#### Real Plaza - Logo.JPGREAL PLAZA

# Área de Tecnologías de la Información

**Proyecto :** Advance

#### Descripción : Estándares de Modelado para el Diseño de Base de Datos

1. **INTRODUCCION**

El objetivo del presente documento es definir los estándares a utilizar para el diseño de base de datos en proyectos de desarrollo de software con Microsoft SQL Server 2008 R2 como el gestor de bases de datos, y en este caso en especifico, para el diseño del Sistema Advance.

Cada formato y estructura definida tiene la finalidad de mantener un orden y control durante el desarrollo de software, por lo tanto el presente documento está dirigido a analistas, diseñadores, analistas programadores y programadores, con el fin de homogeneizar el desarrollo de los sistemas.

1. **DEFINICION DE BASE DE DATOS**

El nombre de la Base de Datos deberá ser corto, descriptivo y que permita determinar fácilmente su propósito. Estará asociado directamente con el aplicativo al cual dará información.   
Se recomienda anteponer las siglas BD para identificar que el objeto en mención es una base de datos seguido del símbolo “\_” y del nombre descriptivo mencionado anteriormente.

Por ejemplo:

Para la aplicación en mención, “Advance”, el nombre de la base de datos podría ser “BD\_ADVANCE” o “BD\_ADV”.

1. **DEFINICION DE TABLA**

El nombre de la tabla seguirá la siguiente nomenclatura:

**XXX\_T\_NOMBRE\_TABLA**

Donde:

* **XXX**: Son 3 letras mayúsculas e indica el sistema o subsistema al que pertenece
* **\_**: Es un símbolo (raya abajo).
* **T**: Letra en mayúscula que identifica el tipo de objeto, en este caso una tabla y será la letra “T”.
* **\_**: Es un símbolo (raya abajo).
* **NOMBRE\_TABLA**: identifica el nombre físico de la tabla, debe ser todo en mayúsculas y en singular. En caso de una tabla temporal se debe añadir el prefijo TEMP.   
  Para separar palabras se debe usar el símbolo “\_”.   
  Debe mantenerse el nombre de la tabla padre en la tabla detalle o hija.

Por ejemplo:

ADV\_T\_INMUEBLE

ADV\_T\_INMUEBLE\_IMAG

ADV\_T\_TEMP\_*NOMBRE\_TABLA*

1. **DEFINICION DE COLUMNA O CAMPO**

El nombre de la columna, o campo, seguirá la siguiente nomenclatura:

**nemonico\_tabla\_c\_znombre\_columna**

Donde:

* **nemonico\_tabla**: Son 4 letras minúsculas e indican la tabla a la que pertenece la columna.  
  **Esta parte pude variar dependiendo de la longitud del nombre de la tabla, en tablas padres puede seguir esta nomenclatura, en tablas hijas, dependientes o detalle esta parte de la nomenclatura puede variar, principalmente aumentar y usar “\_” para separar palabras.**
* **\_**: Es un símbolo (raya abajo).
* **c**: Letra minúscula que identifica el tipo de objeto, en este caso una columna y será la letra “c”.
* **\_**: Es un símbolo (raya abajo).
* **z**: Letra minúscula que identifica el tipo de dato del objeto. **Considerar Tabla 1**.
* **nombre\_columna**: identifica el nombre de la columna, debe ser todo en minúsculas y en singular. Para separar palabras se debe usar el símbolo “\_”.

Por ejemplo:

CREATE TABLE ADV\_T\_INMUEBLE

{

inmu\_c\_iid\_inmu INT NULL,

inmu\_c\_vnomb\_inmu VARCHAR(100) NULL,

inmu\_c\_cabrev\_inmu CHAR(10) NULL

inmu\_c\_earea\_total DECIMAL(8,2) NULL

prop\_c\_vnomb\_prop VARCHAR(200) NULL

}

|  |  |
| --- | --- |
| **CARACTER** | **TIPO DE DATO** |
| C | Char |
| D | Date |
| S | Smalldatetime |
| I | Int |
| N | Numeric |
| V | Varchar |
| B | Bit |
| G | BigInt |
| A | Binary |
| E | Decimal |
| F | Float |
| Z | Datetime |
| O | Money |
| H | Nchar |
| T | NText |
| R | NVarchar |
| L | Real |
| Q | Sql\_Variant |
| X | Text |
| P | Timestamp |
| Y | TinyInt |
| U | Uniqueidentifier |

**Tabla 1 – Abreviatura de tipos de datos**

1. **DEFINICION DE OBJETOS DE INTEGRIDAD REFERENCIAL**
2. **Primary Key**

Un primary key seguirá la siguiente nomenclatura:

**PK\_XXX\_T\_NOMBRE\_TABLA**

Donde:

* **PK**: prefijo que denomina a un Primary Key.
* **\_**: Es un símbolo (raya abajo).
* **XXX\_T\_NOMBRE\_TABLA**: este punto está definido en el tema de definición de tablas.

Por ejemplo:

ALTER TABLE ADV\_T\_INMUEBLE

{

ADD CONSTRAINT PK\_ADV\_T\_INMUEBLE

PRIMARY KEY (inmu\_c\_iid\_inmu)

}

1. **Foreign Key**

Un foreign key seguirá la siguiente nomenclatura:

**FK\_XXX\_T\_NOMBRE\_TABLAORIGEN\_XXX\_T\_NOMBRE\_TABLAREFERENCIA**

Donde:

* **FK**: prefijo que denomina a un Foreign Key.
* **\_**: Es un símbolo (raya abajo).
* **XXX\_T\_NOMBRE\_TABLAORIGEN**: este punto está definido en el tema de definición de tablas, y especifica a la tabla origen.
* **\_**: Es un símbolo (raya abajo).
* **XXX\_T\_NOMBRE\_TABLAREFERENCIA:** este punto está definido en el tema de definición de tablas, y especifica a la tabla de referencia.

Por ejemplo:

ALTER TABLE ADV\_T\_INMUEBLE

{

ADD CONSTRAINT FK\_ADV\_T\_INMUEBLE\_ADV\_T\_PROP

FOREIGN KEY (prop\_c\_vnomb\_prop)

REFERENCES ADV\_T\_PROP (prop\_c\_vnomb\_prop)

ON UPDATE NO ACTION

ON DELETE NO ACTION

}

1. **Default**

Un Default seguirá la siguiente nomenclatura:

**DF\_NEMONICO\_TABLA\_NEMONICO\_COLUMNA**

Donde:

* **DF**: prefijo que denomina a un Default.
* **\_**: Es un símbolo (raya abajo).
* **NEMONICO\_TABLA:** En lugar de colocar el nombre de la tabla con su nomenclatura especifica, se utiliza un nemónico o palabra clave para un mayor entendimiento.
* **\_**: Es un símbolo (raya abajo).
* **NEMONICO\_COLUMNA:** En lugar de colocar el nombre del campo con su nomenclatura especifica, se utiliza un nemónico o palabra clave para un mayor entendimiento.

Por ejemplo:

ALTER TABLE ADV\_T\_INMUEBLE

{

ADD CONSTRAINT DF\_INM\_AREAT

DEFAULT 0.00

FOR inmu\_c\_earea\_total

}

1. **Check**

Un Check seguirá la siguiente nomenclatura:

**CK\_NEMONICO\_TABLA\_NEMONICO\_COLUMNA**

Donde:

* **CF**: prefijo que denomina a un Default.
* **\_**: Es un símbolo (raya abajo).
* **NEMONICO\_TABLA:** En lugar de colocar el nombre de la tabla con su nomenclatura especifica, se utiliza un nemónico o palabra clave para un mayor entendimiento.
* **\_**: Es un símbolo (raya abajo).
* **NEMONICO\_COLUMNA:** En lugar de colocar el nombre del campo con su nomenclatura especifica, se utiliza un nemónico o palabra clave para un mayor entendimiento.

Por ejemplo:

ALTER TABLE ADV\_T\_INMUEBLE

{

ADD CONSTRAINT DC\_INM\_ABREV

CHECK (LEN(inmu\_c\_cabrev\_inmu) = 4)

}

1. **DEFINICION DE INDICE**

El nombre de un Índice seguirá la siguiente nomenclatura:

**IDX\_ XXX\_T\_NOMBRE\_TABLA\_CORRELATIVO**

Donde:

* **IDX**: prefijo que denomina a un Índice.
* **\_**: Es un símbolo (raya abajo).
* **XXX\_T\_NOMBRE\_TABLA:** este punto está definido en el tema de definición de tablas.
* **\_**: Es un símbolo (raya abajo).
* **CORRELATIVO:** número correlativo que indicara el orden del índice.

Por ejemplo:

CREATE INDEX IDX\_ADV\_T\_INMUEBLE\_01

ON ADV\_T\_INMUEBLE (inmu\_c\_iid\_inmu ASC, inmu\_c\_vnomb\_inmu DESC)

1. **DEFINICION DE VARIABLES Y PARAMETROS**

Una variable o parámetro debe seguir la siguiente nomenclatura:

**@XY\_NOMBRE**

Donde:

* **@:** símbolo ‘@’ (arroba) es obligatorio anteponerlo por restricciones propias del manejador de Base de datos.
* **X**: Letra mayúscula que identifica el tipo de objeto que se desea declarar.
  + **‘**P’: Parámetros
  + ‘L’: Variables locales.
* **Y:** Letra mayúscula que identifica el tipo de dato del objeto. Considerar la Tabla 1.
* **\_:** Es un símbolo (raya abajo).
* **NOMBRE:** Se detalla el nombre del objeto, ya sea parámetro o variable, todo en mayúsculas.

Por ejemplo:

@PV\_NOMB\_INM

@LI\_CANT\_FACT

1. **DEFINICION DE PROCEDIMIENTO ALMACENADO**

El nombre de un procedimiento almacenado seguirá la siguiente nomenclatura:

**XXX\_SP\_NEMONICO\_ACCION**

Donde:

* **XXX:** Son 3 letras mayúsculas e indica el sistema o subsistema al que pertenece.
* **\_**: Es un símbolo (raya abajo).
* **SP**: prefijo que denomina a un procedimiento almacenado.
* **\_**: Es un símbolo (raya abajo).
* **NEMONICO:** Nombre de la tabla o modulo al cual hace referencia, es necesario que el nemónico denote un nombre significativo para el mejor manejo y ubicación de procedimiento almacenados. Se puede utilizar el guión bajo (\_) para nemónico que representen palabras compuestas.
* **\_**: Es un símbolo (raya abajo).
* **ACCION:** Se detalla la acción que se ejecutará en el procedimiento almacenado. **Considerar la Tabla 2.**
* El nombre del Procedimiento debe ser singular y en mayúscula.

**Estructura del procedimiento almacenado:**

* Cabecera: Todos los procedimientos almacenados deberán tener los siguientes comentarios en la cabecera.
* Procedimiento : Nombre del procedimiento
* Parámetros : Descripción de Parámetros de entrada y salida
* Descripción : Descripción / Objetivo de funcionalidad del procedimiento.
* Retorno : N/A – Tipo de Dato que devuelve. Ejemplo: String, Table,
* Notas : N/A
* Autor : Nombre y Apellido del autor del Procedimiento
* Fecha y Hora : Fecha y hora de Creación/Modificación de Cambios Importantes
* Modificaciones : Autor, Fecha, Descripción de Modificaciones.
* Declaración de Parámetros
* Parámetros
* Declaración Variables locales
* Mayúsculas y minúsculas
* Sentencias SQL
* Palabras del lenguaje SQL, y funciones de sistema en MAYUSCULAS.
* Sentencias legibles e indentadas (cada cláusula SQL en una línea nueva).

Por ejemplo:

CREATE PROCEDURE ADV\_SP\_INMUEBLE\_ELIMINAR

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Procedimiento : ADV\_SP\_INMUEBLE\_ELIMINAR

Parámetros :

@PV\_NOMB\_INM VARCHAR(100) IN

Descripción : Elimina un inmueble de la base de datos

Retorno : N/A

Notas : N/A

Autor : Cristhian Yesquen

Fecha y hora : 28/08/2012 - 04:34 pm.

Modificaciones : N/A

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

<Declaración de Parámetros Input y/o Output>

AS

<Declaración de Variables locales>

<Sentencias SQL>

GO

|  |  |
| --- | --- |
| **REPRESENTACION DEL ESTANDAR** | **ACCIÓN** |
| INSERTAR | Insertar |
| ELIMINAR | Eliminar |
| ACTUALIZAR | Actualizar |
| SELECCIONAR | Seleccionar un solo registro |
| LISTAR | Listar varios registros sin filtros |
| BUSCAR | Buscar varios registros en funcion a filtros |

**Tabla 2 – Definición de posibles acciones de los procedimientos almacenados y funciones.**

1. **DEFINICION DE TRIGGER**

Un trigger debe seguir la siguiente nomenclatura:

**TR\_XXX\_T\_NOMBRE\_TABLA\_YYY\_CORRELATIVO**

Donde:

* **TR**: prefijo que denomina a un Trigger.
* **\_**: Es un símbolo (raya abajo).
* **XXX\_T\_NOMBRE\_TABLA**: este punto está definido en el tema de definición de tablas.
* **\_**: Es un símbolo (raya abajo).
* **YYY**: Letras mayúscula que representa el tipo de lógica del trigger. Puede ser una letra, o más, dependiendo si el trigger está diseñado para realizar más de un tipo de acción (insert, update,select**). Considerar Tabla 3.**
* **\_**: Es un símbolo (raya abajo).
* **CORRELATIVO:** número correlativo que indicara el orden del trigger.

Por ejemplo:

CREATE TRIGGER TR\_ADV\_T\_INMUEBLE\_I\_01

ON ADV\_T\_INMUEBLE

FOR INSERT

AS

BEGIN

.

.

.

END

|  |  |
| --- | --- |
| **CARÁCTER** | **TIPO DE LOGICA** |
| I | INSERT |
| U | UPDATE |
| D | DELETE |

**Tabla 3 – Letras para definir el tipo de lógica del trigger**

1. **DEFINICION DE CURSOR**

Un cursor debe seguir la siguiente nomenclatura:

**CUR\_NOMBRE\_CURSOR**

Donde:

* **CUR**: prefijo que denomina a un Cursor
* **\_**: Es un símbolo (raya abajo).
* **NOMBRE\_CURSOR** identifica el nombre del cursor, debe ser todo en mayúsculas y en singular. Para separar palabras se debe usar el símbolo “\_”.

Por ejemplo:

-- Se declara un cursor para almacenar los inmuebles que existen en Lima

DECLARE CUR\_INM\_LIMA

CURSOR FOR

SELECT …

FROM …

WHERE …

1. **DEFINICION DE VISTA**

Una vista debe seguir la siguiente nomenclatura:

**XXX\_VW\_NEMONICO\_ACCION**

Donde:

* **XXX:** Son 3 letras mayúsculas e indica el sistema o subsistema al que pertenece.
* **\_**: Es un símbolo (raya abajo).
* **VW**: prefijo que denomina a una vista.
* **\_**: Es un símbolo (raya abajo).
* **NEMONICO:** Nombre de la tabla(s) o modulo al cual hace referencia, es necesario que el nemónico denote un nombre significativo para el mejor manejo y ubicación de procedimiento almacenados. Se puede utilizar el guión bajo (\_) para nemónico que representen palabras compuestas.
* **\_**: Es un símbolo (raya abajo).
* **ACCION:** Se detalla la acción que se visualizara en la vista.  
   **Considerar la Tabla 2.**

**Estructura de la vista:**

* Cabecera: Todos las vistas deberán tener los siguientes comentarios en la cabecera
* Vista : Nombre de la vista.
* Descripción : Descripción / Objetivo de la vista.
* Autor : Nombre y Apellido del autor de la vista.
* Fecha y Hora : Fecha y hora creación o modificaciones.

Modificaciones : Autor, Fecha, Descripción de Modificaciones.

* + Sentencias SQL
* Palabras del lenguaje SQL, y funciones de sistema en MAYUSCULAS.
* Sentencias legibles e indentadas (cada cláusula SQL en una línea nueva)

Por ejemplo:

CREATE VIEW ADV\_VW\_INM\_LISTAR

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Vista : ADV\_VW\_INM\_LISTAR

Descripción : Muestra el listado de los inmuebles

Notas : N/A

Autor : Cristhian Yesquen

Fecha y hora : 20/01/2006 - 04:34 pm.

Modificaciones : N/A

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

AS

<Sentencias SQL>

GO

**BIBLIOGRAFIA**

* *Estándar Base de Datos v0.3 - Jonnathan De La Barra*