Database Design Guideline

* **Principle of RDBMS**

1. **Table & Field names are easily understood like :**

* Customer
  + CustomerID - **PK** - bigint
  + CustomerName
  + IsActive - bit - Set Default value - 1 or 0
  + IsDeleted - bit - Set Default value - 0
  + CreatedByID - **FK** - bigint
  + CreatedDate - Server Date Time UTC
  + ModifiedByID - **FK**- bigint , Default : 0
  + ModifiedDate - Server Date Time UTC - Default – Null
* Customer
  + CustomerAddressID – **PK** – bigint
  + CustomerID – **FK** –bigint
  + Address1
  + Address2
  + City
  + StateID – **FK** - bigint
  + CountryID – **FK** – bigint
  + ZipCode
* Order
  + OrderID – **PK** - bigint
  + CustomerID - FK - bigint
  + Amount
  + IsDeleted - bit - Default value - 0
  + CreatedByID – **FK** - bigint
  + CreatedDate - Server Date Time UTC
  + ModifiedByID – **FK** - bigint, Default : 0
  + ModifiedDate - Server Date Time UTC - Default - Null

1. **Avoid Under Score in field name, it’s difficult to write Underscore frequently SQL statement**.
2. **Try to meaning full name so it would be easy to understand** 
   * ~~CustID~~ instead of using **CustomerID**
   * ~~CUSTName~~ instead of using **CustomerName**
   * Add1/Addline1 instead of **Address1**
3. **Table & field name using Pascal Cashing like,**

* CustomerName
* EmployeeName
* Gender
* EmployeeType

1. **Don’t using special character, Space, Underscore in Table & Field Name**
2. **Avoided field name like : ~~Customer\_Name~~, ~~Emp Name~~, ~~Type01~~, ~~22User~~, ~~123~~**
3. **Always define primary key in Table with BIGINT data-type [\* use data-type as per your requirement]**

* **Cus**tomerID – PK – bigint

1. **Avoid using INT data-type in ID column, always using BIGINT [Because we have no limit of “N” no of records]**
2. **Table names are not pluralized ("User" table not "Users")**
3. **Always define constraint from database side whenever is required like**

* Unique Key
* Null or Not Null
* Default value biding like
* **IsDeleted** field default value = 0
* **IsActive** field default value = 1

1. **Define field description whenever required like,**

* **IsActive :**  1= Active, 0 = Inactive
* **IsDeleted:** 0 = Not Deleted, 1 = Deleted
* **PaymentStatus:** 1= Pending, 2 =Success, 3 = Failed

1. **Always use Auto Identity in Primary key with BIGINT data-type.**
2. **Database Normalization is a technique of organizing the data in the database, refer below link**

* <http://www.dotnet-tricks.com/Tutorial/sqlserver/756N210512-Database-Normalization-Basics.html>
* <http://www.eazynotes.com/pages/database-management-system/normalization.html>
* **Database Store Procedure & Function code standard**

1. **Stored Procedures we should avoid the prefix "sp\_" with the name of procedure. Since "sp\_" prefix is already fixed for system defined stored procedures. When we execute store procedure , it should be search entire system rather than if we use prefix “usp” then the search only our database schema.**
2. **Find below Store Procedure naming conventions, [it’s should be matter of choice, I just add my thought process for easiness.]**

* **AddUpdateCustomer** 
  + Insert/Update operation
* **GetAllCustomerList** 
  + Listing operation
* **GetCustomerByCustomerID** 
  + Single record for edit/view operation
* **GetViewCustomerByCustomerID** 
  + Single record for only view /display operation
* **DeleteCustomer** 
  + Delete customer but internally just flag Update IsDeleted =true
* **UpdateCustomerStatus** 
  + Update IsActive status for customer

1. **Store Procedure Standard**

* **Capitalize reserved words**
* **Main keywords on new line**
* **You can use prefix “usp” but no need to define prefix.**
* **Make Insert/Update store procedure together**
* **Avoid ORDER BY 1, We should write ORDER BY T1.Col1**
* **Avoid SELECT \* FROM Table1, We should write field name always instead of “ \*”**
* **Refer URL : https://www.simple-talk.com/sql/t-sql-programming/transact-sql-formatting-standards-%28coding-styles%29/**
* **Find below User define function naming conventions,**
  + - * fnCustomerFullAddress
      * fnCalculateAge
      * fnFullName
      * fnSpliteString
* **Find below View naming conventions,**
  + - * vwCustomerAddress
      * vwCustomerOrderDetail
* **Find below Trigger function naming conventions,**
  + - * trgCustomerInsert
      * trgCustomerDelete
      * trgCustomerUpdate
* **RDBMS Introduction**

RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems like **MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.**

* **RDBMS Role Management**

1. **Follow the standard of Role Management with username & password of respective database.**
2. **Avoid using Server administrator “sa” users. We can access all database through the ‘sa’ login.**

**Security reason using only respective database user/pass.**

* **RDBMS Testing**

1. **Check your database server performance with bigger amount of data.**
   1. **Add 1 Crore record throughout out the script or loop. Check with database performance.**
   2. **Check searching functionality performance**
   3. **Check Indexing if you require particular field**
   4. **You can user composite index like combination of 1 or more field.**
2. **Indexing :**

**You can use indexing for better performance, but other hand you can decrease the performance of insert/update/delete statement.**

**So be very careful for creating the indexing.**

**Find below link for regarding indexing in RDBMS.**

* **<http://www.tutorialspoint.com/dbms/dbms_indexing.htm>**
* **http://www.verynoisy.com/sql-indexing-dummies/**
* **RDBMS Security [SQL INJECTION]**

Nowadays, Database Security is more important topics. We need to take care of SQL INJECTION in our database.

We need to prevent **SQL INJECTION** in our website.

Find below URL for understanding the concept of SQL INJECTION

* **<https://en.wikipedia.org/wiki/SQL_injection>**
* **<http://www.w3schools.com/sql/sql_injection.asp>**
* **<http://breakthesecurity.cysecurity.org/2010/12/hacking-website-using-sql-injection-step-by-step-guide.html>**
* **<http://www.guru99.com/learn-sql-injection-with-practical-example.html>**
* **http://www.thisislegal.com/tutorials/18**