UADER

Facultad de Ciencia Y Tenología

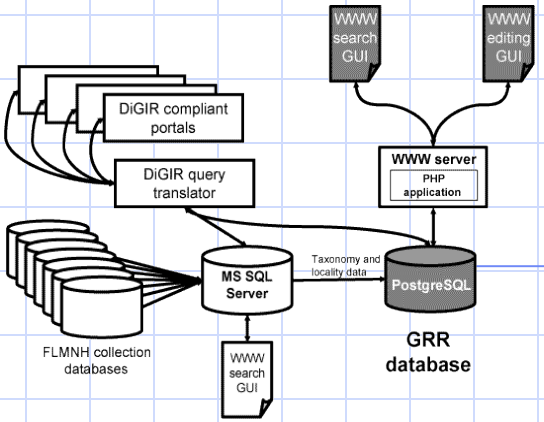
Base de Datos Avanzadas

Base de datos Avanzadas

Repaso Modelo Relacional

Metodologías asociadas

SQL DDLs



Ing. Fernando Sato
fsatopna@gmail.com

Ultima Actualización: 26/03/2018

Sub Lenguaje DDL

Componente de SQL:

- DML (Data Manipulation Language).
- **DDL (Data Definition Language).**
- DCL (Data Control Language).

Permite crear, borrar y en algunos casos actualizar objetos de la base de datos tales como:

Dominios	*** Tablas**	* Vistas
Indices	*Triggers	* Procedure (SP)
Usuarios	* Roles	* y otros

- Incluso una base o schema en si mismo

Estructura sintáctica de DDL

Esta formada por 3 sentencias:

- Create
- Drop
- Alter

Estructura sintáctica - CREATE

CREATE tipo_objeto nombre_objeto
especifiaciones_del_objeto.;

Tipos_Objetos

- **Tablas** *** Vistas** *** etc**

Nombre Objetos:

- Hasta 32 caracteres (letras Nros y “_”)
- Deben comenzar con una letra
- No pueden ser palabras reservadas.

Estructura sintáctica - DROP

Drop tipo_objeto nombre_objeto;

Tipos_Objetos

- **Tablas * Vistas * etc**

Estructura sintáctica - ALTER

Alter tipo_objeto nombre_objeto especifiaciones_de_la_modificación;

Tipos_Objetos

- **Tablas * Vistas * etc**

Dominios CREATE

Un Dominio es un tipo de dato definido por el usuario el cual podemos usar como tipo de dato de una columna.

Sintaxis

```
CREATE DOMAIN nombre_dominio [AS] <tipo_dato>
 [DEFAULT {literal | NULL | USER}]
 [NOT NULL]
 [CHECK (<condicion>)];
```

Ej:

```
CREATE DOMAIN D_MES
      AS INTEGER
       CHECK ( VALUE < 13 );
```

Dominios DROP

Un Dominio no se puede eliminar si esta siendo usado en alguna tabla.

Sintaxis

DROP DOMAIN nombre_dominio;

Ej: DROP DOMAIN D_MES;

Dominios ALTER

Sintaxis

ALTER DOMAIN nombre_dominio {
SET DEFAULT {literal | NULL | USER}
| DROP DEFAULT
| ADD [CONSTRAINT] CHECK (<condición>)
| DROP CONSTRAINT | TYPE tipo_dato};;

Ej:

ALTER domain D_MES DROP CONSTRAINT;

ALTER domain D_MES
ADD CHECK (VALUE BETWEEN 1 AND 12);

;

Tablas

- Constituyen la estructura fundamental de todas las bases de datos.
- Implementan las *Relaciones Base, Temporales y Snapshot.*
- Único objeto que contiene “datos” o “información”.
- Su estructura esta formada por un conjunto dinámico de columnas.
- La información se “guarda” en un conjunto dinámico de filas.
- La información del metadato de una base, se “guarda “ en tablas especiales denominadas tablas del sistema.

Tipos de Tablas

- Tablas de Usuario “Normales”.
- Tablas Temporales. postgerSQL TEMPORARY
- Tablas SnapShot.
- Tablas del Sistema.

Estructura de Una tabla

- Una tabla esta definida por:
  - Un nombre (único para toda la base, incluyendo las vistas).
  - Un conjunto de pares (nombre_atributo, tipo_dato) <Conj Columnas>
  - La clave primaria. *Opcional*.
  - Las claves ajenas. *Opcional*.
  - Las restricciones a imponer a nivel columna y fila. *Opcional*.

```
create table productos
  ( codigo integer not null primary key,
    nombre  varchar(50) unique,
    abreviatura char(6)  );
```

Tipos Primitivos (I)

- **Caracteres:**
  - chararacter(*n*), char(n)
  - character varying(n) varchar(*n*)
- **Numéricos:**
  - integer, Smallint, Double precision,
  - numeric(p,d),
  - Real, float(n)

Tipos Primitivos (II)

- **Fechas:**
  - Date Time TimeStamp
- **Objetos Grandes:**
  - Blob
    - bytea[] Blob: Binarios Largos.
    - text[] Clob: Textos Largos.

Tipo Primitivo (III) - Especial

- **Autoincrementales:**
  - Serial :
    - Este tipo crea automáticamente un objeto secuencia asociado.

```
Create Table nombre (
 id serial,
Nombre varchar(50));
```

Equivale a:

```
Create Table nombre (
 id integer default  DEFAULT
  nextval('table_numero_seq'::regclass),
Nombre varchar(50));
```

Table CREATE

Sintaxis

CREATE [[GLOBAL | LOCAL] { TEMPORARY | TEMP } | UNLOGGED] TABLE [IF NOT EXISTS] table_name ([{

column_name data_type [COLLATE collation]

[column_constraint [...]]

| table_constraint

| LIKE parent_table [like_option ...] } [, ...]])

[INHERITS (parent_table [, ...])]

[WITH (storage_parameter [= value] [, ...])

| WITH OIDS

| WITHOUT OIDS]

[ON COMMIT { PRESERVE ROWS | DELETE ROWS | DROP }] [TABLESPACE tablespace]

Ej

```
create table productos
( codigo integer not null primary key,
  nombre  varchar(50)unique, abreviatura char(6) );
```

Table CREATE (column_constraint)

Constraint de Columnas (column_constraint)

[CONSTRAINT constraint_name]
{ NOT NULL | NULL |
CHECK (expression) | DEFAULT default_expr |
UNIQUE index_parameters | PRIMARY KEY index_parameters|
REFERENCES reftable [(refcolumn)] [MATCH FULL | MATCH PARTIAL | MATCH SIMPLE] [ON DELETE action] [ON UPDATE action] }

Table CREATE (table_constraint)

Constraint de Tabla (table_constraint)

[CONSTRAINT constraint_name]
{ CHECK (expression) |
UNIQUE (column_name [, ...]) index_parameters |
PRIMARY KEY (column_name [, ...]) index_parameters |
FOREIGN KEY (column_name [, ...])
REFERENCES reftable [(refcolumn [, ...])] [
MATCH FULL | MATCH PARTIAL | MATCH SIMPLE]
[ON DELETE action] [ON UPDATE action] }

Table CREATE TEMPORARY

Sintaxis

CREATE TEMPORARY TABLE nombre_tabla

idem tablas tradicionales…

Ej

```
create temporary table tempproductos
( codigo integer not null primary key,
  on commit preserve rows
 );
```

Table DROP

Sintaxis

DROP TABLE nombre_tabla;

Ej: DROP TABLE productos_back;

Table ALTER (I)

Sintaxis

ALTER TABLE table <operation> [, <operation> ...];

<operation> = {ADD <col_def>

| ADD <tconstraint>

| ALTER [COLUMN] column_name <alt_col_clause>

| DROP col

| DROP CONSTRAINT constraint}

<alt_col_clause> = {TO new_col_name

| TYPE new_col_datatype

| POSITION new_col_position};

Table ALTER (II)

Ejemplos

```
ALTER TABLE productos
    ADD precio DECIMAL(6,2),
    DROP abreviatura;
```

```
ALTER TABLE productos ALTER precio TO precio_vigente;
```

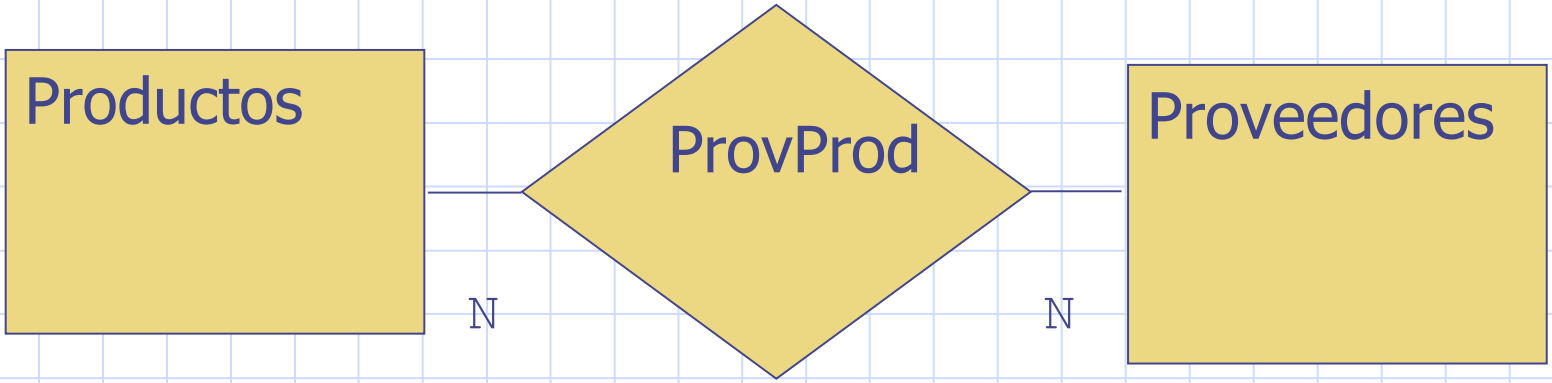
Integridad

Integridad

Claves, Relación con Identificacdores, Tipos, Definición?

Que tipos conoce?

Integridad

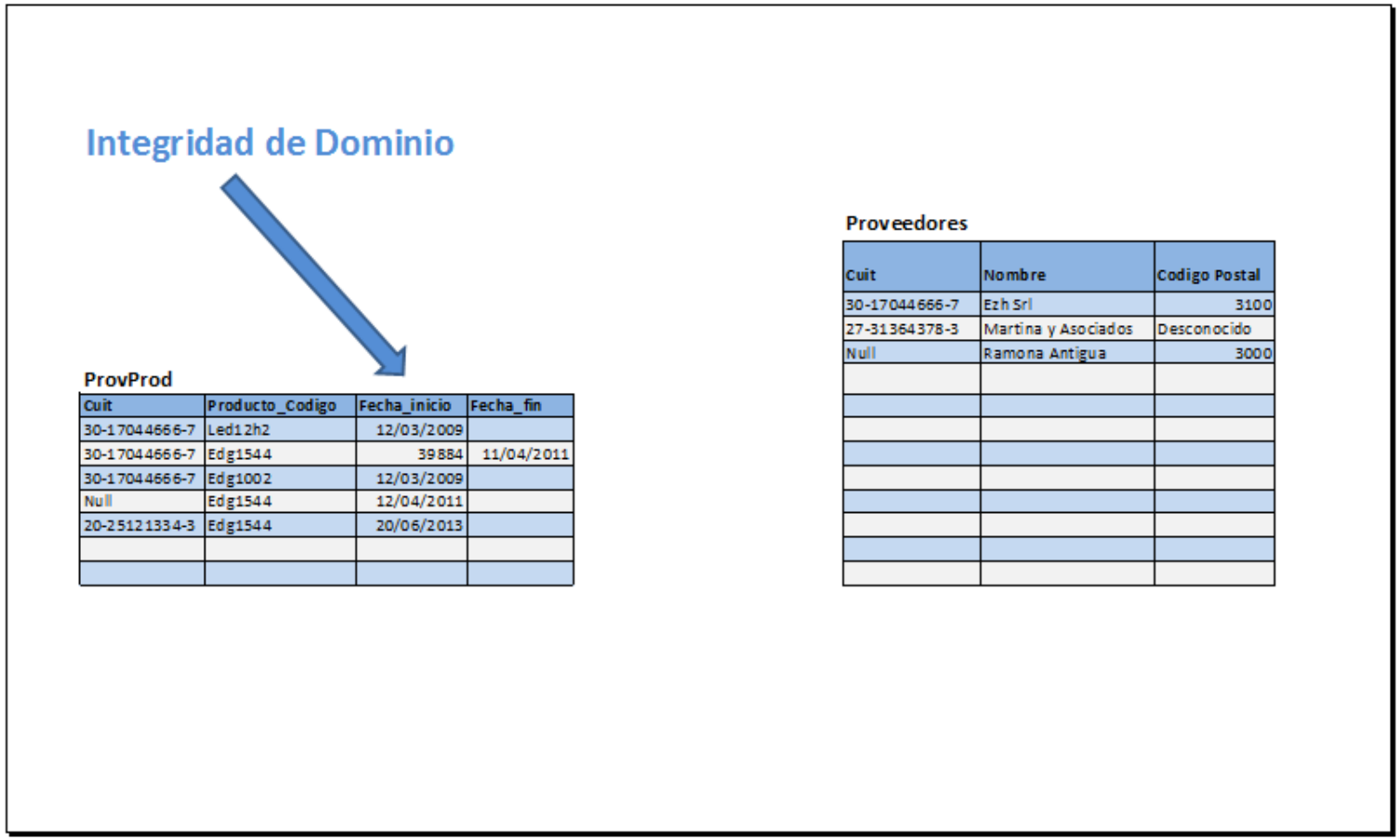


Productos	ProvProd	Proveedores
Codigo	producto_codigo	Cuit
Nombre	cuit	Nombre
Abreviatura	fecha_inicio	Codpos
	fecha_fin	

¿Que restricciones imponemos a este Modelo?

Integridad de Dominio (I)

*La idea es restringir al **dominio real de cada atributo**, consignando la posibilidad de valores nulos incluso.*



ProvProd

Cuit	Producto_Codigo	Fecha_inicio	Fecha_fin
30-17044666-7	Led12h2	12/03/2009	
30-17044666-7	Edg1544	39884	11/04/2011
30-17044666-7	Edg1002	12/03/2009	
Null	Edg1544	12/04/2011	
20-25121334-3	Edg1544	20/06/2013	

Proveedores

Cuit	Nombre	Codigo Postal
30-17044666-7	Ezh Srl	3100
27-31364378-3	Martina y Asociados	Desconocido
Null	Ramona Antigua	3000

Integridad de Dominio (II)

ProvProd

Cuit	Producto_Codigo	Fecha_inicio	Fecha_fin
30-17044666-7	Led12h2	12/03/2009	
30-17044666-7	Edg1544	39884	11/04/2011
30-17044666-7	Edg1002	12/03/2009	
Null	Edg1544	12/04/2011	
20-25121334-3	Edg1544	20/06/2013	

```
Create Table ProvProd (
  Cuit                  Character(11),
  Producto_Codigo       Character Varying(10),
  fecha_inicio          date,
  fecha_fin             date
 );
```

Integridad de Dominio (III)

ProvProd

Cuit	Producto_Codigo	Fecha_inicio	Fecha_fin
30-17044666-7	Led12h2	12/03/2009	
30-17044666-7	Edg1544	59884	11/04/2011
30-17044666-7	Edg1002	12/03/2009	
Null	Edg1544	12/04/2011	
20-25121334-3	Edg1544	20/06/2013	

```
Create Table ProvProd (
  Cuit                  Character(11),
  Producto_Codigo       Character Varying(10),
  fecha_inicio          date
     check(fecha_inicio >= current_date)
     NOT NULL,
  fecha_fin             date
);
```

Integridad de Dominio (IV)

Proveedores

Cuit	Nombre	Codigo Postal
30-17044666-7	Ezh Srl	3100
27-31364378-3	Martina y Asociados	Desconocido
Null	Ramona Antigua	3000

```
Create Table Proveedores(
  Cuit                Character(11),
  Nombre              varchar(60)     Unique,
  CodigoPostal        Integer
 );
```

Integridad a **Nivel Tupla**

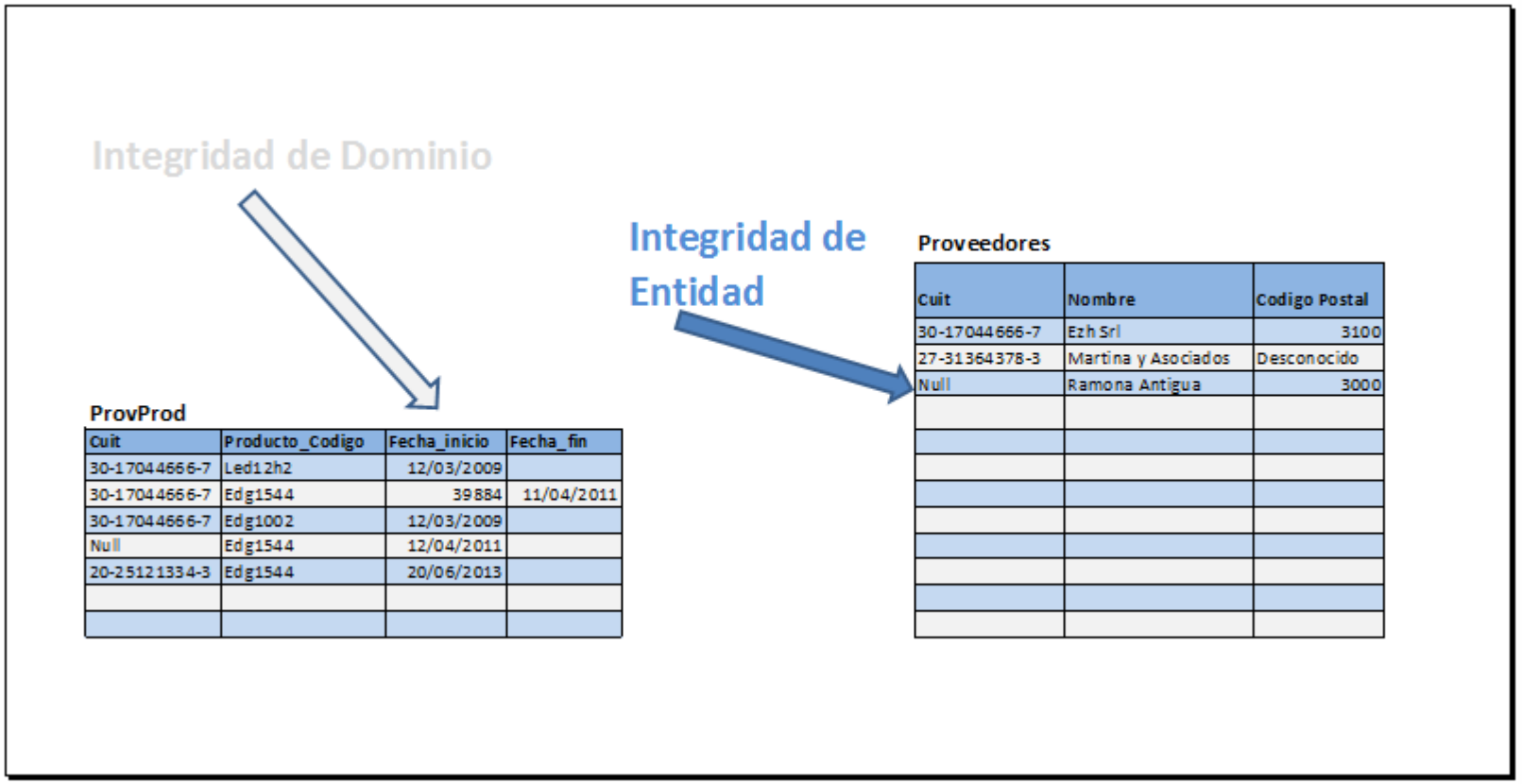
ProvProd

Cuit	Producto_Codigo	Fecha_inicio	Fecha_fin
30-17044666-7	Led12h2	12/03/2009	
30-17044666-7	Edg1544	59884	11/04/2011
30-17044666-7	Edg1002	12/03/2009	
Null	Edg1544	12/04/2011	
20-25121334-3	Edg1544	20/06/2013	

```
Create Table ProvProd (
  Cuit                  Character(11),
  Producto_Codigo       Character Varying(10),
  fecha_inicio          date not null,
  fecha_fin             date,
  check(fecha_inicio <= fecha_fin)
);
```

Integridad de Entidad y PK

- Esta regla de integridad se aplica a las claves primarias de las relaciones base: "*ninguno de los atributos que componen la clave primaria puede ser nulo*".



ProvProd

Cuit	Producto_Codigo	Fecha_inicio	Fecha_fin
30-17044666-7	Led12h2	12/03/2009	
30-17044666-7	Edg1544	39884	11/04/2011
30-17044666-7	Edg1002	12/03/2009	
Null	Edg1544	12/04/2011	
20-25121334-3	Edg1544	20/06/2013	

Proveedores

Cuit	Nombre	Codigo Postal
30-17044666-7	Ezh Srl	3100
27-31364378-3	Martina y Asociados	Desconocido
Null	Ramona Antigua	3000

Integridad de Entidad y PK

Hay 3 formas de definir una clave primaria.

1)Create table tabla (col1 integer not null primary key);

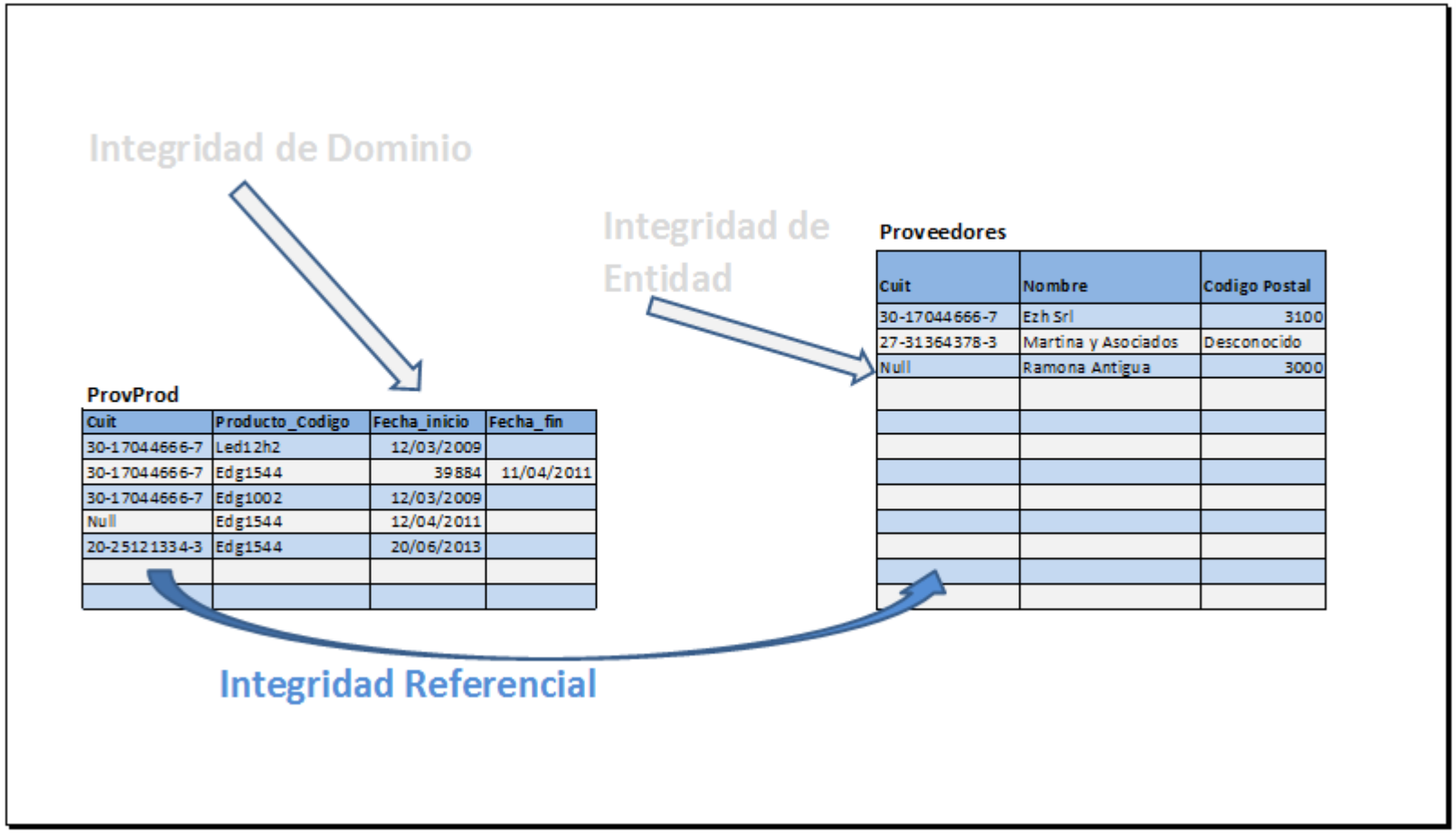
2)Create table tabla (col1 integer not null,
primary key (col1));

3) Luego de creada la tabla

```
Alter table tabla
add constraint pk_table primary key(col1);
```

Integridad Referencial - FK

- La Integridad Referencial se refiere a una relación de dependencia funcional entre dos tablas.



Integridad Referencial – FK (II)

```
ALTER TABLE provprod ADD FOREIGN KEY (Cuit) REFERENCES
   proveedores (cuit)
   ON UPDATE CASCADE
   ON DELETE CASCADE;
```

Importante: Una Clave ajena (FK de Foreign key) tiene todos los atributos no nulos o todos nulos.

Obligatorio: La/s columna/s en la tabla referenciada tienen que ser la PK o deben tener una restricción UNIQUE.

Integridad Referencial – FK (III)

Propagación de actualización:

- NO ACTION **(DEFAULT)**:

Imposibilita la actualización si existen registros que lo referencian por la FK.

- CASCADE

Si la clave primaria a la que hace referencia la clave foranea es borrada o actualizada, en este caso este cambio se propagará a las tablas que tengan la clave foranea, si se definio `on delete cascade` se barranán las filas y en caso de `on update cascade` se actualizarán.

- SET NULL

Si la clave primaria a la que hace referencia la clave foranea es borrada o actualizada, en este caso este cambio se propagará a las tablas que tengan la clave foranea cambiando su valor a NULL.

- SET DEFAULT

Idem seteando el valor al valor default.

Integridad Referencial – FK (IV)

Sintaxis agregado como tconstraint en create table

```
FOREIGN KEY (col [, col …]) REFERENCES other_table
  (col [, col …])
[ON DELETE {NO ACTION|CASCADE|SET DEFAULT|SET NULL}]
[ON UPDATE {NO ACTION|CASCADE|SET DEFAULT|SET NULL}];
```

Sintaxis agregado como tconstraint en alter table

```
ALTER TABLE tabla ADD FOREIGN KEY (col1 [,col2] )
  REFERENCES tabla_referenciada(col1 [,col2])
  ON UPDATE CASCADE
  ON DELETE CASCADE;
```

Ejercicio Propuesto

Un estudio de arquitectura desea crear una BD para gestionar sus proyectos.

Nos dan las siguientes especificaciones:

- Cada proyecto tiene un código y un nombre. No hay dos proyectos que tengan el mismo nombre. Por otro lado se pretende que el código sea un autoincremental. Un proyecto tiene uno y solo un jefe de proyecto y un jefe de proyecto sólo puede estar involucrado en un proyecto, en mas de un proyecto o en ninguno.

- De cada jefe de proyecto se desean recoger sus datos personales (dni, cuil, nombre, dirección y teléfono). No hay dos jefes de proyecto con el mismo nombre.

Ejercicio Propuesto

- Cada proyecto tiene asignado un conjunto de recursos humanos, de cada recurso humano se desea registrar (dni, cuil, nombre, dirección, teléfono y las horas afectadas al proyecto). No hay dos recursos humanos con el mismo nombre. Un recurso humano puede ser parte de mas de un proyecto.

- Cada proyecto tiene planificadas 3 etapas o mas, de cada etapa se desea registrar un código único, un nombre y un responsable. El responsable puede estar o no asignado.

Genere el modelo necesario, imponiendo las restricciones al dominio que crea conveniente, justifique cada decisión.

Fuentes

Libro: Fundamentos de Bases de Datos
Autor: Silberschatz / Korth / Sudarshan
Editorial: Mc Graw Hill

http://www.postgresql.org/docs

Libro: Introducción a los SISTEMAS DE BASES DE DATOS
Autor: C.J. Date
Editorial: Addison Wesley