

## BDF – Database Assignment

In this assignment you will be performing several tasks against your database.

- Create a new database and import a table with data.
- Normalize the “usersNotNormalized” table.
- Create a normalized table for your project.
- Insert, Select, Update, & Delete data.
- Perform a SQL Dump (Export)

After completing this assignment, you will be submitting this document and a SQL Dump in the following naming convention.

All lowercase **lastname\_firstname\_database.pdf**  
**lastname\_firstname\_database.sql**

Paste your answer in the marked areas below. To expand this area, highlight your answer, open your “Formatting Palette”, Styles section and select the “Answer” style.

Answers go here.

We suggest you watch the following videos before attempting this assignment.

- <http://vimeo.com/24637589>

---

## Part 1

Create a new database and import a table with data.

The “.sql” file with this assignment is called SQL Dump. This file is an export of all objects and data from a database. You will use this file to import a table and sample data.

Start up Mamp and Sequel Pro. You will need to create a new database called “bdfYYMM” (replace YYMM with the current year and month). Once the database is created, open the query tab, and then drag the SQL file into the window. Once there, click “Run All” to execute the SQL and import the tables and data to your database.

If successful, you can click on the table to the left, then the “content” tab to view the imported data.

userId	firstname	lastname	username	password	dob	userStatus	userType	createdDate
1	John	Doe	jdoe	pass123	1991-01-23	active	administrator	2012-03-14 12:34:44
2	Bruce	Wayne	bbat	dog334	1993-07-10	active	client	2012-03-15 17:54:09
3	Dave	Banner	dHulk	apple3	1990-05-19	deleted	Client	2012-03-15 22:01:34
4	Tony	Stark	ts101	couch	1996-03-01	Active	cleint	2012-03-16 08:01:56
5	Peter	Parker	spider1	web777	1987-04-27	active	clients	2012-03-17 10:24:25

---

## Part 2

Normalize the table “usersNotNormalized”.

Database normalization is the process of optimising the structure of data for several reasons including:

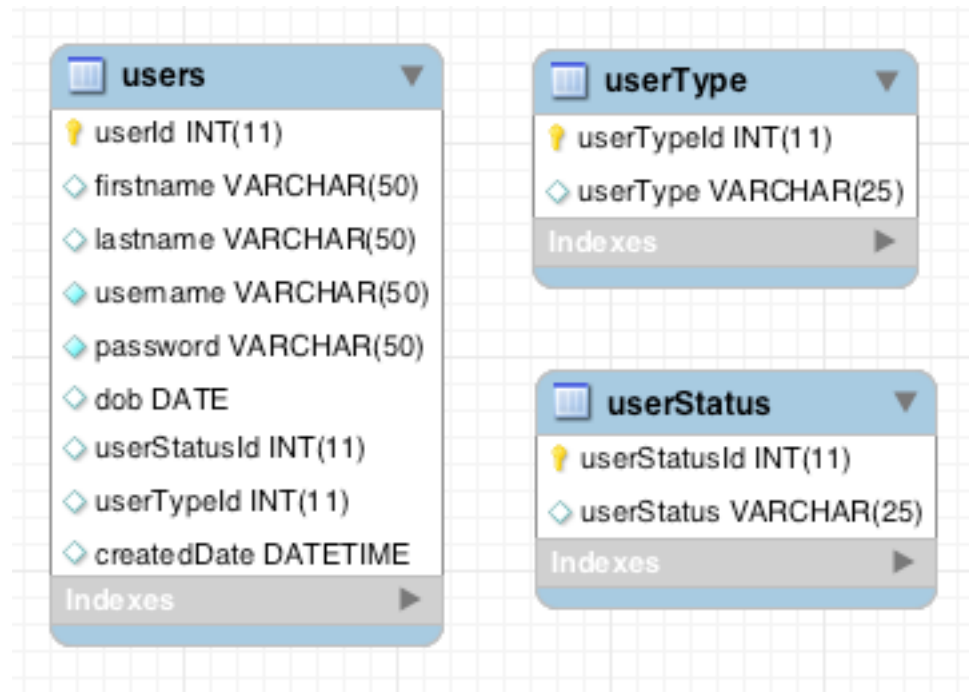
- Performance
- Consistency
- Eliminate Duplicate Data
- Forward Compatibility

The table we imported, usersNotNormalized, is not normalized. If we review each column, we can see there are 2 columns, (userStatus & userType) which have repeating groups of data.

Every user should have a status of “active” or “deleted”. However, some of the columns have capital letters or misspellings. The same is true for type. Some types are “client” while another is “Clients”.

It may be clear to the human eye these values should have the same meaning, but your computer views them as different values.

Create 3 new tables to normalize this table. Separate the values for status and type to separate table. Create a new “users” table to utilize these new tables. Your tables should resemble the ER diagram below.



Users (Not Normalized)

userId	firstname	lastname	username	password	dob	userStatus	userType	createdAt
1	John	Doe	jdoe	pass123	1991-01-23	active	administrator	2012-03-14 12:34:44

Users (Normalized)

userId	firstname	lastname	username	password	dob	userStatusId	userTypeId	createdAt
1	John	Doe	jdoe	pass123	1991-01-23	1	2	2012-03-14 12:34:44

userStatusId	userStatus
1	active
2	deleted

userTypeId	userType
1	client
2	administrator

You may use Sequel Pro's GUI to create these tables but you are required to paste your "create table" statements below. Make sure the tables are created with type = InnoDB.

Answers go here.

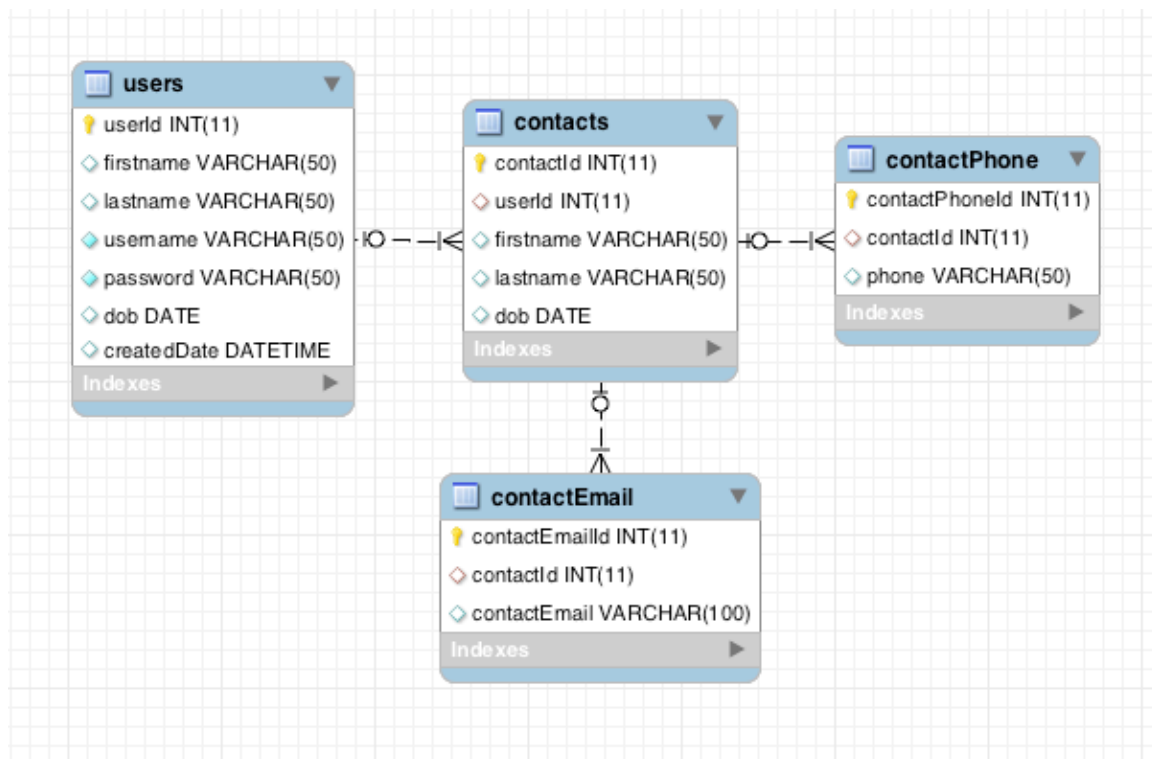
```
CREATE TABLE `users` (  
    `userId` int(11) unsigned NOT NULL AUTO_INCREMENT,  
    `firstname` varchar(50) DEFAULT NULL,  
    `lastname` varchar(50) DEFAULT NULL,  
    `username` varchar(50) NOT NULL DEFAULT '',  
    `password` varchar(50) NOT NULL DEFAULT '',  
    `dob` date DEFAULT NULL,  
    `userStatusId` int(11) DEFAULT NULL,  
    `userId` int(11) DEFAULT NULL,  
    `createdDate` datetime DEFAULT NULL,  
    PRIMARY KEY (`userId`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
CREATE TABLE `userStatus` (  
    `userStatusId` int(11) unsigned NOT NULL  
    AUTO_INCREMENT,  
    `userStatus` varchar(25) DEFAULT NULL,  
    PRIMARY KEY (`userStatusId`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `userType` (  
    `userId` int(11) unsigned NOT NULL AUTO_INCREMENT,  
    `userType` varchar(25) DEFAULT NULL,  
    PRIMARY KEY (`userId`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

**Part 3**

Create a normalized table for your project.

Create a few related tables (minimum of 2) that will be needed for your project that you created in the preproduction assignment. For example, if your project were an address book, you would need to store information about your contacts, their phones and email addresses. You would create a contacts table, contactEmail table and contactPhone table to store your data. Foreign Keys are not required for this assignment.



You may use Sequel Pro's GUI to create these tables but you are required to paste your "create table" statements below. Make sure the tables are created with type = InnoDB.

Answers go here.

```
CREATE TABLE `usersPetOwners` (  
  `userId` int(11) unsigned NOT NULL AUTO_INCREMENT,
```

```
`firstName` varchar(50) DEFAULT NULL,  
  
`lastName` varchar(50) DEFAULT NULL,  
  
`userName` varchar(50) NOT NULL DEFAULT '',  
  
`password` varchar(50) NOT NULL DEFAULT '',  
  
PRIMARY KEY (`userId`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
CREATE TABLE `pets` (  
  
    `petId` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  
    `petType` varchar(50) NOT NULL DEFAULT '',  
  
    `sexOfPet` varchar(50) DEFAULT NULL,  
  
    `dob` date DEFAULT NULL,  
  
    PRIMARY KEY (`petId`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
CREATE TABLE `breeds` (  
  
    `breedId` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  
    `breedOfPet` varchar(50) NOT NULL DEFAULT '',  
  
    `size` varchar(50) NOT NULL DEFAULT '',  
  
    PRIMARY KEY (`breedId`)
```

## BDF – Database Assignment

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```



#### Part 4

CRUD - Insert, Select, Update, & Delete data.

Write 6 SQL statements:

1. Insert 3 rows into your new table from part 3, Use 3 separate SQL statements.
2. Update only one of the rows you just inserted. (Hint: Reference the primary key of the table in your where condition)
3. Select all of the rows from your table.
4. Finally, delete one row from your database. (Hint: Reference the primary key of the table in your where condition)

Paste 6 SQL statements here.

```
INSERT INTO usersPetOwners (firstName, lastName,
userName, password)
VALUES('Kim', 'Anderson', 'kimanderson', 'password');
```

```
INSERT INTO usersPetOwners (firstName, lastName,
userName, password)
VALUES('Jane', 'Doe', 'janedoe', 'password');
```

```
INSERT INTO usersPetOwners (firstName, lastName,
userName, password)
VALUES('Rick', 'Grimes', 'walkerHunter', 'password');
```

```
UPDATE usersPetOwners
```

```
SET
```

```
    firstName = 'Kimberly'
```

```
WHERE userId = 3;
```

```
SELECT *
```

```
FROM UsersPetOwners;
```

```
DELETE FROM UsersPetOwners
```

```
WHERE userId = 1;
```

---

## Part 5

Perform a SQL Dump of your database.

Make sure when you export your database, you choose “SQL Dump”. This will produce a “.sql” file. If you open this file, you will see SQL such as “Create table...” and “Insert into...”.