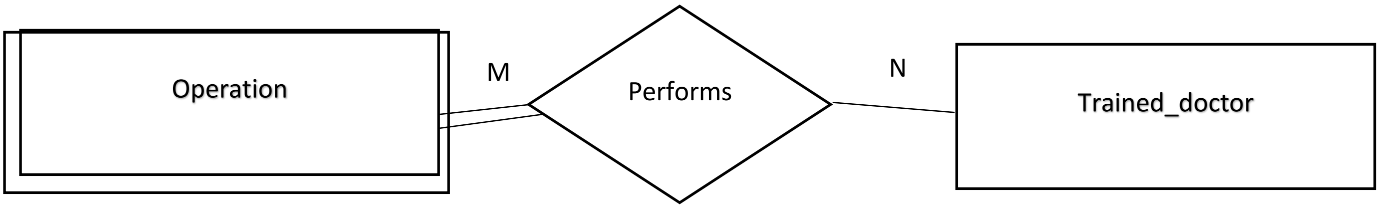
Cardinality Choices



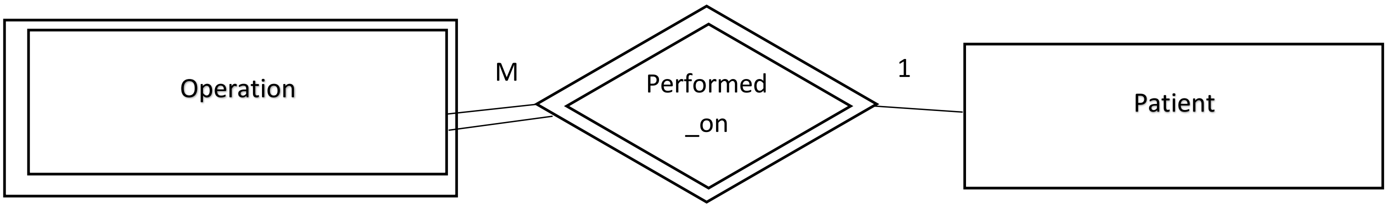
A trainee doctor can be trained by 1 trained doctor (senior doctor). 1 trained doctor can mentor many trainee doctors. Partial participation as trained doctor does not have to be a mentor. This is a 1:M relationship. States in the schema that trainee doctors "may have a mentor, who is also a senior doctor", so partial participation. Mentor may not have to be doctor then.



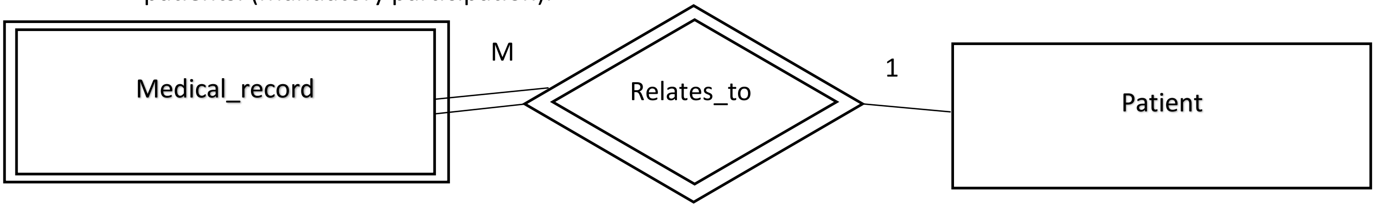
1 trained doctor can write many medical records. 1 medical record is written by 1 doctor. (M: 1). Trained doctors can exist without any medical records (partial participation). Schema states that medical records cannot exist without doctors. (Mandatory participation). "Such record has been made by exactly one doctor".



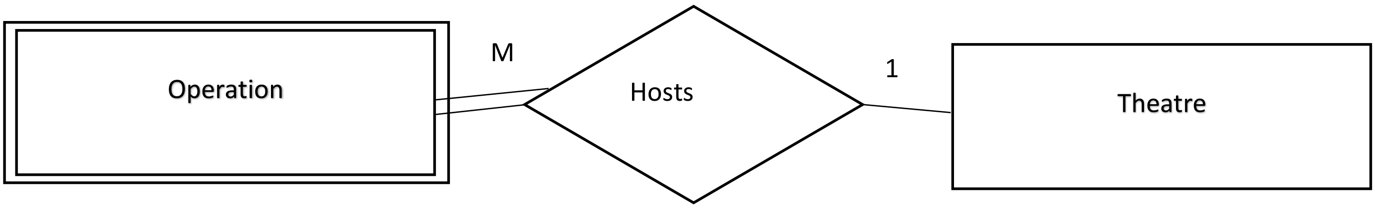
1 or more trained doctors can perform an operation. Many operations can be performed by 1 doctor. (M: N). Trained doctors can exist without any operations (partial participation). Operations cannot exist without doctors. (Mandatory participation). Schema states: "Operations are only put on record once we know who the **doctors are** and who the patient is" – only recorded when we can a doctor.



Schema states: "An operation is carried out by one or several doctors on a single patient", so 1 patient can be present in many operations. One operation is performed on 1 patient. (M: 1). Patients can exist without any operations (partial participation). Operations cannot exist without patients. (Mandatory participation).



One medical records is related to one patient. 1 patient can have many medical records (M: 1). Patient can exist without any medical records (partial participation). Medical records cannot exist without patients. (Mandatory participation), supported as schema states: "Operations are only put on record once we know who the doctors are and who the **patient** is"



Schema states: "For each operation we store the theatre number". One operations is held in one theatre. 1 theatre can host many operations (M: 1). Theatre can exist without any operations (partial participation). Operations cannot exist without operating theatres. (Mandatory participation).

Question 2

Supplier (supplier_no, supplier_name)
primary key supplier_no

City (name, state, population)
primary key (name, state)

Project (proj_no, estimated_cost, supplier)
primary key proj_no
foreign key supplier references Supplier(supplier_no)
supplier NOT NULL

PhoneNumber (phone_number, employeeID)
primary key (phone_number, employeeID)
foreign key employeeID references Employee(nis_no)

Employee (nis_no, street_name, postcode, house_no, married_to, m_date)
primary key nis_no
foreign key married_to references Employee (nationalins_no)

Possible candidate key for Employee could be a composite of street_name, postcode and house_no if truly unique, this however wouldn't be the case if two employees lived together.

Participates (nis_no, proj_no, contract_no, name, state)
primary key (nis_no, proj_no)
foreign key nis_no references Employee (nis_no)
foreign key proj_no references Project (proj_no)
foreign key (name, state) references City (name, state)

name and state are not part of the primary key as a certain employee and project will only come from one city. This means that it would be redundant to have the city attributes in the primary key as the nis_no and proj_no create a unique key. i.e. once nis_no is fixed and proj_no is fixed, we already know the city.

Question 3

1. An employee cannot be married to themselves. In this recursive relationship, an employee if married must be married to a different employee, so the nis_no for person and married_to are different. This is shown through the foreign key declaration of **married_to**. This is shown through making the primary and foreign key distinct in the schema, but is not shown in the ERD. Example: foreign key married_to references Employee (**nationalins_no**)
   Here a different name is used for the key, to ensure that the primary key of employee is separate/distinct to the married_to variable foreign key, (which references a primary key of another employee)

2. As we know, the is_married_To relationship is 1:1. This means that one employee can only be married to one other employee. This means that employees can be married only to other employees, when they in fact could be married to someone that is not an employee. This means that the ERD can only represent marriages between two employees, not two people, as one of the employees may be married to someone who is not an employee. So a constraint that is not in the ERD, is that an employee could be married to a person that is not an employee.