Rabya Ghafoor

Database Management Project

Healthcare Database

1)- Define the information content of your database.

Normally, hospitals and medical offices would store their patient data in a traditional file system such as Microsoft Excel, Open office (compatible with windows, MAC, Linux) or Google doc spreadsheets. The major drawback of using such a file system is that the data is apart of the application program which works only with a specific type of application that all users must be trained to use. This healthcare management database is created to store data records of OBGYN patients with gynecological cancers, patient assessment, and treatment information. This database will provide an important way of monitoring and improving the value of healthcare services which will result in better patient care. Few data records are as below: 1) Patients Personnel Information 2) Care Center Name 3) Details about patients most recent office visit 4) Treatment results 5) Medication prescribed 6) The database administrator in a healthcare setting is often called the Medical Record Administrator - who has the additional responsibility of managing patient related information, accreditation, and re- imbursement information too. The Entity of the database may be below: 1) Physician 2) patients 3) Care Center 4) Visit 6) History 7) Diagnosis 8) Assessment 9) Medicines 10) Order etc. etc.

a)-Define a set of entities and appropriate attributes for each entity.

 Minimum 10 entities.

In the given relationship diagram below the entities and associated attributes are summarized in table below

Entities	Attributes
Care Center	Care Center ID, Care Center Name
Nurse	Nurse ID, Nurse Name
Bed	Bed #, Examination Room #, Room Type, Block Floor, Block Code, Room Availability
Patient Info	Medical Record #, Patient Name, Bed #
Assessment	Appointment ID, Appointment Datetime, Visit Notes, Patient Weight, Patient Height, Patient Blood Pressure, Patient Pulse, Patient Temperature
Physician	Physician ID, Physician Name, Subspecialty, Salary
Physician Dx	Diagnosis Datetime
Diagnosis	Diagnosis Code, Diagnosis Name
Medication Consumption	Consume Date, Dose
Medication	Medication Name, Total Cost
Treatment	Treatment Code, Treatment Datetime, Treatment Name, Result
Order Note	Order ID, Order Datetime

b)-Define a set of relationships that might exist between/among entities and attributes. Such

 relationships may include one-to-one, one-to-many and many-to-many associations.

- Each patient is assigned to one bed (one-to-one)
- The physician admits or refers more than one patient (one-to-many)
- A physician can make more than one diagnoses (one-to-many)
- A patient can have more than one diagnosis (one-to-many)
- A patient can consume more than one item/medication (one-to-many)
- A patient can receive more than one order of tests or medications (one-to-many)
- An order can consists of more than one treatment and vice versa (many-to-many)
- A patient can have more than one assessment (one-to-many)

c)-Define a set of constraints that may be imposed on data.

A not null constraint will be imposed to prevent null values from being entered in one or more columns within a table.

A primary key implied as (+) and foreign key implied as (-) will be used to define relationships between tables.

A Check constraint on certain columns to checks for a specific condition before inserting data into a table. If the data passes all the Check constraints then the data will be inserted into the table otherwise the data for insertion will be discarded.

2. Define an ER diagram

Care Center
🔑 Care Center ID	char(6)
▪ Care Center Name	varchar(30)

Nurse_Care Center
🔑 Nurse ID	char(5)
🔑 Care Center ID	char(5)

Nurse
🔑 Nurse ID	char(5)
Nurse Name	varchar(20)
Subspecialty	varchar(50)
Salary	char(10)

Bed
🔑 Bed number	char(6)
Room number	char(5)
Care Center ID	char(5)
Room type	varchar(20)
Block floor	char(2)
Block code	char(2)
room availability	varchar(5)
Column	Integer(10)

Assessment
🔑 Appointment ID	char(6)
Appointment date time	timestamp
Visit Notes	varchar(100)
Patient Weight	char(5)
Patient Blood Pressure	char(5)
Patient Pulse	char(5)
Patient Temperature	char(5)
Patient medical record number	char(9)
Physician ID	char(9)
Nurse ID	char(9)

Patient
🔑 medical record number	char(6)
patient name	varchar(20)
Bed number	char(9)
Physician ID	char(9)

Physician
🔑 Physician ID	char(6)
Physician Name	varchar(20)
Subspecialty	varchar(50)
Salary	char(5)

Physician Dx
Diagnosis date time	timestamp
Physician ID	char(9)
Patient medical record number	char(9)
Diagnosis Code	varchar(9)

Medication Consumption
Consumption Date	timestamp
Dose	char(5)
Medication code	char(9)
Patient medical record number	char(9)

Diagnosis
🔑 Diagnosis Code	varchar(6)
Diagnosis Name	varchar(40)

Order
🔑 Order ID	char(6)
Order Date time	timestamp
Medication code	char(9)
Physician ID	char(9)
Patient medical record number	char(9)

Medication
🔑 Medication Code	char(6)
Medication Name	varchar(50)
Total Cost	char(10)

Order_Treatment
🔑 OrderOrder ID	char(5)
🔑 TreatmentTreatment ID	integer(10)

Treatment
🔑 Treatment ID	Integer(10)
Treatment date time	timestamp
Treatment Name	varchar(20)
Results	varchar(100)
Patient medical record number	char(9)
Physician Physician ID	char(9)

3)- Define a relational schema for your database design.

Here is a relational database schema

Make sure that you have both one-to-many and many-to-many associations.

a)-Define one or more realistic key(s) for every relation scheme. Use both simple and composite keys.

Treatment (<u>Treatment ID, Order ID,</u> Treatment Datetime, Treatment Name, Result, Patient Medical Record #, Physician ID)

Order Note (<u>Order ID, Treatment ID,</u> Order Datetime, Medicine ID, Patient Medical Record, Physician ID,)

Medicine Consumption (<u>Medicine ID,</u> Consumption Date, Dose, Patient Medical Record #)

Medication (<u>Medicine ID,</u> Medicine Name, Total Cost)

Patient Info (<u>Patient Medical Record #,</u> Patient Name, Bed #, Physician ID)

Physician Dx (<u>Physician ID, Patient Medical Record #,</u> Diagnosis Datetime, Diagnosis Code)

Physician (Physician ID, Physician Name, Subspecialty, Salary)

Nurse (<u>Nurse ID, Center ID</u>, Nurse Name)

Care Center (<u>Center ID, Nurse ID</u>, Care Center Name)

Physician ID is a primary key in physician table and a foreign key in physician diagnosis table to show that a physician can make more than one diagnosis in the same way that each patient can be diagnosed with more than one condition (one-to-many)

Patient medical record # is a primary key in patient info and a foreign key in treatment to show that one patient can receive more than one treatment (one-to-many relationship)

The Order Note and Treatment table have a composite key Order ID and Treatment ID to show a many-to-many relationship.

The Nurse and Care Center table have a composite key Nurse ID and Center ID to show a many-to-many relationship since more than one nurse can work for the same Care Center and one nurse can work at more than Care Center.

b)-Define a realistic set of Functional / Multi-Valued Dependencies (when appropriate) for every

relation scheme.

R1:

Physician ID	Physician Name	Order ID	Order Datetime

R2:

Order ID	Treatment ID	Treatment Name	Treatment Datetime	Order Datetime	Treatment Result

R3:

Physician ID	Physician Name	Diagnosis Datetime

C-Check whether your relational schema is in 2NF, 3NF, BCNF, 4NF.

The first relationship is in 1NF since not all attributes or non-key fields are not entirely dependent on the primary key as opposed to third relationship where all attributes are dependent on the single primary key.

The primary key is a composite key on order ID and treatment ID. But the value of treatment result depends only on treatment ID, not on the entire primary key.

d)-Put your relational schema in the highest normal form that is possible.

Note that, every relation scheme should be in a specific normal form to have the relational

schema in that normal form.

To normalize this table, move treatment result to a second table as shown below. Removing treatment result bring this table into Third Normal Form

Order Note (Order ID, Treatment ID, Treatment Name, Treatment Datetime, Order Datetime)
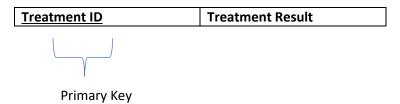
Treatment (Treatment ID, Treatment Result)

Order Note

Order ID	Treatment ID	Treatment Name	Treatment Datetime	Order Datetime

        Primary Key

Treatment

Treatment ID	Treatment Result

        Primary Key

4) Implementation: Create your database using Oracle, or MySQL, or… to Perform the following operations.

A) You are required to execute SQL queries that include the following operations. For each query, provide the

SQL statements along with the output. For each of the following, try different SQL statements (i.e., using one

relation, more than one relations,…).

Create tables:

```
create table if not exists center
(
center_id char(5) not null primary key,
center_name varchar(30),
nurse_id char(5) not null,
foreign key(nurse_id) references nurse(nurse_id)
);
```

Populating the tables:

```
insert into nurse values('N-202', 'Sarah Kinsley', 'C-225', 'Gynecologic Oncology', 167923 );
insert into nurse values('N-201', 'Andrea Smith', 'C-281', 'Gynecologic Oncology', 134908);
insert into nurse values('N-212', 'Martha Stewart', 'C-201', 'Gynecologic Oncology', 155678);
insert into nurse values('N-009', 'Lauren Khols', 'C-232', 'Gynecologic Oncology', 160406);
insert into nurse values('N-150', 'Monica Norton', 'C-281', 'Gynecologic Oncology', 162856);
insert into nurse values('N-230', 'Drashty Mody', 'C-217', 'Maternal-Fetal Medicine', 125000);
```

Select: select involving one/more conditions in Where Clause

a. Find the name of patients whose first name starts with a 'C' and room number is less than 30

Query

```
select patient_name
from patient_info
where patient_name LIKE 'C%' and bed_number < 30;
```

Result

patient_name
Casandra Smith

b. Find the patient's name who's primary OBGYN doctor's name is Desai.

Query

```
select patient_name from patient_info, physician
where physician_name = 'Desai'
and patient_info.physician_id = physician.physician_id;
```

Result

patient_name
Hadassa Rocha

2. select with aggregate functions (i.e., SUM, MIN, MAX, AVG, COUNT)

a. Find the number of available rooms

Query

```sql
SELECT count(*) "Number of available rooms"
FROM bed
WHERE available ='true';
```

Result

Number of available rooms
7

b. Count the number of unavailable rooms for each block in each floor.

Query

```sql
SELECT block_floor as "Floor",
       count(*) AS  "No of available
FROM bed
WHERE available='true'
GROUP BY block_floor
HAVING count(*) =
  (SELECT max(zz) AS highest_total
    FROM
      ( SELECT block_floor ,
              count(*) AS zz
        FROM bed
        WHERE available='true'
        GROUP BY block_floor ) AS t );
```

Result

Floor	No of available rooms
1	7

3. select with Having, Group By, Order By clause

a. Count the number of unavailable rooms for each block in each floor.

Query

```
SELECT block_floor AS "Floor",
       block_code AS "Block",
       count(*) "Number of unavailable rooms"
FROM bed
WHERE available='false'
GROUP BY block_floor,
         block_code
ORDER BY block_floor,
         block_code;
```

Result

Floor	Block	Number of unavailable rooms
1	1	1
1	2	1

b. Find all nurses who are on call for single room types only.

Query

```
select nurse_name from nurse
where nurse.center_id in
(select center_id from bed
where room_type = 'Single'
group by block_code);
```

Result

nurse_name
Lauren Khols
Monica Norton
Andrea Smith
Drashty Mody

4. Nested Select

a. Show visit notes from all patients whose attending nurse's name starts with the letter 'M' or 'D'.

Query

```sql
select visit_notes from assessment
where assessment.nurse_id in
(select nurse_id from nurse
where full_name LIKE 'M%' OR 'D%');
```

Result

visit_notes
pelvic pressure
right ovary pain

b. Find the names of patients who have undergone a hysterectomy.

Query

```sql
select patient_name from patient_info
where patient_info.patient_medical_record in
(select patient_medical_record from treatment
where treatment_name = 'Hysterectomy');
```

Result

patient_name
Christina Ware

c. Find the name of patients and treating physician who's new GYN appointment was scheduled for on the 17th of April at 11:30 AM.

Query

```sql
SELECT patient_name AS "Name of the patient",
       full_name AS "Name of the physician",
       a.appointment_datetime
FROM patient_info patient_name
JOIN assessment a ON a.patient_medical_record=patient_name.patient_medical_record
JOIN physician p ON a.physician_id=p.physician_id
WHERE appointment_datetime ='2021-04-17 11:30:00';
```

Result

Name of the patient	Name of the physician	appointment_datetime
Katie Hodgson	Desai	2021-04-17 11:30:00
Abeeha Malik	Kochi	2021-04-17 11:30:00

5. select involving the Union operation

a. Find the names of the nurses who work at Bristow ONGYN Associates, attending a patient scheduled for 17th of April at 11:30 AM, or both.

Query

```sql
select nurse_name from nurse where nurse.nurse_id in
(select nurse_id from center where center_name = 'Bristow OBGYN Associates')
UNION
select nurse_name from nurse where nurse.nurse_id in
(select nurse_id from assessment where appointment_datetime = '2021-04-17 11:30:00');
```

Result

nurse_name
Martha Stewart
Andrea Smith

b. Return the patient weight of the patients with a weight greater than 130 pounds.

Query

```sql
SELECT patient_name AS "Name of the patient",
       a.patient_weight
FROM patient_info patient_name
JOIN assessment a ON a.patient_medical_record=patient_name.patient_medical_record
where patient_weight > 130
UNION
SELECT patient_name AS "Name of the patient",
       a.patient_weight
FROM patient_info patient_name
JOIN assessment a ON a.patient_medical_record=patient_name.patient_medical_record
where patient_weight < 130;
```

Result

Name of the patient	patient_weight
Abeeha Malik	138.5
Christina Ware	145.4
Casandra Smith	138.4
Hadassa Rocha	155.3
Katie Hodgson	115.4
Maria Ghafoor	120.7

6. insert one tuple into a table

Before Query

treatment_id	treatment_datetime	treatment_name	result	patient_medical_record	physician_id
T-01	2021-03-15 10:30:00	Oophorectomy	G2 Right Overian Cancer	MR-14	P-12
T-02	2020-03-15 10:30:00	Hysterectomy	G1 Endometrioid Cancer	MR-31	P-24
T-05	2018-07-30 01:30:00	Laparotomy.	G3 Endometrioid Cancer	MR-41	P-15
T-10	2019-03-21 02:30:00	Myomectomy	Submucosal uterine fibriod	MR-91	P-41
T-12	2021-05-21 11:00:00	Hysteroscopy/D&C	Endometrial Polyps	MR-22	P-13

Query

```
INSERT INTO treatment (treatment_id, treatment_datetime, treatment_name, result, patient_medical_record,
VALUES ('T-18', '2017-12-24 10:30:00', 'Vaginectomy', 'Clear cell adenocarcinoma of the vagina', 'MR-86'
select * from treatment;
```

Result

treatment_id	treatment_datetime	treatment_name	result	patient_medical_record	physician_id
T-01	2021-03-15 10:30:00	Oophorectomy	G2 Right Overian Cancer	MR-14	P-12
T-02	2020-03-15 10:30:00	Hysterectomy	G1 Endometrioid Cancer	MR-31	P-24
T-05	2018-07-30 01:30:00	Laparotomy.	G3 Endometrioid Cancer	MR-41	P-15
T-10	2019-03-21 02:30:00	Myomectomy	Submucosal uterine fibriod	MR-91	P-41
T-12	2021-05-21 11:00:00	Hysteroscopy/D&C	Endometrial Polyps	MR-22	P-13
T-18	2017-12-24 10:30:00	Vaginectomy	Clear cell adenocarcinoma of the vagina	MR-86	P-18

7. insert a set of tuples (by using another select statement) insert involving two tables.

Before Query

physician_id	physician_name	subspecialty	salary
P-10	Brown	Reproductive Endocrinology	258000
P-11	Rose	Urogynecology	315034
P-12	Bicher	Gynecologic Oncology	331000
P-13	Smith	Gynecologic Oncology	328790
P-15	Anderson	Gynecologic Oncology	335246
P-17	Osbert	Urogynecology	348650
P-18	Barlow	Maternal-Fetal Medicine	505610
P-19	Gandhi	Maternal-Fetal Medicine	448789
P-20	Somoza	Reproductive Endocrinology	250436
P-24	Kochi	Gynecologic Oncology	397689
P-29	Grant	Maternal-Fetal Medicine	567843
P-41	Desai	Gynecologic Oncology	295550

Query

```sql
INSERT INTO physician(physician_id, full_name, subspecialty, salary)
SELECT center_id, full_name, subspecialty, salary
FROM nurse
WHERE center_id = 'C-225';
```

Result

physician_id	full_name	subspecialty	salary
C-225	Sarah Kinsley	Gynecologic Oncology	167923
P-10	Brown	Reproductive Endocrinology	258000
P-11	Rose	Urogynecology	315034
P-12	Bicher	Gynecologic Oncology	331000
P-13	Smith	Gynecologic Oncology	328790

8. delete one tuple or a set of tuples: from one table, from multiple tables.

a. Delete all patients with height less than 5.0

Before Query

appointment_id	appointment_datetime	visit_notes	patient_weight	patient_height	patient_bp	patient_pulse	patient_temp	patient_medical_record	nurse_id	physician_id
V-01	2021-04-17 11:30:00	abnormal vaginal bleeding	115.4	5.1	122/98	98.6	73	MR-22	N-201	P-41
V-02	2021-04-17 11:30:00	right ovary pain	138.5	5.8	135/90	97.5	65	MR-14	N-212	P-24
V-10	2018-06-15 12:43:06	pelvic pain	120.7	4.5	110/71	96.5	90	MR-41	N-202	P-13
V-12	2019-08-21 03:15:23	bleeding after menopause	145.4	5.5	128/83	98.1	84	MR-31	N-230	P-24
V-15	2020-03-08 08:45:51	vaginal pain	138.4	5.3	119/75	97.4	91	MR-86	N-009	P-18
V-31	2019-02-24 12:15:10	pelvic pressure	155.3	5.8	134/91	97.6	86	MR-91	N-150	P-12

Query

```sql
DELETE FROM assessment
WHERE patient_height < 5.0 ;
select * from assessment;
```

Result

appointment_id	appointment_datetime	visit_notes	patient_weight	patient_height	patient_bp	patient_pulse	patient_temp	patient_medical_record	nurse_id	physician_id
V-01	2021-04-17 11:30:00	abnormal vaginal bleeding	115.4	5.1	122/98	98.6	73	MR-22	N-201	P-41
V-02	2021-04-17 11:30:00	right ovary pain	138.5	5.8	135/90	97.5	65	MR-14	N-212	P-24
V-12	2019-08-21 03:15:23	bleeding after menopause	145.4	5.5	128/83	98.1	84	MR-31	N-230	P-24
V-15	2020-03-08 08:45:51	vaginal pain	138.4	5.3	119/75	97.4	91	MR-86	N-009	P-18
V-31	2019-02-24 12:15:10	pelvic pressure	155.3	5.8	134/91	97.6	86	MR-91	N-150	P-12

b. Delete appointment date for patient with medical record number MR-22 from the assessment table and delete their scheduled treatment date in the treatment table:

Before Query

treatment_id	treatment_datetime	treatment_name	result	patient_medical_record	physician_id
T-01	2021-03-15 10:30:00	Oophorectomy	G2 Right Overian Cancer	MR-14	P-12
T-02	2020-03-15 10:30:00	Hysterectomy	G1 Endometrioid Cancer	MR-31	P-24
T-05	2018-07-30 01:30:00	Laparotomy.	G3 Endometrioid Cancer	MR-41	P-15
T-10	2019-03-21 02:30:00	Myomectomy	Submucosal uterine fibriod	MR-91	P-41
T-12	2021-05-21 11:00:00	Hysteroscopy/D&C	Endometrial Polyps	MR-22	P-13
T-18	2017-12-24 10:30:00	Vaginectomy	Clear cell adenocarcinoma of the vagina	MR-86	P-18

Query

```sql
DELETE assessment,  treatment FROM assessment
INNER join treatment on assessment.patient_medical_record = treatment.patient_medical_record
where assessment.patient_medical_record = 'MR-22' ;
select * from assessment;
select * from treatment;
```

Result

treatment_id	treatment_datetime	treatment_name	result	patient_medical_record	physician_id
T-01	2021-03-15 10:30:00	Oophorectomy	G2 Right Overian Cancer	MR-14	P-12
T-02	2020-03-15 10:30:00	Hysterectomy	G1 Endometrioid Cancer	MR-31	P-24
T-05	2018-07-30 01:30:00	Laparotomy.	G3 Endometrioid Cancer	MR-41	P-15
T-10	2019-03-21 02:30:00	Myomectomy	Submucosal uterine fibriod	MR-91	P-41
T-18	2017-12-24 10:30:00	Vaginectomy	Clear cell adenocarcinoma of the vagina	MR-86	P-18

appointment_id	appointment_datetime	visit_notes	patient_weight	patient_height	patient_bp	patient_pulse	patient_temp	patient_medical_record	nurse_id	physician_id
V-02	2021-04-17 11:30:00	right ovary pain	138.5	5.8	135/90	97.5	65	MR-14	N-212	P-24
V-12	2019-08-21 03:15:23	bleeding after menopause	145.4	5.5	128/83	98.1	84	MR-31	N-230	P-24
V-15	2020-03-08 08:45:51	vaginal pain	138.4	5.3	119/75	97.4	91	MR-86	N-009	P-18
V-31	2019-02-24 12:15:10	pelvic pressure	155.3	5.8	134/91	97.6	86	MR-91	N-150	P-12

9. update one tuple or a set of tuples: from one table, from multiple tables.

a. Update each price of medication by fifty percent

Before Query

med_number	med_name	med_totalcost
M-01	Provera	24.19
M-04	Lysteda	179.55
M-07	Lupron Depot	1393.33
M-15	Progestrone	26.00
M-19	Metformin	12.88

Query

```
update medication
set medication.med_totalcost = medication.med_totalcost * 0.5;
select * from medication;
```

Result

med_number	med_name	med_totalcost
M-01	Provera	24.19
M-04	Lysteda	179.55
M-07	Lupron Depot	1393.33
M-15	Progestrone	26.00
M-19	Metformin	12.88

b. Update medication name, dose, for medication id 'M-19'.

Query

```
update medication, med_consumption
set med_consumption.dose= 'Taxol 175mg/m2 IV + carbo 5-6 mg·min/mL IV',
medication.med_name = 'Carbo/Taxol'
where medication.med_number = 'M-19'
AND med_consumption.med_number = 'M-19';
```

Result

med_number	med_name	med_totalcost
M-01	Provera	24.19
M-04	Lysteda	179.55
M-07	Lupron Depot	1393.33
M-15	Progestrone	26.00
M-19	Carbo/Taxol	12.88

consume_date	dose	med_number	patient_medical_record
2021-04-21	5mg	M-01	MR-22
2021-01-27	Taxol 175mg/m2 IV + carbo 5-6 mg·min/mL IV	M-19	MR-14
2018-06-20	11.25mg	M-07	MR-41
2020-03-15	200mg	M-15	MR-31
2019-02-26	3900mg	M-04	M-91

10. Create View: - based on one relation and more than one relation: one relation

Query

```
Create view Single as
Select * from bed
where room_type ="Single";
```

More than one relation – a view of dx names of those codes that start with the letter 'N'

```
create view DxStartWithN as
select * from dx
where dx.dx_code in
(select dx_code from physician_dx
where dx_code LIKE 'N%');
```

11. operate on View (i.e., select, insert, delete, update,..):

Query

```
Select * from Single
where available = 'false';
```

Result

bed_number	room_number	room_type	block_floor	block_code	available	center_id
04	R-201	Single	1	2	false	C-217
28	R-102	Single	1	1	false	C-281

Query

```sql
INSERT INTO Single (bed_number, room_number, room_type, block_floor, block_code, available, center_id)
VALUES (45, 'R-103', 'Single', '1', '3', 'true', 'C-201');
SELECT * from Single;
```

Result

bed_number	room_number	room_type	block_floor	block_code	available	center_id
04	R-201	Single	1	2	false	C-217
16	R-217	Single	1	2	true	C-215
28	R-102	Single	1	1	false	C-281
33	R-105	Single	1	1	true	C-232
45	R-103	Single	1	3	true	C-201

Before Query

dx_code	dx_name
N83.2	Ovarian Cyst
N89.8	Vaginal Cyst
N92.0	Menorrhagia
N95.0	Postmenopausal bleeding

Query

```sql
DELETE FROM  DxStartWithN
WHERE dx_name LIKE '%Cyst%';
SELECT * FROM DxStartWithN;
```

Result

dx_code	dx_name
N92.0	Menorrhagia
N95.0	Postmenopausal bleeding

Before Query

bed_number	room_number	room_type	block_floor	block_code	available	center_id
04	R-201	Single	1	2	false	C-217
16	R-217	Single	1	2	true	C-215
28	R-102	Single	1	1	false	C-281
33	R-105	Single	1	1	true	C-232
45	R-103	Single	1	3	true	C-201

Query

```sql
update Single
set Single.available = 'true';
SELECT * FROM Single;
```

Result

bed_number	room_number	room_type	block_floor	block_code	available	center_id
04	R-201	Single	1	2	true	C-217
16	R-217	Single	1	2	true	C-215
28	R-102	Single	1	1	true	C-281
33	R-105	Single	1	1	true	C-232
45	R-103	Single	1	3	true	C-201

B) Also, create at least 4 different practical/useful triggers (written in following tasks: MySQL) for your database to perform the

enforcing referential integrity: Delete patient assessment when patients are deleted. To trigger this, you will delete a tuple or multiple tuples, from the patient_info table.

Query

```
Delimiter $$
create trigger assessment_delete after delete on patient_info
for each row
begin
delete from assessment where patient_medical_record = old.patient_medical_record;
end;
$$


-- Activitation
delete from patient_info where patient_medical_record = 'MR-41';
select * from assessment;
select * from physician_info;
```

Result

appointment_id	appointment_datetime	visit_notes	patient_weight	patient_height	patient_bp	patient_pulse	patient_temp	patient_medical_record	nurse_id	physician_id
V-01	2021-04-17 11:30:00	abnormal vaginal bleeding	115.4	5.1	122/98	98.6	73	MR-22	N-201	P-41
V-02	2021-04-17 11:30:00	right ovary pain	138.5	5.8	135/90	97.5	65	MR-14	N-212	P-24
V-12	2019-08-21 03:15:23	bleeding after menopause	145.4	5.5	128/83	98.1	84	MR-31	N-230	P-24
V-15	2020-03-08 08:45:51	vaginal pain	138.4	5.3	119/75	97.4	91	MR-86	N-009	P-18
V-31	2019-02-24 12:15:10	pelvic pressure	155.3	5.8	134/91	97.6	86	MR-91	N-150	P-12

patient_medical_record	patient_name	bed_number	physician_id
MR-12	Het Patel	01	P-10
MR-14	Abeeha Malik	02	P-12
MR-22	Katie Hodgson	03	P-13
MR-24	Hanan Ismail	4	P-11
MR-29	Lina Freedy	10	P-18
MR-31	Christina Ware	16	P-24
MR-65	Victoria Lopez	23	P-17
MR-86	Casandra Smith	28	P-29
MR-91	Hadassa Rocha	33	P-41

enforcing attribute domain constraints: Setting a salary limit - varchar constraints. To trigger this, you will add a salary raise of greater than 5% through an update, which will ensure that there are no salary raises greater than 5% when triggered.

Query

```
Delimiter $$
Create trigger LimitSalaryRaise
after update on physician
For each row
Begin
if (new.salary - old.salary) > (0.05*old.salary) then
insert into mylog values("Physician salary is raised more than 5 percent");
end if;
end;
$$
-- Activating Trigger
update physician set salary=salary+1000 where physician_name= 'Rose' ;
select * from physician
```

Before Query

physician_id	physician_name	subspecialty	salary
?-10	Brown	Reproductive Endocrinology	258000
?-11	Rose	Urogynecology	315034
?-12	Bicher	Gynecologic Oncology	331000
?-13	Smith	Gynecologic Oncology	328790
?-15	Anderson	Gynecologic Oncology	335246
?-17	Osbert	Urogynecology	348650
?-18	Barlow	Maternal-Fetal Medicine	505610
?-19	Gandhi	Maternal-Fetal Medicine	448789
?-20	Somoza	Reproductive Endocrinology	250436
?-24	Kochi	Gynecologic Oncology	397689

Result

physician_id	physician_name	subspecialty	salary
P-10	Brown	Reproductive Endocrinology	258000
P-11	Rose	Urogynecology	318034
P-12	Bicher	Gynecologic Oncology	331000
P-13	Smith	Gynecologic Oncology	328790
P-15	Anderson	Gynecologic Oncology	335246
P-17	Osbert	Urogynecology	348650
P-18	Barlow	Maternal-Fetal Medicine	505610
P-19	Gandhi	Maternal-Fetal Medicine	448789
P-20	Somoza	Reproductive Endocrinology	250436
P-24	Kochi	Gynecologic Oncology	397689

creating database log

Query

```sql
Create table MyLog (message varchar(70));
Delimiter $$
create trigger add_physician after insert on physician
for each row
begin
insert into MyLog values(concat("physician has been added by", current_user(), "on", current_date()));
end;
$$
insert into physician values('P-51','Jackson', 'Reproductive Endocrinology', 350547);
insert into physician values('P-76', 'Trueman', 'Maternal-Fetal Medicine', 515845);
select * from physician;
select * from MyLog;
```

Result

message
physician has been added byroot@localhoston2...
physician has been added byroot@localhoston2...
physician has been added byroot@localhoston2...
physician has been added byroot@localhoston2...
physician has been added byroot@localhoston2...

physician_id	physician_name	subspecialty	salary
P-10	Brown	Reproductive Endocrinology	258000
P-11	Rose	Urogynecology	315034
P-13	Smith	Gynecologic Oncology	328790
P-15	Anderson	Gynecologic Oncology	335246
P-17	Osbert	Urogynecology	348650
P-18	Barlow	Maternal-Fetal Medicine	505610
P-19	Gandhi	Maternal-Fetal Medicine	448789
P-20	Somoza	Reproductive Endocrinology	275479.6
P-24	Kochi	Gynecologic Oncology	397689
P-29	Grant	Maternal-Fetal Medicine	567843
P-41	Desai	Gynecologic Oncology	295550
P-51	Jackson	Reproductive Endocrinology	350547
P-76	Trueman	Maternal-Fetal Medicine	515845

Query

```
Delimiter $$
create trigger update_physician after update on physician
for each row
begin
insert into MyLog values(concat('physician has been updated by ',current_user(), ' on ',current_date()));
end;
$$
update physician set salary=salary + 0.10 * salary where physician_name ='Somoza';
select * from physician;
select * from MyLog;
```

Result

Physician salary is raised more than 5 percent
physician has been updated by root@localhost ...

physician_id	physician_name	subspecialty	salary
P-10	Brown	Reproductive Endocrinology	258000
P-11	Rose	Urogynecology	315034
P-13	Smith	Gynecologic Oncology	328790
P-15	Anderson	Gynecologic Oncology	335246
P-17	Osbert	Urogynecology	348650
P-18	Barlow	Maternal-Fetal Medicine	505610
P-19	Gandhi	Maternal-Fetal Medicine	448789
P-20	Somoza	Reproductive Endocrinology	275479.6
P-24	Kochi	Gynecologic Oncology	397689
P-29	Grant	Maternal-Fetal Medicine	567843
P-41	Desai	Gynecologic Oncology	295550

Query

```
Delimiter $$
create trigger delete_physician after delete on physician
for each row
begin
insert into MyLog values(concat('physician has been deleted by ',current_user(), ' on ',current_date()));
end;
$$
delete from physician where physician_name ='Bicher';
select * from physician;
select * from MyLog;
```

Result

physician has been updated by root@localhost ...

physician_id	physician_name	subspecialty	salary
P-10	Brown	Reproductive Endocrinology	258000
P-11	Rose	Urogynecology	315034
P-13	Smith	Gynecologic Oncology	328790
P-15	Anderson	Gynecologic Oncology	335246
P-17	Osbert	Urogynecology	348650
P-18	Barlow	Maternal-Fetal Medicine	505610
P-19	Gandhi	Maternal-Fetal Medicine	448789
P-20	Somoza	Reproductive Endocrinology	250436
P-24	Kochi	Gynecologic Oncology	397689
P-29	Grant	Maternal-Fetal Medicine	567843
P-41	Desai	Gynecologic Oncology	295550

gathering statistics: make a summary table on employee salaries. To trigger this a new employee will be added, and we will receive a summary table which gives us descriptive statistics.

Query

```
create table salary_summary(subspeciality varchar(30),minsalary double, maxsalary double, avgsalary double);
Delimiter $$
create trigger salary_insert after insert on physician
for each row
begin
delete from salary_summary;
insert salary_summary
select subspeciality, min(salary),max(salary),avg(salary) from physician group by subspeciality;
end;
$$
insert into physician values ('P-09', 'Johnson', 'Urogynecology', '337834');
insert into physician values ('P-25', 'Roberts', 'Gynecologic Oncology', '382456');
select * from salary_summary;
```

Result

subspeciality	minsalary	maxsalary	avgsalary
Urogynecology	315034	348650	333839.3333333333
Reproductive Endocrinology	258000	350547	294675.5333333333
Gynecologic Oncology	295550	397689	347946.2
Maternal-Fetal Medicine	448789	567843	509521.75