**MySQL Workbench Tutorial & MySQL Introduction**

## What is MySQL?

MySQL is an open source relational database.

MySQL is cross platform which means it runs on a number of different platforms such as Windows, Linux, and Mac OS etc.

**In this tutorial, you will learn-**

* [What is MySQL?](https://www.guru99.com/introduction-to-mysql-workbench.html#1)
* [Why use MySQL?](https://www.guru99.com/introduction-to-mysql-workbench.html#2)
* [Introducing MySQL Workbench](https://www.guru99.com/introduction-to-mysql-workbench.html#3)
* [MySQL workbench- Modeling and Design tool](https://www.guru99.com/introduction-to-mysql-workbench.html#4)
* [MySQL workbench - SQL development tool](https://www.guru99.com/introduction-to-mysql-workbench.html#5)
* [MySQL workbench - Administration tool](https://www.guru99.com/introduction-to-mysql-workbench.html#6)
* [Install MySQL workbench Guide](https://www.guru99.com/introduction-to-mysql-workbench.html#7)

## Why use MySQL?

There are a number of relational database management systems on the market.

Examples of relational databases include Microsoft SQL Server, Microsoft Access, Oracle, DB2 etc.

One may ask why we would choose MySQL over the other database management systems.

The answer to this question depends on a number of factors.

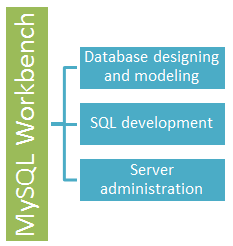
**Let's look at the strengths of MySQL compared to over relational databases such as SQL Server**-

* MySQL supports multiple storage engines each with its own specifications while other systems like SQL server only support a single storage engine. In order to appreciate this statement, let's look at two of the storage engines supported by MySQL.
* InnoDB: - its default storage engine provided with MySQL as of version 5.5. InnoDB supports foreign keys for referential integrity and also supports ACID-standard transactions.
* MyISAM: - it was the default storage engine for MySQL prior to version 5.5. MyISAM lacks support for transactions. Its advantages over InnoDB include simplicity and high performance.
* MySQL has high performance compared to other relation database systems. This is due to its simplicity in design and support for multiple-storage engines.
* Cost effective, it's relatively cheaper in terms of cost when compared to other relational databases. In fact, the community edition is free. The commercial edition has a licensing fee which is also cost effective compared to licensing fees for products such as Microsoft SQL Server.
* Cross platform - MySQL works on many platforms which means it can be deployed on most machines. Other systems such as MS SQL Server only run on the windows platform.

In order to interact with MySQL, you will need a **server access tool**that can communicate with MySQL server. MySQL supports multiple user connections.

## 

## Introducing MySQL Workbench

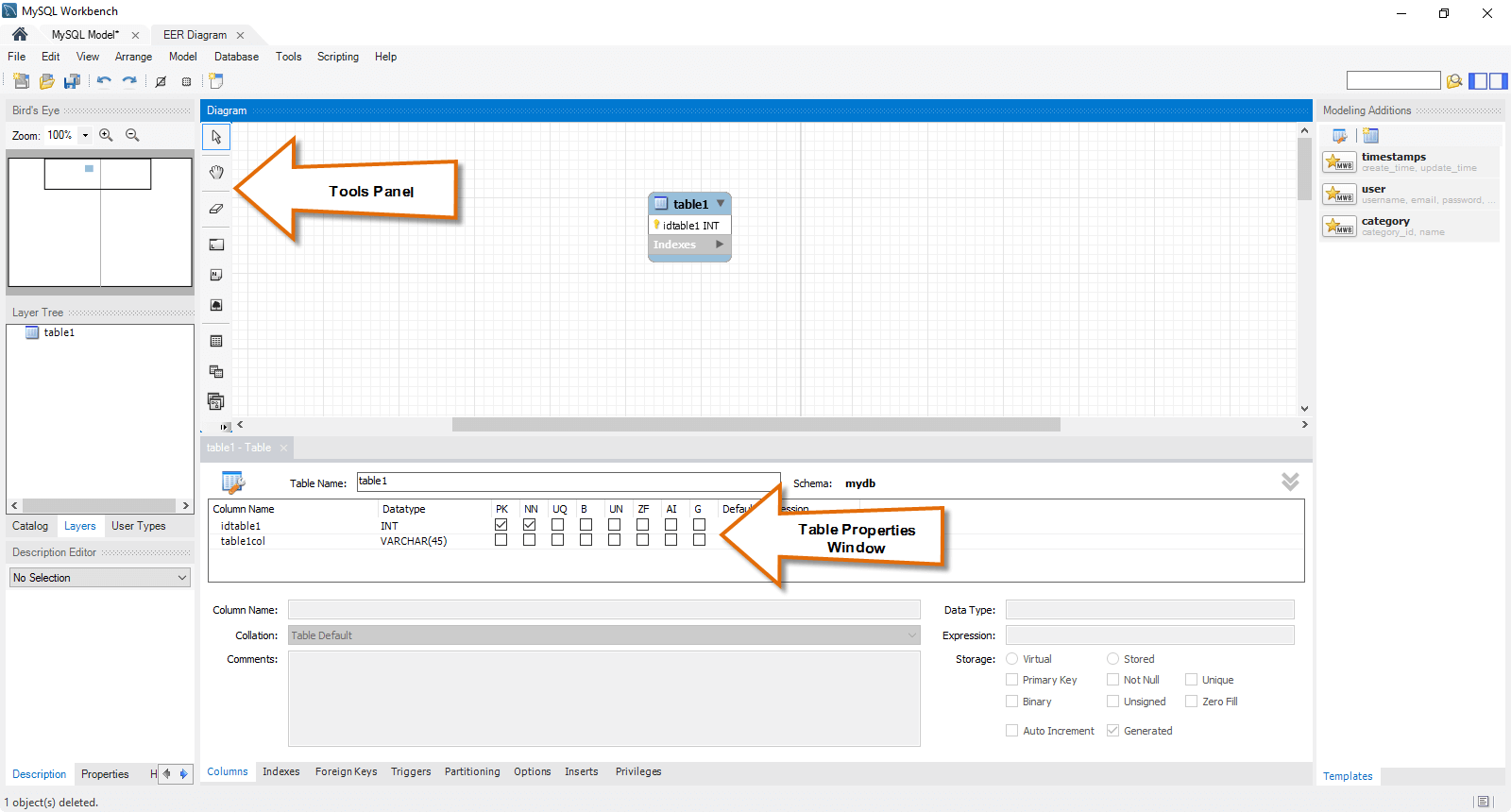
[](https://www.guru99.com/images/mysql_workbench.png)

MySQLWorkbench is a **Visual database designing and modeling** access tool for MySQL server relational database. It facilitates creation of new physical data models and modification of existing MySQL databases with reverse/forward engineering and change management functions.

## Getting Started MySQL workbench- Modeling and Design tool

* Models are at the core of most valid and high performance databases. MySQLworkbench has tools that allow developers and database administrators visually create physical database design models that can be easily translated into MySQL databases using forward engineering.
* MySQL workbench supports creation of multiple models in the same environment.
* It supports all objects such as tables, views, stored procedures, triggers,  etc. that make up a database.
* MySQL workbench has a built in model validating utility that reports any issues that might be found to the data modeler.
* It also allows for different modeling notations and can be extended by using LUA a scripting language.

The figure shown below shows the modeling window for MySQLWorkbench.

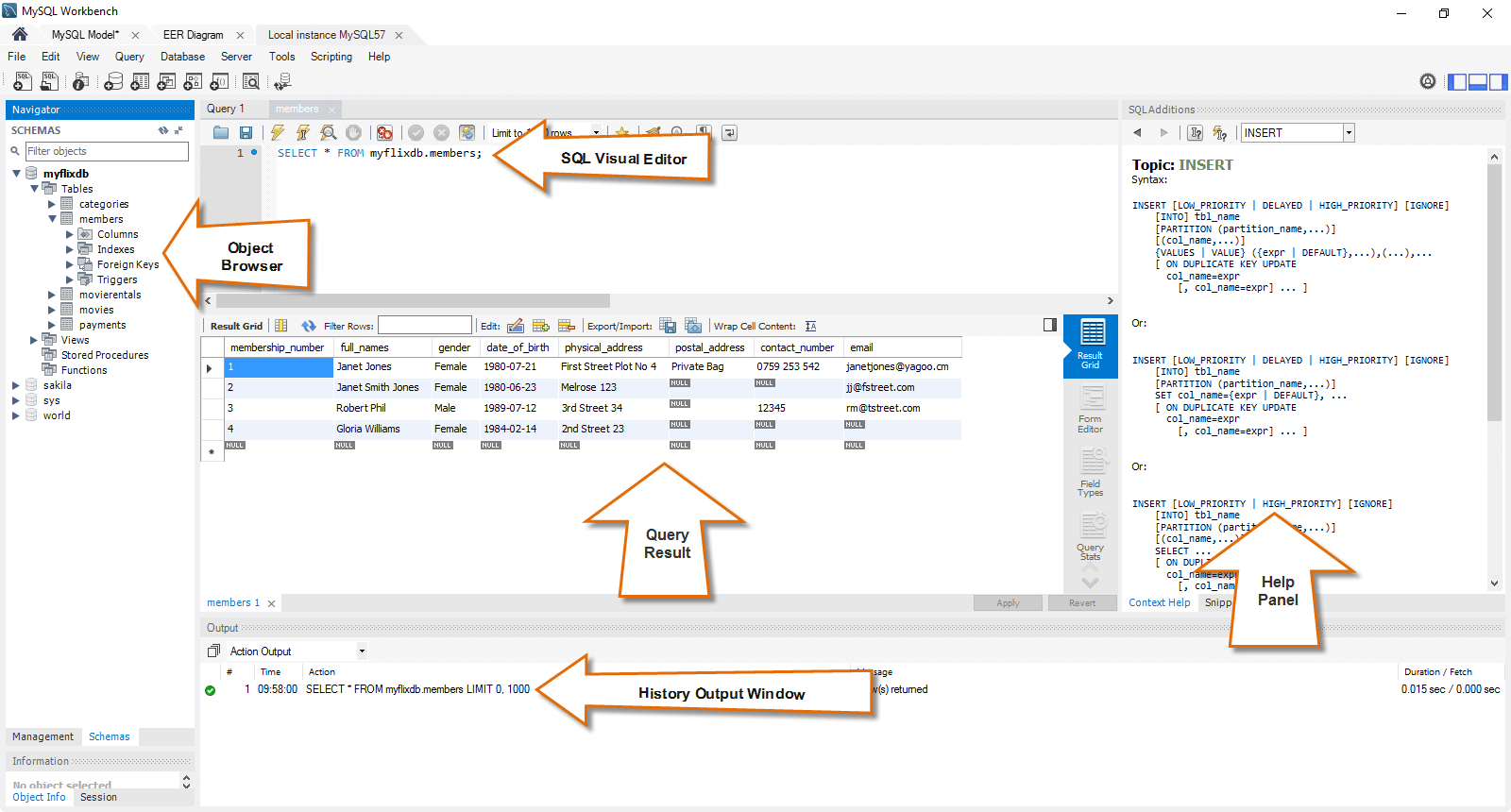
[](https://www.guru99.com/images/MSQLWorkBench.png)

## MySQL workbench - SQL development tool

Structured Query Language (SQL) allows us to manipulate our relational databases. SQL is at the heart of all relational databases.

* MySQLworkbench, has built in SQL visual editor.
* The Visual SQL editor allows developers to build, edit and run queries against MySQL server databases. It has utilities for viewing data and exporting it.
* Its syntax color highlighters help developers easily write and debug SQL statements.
* Multiple queries can be run and results automatically displayed in different tabs.
* The queries are also saved in the history panel for later retrieval and running.

The figure shown below shows the SQL development window for MySQL Workbench.

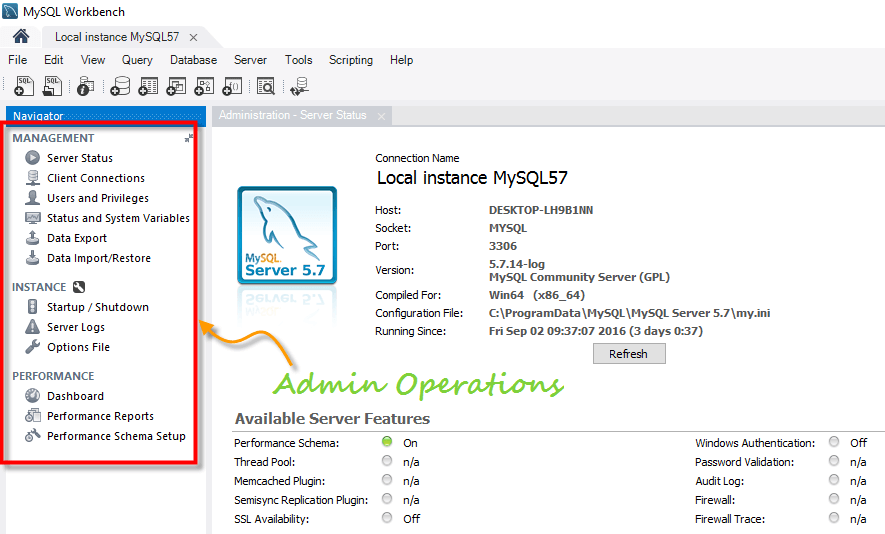
[](https://www.guru99.com/images/mysql_workbench_sql_development.png)

## MySQL workbench - Administration tool

Server administration plays a critical role in securing the data of the company. The major issues concerning server administration are users' management, server configuration, server logs and many more. Workbench MySQL has the following features that simplify the process of MySQL server administration;

* **User administration** - visual utility for managing users that lets database administrators easily add new and remove existing users if need arises, grant and drop privileges and view user profiles.
* **Server configuration** - allows for advanced configuration of the server and fine tuning for optimal performance.
* **Database backup and restorations** - visual tool for exporting/importing MySQL dump files. MySQL dump files contain SQL scripts for creating databases, tables, views, stored procedures and insertion of data.
* **Server logs** - visual tool for viewing MySQL server logs. The logs include error logs, binary logs and InnodDB logs. These logs come in handy when performing diagnosis on the server. The figure shown below shows the modeling window for MySQL Workbench.

The figure shown below shows  the Admin panel for Workbench MySQL .

[](https://www.guru99.com/images/mysql_workbench_administration.png)

## Install MySQL workbench Guide(for Windows)

This is a 2 step process

1)      Install [MySQL Community Server](http://dev.mysql.com/downloads/mysql/)

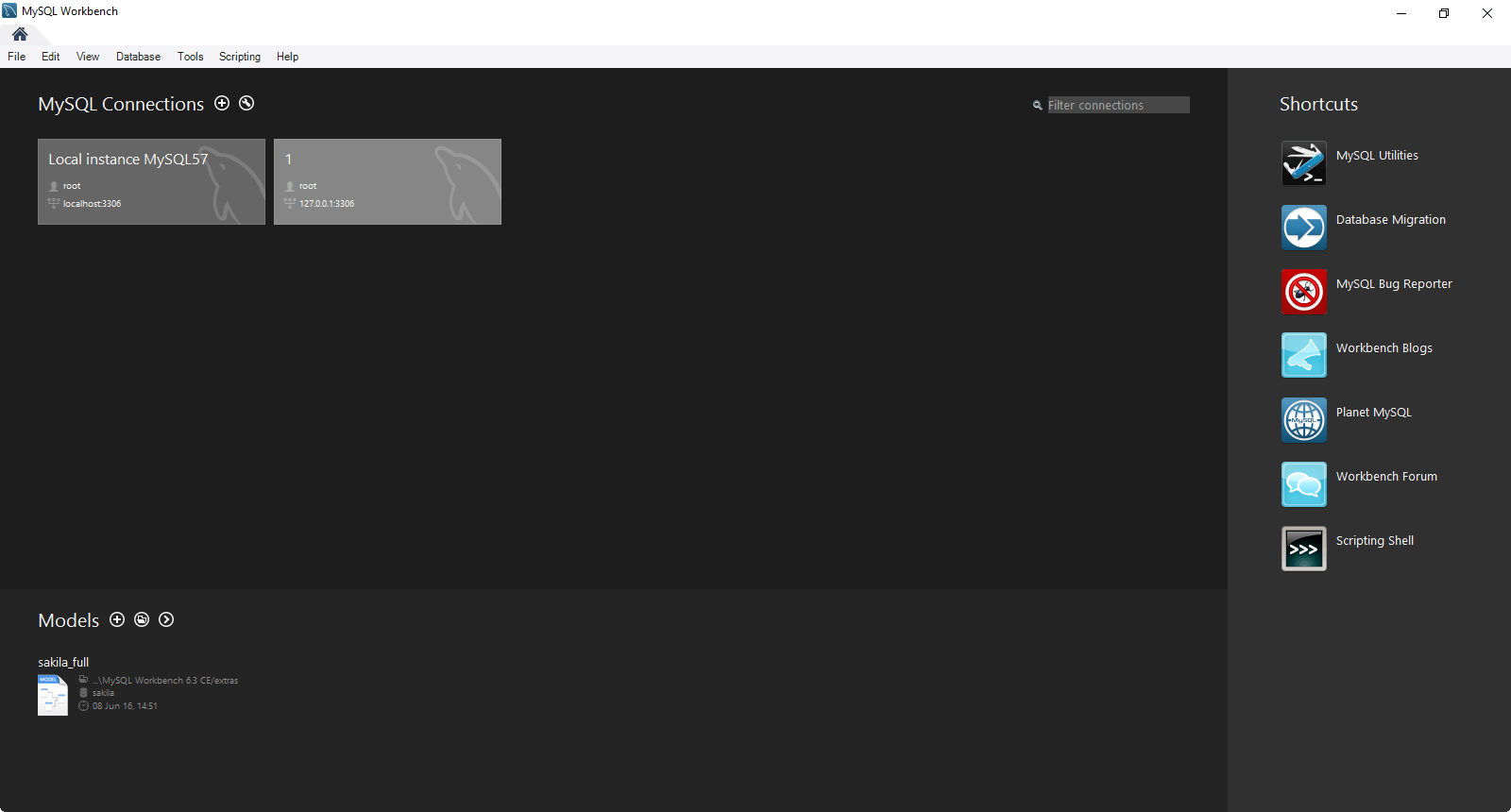
2)      Install [MySQL workbench](http://dev.mysql.com/downloads/workbench/). - You can install the workbench using a zip file or an msi installer (recommended)

Note: You will require Administrator or Power User Privileges to perform installation.

**Getting Started**

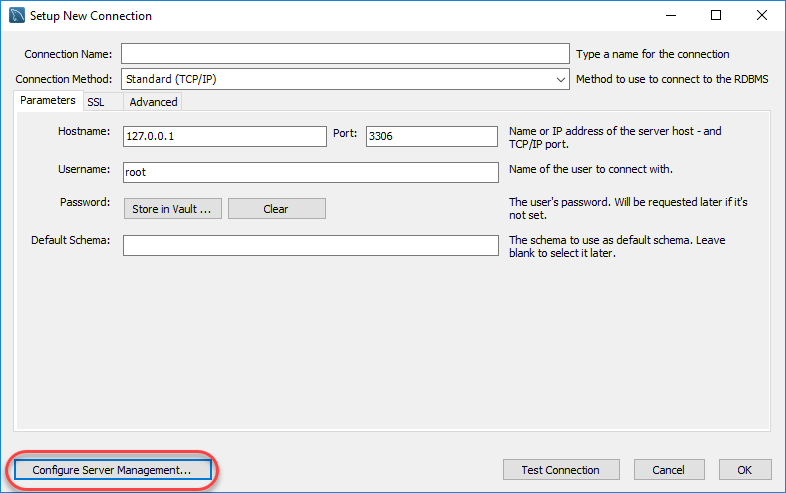
Once you have finished installing above you need to set up MySQL Workbench as shown below -

**1.**First step is launching the Workbench MySQL . What you see is called **Home Window**

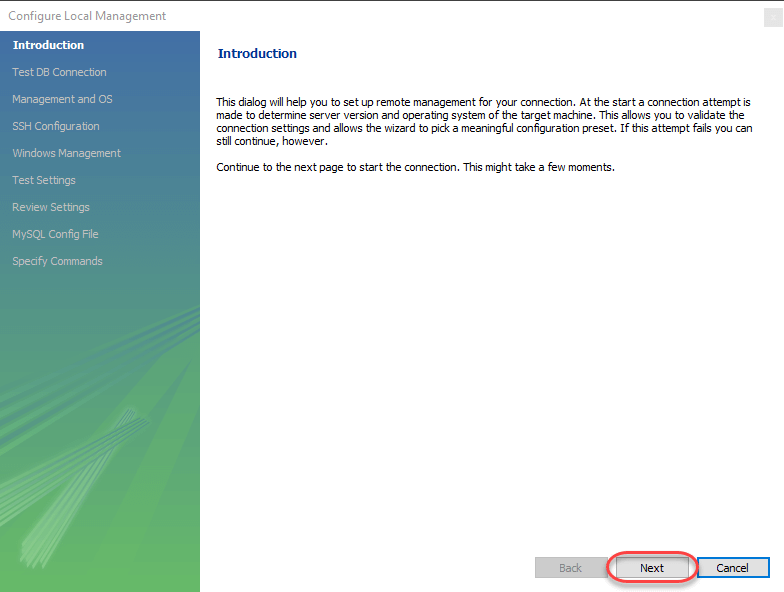
**[](https://www.guru99.com/images/mysql_work_bench_home.png)**

2. Next you need to create your MySQL Server Connection which contains details about target database server including how to connect to it. Click **" +  "** in MySQL Workbench Home Window. This will open **Setup New Connection**. Wizard

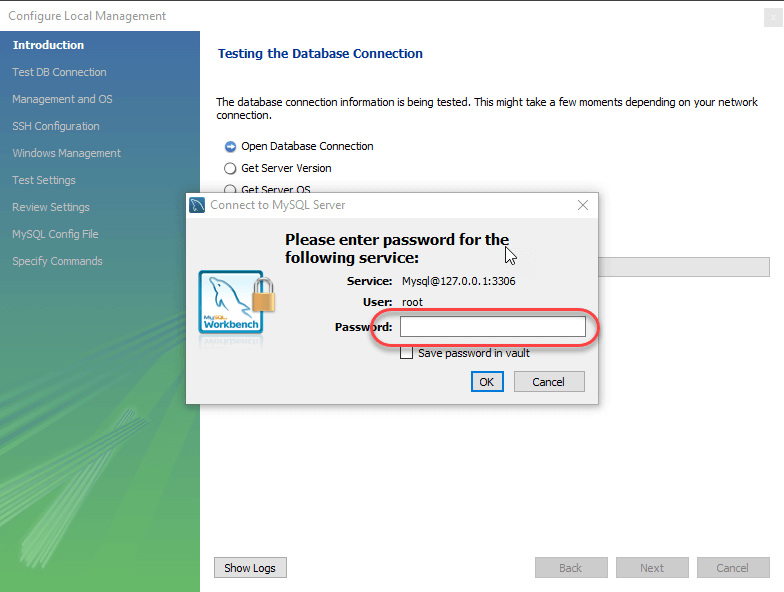
3. As a beginner you can create a connection for a locally installed server. Click **Configure Server Management**button in **Setup New Connection** window to check the cofiguration of the MySQL server.

[](https://www.guru99.com/images/workbench-instance-2.png)

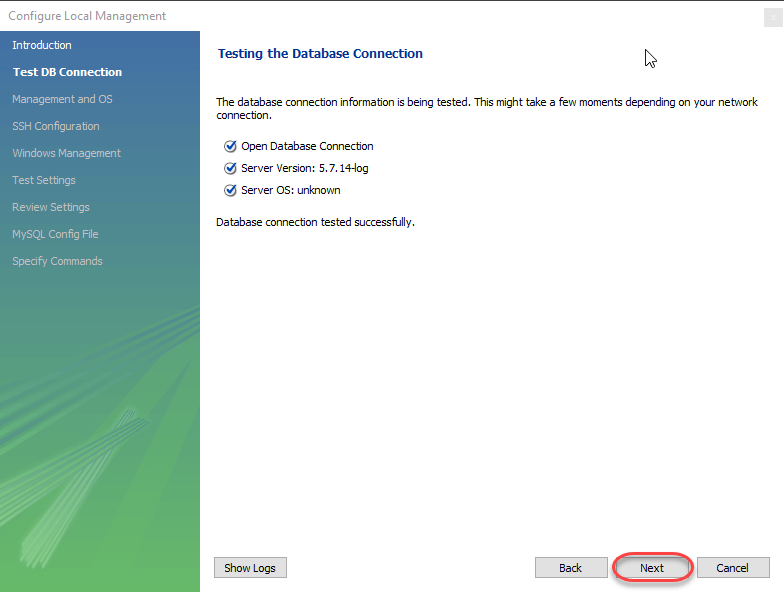
4. A new window opens named **Configure Local Management** . Click Next button to continue.

[](https://www.guru99.com/images/workbench-instance-3.png)

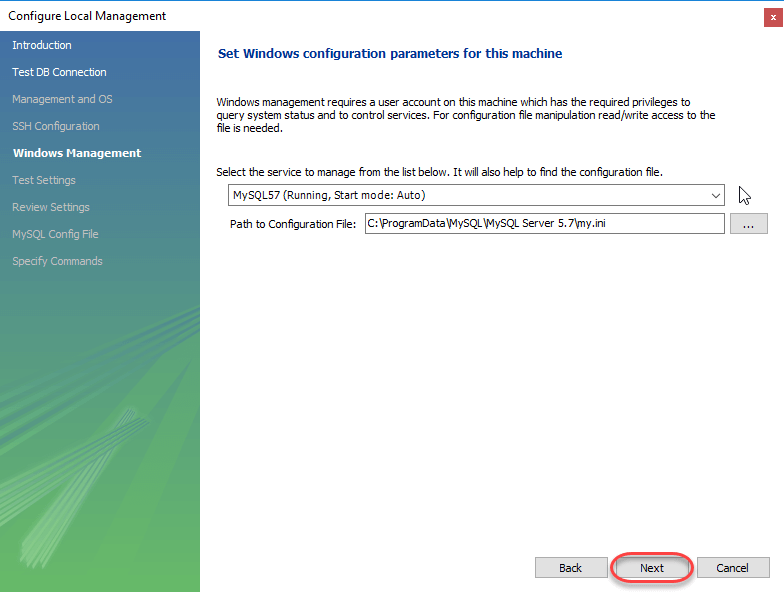
5. Next the Wizard will test connections to database. If test fails, go back and correct database connection parameters.5. Next it will open a pop up window asking your root password to test your connection with the local mysql server instance. The password is the one you set during installation of MySQL Workbench. Enter your password and press **OK**

[](https://www.guru99.com/images/workbench-instance-4.png)

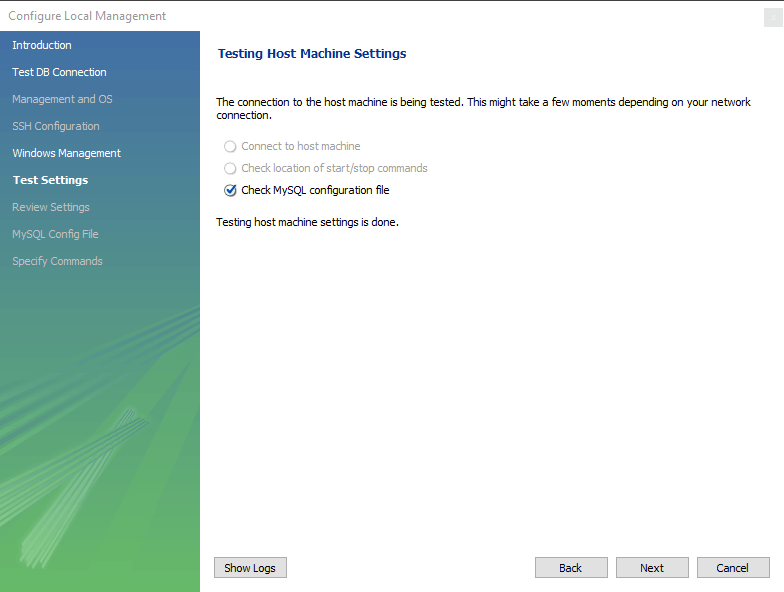
6. Next the Wizard will test connections to database. If test fails, go back and correct database connection parameters. Else if all tests are sucessful click Next to continue.

[](https://www.guru99.com/images/workbench-instance-5.png)

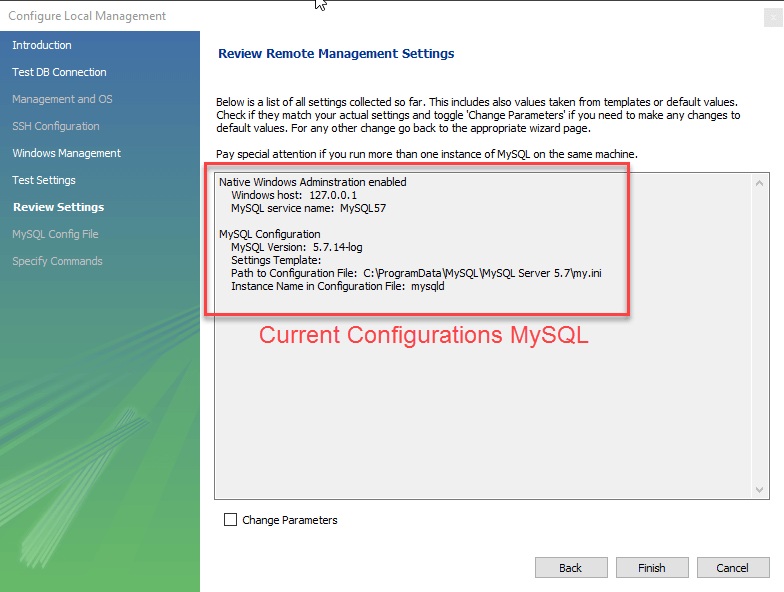
7. After that a new wizard will open about Local Service Management - It lets you switch between multiple mysql severs installed on one machines. As a beginner you can bypass this and click **Next** to continue.

[](https://www.guru99.com/images/workbench-instance-6.png)

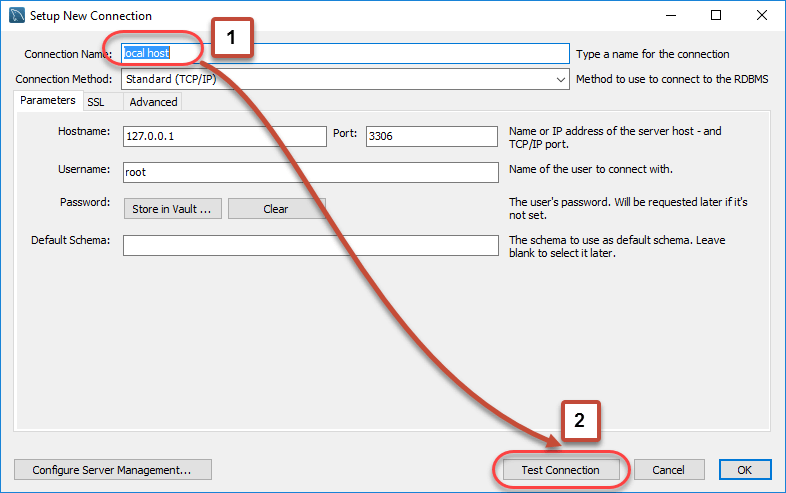
8. The Wizard will then check ability to access MySQL Server Configuration File, and test start/stop commands.

[](https://www.guru99.com/images/workbench-instance-7.png)

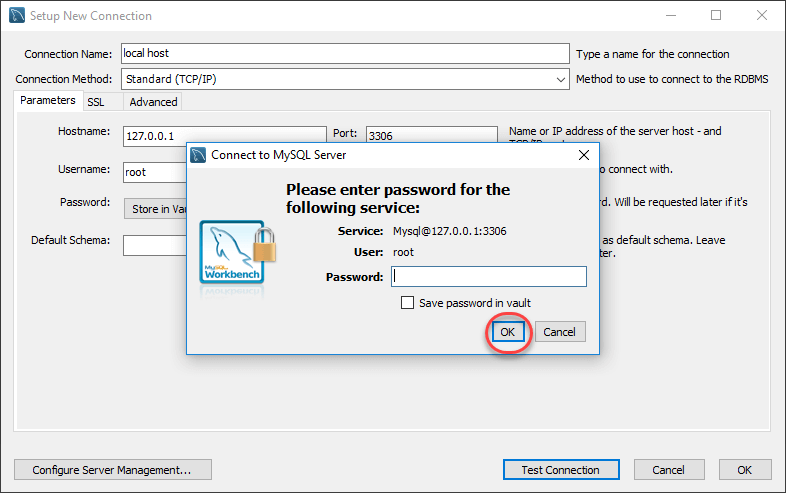
9. Next you can review current configurations. After reviewing the configurations, Click Finish to finsh server cofiguration

[](https://www.guru99.com/images/workbench-instance-7-1.png)

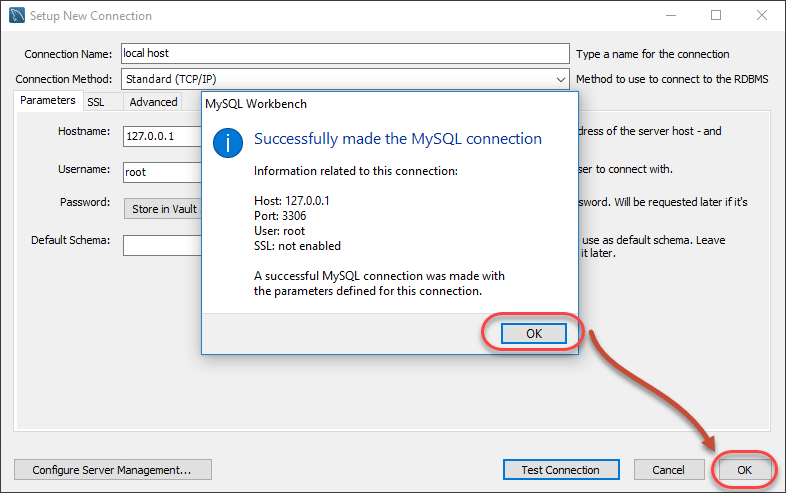
10. Next Step is to setup a connection, which can be used to connect to server. If you have not created a connection already, you can use the default values given. Click on Test Connection [ 2 ] after entering the Connection Name [ 1 ].

[](https://www.guru99.com/images/workbench-instance-8.png)

11. A new dialog box will open asking you password to root/selected user. If your MySQL root user has a password, you can enter that using Store in Vault feature. Click OK.

[](https://www.guru99.com/images/workbench-instance-9.png)

If the entered password for the user is correct then the following screen will show. Click on **both OK** buttons and you will be good to go.

[](https://www.guru99.com/images/workbench-instance-10.png)

A new instance is shown in the homepage

[](https://www.guru99.com/images/workbench-instance-11.png)

Double click and start querying.

## Summary

* MySQL is an open source relational database that is cross platform.
* MySQL supports multiple storage engines which greatly improve the server performance tuning and flexibility. Prior to version 5.5, the default storage engine was MyISAM which lacked support for transactions, as of version 5.5; the default storage engine is InnoDB which supports transactions and foreign keys.
* MySQL server can be administered using a number of server access mysql tools which include both commercial and open source products. Popular examples include;
* **phpMyAdmin** - cross platform web based open source server access tool
* **SQLYog** - targeted at the windows platform, desktop commercial server access tool
* **MySQL** **workbench** - cross platform open source server access tool.
* MySQL workbench is an integrated development environment for MySQL server. It has utilities for database modeling and designing, SQL development and server administration.

# Database Design Tutorial: Learn Data Modeling

## What is Database Design?

Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems

It helps produce  database systems

1. That meet the requirements of the users
2. Have high performance.

The main objectives of database designing are to produce logical and physical designs models of the proposed database system.

 The logical model concentrates on the data requirements and the data to be stored independent of physical considerations. It does not concern itself with how the data will be stored or where it will be stored physically.

 The physical data design model involves translating the logical design of the database onto physical media using hardware resources and software systems such as database management systems (DBMS).

**In this tutorial, you will learn-**

* [What is Database Design?](https://www.guru99.com/database-design.html#1)
* [Why Database Design is Important ?](https://www.guru99.com/database-design.html#2)
* [Database development life cycle](https://www.guru99.com/database-design.html#3)
* [Requirements analysis](https://www.guru99.com/database-design.html#4)
* [Database designing](https://www.guru99.com/database-design.html#5)
* [Implementation](https://www.guru99.com/database-design.html#6)
* [Types of Database Techniques](https://www.guru99.com/database-design.html#7)

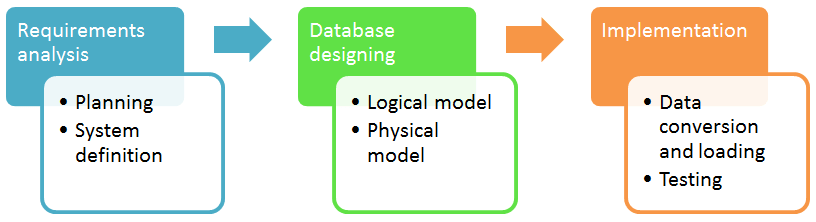
## Why Database Design is Important ?

Database designing is crucial to **high performance** database system.

  Apart from improving the performance, properly designed database are easy to maintain, improve data consistency and are cost effective in terms of disk storage space.

Note , the genius of a database is in its design . Data operations using SQL is relatively simple

## Database development life cycle

[](https://www.guru99.com/images/DatabaseDesignProcess(1).png)

The database development life cycle has a number of stages that are followed when developing database systems.

The steps in the development life cycle do not necessary have to be followed religiously in a sequential manner.

On small database systems, the database system development life cycle is usually very simple and does not involve a lot of steps.

In order to fully appreciate the above diagram, let's look at the individual components listed in each step.

## Requirements analysis

* **Planning** - This stages concerns with planning of entire Database Development Life Cycle  It  takes into consideration the Information Systems strategy of the organization.
* **System definition** - This stage defines the scope and boundaries of the proposed database system.

## Database designing

* **Logical model** - This stage is concerned with developing a database model based on requirements. The entire design is on paper without any physical implementations or specific DBMS considerations.
* **Physical model** - This stage implements the logical model of the database taking into account the DBMS and physical implementation factors.

## Implementation

* **Data conversion and loading** - this stage is concerned with importing and converting data from the old system into the new database.
* **Testing** - this stage is concerned with the identification of errors  in the newly implemented system .It checks the database against requirement specifications.

## Two Types of Database Techniques

1. **Normalization**
2. **ER Modeling**

Let's study them one by one

# What is Normalization? 1NF, 2NF, 3NF & BCNF with Examples

## What is Normalization?

Normalization is a database design technique which organizes tables in a manner that reduces redundancy and dependency of data.

It divides larger tables to smaller tables and links them using relationships.

In this tutorial, you will learn-

* [Database Normal Forms](https://www.guru99.com/database-normalization.html#9)
* [1NF Rules](https://www.guru99.com/database-normalization.html#2)
* [What is a KEY?](https://www.guru99.com/database-normalization.html#3)
* [What is Composite Key](https://www.guru99.com/database-normalization.html#4)
* [2NF Rules](https://www.guru99.com/database-normalization.html#10)
* [Database - Foreign Key](https://www.guru99.com/database-normalization.html#5)
* [What are transitive functional dependencies?](https://www.guru99.com/database-normalization.html#6)
* [3NF Rules](https://www.guru99.com/database-normalization.html#7)
* [Boyce-Codd Normal Form (BCNF)](https://www.guru99.com/database-normalization.html#8)

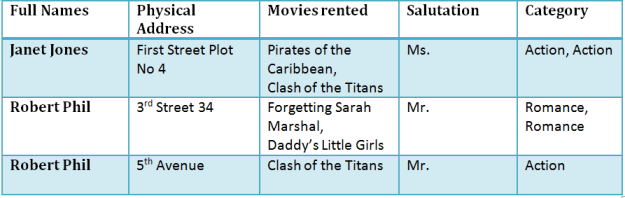
The inventor of the relational model Edgar Codd proposed the theory of normalization with the introduction of First Normal Form, and he continued to extend theory with Second and Third Normal Form. Later he joined with Raymond F. Boyce to develop the theory of Boyce-Codd Normal Form.

Theory of Data Normalization in SQL is still being developed further. For example, there are discussions even on 6th Normal Form. **However, in most practical applications, normalization achieves its best in 3rd Normal Form**. The evolution of Normalization theories is illustrated below-

[What is Normalization? 1NF, 2NF, 3NF & BCNF with Examples](https://www.guru99.com/images/NormalizationProcess(1).png)

### Database Normalization Examples -

Assume a video library maintains a database of movies rented out. Without any normalization, all information is stored in one table as shown below.

[[](https://www.guru99.com/images/NormalizationTable1.png)](https://www.guru99.com/images/NormalizationTable1.png)

[Table 1](https://www.guru99.com/images/NormalizationTable1.png)

Here you see **Movies Rented column has multiple values**.

## Database Normal Forms

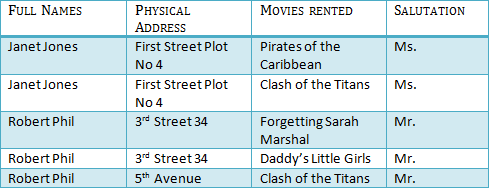
Now let's move into 1st Normal Forms

## ****1NF (First Normal Form) Rules****

* Each table cell should contain a single value.
* Each record needs to be unique.

The above table in 1NF-

### 1NF Example

[[](https://www.guru99.com/images/1NF.png)](https://www.guru99.com/images/1NF.png)

[Table 1: In 1NF Form](https://www.guru99.com/images/1NF.png)

Before we proceed let's understand a few things --

## What is a KEY?

A KEY is a value used to identify a record in a table uniquely. A KEY could be a single column or combination of multiple columns

Note: Columns in a table that are NOT used to identify a record uniquely are called non-key columns.

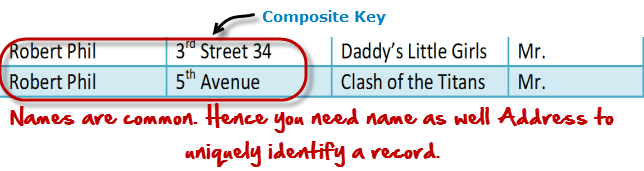
What is a Primary Key?

|  |  |
| --- | --- |
| [What is Normalization? 1NF, 2NF, 3NF & BCNF with Examples](https://www.guru99.com/images/PrimaryKey.png) | A primary is a single column value used to identify a database record uniquely.  It has following attributes   * A primary key cannot be NULL * A primary key value must be unique * The primary key values cannot be changed * The primary key must be given a value when a new record is inserted. |

## What is Composite Key?

A composite key is a primary key composed of multiple columns used to identify a record uniquely

In our database, we have two people with the same name Robert Phil, but they live in different places.

[](https://www.guru99.com/images/CompositeKey.png)

Hence, we require both Full Name and Address to identify a record uniquely. That is a composite key.

Let's move into second normal form 2NF

## 2NF (Second Normal Form) Rules

* Rule 1- Be in 1NF
* Rule 2- Single Column Primary Key

It is clear that we can't move forward to make our simple database in 2nd Normalization form unless we partition the table above.

[[](https://www.guru99.com/images/Table2.png)](https://www.guru99.com/images/Table2.png)

[Table 1](https://www.guru99.com/images/Table2.png)

[[](https://www.guru99.com/images/Table1.png)](https://www.guru99.com/images/Table1.png)

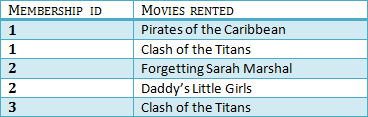
[Table 2](https://www.guru99.com/images/Table1.png)

We have divided our 1NF table into two tables viz. Table 1 and Table2. Table 1 contains member information. Table 2 contains information on movies rented.

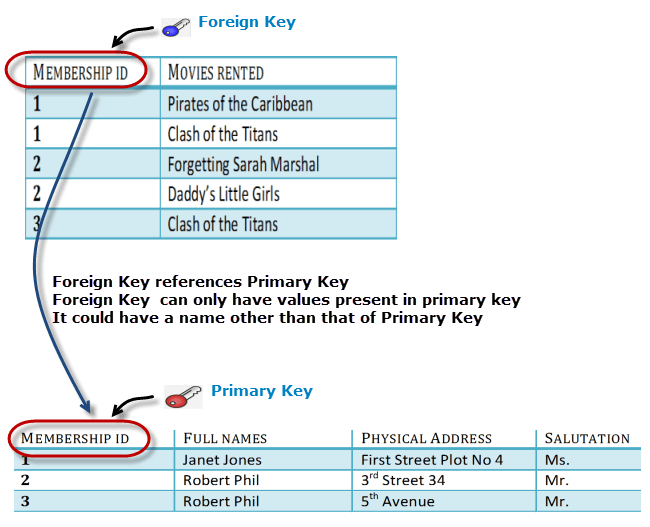
We have introduced a new column called Membership\_id which is the primary key for table 1. Records can be uniquely identified in Table 1 using membership id

## Database - Foreign Key

In Table 2, Membership\_ID is the Foreign Key

[](https://www.guru99.com/images/foreign_key_table.png)

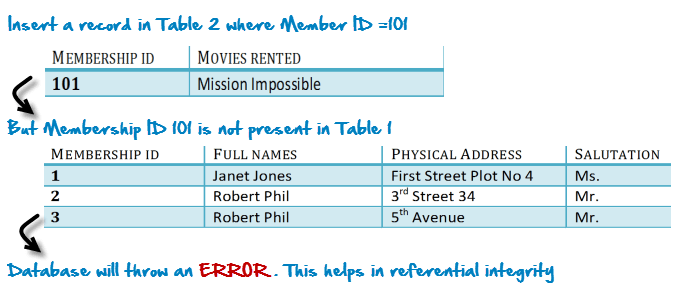
|  |  |
| --- | --- |
| [What is Normalization? 1NF, 2NF, 3NF & BCNF with Examples](https://www.guru99.com/images/ForeignKey.png) | Foreign Key references the primary key of another Table! It helps connect your Tables   * A foreign key can have a different name from its primary key * It ensures rows in one table have corresponding rows in another * Unlike the Primary key, they do not have to be unique. Most often they aren't * Foreign keys can be null even though primary keys can not |

[](https://www.guru99.com/images/ForeignKeyRelationWithPrimary.png)

Why do you need a foreign key?

Suppose an idiot inserts a record in Table B such as

You will only be able to insert values into your foreign key that exist in the unique key in the parent table. This helps in referential integrity.

[](https://www.guru99.com/images/WhyDataBaseIsImportant.png)

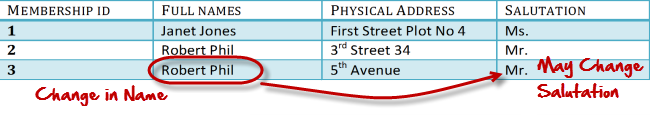
The above problem can be overcome by declaring membership id  from Table2  as foreign key of membership id from Table1

Now, if somebody tries to insert a value in the membership id field that does not exist in the parent table, an error will be shown!

## What are transitive functional dependencies?

A transitive functional dependency is when changing a non-key column, might cause any of the other non-key columns to change

Consider the table 1. Changing the non-key column Full Name may change Salutation.

[](https://www.guru99.com/images/transitive_functional_dependencies.png)

Let's move into 3NF

## 3NF (Third Normal Form) Rules

* Rule 1- Be in 2NF
* Rule 2- Has no transitive functional dependencies

To move our 2NF table into 3NF, we again need to again divide our table.

### 3NF Example

[[](https://www.guru99.com/images/2NFTable1.png)](https://www.guru99.com/images/2NFTable1.png)

[TABLE 1](https://www.guru99.com/images/2NFTable1.png)

[[](https://www.guru99.com/images/2NFTable2.png)](https://www.guru99.com/images/2NFTable2.png)

[Table 2](https://www.guru99.com/images/2NFTable2.png)

[[](https://www.guru99.com/images/2NFTable3.png)](https://www.guru99.com/images/2NFTable3.png)

[Table 3](https://www.guru99.com/images/2NFTable3.png)

We have again divided our tables and created a new table which stores Salutations.

There are no transitive functional dependencies, and hence our table is in 3NF

In Table 3 Salutation ID is primary key, and in Table 1 Salutation ID is foreign to primary key in Table 3

Now our little example is at a level that cannot further be decomposed to attain higher forms of normalization. In fact, it is already in higher normalization forms. Separate efforts for moving into next levels of normalizing data are normally needed in complex databases.  However, we will be discussing next levels of normalizations in brief in the following.

## Boyce-Codd Normal Form (BCNF)

Even when a database is in 3rd Normal Form, still there would be anomalies resulted if it has more than one **Candidate**Key.

Sometimes is BCNF is also referred as **3.5 Normal Form.**

### 4NF (Fourth Normal Form) Rules

If no database table instance contains two or more, independent and multivalued data describing the relevant entity, then it is in 4th Normal Form.

### 5NF (Fifth Normal Form) Rules

A table is in 5th Normal Form only if it is in 4NF and it cannot be decomposed into any number of smaller tables without loss of data.

### 6NF (Sixth Normal Form) Proposed

6th Normal Form is not standardized, yet however, it is being discussed by database experts for some time. Hopefully, we would have a clear & standardized definition for 6th Normal Form in the near future...

That's all to Normalization!!!

### Summary

* Database designing is critical to the successful implementation of a database management system that meets the data requirements of an enterprise system.
* Normalization helps produce database systems that are cost-effective and have better security models.
* Functional dependencies are a very important component of the normalize data process
* Most database systems are normalized database up to the third normal forms.
* A primary key uniquely identifies are record in a Table and cannot be null
* A foreign key helps connect table and references a primary key