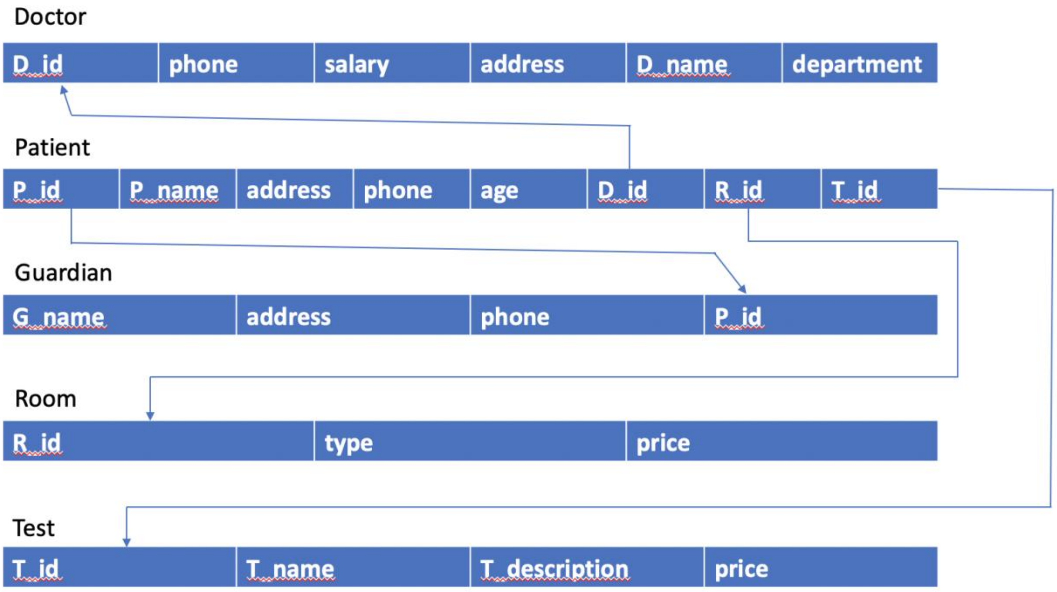
[Jiaying]_[Yu]_[HDCSDEV_INT]_IDB_TABA

Q1.



Doctor

D_id | phone | salary | address | D_name | department

Patient

P_id | P_name | address | phone | age | D_id | R_id | T_id

Guardian

G_name | address | phone | P_id

Room

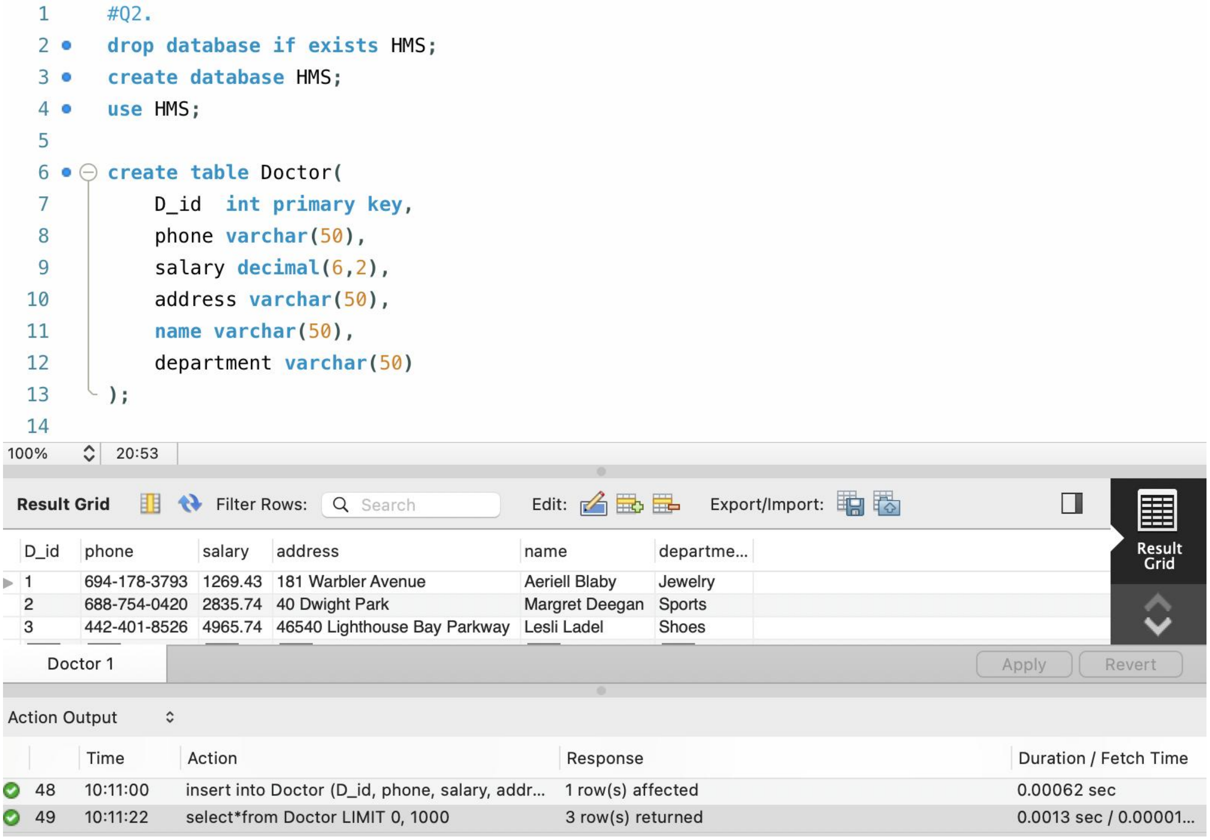
R_id | type | price

Test

T_id | T_name | T_description | price

Q2.

Table1 Doctor

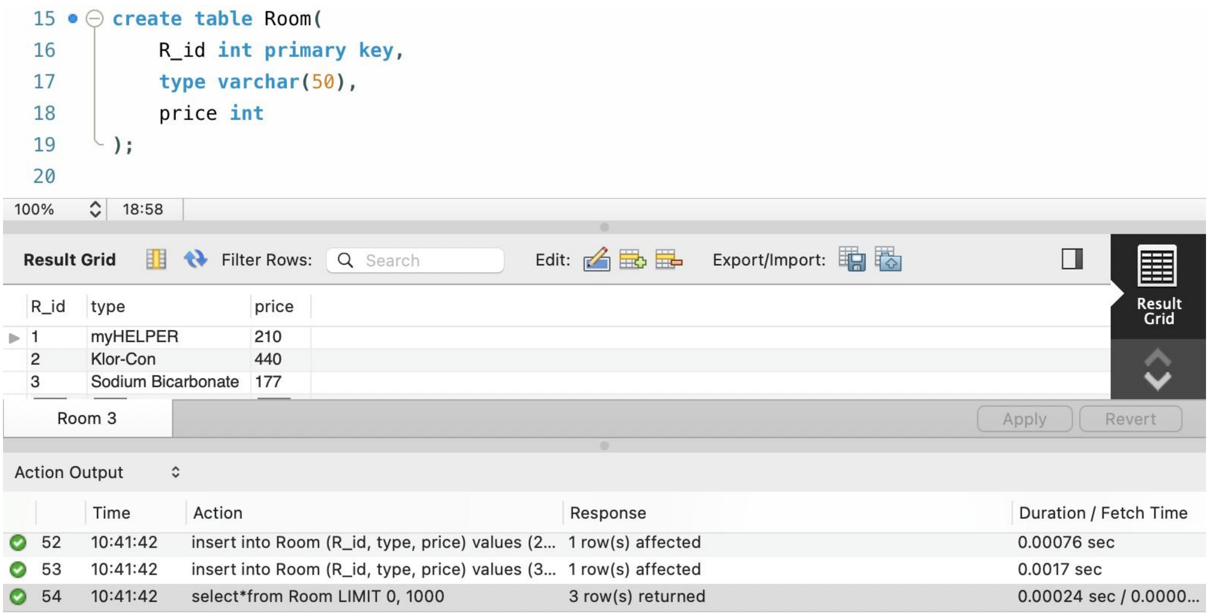


```
#Q2.
drop database if exists HMS;
create database HMS;
use HMS;

create table Doctor(
    D_id  int primary key,
    phone varchar(50),
    salary decimal(6,2),
    address varchar(50),
    name varchar(50),
    department varchar(50)
);

```

D_id	phone	salary	address	name	departme...
1	694-178-3793	1269.43	181 Warbler Avenue	Aeriell Blaby	Jewelry
2	688-754-0420	2835.74	40 Dwight Park	Margret Deegan	Sports
3	442-401-8526	4965.74	46540 Lighthouse Bay Parkway	Lesli Ladel	Shoes

	Time	Action	Response	Duration / Fetch Time
48	10:11:00	insert into Doctor (D_id, phone, salary, addr...	1 row(s) affected	0.00062 sec
49	10:11:22	select*from Doctor LIMIT 0, 1000	3 row(s) returned	0.0013 sec / 0.00001...

Table2 room

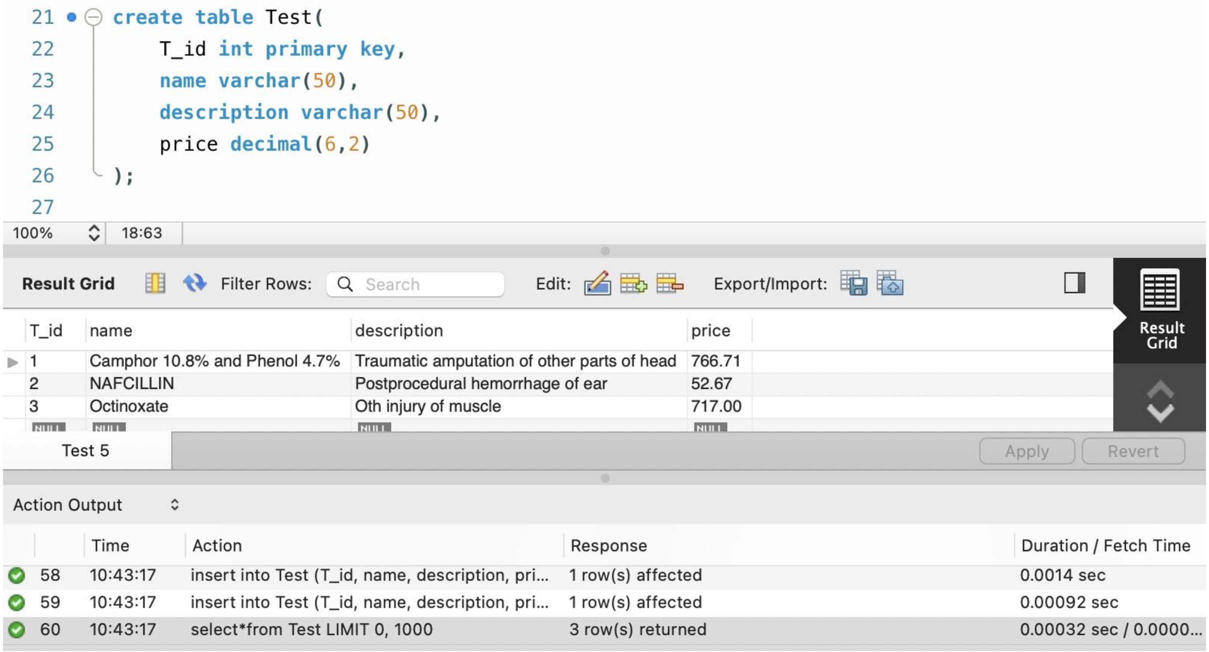


```sql
create table Room(
	R_id int primary key,
	type varchar(50),
	price int
);
```

R_id	type	price
1	myHELPER	210
2	Klor-Con	440
3	Sodium Bicarbonate	177

	Time	Action	Response	Duration / Fetch Time
52	10:41:42	insert into Room (R_id, type, price) values (2...	1 row(s) affected	0.00076 sec
53	10:41:42	insert into Room (R_id, type, price) values (3...	1 row(s) affected	0.0017 sec
54	10:41:42	select*from Room LIMIT 0, 1000	3 row(s) returned	0.00024 sec / 0.0000...

Table3 Test

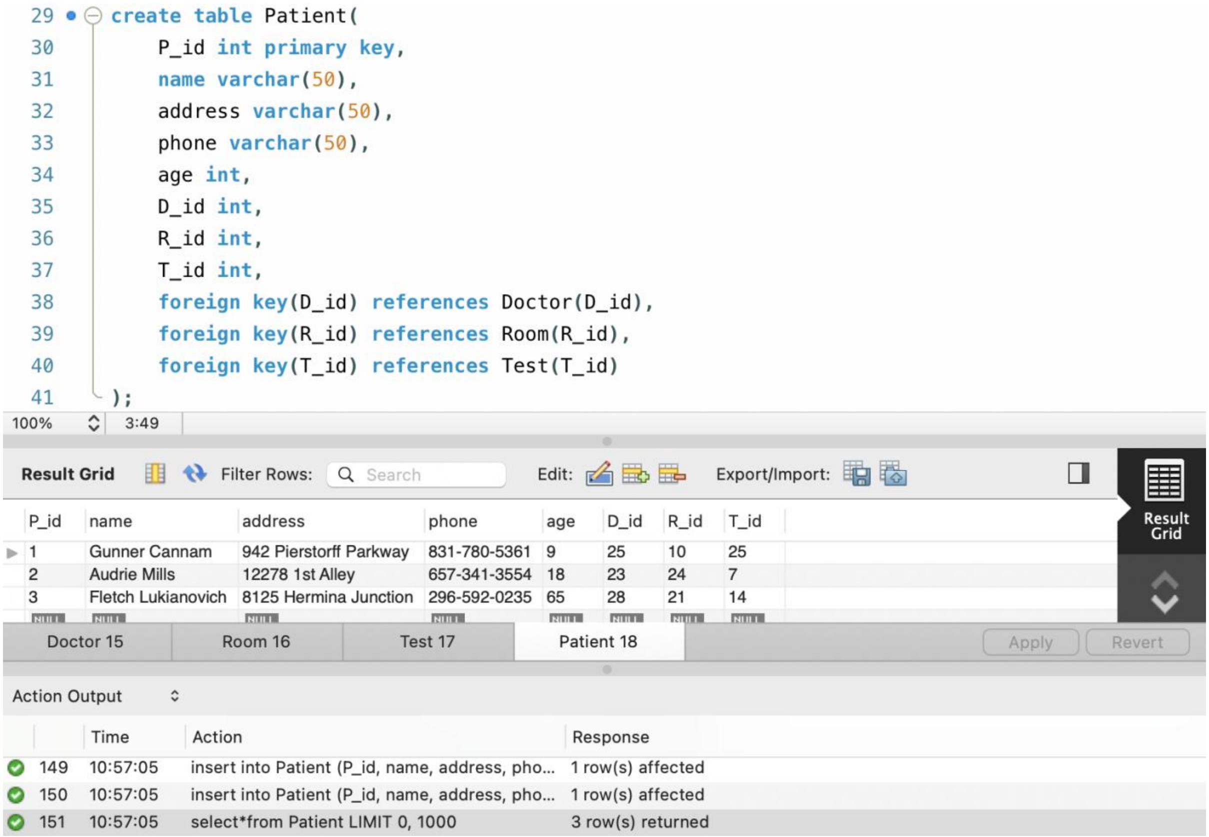


```sql
create table Test(
	T_id int primary key,
	name varchar(50),
	description varchar(50),
	price decimal(6,2)
);
```

T_id	name	description	price
1	Camphor 10.8% and Phenol 4.7%	Traumatic amputation of other parts of head	766.71
2	NAFCILLIN	Postprocedural hemorrhage of ear	52.67
3	Octinoxate	Oth injury of muscle	717.00
NULL	NULL	NULL	NULL

	Time	Action	Response	Duration / Fetch Time
58	10:43:17	insert into Test (T_id, name, description, pri...	1 row(s) affected	0.0014 sec
59	10:43:17	insert into Test (T_id, name, description, pri...	1 row(s) affected	0.00092 sec
60	10:43:17	select*from Test LIMIT 0, 1000	3 row(s) returned	0.00032 sec / 0.0000...

Table4 Patient

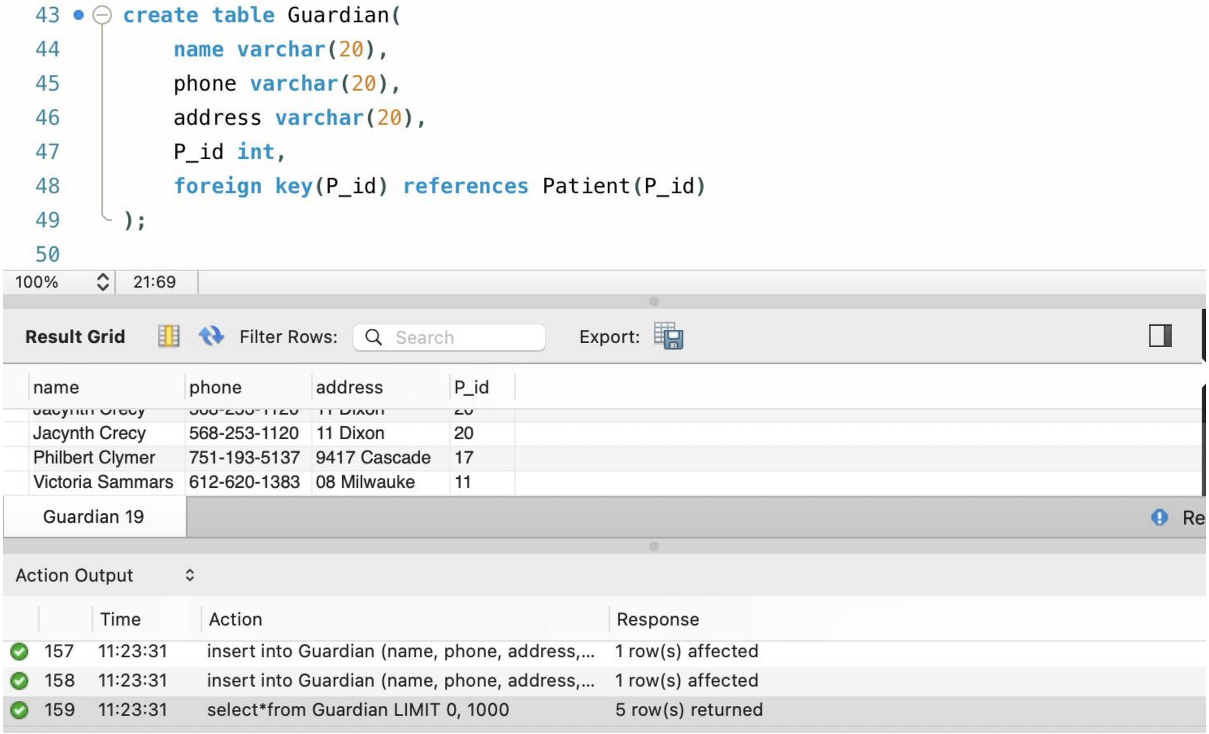


```
create table Patient(
    P_id int primary key,
    name varchar(50),
    address varchar(50),
    phone varchar(50),
    age int,
    D_id int,
    R_id int,
    T_id int,
    foreign key(D_id) references Doctor(D_id),
    foreign key(R_id) references Room(R_id),
    foreign key(T_id) references Test(T_id)
);
```

P_id	name	address	phone	age	D_id	R_id	T_id
1	Gunner Cannam	942 Pierstorff Parkway	831-780-5361	9	25	10	25
2	Audrie Mills	12278 1st Alley	657-341-3554	18	23	24	7
3	Fletch Lukianovich	8125 Hermina Junction	296-592-0235	65	28	21	14
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
     Time      Action                                        Response
149  10:57:05  insert into Patient (P_id, name, address, pho...  1 row(s) affected
150  10:57:05  insert into Patient (P_id, name, address, pho...  1 row(s) affected
151  10:57:05  select*from Patient LIMIT 0, 1000             3 row(s) returned
```

Table5 Guardian

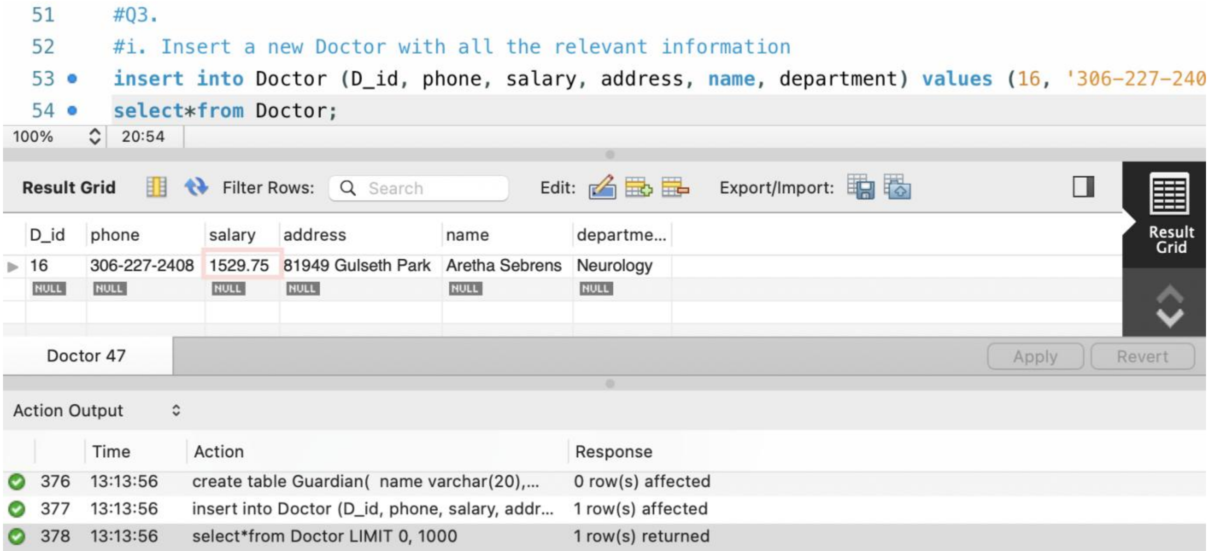


```
create table Guardian(
    name varchar(20),
    phone varchar(20),
    address varchar(20),
    P_id int,
    foreign key(P_id) references Patient(P_id)
);
```

name	phone	address	P_id
Jacynth Crecy	568-253-1120	11 Dixon	20
Philbert Clymer	751-193-5137	9417 Cascade	17
Victoria Sammars	612-620-1383	08 Milwauke	11

```
     Time      Action                                          Response
157  11:23:31  insert into Guardian (name, phone, address,...  1 row(s) affected
158  11:23:31  insert into Guardian (name, phone, address,...  1 row(s) affected
159  11:23:31  select*from Guardian LIMIT 0, 1000              5 row(s) returned
```

Q3. i

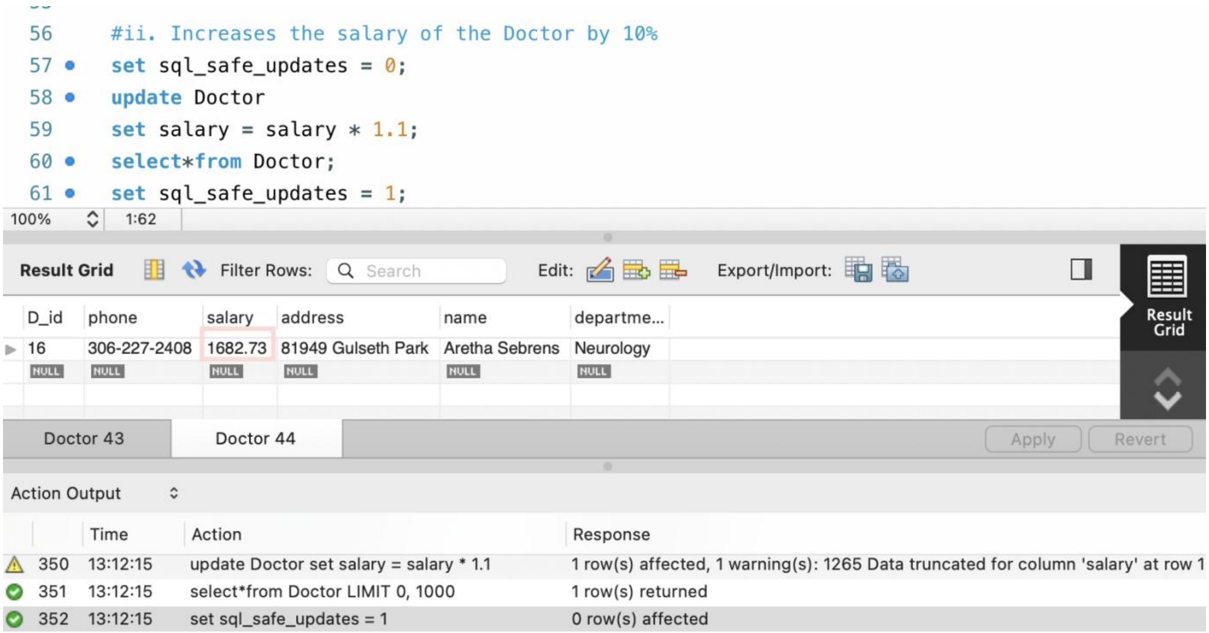


```
#Q3.
#i. Insert a new Doctor with all the relevant information
insert into Doctor (D_id, phone, salary, address, name, department) values (16, '306-227-240
select*from Doctor;
```

D_id	phone	salary	address	name	departme...
16	306-227-2408	1529.75	81949 Gulseth Park	Aretha Sebrens	Neurology
NULL	NULL	NULL	NULL	NULL	NULL

	Time	Action	Response
376	13:13:56	create table Guardian(name varchar(20),...	0 row(s) affected
377	13:13:56	insert into Doctor (D_id, phone, salary, addr...	1 row(s) affected
378	13:13:56	select*from Doctor LIMIT 0, 1000	1 row(s) returned

ii.

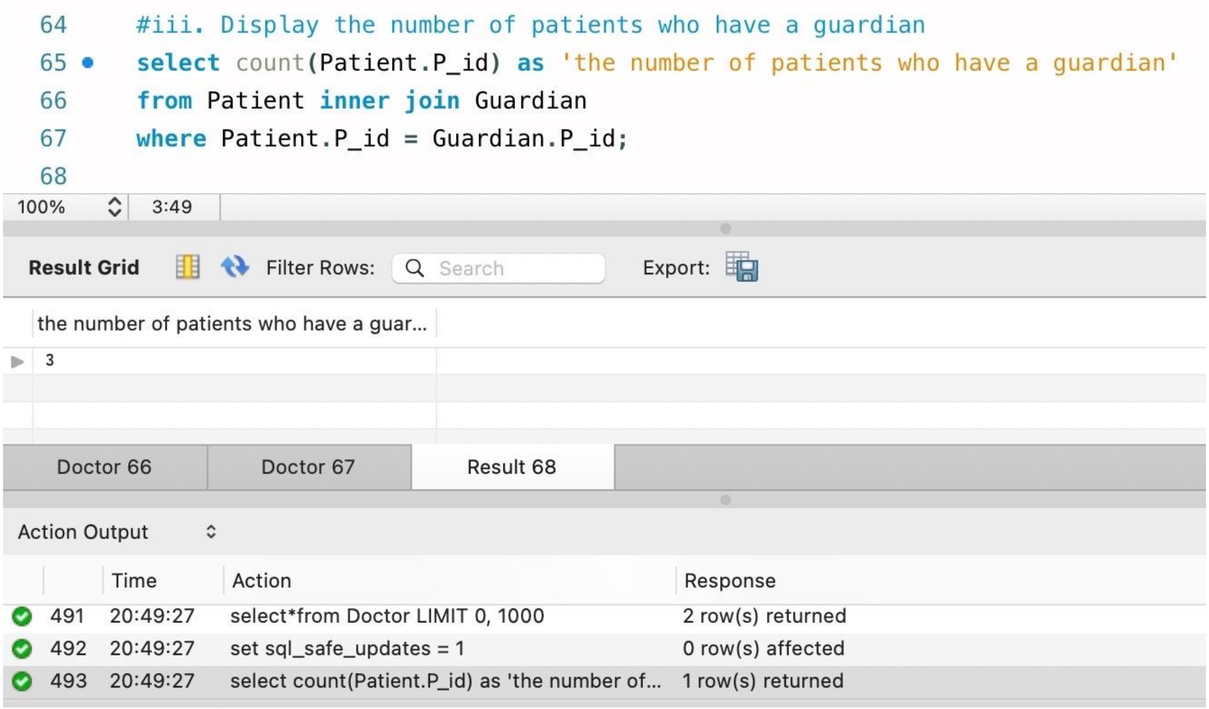


```
#ii. Increases the salary of the Doctor by 10%
set sql_safe_updates = 0;
update Doctor
set salary = salary * 1.1;
select*from Doctor;
set sql_safe_updates = 1;
```

D_id	phone	salary	address	name	departme...
16	306-227-2408	1682.73	81949 Gulseth Park	Aretha Sebrens	Neurology
NULL	NULL	NULL	NULL	NULL	NULL

	Time	Action	Response
⚠ 350	13:12:15	update Doctor set salary = salary * 1.1	1 row(s) affected, 1 warning(s): 1265 Data truncated for column 'salary' at row 1
351	13:12:15	select*from Doctor LIMIT 0, 1000	1 row(s) returned
352	13:12:15	set sql_safe_updates = 1	0 row(s) affected

iii.

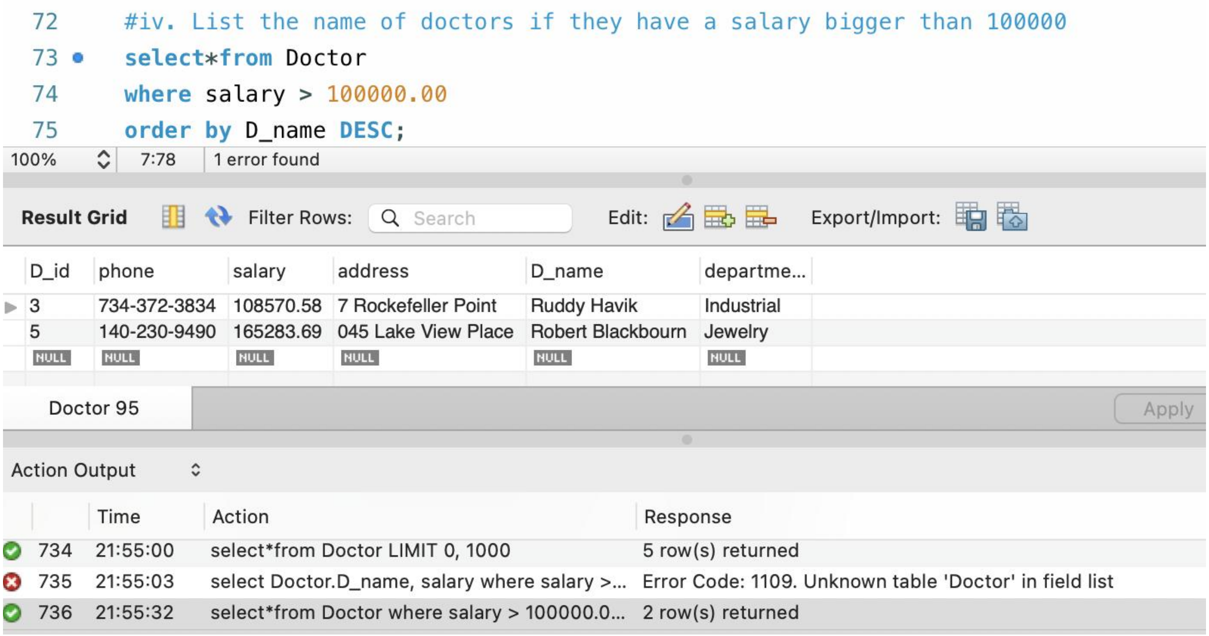


```
#iii. Display the number of patients who have a guardian
select count(Patient.P_id) as 'the number of patients who have a guardian'
from Patient inner join Guardian
where Patient.P_id = Guardian.P_id;
```

the number of patients who have a guar...
3

		Time	Action	Response
✓	491	20:49:27	select*from Doctor LIMIT 0, 1000	2 row(s) returned
✓	492	20:49:27	set sql_safe_updates = 1	0 row(s) affected
✓	493	20:49:27	select count(Patient.P_id) as 'the number of...	1 row(s) returned

iv.



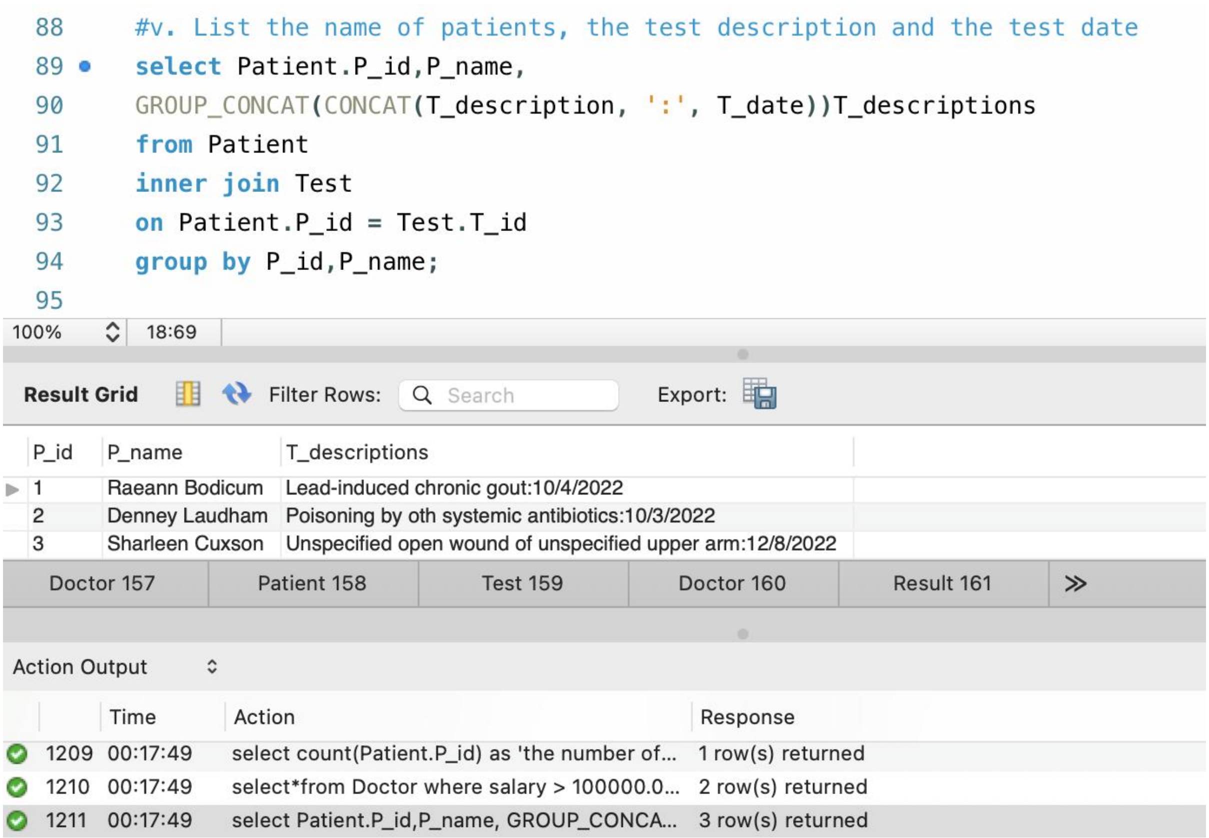
```
#iv. List the name of doctors if they have a salary bigger than 100000
select*from Doctor
where salary > 100000.00
order by D_name DESC;
```

1 error found

D_id	phone	salary	address	D_name	departme...
3	734-372-3834	108570.58	7 Rockefeller Point	Ruddy Havik	Industrial
5	140-230-9490	165283.69	045 Lake View Place	Robert Blackbourn	Jewelry
NULL	NULL	NULL	NULL	NULL	NULL

		Time	Action	Response
✓	734	21:55:00	select*from Doctor LIMIT 0, 1000	5 row(s) returned
✗	735	21:55:03	select Doctor.D_name, salary where salary >...	Error Code: 1109. Unknown table 'Doctor' in field list
✓	736	21:55:32	select*from Doctor where salary > 100000.0...	2 row(s) returned

v.

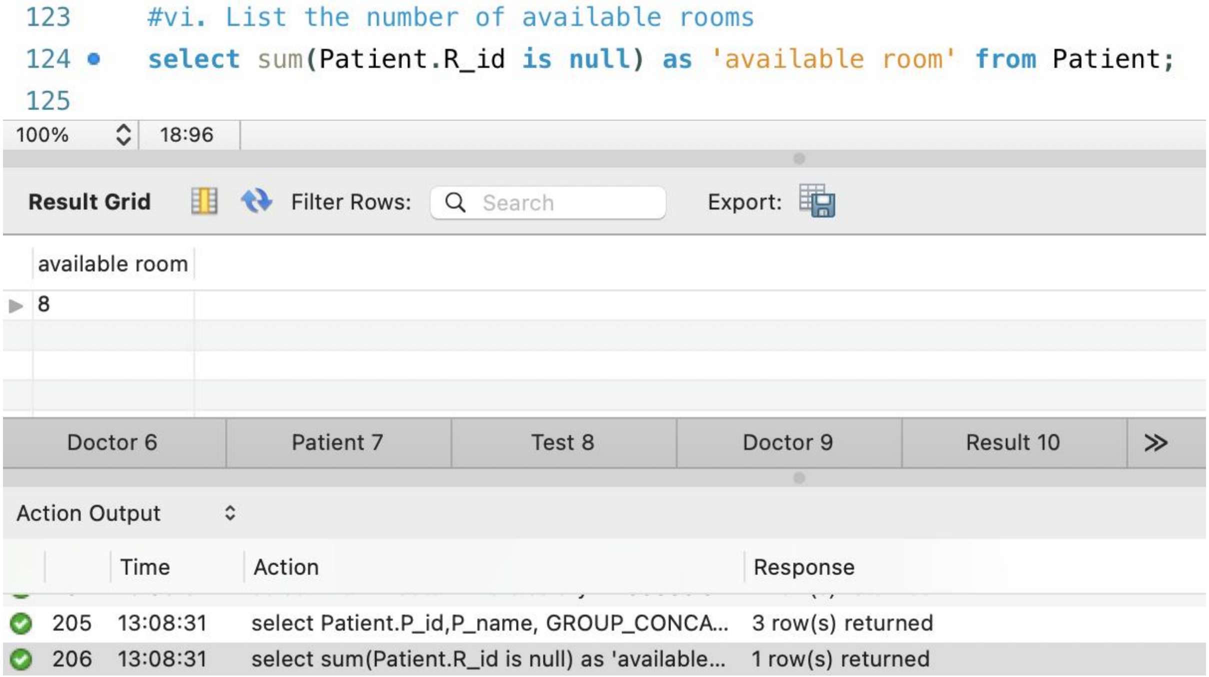


```sql
#v. List the name of patients, the test description and the test date
select Patient.P_id,P_name,
GROUP_CONCAT(CONCAT(T_description, ':', T_date))T_descriptions
from Patient
inner join Test
on Patient.P_id = Test.T_id
group by P_id,P_name;

```

P_id	P_name	T_descriptions
1	Raeann Bodicum	Lead-induced chronic gout:10/4/2022
2	Denney Laudham	Poisoning by oth systemic antibiotics:10/3/2022
3	Sharleen Cuxson	Unspecified open wound of unspecified upper arm:12/8/2022

```
      Time      Action                                        Response
1209  00:17:49  select count(Patient.P_id) as 'the number of...  1 row(s) returned
1210  00:17:49  select*from Doctor where salary > 100000.0...    2 row(s) returned
1211  00:17:49  select Patient.P_id,P_name, GROUP_CONCA...       3 row(s) returned
```

vi.



```sql
#vi. List the number of available rooms
select sum(Patient.R_id is null) as 'available room' from Patient;

```

available room
8

```
     Time      Action                                          Response
205  13:08:31  select Patient.P_id,P_name, GROUP_CONCA...        3 row(s) returned
206  13:08:31  select sum(Patient.R_id is null) as 'available...  1 row(s) returned
```

MySQL code—Jiaying Yu—x22142762

```
#Q2.
drop database if exists HMS;
create database HMS;
use HMS;
set foreign_key_checks = 0;

create table Doctor(
        D_id  int primary key,
        phone varchar(50),
        salary decimal(10,2),
        address varchar(50),
        D_name varchar(50),
        department varchar(50)
);

create table Room(
        R_id int primary key,
        type varchar(50),
        price int
);

create table Test(
        T_id int primary key,
        T_name varchar(50),
        T_description varchar(50),
        price decimal(10,2),
        T_date varchar(50)
);

create table Patient(
        P_id int primary key,
        P_name varchar(50),
        address varchar(50),
        phone varchar(50),
        age int,
        D_id int,
        R_id int,
        T_id int,
   foreign key(D_id) references Doctor(D_id),
   foreign key(R_id) references Room(R_id),
   foreign key(T_id) references Test(T_id)
);

create table Guardian(
        G_name varchar(20),
        phone varchar(20),
        address varchar(20),
        P_id int,
    foreign key(P_id) references Patient(P_id)
);
#Q3.
#i. Insert a new Doctor with all the relevant information
insert into Doctor (D_id, phone, salary, address, D_name, department) values (16,
'306-227-2408', 1529.75, '81949 Gulseth Park', 'Aretha Sebrens', 'Neurology');
insert into Doctor (D_id, phone, salary, address, D_name, department) values (3,
'734-372-3834', 108570.58, '7 Rockefeller Point', 'Ruddy Havik', 'Industrial');
insert into Doctor (D_id, phone, salary, address, D_name, department) values (4,
'446-874-4249', 76898.19, '0 Sloan Street', 'Johny Dudenie', 'Books');
insert into Doctor (D_id, phone, salary, address, D_name, department) values (5,
'140-230-9490', 165283.69, '045 Lake View Place', 'Robert Blackbourn', 'Jewelry');
insert into Doctor (D_id, phone, salary, address, D_name, department) values (6,
'205-441-7469', 43394.05, '03 Forest Run Circle', 'Loreen Loyd', 'Shoes');
select*from Doctor;

#ii. Increases the salary of the Doctor by 10%
set sql_safe_updates = 0;
update Doctor
set salary = salary * 1.1;
select*from Doctor;
set sql_safe_updates = 1;

#iii. Display the number of patients who have a guardian
select count(Patient.P_id) as 'the number of patients who have a guardian'
from Patient inner join Guardian
where Patient.P_id = Guardian.P_id;

#iv. List the name of doctors if they have a salary bigger than 100000
select*from Doctor
where salary > 100000.00
order by D_name DESC;

#v. List the name of patients, the test description and the test date
select Patient.P_id,P_name,
GROUP_CONCAT(CONCAT(T_description, ':', T_date))T_descriptions
from Patient
inner join Test
on Patient.P_id = Test.T_id
group by P_id,P_name;

#vi. List the number of available rooms
select sum(Patient.R_id is null) as 'available room' from Patient;
```

Q4. Discuss the BASE characteristics of non-relational databases by choosing one of the contexts below. Discuss the types of non-relational databases and the scenarios suitable to use. 22142762-Social media

For ‘BASE’, ’BA’ is Basically Available, ‘S’ is Soft State, ‘E’ is Eventually Consistent. For social media, it consist of 5 main parts. Profile, Posts, Relationships and Media. Profile: stores basic information about users; Posts: gives users the ability to share their thoughts; Relationships: creates connections between different users; Media: allows users to share photos or videos. Based on these features, social network data is more flexible and difficult to manage.

For example, MongoDB works with large amounts of data. It has different concentrations, while sharing the advantages of NoSQL. Social media is spread in various domains, which need various features at the same time. In addition, NoSQL plays an important role in Big Data in social media. NoSQL data storage systems are ideal for applications such as social media that need to handle very large amounts of semi-structured data. It allows some parts of the distributed system to fail while the rest of the system is still available, which is why NOSQL data storage systems are used to a large extent in contemporary social media, given the large number of users and the freedom to personalize content.

For the application of Soft State and Eventually Consistent in social media, first of all, technicians do not need to spend a lot of time learning how to code for MongoDB, compared to traditional SQL. In addition, scalability is one of the main advantages for companies using MongoDB.

When the data volume increases, through the process of storing data records on multiple machine nodes, the growth of data volume and the demand for read and write operations can be supported by adding more machines to handle huge data. This also applies to the scenarios where NOSQL is suitable for use.

Therefore, it is a better way to handle the growing data in social networks. Then, MongoDB is a document-oriented database, where a collection contains different documents with the number of fields. The content and size of the documents can vary from one file to another. This feature makes MongoDB schema-less, even if these documents need to be 'changed' at any time. This makes 'change' operations easier to implement, especially when documents are separated on different nodes of the machine. In addition, MongoDB defines collection-level indexes and supports indexing of any file in a document, allowing for more efficient query execution.

Q5. Differentiate between Authorisation and Authentication with a suitable example to elaborate?

Authentication is the process of verifying a user's identity, while authorization is the process of verifying what a user can access, with authentication visible to the user and partially changeable by the user. Authorization is not visible or changeable by the user. For example, when a user goes through security at an airport, the user is asked to present their ID to verify their identity without authorizing the user's ID.

With the help of a suitable diagram, explain the various Database Security Levels?

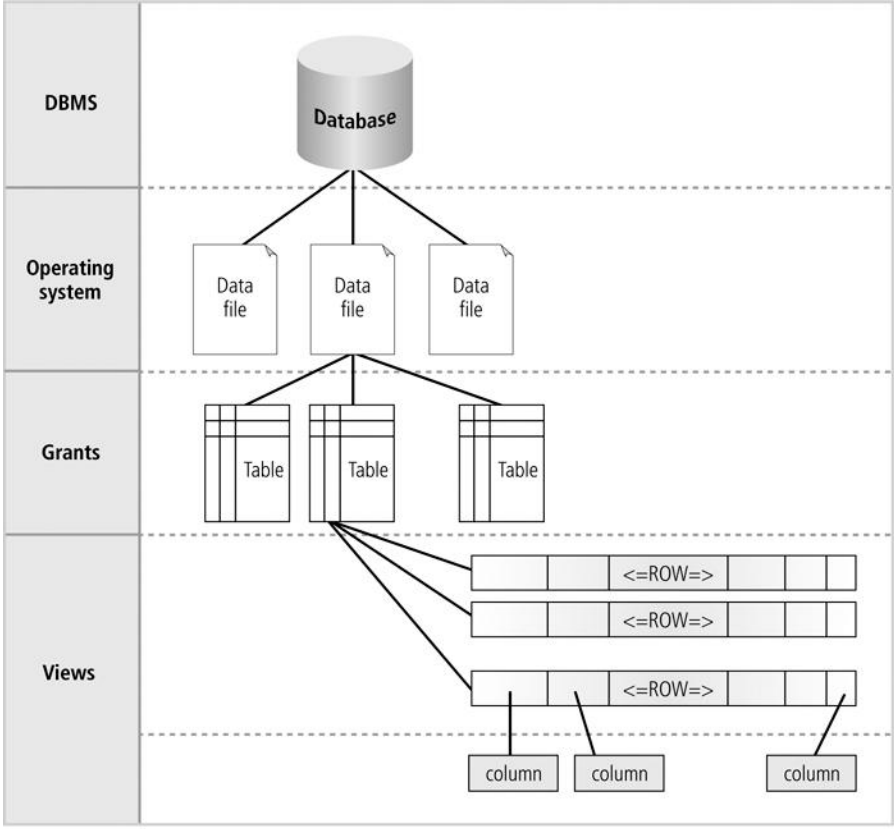


FIGURE 1-11 Levels of database security

Database security levels encompasses security that ensures that databases are not accessed, modified, or permissions are not deleted. This is reflected in the Operating system, grants and views. Since databases define a business-critical resource, database security is an important part of the security of entire information systems in some organizations.

How Discretionary Access Control in SQL maintains security?

As for server security, the best course of action is to secure the hardware and use internal system sockets to MySQL and make sure to block any network access to the MySQL server. Ensure that users use both system privileges and MySQL privileges to allow as little access as possible. For certain scripts, users may consider write-only authentication. However, no encryption method is foolproof because users will always need to decrypt, so keys must be stored. Of course, the user can store the key in one place, and if a breach is detected in the system, the user can destroy the file, rendering the data unusable.

Discuss why Availability is an important aspect of secure DBMS?

Availability ensures that systems, applications, and data are available and accessible to authorized users when needed. Availability plays an important role in the Database management system because the network, systems and applications must be constantly up and running, and because availability ensures that critical business processes remain uninterrupted.

Ensure data availability
1. Backing up data for logical errors and failures;
2. Storing data with different methods, such as storage networking technologies: DAS, NAS, SAN, or CDM backup all-in-one.