Install Mysql
Download link :https://dev.mysql.com/downloads/

Download MySQL Sample Database (classicmodels)
Sample Database Download Link: https://www.mysqltutorial.org/mysql-sample-database.aspx

What is Database Testing?

Database Testing is a type of software testing that checks the schema, tables, triggers, etc. of the Database under test.

The 3 types of Database Testing are

1. Structural Testing
2. Functional Testing
3. Non-functional Testing

Functional Database Testing

Functional Database Testing is a type of database testing that is used to validate the functional requirements of a database from the end-user's perspective. The main goal of functional database testing is to test whether the transactions and operations performed by the end-users which are related to the database works as expected or not.

Following are the basic conditions that need to be observed for database validations.

- Whether the field is mandatory while allowing NULL values on that field?
- Whether the length of each field is of sufficient size?
- Whether all similar fields have the same names across tables?
- Whether there are any computed fields present in the Database?

Non-functional testing

Non-functional testing in the context of database testing can be categorized into various categories as required by the business requirements. These can be load testing, Stress Testing, Security Testing, Usability Testing, and Compatibility Testing, and so on.

Stored Procedure Testing

What is stored procedure

- A stored procedure is a block of SQL statements.
- We can save stored procedures and can be reused multiple times.
- We can also pass parameters to a stored procedure.

Advantage

- Reduce network traffic

- Centralize business logic in the database
- Make database more secure

Checking data integrity and consistency

Following checks are important

- ☐ Whether the data is logically well organized?
- ☐ Whether the data stored in the tables is correct and as per the business requirements?
- ☐ Whether there are any unnecessary data present in the application under test?
- ☐ Whether the data has been stored as per as the requirement with respect to data which has been updated from the user interface?
- ☐ Whether the TRIM operations performed on the data before inserting data into the Database under test?
- ☐ Whether the transactions have been performed according to the business requirement specifications and whether the results are correct or not?
- ☐ Whether the data has been properly committed if the transaction has been successfully executed?
- ☐ Whether the data has been rolled backed successfully if the transaction has not been executed successfully by the end-user?
- ☐ Whether the data has been rolled backed if the transaction has not been executed successfully and multiple heterogeneous databases have been involved in the transaction in question?
- ☐ Whether all the transactions have been executed by using the required design procedures as specified by the system business requirements?

Login and User Security

The validations of the login and user security credentials need to take into consideration the following things.

1. Whether the application prevents the user from proceeding further in the application in case of a
    - invalid username but valid password
    - valid username but invalid password.
    - invalid username and invalid password.
2. Whether the user is allowed to perform only those specific operations which are specified by the business requirements?
3. Whether the data is secured from unauthorized access?
4. Whether there are different user roles created with different permissions?
5. Whether all the users have required levels of access on the specified Database as required by the business specifications?

6. Check that sensitive data like passwords, credit card numbers are encrypted and not stored as plain text in Database. It is a good practice to ensure all accounts should have passwords that are complex and not easily guessed.

Here is some database testing checklist you might have used:

- ☐ Check if correct data is getting saved in database upon successful page submit
- ☐ Check values for columns which are not accepting null values
- ☐ Check for data integrity. Data should be stored in single or multiple tables based on design
- ☐ Index names should be given as per the standards e.g. IND_<Tablename>_<ColumnName>
- ☐ Tables should have primary key column
- ☐ Null values should not be allowed for Primary key column
- ☐ Table columns should have description information available (except for audit columns like created date, created by etc.)
- ☐ For every database add/update operation log should be added
- ☐ Required table indexes should be created
- ☐ Data should be rolled back in case of failed transactions
- ☐ Check if data is committed to database only when the operation is successfully completed
- ☐ Check if input data is not truncated while saving. Field length shown to user on page and in database schema should be same
- ☐ Check numeric fields with minimum, maximum, and float values
- ☐ Test stored procedures and triggers with sample input data
- ☐ Database name should be given as per the application type i.e. test, UAT, sandbox, live (though this is not a standard it is helpful for database maintenance)
- ☐ Database logical names should be given according to database name (again this is not standard but helpful for DB maintenance)
- ☐ Stored procedures should not be named with prefix “sp_”
- ☐ Check values for table audit columns (like createddate, createdby, updatedate, updatedby, isdeleted, deleted date, deleted by etc.) are populated properly
- ☐ Check numeric fields with negative values (for both acceptance and non-acceptance)
- ☐ Check if radio button and dropdown list options are saved correctly in database
- ☐ Check if database fields are designed with correct data type and data length
- ☐ Check if all table constraints like Primary key, Foreign key etc. are implemented correctly
- ☐ Input field leading and trailing spaces should be truncated before committing data to database

Database Testing | How To Test Schema of Database Table | Test Cases

MySQL Sample Database Schema

The MySQL sample database schema consists of the following tables:

- **Customers**: stores customer's data.
- **Products**: stores a list of scale model cars.
- **ProductLines**: stores a list of product line categories.
- **Orders**: stores sales orders placed by customers.
- **OrderDetails**: stores sales order line items for each sales order.
- **Payments**: stores payments made by customers based on their accounts.
- **Employees**: stores all employee information as well as the organization structure such as who reports to whom.
- **Offices**: stores sales office data.

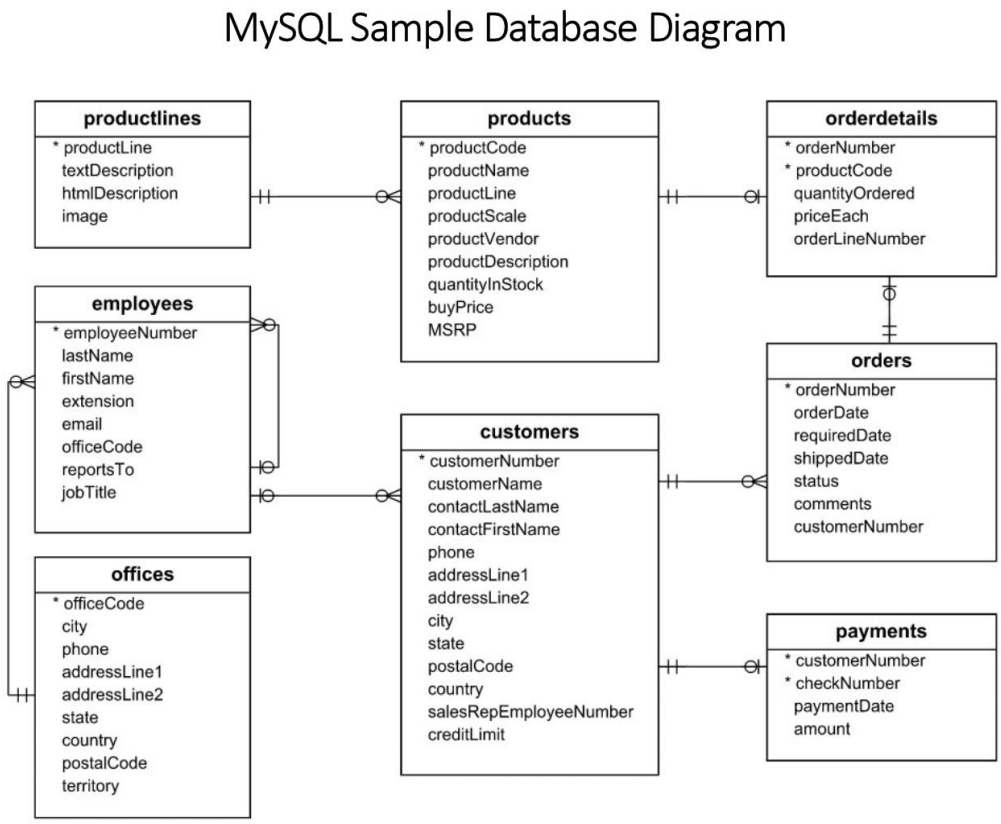

Table Name	Columns	Data Types & Size	Null	Keys
Customers	customerNumber	int(11)	No	PRI
	customerName	varchar(50)	No	
	contactLastName	varchar(50)	No	
	contactFirstName	varchar(50)	No	
	phone	varchar(50)	No	
	addressLine1	varchar(50)	No	
	addressLine2	varchar(50)	Yes	
	city	varchar(50)	No	
	state	varchar(50)	Yes	
	postalCode	varchar(50)	Yes	
	country	varchar(50)	No	
	salesRepEmployeeNumber	int(11)	Yes	MUL
	creditLimit	decimal(10,2)	Yes	
Products	productCode	varchar(15)	No	PRI
	productName	varchar(70)	No	
	productLine	varchar(50)	No	
	ProductScale	varchar(10)	No	
	productVendor	varchar(50)	No	
	productDescription	text	No	
	quantityInStock	smallint(6)	No	
	buyPrice	decimal(10,2)	No	
	MSRP	decimal(10,2)	No	

productLines	productLine	varchar(50)	No	PRI
	textDescription	varchar(50)	Yes	
	htmlDescription	mediumtext	Yes	
	image	mediumblob	Yes	

Orders	orderNumber	int(11)	No	PRI
	orderDate	date	No	
	requiredDate	date	No	
	shippedDate	date	Yes	
	status	varchar(15)	No	
	comments	text	Yes	
	customerNumber	int(11)	No	MUL

OrderDetails	orderNumber	int(11)	No	PRI
	productCode	varchar(15)	No	PRI
	quantityOrdered	int(11)	No	
	priceEach	decimal(10,2)	No	
	orderLineNumber	smallint(6)	No	

Payments	customerNumber	int(11)	No	PRI
	checkNumber	varchar(50)	No	PRI
	paymentDate	date	No	
	amount	decimal(10,2)	No	

Employee	employeeNumber	int(11)	No	PRI
	lastName	varchar(50)	No	
	firstName	varchar(50)	No	
	extension	varchar(10)	No	
	email	varchar(100)	No	
	officeCode	varchar(10)	No	MUL
	reportsTo	int(11)	Yes	MUL
	jobTitle	varchar(50)	No	

Offices	officeCode	varchar(10)	No	PRI
	city	varchar(50)	No	
	phone	varchar(50)	No	
	addressLine1	varchar(50)	No	
	addressLine2	varchar(50)	Yes	
	state	varchar(50)	Yes	
	country	varchar(50)	No	
	postalCode	varchar(15)	No	
	territory	varchar(10)	No	

Test Case

- ☐ Check table presence in database schema
- ☐ Check table name convention
- ☐ Check number of columns in table
- ☐ Check column name in table
- ☐ Check data types of column in table
- ☐ Check size of the column in a table
- ☐ Check nulls fields in table
- ☐ Check column keys in table

QUERY FOR Test case(MySql)

```
use classicmodels;

show tables;

select * from customers;

select count(*) as NumberOfColumns from information_schema.COLUMNS where
TABLE_NAME = 'customers';

select COLUMN_NAME from information_schema.COLUMNS where
TABLE_NAME='customers';

select COLUMN_NAME,DATA_TYPE from information_schema.COLUMNS where
TABLE_NAME = 'customers';

select COLUMN_NAME,COLUMN_TYPE from information_schema.COLUMNS where
TABLE_NAME = 'customers';

select COLUMN_NAME,IS_NULLABLE from information_schema.COLUMNS where
TABLE_NAME = 'customers';

select COLUMN_NAME,COLUMN_KEY from information_schema.COLUMNS where
TABLE_NAME = 'customers';
```