HOSPITAL MANAGEMENT SYSTEM

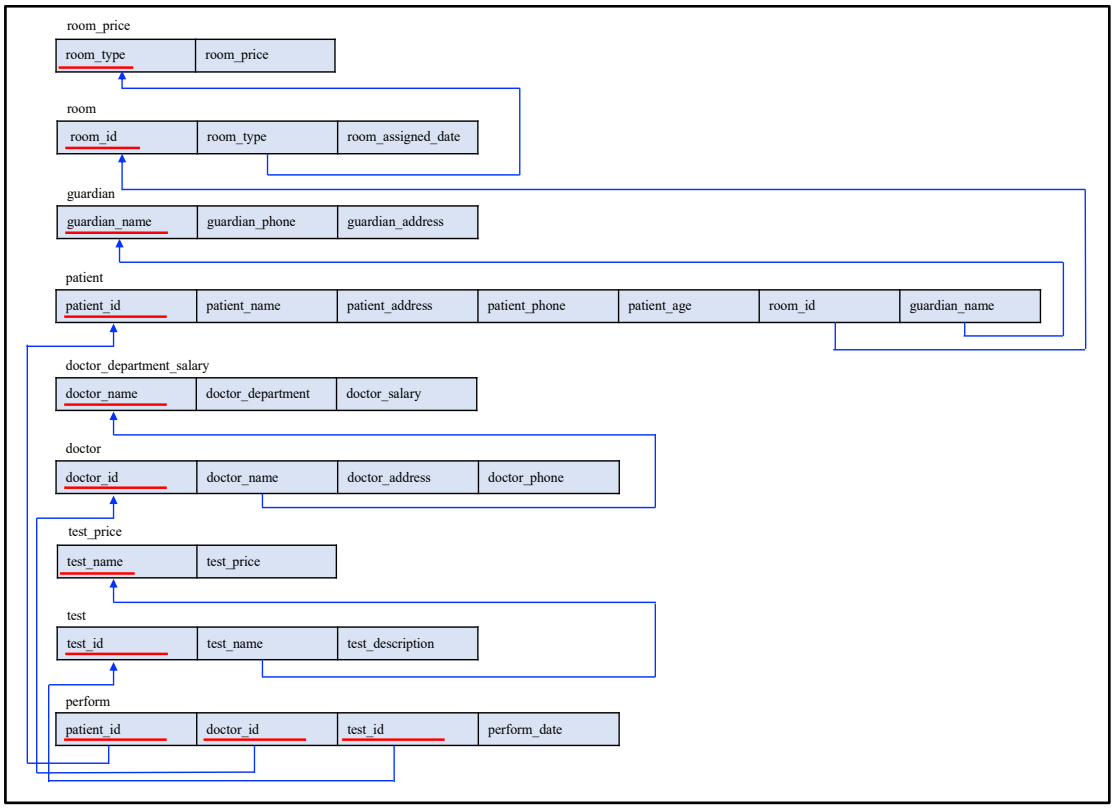
Introduction to Databases

- Semester 1, 2022/23 –

Submitted by

DONGHYEOK LEE (21234175)

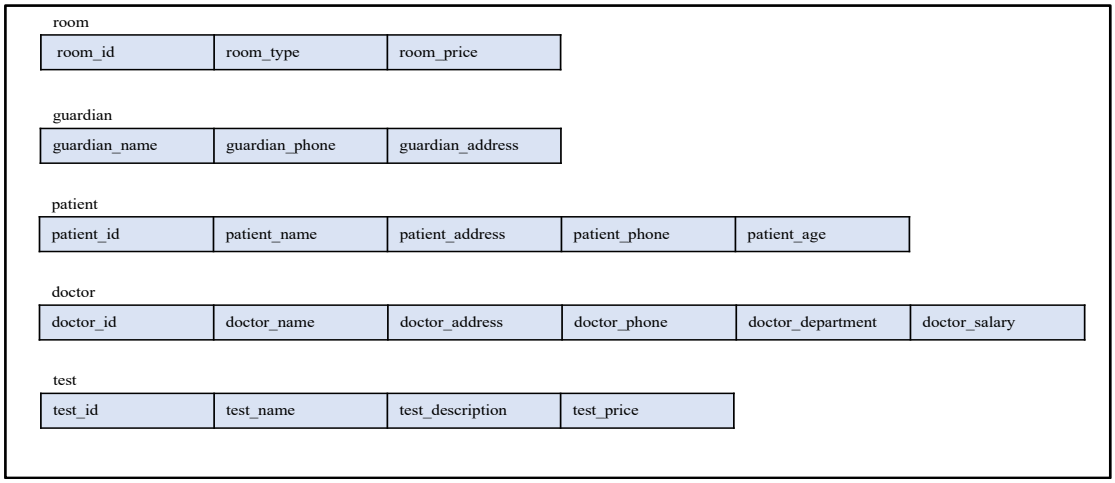
1. Relational model of hospital management system.



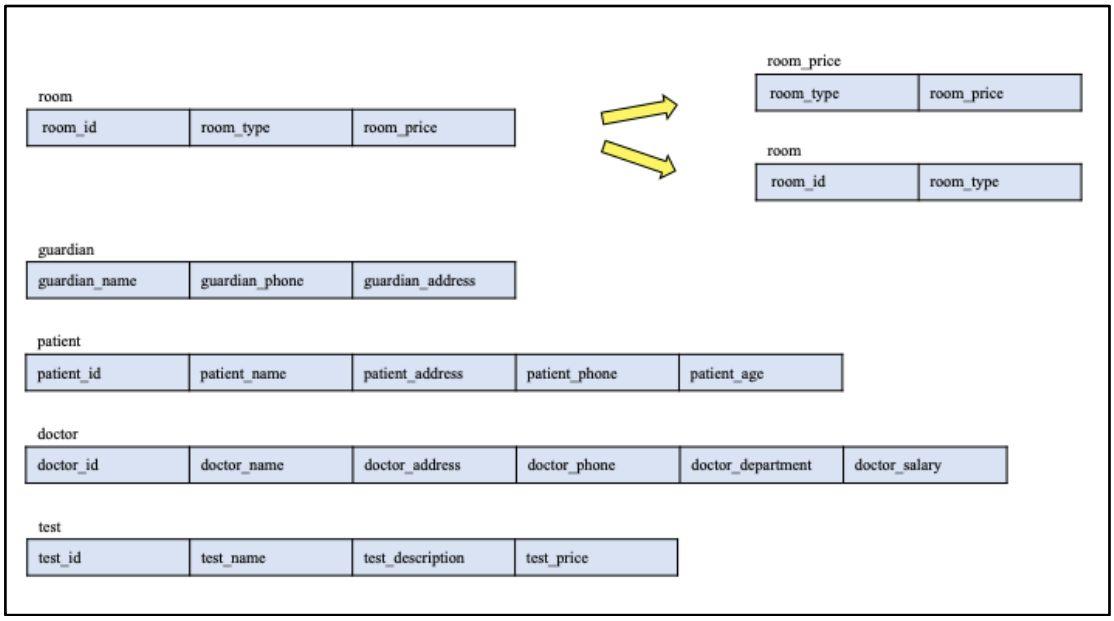
Above relational model was converted with 1st, 2nd, and 3rd normal form and below steps

[Nomarlization]

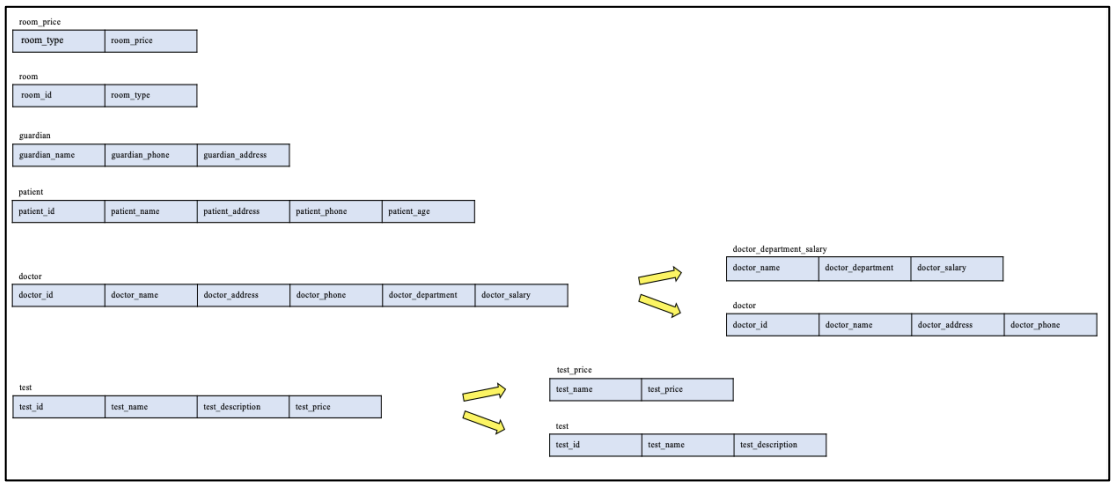
1) 1st normalization: Repeating attributes removed



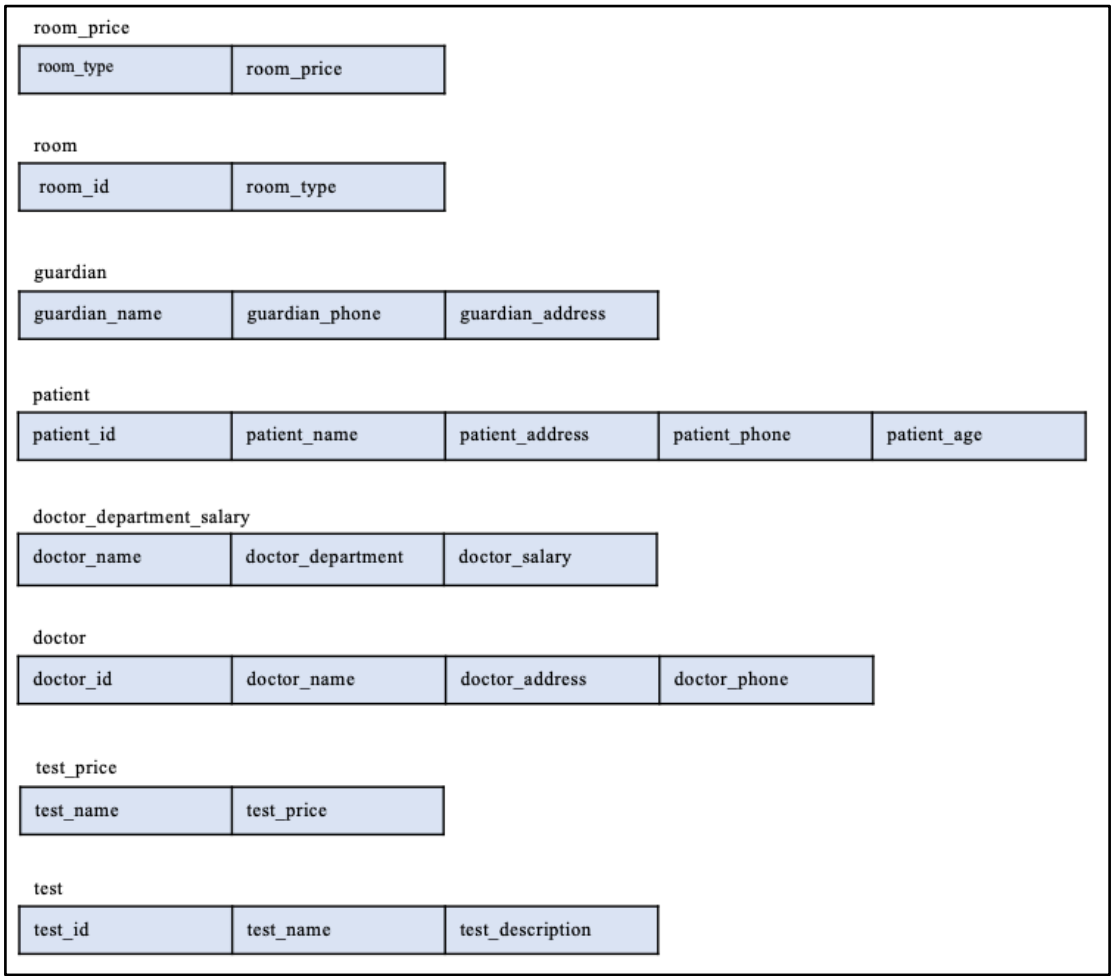
2) 2nd normalization: No partial dependencies



3) 3rd normalization: Non-key dependencies removed



4) After normalization



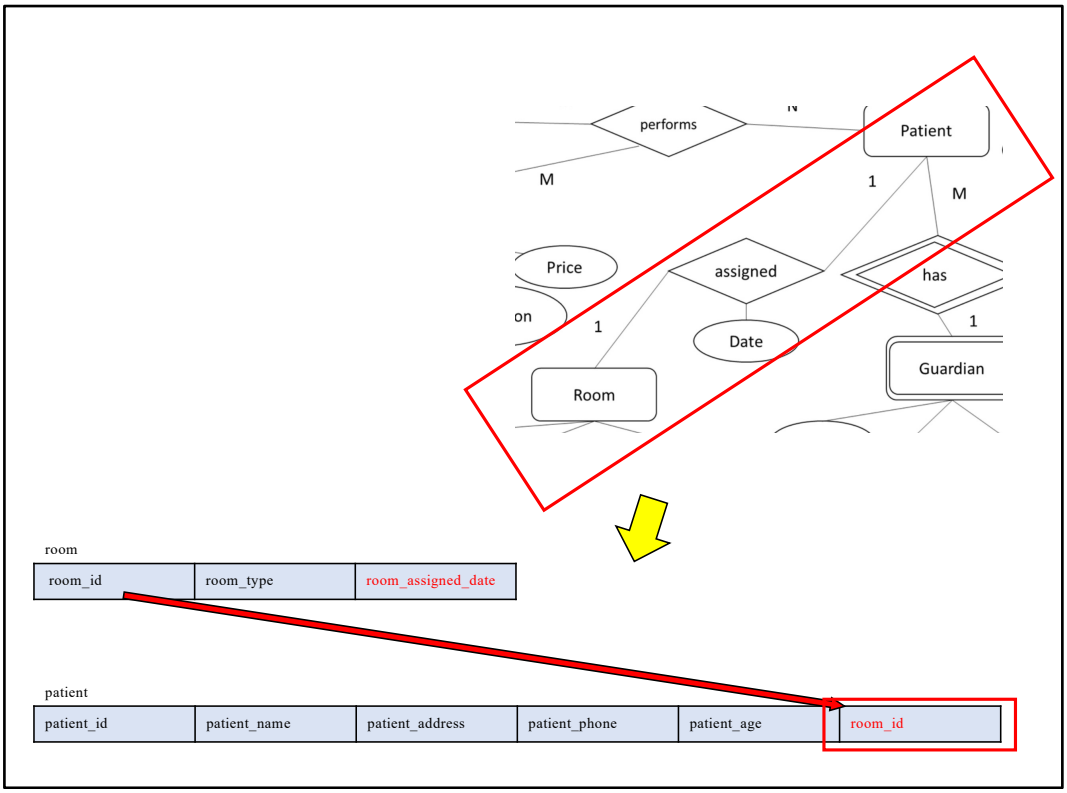
And then, above tables is processing with ER-to-Relational Mapping Algorithm (7 steps)

[ER-to-Relational Mapping Algorithm (7 steps)]

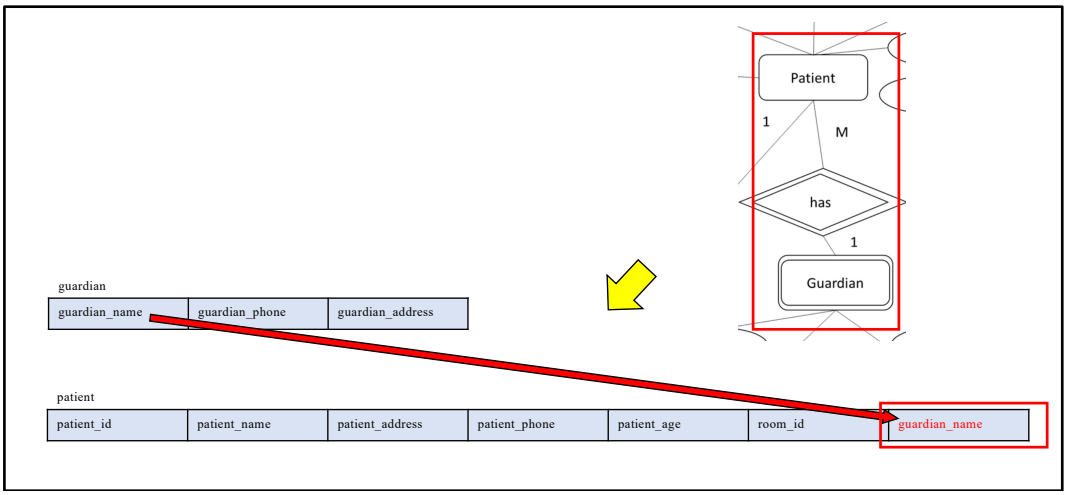
1) Step 1: Mapping of Regular Entity Types.
   - The same with above table.

2) Step 2: Mapping of Weak Entity Types.
   - Guardian is a weak entity type. So, use Guardian_name as a foreign key of patient table
   - Guardian_name is used for composite key of patient table.

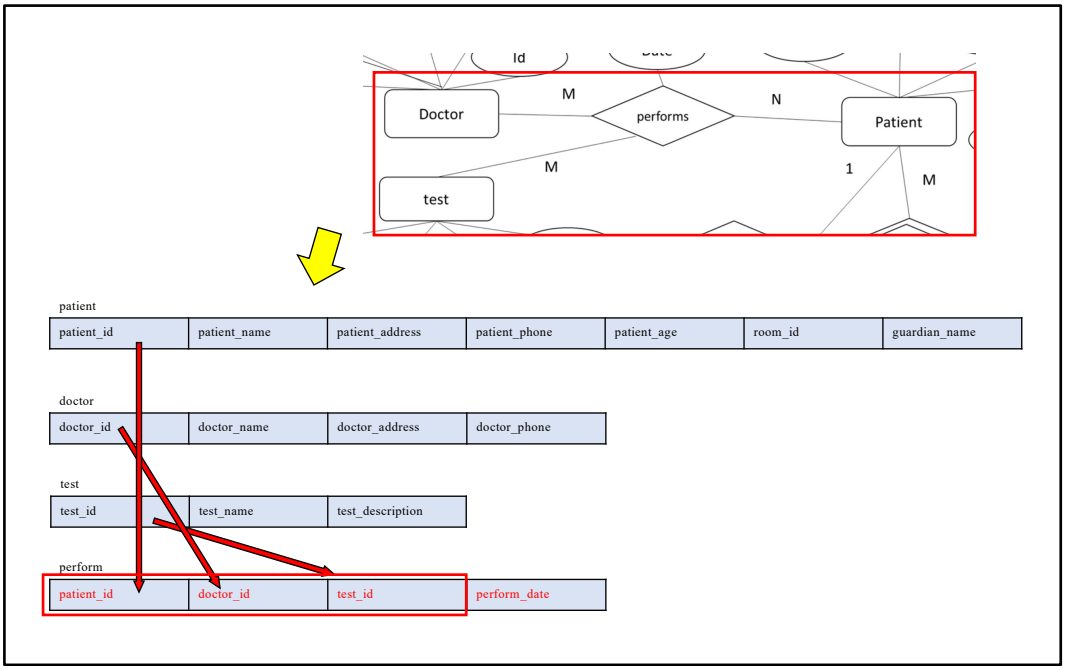
3) Step 3: Mapping of Binary 1:1 Relationship Types.
   - Room – Patient.



4) Step 4: Mapping of Binary 1:N Relationship Types.
   - Guardian – Patient.

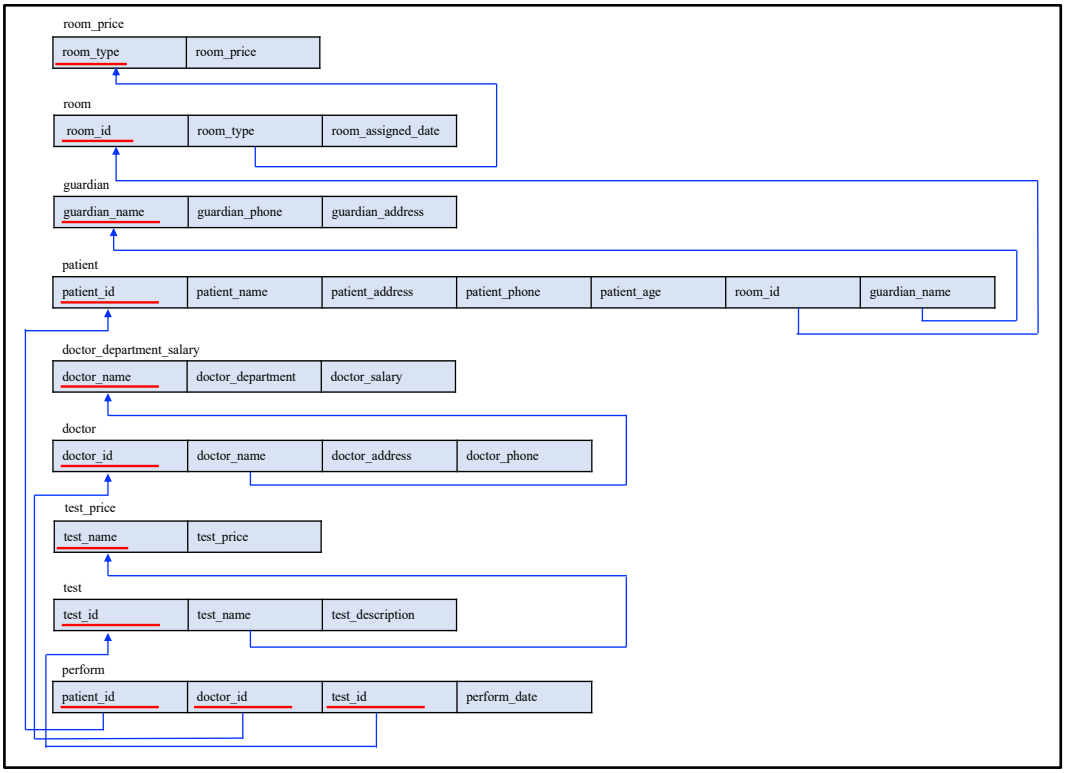


5) Step 5: Mapping of Binary M:N Relationship Types.
   - Patient – Doctor – Test.

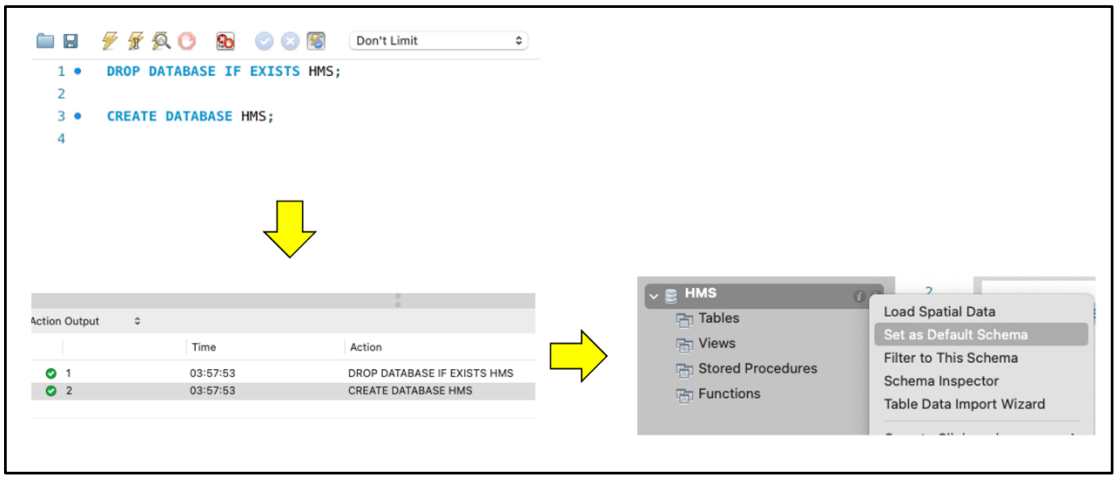


6) Step 6: Mapping of Multivalued Attributes .
   - There is no multivalued attribute in this database.

7) Step 7: Mapping of N-ary Relationship Types.



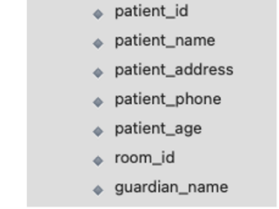
2.1. Create ‘HMS’ database using DDL.



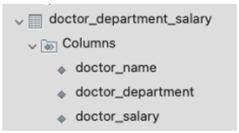
2.2. Create all tables of ‘HMS’ database using DDL.

```sql
DROP TABLE IF EXISTS guardian;
create table guardian (
    guardian_name VARCHAR(50) not null,
    guardian_phone VARCHAR(50),
    guardian_address VARCHAR(50),
    PRIMARY KEY (guardian_name)
);
```

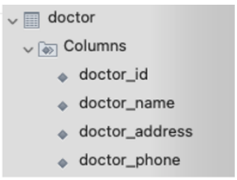

```sql
DROP TABLE IF EXISTS patient;
create table patient (
    patient_id INT not null,
    patient_name VARCHAR(50),
    patient_address VARCHAR(50),
    patient_phone VARCHAR(50),
    patient_age INT,
    room_id INT,
    guardian_name VARCHAR(50),
    PRIMARY KEY (patient_id)
);
Alter table patient Add foreign key (room_id) references room (room_id);
Alter table patient Add foreign key (guardian_name) references guardian (guardian_name);
```




```sql
DROP TABLE IF EXISTS doctor_department_salary;
create table doctor_department_salary (
    doctor_name VARCHAR(50) not null,
    doctor_department VARCHAR(23),
    doctor_salary INT,
    PRIMARY KEY (doctor_name)
);
```

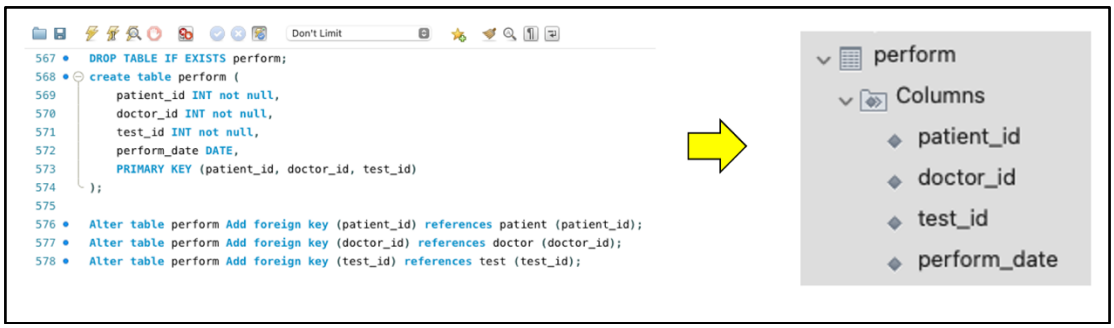



```sql
DROP TABLE IF EXISTS doctor;
create table doctor (
    doctor_id INT not null,
    doctor_name VARCHAR(50),
    doctor_address VARCHAR(50),
    doctor_phone VARCHAR(50),
    PRIMARY KEY (doctor_id)
);
Alter table doctor Add foreign key (doctor_name) references doctor_department_salary (doctor_name);
```



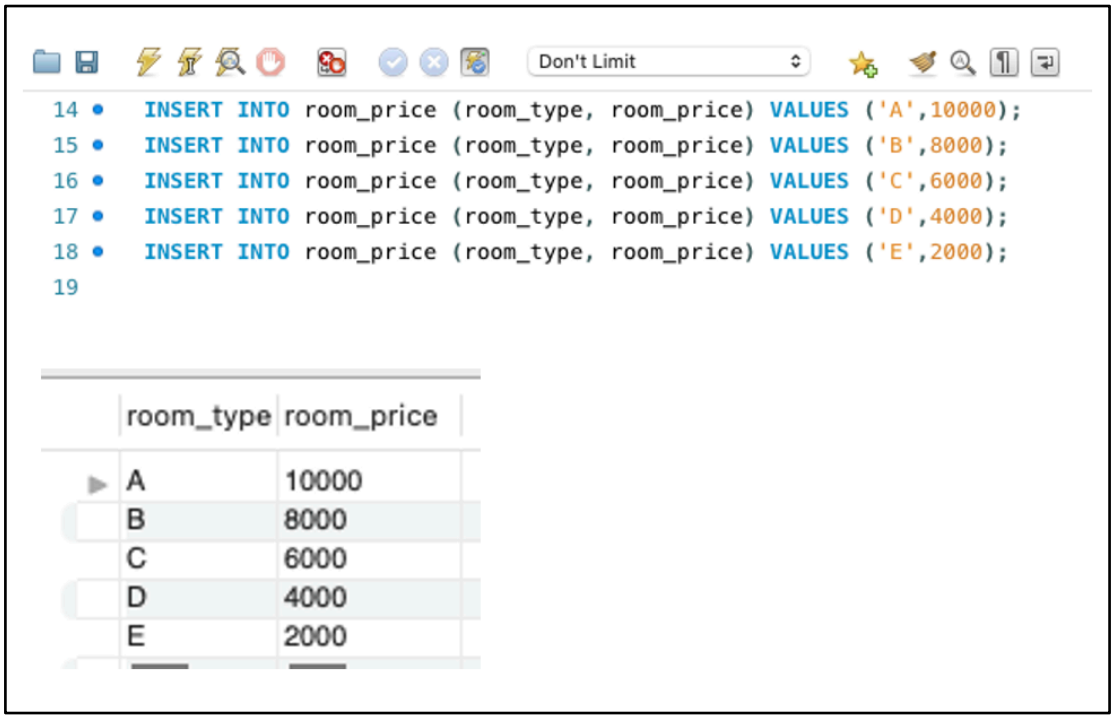

```sql
DROP TABLE IF EXISTS test_price;
create table test_price (
    test_name VARCHAR(50) not null,
    test_price INT,
    PRIMARY KEY (test_name)
);
```


```sql
DROP TABLE IF EXISTS test;
create table test (
    test_id INT not null,
    test_name VARCHAR(50),
    test_description  VARCHAR(50),
    PRIMARY KEY (test_id)
);
Alter table test Add foreign key (test_name) references test_price (test_name);
```

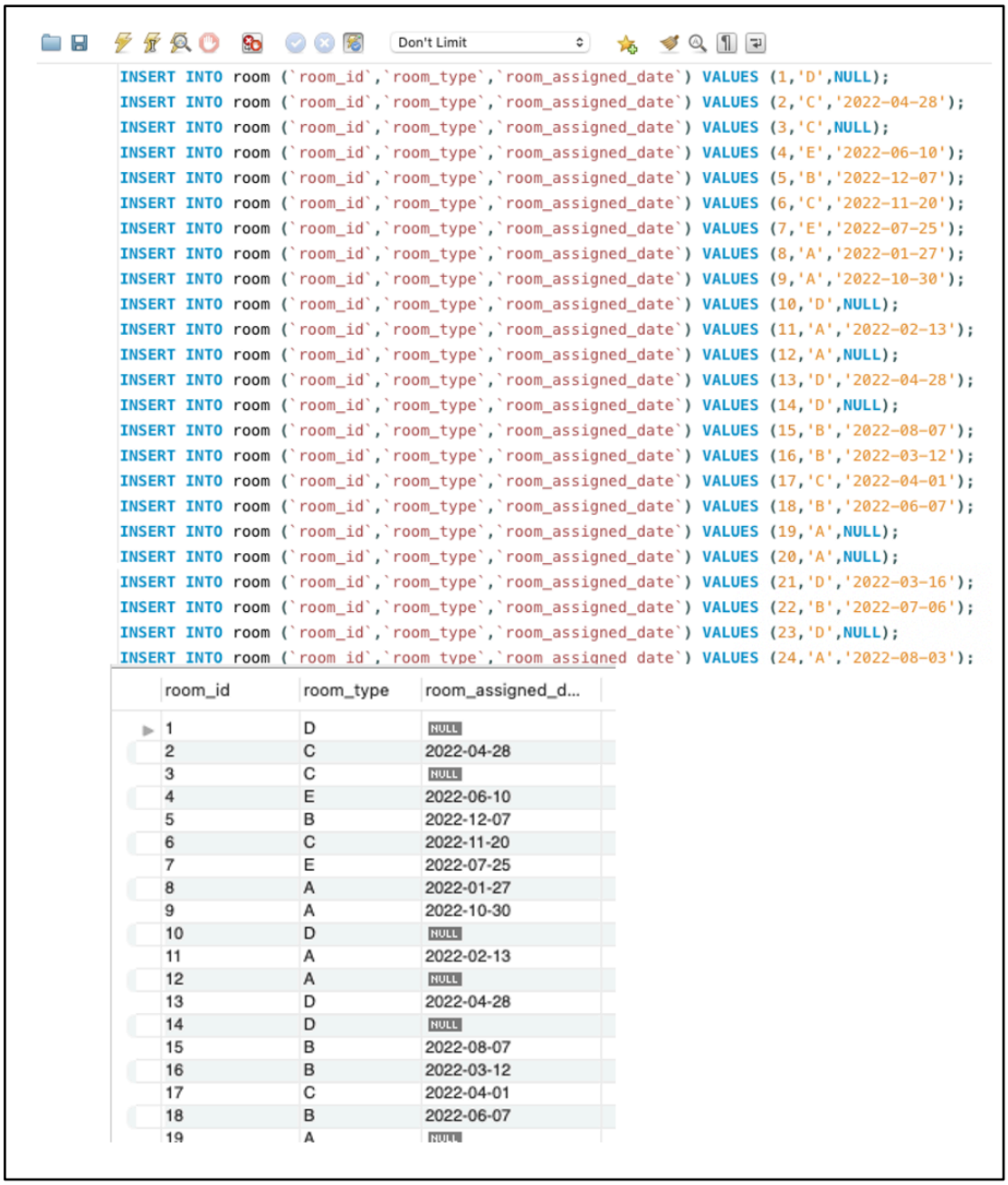


2.3. Fill tables with some data using Mockaroo.

1) I assume that there are 5 room type and according to these types, room price will be defined. It has information of Room type (Primary Key) and Room price.



2) I assume that there are 150 rooms and among these rooms, some are available, and the others are unavailable because they're already assigned. It has information of Room id (Primary Key) and Room type (Foreign key), and Room assigned date.



```
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (1,'D',NULL);
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (2,'C','2022-04-28');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (3,'C',NULL);
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (4,'E','2022-06-10');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (5,'B','2022-12-07');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (6,'C','2022-11-20');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (7,'E','2022-07-25');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (8,'A','2022-01-27');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (9,'A','2022-10-30');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (10,'D',NULL);
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (11,'A','2022-02-13');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (12,'A',NULL);
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (13,'D','2022-04-28');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (14,'D',NULL);
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (15,'B','2022-08-07');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (16,'B','2022-03-12');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (17,'C','2022-04-01');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (18,'B','2022-06-07');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (19,'A',NULL);
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (20,'A',NULL);
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (21,'D','2022-03-16');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (22,'B','2022-07-06');
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (23,'D',NULL);
INSERT INTO room (`room_id`,`room_type`,`room_assigned_date`) VALUES (24,'A','2022-08-03');
```

room_id	room_type	room_assigned_d...
1	D	NULL
2	C	2022-04-28
3	C	NULL
4	E	2022-06-10
5	B	2022-12-07
6	C	2022-11-20
7	E	2022-07-25
8	A	2022-01-27
9	A	2022-10-30
10	D	NULL
11	A	2022-02-13
12	A	NULL
13	D	2022-04-28
14	D	NULL
15	B	2022-08-07
16	B	2022-03-12
17	C	2022-04-01
18	B	2022-06-07
19	A	NULL

3) I assume that there are 50 guardians. They are providing service for patience(s) (1 – many). They have information of Guardian name (Primary Key) and Guardian phone, and Guardian address.

```
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Maura Muddle', '513-996-7560', '53 Raven Alley');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Cyndy Houseman', '281-444-5948', '75 Stang Avenue');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Keir Graver', '784-267-4125', '73 Veith Street');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Eddie Annable', '826-703-9472', '2927 Almo Alley');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Walden Seamon', '985-387-1969', '50 Hoffman Court');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Nicolai Bartoszewski', '205-822-0792', '4 Westerfield Crossing');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Bell Hollyar', '219-616-9606', '771 Iowa Trail');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Jean Brennans', '426-861-6010', '0 Badeau Park');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Ignacius Flintoffe', '582-475-0466', '355 Jenna Court');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Rachel Tousey', '483-729-5190', '1 Raven Center');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Tann Augie', '794-434-7303', '4709 Menomonie Place');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Stefa Lathey', '326-803-4070', '797 Duke Junction');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Crin Aguilar', '180-624-6632', '932 Sherman Avenue');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Raine McMurthy', '876-632-7657', '69 North Crossing');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Janot Suttill', '915-621-7040', '39545 Park Meadow Drive');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Hill Kimmings', '871-744-1508', '0934 Oriole Street');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Agnella Puleque', '764-248-1954', '598 Killdeer Drive');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Tybi Burrow', '979-371-9695', '141 Mccormick Road');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Milicent Skuce', '472-144-4587', '146 Charing Cross Plaza');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Arlene Daborne', '219-224-1381', '31755 Graedel Place');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Krysta Brass', '719-753-1487', '4 Emmet Lane');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Jenine Prettejohns', '439-648-0053', '88521 8th Road');
insert into guardian (guardian_name, guardian_phone, guardian_address) values ('Bibby Hesbrook', '992-659-3158', '233 Jenna Circle');
```

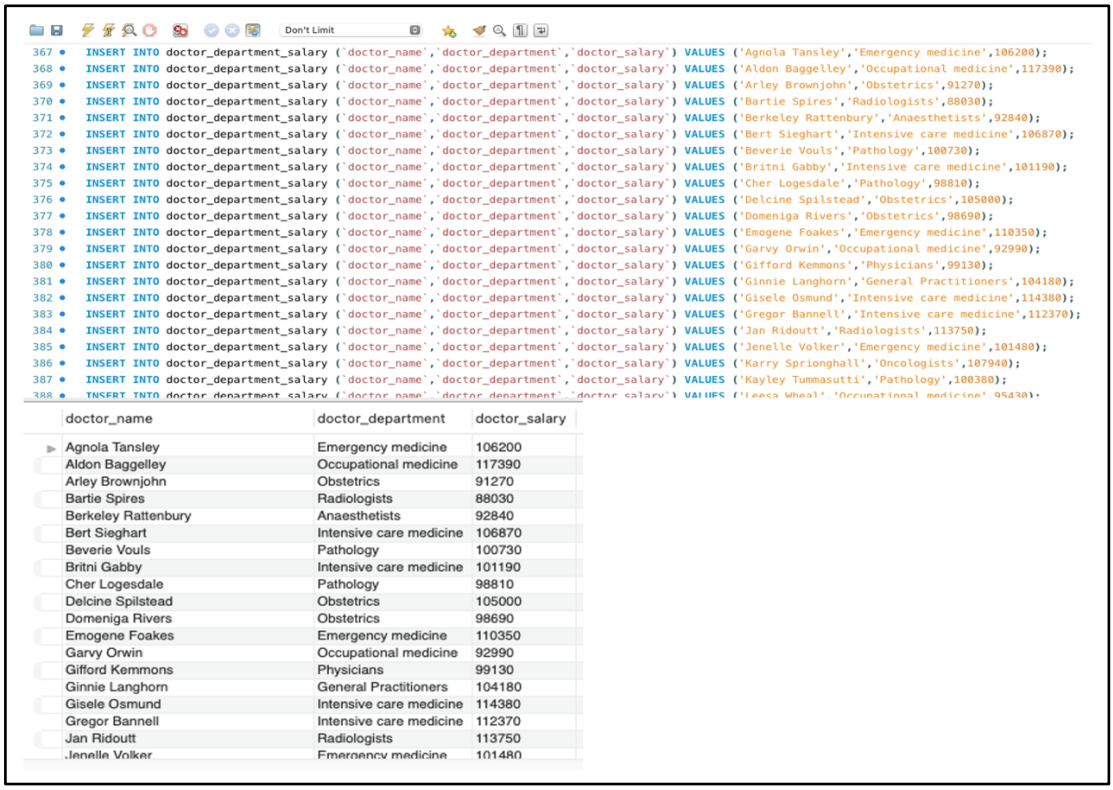
guardian_name	guardian_phone	guardian_address
Adrian Carrick	863-811-1148	008 Miller Lane
Agnella Puleque	764-248-1954	598 Killdeer Drive
Aguie Popplewell	273-871-0215	679 Gulseth Way
Arlee Rubertelli	541-323-0979	74488 Veith Way
Arlene Daborne	219-224-1381	31755 Graedel Place
Bell Hollyar	219-616-9606	771 Iowa Trail
Bibby Hesbrook	992-659-3158	233 Jenna Circle
Bob Layson	946-156-9108	13931 Longview Place
Brooks Walkling	818-680-4329	87 Glacier Hill Avenue
Carey Patillo	829-127-0380	07216 Donald Circle
Crin Aguilar	180-624-6632	932 Sherman Avenue
Cyndy Houseman	281-444-5948	75 Stang Avenue
Dana Rudolfer	120-115-7676	8769 Troy Circle
Eddie Annable	826-703-9472	2927 Almo Alley
Elizabet Mussetti	567-984-1068	0692 Melody Court
Esta Massy	957-452-8333	44 Monica Road
Gerome Bendixen	364-791-1583	5 Waywood Crossing
Herbie Kerwin	466-551-5315	18 Oxford Drive

4) I assume that there are 100 patients. They have some information of Patient id, Patient name, Patient address, Patient phone, Patient age, Room id (Foreign Key), and Guardian name (Foreign Key))

```
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1001,'Sigismund Luppitt','6502 Rowland Circle','963-277-8367',
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1002,'Jacynth Raubenheimer','1 Pleasure Avenue','394-616-7509'
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1003,'Dominic Shutte','9 Columbus Trail','912-371-2412',31,'3'
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1004,'Kerry Caw','516 Carberry Lane','137-628-5869',26,'4','Ed
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1005,'Zea Marchington','29 Lakewood Gardens Hill','764-440-297
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1006,'Giuditta McLenaghan','34 Charing Cross Terrace','202-476
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1007,'Gladi Averill','24405 Rigney Pass','596-848-0516',36,'7'
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1008,'Jeannette Tuft','0 Grasskamp Hill','865-284-5483',36,'8'
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1009,'Idalina Calken','12493 Lindbergh Trail','357-574-5759',3
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1010,'Flossi Jacob','45286 Heffernan Drive','668-727-5640',28,
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1011,'Isidor Wills','8 Bowman Center','982-510-4361',15,'11','
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1012,'Guillema Danzelman','01562 Rusk Lane','622-561-0714',29,
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1013,'Claus Gumley','70 Lakewood Avenue','893-340-7088',10,'13
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1014,'Geneva Dillimore','6777 Golf Alley','733-801-5087',40,'1
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1015,'Katheryn Sabathier','48634 Mcguire Lane','656-965-8398',
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1016,'Winni Canto','5 Lakewood Gardens Place','120-285-3161',4
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1017,'Schuyler Daymont','3 Fieldstone Crossing','698-678-6960'
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1018,'Brice Rastall','0100 Kim Park','589-797-4510',21,'18','T
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1019,'Arthur Windybank','908 Carey Circle','967-358-5098',24,'
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1020,'Terrie Winthrop','32948 Hollow Ridge Place','923-572-602
INSERT INTO patient (`patient_id`,`patient_name`,`patient_address`,`patient_phone`,`patient_age`,`room_id`,`guardian_name`) VALUES (1021,'Cori McSperrin','50 Trailsway Parkway','368-396-7827',32
```

patient_id	patient_name	patient_address	patient_pho...	patient_age	room_id	guardian_name
1001	Sigismund Luppitt	6502 Rowland Circle	963-277-8367	36	1	NULL
1002	Jacynth Raubenheimer	1 Pleasure Avenue	394-616-7509	47	2	Cyndy Houseman
1003	Dominic Shutte	9 Columbus Trail	912-371-2412	31	3	Keir Graver
1004	Kerry Caw	516 Carberry Lane	137-628-5869	26	4	Eddie Annable
1005	Zea Marchington	29 Lakewood Gardens Hill	764-440-2976	14	5	Walden Seamon
1006	Giuditta McLenaghan	34 Charing Cross Terrace	202-476-9429	10	6	Nicolai Bartoszewski
1007	Gladi Averill	24405 Rigney Pass	596-848-0516	36	7	Bell Hollyar
1008	Jeannette Tuft	0 Grasskamp Hill	865-284-5483	36	8	NULL
1009	Idalina Calken	12493 Lindbergh Trail	357-574-5759	32	9	Ignacius Flintoffe
1010	Flossi Jacob	45286 Heffernan Drive	668-727-5640	28	10	Rachel Tousey
1011	Isidor Wills	8 Bowman Center	982-510-4361	15	11	Tann Augie
1012	Guillema Danzelman	01562 Rusk Lane	622-561-0714	29	12	Stefa Lathey
1013	Claus Gumley	70 Lakewood Avenue	893-340-7088	10	13	Crin Aguilar
1014	Geneva Dillimore	6777 Golf Alley	733-801-5087	40	14	Raine McMurthy
1015	Katheryn Sabathier	48634 Mcguire Lane	656-965-8398	24	15	Janot Suttill
1016	Winni Canto	5 Lakewood Gardens Pla...	120-285-3161	44	16	NULL
1017	Schuyler Daymont	3 Fieldstone Crossing	698-678-6960	32	17	Agnella Puleque
1018	Brice Rastall	0100 Kim Park	589-797-4510	21	18	Tybi Burrow
1019	Arthur Windybank	908 Carey Circle	967-358-5098	24	19	Milicent Skuce

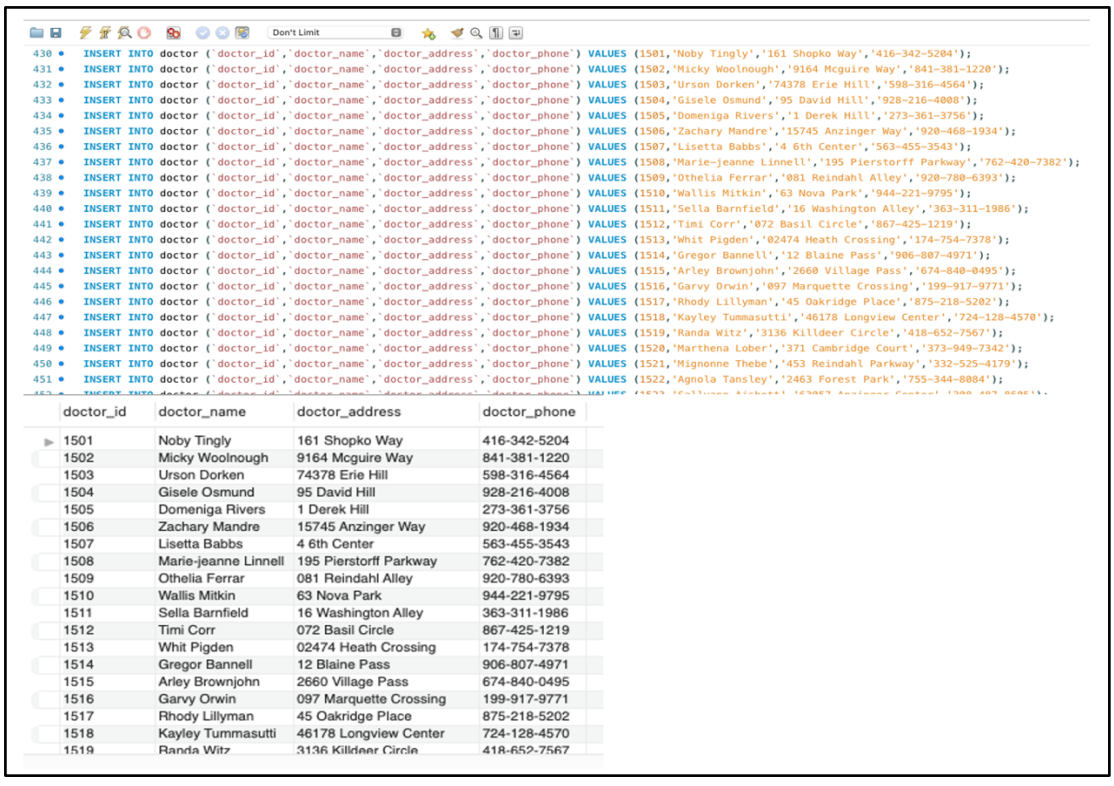
5) I assume that there are 50 doctors. They have some information of Doctor name (Primary Key), Doctor department, and Doctor salary.



```
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Agnola Tansley','Emergency medicine',106200);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Aldon Baggelley','Occupational medicine',117390);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Arley Brownjohn','Obstetrics',91270);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Bartie Spires','Radiologists',88030);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Berkeley Rattenbury','Anaesthetists',92840);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Bert Sieghart','Intensive care medicine',106870);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Beverie Vouls','Pathology',100730);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Britni Gabby','Intensive care medicine',101190);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Cher Logesdale','Pathology',98810);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Delcine Spilstead','Obstetrics',105000);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Domeniga Rivers','Obstetrics',98690);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Emogene Foakes','Emergency medicine',110350);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Garvy Orwin','Occupational medicine',92990);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Gifford Kemmons','Physicians',99130);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Ginnie Langhorn','General Practitioners',104180);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Gisele Osmund','Intensive care medicine',114380);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Gregor Bannell','Intensive care medicine',112370);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Jan Ridoutt','Radiologists',113750);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Jenelle Volker','Emergency medicine',101480);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Karry Sprionghall','Oncologists',107940);
INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Kayley Tummasutti','Pathology',100380);
```

doctor_name	doctor_department	doctor_salary
Agnola Tansley	Emergency medicine	106200
Aldon Baggelley	Occupational medicine	117390
Arley Brownjohn	Obstetrics	91270
Bartie Spires	Radiologists	88030
Berkeley Rattenbury	Anaesthetists	92840
Bert Sieghart	Intensive care medicine	106870
Beverie Vouls	Pathology	100730
Britni Gabby	Intensive care medicine	101190
Cher Logesdale	Pathology	98810
Delcine Spilstead	Obstetrics	105000
Domeniga Rivers	Obstetrics	98690
Emogene Foakes	Emergency medicine	110350
Garvy Orwin	Occupational medicine	92990
Gifford Kemmons	Physicians	99130
Ginnie Langhorn	General Practitioners	104180
Gisele Osmund	Intensive care medicine	114380
Gregor Bannell	Intensive care medicine	112370
Jan Ridoutt	Radiologists	113750
Jenelle Volker	Emergency medicine	101480

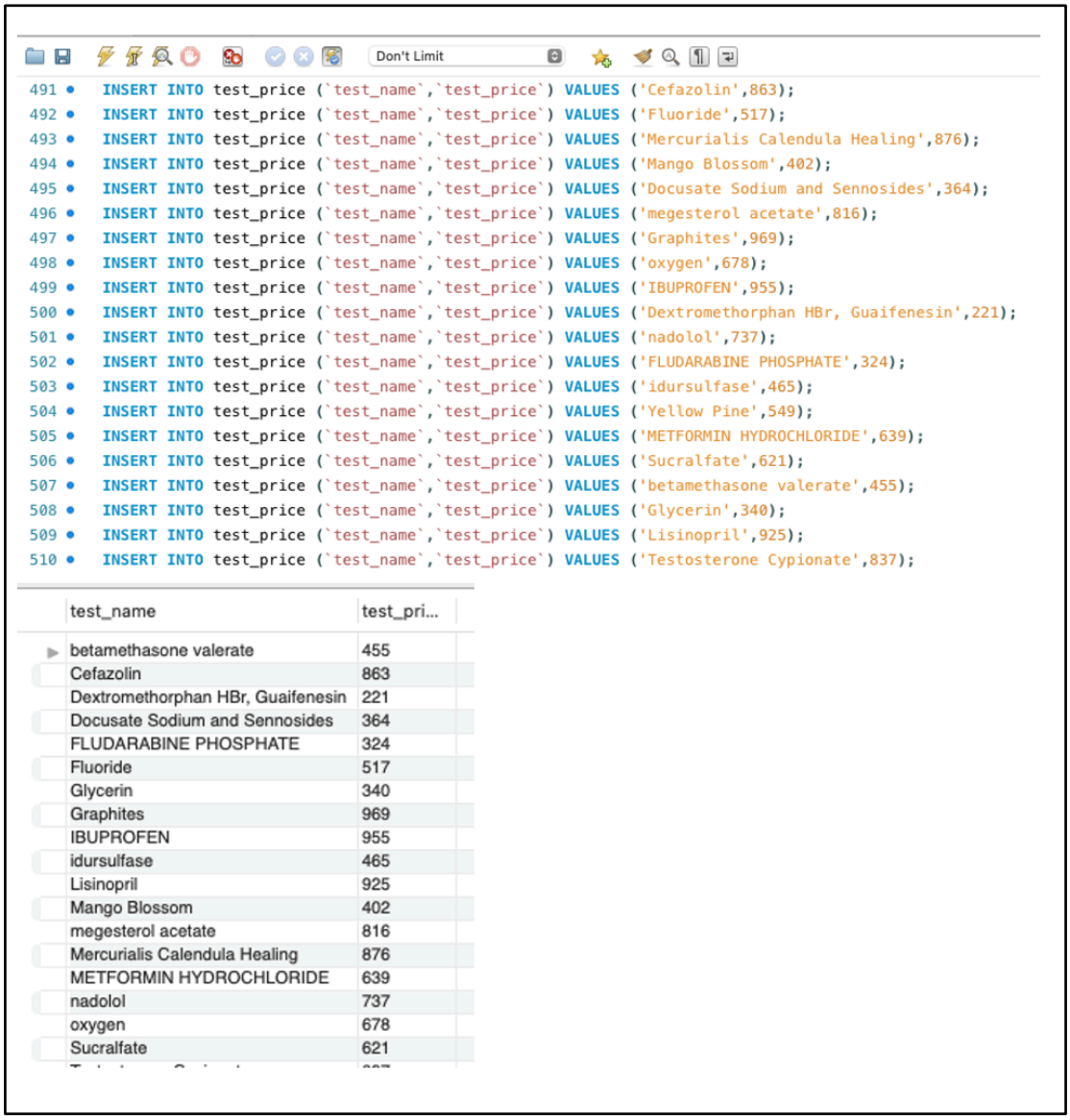
6) I assume that there are 50 doctors. With above the table, this table have some information related to doctor of Doctor id (Primary Key), Doctor name (Foreign Key), Doctor address, and Doctor phone.



```
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1501,'Noby Tingly','161 Shopko Way','416-342-5204');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1502,'Micky Woolnough','9164 Mcguire Way','841-381-1220');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1503,'Urson Dorken','74378 Erie Hill','598-316-4564');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1504,'Gisele Osmund','95 David Hill','928-216-4008');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1505,'Domeniga Rivers','1 Derek Hill','273-361-3756');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1506,'Zachary Mandre','15745 Anzinger Way','920-468-1934');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1507,'Lisetta Babbs','4 6th Center','563-455-3543');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1508,'Marie-jeanne Linnell','195 Pierstorff Parkway','762-420-7382');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1509,'Othelia Ferrar','081 Reindahl Alley','920-780-6393');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1510,'Wallis Mitkin','63 Nova Park','944-221-9795');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1511,'Sella Barnfield','16 Washington Alley','363-311-1986');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1512,'Timi Corr','072 Basil Circle','867-425-1219');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1513,'Whit Pigden','02474 Heath Crossing','174-754-7378');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1514,'Gregor Bannell','12 Blaine Pass','906-807-4971');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1515,'Arley Brownjohn','2660 Village Pass','674-840-0495');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1516,'Garvy Orwin','097 Marquette Crossing','199-917-9771');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1517,'Rhody Lillyman','45 Oakridge Place','875-218-5202');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1518,'Kayley Tummasutti','46178 Longview Center','724-128-4570');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1519,'Randa Witz','3136 Killdeer Circle','418-652-7567');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1520,'Marthena Lober','371 Cambridge Court','373-949-7342');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1521,'Mignonne Thebe','453 Reindahl Parkway','332-525-4179');
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1522,'Agnola Tansley','2463 Forest Park','755-344-8084');
```

doctor_id	doctor_name	doctor_address	doctor_phone
1501	Noby Tingly	161 Shopko Way	416-342-5204
1502	Micky Woolnough	9164 Mcguire Way	841-381-1220
1503	Urson Dorken	74378 Erie Hill	598-316-4564
1504	Gisele Osmund	95 David Hill	928-216-4008
1505	Domeniga Rivers	1 Derek Hill	273-361-3756
1506	Zachary Mandre	15745 Anzinger Way	920-468-1934
1507	Lisetta Babbs	4 6th Center	563-455-3543
1508	Marie-jeanne Linnell	195 Pierstorff Parkway	762-420-7382
1509	Othelia Ferrar	081 Reindahl Alley	920-780-6393
1510	Wallis Mitkin	63 Nova Park	944-221-9795
1511	Sella Barnfield	16 Washington Alley	363-311-1986
1512	Timi Corr	072 Basil Circle	867-425-1219
1513	Whit Pigden	02474 Heath Crossing	174-754-7378
1514	Gregor Bannell	12 Blaine Pass	906-807-4971
1515	Arley Brownjohn	2660 Village Pass	674-840-0495
1516	Garvy Orwin	097 Marquette Crossing	199-917-9771
1517	Rhody Lillyman	45 Oakridge Place	875-218-5202
1518	Kayley Tummasutti	46178 Longview Center	724-128-4570
1519	Randa Witz	3136 Killdeer Circle	418-652-7567

7) I assume that there are 20 tests. It has some information of Test name (Primary Key), and Test price.



```
491 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('Cefazolin',863);
492 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('Fluoride',517);
493 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('Mercurialis Calendula Healing',876);
494 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('Mango Blossom',402);
495 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('Docusate Sodium and Sennosides',364);
496 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('megesterol acetate',816);
497 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('Graphites',969);
498 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('oxygen',678);
499 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('IBUPROFEN',955);
500 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('Dextromethorphan HBr, Guaifenesin',221);
501 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('nadolol',737);
502 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('FLUDARABINE PHOSPHATE',324);
503 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('idursulfase',465);
504 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('Yellow Pine',549);
505 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('METFORMIN HYDROCHLORIDE',639);
506 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('Sucralfate',621);
507 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('betamethasone valerate',455);
508 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('Glycerin',340);
509 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('Lisinopril',925);
510 • INSERT INTO test_price (`test_name`,`test_price`) VALUES ('Testosterone Cypionate',837);
```

test_name	test_pri...
betamethasone valerate	455
Cefazolin	863
Dextromethorphan HBr, Guaifenesin	221
Docusate Sodium and Sennosides	364
FLUDARABINE PHOSPHATE	324
Fluoride	517
Glycerin	340
Graphites	969
IBUPROFEN	955
idursulfase	465
Lisinopril	925
Mango Blossom	402
megesterol acetate	816
Mercurialis Calendula Healing	876
METFORMIN HYDROCHLORIDE	639
nadolol	737
oxygen	678
Sucralfate	621

8) I assume that there are 40 Test ids. It has some information of Test id (Primary Key), Test name (Foreign Key) and Test description.

```
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2001,'Cefazolin','CA-NT');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2002,'Fluoride','US-AK');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2003,'Mercurialis Calendula Healing','CD-NK');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2004,'Mango Blossom','NL-NH');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2005,'Docusate Sodium and Sennosides','US-AK');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2006,'megesterol acetate','DE-BY');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2007,'Graphites','RU-VLG');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2008,'oxygen','US-FL');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2009,'IBUPROFEN','VE-G');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2010,'Dextromethorphan HBr, Guaifenesin','ET-SN');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2011,'nadolol','US-WI');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2012,'FLUDARABINE PHOSPHATE','EG-SIN');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2013,'idursulfase','RU-AST');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2014,'Yellow Pine','SD-25');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2015,'METFORMIN HYDROCHLORIDE','US-MI');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2016,'Sucralfate','US-NM');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2017,'betamethasone valerate','AU-WA');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2018,'Glycerin','MV-01');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2019,'Lisinopril','AU-QLD');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2020,'Testosterone Cypionate','CN-53');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2021,'Cefazolin','MZ-I');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2022,'Fluoride','JP-28');
INSERT INTO test (`test_id`,`test_name`,`test_description`) VALUES (2023,'Mercurialis Calendula Healing','AU-QLD');
```

test_id	test_name	test_description
2001	Cefazolin	CA-NT
2002	Fluoride	US-AK
2003	Mercurialis Calendula Healing	CD-NK
2004	Mango Blossom	NL-NH
2005	Docusate Sodium and Sennosides	US-AK
2006	megesterol acetate	DE-BY
2007	Graphites	RU-VLG
2008	oxygen	US-FL
2009	IBUPROFEN	VE-G
2010	Dextromethorphan HBr, Guaifenesin	ET-SN
2011	nadolol	US-WI
2012	FLUDARABINE PHOSPHATE	EG-SIN
2013	idursulfase	RU-AST
2014	Yellow Pine	SD-25
2015	METFORMIN HYDROCHLORIDE	US-MI
2016	Sucralfate	US-NM
2017	betamethasone valerate	AU-WA
2018	Glycerin	MV-01
2019	Lisinopril	AU-QLD

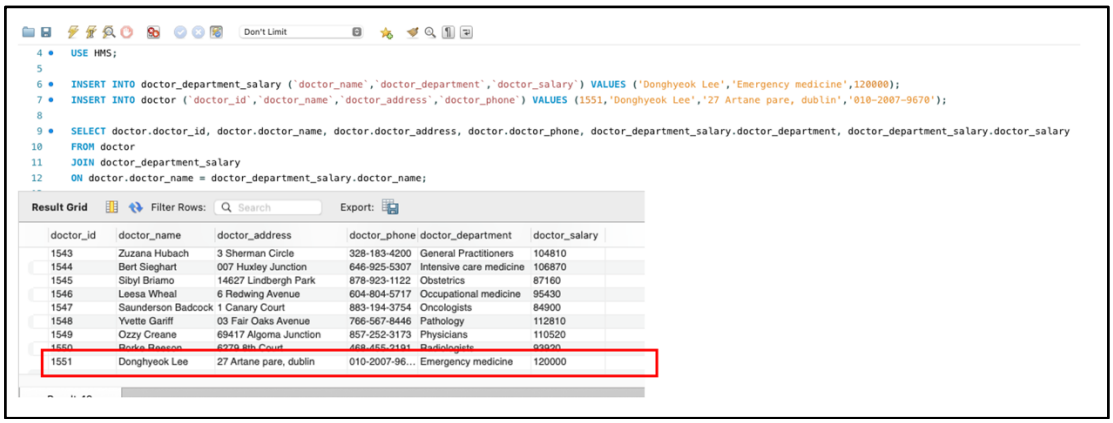
9) I assume that there are 150 times of Perform(test). In each time, there are some information of Patient id, Doctor id, test id (Primary keys), and perform date.

```
580 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1001,1501,2001,'2023-01-02');
581 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1001,1550,2026,'2023-03-30');
582 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1002,1502,2002,'2023-05-31');
583 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1002,1549,2027,'2023-05-28');
584 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1003,1503,2003,'2023-02-16');
585 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1003,1548,2028,'2023-02-21');
586 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1004,1504,2004,'2023-04-27');
587 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1004,1547,2029,'2023-09-27');
588 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1005,1505,2005,'2023-10-04');
589 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1005,1546,2030,'2023-12-05');
590 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1006,1506,2006,'2023-06-06');
591 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1006,1545,2031,'2023-12-10');
592 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1007,1507,2007,'2023-12-29');
593 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1007,1544,2032,'2023-04-24');
594 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1008,1508,2008,'2023-01-15');
595 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1008,1543,2033,'2023-06-20');
596 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1009,1509,2009,'2023-12-24');
597 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1009,1542,2034,'2023-06-20');
598 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1010,1510,2010,'2023-12-21');
599 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1010,1541,2035,'2023-03-13');
600 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1011,1511,2011,'2023-07-10');
601 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1011,1540,2036,'2023-12-25');
602 • INSERT INTO perform (`patient_id`,`doctor_id`,`test_id`,`perform_date`) VALUES (1012,1512,2012,'2023-05-11');
```

patient_id	doctor_id	test_id	perform_date
1001	1501	2001	2023-01-02
1001	1550	2026	2023-03-30
1002	1502	2002	2023-05-31
1002	1549	2027	2023-05-28
1003	1503	2003	2023-02-16
1003	1548	2028	2023-02-21
1004	1504	2004	2023-04-27
1004	1547	2029	2023-09-27
1005	1505	2005	2023-10-04
1005	1546	2030	2023-12-05
1006	1506	2006	2023-06-06
1006	1545	2031	2023-12-10
1007	1507	2007	2023-12-29
1007	1544	2032	2023-04-24
1008	1508	2008	2023-01-15
1008	1543	2033	2023-06-20
1009	1509	2009	2023-12-24
1009	1542	2034	2023-06-20
1010	1510	2010	2023-12-21

3.1. Insert a new Doctor with all the relevant information

(Insert new Doctor who is Donghyeok Lee with all the relevant information)

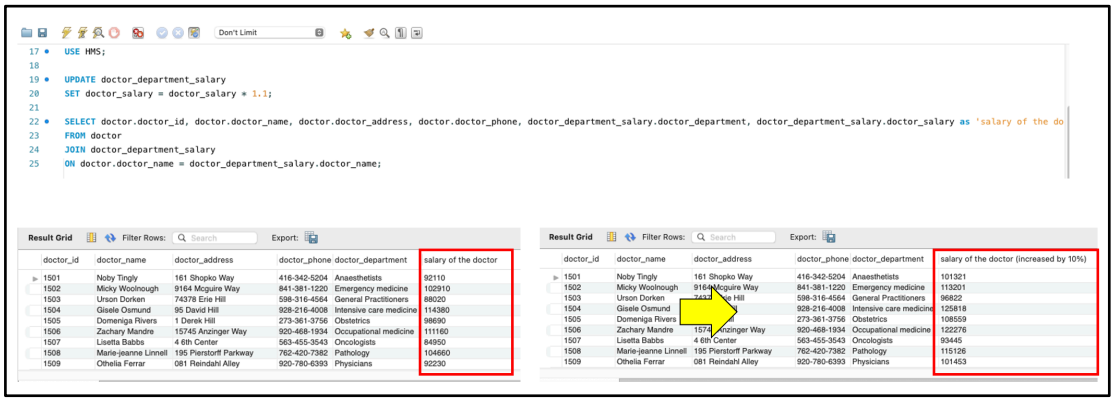


```
USE HMS;

INSERT INTO doctor_department_salary (`doctor_name`,`doctor_department`,`doctor_salary`) VALUES ('Donghyeok Lee','Emergency medicine',120000);
INSERT INTO doctor (`doctor_id`,`doctor_name`,`doctor_address`,`doctor_phone`) VALUES (1551,'Donghyeok Lee','27 Artane pare, dublin','010-2007-9670');

SELECT doctor.doctor_id, doctor.doctor_name, doctor.doctor_address, doctor.doctor_phone, doctor_department_salary.doctor_department, doctor_department_salary.doctor_salary
FROM doctor
JOIN doctor_department_salary
ON doctor.doctor_name = doctor_department_salary.doctor_name;
```

Result Grid

doctor_id	doctor_name	doctor_address	doctor_phone	doctor_department	doctor_salary
1543	Zuzana Hubach	3 Sherman Circle	328-183-4200	General Practitioners	104810
1544	Bert Sieghart	007 Huxley Junction	646-925-5307	Intensive care medicine	106870
1545	Sibyl Briamo	14627 Lindbergh Park	878-923-1122	Obstetrics	87160
1546	Leesa Wheal	6 Redwing Avenue	604-804-5717	Occupational medicine	95430
1547	Saunderson Badcock	1 Canary Court	883-194-3754	Oncologists	84900
1548	Yvette Gariff	03 Fair Oaks Avenue	766-567-8446	Pathology	112810
1549	Ozzy Creane	69417 Algoma Junction	857-252-3173	Physicians	110520
1551	Donghyeok Lee	27 Artane pare, dublin	010-2007-96...	Emergency medicine	120000

3.2. Increases the salary of the Doctor by 10%



```
USE HMS;

UPDATE doctor_department_salary
SET doctor_salary = doctor_salary * 1.1;

SELECT doctor.doctor_id, doctor.doctor_name, doctor.doctor_address, doctor.doctor_phone, doctor_department_salary.doctor_department, doctor_department_salary.doctor_salary as 'salary of the do
FROM doctor
JOIN doctor_department_salary
ON doctor.doctor_name = doctor_department_salary.doctor_name;
```

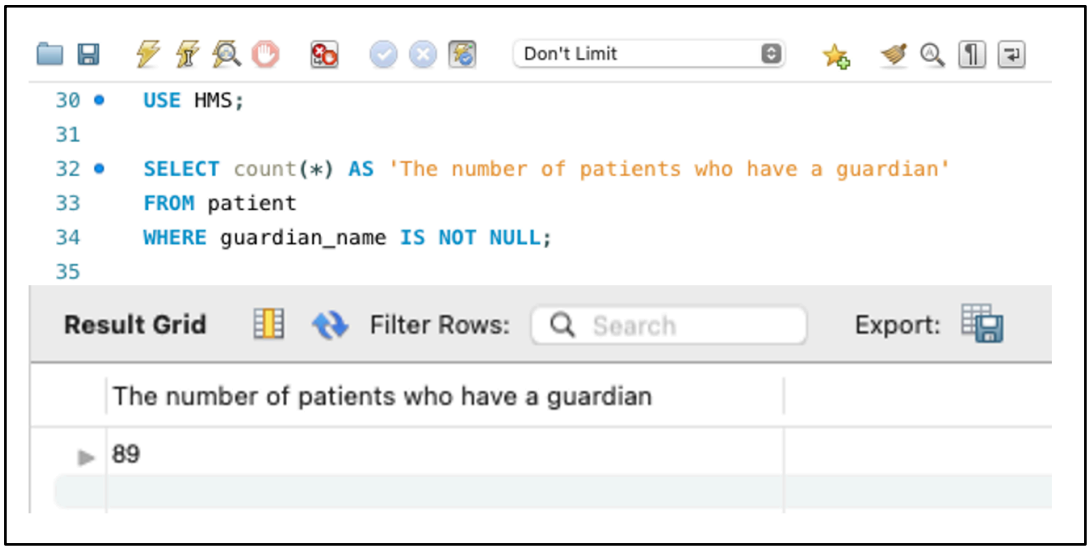
Result Grid

doctor_id	doctor_name	doctor_address	doctor_phone	doctor_department	salary of the doctor
1501	Noby Tingly	161 Shopko Way	416-342-5204	Anaesthetists	92110
1502	Micky Woolnough	9164 Mcguire Way	841-381-1220	Emergency medicine	102910
1503	Urson Dorken	74378 Erie Hill	598-316-4564	General Practitioners	88020
1504	Gisele Osmund	95 David Hill	928-216-4008	Intensive care medicine	114380
1505	Domeniga Rivers	1 Derek Hill	273-361-3756	Obstetrics	98690
1506	Zachary Mandre	15745 Anzinger Way	920-468-1934	Occupational medicine	111160
1507	Lisetta Babbs	4 6th Center	563-455-3543	Oncologists	84950
1508	Marie-jeanne Linnell	195 Pierstorff Parkway	762-420-7382	Pathology	104660
1509	Othelia Ferrar	081 Reindahl Alley	920-780-6393	Physicians	92230

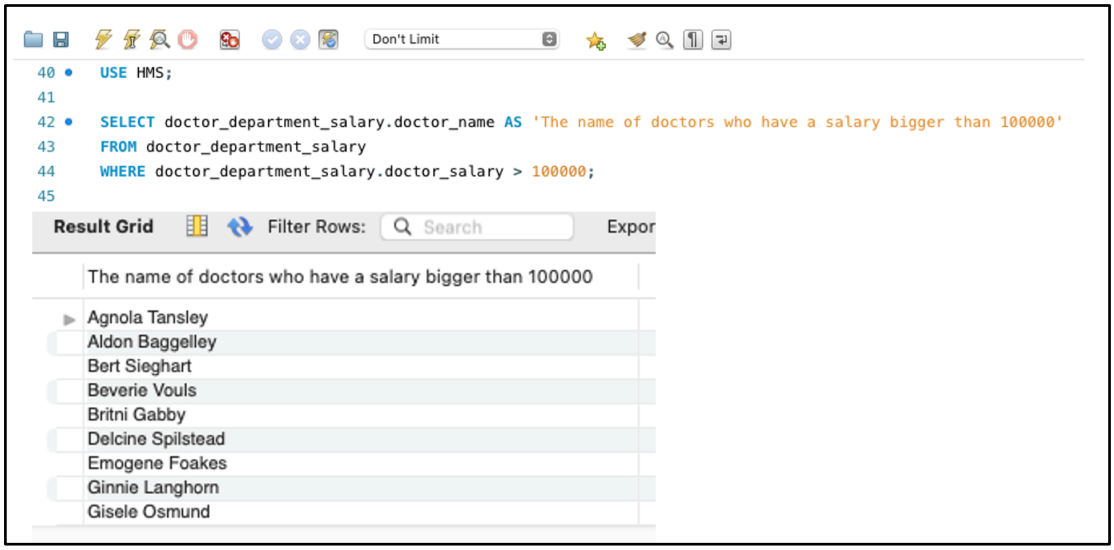
Result Grid

doctor_id	doctor_name	doctor_address	doctor_phone	doctor_department	salary of the doctor (increased by 10%)
1501	Noby Tingly	161 Shopko Way	416-342-5204	Anaesthetists	101321
1502	Micky Woolnough	9164 Mcguire Way	841-381-1220	Emergency medicine	113201
1503	Urson Dorken	74378 Erie Hill	598-316-4564	General Practitioners	96822
1504	Gisele Osmund	95 David Hill	928-216-4008	Intensive care medicine	125818
1505	Domeniga Rivers	1 Derek Hill	273-361-3756	Obstetrics	108559
1506	Zachary Mandre	15745 Anzinger Way	920-468-1934	Occupational medicine	122276
1507	Lisetta Babbs	4 6th Center	563-455-3543	Oncologists	93445
1508	Marie-jeanne Linnell	195 Pierstorff Parkway	762-420-7382	Pathology	115126
1509	Othelia Ferrar	081 Reindahl Alley	920-780-6393	Physicians	101453

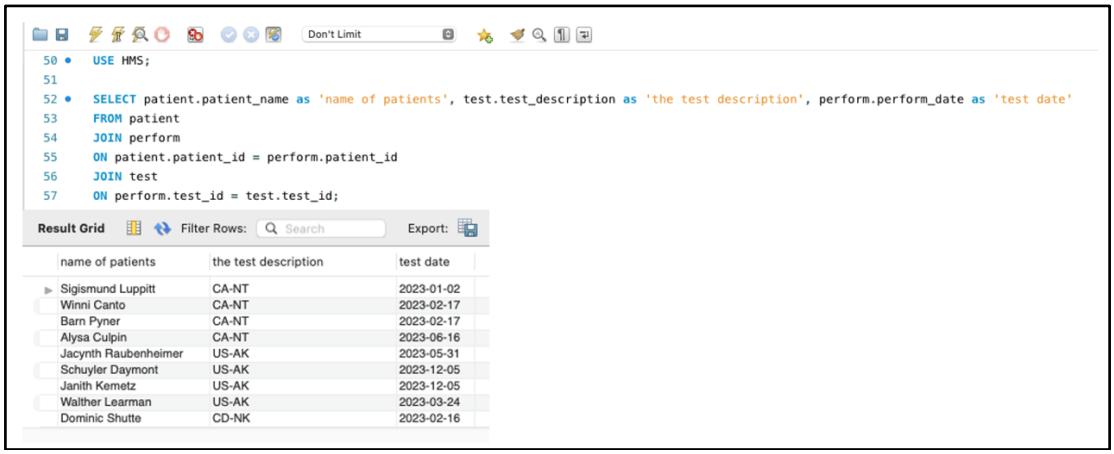
3.3. Display the number of patients who have a guardian



3.4. List the name of doctors if they have a salary bigger than 100000



3.5. List the name of patients, the test description, and the test date

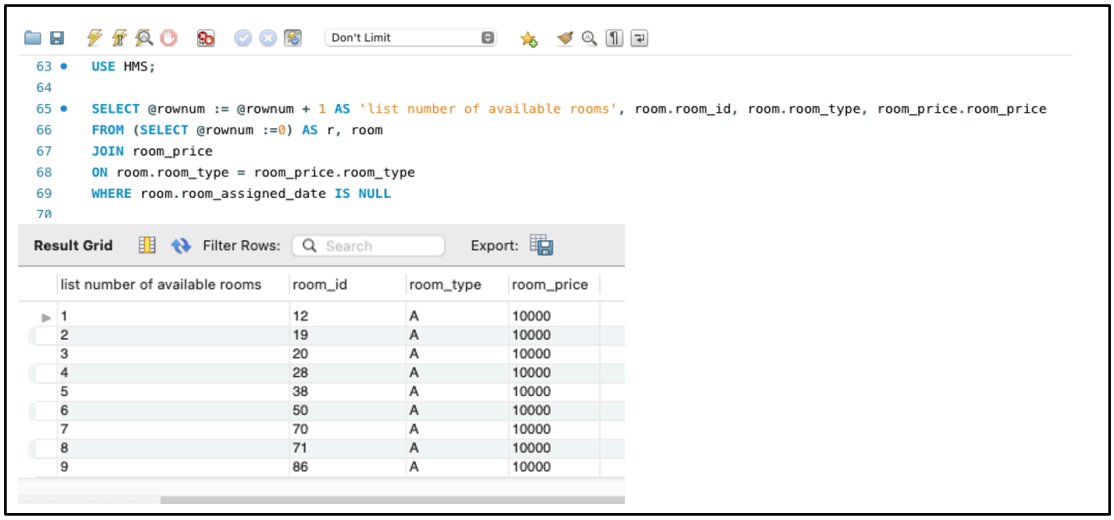


```
50 •  USE HMS;
51
52 •  SELECT patient.patient_name as 'name of patients', test.test_description as 'the test description', perform.perform_date as 'test date'
53    FROM patient
54    JOIN perform
55    ON patient.patient_id = perform.patient_id
56    JOIN test
57    ON perform.test_id = test.test_id;
```

Result Grid | Filter Rows: Search | Export:

name of patients	the test description	test date
Sigismund Luppitt	CA-NT	2023-01-02
Winni Canto	CA-NT	2023-02-17
Barn Pyner	CA-NT	2023-02-17
Alysa Culpin	CA-NT	2023-06-16
Jacynth Raubenheimer	US-AK	2023-05-31
Schuyler Daymont	US-AK	2023-12-05
Janith Kemetz	US-AK	2023-12-05
Walther Learman	US-AK	2023-03-24
Dominic Shutte	CD-NK	2023-02-16

3.6. List the number of available rooms



```
63 •  USE HMS;
64
65 •  SELECT @rownum := @rownum + 1 AS 'list number of available rooms', room.room_id, room.room_type, room_price.room_price
66    FROM (SELECT @rownum :=0) AS r, room
67    JOIN room_price
68    ON room.room_type = room_price.room_type
69    WHERE room.room_assigned_date IS NULL
70
```

Result Grid | Filter Rows: Search | Export:

list number of available rooms	room_id	room_type	room_price
1	12	A	10000
2	19	A	10000
3	20	A	10000
4	28	A	10000
5	38	A	10000
6	50	A	10000
7	70	A	10000
8	71	A	10000
9	86	A	10000

4.1. Non-Relational Database Management System
It is for Big Data Processing. From traditional relational database management systems (RDBMS), This is a non-relational DBMS designed differently, and its strength is that it can flexibly handle huge amounts of data. Examples of NoSQL-based systems include Apache Cassandra, Hadoop, and MongoDB.

4.2. Key factor about NoSQL : **Big Data from IoT**
Key factor driving the development and adoption of NoSQL (Not only SQL) is the explosion of data volume and diversity. Since the advent of the Internet in the 1990s, data of different types and sizes has been pouring in, and the amount of data has only continued to grow.

Furthermore, today's data sources really range from **big data from IoT devices**. Rigid table-style relational databases are no longer suitable for today's environment, where **such** diverse types of data sources are constantly emerging.

4.3. The BASE characteristics of non-relational databases with **Big data from IoT**

Characteristics	Description	Example with Big data from IoT
Basically **A**vailable	Ensuring the system's available in any events.	Even though main database that receive data from IoT suddenly shot down, copy database will be made and then operate normally.
Soft-state	Even though there is no interactions by eventual consistency, the state of data can change	When the database has any problem related to eventually consistency, this database input the data form IoT automatically to maintain consistency
Eventually Consistency	The system will be consistent after the program input. The data would be replicated to other nodes and eventually get a consistent state.	When the database doesn't maintain consistency, the system figures out the problem and maintain consistency again.

4.4. The types of non-relational databases and the scenarios suitable to use
1) Document-based type
Document databases have similar basic ideas to key-value stores. However, the structure is a little more complicated in that 'documents' include data, and each document is assigned its own key.

Document databases allow you to browse the contents of documents, allowing you to perform additional search processes. Unlike RDBMS, which requires a static schema, document databases have flexible schemas that are defined based on the content of the document.
(Tools: MongoDB, CouchDB and others)

2) Key-Value type

The key-value database consists of keys and values with all the data indexed, making it the simplest of the NoSQL databases. The key-value database uses a hash mechanism that allows you to quickly find the value associated with it when given a key.

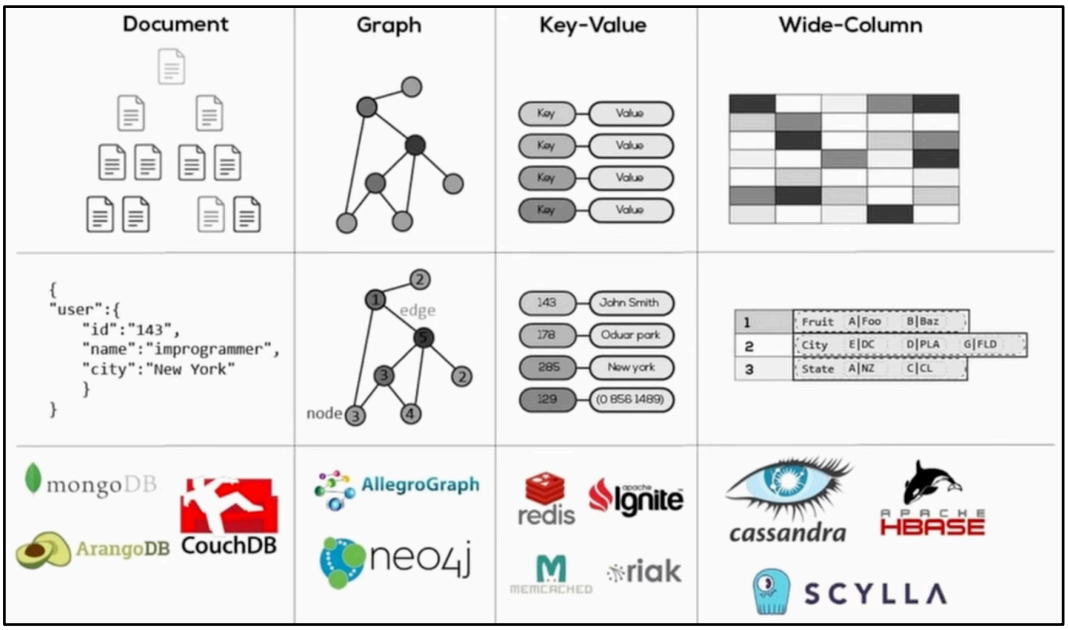
Hash mechanisms provide a constant amount of access, so you can maintain a high level of performance even in large databases.
(Tools: Redis, Amazon DynamoDB, Riak, Oracle NoSQL and others)

3) Graph-based type
Graph databases store data in a graphical way that uses the relationship between data. The nodes in the graph represent each data item, and the lines represent the relationship.

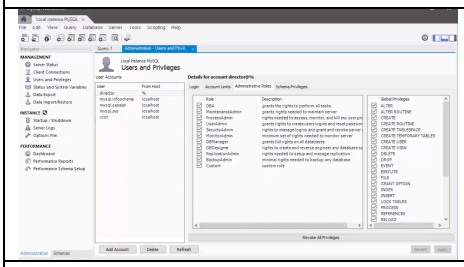
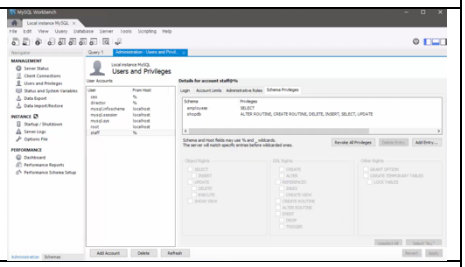
Graph databases handle extremely complex and highly connected data and have a level of relationship and join capabilities that go beyond RDBMS.
(Tools: DataStax Graph, Neo4J, JanusGraph, Amazon Neptune and others)

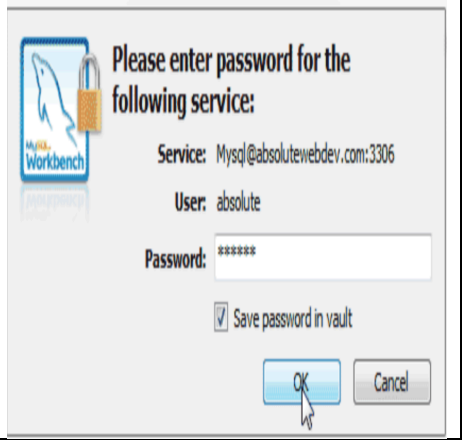
4) Column-based type
Column-based type databases arrange data in rows and columns, but differ from RDBMS. Column-based type databases, also known as Wide-Column store or partition row store, speed up queries by providing the option to organize associated rows into partitions and store them.

In addition, the table format is more flexible than the RDBMS. For example, with Cassandra, all rows in the table do not need to contain values for all columns. Column-based type databases use hash to retrieve rows from tables, such as key-value and document databases.
(Tools: Cassandra, HBase, Google Bigtable, and others)



5.1. Authorisation and Authentication

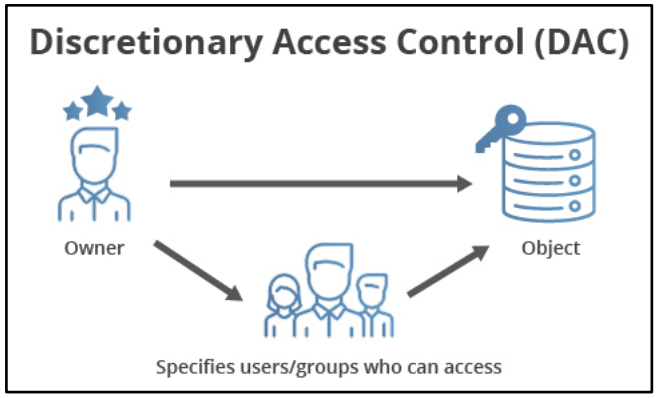
	Authorisation
Definition	The granting of a right or privilege, which enables a subject to have legitimate access to a system or a system's object
Example	Host would like to make the user of director and give the right of DBA (Administrative role – DBA) or any other things. → This processing is called 'Authorisation'   In workbench, it's possible to grant right or privilege to any users who host made. And all each user who host made can be given all different right related to accessing database.

	Authentication
Definition	Database authentication is that when user would like to attempt to access to the database that is secured, the system confirm that activities form he or she is authorized or not.
Example	 

5.2. Database Security Levels and how Discretionary Access Control in SQL maintains security

There are 3 levels of database security. These are database level, access level, and perimeter level. This security appears when the database is being in the system. Security approach at the border level decides who can and cannot get into databases. Each level requires one of a kind security arrangements.

Among the models considering database security, the fact that a database owner can arbitrarily set a policy on security is called discretionary access control. This method has been applied to database management systems for quite a long time. This method maintains system security by only giving users the specified form of read, write, or modify when giving users access to specific data files, records, or fields.



5.3. why Availability is an important aspect of secure DBMS and how it can be ensured.

This is meaning that the database's owner guarantee their data is available to every parties who are using their information. If this availability can not be maintained, it has huge impact to owner's business or any reputation. So, anyone or any corporate is keeping this availability. So, it's very important to ensure availability. Here are some methods how it can be ensured.

Method	Description
Data Back-Up	Data Back-up guarantees that data will not lost permanently, can be import, and is rapidly made available in case of a memory loss.
Avoid Single Points of Failure	Businesses have diverse access routes. This way, in case of failure of one access route or network, others continue to provide data .
Use the proper Tools	Using the right data can loss prevention tools. And It can help to protect data to a data from other risk.
Using Software Infrastructure	Businesses should try to use software-defined infrastructure and storage wherever possible, as it helps enhance data availability.
Eliminate dummy Data	If data is outdated, businesses have to archive them in the other storage system. Or securely dispose of it directly. This process will reduce clutter in the storage center and support that only the useful data is connecting users.
Erasure Coding	If data is saved in using erasure coding, then in case of data collapse or loss, it can be reconstructed using the other components stored other.
Cloud server for storage	Making data more efficiently, lots of businesses opt to move their data either whole into the cloud storage or into the hybrid model.