Heather Dumond
Department of Computer Science
Database Management Systems (CSC570.001)
Spring 2017 (40 pts)
Mehdi Owrang

You need to provide a complete copy of the entire project:

1. Inputs / outputs
2. Sql statements / sql results
3. Codes for triggers
4. Codes for HTML/Coldfuion
5. On-line operations
6. Screenshots

> INTRODUCTION <

For this project, I chose to create a hospital management database. I figured this would be a reasonable project to take on because it will be clear to me, as the designer, whether my entities and their relationships are reasonable or not. It also has practical real-world application as we will all likely encounter a hospital at some point in our lives, and the medical tech world has limitless possibilities for programmers and database engineers.

> DEFINING OUR ENTITIES <

1 - Define the information content of your database

a) Define a set of entities and appropriate attributes for each entity**. Minimum 10 entities**.

Entities & Respective Attributes

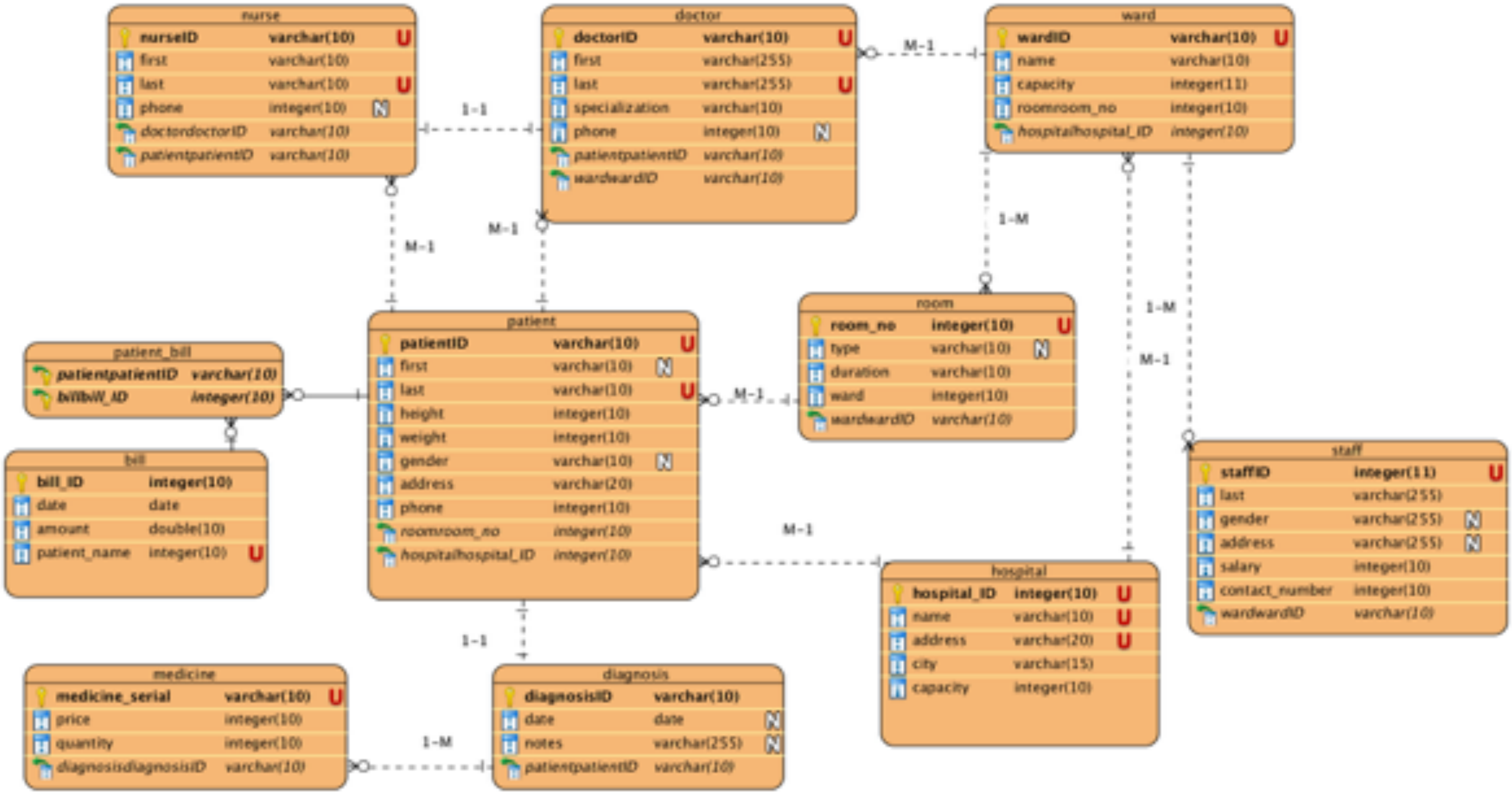
1. **Nurse**
    1. First
    2. Last
    3. Phone
2. **Doctor**
    1. First
    2. Last
    3. Specialization
    4. Phone

3. **W**ard
    1. Name
    2. Capacity
    3. RoomNo
4. **Bill**
    1. date
    2. amount
    3. Patient name
5. **Patient**
    1. first
    2. last
    3. height
    4. weight
    5. gender
    6. address
    7. phone
    8. room no
6. **Medicine**
    1. price
    2. quantity
7. **Diagnosis**
    1. date
    2. notes
8. **Hospital**
    1. name
    2. address
    3. city

    4. capacity

9. **Room**
    1. type
    2. duration
    3. ward
10. **Staff**
    1. last
    2. gender
    3. address
    4. salary
    5. contact_number

2) Define an E-R Diagram for your database design.



3) Define a relational schema for your database design.

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

nurse (nurseID, nurse-first, nurse-last, nurse-phone, doctorID, patientID)

doctor (doctorID, doctor-first, doctor-last, doctor-specialization, doctor-phone, patientID, wardID)

ward (wardID, ward-name, ward-capacity, room_no, hospital_ID)

patient_bill (patientID, bill_ID)

bill (bill_ID, bill-date, bill-amount, bill-patient_name)

medicine (medicine_serial, medicine-price, medicine-quantity, diagnosisID)

patient (patientID, patient-first, patient-last, patient-height, patient-height, patient-weight, patient-gender, patient-address, patient-phone, room_no, hospital_ID)

diagnosis (diagnosisID, diagnosis-date, diagnosis-notes, patientID)

hospital (hospital_ID, hospital-name, hospital-address, hospital-city, hospital-city, hospital-capacity)

room (room_no, room-type, room-duration, room-ward, wardID)

staff (staffID, staff-last, staff-gender, staff-address, staff-salary, staff-contact_number, wardID)

> RELATIONSHIPS <

1-1

- Nurse > Doctor
- Diagnosis > Patient

1-M

- Patient > Nurse
- Patient > Doctor
- Diagnosis > Medicine
- Hospital > Patient
- Hospital > Ward
- Ward > Staff
- Ward > Room
- Room > Patient

M-M

- `Patient > Bill`

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

Functional Dependencies

(A —> B = B is functionally dependent on A)

- diagnosisID > medicine_serial
- patientID > first, last
- hospital_ID > wardID
- wardID > Room_number
- Ward > StaffID
- patientID > doctorID
- doctorID > nurseID
- patientID > bill_ID

Multi-valued Dependencies

- patientID ->> first
- patientID ->> last
- patientID ->> address
- patientID ->> phone
- doctorID ->> specialization
- doctorID ->> last
- nurse ->> last
- patientID ->> phone
- ward ->> capacity
- ward ->> room_no
- room_no ->> wardID
- room ->> type

- staffID ->> last
- staffID ->> wardID
- staffID ->> contact_number
- diagnosisID ->> patientID
- hospitalID ->> name
- hospitalID ->> address

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

TABLE	PRIMARY KEY	FOREIGN KEY	NORMALIZATION
nurse	nurseID	doctorID, patientID	4NF
doctor	doctorID	patientID, wardID	4NF
ward	wardID	hospital_ID	2NF
bill	bill_ID	patientID	2NF
medicine	medicine_serial	diagnosisID	3NF
patient	patientID	room_no, hospital_ID	2NF
diagnosis	diagnosisID	patientID	2NF
hospital	hospital_ID		1NF
room	room_no	wardID	3NF
staff	staffID	wardID	3NF

4) Implementation: Create your database using MySQL 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.

> Create tables <

SQL STATEMENTS:

**SQL code to generate and populate tables will be included in the zip file as a separate document but below I have included a version as well.

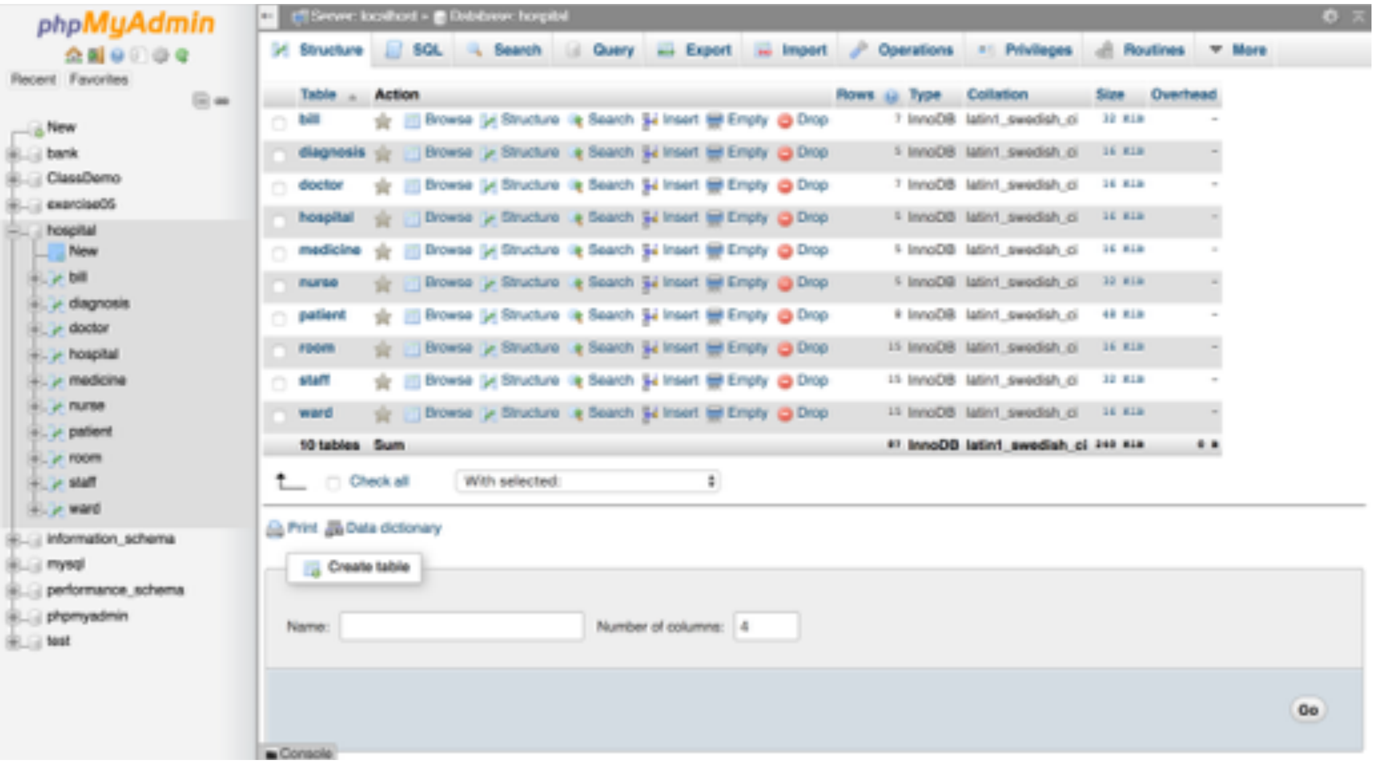
```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

-- Database: `hospital`
-- Table structure for table `bill`

CREATE TABLE `bill` (
  `billID` int(10) NOT NULL,
  `date` date NOT NULL,
  `amount` double NOT NULL,
  `patientID` varchar(9) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

-- Table structure for table `diagnosis`

CREATE TABLE `diagnosis` (
  `diagnosisID` int(11) NOT NULL,
  `date` date NOT NULL,
  `notes` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `doctor`

CREATE TABLE `doctor` (
  `doctorID` varchar(9) NOT NULL,
  `first` varchar(255) NOT NULL,
  `last` varchar(255) NOT NULL,
  `wardID` varchar(255) NOT NULL,
  `hospitalID` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Table structure for table `hospital`

CREATE TABLE `hospital` (
  `hospitalID` varchar(255) NOT NULL,
  `name` varchar(255) NOT NULL,
  `address` varchar(255) NOT NULL,
  `city` varchar(255) NOT NULL,
  `capacity` int(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `medicine`
--

CREATE TABLE `medicine` (
  `medicine_serial` varchar(10) NOT NULL,
  `price` int(10) NOT NULL,
  `quantity` int(10) NOT NULL,
  `diagnosisID` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `nurse`
--
```

```
CREATE TABLE `nurse` (
  `nurseID` varchar(10) NOT NULL,
  `first` varchar(10) NOT NULL,
  `last` varchar(10) NOT NULL,
  `phone` int(10) NOT NULL,
  `doctorID` varchar(9) NOT NULL,
  `patientID` varchar(9) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `patient`
--

CREATE TABLE `patient` (
  `patientID` varchar(9) NOT NULL,
  `first` varchar(20) NOT NULL,
  `last` varchar(20) NOT NULL,
  `weight` int(3) NOT NULL,
  `height_cm` int(255) NOT NULL,
  `gender` varchar(20) NOT NULL,
  `address` varchar(50) NOT NULL,
  `phone_number` int(10) NOT NULL,
  `doctorID` varchar(9) NOT NULL,
  `wardID` varchar(255) NOT NULL,
  `diagnosisID` int(11) NOT NULL,
  `hospitalID` varchar(10) NOT NULL,
  `room_no` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `room`
--

CREATE TABLE `room` (
  `room_no` int(10) NOT NULL,
  `type` varchar(10) NOT NULL,
  `duration` varchar(10) NOT NULL,
  `wardID` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `staff`
--

CREATE TABLE `staff` (
  `staffID` int(11) NOT NULL,
  `name` int(11) NOT NULL,
  `wardID` varchar(255) NOT NULL,
  `gender` int(11) NOT NULL,
  `address` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-- --------------------------------------------------------

--
-- Table structure for table `ward`
--

CREATE TABLE `ward` (
  `wardID` varchar(255) NOT NULL,
  `name` varchar(255) NOT NULL,
  `capacity` int(11) NOT NULL,
  `hospitalID` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Indexes for dumped tables
--

--
-- Indexes for table `bill`
--
ALTER TABLE `bill`
  ADD PRIMARY KEY (`billID`),
  ADD UNIQUE KEY `patientID` (`patientID`);

--
-- Indexes for table `diagnosis`
--
ALTER TABLE `diagnosis`
  ADD PRIMARY KEY (`diagnosisID`);

--
-- Indexes for table `doctor`
--
ALTER TABLE `doctor`
  ADD PRIMARY KEY (`doctorID`),
  ADD KEY `hospitalID` (`hospitalID`);

--
-- Indexes for table `hospital`
--
ALTER TABLE `hospital`
  ADD PRIMARY KEY (`hospitalID`);

--
-- Indexes for table `medicine`
--
ALTER TABLE `medicine`
  ADD PRIMARY KEY (`medicine_serial`);

--
-- Indexes for table `nurse`
--
ALTER TABLE `nurse`
  ADD PRIMARY KEY (`nurseID`),
  ADD UNIQUE KEY `last` (`last`);

--
```

```
-- Indexes for table `patient`
--
ALTER TABLE `patient`
  ADD PRIMARY KEY (`patientID`),
  ADD KEY `doctorID` (`doctorID`),
  ADD KEY `diagnosisID` (`diagnosisID`);

--
-- Indexes for table `room`
--
ALTER TABLE `room`
  ADD PRIMARY KEY (`room_no`);

--
-- Indexes for table `staff`
--
ALTER TABLE `staff`
  ADD PRIMARY KEY (`staffID`),
  ADD KEY `wardID` (`wardID`);

--
-- Indexes for table `ward`
--
ALTER TABLE `ward`
  ADD PRIMARY KEY (`wardID`);

--
-- Constraints for dumped tables
--

--
-- Constraints for table `doctor`
--
ALTER TABLE `doctor`
  ADD CONSTRAINT `hospitalID` FOREIGN KEY (`hospitalID`) REFERENCES `hospital` (`hospitalID`);

--
-- Constraints for table `patient`
--
ALTER TABLE `patient`
  ADD CONSTRAINT `diagnosisID` FOREIGN KEY (`diagnosisID`) REFERENCES `diagnosis`
(`diagnosisID`),
  ADD CONSTRAINT `doctorID` FOREIGN KEY (`doctorID`) REFERENCES `doctor` (`doctorID`);

--
-- Constraints for table `staff`
--
ALTER TABLE `staff`
  ADD CONSTRAINT `wardID` FOREIGN KEY (`wardID`) REFERENCES `ward` (`wardID`);
```

OUTPUT:



Server: localhost » Database: hospital

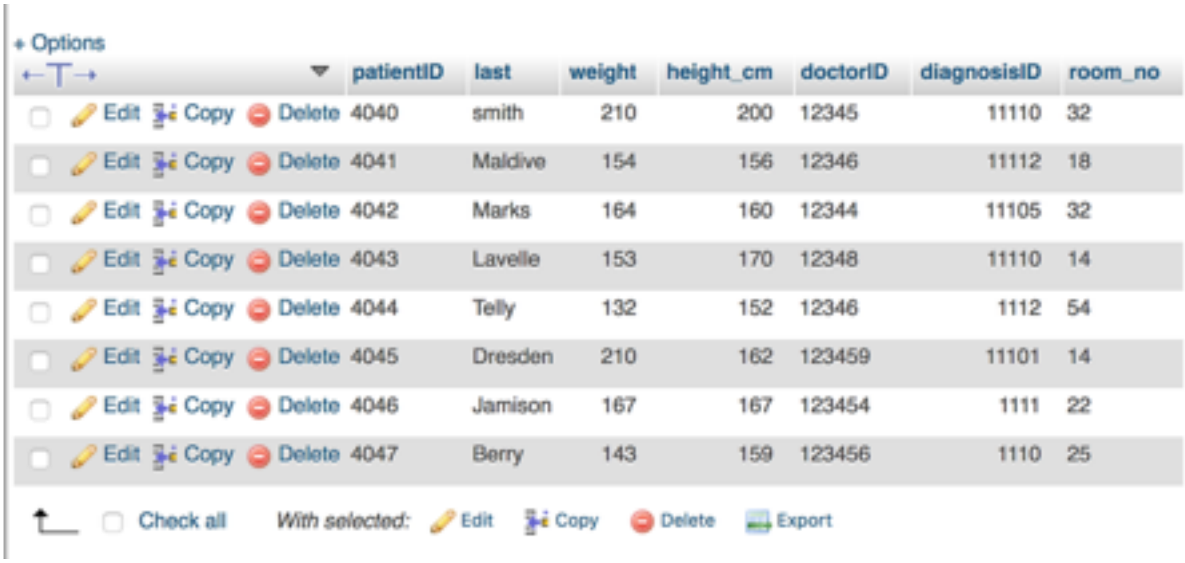
	Table	Action	Rows	Type	Collation	Size	Overhead
□	bill	Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	32 KiB	-
□	diagnosis	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16 KiB	-
□	doctor	Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	16 KiB	-
□	hospital	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16 KiB	-
□	medicine	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16 KiB	-
□	nurse	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	32 KiB	-
□	patient	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	48 KiB	-
□	room	Browse Structure Search Insert Empty Drop	15	InnoDB	latin1_swedish_ci	16 KiB	-
□	staff	Browse Structure Search Insert Empty Drop	15	InnoDB	latin1_swedish_ci	32 KiB	-
□	ward	Browse Structure Search Insert Empty Drop	15	InnoDB	latin1_swedish_ci	16 KiB	-
	10 tables	Sum	87	InnoDB	latin1_swedish_ci	240 KiB	0 B

> Select Statements <

1. **Select involving one/more conditions in Where clause**

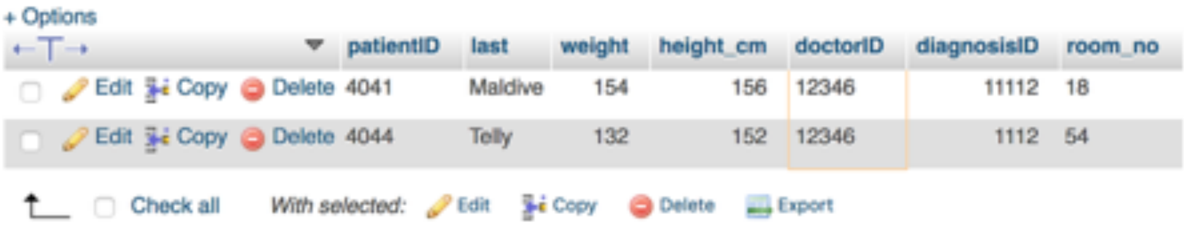
```
SELECT * FROM patient where doctorID = 12346
```

Before Query:



	patientID	last	weight	height_cm	doctorID	diagnosisID	room_no
□ Edit Copy Delete	4040	smith	210	200	12345	11110	32
□ Edit Copy Delete	4041	Maldive	154	156	12346	11112	18
□ Edit Copy Delete	4042	Marks	164	160	12344	11105	32
□ Edit Copy Delete	4043	Lavelle	153	170	12348	11110	14
□ Edit Copy Delete	4044	Telly	132	152	12346	1112	54
□ Edit Copy Delete	4045	Dresden	210	162	123459	11101	14
□ Edit Copy Delete	4046	Jamison	167	167	123454	1111	22
□ Edit Copy Delete	4047	Berry	143	159	123456	1110	25

After Query:

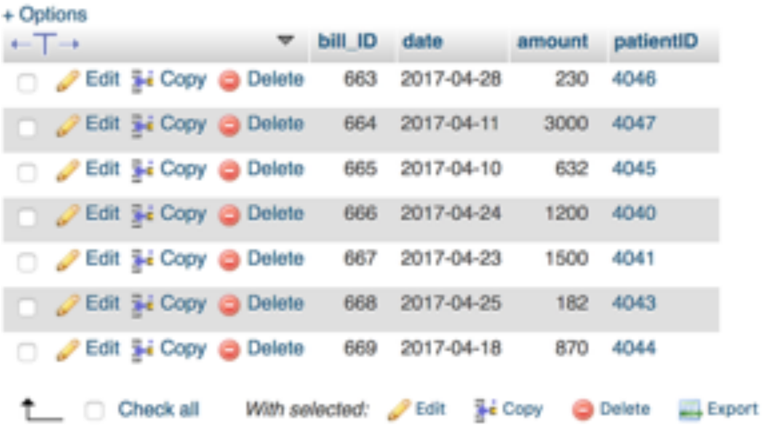


	patientID	last	weight	height_cm	doctorID	diagnosisID	room_no
□ Edit Copy Delete	4041	Maldive	154	156	12346	11112	18
□ Edit Copy Delete	4044	Telly	132	152	12346	1112	54

2. Select with aggregate functions (sum, min, max, avg, count) + **GROUP BY (#3)**

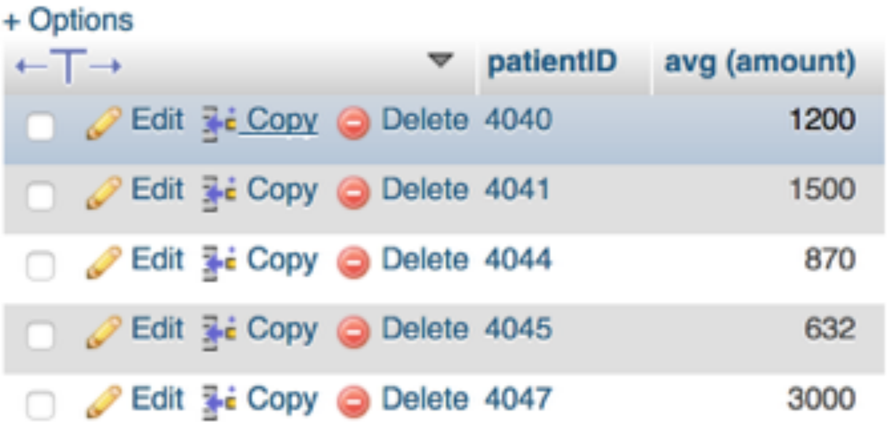
```
SELECT patientID, avg (amount) FROM bill  group by patientID

having avg(amount) > 500;
```

Before Query:



	bill_ID	date	amount	patientID
□ Edit Copy Delete	663	2017-04-28	230	4046
□ Edit Copy Delete	664	2017-04-11	3000	4047
□ Edit Copy Delete	665	2017-04-10	632	4045
□ Edit Copy Delete	666	2017-04-24	1200	4040
□ Edit Copy Delete	667	2017-04-23	1500	4041
□ Edit Copy Delete	668	2017-04-25	182	4043
□ Edit Copy Delete	669	2017-04-18	870	4044

After Query:



	patientID	avg (amount)
□ Edit Copy Delete	4040	1200
□ Edit Copy Delete	4041	1500
□ Edit Copy Delete	4044	870
□ Edit Copy Delete	4045	632
□ Edit Copy Delete	4047	3000

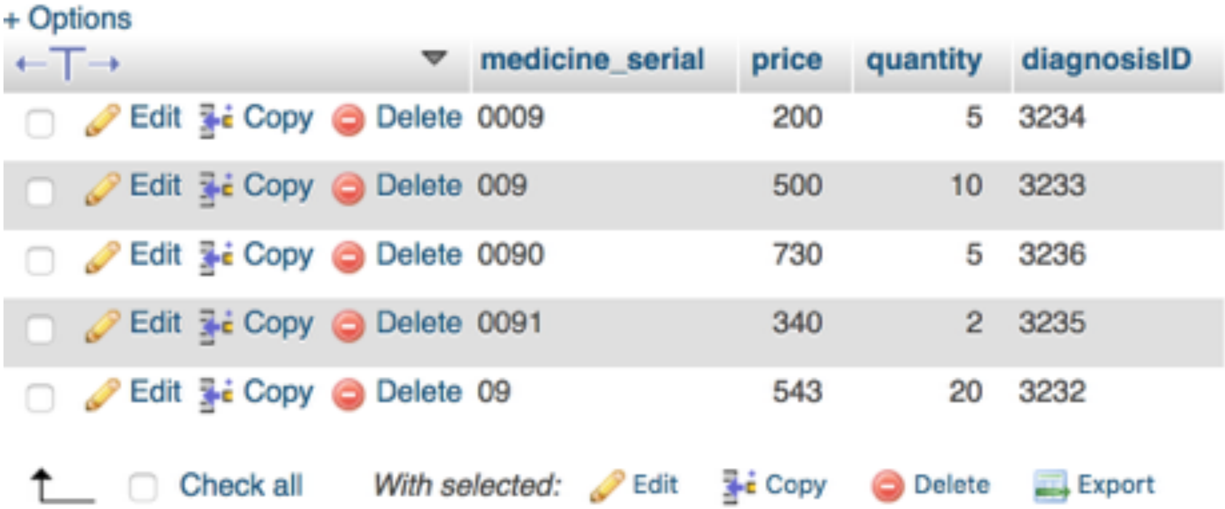
3. **Nested Select**

```
SELECT wardID FROM ward
  WHERE hospitalID='BBBH17'
(SELECT hospitalID FROM ward
 WHERE city='Brighton')
```

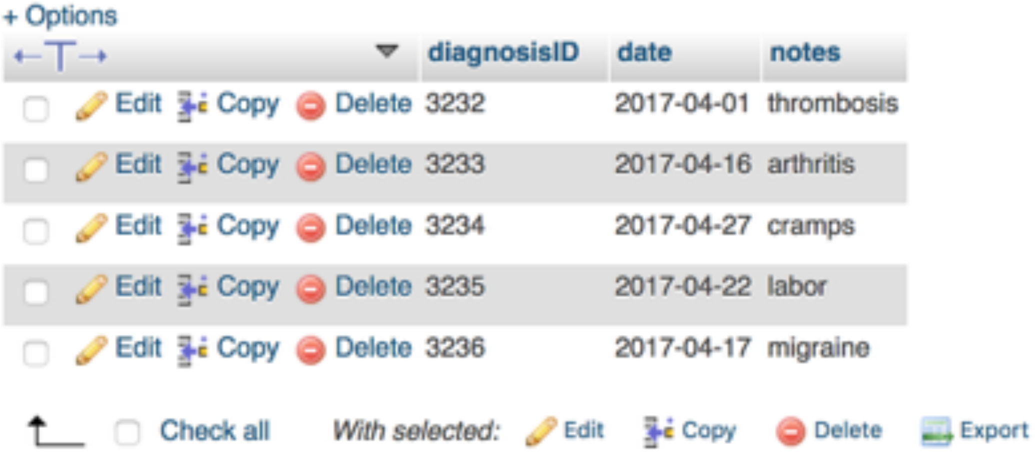
4. **Select involving the Union operation**

```
SELECT medicine.medicine_serial, diagnosis.diagnosisID, medicine.price
FROM medicine INNER JOIN  diagnosis ON
diagnosis.diagnosisID=medicine.diagnosisID;
```

Before Query:

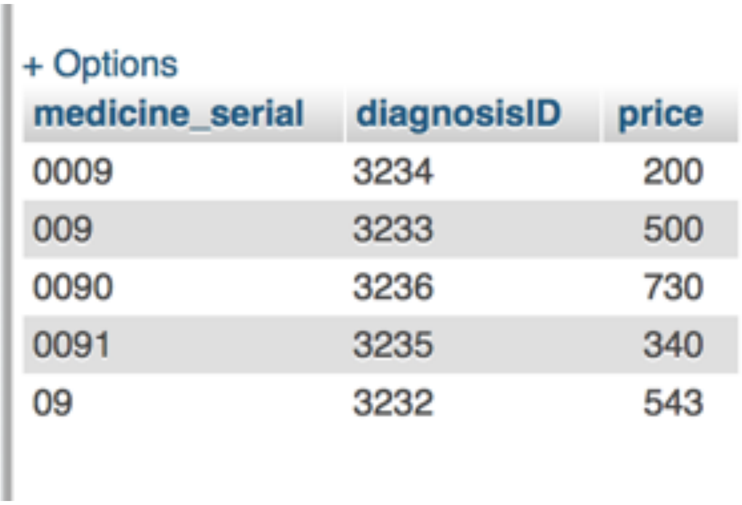


medicine_serial	price	quantity	diagnosisID
0009	200	5	3234
009	500	10	3233
0090	730	5	3236
0091	340	2	3235
09	543	20	3232



diagnosisID	date	notes
3232	2017-04-01	thrombosis
3233	2017-04-16	arthritis
3234	2017-04-27	cramps
3235	2017-04-22	labor
3236	2017-04-17	migraine

After Query:



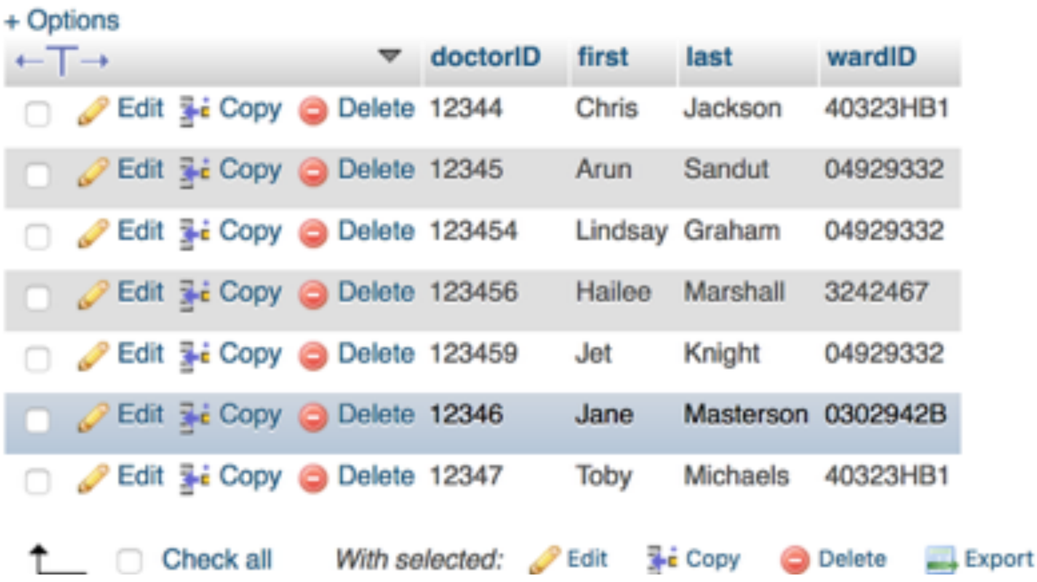
medicine_serial	diagnosisID	price
0009	3234	200
009	3233	500
0090	3236	730
0091	3235	340
09	3232	543

> Insert Statements <

1. Insert one tuple into a table

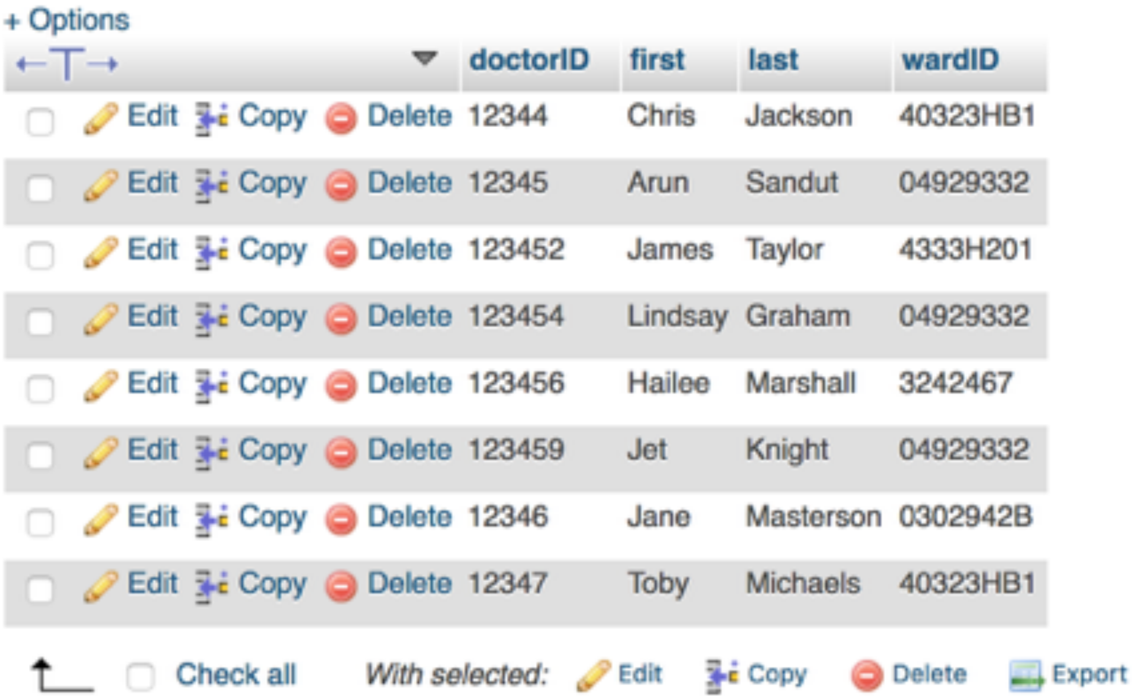
```
INSERT INTO doctor (doctorID, first, last, wardID)

VALUES (123452, James, Taylor, 4333H201);
```

Before Query:



	doctorID	first	last	wardID
□ Edit Copy Delete	12344	Chris	Jackson	40323HB1
□ Edit Copy Delete	12345	Arun	Sandut	04929332
□ Edit Copy Delete	123454	Lindsay	Graham	04929332
□ Edit Copy Delete	123456	Hailee	Marshall	3242467
□ Edit Copy Delete	123459	Jet	Knight	04929332
□ Edit Copy Delete	12346	Jane	Masterson	0302942B
□ Edit Copy Delete	12347	Toby	Michaels	40323HB1

After Query

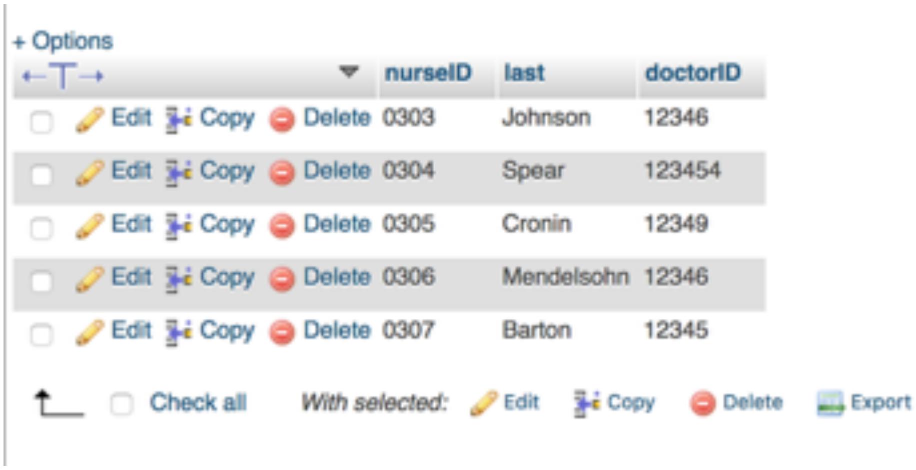


doctorID	first	last	wardID
12344	Chris	Jackson	40323HB1
12345	Arun	Sandut	04929332
123452	James	Taylor	4333H201
123454	Lindsay	Graham	04929332
123456	Hailee	Marshall	3242467
123459	Jet	Knight	04929332
12346	Jane	Masterson	0302942B
12347	Toby	Michaels	40323HB1

2. Insert a set of tuples

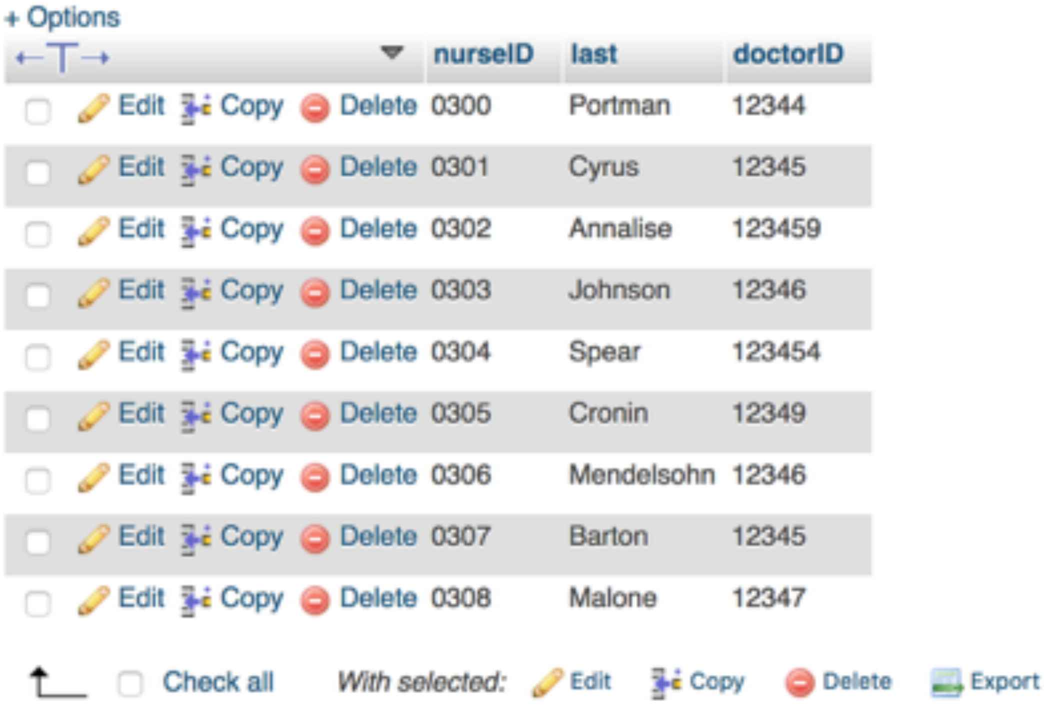
```
INSERT INTO nurse (nurseID, last, doctorID)
       VALUES (0300, Portman, 12344),
              (0301, Cyrus, 12345);
              (0302, Annalise, 123459);
              (0308, Malone, 12347);
```

Before Query:



nurseID	last	doctorID
0303	Johnson	12346
0304	Spear	123454
0305	Cronin	12349
0306	Mendelsohn	12346
0307	Barton	12345

After Query:



	nurseID	last	doctorID
□ Edit Copy Delete	0300	Portman	12344
□ Edit Copy Delete	0301	Cyrus	12345
□ Edit Copy Delete	0302	Annalise	123459
□ Edit Copy Delete	0303	Johnson	12346
□ Edit Copy Delete	0304	Spear	123454
□ Edit Copy Delete	0305	Cronin	12349
□ Edit Copy Delete	0306	Mendelsohn	12346
□ Edit Copy Delete	0307	Barton	12345
□ Edit Copy Delete	0308	Malone	12347

3. Insert involving two tables

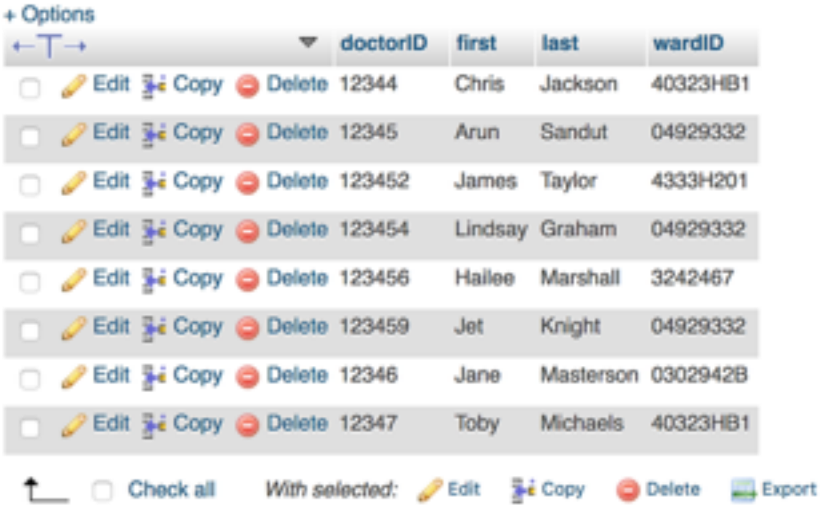
```
INSERT INTO bill ('last', 'weight', 'diagnosisID')

SELECT * FROM patient WHERE patientID = ‘4040';
```

> Delete Statements <

1. delete one tuple or a set of tuples from one table

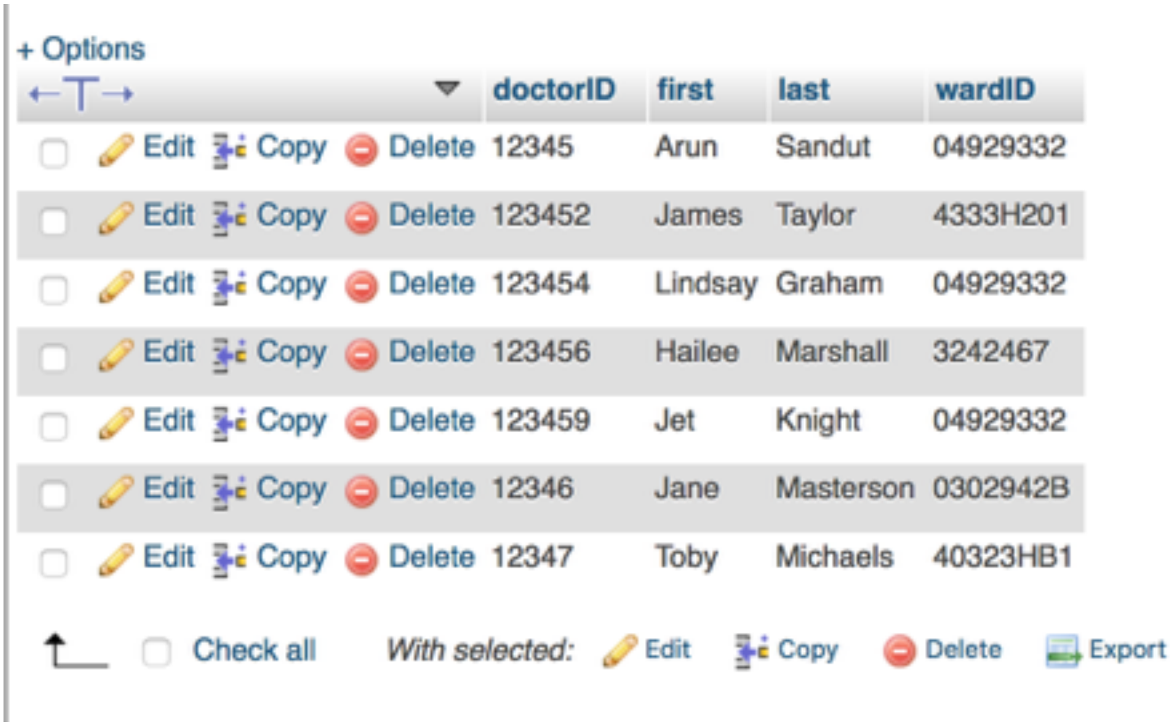
```
DELETE FROM doctor WHERE doctorID = ‘12344’
```

Before Query:



	doctorID	first	last	wardID
□ Edit Copy Delete	12344	Chris	Jackson	40323HB1
□ Edit Copy Delete	12345	Arun	Sandut	04929332
□ Edit Copy Delete	123452	James	Taylor	4333H201
□ Edit Copy Delete	123454	Lindsay	Graham	04929332
□ Edit Copy Delete	123456	Hailee	Marshall	3242467
□ Edit Copy Delete	123459	Jet	Knight	04929332
□ Edit Copy Delete	12346	Jane	Masterson	0302942B
□ Edit Copy Delete	12347	Toby	Michaels	40323HB1

After Query:



+ Options

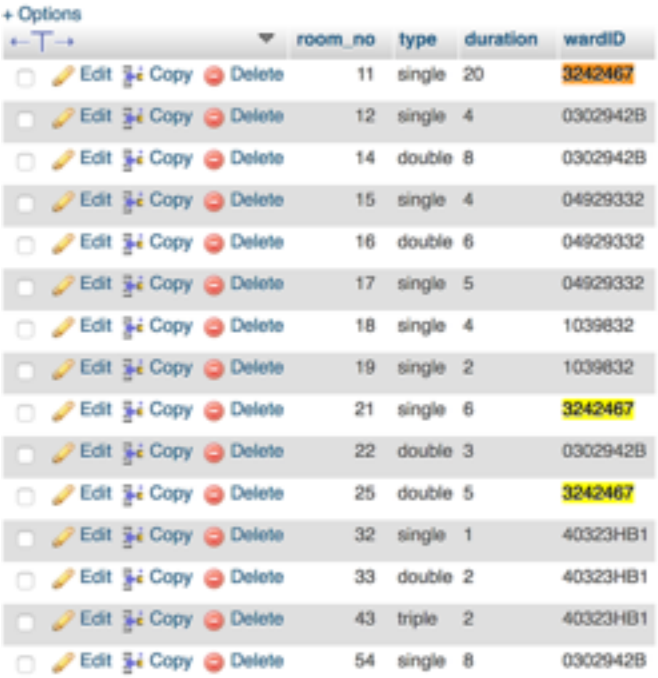
←T→	doctorID	first	last	wardID
□ Edit Copy Delete	12345	Arun	Sandut	04929332
□ Edit Copy Delete	123452	James	Taylor	4333H201
□ Edit Copy Delete	123454	Lindsay	Graham	04929332
□ Edit Copy Delete	123456	Hailee	Marshall	3242467
□ Edit Copy Delete	123459	Jet	Knight	04929332
□ Edit Copy Delete	12346	Jane	Masterson	0302942B
□ Edit Copy Delete	12347	Toby	Michaels	40323HB1

□ Check all *With selected:* Edit Copy Delete Export

2. Delete one tuple from multiple tables

```
 DELETE ward.*, room.*
   FROM ward
         INNER JOIN room ON ward.wardID = room.wardID
  WHERE (ward.wardID)='3242467';
```

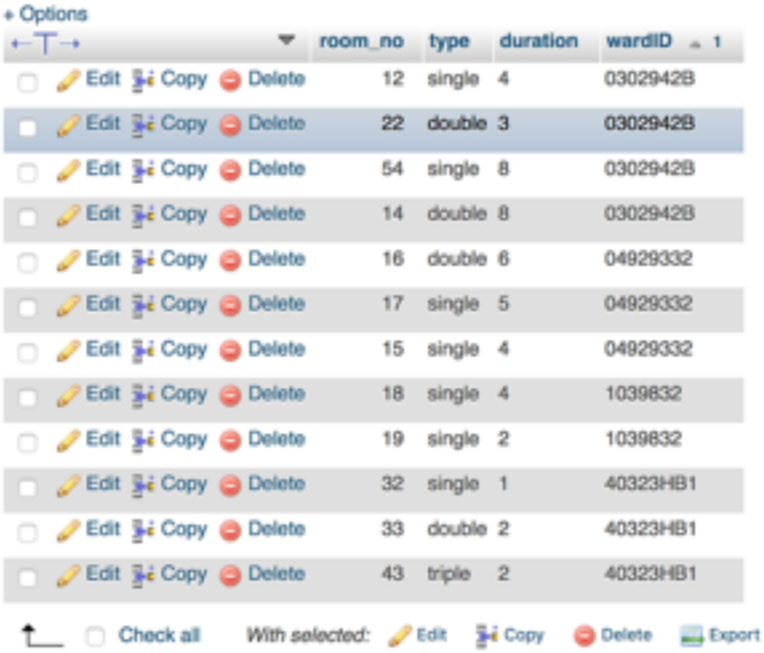
Before Query:



+ Options

←T→	room_no	type	duration	wardID
□ Edit Copy Delete	11	single	20	3242467
□ Edit Copy Delete	12	single	4	0302942B
□ Edit Copy Delete	14	double	8	0302942B
□ Edit Copy Delete	15	single	4	04929332
□ Edit Copy Delete	16	double	6	04929332
□ Edit Copy Delete	17	single	5	04929332
□ Edit Copy Delete	18	single	4	1039832
□ Edit Copy Delete	19	single	2	1039832
□ Edit Copy Delete	21	single	6	3242467
□ Edit Copy Delete	22	double	3	0302942B
□ Edit Copy Delete	25	double	5	3242467
□ Edit Copy Delete	32	single	1	40323HB1
□ Edit Copy Delete	33	double	2	40323HB1
□ Edit Copy Delete	43	triple	2	40323HB1
□ Edit Copy Delete	54	single	8	0302942B

After Query:



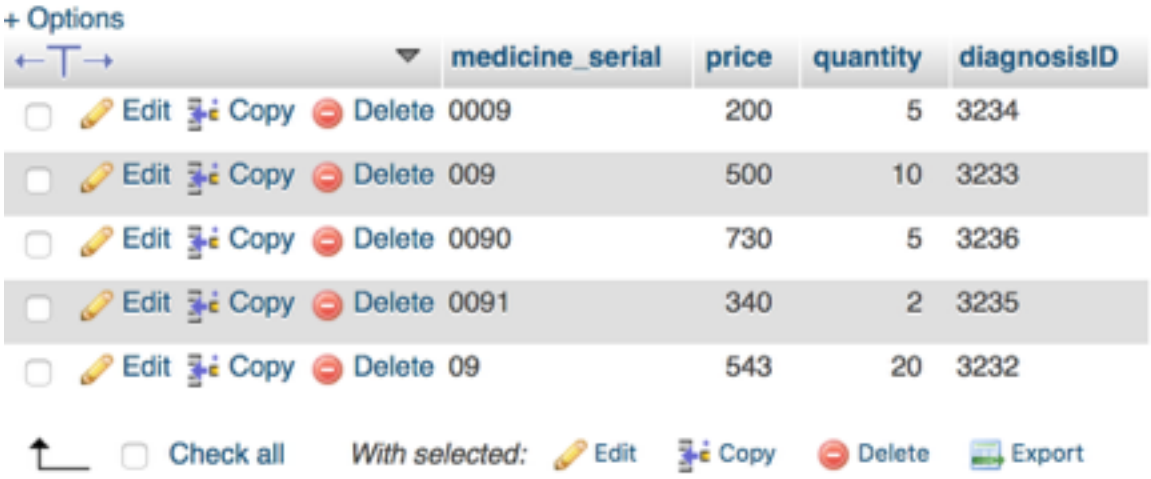
room_no	type	duration	wardID
12	single	4	03029428
22	double	3	03029428
54	single	8	03029428
14	double	8	03029428
16	double	6	04929332
17	single	5	04929332
15	single	4	04929332
18	single	4	1039832
19	single	2	1039832
32	single	1	40323HB1
33	double	2	40323HB1
43	triple	2	40323HB1

> Update Statements <

1. **Update One Tuple from One Table**

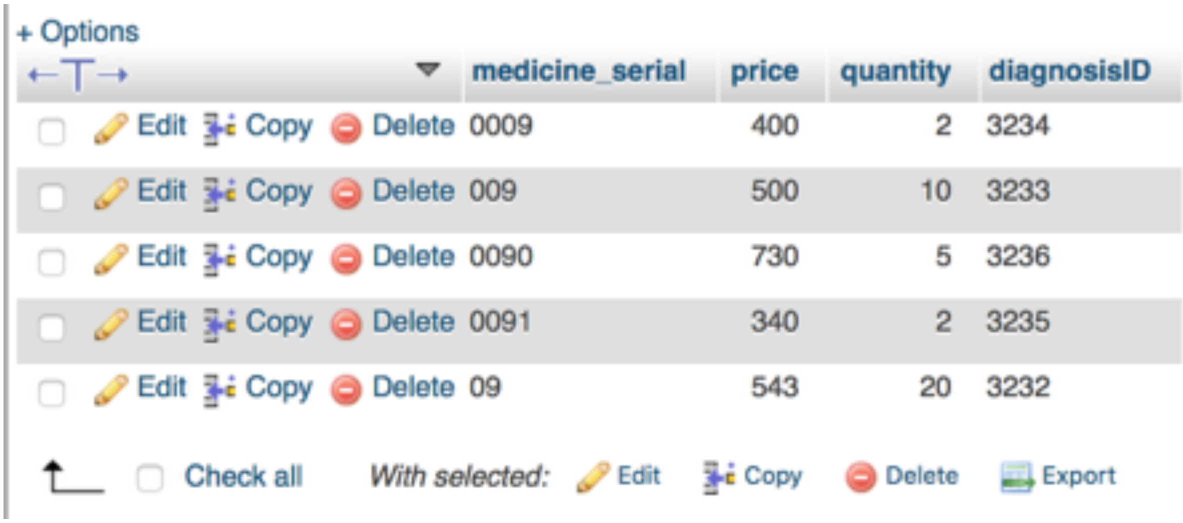
```
UPDATE medicine
SET price = '400', amount = '2',
WHERE medicine_serial = '0009';
```

Before Query:



medicine_serial	price	quantity	diagnosisID
0009	200	5	3234
009	500	10	3233
0090	730	5	3236
0091	340	2	3235
09	543	20	3232

After Query:



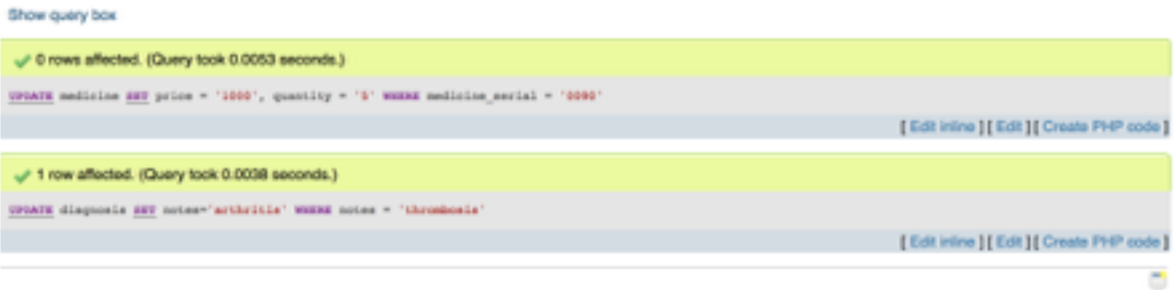
←T→	medicine_serial	price	quantity	diagnosisID
□ Edit Copy Delete	0009	400	2	3234
□ Edit Copy Delete	009	500	10	3233
□ Edit Copy Delete	0090	730	5	3236
□ Edit Copy Delete	0091	340	2	3235
□ Edit Copy Delete	09	543	20	3232

2. **Update more than 1 tuple**

UPDATE medicine **SET** price = ‘1000’, amount = ‘5’, **WHERE** medicine_serial = ‘0090’;

UPDATE diagnosis **SET** notes = ‘arthritis’, **WHERE** notes = ‘thrombosis’;

After Query:



✓ 0 rows affected. (Query took 0.0053 seconds.)

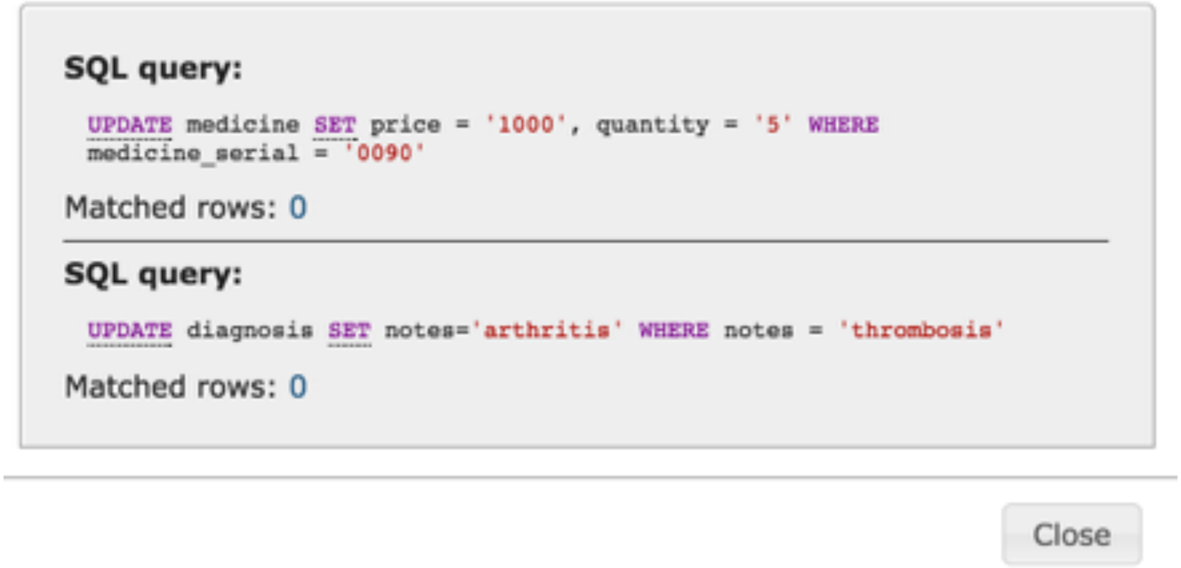
```
UPDATE medicine SET price = '1000', quantity = '5' WHERE medicine_serial = '0090'
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row affected. (Query took 0.0038 seconds.)

```
UPDATE diagnosis SET notes='arthritis' WHERE notes = 'thrombosis'
```

[Edit inline] [Edit] [Create PHP code]



SQL query:

```
UPDATE medicine SET price = '1000', quantity = '5' WHERE
medicine_serial = '0090'
```

Matched rows: 0

SQL query:

```
UPDATE diagnosis SET notes='arthritis' WHERE notes = 'thrombosis'
```

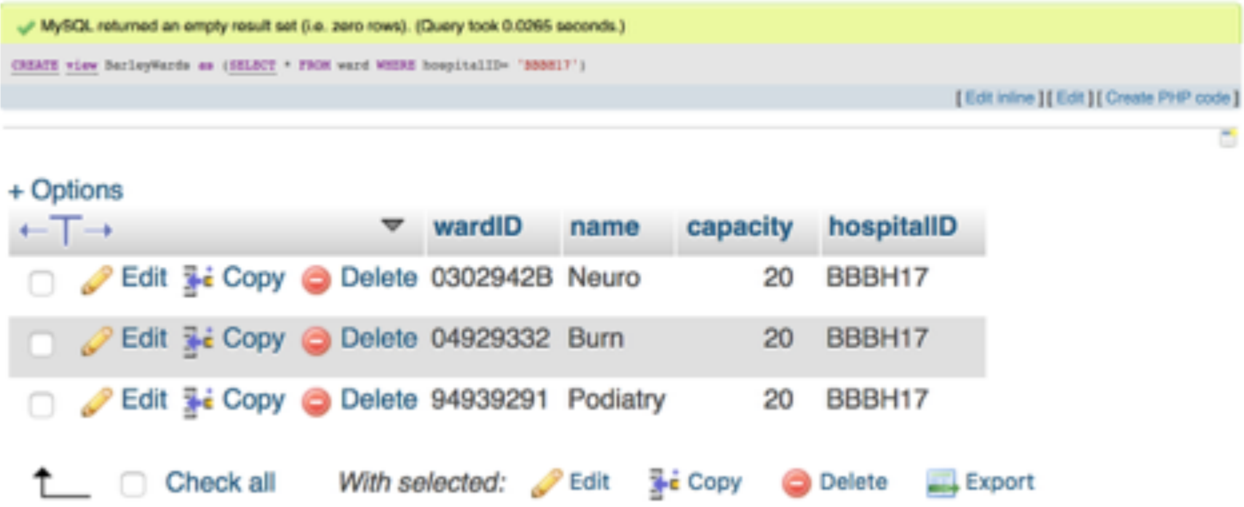
Matched rows: 0

> Create View <

1. Create View

CREATE view BarleyWards as (**SELECT** * **FROM** ward **WHERE** hospital= ‘BBBH17’);

After Query:



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0265 seconds.)

```
CREATE view BarleyWards as (SELECT * FROM ward WHERE hospitalID= 'BBBH17')
```

[Edit inline] [Edit] [Create PHP code]

+ Options

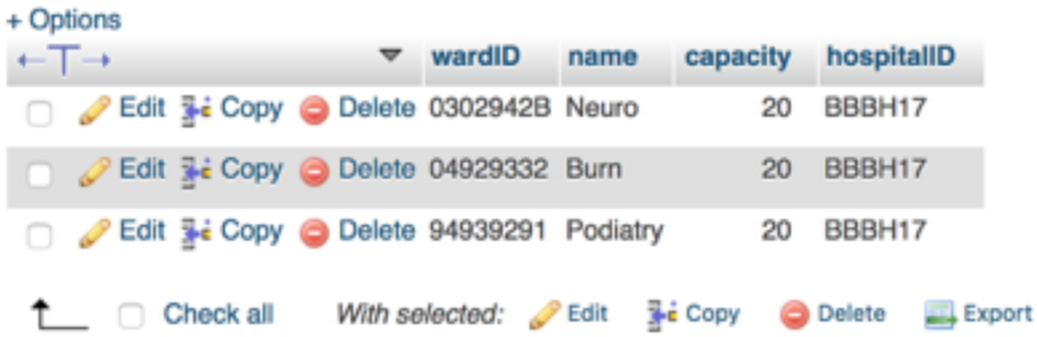
←T→	wardID	name	capacity	hospitalID
□ Edit Copy Delete	0302942B	Neuro	20	BBBH17
□ Edit Copy Delete	04929332	Burn	20	BBBH17
□ Edit Copy Delete	94939291	Podiatry	20	BBBH17

□ Check all *With selected:* Edit Copy Delete Export

2. Insert From Barley

INSERT into barleywards values (‘0302934S’, ‘ICU’, ‘BBBH17’);

Before Query:

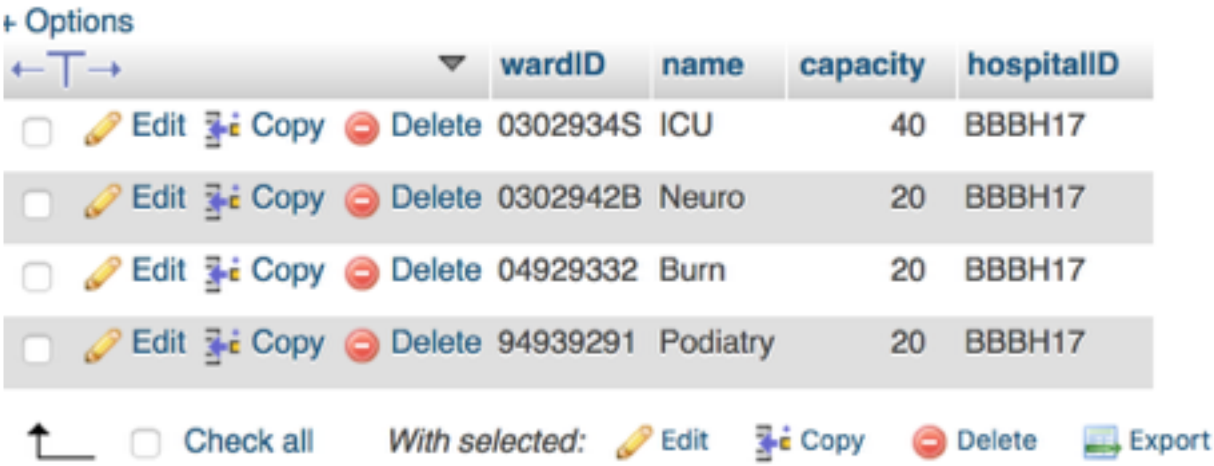


+ Options

←T→	wardID	name	capacity	hospitalID
□ Edit Copy Delete	0302942B	Neuro	20	BBBH17
□ Edit Copy Delete	04929332	Burn	20	BBBH17
□ Edit Copy Delete	94939291	Podiatry	20	BBBH17

□ Check all *With selected:* Edit Copy Delete Export

After Query:

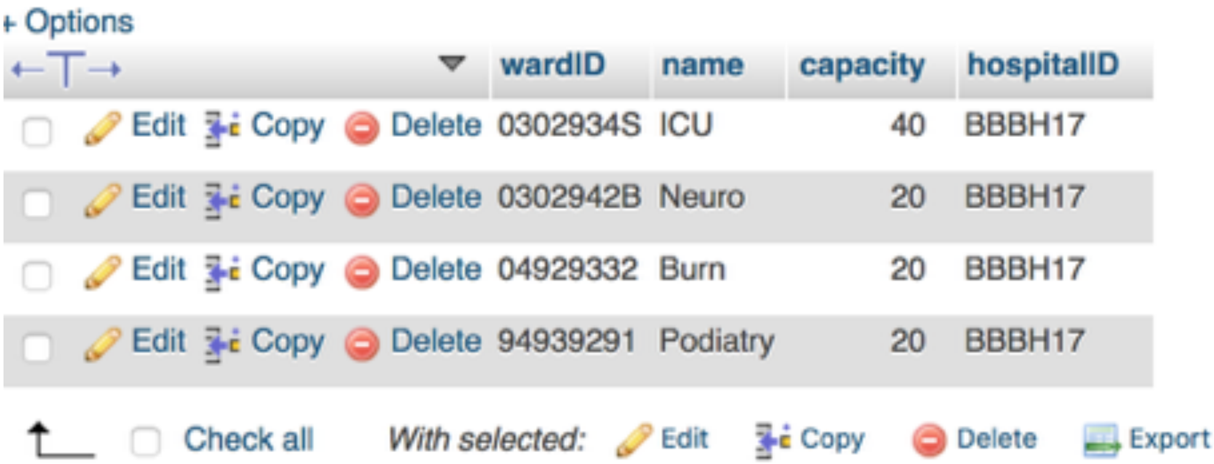


wardID	name	capacity	hospitalID
0302934S	ICU	40	BBBH17
0302942B	Neuro	20	BBBH17
04929332	Burn	20	BBBH17
94939291	Podiatry	20	BBBH17

```
3. Update barley wards
```

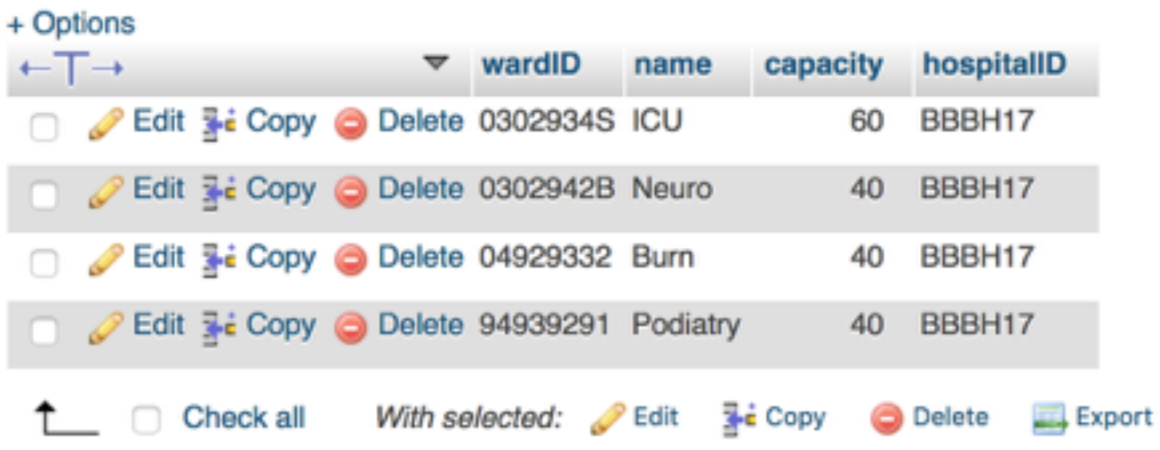
```
UPDATE barley ward set salary = capacity+20;
```

Before Query:



wardID	name	capacity	hospitalID
0302934S	ICU	40	BBBH17
0302942B	Neuro	20	BBBH17
04929332	Burn	20	BBBH17
94939291	Podiatry	20	BBBH17

After Query:



	wardID	name	capacity	hospitalID
□ Edit Copy Delete	0302934S	ICU	60	BBBH17
□ Edit Copy Delete	0302942B	Neuro	40	BBBH17
□ Edit Copy Delete	04929332	Burn	40	BBBH17
□ Edit Copy Delete	94939291	Podiatry	40	BBBH17

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

Trigger meant to prevent overdose of medication:

```
create trigger medicine_dosage_max
before update of medicine
for each row
      when (new.quantity > 10 )
            set new.quantity = 10;
```

Trigger meant to ensure rooms aren't being oversaturated/overpopulated past capacity:

```
create trigger room_max
before update of room
      SELECT capacity, COUNT (distinct room_no)
      FROM room, patient
      WHERE room.room_no = patient.room_no
      group by room_no

      DELETE room FROM patient WHERE count > 3;
```

Trigger to enforce referential integrity:

```
create trigger
for insert_trigger
on ward
for insert as
if (select
count(*)
    from ward, hospital
    where ward.hospitalID =
inserted.hospitalID) !=
    @@rowcount

/* Cancel the insert and print a message.*/
  begin
    rollback transaction
    print
"No, this ward does not exist in
    any valid hospital."
  end
/*
Otherwise, allow it. */
else
  print "Added! This ward does exist in the aforementioned hospital."
```

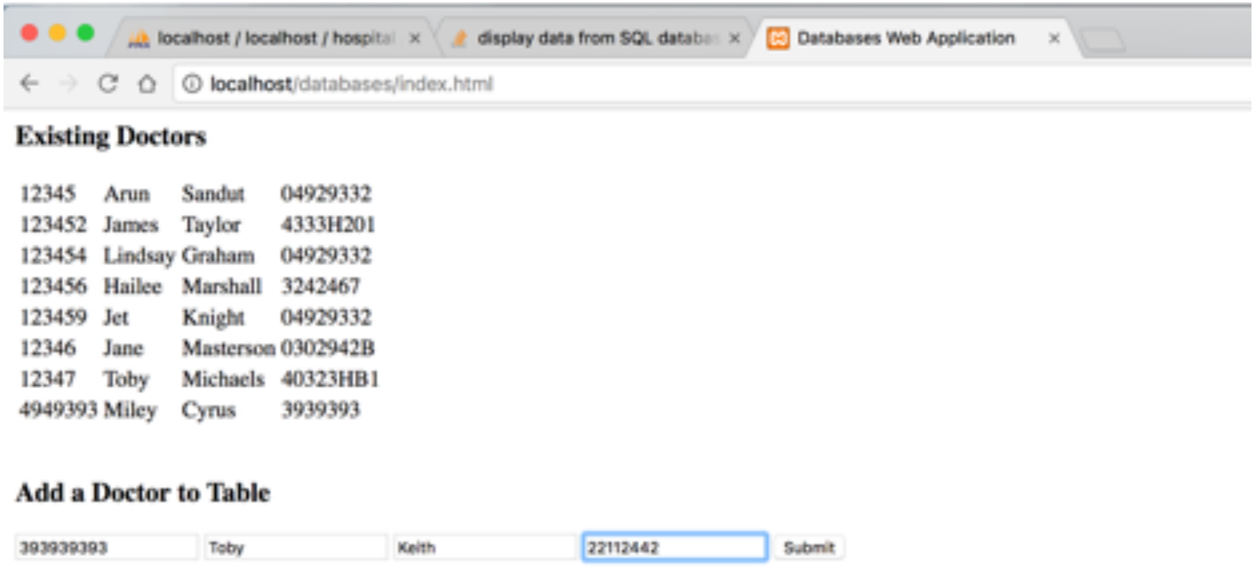
<u>*Trigger Meant to Prevent Duplicate Entries:*</u>

```
CREATE TRIGGER dupe_check

      AFTER UPDATE ON patient

      FOR EACH ROW

      BEGIN

       IF EXISTS (SELECT last FROM patient WHERE last=patient.last)

           DELETE * FROM patient;
```

C) Use MySQL and ColdFusion to create a Web-based application to enable the user to do the following operations:

<u>> ADDING A RECORD <</u>

<u>SCREENSHOT BEFORE INSERT:</u>


localhost / localhost / hospital
display data from SQL databa
Databases Web Application
localhost/databases/index.html

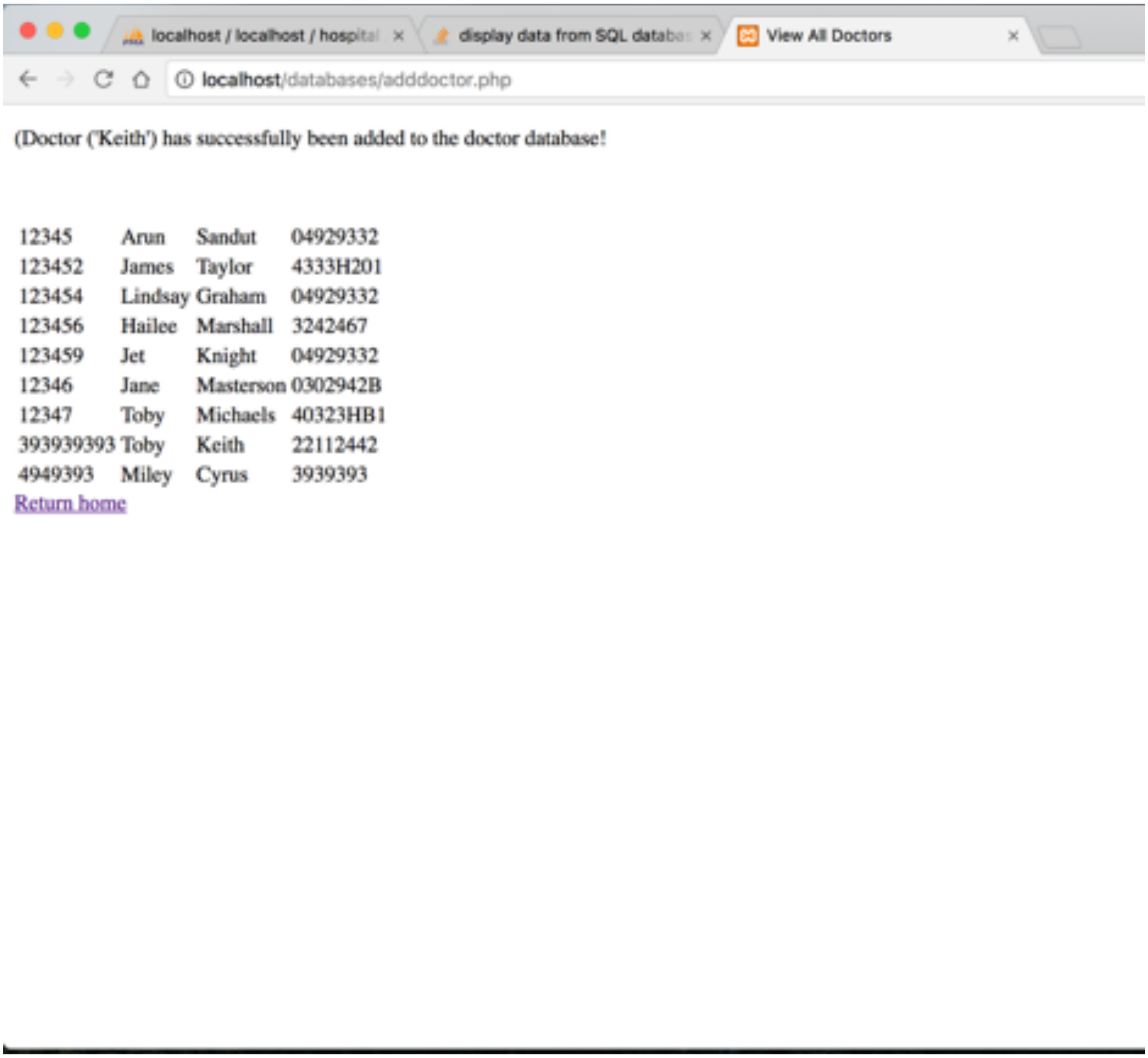
Existing Doctors

12345	Arun	Sandut	04929332
123452	James	Taylor	4333H201
123454	Lindsay	Graham	04929332
123456	Hailee	Marshall	3242467
123459	Jet	Knight	04929332
12346	Jane	Masterson	0302942B
12347	Toby	Michaels	40323HB1
4949393	Miley	Cyrus	3939393

Add a Doctor to Table

393939393 | Toby | Keith | 22112442 | Submit

SCREENSHOT AFTER INSERT:


localhost / localhost / hospital
display data from SQL databa
View All Doctors
localhost/databases/adddoctor.php

(Doctor ('Keith') has successfully been added to the doctor database!

12345	Arun	Sandut	04929332
123452	James	Taylor	4333H201
123454	Lindsay	Graham	04929332
123456	Hailee	Marshall	3242467
123459	Jet	Knight	04929332
12346	Jane	Masterson	0302942B
12347	Toby	Michaels	40323HB1
393939393	Toby	Keith	22112442
4949393	Miley	Cyrus	3939393

Return home

HTML CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Databases Web Application</title>
```

```
</head>

<body>

  <big> <b>Existing Doctors</b></big>

<br><br>

  <div id="usersDiv"></div>

  <script>

    //Set up a HTTP request to users.php, it will generate a list of all
entries

    //from the Users table in the ClassDemo database

    //NOTE: The code below uses AJAX to make a HTTP GET request to a php file

    var xmlhttp = new XMLHttpRequest();

    xmlhttp.onreadystatechange = function() {

      if(this.readyState == 4 && this.status == 200) {

        //responseText is generated by nouns.php

        document.getElementById("usersDiv").innerHTML = this.responseText;

      }

    }

    xmlhttp.open("GET", "viewall.php");

    xmlhttp.send();

  </script>

<br><br>


  <!-- Add a record


  Delete a record


  Update a record


  Query (at least 3 select statements on one relation) -->
  <big> <b>Add a Doctor to Table</b></big>
  <br><br>
  <form action="adddoctor.php" method="post">
    <input name="doctorID" type="text" placeholder="Doctor ID">
    <input name="first" type="text" placeholder="First">
    <input name="last" type="text" placeholder="Last">
    <input name="wardID" type="text" placeholder="Ward ID">
    <input type="submit"></input>

  </form>
  <table></table>
  <br><br>

</body>
</html>
```

PHP CODE:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>View All Doctors</title>
  </head>
  <body>
    <p>
    <?
      $host = "localhost";
      $user = "root";
      $password = "";


      $mysqli = new mysqli($host, $user, $password, "hospital");


      if ($mysqli->connect_errno) {
        die("Error: " . $mysqli->connect_errno);
      }
      $doctorID = $_POST["doctorID"];
      $first = $_POST["first"];
      $last = $_POST["last"];
      $wardID = $_POST["wardID"];


      $sql = "INSERT INTO doctor (doctorID, first, last, wardID) VALUES
('$doctorID','$first', '$last', '$wardID')";


      $mysqli->query($sql);


      echo "(Doctor ('$last') has successfully been added to the doctor
database!";
    ?>
    </p>
    <br><br>
    <div id="usersDiv"></div>
    <script>
      //Set up a HTTP request to users.php, it will generate a list of all
entries
      //from the Users table in the ClassDemo database
```

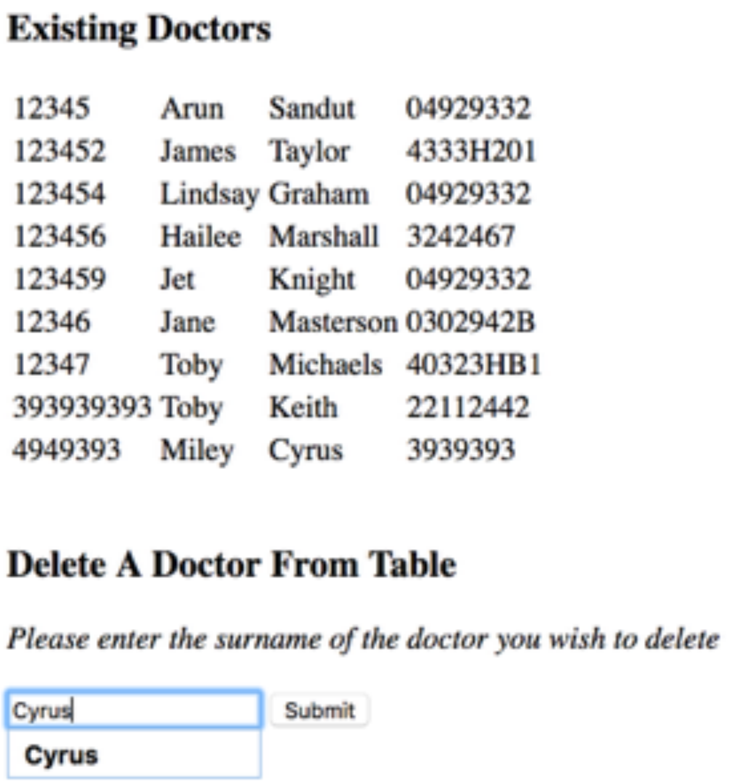
```
   //NOTE: The code below uses AJAX to make a HTTP GET request to a php file
   var xmlhttp = new XMLHttpRequest();
   xmlhttp.onreadystatechange = function() {
    if(this.readyState == 4 && this.status == 200) {
     //responseText is generated by nouns.php
     document.getElementById("usersDiv").innerHTML = this.responseText;
    }
   }
   xmlhttp.open("GET", "viewall.php");
   xmlhttp.send();
  </script>

 <a href="index.html">Return home</a>
 </body>
</html>
```

<u>VIEW ALL SCRIPT CODE:</u>

```
<?php
$connection = mysql_connect('localhost', 'root', ''); //The Blank string is the password
mysql_select_db('hospital');

$query = "SELECT * FROM doctor"; //You don't need a ; like you do in SQL
$result = mysql_query($query);

echo "<table>"; // start a table tag in the HTML

while($row = mysql_fetch_array($result)){   //Creates a loop to loop through results
echo "<tr><td>" . $row['doctorID'] . "</td><td>" . $row['first'] . "</
td><td>" . $row['last'] . "</td><td>" . $row['wardID'] . "</td></tr>";  //
$row['index'] the index here is a field name

}



echo "</table>"; //Close the table in HTML



mysql_close(); //Make sure to close out the database connection



?>
```

> DELETING A RECORD <

SCREENSHOT BEFORE DELETE:



Existing Doctors

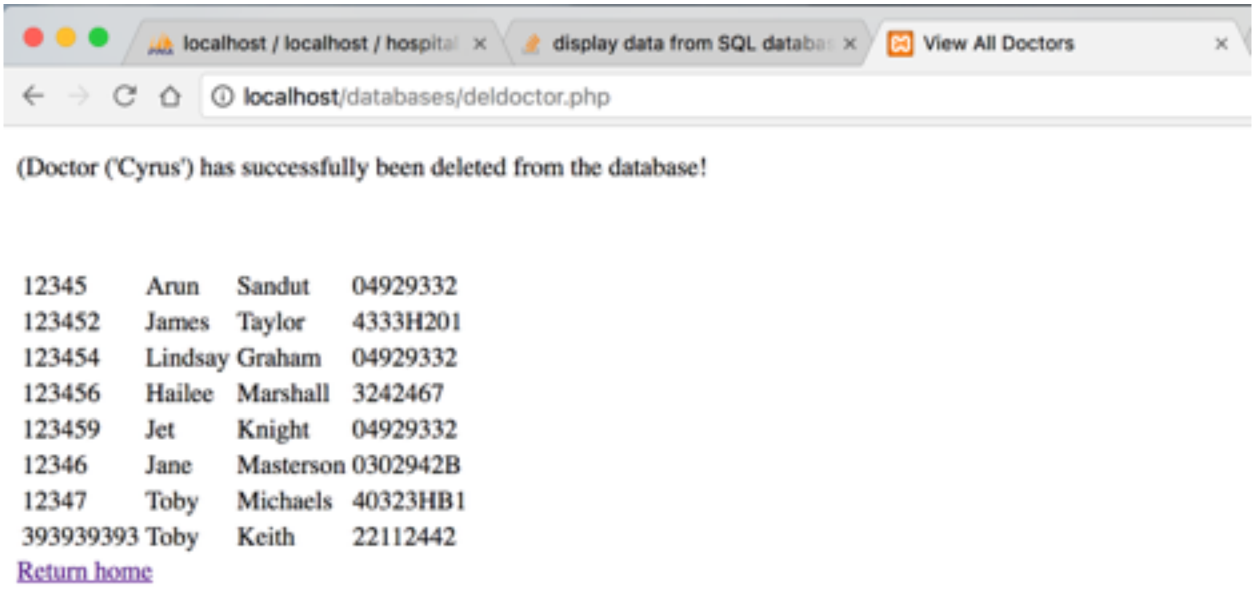
12345	Arun	Sandut	04929332
123452	James	Taylor	4333H201
123454	Lindsay	Graham	04929332
123456	Hailee	Marshall	3242467
123459	Jet	Knight	04929332
12346	Jane	Masterson	0302942B
12347	Toby	Michaels	40323HB1
393939393	Toby	Keith	22112442
4949393	Miley	Cyrus	3939393

Delete A Doctor From Table

Please enter the surname of the doctor you wish to delete

Cyrus

SCREENSHOT AFTER DELETE:


localhost / localhost / hospital
display data from SQL databa
View All Doctors
localhost/databases/deldoctor.php

(Doctor ('Cyrus') has successfully been deleted from the database!

12345	Arun	Sandut	04929332
123452	James	Taylor	4333H201
123454	Lindsay	Graham	04929332
123456	Hailee	Marshall	3242467
123459	Jet	Knight	04929332
12346	Jane	Masterson	0302942B
12347	Toby	Michaels	40323HB1
393939393	Toby	Keith	22112442

Return home

HTML CODE:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Databases Web Application</title>

</head>

<body>

  <big> <b>Existing Doctors</b></big>

<br><br>

  <div id="usersDiv"></div>

  <script>
```

```
    //Set up a HTTP request to users.php, it will generate a list of all
entries
    //from the Users table in the ClassDemo database
    //NOTE: The code below uses AJAX to make a HTTP GET request to a php file
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
      if(this.readyState == 4 && this.status == 200) {
        //responseText is generated by nouns.php
        document.getElementById("usersDiv").innerHTML = this.responseText;
      }
    }
    xmlhttp.open("GET", "viewall.php");
    xmlhttp.send();
  </script>
<br><br>


  <!-- Add a record

  Delete a record

  Update a record

  Query (at least 3 select statements on one relation) -->


  <big> <b>Delete A Doctor From Table</b></big><br>
  <br><i>Please enter the surname of the doctor you wish to delete </i>
  <br><br>
  <form action="deldoctor.php" method="post">
    <input name="last" type="text" placeholder="Last">
    <input type="submit"></input>

  </form>
  <table></table>
  <br><br>

</body>
</html>
```

PHP CODE (deldoctor.php):

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>View All Doctors</title>
  </head>
  <body>
    <p>
    <?
      $host = "localhost";
      $user = "root";
      $password = "";

      $mysqli = new mysqli($host, $user, $password, "hospital");

      if ($mysqli->connect_errno) {
        die("Error: " . $mysqli->connect_errno);
      }
      $last = $_POST["last"];

      $sql = "DELETE FROM doctor WHERE last = '($last)'";

      $mysqli->query($sql);

      echo "(Doctor ('$last') has successfully been deleted from the
database!";
    ?>
    </p>
    <br><br>
    <div id="usersDiv"></div>
    <script>
      //Set up a HTTP request to users.php, it will generate a list of all
entries
      //from the Users table in the ClassDemo database
      //NOTE: The code below uses AJAX to make a HTTP GET request to a php
file
      var xmlhttp = new XMLHttpRequest();
      xmlhttp.onreadystatechange = function() {
        if(this.readyState == 4 && this.status == 200) {
          //responseText is generated by nouns.php
          document.getElementById("usersDiv").innerHTML = this.responseText;
        }
      }
      xmlhttp.open("GET", "viewall.php");
      xmlhttp.send();
    </script>
```

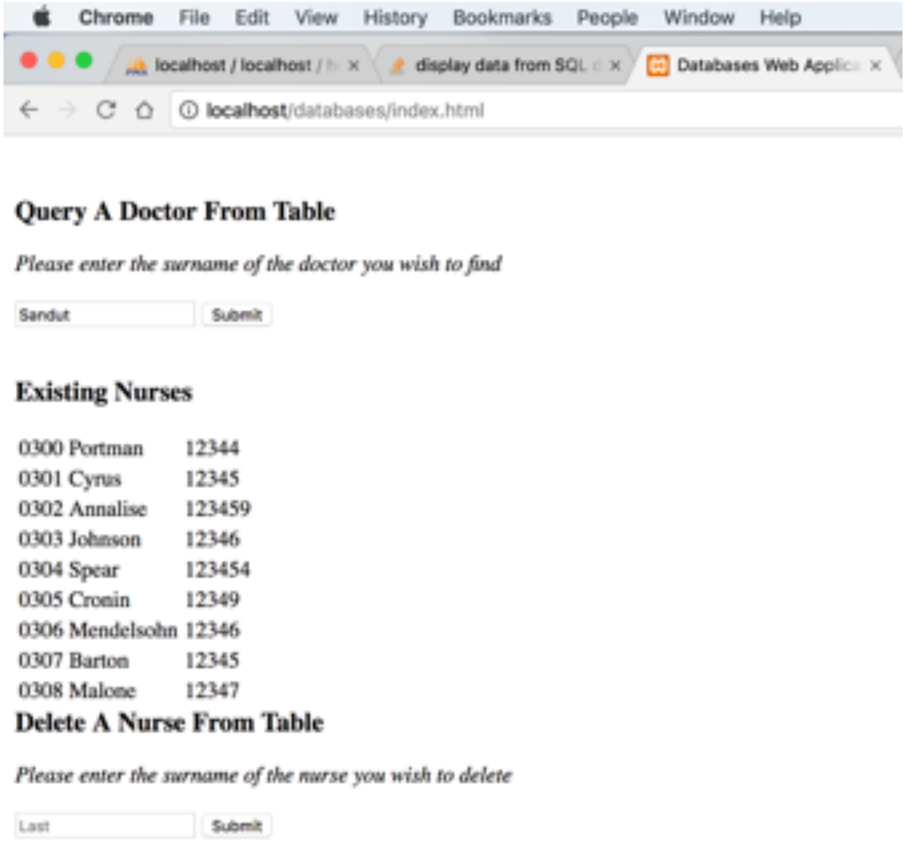
```
  <a href="index.html">Return home</a>

  </body>

</html>
```

PHP VIEW ALL SCRIPT:

```
<?php

$connection = mysql_connect('localhost', 'root', ''); //The Blank string is
the password

mysql_select_db('hospital');



$query = "SELECT * FROM doctor"; //You don't need a ; like you do in SQL

$result = mysql_query($query);



echo "<table>"; // start a table tag in the HTML



while($row = mysql_fetch_array($result)){   //Creates a loop to loop through
results

echo "<tr><td>" . $row['doctorID'] . "</td><td>" . $row['first'] . "</
td><td>" . $row['last'] . "</td><td>" . $row['wardID'] . "</td></tr>";  //
$row['index'] the index here is a field name

}



echo "</table>"; //Close the table in HTML



mysql_close(); //Make sure to close out the database connection



?>
```

> SELECTING A RECORD <

SCREENSHOT BEFORE QUERY #1:



Query A Doctor From Table

Please enter the surname of the doctor you wish to find

Sandut Submit

Existing Nurses

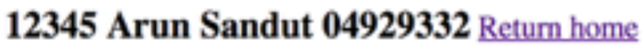
0300 Portman	12344
0301 Cyrus	12345
0302 Annalise	123459
0303 Johnson	12346
0304 Spear	123454
0305 Cronin	12349
0306 Mendelsohn	12346
0307 Barton	12345
0308 Malone	12347

Delete A Nurse From Table

Please enter the surname of the nurse you wish to delete

Last Submit

SCREENSHOT AFTER QUERY #1



12345 Arun Sandut 04929332 Return home

HTML:

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Databases Web Application</title>
</head>
<body>
  <!-- <big> <b>Existing Doctors</b></big>
<br><br>
  <div id="usersDiv"></div> -->
  <!-- <script>
    //Set up a HTTP request to users.php, it will generate a list of all
entries
    //from the Users table in the ClassDemo database
    //NOTE: The code below uses AJAX to make a HTTP GET request to a php file
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
      if(this.readyState == 4 && this.status == 200) {
        //responseText is generated by nouns.php
        document.getElementById("usersDiv").innerHTML = this.responseText;
      }
    }
    xmlhttp.open("GET", "viewalldoctors.php");
    xmlhttp.send();
  </script> -->
<br><br>

  <!-- Add a record
  Delete a record

  Update a record

  Query (at least 3 select statements on one relation) -->


  <big> <b>Query A Doctor From Table</b></big><br>
  <br><i>Please enter the surname of the doctor you wish to find </i>
  <br><br>
  <form action="deldoctor.php" method="post">
    <input name="last" type="text" placeholder="Last">
    <input type="submit"></input>

  </form>
  <table></table>
  <br><br>

  <big> <b>Existing Nurses</b></big>
<br><br>
  <div id="nursediv"></div>
  <script>
    //Set up a HTTP request to users.php, it will generate a list of all
entries
    //from the Users table in the ClassDemo database
    //NOTE: The code below uses AJAX to make a HTTP GET request to a php file
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
      if(this.readyState == 4 && this.status == 200) {
        //responseText is generated by nouns.php
        document.getElementById("nursediv").innerHTML = this.responseText;
      }
    }
    xmlhttp.open("GET", "viewallnurse.php");
    xmlhttp.send();
  </script>
  <big> <b>Delete A Nurse From Table</b></big><br>
  <br><i>Please enter the surname of the nurse you wish to delete </i>
  <br><br>
  <form action="delnurse.php" method="post">
    <input name="nlast" type="text" placeholder="Last">
    <input type="submit"></input>

  </form>
  <table></table>
  <br><br>

</body>
</html>
```

PHP:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>View All Doctors</title>
  </head>
  <body>
```

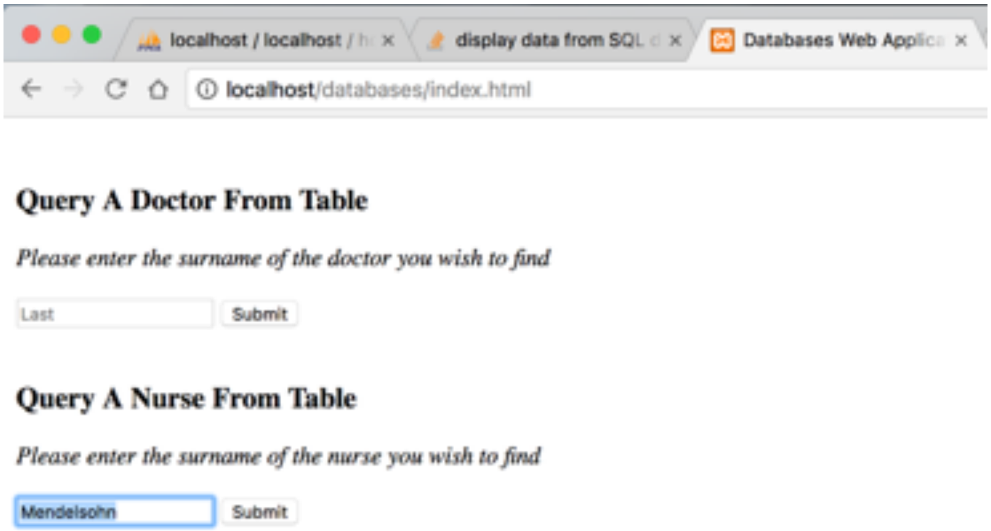
```
    <p>
    <?
      $host = "localhost";
      $user = "root";
      $password = "";

      $mysqli = new mysqli($host, $user, $password, "hospital");

      if ($mysqli->connect_errno) {
        die("Error: " . $mysqli->connect_errno);
      }
      $last = $_POST["last"];
      $sql="SELECT * FROM doctor WHERE last = '$last'";
      $result= $mysqli->query($sql);
    ?>
    </p>
  <a href="index.html">Return home</a>
  </body>
</html>
```

2nd PHP CODE:

```
<?php
$connection = mysql_connect('localhost', 'root', ''); //The Blank string is the password
mysql_select_db('hospital');


$query = "SELECT * FROM doctor"; //You don't need a ; like you do in SQL
$result = mysql_query($query);
```

```
echo "<table>"; // start a table tag in the HTML


while($row = mysql_fetch_array($result)){   //Creates a loop to loop through
results

echo "<tr><td>" . $row['doctorID'] . "</td><td>" . $row['first'] . "</
td><td>" . $row['last'] . "</td><td>" . $row['wardID'] . "</td></tr>";  //
$row['index'] the index here is a field name

}


echo "</table>"; //Close the table in HTML


mysql_close(); //Make sure to close out the database connection


?>
```

SCREENSHOT BEFORE QUERY #2



Query A Doctor From Table

Please enter the surname of the doctor you wish to find

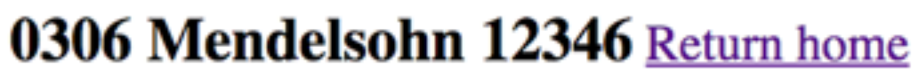
Last

Query A Nurse From Table

Please enter the surname of the nurse you wish to find

Mendelsohn

SCREENSHOT AFTER QUERY #2



0306 Mendelsohn 12346 Return home

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Databases Web Application</title>
</head>
<body>
  <!-- <big> <b>Existing Doctors</b></big>
<br><br>
  <div id="usersDiv"></div> -->
  <!-- <script>
    //Set up a HTTP request to users.php, it will generate a list of all entries
    //from the Users table in the ClassDemo database
    //NOTE: The code below uses AJAX to make a HTTP GET request to a php file
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
      if(this.readyState == 4 && this.status == 200) {
        //responseText is generated by nouns.php
        document.getElementById("usersDiv").innerHTML = this.responseText;

      }

    }

    xmlhttp.open("GET", "viewalldoctors.php");

    xmlhttp.send();

  </script> -->

<br><br>


  <!-- Add a record


  Delete a record


  Update a record


  Query (at least 3 select statements on one relation) -->


  <big> <b>Query A Doctor From Table</b></big><br>

  <br><i>Please enter the surname of the doctor you wish to find </i>

  <br><br>

  <form action="deldoctor.php" method="post">

    <input name="last" type="text" placeholder="Last">

    <input type="submit"></input>


  </form>

  <table></table>

  <br><br>


  <!-- <big> <b>Existing Nurses</b></big>

<br><br> -->
```

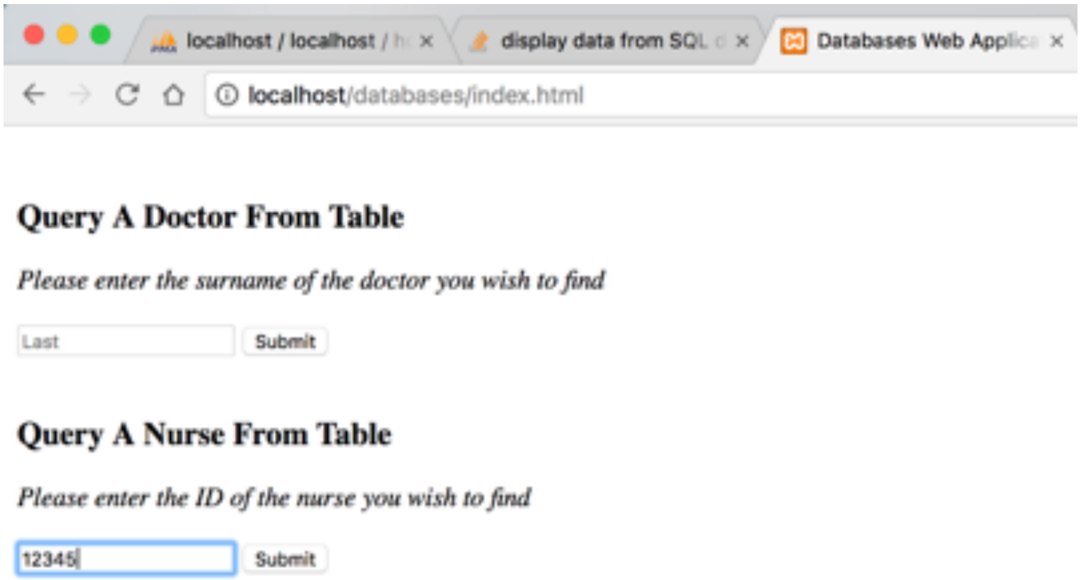
```
  <!-- <div id="nursediv"></div>
  <script>
    //Set up a HTTP request to users.php, it will generate a list of all
entries
    //from the Users table in the ClassDemo database
    //NOTE: The code below uses AJAX to make a HTTP GET request to a php file
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
      if(this.readyState == 4 && this.status == 200) {
        //responseText is generated by nouns.php
        document.getElementById("nursediv").innerHTML = this.responseText;
      }
    }
    xmlhttp.open("GET", "viewallnurse.php");
    xmlhttp.send();
  </script> -->
  <big> <b>Query A Nurse From Table</b></big><br>
  <br><i>Please enter the ID of the nurse you wish to find </i>
  <br><br>
  <form action="delnurse.php" method="post">
    <input name="nlast" type="text" placeholder="Last">
    <input type="submit"></input>

  </form>
  <table></table>
  <br><br>

</body>
```

```
</html>
```

PHP CODE:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>View All Doctors</title>
  </head>
  <body>
    <p>
    <?
      $host = "localhost";
      $user = "root";
      $password = "";

      $mysqli = new mysqli($host, $user, $password, "hospital");

      if ($mysqli->connect_errno) {
        die("Error: " . $mysqli->connect_errno);
      }
      $last = $_POST["last"];
      $sql="SELECT * FROM nurse WHERE last = '$last'";
      $result= $mysqli->query($sql);
    ?>
    </p>
  <a href="index.html">Return home</a>

  </body>

</html>
```

SCREENSHOT BEFORE QUERY #3



Query A Doctor From Table

Please enter the surname of the doctor you wish to find

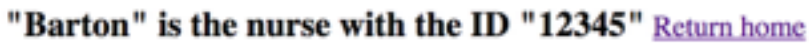
Last

Query A Nurse From Table

Please enter the ID of the nurse you wish to find

12345

SCREENSHOT AFTER QUERY #3



"Barton" is the nurse with the ID "12345" Return home

HTML CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Databases Web Application</title>
</head>
<body>
  <!-- <big> <b>Existing Doctors</b></big>
<br><br>
  <div id="usersDiv"></div> -->
  <!-- <script>
    //Set up a HTTP request to users.php, it will generate a list of all entries
    //from the Users table in the ClassDemo database
    //NOTE: The code below uses AJAX to make a HTTP GET request to a php file
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
      if(this.readyState == 4 && this.status == 200) {
        //responseText is generated by nouns.php
        document.getElementById("usersDiv").innerHTML = this.responseText;
      }
    }
    xmlhttp.open("GET", "viewalldoctors.php");
    xmlhttp.send();
  </script> -->
<br><br>
```

```
  <!-- Add a record

  Delete a record

  Update a record

  Query (at least 3 select statements on one relation) -->


  <big> <b>Query A Doctor From Table</b></big><br>
  <br><i>Please enter the surname of the doctor you wish to find </i>
  <br><br>
  <form action="deldoctor.php" method="post">
    <input name="last" type="text" placeholder="Last">
    <input type="submit"></input>

  </form>
  <table></table>
  <br><br>


  <!-- <big> <b>Existing Nurses</b></big>
<br><br> -->
  <!-- <div id="nursediv"></div>
  <script>
    //Set up a HTTP request to users.php, it will generate a list of all
entries
    //from the Users table in the ClassDemo database
    //NOTE: The code below uses AJAX to make a HTTP GET request to a php file
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
      if(this.readyState == 4 && this.status == 200) {
        //responseText is generated by nouns.php
        document.getElementById("nursediv").innerHTML = this.responseText;
      }
    }
    xmlhttp.open("GET", "viewallnurse.php");
    xmlhttp.send();
  </script> -->
  <big> <b>Query A Nurse From Table</b></big><br>
  <br><i>Please enter the ID of the nurse you wish to find </i>
  <br><br>
  <form action="delnurse.php" method="post">
    <input name="nurseID" type="text" placeholder="ID">
    <input type="submit"></input>

  </form>
  <table></table>
  <br><br>

</body>
</html>
```

PHP CODE:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>View All Doctors</title>
  </head>
  <body>
    <p>
    <?
      $host = "localhost";
      $user = "root";
      $password = "";

      $mysqli = new mysqli($host, $user, $password, "hospital");

      if ($mysqli->connect_errno) {
        die("Error: " . $mysqli->connect_errno);
      }
      $nurseID = $_POST["nurseID"];
      $sql="SELECT name FROM nurse WHERE nurseID = '$nurseID'";
      $result= $mysqli->query($sql);
    ?>
    </p>
  <a href="index.html">Return home</a>
  </body>
</html>
```