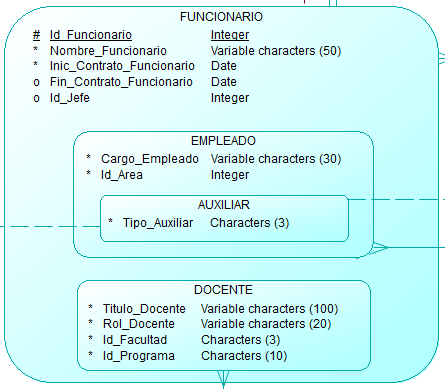
**RESOLUCIÓN DE SUPERTIPOS EN SQL**



Se diseña una tabla FUNCIONARIO que recoja todos los atributos de la imagen anterior junto con un atributo *tipo\_funcionario*. Los atributos obligatorios de los subtipos pasarán a ser opcionales.

FUNCIONARIO

# id\_funcionario Integer

+ nombre\_funcionario Varchar2(50)

+ inic\_contrato\_funcionario Date

o fin\_contrato\_funcionario Date

o id\_jefe Integer

o tipo\_funcionario Char(1)

o cargo\_empleado Varchar2(30)

o id\_area Integer

o titulo\_docente Varchar2(100)

o rol\_docente Varchar2(20)

o id\_facultad Char(3)

o id\_programa Char(10)

o tipo\_auxiliar Char(3)

La integridad por valor para el atributo *tipo\_funcionario* será ‘E’ para empleado y ‘D’ para docente.

Los atributos en color amarillo representan llaves foráneas.

El otro supertipo a resolver es el siguiente:



Se diseña una tabla LOCACION que recoja todos los atributos anteriores y uno adicional llamado tipo\_locacion, Los atributos obligatorios de los subtipos pasarán a ser opcionales:

LOCACION

# id\_locacion Integer

# id\_edificio Integer

+ nombre\_locacion Varchar2(100)

o tipo\_locacion Char(1)

o capacidad\_salon Integer

o tipo\_salon Char(3)

o puestos\_oficina Integer

o id\_area Integer

o id\_auxiliar Integer

o acreditacion\_laboratorio Varchar2(20)

La integridad por valor para el atributo tipo\_locacionserá ‘S’ para salón, ‘O’ para oficina y ‘L’ para laboratorio.

Los atributos en color amarillo representan llaves foráneas.

A continuación se presenta el SQL de creación completo. Se marcarán en color rojo las líneas a eliminar y en color verde las líneas a agregar o modificar.

/\*==============================================================\*/

/\* DBMS name: ORACLE Version 11g \*/

/\* Created on: 25/06/2020 12:34:49 p. m. \*/

/\*==============================================================\*/

alter table ACTIVO

drop constraint FK\_COMPRA\_IN\_ACTIVO;

alter table ACTIVO

drop constraint FK\_LOCACION\_IN\_ACTIVO;

alter table ACTIVO

drop constraint FK\_LAB\_IN\_ACTIVO;

alter table ACTIVO

drop constraint FK\_SALON\_IN\_ACTIVO;

alter table ACTIVO

drop constraint FK\_ACTIVO\_USADO\_POR\_FUN;

alter table ACTIVO

drop constraint FK\_DOCENTE\_IN\_ACTIVO;

alter table AREA

drop constraint FK\_DEPARTAMENTO\_IN\_AREA;

alter table AREA

drop constraint FK\_FACULTAD\_IN\_AREA;

alter table FUNCIONARIO

drop constraint FK\_AREA\_IN\_FUNCIONARIO;

alter table FUNCIONARIO

drop constraint FK\_JEFE\_IN\_FUNCIONARIO;

alter table DEPARTAMENTO

drop constraint FK\_FACULTAD\_IN\_DEPT;

alter table DEPARTAMENTO

drop constraint FK\_VICER\_IN\_DEPT;

alter table FUNCIONARIO

drop constraint FK\_PROGRAMA\_IN\_FUNCIONARIO;

alter table DOCENTE

drop constraint FK\_JEFE\_IN\_DOCENTE;

alter table EDIFICIO

drop constraint FK\_SEDE\_IN\_EDIFICIO;

alter table LOCACION

drop constraint FK\_AUXILIAR\_IN\_LAB;

alter table LABORATORIO

drop constraint EK\_EDIFICIO\_IN\_LAB;

alter table LOCACION

drop constraint FK\_AREA\_IN\_OFICINA;

alter table LOCACION

drop constraint FK\_EDIFICIO\_IN\_LOCACION;

alter table PROCEDIMIENTO

drop constraint FK\_FUN\_ENTREGA\_IN\_PROC;

alter table PROCEDIMIENTO

drop constraint FK\_DOCENTE\_ENTREGA\_IN\_PROC;

alter table PROCEDIMIENTO

drop constraint FK\_FUNCIONARIO\_IN\_PROC;

alter table PROCEDIMIENTO

drop constraint FK\_ACTIVO\_IN\_PROC;

alter table PROGRAMA

drop constraint FK\_FACULTAD\_IN\_PROGRAMA;

alter table SALON

drop constraint FK\_EDIFICIO\_IN\_SALON;

drop index SE\_ENCUENTRA\_EN2\_FK;

drop index USADO\_POR2\_FK;

drop index ACTIVO\_COMPRA\_FK;

drop table ACTIVO cascade constraints;

drop index AREA\_DEPARTAMENTO\_FK;

drop index AREA\_FACULTAD\_FK;

drop table AREA cascade constraints;

drop index TRABAJA\_EN\_FK;

drop table FUNCIONARIO cascade constraints;

drop table COMPRA cascade constraints;

drop index DEPARTAMENTO\_FACULTAD\_FK;

drop index DEPATAMENTO\_VICERRECTORIA\_FK;

drop table DEPARTAMENTO cascade constraints;

drop index TRABAJA\_FK;

drop table DOCENTE cascade constraints;

drop index EDIFICIO\_SEDE\_FK;

drop table EDIFICIO cascade constraints;

drop table FACULTAD cascade constraints;

drop index LOCACION\_EDIFICIO2\_FK;

drop index AUXILIAR\_LABORATORIO\_FK;

drop table LABORATORIO cascade constraints;

drop index LOCACION\_EDIFICIO\_FK;

drop index AREA\_OFICINA\_FK;

drop table LOCACION cascade constraints;

drop index ENTREGA2\_FK;

drop index PROCEDIMIENTO\_ACTIVO\_FK;

drop index PROCEDIMIENTO2\_FK;

drop table PROCEDIMIENTO cascade constraints;

drop index PROGRAMA\_FACULTAD\_FK;

drop table PROGRAMA cascade constraints;

drop index LOCACION\_EDIFICIO3\_FK;

drop table SALON cascade constraints;

drop table SEDE cascade constraints;

drop table VICERRECTORIA cascade constraints;

/\*==============================================================\*/

/\* Table: ACTIVO \*/

/\*==============================================================\*/

create table ACTIVO

(

ID\_ACTIVO INTEGER not null,

ID\_COMPRA INTEGER not null,

REFERENCIA\_ACTIVO VARCHAR2(40) not null,

CATEGORIA\_ACTIVO VARCHAR2(3) not null

constraint CKC\_CATEGORIA\_ACTIVO\_ACTIVO check (CATEGORIA\_ACTIVO in ('CAE','DTS','LAB','LOG','SGT')),

NOMBRE\_ACTIVO VARCHAR2(40) not null,

MARCA\_ACTIVO VARCHAR2(25) not null,

VALOR\_ACTIVO NUMBER(16,2) not null,

DEPRECIACION\_ACTIVO NUMBER(5,2) not null,

ID\_FUNCIONARIO INTEGER,

ID\_EDIFICIO INTEGER,

ID\_LOCACION INTEGER,

MATERIAL\_ACTIVO VARCHAR2(20),

COLOR\_ACTIVO VARCHAR2(20),

TAMANO\_ACTIVO VARCHAR2(20),

COMPONENTES\_ACTIVO VARCHAR2(120),

constraint PK\_ACTIVO primary key (ID\_ACTIVO)

);

/\*==============================================================\*/

/\* Index: ACTIVO\_COMPRA\_FK \*/

/\*==============================================================\*/

create index ACTIVO\_COMPRA\_FK on ACTIVO (

ID\_COMPRA ASC

);

/\*==============================================================\*/

/\* Index: USADO\_POR2\_FK \*/

/\*==============================================================\*/

create index USADO\_POR2\_FK on ACTIVO (

ID\_FUNCIONARIO ASC

);

/\*==============================================================\*/

/\* Index: SE\_ENCUENTRA\_EN2\_FK \*/

/\*==============================================================\*/

create index SE\_ENCUENTRA\_EN2\_FK on ACTIVO (

ID\_LOCACION ASC,

ID\_EDIFICIO ASC

);

/\*==============================================================\*/

/\* Table: AREA \*/

/\*==============================================================\*/

create table AREA

(

ID\_AREA INTEGER not null,

NOMBRE\_AREA VARCHAR2(70) not null,

ID\_FACULTAD CHAR(3),

ID\_DEPARTAMENTO INTEGER,

constraint PK\_AREA primary key (ID\_AREA)

);

/\*==============================================================\*/

/\* Index: AREA\_FACULTAD\_FK \*/

/\*==============================================================\*/

create index AREA\_FACULTAD\_FK on AREA (

ID\_FACULTAD ASC

);

/\*==============================================================\*/

/\* Index: AREA\_DEPARTAMENTO\_FK \*/

/\*==============================================================\*/

create index AREA\_DEPARTAMENTO\_FK on AREA (

ID\_DEPARTAMENTO ASC

);

/\*==============================================================\*/

/\* Table: FUNCIONARIO \*/

/\*==============================================================\*/

create table FUNCIONARIO

(

ID\_FUNCIONARIO INTEGER not null,

NOMBRE\_FUNCIONARIO VARCHAR2(50) not null,

INIC\_CONTRATO\_FUNCIONARIO DATE not null,

FIN\_CONTRATO\_FUNCIONARIO DATE ,

ID\_JEFE INTEGER ,

TIPO\_FUNCIONARIO CHAR(1) not null

constraint CKC\_TIPO\_FUN check (TIPO\_FUNCIONARIO in ('E', 'D')),

CARGO\_EMPLEADO VARCHAR2(30) ,

ID\_AREA INTEGER ,

TITULO\_DOCENTE VARCHAR2(100) ,

ROL\_DOCENTE VARCHAR2(20) ,

ID\_FACULTAD CHAR(3) ,

ID\_PROGRAMA CHAR(10) ,

TIPO\_AUXILIAR CHAR(3)

constraint CKC\_TIPO\_AUXILIAR\_FUN check (TIPO\_AUXILIAR in ('INV','MAN','LAB')),

constraint PK\_FUNCIONARIO primary key (ID\_FUNCIONARIO)

);

/\*==============================================================\*/

/\* Index: TRABAJA\_EN\_FK \*/

/\*==============================================================\*/

create index TRABAJA\_EN\_FK on FUNCIONARIO (

ID\_AREA ASC

);

/\*==============================================================\*/

/\* Table: COMPRA \*/

/\*==============================================================\*/

create table COMPRA

(

ID\_COMPRA INTEGER not null,

FECHA\_COMPRA DATE not null,

VALOR\_COMPRA NUMBER(16,2) not null,

constraint PK\_COMPRA primary key (ID\_COMPRA)

);

/\*==============================================================\*/

/\* Table: DEPARTAMENTO \*/

/\*==============================================================\*/

create table DEPARTAMENTO

(

ID\_DEPARTAMENTO INTEGER not null,

NOMBRE\_DEPARTAMENTO VARCHAR2(70) not null,

DIRECTOR\_DEPARTAMENTO VARCHAR2(50) not null,

ID\_VICERRECTORIA CHAR(3),

ID\_FACULTAD CHAR(3),

constraint PK\_DEPARTAMENTO primary key (ID\_DEPARTAMENTO)

);

/\*==============================================================\*/

/\* Index: DEPATAMENTO\_VICERRECTORIA\_FK \*/

/\*==============================================================\*/

create index DEPATAMENTO\_VICERRECTORIA\_FK on DEPARTAMENTO (

ID\_VICERRECTORIA ASC

);

/\*==============================================================\*/

/\* Index: DEPARTAMENTO\_FACULTAD\_FK \*/

/\*==============================================================\*/

create index DEPARTAMENTO\_FACULTAD\_FK on DEPARTAMENTO (

ID\_FACULTAD ASC

);

/\*==============================================================\*/

/\* Table: DOCENTE \*/

/\*==============================================================\*/

create table DOCENTE

(

ID\_FUNCIONARIO INTEGER not null,

NOMBRE\_FUNCIONARIO VARCHAR2(50),

INIC\_CONTRATO\_FUNCIONARIO DATE,

FIN\_CONTRATO\_FUNCIONARIO DATE,

ID\_JEFE INTEGER,

TITULO\_DOCENTE VARCHAR2(100) not null,

ROL\_DOCENTE VARCHAR2(20) not null,

ID\_FACULTAD CHAR(3) not null,

ID\_PROGRAMA CHAR(10) not null,

constraint AK\_IDENTIFIER\_1\_DOCENTE unique (ID\_FUNCIONARIO)

);

/\*==============================================================\*/

/\* Index: TRABAJA\_FK \*/

/\*==============================================================\*/

create index TRABAJA\_FK on DOCENTE (

ID\_FACULTAD ASC,

ID\_PROGRAMA ASC

);

/\*==============================================================\*/

/\* Table: EDIFICIO \*/

/\*==============================================================\*/

create table EDIFICIO

(

ID\_SEDE INTEGER not null,

ID\_EDIFICIO INTEGER not null,

NOMBRE\_EDIFICIO VARCHAR2(50) not null,

constraint PK\_EDIFICIO primary key (ID\_EDIFICIO)

);

/\*==============================================================\*/

/\* Index: EDIFICIO\_SEDE\_FK \*/

/\*==============================================================\*/

create index EDIFICIO\_SEDE\_FK on EDIFICIO (

ID\_SEDE ASC

);

/\*==============================================================\*/

/\* Table: FACULTAD \*/

/\*==============================================================\*/

create table FACULTAD

(

ID\_FACULTAD CHAR(3) not null

constraint CKC\_ID\_FACULTAD\_FACULTAD check (ID\_FACULTAD in ('AGR','JUR','ECO','ING','CIE')),

NOMBRE\_FACULTAD VARCHAR2(60) not null,

DECANO\_FACULTAD VARCHAR2(50) not null,

constraint PK\_FACULTAD primary key (ID\_FACULTAD)

);

/\*==============================================================\*/

/\* Table: LABORATORIO \*/

/\*==============================================================\*/

create table LABORATORIO

(

ID\_EDIFICIO INTEGER not null,

ID\_LOCACION INTEGER not null,

NOMBRE\_LOCACION VARCHAR2(100),

ID\_AUXILIAR INTEGER not null,

ACREDITACION\_LABORATORIO VARCHAR2(20),

constraint PK\_LABORATORIO primary key (ID\_LOCACION, ID\_EDIFICIO),

constraint AK\_IDENTIFIER\_1\_LABORATO unique (ID\_LOCACION)

);

/\*==============================================================\*/

/\* Index: LOCACION\_EDIFICIO2\_FK \*/

/\*==============================================================\*/

create index LOCACION\_EDIFICIO2\_FK on LABORATORIO (

ID\_EDIFICIO ASC

);

/\*==============================================================\*/

/\* Table: LOCACION \*/

/\*==============================================================\*/

create table LOCACION

(

ID\_EDIFICIO INTEGER not null,

ID\_LOCACION INTEGER not null,

NOMBRE\_LOCACION VARCHAR2(100) not null,

TIPO\_LOCACION CHAR(1) not null

constraint CKC\_TIPO\_LOCACION check (TIPO\_LOCACION in ('S','O','L')),

CAPACIDAD\_SALON INTEGER ,

TIPO\_SALON CHAR(3)

constraint CKC\_TIPO\_SALON check (TIPO\_SALON is null or (TIPO\_SALON in ('AUD','TAL','COM'))),

PUESTOS\_OFICINA INTEGER ,

ID\_AREA INTEGER ,

ID\_AUXILIAR INTEGER ,

ACREDITACION\_LAB VARCHAR2(20) ,

constraint PK\_LOCACION primary key (ID\_EDIFICIO, ID\_LOCACION)\_

constraint AK\_IDENTIFIER\_1\_OFICINA unique (ID\_LOCACION)

);

/\*==============================================================\*/

/\* Index: AUXILIAR\_LABORATORIO\_FK \*/

/\*==============================================================\*/

create index AUXILIAR\_LABORATORIO\_FK on LOCACION (

ID\_AUXILIAR ASC

);

/\*==============================================================\*/

/\* Index: AREA\_OFICINA\_FK \*/

/\*==============================================================\*/

create index AREA\_OFICINA\_FK on LOCACION (

ID\_AREA ASC

);

/\*==============================================================\*/

/\* Index: LOCACION\_EDIFICIO\_FK \*/

/\*==============================================================\*/

create index LOCACION\_EDIFICIO\_FK on LOCACION (

ID\_EDIFICIO ASC

);

/\*==============================================================\*/

/\* Table: PROCEDIMIENTO \*/

/\*==============================================================\*/

create table PROCEDIMIENTO

(

ID\_PROCE INTEGER not null,

ID\_ACTIVO INTEGER not null,

ID\_AUXILIAR INTEGER not null,

TIPO\_PROCEDIMIENTO CHAR(1) not null

constraint CKC\_TIPO\_PROCEDIMIENT\_PROCEDIM check (TIPO\_PROCEDIMIENTO in ('E','P','R','M')),

FECHA\_PROCEDIMIENTO DATE not null,

ESTADO\_REVISION CHAR(1)

constraint CKC\_ESTADO\_REVISION\_PROCEDIM check (ESTADO\_REVISION is null or (ESTADO\_REVISION in ('R','M'))),

OBSERVACIONES VARCHAR2(150),

FUNCIONARIO\_ENTREGA INTEGER,

constraint PK\_PROCEDIMIENTO primary key (ID\_PROCE)

);

/\*==============================================================\*/

/\* Index: PROCEDIMIENTO2\_FK \*/

/\*==============================================================\*/

create index PROCEDIMIENTO2\_FK on PROCEDIMIENTO (

ID\_AUXILIAR ASC

);

/\*==============================================================\*/

/\* Index: PROCEDIMIENTO\_ACTIVO\_FK \*/

/\*==============================================================\*/

create index PROCEDIMIENTO\_ACTIVO\_FK on PROCEDIMIENTO (

ID\_ACTIVO ASC

);

/\*==============================================================\*/

/\* Index: ENTREGA2\_FK \*/

/\*==============================================================\*/

create index ENTREGA2\_FK on PROCEDIMIENTO (

FUNCIONARIO\_ENTREGA ASC

);

/\*==============================================================\*/

/\* Table: PROGRAMA \*/

/\*==============================================================\*/

create table PROGRAMA

(

ID\_FACULTAD CHAR(3) not null,

ID\_PROGRAMA CHAR(10) not null,

NOMBRE\_PROGRAMA VARCHAR2(80) not null,

constraint PK\_PROGRAMA primary key (ID\_FACULTAD, ID\_PROGRAMA)

);

/\*==============================================================\*/

/\* Index: PROGRAMA\_FACULTAD\_FK \*/

/\*==============================================================\*/

create index PROGRAMA\_FACULTAD\_FK on PROGRAMA (

ID\_FACULTAD ASC

);

/\*==============================================================\*/

/\* Table: SALON \*/

/\*==============================================================\*/

create table SALON

(

ID\_EDIFICIO INTEGER not null,

ID\_LOCACION INTEGER not null,

NOMBRE\_LOCACION VARCHAR2(100),

CAPACIDAD\_SALON INTEGER not null,

TIPO\_SALON CHAR(3)

constraint CKC\_TIPO\_SALON\_SALON check (TIPO\_SALON is null or (TIPO\_SALON in ('AUD','TAL','COM'))),

constraint PK\_SALON primary key (ID\_LOCACION, ID\_EDIFICIO),

constraint AK\_IDENTIFIER\_1\_SALON unique (ID\_LOCACION)

);

/\*==============================================================\*/

/\* Index: LOCACION\_EDIFICIO3\_FK \*/

/\*==============================================================\*/

create index LOCACION\_EDIFICIO3\_FK on SALON (

ID\_EDIFICIO ASC

);

/\*==============================================================\*/

/\* Table: SEDE \*/

/\*==============================================================\*/

create table SEDE

(

ID\_SEDE INTEGER not null,

NOMBRE\_SEDE VARCHAR2(70) not null,

constraint PK\_SEDE primary key (ID\_SEDE)

);

/\*==============================================================\*/

/\* Table: VICERRECTORIA \*/

/\*==============================================================\*/

create table VICERRECTORIA

(

ID\_VICERRECTORIA CHAR(3) not null

constraint CKC\_ID\_VICERRECTORIA\_VICERREC check (ID\_VICERRECTORIA in ('FOR','INV','EXT','ADM')),

NOMBRE\_VICERRECTORIA VARCHAR2(50) not null,

VICERRECTOR\_VICERRECTORIA VARCHAR2(50) not null,

constraint PK\_VICERRECTORIA primary key (ID\_VICERRECTORIA)

);

alter table ACTIVO

add constraint FK\_COMPRA\_IN\_ACTIVO foreign key (ID\_COMPRA)

references COMPRA (ID\_COMPRA);

alter table ACTIVO

add constraint FK\_LOCACION\_IN\_ACTIVO foreign key (ID\_EDIFICIO, ID\_LOCACION)

references LOCACION (ID\_EDIFICIO, ID\_LOCACION);

alter table ACTIVO

add constraint FK\_LAB\_IN\_ACTIVO foreign key (ID\_LOCACION, ID\_EDIFICIO)

references LABORATORIO (ID\_LOCACION, ID\_EDIFICIO);

alter table ACTIVO

add constraint FK\_SALON\_IN\_ACTIVO foreign key (ID\_LOCACION, ID\_EDIFICIO)

references SALON (ID\_LOCACION, ID\_EDIFICIO);

alter table ACTIVO

add constraint FK\_ACTIVO\_USADO\_POR\_FUN foreign key (ID\_FUNCIONARIO)

references FUNCIONARIO (ID\_FUNCIONARIO);

alter table ACTIVO

add constraint FK\_DOCENTE\_IN\_ACTIVO foreign key (ID\_FUNCIONARIO)

references DOCENTE (ID\_FUNCIONARIO);

alter table AREA

add constraint FK\_DEPARTAMENTO\_IN\_AREA foreign key (ID\_DEPARTAMENTO)

references DEPARTAMENTO (ID\_DEPARTAMENTO);

alter table AREA

add constraint FK\_FACULTAD\_IN\_AREA foreign key (ID\_FACULTAD)

references FACULTAD (ID\_FACULTAD);

alter table FUNCIONARIO

add constraint FK\_AREA\_IN\_FUNCIONARIO foreign key (ID\_AREA)

references AREA (ID\_AREA);

alter table FUNCIONARIO

add constraint FK\_JEFE\_IN\_FUNCIONARIO foreign key (ID\_JEFE)

references FUNCIONARIO (ID\_FUNCIONARIO);

alter table DEPARTAMENTO

add constraint FK\_FACULTAD\_IN\_DEPT foreign key (ID\_FACULTAD)

references FACULTAD (ID\_FACULTAD);

alter table DEPARTAMENTO

add constraint FK\_VICER\_IN\_DEPT foreign key (ID\_VICERRECTORIA)

references VICERRECTORIA (ID\_VICERRECTORIA);

alter table FUNCIONARIO

add constraint FK\_PROGRAMA\_IN\_FUNCIONARIO foreign key (ID\_FACULTAD, ID\_PROGRAMA)

references PROGRAMA (ID\_FACULTAD, ID\_PROGRAMA);

alter table DOCENTE

add constraint FK\_JEFE\_IN\_DOCENTE foreign key (ID\_JEFE)

references DOCENTE (ID\_FUNCIONARIO);

alter table EDIFICIO

add constraint FK\_SEDE\_IN\_EDIFICIO foreign key (ID\_SEDE)

references SEDE (ID\_SEDE);

alter table LOCACION

add constraint FK\_AUXILIAR\_IN\_LAB foreign key (ID\_AUXILIAR)

references FUNCIONARIO (ID\_FUNCIONARIO);

alter table LABORATORIO

add constraint EK\_EDIFICIO\_IN\_LAB foreign key (ID\_EDIFICIO)

references EDIFICIO (ID\_EDIFICIO);

alter table LOCACION

add constraint FK\_AREA\_IN\_OFICINA foreign key (ID\_AREA)

references AREA (ID\_AREA);

alter table LOCACION

add constraint FK\_EDIFICIO\_IN\_LOCACION foreign key (ID\_EDIFICIO)

references EDIFICIO (ID\_EDIFICIO);

alter table PROCEDIMIENTO

add constraint FK\_FUN\_ENTREGA\_IN\_PROC foreign key (FUNCIONARIO\_ENTREGA)

references FUNCIONARIO (ID\_FUNCIONARIO);

alter table PROCEDIMIENTO

add constraint FK\_DOCENTE\_ENTREGA\_IN\_PROC foreign key (FUNCIONARIO\_ENTREGA)

references DOCENTE (ID\_FUNCIONARIO);

alter table PROCEDIMIENTO

add constraint FK\_FUNCIONARIO\_IN\_PROC foreign key (ID\_AUXILIAR)

references FUNCIONARIO (ID\_FUNCIONARIO);

alter table PROCEDIMIENTO

add constraint FK\_ACTIVO\_IN\_PROC foreign key (ID\_ACTIVO)

references ACTIVO (ID\_ACTIVO);

alter table PROGRAMA

add constraint FK\_FACULTAD\_IN\_PROGRAMA foreign key (ID\_FACULTAD)

references FACULTAD (ID\_FACULTAD);

alter table SALON

add constraint FK\_EDIFICIO\_IN\_SALON foreign key (ID\_EDIFICIO)

references EDIFICIO (ID\_EDIFICIO);

En color fucsia están los índices para la relación intermedia PROCEDIMIENTO. No estoy seguro si se pueden eliminar sin inconveniente. Me parece haber entendido eso en clase.

Finalmente, se agregarán los CHECK de tablas para cada supertipo generado:

/\*==============================================================\*/

/\* CHECK TABLE FUNCIONARIO \*/

/\*==============================================================\*/

alter table FUNCIONARIO

add constraint FUNCIONARIO\_CHK\_TABLE

CHECK (

(TIPO\_FUNCIONARIO = 'E'

AND CARGO\_EMPLEADO IS NOT NULL

AND ID\_AREA IS NOT NULL

AND TITULO\_DOCENTE IS NULL

AND ROL\_DOCENTE IS NULL

AND ID\_PROGRAMA IS NULL

AND ID\_FACULTAD IS NULL) OR

(TIPO\_FUNCIONARIO = 'D'

AND CARGO\_EMPLEADO IS NULL

AND ID\_AREA IS NULL

AND TIPO\_AUXILIAR IS NULL

AND TITULO\_DOCENTE IS NOT NULL

AND ROL\_DOCENTE IS NOT NULL

AND ID\_PROGRAMA IS NOT NULL

AND ID\_FACULTAD IS NOT NULL)

);

/\*==============================================================\*/

/\* CHECK TABLE PROCEDIMIENTO \*/

/\*==============================================================\*/

alter table PROCEDIMIENTO

add constraint PROCEDIMIENTO\_CHK\_TABLE

CHECK (

(TIPO\_PROCEDIMIENTO = 'R'

AND FUNCIONARIO\_ENTREGA IS NULL) OR

(TIPO\_PROCEDIMIENTO = 'P'

AND FUNCIONARIO\_ENTREGA IS NULL) OR

(TIPO\_PROCEDIMIENTO = 'M'

AND ESTADO\_REVISION IS NULL

AND FUNCIONARIO\_ENTREGA IS NULL) OR

(TIPO\_PROCEDIMIENTO = 'E'

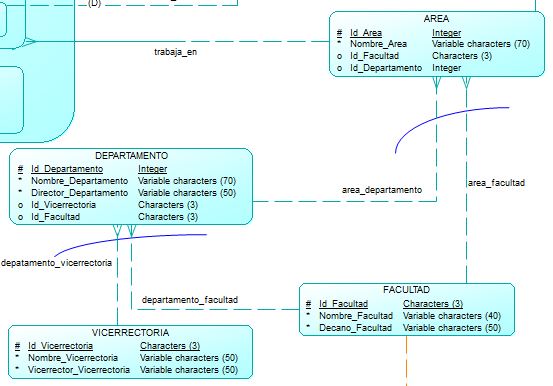
AND ESTADO\_REVISION IS NULL

AND FUNCIONARIO\_ENTREGA IS NOT NULL)

);

**RESOLUCIÓN DE ARCOS CON CHECK CONSTRAINTS**

Los arcos a solucionar son los siguientes:



Se usará el siguiente código SQL para garantizar la integridad referencial entre estas relaciones.

alter table DEPARTAMENTO

add constraint DEPARTAMENTO\_ARC\_CHK

check (

(id\_vicerrectoria IS NOT NULL

and id\_facultad IS NULL) OR

(id\_vicerrectoria IS NULL

and id\_facultad IS NOT NULL)

);

alter table AREA

add constraint AREA\_ARC\_CHK

check (

(id\_facultad IS NOT NULL

and id\_departamento IS NULL) OR

(id\_facultad IS NULL

and id\_departamento IS NOT NULL)

);

**SIGUIENTES PASOS**

Verificar como se garantiza la relación 1-1. Hay que hacer algo relacionado con el comando UNIQUE y los índices pero no recuerdo qué es o cómo se hace.

Verificar si es seguro eliminar los índices de la relación intermedia y que están en color fucsia en el código SQL de este documento.

Verificar si es posible implementar disparadores o *triggers* para garantizar lo siguiente:

* Que solo los auxiliares cuyo tipo es INV puedan hacer revisiones, entregas y firmas de paz y salvos
* Que solo los auxiliares de tipo MAN puedan hacer mantenimientos
* Que solo los auxiliares de tipo LAB puedan ser relacionados en la tabla LABORATORIO como auxiliar encargado.