

ESTÁNDAR DE PROGRAMACIÓN DE BASE DE DATOS

1. Definición General de Objetos de BD:

Todos los objetos de la BD deben tener un prefijo correspondiente a la sigla del sistema (de 3 a 5 caracteres), más el tipo de objeto (de tamaño 2).

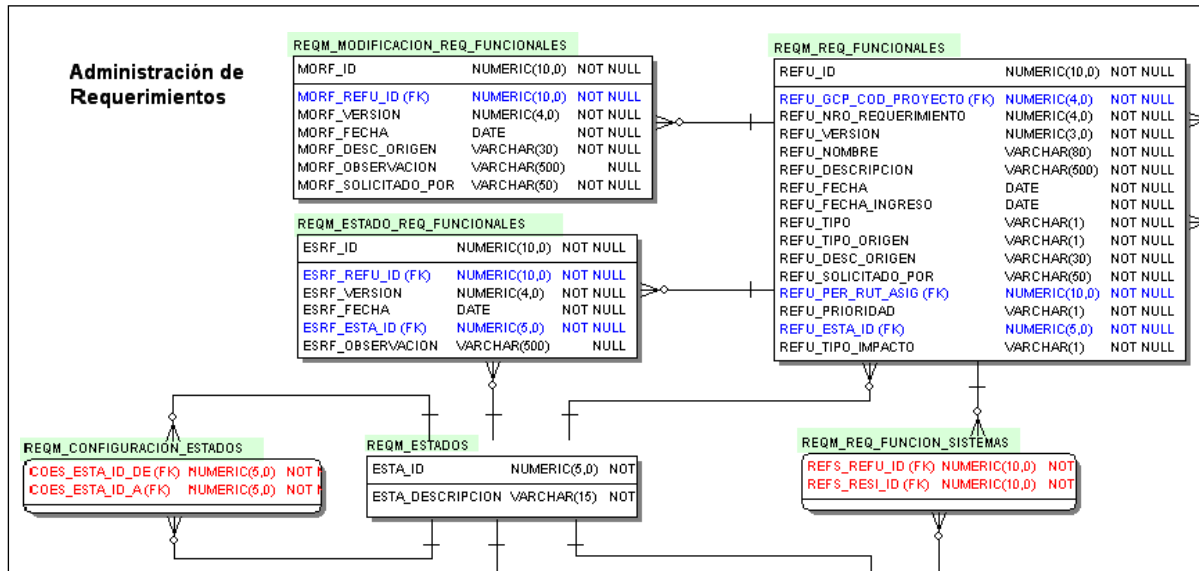
Ejemplo: Sistema de Administración de Requerimientos **Prefijo => REQM**

Sintaxis de Objetos:

Formato	<Prefijo>_<ID>_<Nombre>		
Objeto	ID	Ejemplo	Observación
Procedure	SP	REQM_SP_CARGA_DATOS	
Function	FC	REQM_FC_VALIDA_DV_RUT	
Package	PK	REQM_PK_ING_DOCTOS	Los SP y FC contenidas no llevarán prefijo (solo <ID>_<Nombre>): SP_PRUEBA, FC_VALIDA
Sequence	SQ	REQM_SQ_ID_MOVTO	Si se asocia a la PK de una tabla en <Nombre> usar prefijo de ella, ejemplo: REQM_SQ_MOMA (ver pto 2.a)
View	VW	REQM_VW_MOVIMIENTOS	
Table	-	REQM_REQUERIMIENTOS	Sin ID (ver pto 2).

ESTÁNDAR DE PROGRAMACIÓN DE BASE DE DATOS

2. Definición de Tablas y Objetos Relacionados



- a. Definir prefijo para identificar cada tabla (de 3 a 5 caracteres), se utilizará para la definición de sus objetos relacionados.

Ejemplo:

Tabla	Prefijo
REQM_REQ_FUNCIONALES	REFU
REQM_MODIFICACION_REQ_FUNCIONALES	MORF
REQM_ESTADO_REQ_FUNCIONAL	ESRF
REQM_ESTADOS	ESTA
REQM_CONFIGURACION_ESTADOS	COES
REQM_REQ_FUNCION_SISTEMAS	REFS

- b. COLUMNAS: <PrefijoTabla>_<DescColumna> (tb en vistas)

Ejemplo: Para REQM_REQ_FUNCIONALES => REFU_NOMBRE

ESTÁNDAR DE PROGRAMACIÓN DE BASE DE DATOS

REQM_REQ_FUNCIONALES		
REFU_ID	NUMERIC(10,0)	NOT NULL
REFU_GCP_COD_PROYECTO (FK)	NUMERIC(4,0)	NOT NULL
REFU_NRO_REQUERIMIENTO	NUMERIC(4,0)	NOT NULL
REFU_VERSION	NUMERIC(3,0)	NOT NULL
REFU_NOMBRE	VARCHAR(80)	NOT NULL
REFU_DESCRIPCION	VARCHAR(500)	NOT NULL
REFU_FECHA	DATE	NOT NULL
REFU_FECHA_INGRESO	DATE	NOT NULL
REFU_TIPO	VARCHAR(1)	NOT NULL
REFU_TIPO_ORIGEN	VARCHAR(1)	NOT NULL
REFU_DESC_ORIGEN	VARCHAR(30)	NOT NULL
REFU_SOLICITADO_POR	VARCHAR(50)	NOT NULL
REFU_PER_RUT_ASIG (FK)	NUMERIC(10,0)	NOT NULL
REFU_PRIORIDAD	VARCHAR(1)	NOT NULL
REFU_ESTA_ID (FK)	NUMERIC(5,0)	NOT NULL
REFU_TIPO_IMPACTO	VARCHAR(1)	NOT NULL

c. PRIMARY KEY: <PrefijoTabla>_PK

Ejemplo: Para REQM_REQ_FUNCIONALES => REFU_PK

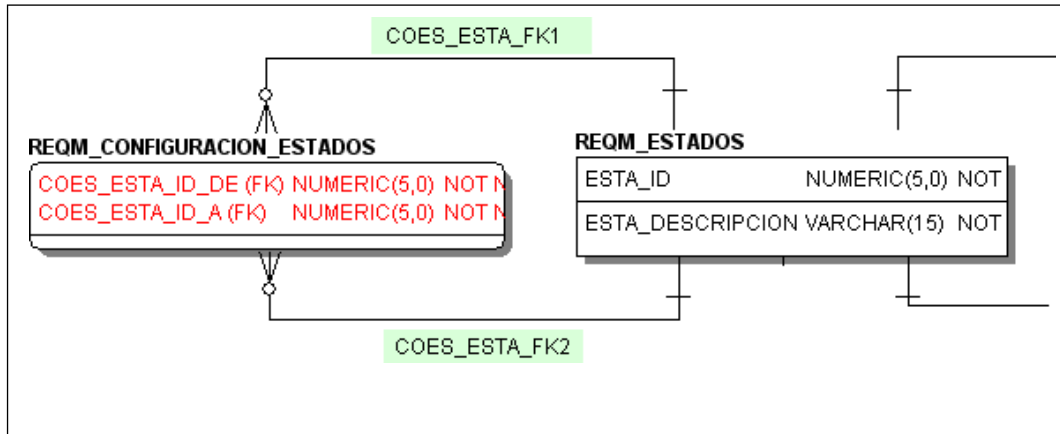
Attributes	Keys	Relationships
Define alternate keys and inversion entries below. Use Attributes tab for		
Name	Type	
REFU_PK	Primary Key	REFU_ID

d. FOREIGN KEY: <PrefijoTablaHija>_<PrefijoTablaPadre>_FK<nro>

Ejemplo: Para REQM_CONFIGURACION_ESTADOS => COES_ESTA_FK1 y COES_ESTA_FK2

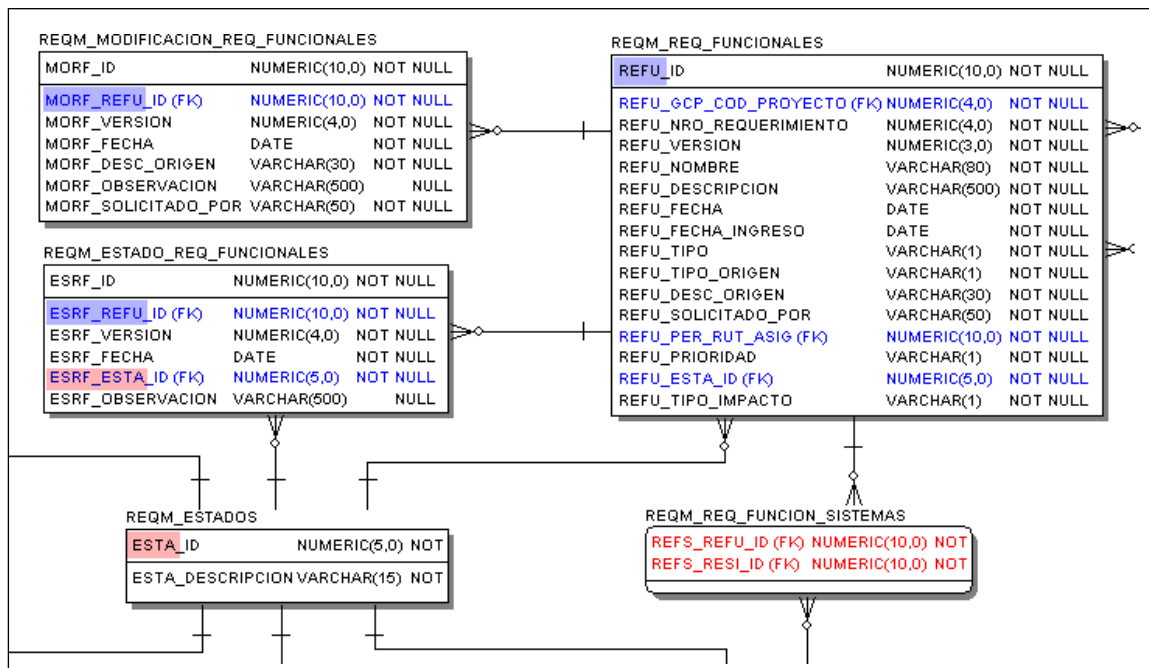
Nota: Tenga o no, una o más referencias.

ESTÁNDAR DE PROGRAMACIÓN DE BASE DE DATOS



- Para el caso de las columnas con FOREIGN KEY deben definirse:

<PrefijoTablaHija>_<PrefijoTablaOrigen>_<DescAtributo>

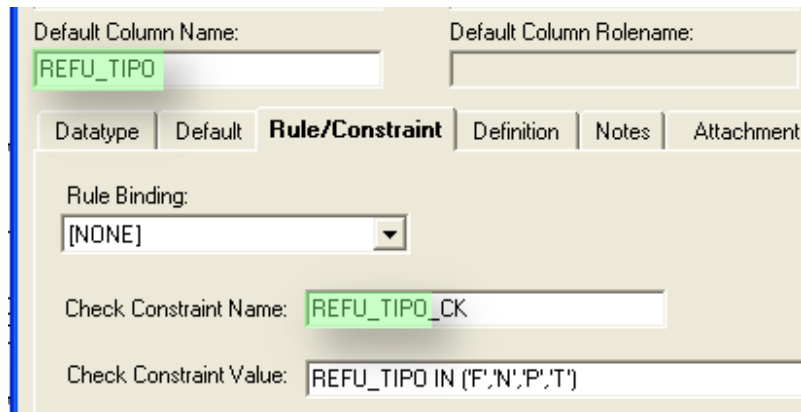


- Los FOREIGN KEY deben generar índices en tablas hijas. El nombre del índice asociado a una FK es **<NombreFK>_I**

Ejemplo: Para la FK MORF_REFU_FK1 el nombre del índice es MORF_REFU_FK1_I

ESTÁNDAR DE PROGRAMACIÓN DE BASE DE DATOS

e. CHECK CONSTRAINT: <Atributo>_CK



Default Column Name: REFU_TIPO

Default Column Rolename:

Datatype | Default | **Rule/Constraint** | Definition | Notes | Attachment

Rule Binding: [NONE]

Check Constraint Name: REFU_TIPO_CK

Check Constraint Value: REFU_TIPO IN ('F','N','P','T')

f. INDICES: <PrefijoTabla>_IDX<nro>

Ejemplo: Para REQM_REQ_FUNCIONALES => REFU_IDX01

g. TRIGGERS: <PrefijoTabla>_TRG<nro>

Ejemplo: Para REQM_REQ_FUNCIONALES => REFU_TRG01

ESTÁNDAR DE PROGRAMACIÓN DE BASE DE DATOS

3. Definición Objetos de Programación

a. Uso de Variables:

Prefijo	Tipo
VN_	variable numérica local
VC_	variable alfanumérica local
VD_	variable de tipo fecha local
VB_	variable booleana local

Las variables relacionadas con campos de la base de datos conservan el nombre del campo anteponiendo el prefijo VN_...ó VV_... etc., según corresponda.

b. Uso de Constantes:

Prefijo	Tipo
CN_	constante numérica local
CC_	constante alfanumérica local
CD_	constante de tipo fecha local
CB_	constante booleana local

c. Uso de Parámetros:

Aplicable a Parámetros pasados entre Procedimientos o Funciones.

Prefijo	Tipo
PN_	parámetro numérico
PV_	parámetro alfanumérico
PD_	parámetro de tipo fecha
PB_	parámetro booleano

d. Documentación del Código

Para triggers, procedures, functions, packages se requiere documentar el código indicando el propósito que cumple la unidad de programa y las modificaciones realizadas, esta documentación irá al inicio de la unidad de programa de la sgte manera:

ESTÁNDAR DE PROGRAMACIÓN DE BASE DE DATOS

Procedure <Nombre> is

/*****

Autor: <Nombre Programador>

Fecha: <99/99/9999>

Objetivo: <Breve descripción del objetivo>

Referencia: <N°Soporte/OT>

Modificaciones:

1. <Autor>, <Fecha> (Referencia):
<Breve descripción de la modificación>
- 2.

*****/

Begin

<líneas de código> --(M.<N°Modificación>)

End;

Ejemplo:

Procedure REQM_CARGA_DATOS is

/*****

Autor: Juan Pérez

Fecha: 02/03/2010

Objetivo: Carga los requerimientos funcionales y de sistema definidos en proyecto Análisis a proyecto de Construcción.

Referencia: RE156231

Modificaciones:

1. B Vásquez, 05/05/2010 (RE231452):
Reinicia el estado de los requerimientos a "Aceptado", antes los traspasaba en el estado que se encontraban.
- 2.

*****/

Begin

```
...
INSERT INTO REQM_REQ_FUNCIONALES
  (REFU_ID, REFU_GCP_COD_PROYECTO, REFU_NRO_REQUERIMIENTO, REFU_VERSION,
  REFU_NOMBRE, REFU_DESCRIPCION, REFU_FECHA, REFU_TIPO, REFU_FECHA_EDO)
SELECT FC_CONS_SEQ_REQ_FUNC, P_ID_PROY_NVO, REFU_NRO_REQUERIMIENTO, REFU_VERSION,
REFU_NOMBRE, REFU_DESCRIPCION, REFU_FECHA, REFU_TIPO, 'A'      --(M.1)
FROM REQM_REQ_FUNCIONALES
WHERE REFU_GCP_COD_PROYECTO = P_ID_PROY_ORIG
ORDER BY REFU_NRO_REQUERIMIENTO;
...

```

End;

ESTÁNDAR DE PROGRAMACIÓN DE BASE DE DATOS

4. Consideraciones Generales

- No pueden definirse PK compuestas con un número de campos mayor a 6, si se requiere debe ser con el v°b° de otro JP que designe el Subgerente de Consultoría.
- En general evitar campos tipo CHAR, en particular NO usar este tipo de datos en PK, si se requiere debe ser con el v°b° de otro JP que designe el Subgerente de Consultoría.
- ***Si el desarrollo es en Oracle, para la implementación de funcionalidades (opción de menú) del sistema, se deben utilizar “package” para la implementación de procedimientos almacenados. Por otra parte, los packages deben ser creados uno por funcionalidad.***