# Estándar SQL; Creación de tablas y poblamiento

Se detallan las siguientes normas:

* Creación de tablas: Sintaxis SQL para la creación de tablas. Sera ingresada en la opción SQL de PgAdmin.
* Convención de nombres: Estándar de notación de atributos, y entidades.
* Poblamiento de la BD: Sintaxis SQL para la inserción de datos en la BD;
* Consideraciones de integridad referencial: Propiedades que deberán ser agregadas a la creación de tablas para permitir la integridad referencial.
* Ejemplo

## Creación de tablas

create table <nombre tabla> (

<nombre atributo> <tipo> <propiedad>,

<Otro nombre atributo> <tipo><propiedad>,

Constraint PK\_<nombre tabla> primary key (<nombre atributo>) < propiedad >

<Otro nombre atributo foráneo> <tipo> references <nombre tabla> (<nombre clave foránea >) <Propiedades de integridad referencial> );

* Nombre tabla: Nombre de la entidad, primera letra en mayúscula (Ver anexo)
* Nombre atributo: Minúscula, si es palabra compuesta separada por Primera letra de la segunda palabra en mayúscula.
* Tipo: Pueden ser:
  + bool
  + serial : Este tipo de datos permite que un atributo se incremente de forma automática en una unidad, Es utilizado para los identificadores.
  + char(n)
  + date
  + money
  + time
  + varchar(n)

Para ver otros tipos consultar <http://www.postgresql.org/docs/7.4/interactive/datatype.html> columna “aliases”

Para saber los valores

* Propiedad: (Ver anexo)
  + NOT NULL
  + NULL
  + Propiedad de integridad referencial: Esto es aplicado solo a las tablas que contienen clave foránea, y es aplicado para mantener la integridad de una tabla ante atributos apuntados desde otras tablas y ante actualizaciones de las claves foráneas.

La sentencia que se utilizara es la siguiente:

MATCH FULL ON DELETE RESTRICT ON UPDATE CASCADE

**Consideración integridad referencial:**

Deberán ser creadas primero las tablas que no contienen claves foráneas. Es decir se debe resolver antes la dependencia.

## Poblamiento BD

insert into <nombre tabla> (<nombre atributo>, <nombre atributo>)

values ('<valor>',’ <valor>’);

* Nombre tabla: Nombre de la entidad, primera letra en mayúscula (Ver anexo)
* Nombre atributo: Minúscula, si es palabra compuesta separada por Primera letra de la segunda palabra en mayúscula.
* Valor: Dato que será ingresado a la BD, este valor debe ser del mismo tipo del que fue declarado en la creación de la tabla.

**Consideración integridad referencial:**

Si el valor de la clave foránea no existe en la tabla, es decir no fue previamente insertado, el sistema de BD arrojara un problema de ejecución tal como el siguiente:

ERROR: insert or update on table "<nombre table>" violates foreign key constraint "<nombre atributo foraneo>"

## Ejemplo

Ejemplo de creación de tablas

create table ciudad (

nombre varchar(80) primary key,

ubicacion point

);

create table tiempo (

ciudadNombre varchar(80) references ciudad(nombre) MATCH FULL ON DELETE RESTRICT ON UPDATE CASCADE,

temperatura int

);

INSERT INTO ciudad (nombre, ubicacion)

VALUES ('Curico', '12,30');

# Anexo

**Notacion**

* Los atributos en azul: Indican clave foránea
* Los atributos subrayados: Indican clave primaria.
* Los atributos en paréntesis: Indican el valor por defecto del atributo que está a la izquierda.
* En paréntesis cuadrados: Tipos de datos de cada atributo.

Usuario(nombre[char],

aMaterno[char],

aPaterno[char],

usuario[char],

idUsuario[serial],

contrasena[char],

cargo[char],

pRegistrar[bool],

pEditar[bool],

pEliminar[bool],

pPurgar[bool],

servicioNombre[char]

)

Servicio(nombre[char],

descripcion[char]

)

Cita(fecha[date],

horaInicio[time],

servicioNombre[char],

clienteRut [int]

)

ClienteWeb(rut[int],

telefono1[int],

telefono2[int],

nombre[char],

correo[char],

contraseña[char]

)

Aviso(descripcion[char],

fecha[date],

hora[time],

clienteWebRut[int]

)

ClientePresencial (rut[int],

telefono1[int],

telefono2[int],

nombre[char],

correo[char],

estado[bool],

domicilioNumero[int] ,

domicilioComuna[char],

domicilioCalle[char]

)

Producto(nombre[char],

Precio[Money],

descripcion[char],

categoria[char],

clientePresencialRut[int],

codigoBarra[int]

)

Mascota(rutCliente[char],

nombre[char],

fechaNacimiento[char],

chase[char],

raza[char],

sexo[char],

estado[bool],

foto[abytes]

)

Atencion(clienteRut[int],

mascotaNombre[char],

mascotaFechaNacimiento[date],

servicio[char],

hora[time],

fecha[date]

)

AtencionPostOperatorio(medicamento[char],

alimento[char],

indicaciones[char],

servicio(pabellon) [char],

hora[time],

fecha[date]

)

AtencionPreOperatorio(observaciones[char],

sintomas[char],

diagnosticos[char],

servicio(pabellon) [char],

hora[time],

fecha[date]

)

AtencionCirugia(estado[bool],

diagnostico[char],

tipo[char],

servicio(pabellon) [char],

hora[time],

fecha[date]

)

AtencionConsulta(anamnesis,

servicio(policlinico) [char],

hora[time],

fecha[date]

)

AtencionControl(descripción[char],

recomendaciones[char],

proximaFecha[char],

servicio(policlinico)[char],

hora[time],

fecha[date]

);

AtencionVacuna(precio[money,

nombre[char],

descripcion[char],

fechaCaducidad[date],

servicio(policlinico) [char],

hora[time],

fecha[date]

);

AtencionAlojamiento(canil[char],

fechaSalida[date],

comentario[char],

solicitudes[char],

servicio(hoteleria) [char],

hora[time],

fecha[date]

);

AtencionBaño(tipo,

comentario[char],

servicio(peluqueria) [char],

hora[time],

fecha[date]

);

AtencionCorte(categoría[char],

nombre[char],

descripcion[char],

raza[char],

foto,

servicio(peluqueria) [char],

hora[time],

fecha[date]

);