

ENUNCIADO 1

DIAGRAMA ENTIDAD RELACION

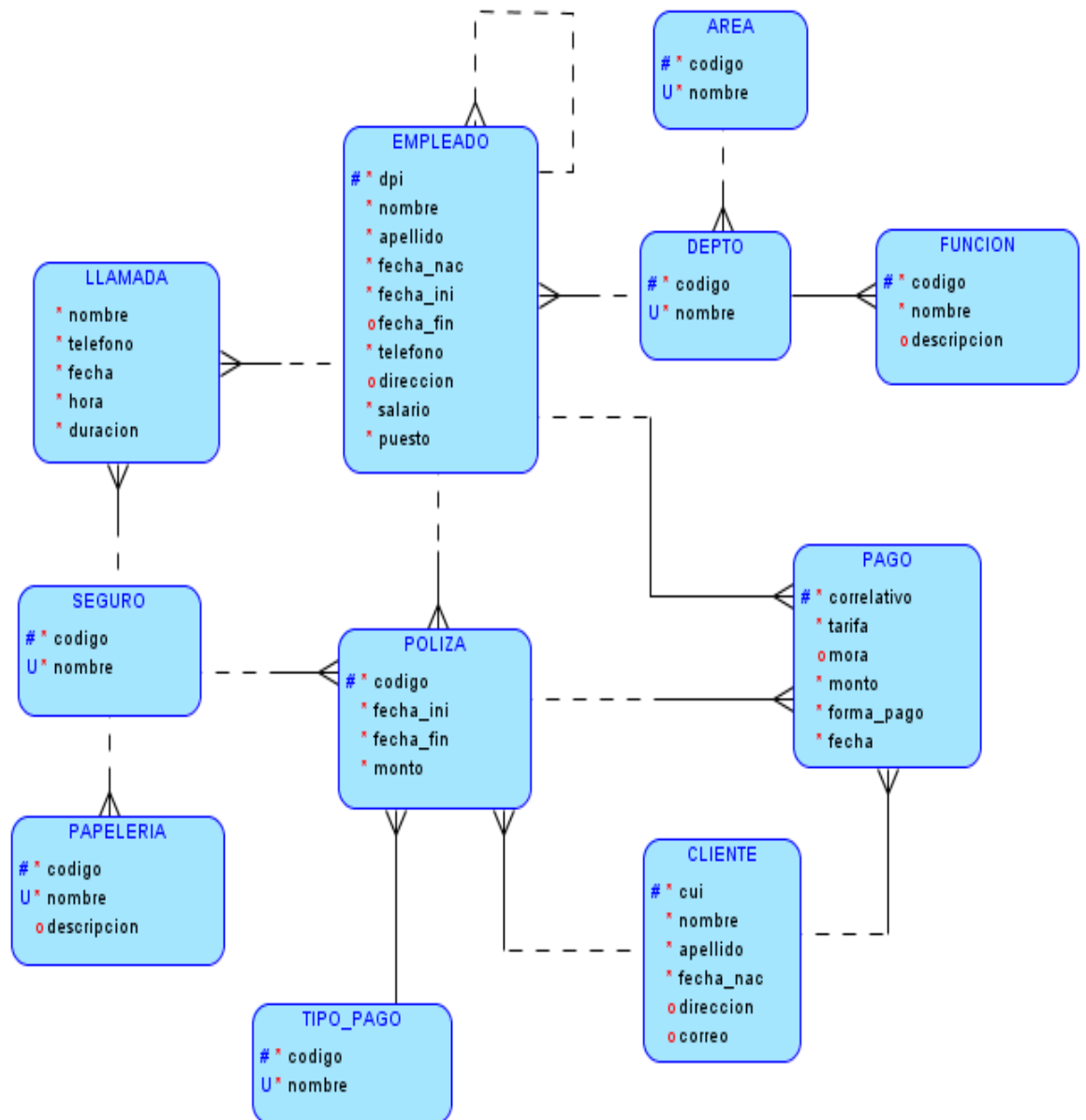
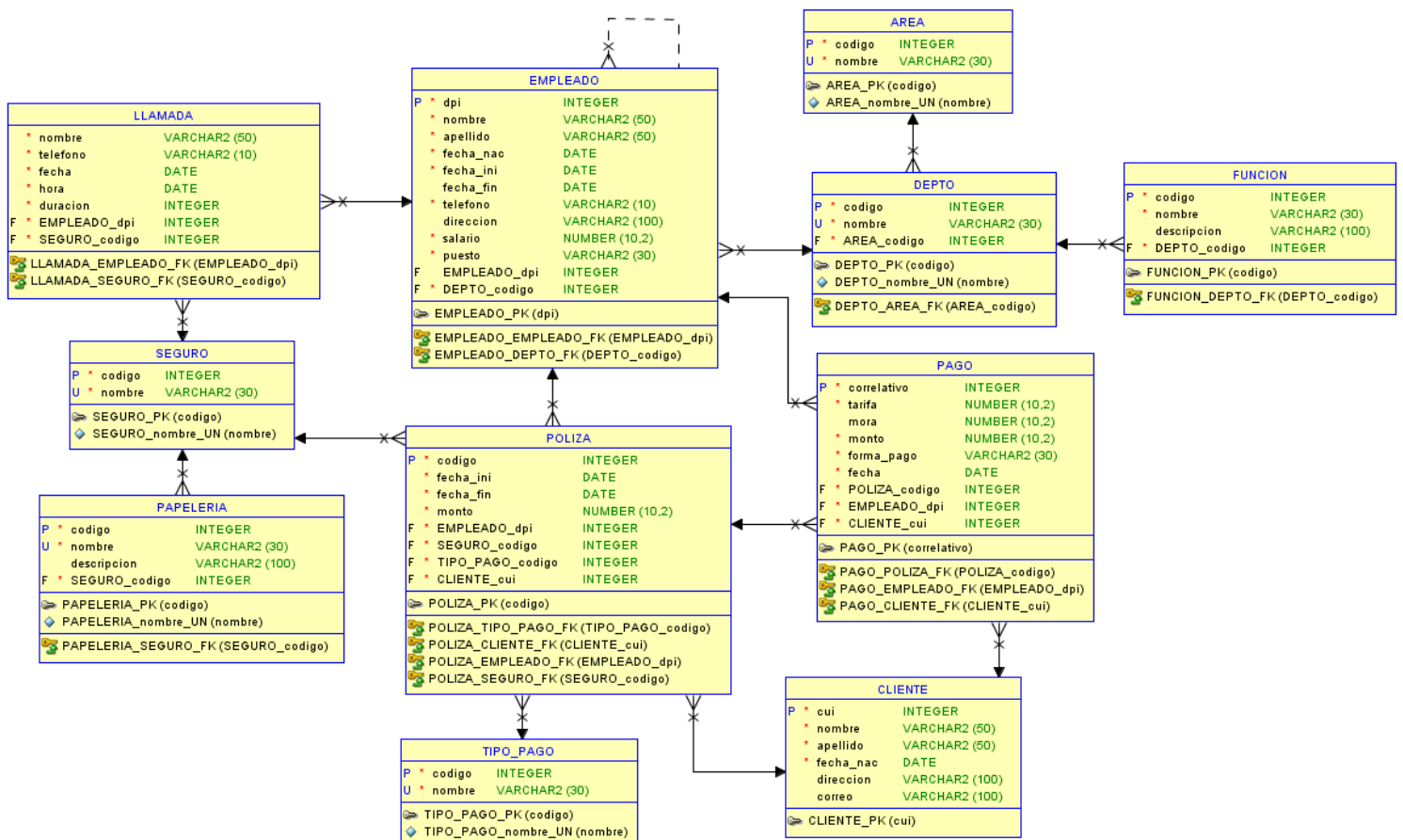


DIAGRAMA MODELO RELACIONAL



DESCRIPCION DE ENTIDADES

Nombre Entidad	Descripción
FUNCION	Funciones que deben de cumplir cada departamento.
AREA	La aseguradora se divide en áreas para una mejor organización de los empleados.
DEPTO	Las áreas se dividen en departamentos para que cada empleado tenga un área de trabajo.
EMPLEADO	Atiende a clientes y da información sobre los seguros.
LLAMADA	Se guarda los registros de clientes que llamaron para información de seguros.
PAGO	Pagos que realiza un cliente a una determinada póliza.
POLIZA	Guarda el contrato que firma el cliente al adquirir un seguro.
SEGURO	Registra los seguros que cubre la a seguridad, además
PAPELERIA	Esta entidad se almacena la información de papelería que se debe presentar para un seguro.
TIPO_PAGO	La entidad guarda el tipo de pago de un seguro si es semestralmente, semanalmente, diariamente, etc.
CLIENTE	Registra al cliente que contrata una póliza.

DESCRIPCION DE ATRIBUTOS POR ENTIDAD

AREA

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	SI	30 caracteres máximo.

FUNCION

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	SI	30 caracteres máximo.
Descripción	Varchar		NO	NO	100 caracteres máximo, especifica la función de un departamento.
DEPTO_codigo	Integer	Foránea	SI	NO	Código del departamento.

DEPTO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	SI	30 caracteres máximo.
AREA_codigo	Integer	Foránea	SI	NO	Código de área.

EMPLEADO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Dpi	Integer	Primaria	SI	SI	Dpi único del empleado.
Nombre	Varchar		SI	NO	50 caracteres máximo se puede ingresar 2 o más nombres.
Apellido	Varchar		SI	NO	50 caracteres máximo se puede ingresar 2 o más apellidos.
Fecha_nac	Date		SI	NO	Fecha nacimiento para calcular la edad.
Fecha_ini	Date		SI	NO	Fecha de inicio del contrato de un empleado.
Fecha_fin	Date		NO	NO	Fecha que deja de trabajar un empleado es opcional por que no se sabe cuándo va a dejar de trabajar en la aseguradora.
Teléfono	Varchar		SI	NO	10 caracteres máximo.
Dirección	Varchar		NO	NO	100 caracteres máximo.
Salario	Number		SI	NO	Es un número de 2 decimales.
Puesto	Varchar		SI	NO	30 caracteres máximo.
EMPLEADO_dpi	Integer	Foránea	NO	NO	Código del jefe.

DEPTO_cod	Integer	Foránea	SI	NO	Código del departamento.
-----------	---------	---------	----	----	--------------------------

LLAMADA

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Teléfono	Varchar		SI	NO	10 caracteres máximo.
Nombre	Varchar		SI	SI	50 caracteres máximo. Almacena el nombre del cliente que llamo.
Fecha	Date		SI	NO	Fecha que se registró la llamada.
Hora	Date		SI	NO	Hora en la que se registró la llamada.
Duración	Integer		SI	NO	
EMPLEADO_dpi	Integer	Foránea	SI	NO	Código del empleado que atendió.
SEGURO_codigo	Integer	Foránea	SI	NO	Código del seguro que brindó información.

SEGURO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	SI	30 caracteres máximo.

PAPELERIA

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	SI	30 caracteres máximo.
Descripción	Varchar		NO	NO	100 caracteres máximo. Descripción de la papelería que se necesita.
SEGURO_codigo	Integer	Foránea	SI	NO	Código de seguro que necesita la papelería.

POLIZA

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Fecha_ini	Date		SI	NO	Fecha inicial del contrato del seguro.
Fecha_fin	Date		SI	NO	Fecha en que finaliza el contrato del seguro.
Monto	Number		SI	NO	Monto total a pagar.
EMPLEADO_dpi	Integer	Foránea	SI	NO	Código del empleado que realizo la póliza.

SEGURO_codigo	Integer	Foránea	SI	NO	Código del seguro de la póliza.
TIPO_PAGO_codigo	Integer	Foránea	SI	NO	Código del tipo de pago.
CLIENTE_cui	Integer	Foránea	SI	NO	Cliente que pidió la póliza.

TIPO DE PAGO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	SI	30 caracteres máximo.

PAGO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Correlativo	Integer	Primaria	SI	SI	Código de identificación.
Tarifa	Number		SI	NO	Tarifa que pagara el cliente.
Mora	Number		NO	NO	Mora en caso exista.
Monto	Number		SI	NO	Total a pagar.
Forma_pago	Varchar		SI	NO	Especifica si es tarjeta de crédito, débito o en efectivo.
Fecha	Date		SI	NO	Fecha en que se realizó el pago.
POLIZA_codigo	Integer	Foránea	SI	NO	Código de la póliza que se pagó.
EMPLEADO_cui	Integer	Foránea	SI	NO	Empleado que recibió el pago.
CLIENTE_cui	Integer	Foránea	SI	NO	Cliente que realizó el pago.

CLIENTE

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Cui	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	NO	50 caracteres máximo.
Apellido	Varchar		SI	NO	50 caracteres máximo.
Fecha_nac	Date		SI	NO	
Dirección	Varchar		NO	NO	100 caracteres máximo el cliente puede dar información de su dirección.
Correo	Varchar		SI	NO	100 caracteres máximo

DESCRIPCION DE RELACIONES ENTRE ENTIDADES

Entidad1	Entidad2	Relación N:N	Descripción
----------	----------	-----------------	-------------

AREA	DEPTO	1:M	En área puede haber uno o varios departamentos
DEPTO	FUNCION	1:M	Un departamento tiene varias funciones que realizan en la empresa que son únicas para cada departamento.
DEPTO	EMPLEADO	1:M	Un departamento tiene uno o varios empleados que trabajan.
EMPLEADO	EMPLEADO	1:M	Un empleado puede ser jefe de otros empleados.
EMPLEADO	LLAMADA	1:M	Un empleado puede recibir varias llamadas.
EMPLEADO	POLIZA	1:M	Un empleado puede realizar el trámite de una póliza.
EMPLEADO	PAGO	1:M	Un empleado recibe el pago de las pólizas.
SEGURO	LLAMADA	1:M	Un seguro puede ser informado en las llamadas.
SEGURO	PAPELERIA	1:M	Un seguro puede tener papelería que es un requisito para el contrato del seguro.
SEGURO	POLIZA	1:M	Un seguro está en una o varias pólizas que adquiera cualquier cliente.
POLIZA	PAGO	1:M	Para la póliza se puede hacer en varios pagos.
TIPO_PAGO	POLIZA	1:M	En tipo de pago puede estar en varias pólizas.
CLIENTE	POLIZA	1:M	Un cliente realiza el contrato con varias pólizas.
CLIENTE	PAGO	1:M	Un cliente puede realizar el pago de varias pólizas.

DESCRIPCION DE LAS RESTRICCIONES

Unique: La restricción unique identifica de manera única a cada fila de una tabla.

Primary Key: Identifica de manera única cada fila de una tabla.

Foreign key: Es una referencia a otra tabla. Puede aceptar valores nulos.

CONSIDERACIONES QUE SE TOMARON AL MOMENTO DE DISEÑAR EL DIAGRAMA

1. No se tomó el cómo campo la edad porque se puede calcular con la fecha de nacimiento además, cada vez que una persona cumpla años se debe de modificar en la base de datos.
2. Para la entidad empleado se hizo recursiva porque puede tener jefe.
3. La llamada puede ser anónima, que solo quiera saber la información de los seguros, por lo que no se relaciona con la entidad cliente.
4. En cuanto al horario de oficina que es de 8:00 a 17:00 horas se agrega con constraint a la hora de la entidad Llamada.
5. Realizar un trigger que cuente cuantas pólizas tiene a su nombre un cliente y si es menor a 5. Otra forma de hacerlo es crear un campo en cliente que es el número de pólizas que

- tiene y realizar un trigger que calcule el número de pólizas y colocar un constraint que sea menor a 5.
6. Cuando una persona quiera adquirir un seguro debe ser mayor a 18 años por lo que se puede realizar con un constraint calculando la edad.
 7. El código de la póliza iniciándola en 10000 y creando un auto incrementable en la base de datos.

SCRIPT PARA EL DBMS DE ORACLE

```
create table AREA (  
    codigo integer not null,  
    nombre varchar(30) constraint AREA_nombre_UN unique not null,  
    constraint AREA_PK primary key (codigo)  
);
```

```
create table CLIENTE (  
    cui integer not null,  
    nombre varchar(50) not null,  
    apellido varchar(50) not null,  
    fecha_nac date not null,  
    direccion varchar(100),  
    correo varchar(100),  
    constraint CLIENTE_PK primary key (cui)  
);
```

```
create table DEPTO (  
    codigo integer not null,  
    nombre varchar(30) constraint DEPTO_nombre_UN unique not null,  
    AREA_codigo integer not null,  
    constraint DEPTO_PK primary key (codigo),  
    constraint DEPTO_AREA_FK foreign key (AREA_codigo) references AREA (codigo)  
);
```

```
create table EMPLEADO (  
    dpi integer not null,
```

```

nombre    varchar(50) not null,
apellido  varchar(50) not null,
fecha_nac date not null,
fecha_ini date not null,
fecha_fin date,
telefono  varchar(10) not null,
direccion varchar(100),
salario   number(10,2) not null,
puesto    varchar(30) not null,
EMPLEADO_dpi integer,
DEPTO_codigo integer not null,
constraint EMPLEADO_PK primary key (dpi),
constraint EMPLEADO_EMPLEADO_FK foreign key (EMPLEADO_dpi) references
EMPLEADO (dpi),
constraint EMPLEADO_DEPTO_FK foreign key (DEPTO_codigo) references DEPTO (codigo)
);

```

```

create table FUNCION (
    codigo    integer not null ,
    nombre    varchar (30) not null ,
    descripcion varchar (100) ,
    DEPTO_codigo integer not null,
    constraint FUNCION_PK primary key (codigo),
    constraint FUNCION_DEPTO_FK foreign key (DEPTO_codigo) references DEPTO (codigo)
);

```

```

create table LLAMADA (
    nombre    varchar(50) not null,
    telefono  varchar(10) not null,
    fecha     date not null,
    hora      date not null,

```



```

    duracion    integer not null,

    EMPLEADO_dpi integer not null,

    SEGURO_codigo integer not null,

    constraint LLAMADA_EMPLEADO_FK foreign key (EMPLEADO_dpi) references EMPLEADO
(dpi),

    constraint LLAMADA_SEGURO_FK foreign key (SEGURO_codigo) references SEGURO
(codigo)

);

```

```

create table PAGO (

    correlativo integer not null,

    tarifa      NUMBER(10,2) not null,

    mora       NUMBER(10,2),

    monto       NUMBER(10,2) not null,

    forma_pago  varchar(30) not null,

    fecha       date not null,

    POLIZA_codigo integer not null,

    EMPLEADO_dpi integer not null,

    CLIENTE_cui integer not null,

    constraint PAGO_PK primary key (correlativo),

    constraint PAGO_CLIENTE_FK foreign key (CLIENTE_cui) references CLIENTE (cui),

    constraint PAGO_EMPLEADO_FK foreign key (EMPLEADO_dpi) references EMPLEADO
(dpi),

    constraint PAGO_POLIZA_FK foreign key (POLIZA_codigo) references POLIZA (codigo)

);

```

```

create table PAPELERIA (

    codigo      integer not null,

    nombre      varchar(30) constraint PAPELERIA_nombre_UN unique not null,

    descripcion varchar(100),

    SEGURO_codigo integer not null,

    constraint PAPELERIA_PK primary key (codigo),

```

```
    constraint PAPELERIA_SEGURO_FK foreign key (SEGURO_codigo) references SEGURO
(codigo)
);
```

```
create table POLIZA (
    codigo      integer not null,
    fecha_ini   date not null,
    fecha_fin   date not null,
    monto       NUMBER(10,2) not null,
    EMPLEADO_dpi integer not null,
    SEGURO_codigo integer not null,
    TIPO_PAGO_codigo integer not null,
    CLIENTE_cui integer not null,
    constraint POLIZA_PK primary key (codigo),
    constraint POLIZA_CLIENTE_FK foreign key (CLIENTE_cui) references CLIENTE (cui),
    constraint POLIZA_EMPLEADO_FK foreign key (EMPLEADO_dpi) references EMPLEADO
(dpi),
    constraint POLIZA_SEGURO_FK foreign key (SEGURO_codigo) references SEGURO
(codigo),
    constraint POLIZA_TIPO_PAGO_FK foreign key (TIPO_PAGO_codigo) references
TIPO_PAGO (codigo)
);
```

```
create table SEGURO (
    codigo integer not null,
    nombre varchar(30) constraint SEGURO_nombre_UN unique not null,
    constraint SEGURO_PK primary key (codigo)
);
```

```
create table TIPO_PAGO (
    codigo integer not null,
    nombre varchar(30) constraint TIPO_PAGO_nombre_UN unique not null,
```

constraint TIPO_PAGO_PK primary key (codigo)

);

ENUNCIADO 2

DIAGRAMA ENTIDAD RELACION

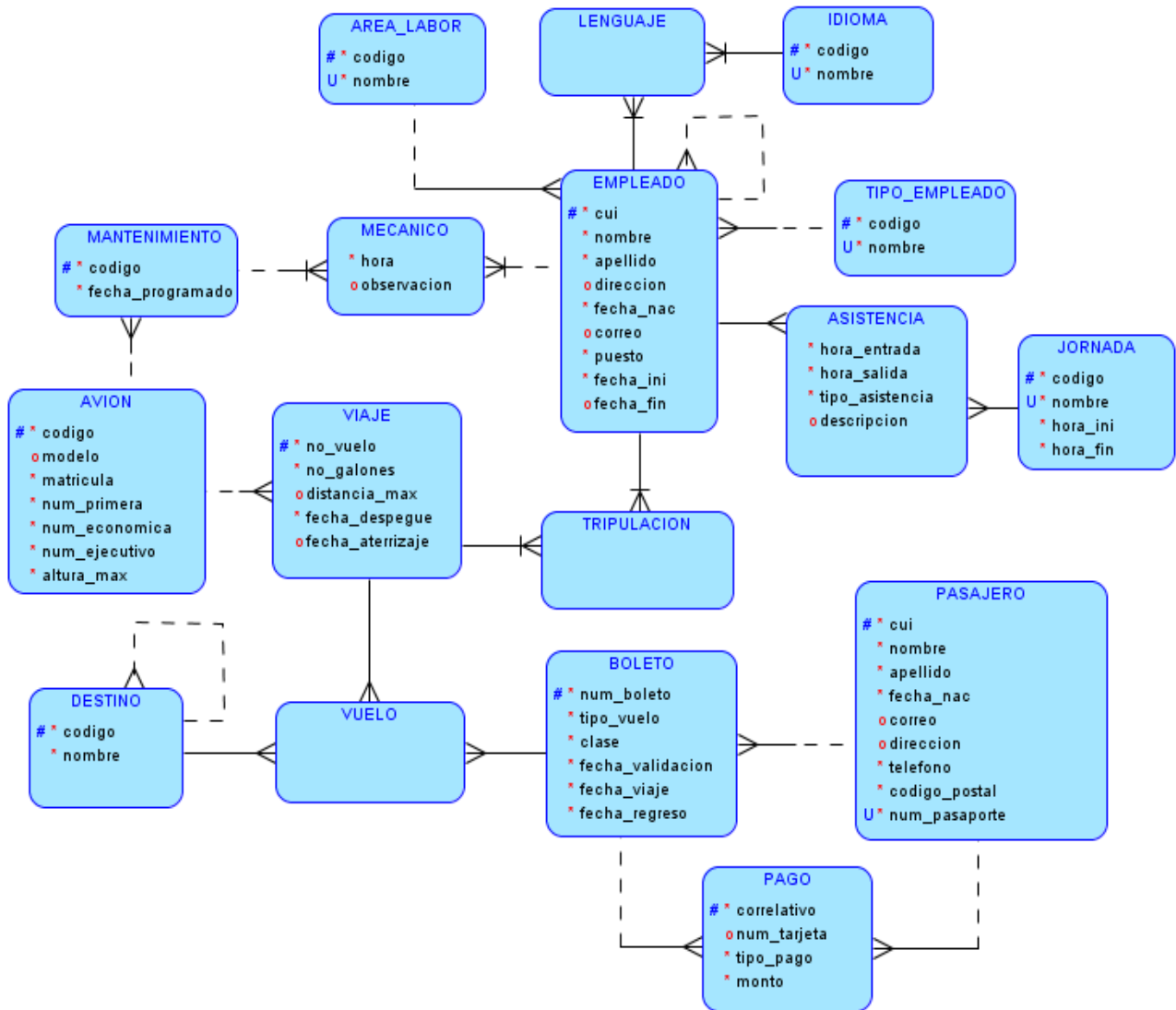
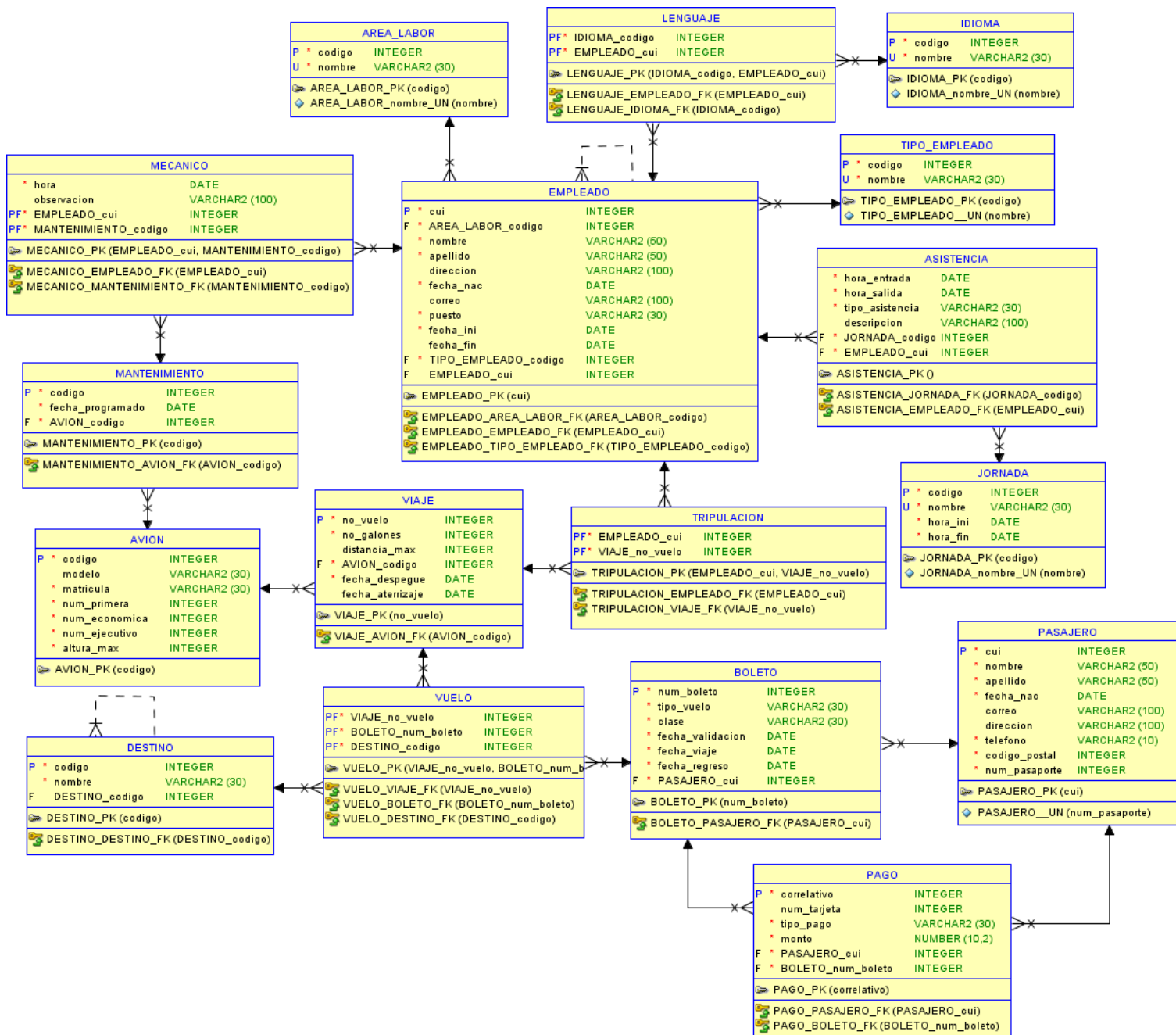


DIAGRAMA MODELO RELACIONAL



DESCRIPCION DE ENTIDADES

Nombre Entidad	Descripción
AREA_LABOR	Lugar donde un empleado labora. En donde una aerolínea necesita un área para cada funcionamiento de la organización.
LENGUAJE	Registra el lenguaje que pueden hablar los empleados.
IDIOMA	Es el idioma de pasajeros que pueden llegar a la aerolínea.
EMPLEADO	Empleados que atienden la aerolínea.
TIPO_EMPLEADO	Un empleado puede ser gerente, mecánico, mantenimiento, etc.
MECANICO	Son los mecánicos que le dan mantenimiento a un avión.

MANTENIMIENTO	Registra cuando es su próximo mantenimiento.
AVION	Registra los aviones que hay en la aerolínea.
ASISTENCIA	Registra las asistencias del empleado en el área laboral.
JORNADA	Jornada que puede haber en la organización.
VIAJE	En un viaje se registra el número de vuelo, el número de galones que tiene el avión, y la distancia máxima que puede recorrer en ese viaje.
TRIPULACION	Se registra que tripulantes están en el viaje.
DESTINO	Es el destino que puede haber en un viaje ejemplo ciudad, país, continente, isla, etc.
VUELO	Registra los boletos de los pasajeros que van en ese viaje.
BOLETO	Es el permiso que un pasajero pueda entrar en un avión.
PASAJERO	Registra los pasajeros que compran boleto.
PAGO	La entidad registra los pagos que realiza el pasajero con un boleto.

DESCRIPCION DE ATRIBUTOS POR ENTIDAD

IDIOMA

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	SI	30 caracteres máximo.

LENGUAJE

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
IDIOMA_codigo	Integer	Primaria, Foránea	SI	NO	Idioma que habla un empleado.
EMPLEADO_cui	Integer	Primaria, Foránea	SI	NO	Empleado que habla el idioma.

AREA LABOR

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	SI	30 caracteres máximo.

TIPO EMPLEADO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	SI	30 caracteres máximo.

EMPLEADO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Cui	Integer	Primaria	SI	SI	Código de identificación.

Nombre	Varchar		SI	NO	50 caracteres máximo. Donde pueden escribir todos sus nombres.
Apellido	Varchar		SI	NO	50 caracteres máximo. Donde se puede ingresar sus apellidos.
Dirección	Varchar		NO	NO	100 caracteres máximo. Dirección del empleado.
Fecha_nac	Date		SI	NO	Fecha del empleado.
Correo	Varchar		NO	NO	100 caracteres máximo. Correo que pueda tener el empleado.
Puesto	Varchar		SI	NO	30 caracteres máximo. Puesto que labora.
Fecha_ini	Date		SI	NO	Fecha inicial del contrato.
Fecha_fin	Date		NO	NO	Fecha final del contrato.
AREA_LABOR_codigo	Integer	Foránea	SI	NO	Área a la que pertenece.
TIPO_EMPLEADO_codigo	Integer	Foránea	SI	NO	Tipo de empleado que es.
EMPLEADO_cui	Integer	Foránea	NO	NO	Jefe del empleado.

MANTENIMIENTO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Fecha	Date		SI	NO	Fecha para su próximo mantenimiento.
AVION_codigo	Integer		SI	NO	Avión al que se le dará mantenimiento.

AVION

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Modelo	Varchar		SI	NO	Modelo del avión.
Matricula	Varchar		SI	NO	Matricula del avión.
Num_primera	Integer		SI	NO	Numero de primera clase que se puede viajar.

Num_economica	Integer		SI	NO	Numero de clase económica.
Num_ejecutivo	Integer		SI	NO	Numero de clase ejecutivo.
Altura_max	Integer		SI	NO	Altura máxima que puede alcanzar el avión.

VIAJE

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
No_vuelo	Integer	Primaria	SI	SI	Código de identificación.
No_galones	Integer		SI	NO	Numero de galones de gasolina que usa en el viaje.
Distancia_max	Integer		NO	NO	Distancia que puede recorrer el avión.
AVION_codigo	Integer		SI	NO	Avión que realizara el viaje.
Fecha_despegue	Date		SI	NO	Fecha en la despegara el avión.
Fecha_aterrizaje	Date		NO	NO	Fecha en el que el avión aterrizara.

TRIPULACION

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
EMPLEADO_cui	Integer	Primaria, Foránea	SI	NO	Empleado que será un tripulante.
VIAJE_no_vuelo	Integer	Primaria, Foránea	SI	NO	Viaje donde será tripulante.

JORNADA

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	SI	30 caracteres máximo.
Hora_ini	Date		SI	NO	Hora de inicio de la jornada.
Hora_fin	Date		SI	NO	Hora de fin de la jornada.

ASISTENCIA

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Hora_entrada	Date		SI	NO	Hora de entrada de un empleado.
Hora_salida	Date		SI	NO	Hora de salida de un empleado.
Tipo_asistencia	Varchar		SI	NO	30 caracteres máximo. Puede ser inasistencia o asistencia.

Descripción	Varchar		NO	NO	100 caracteres máximo. Define el porqué de la asistencia o inasistencia.
JORNADA_codigo	Integer	Foránea	SI	NO	Jornada en que labora el empleado.
EMPLEADO_cui	Integer	Foránea	SI	NO	Empleado que se le toma asistencia.

DESTINO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	NO	30 caracteres máximo.
DESTINO_codigo	Integer	Foránea	NO	NO	En el destino puede pasar por varios más, ejemplo para ir a Estados Unidos Nueva York, su destino es paradas continuas en aeropuertos hasta llegar a la ciudad.

VUELO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
VIAJE_no_vuelo	Integer	Primaria, Foránea	SI	NO	Viaje en la que realizara el vuelo.
BOLETO_num_boleto	Integer	Primaria, Foránea	SI	NO	Boleto que puede estar en el vuelo.
DESTINO_codigo	Integer	Primaria, Foránea	SI	NO	Destino del vuelo.

BOLETO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Num_boleto	Integer	Primaria	SI	SI	Código de identificación.
Tipo_vuelo	Varchar		SI	NO	30 caracteres máximo. Directo o haya aterrizajes continuos.
Clase	Varchar		SI	NO	30 caracteres máximo. Específica en clase del avión desea viajar.
Fecha_validacion	Date		SI	NO	Fecha en la que es válida el boleto.
Fecha_viaje	Date		SI	NO	Fecha en la que se viajar con el boleto.
Fecha_regreso	Date		SI	NO	Fecha en la que debe regresar el pasajero.
PASAJERO_cui	Integer	Foránea	SI	NO	Pasajero que adquirió el boleto.

PASAJERO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Cui	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	NO	50 caracteres máximo.
Apellido	Varchar		SI	NO	50 caracteres máximo.
Fecha_nac	Date		SI	NO	Fecha de nacimiento del pasajero.
Correo	Varchar		NO	NO	100 caracteres máximo.
Dirección	Varchar		NO	NO	100 caracteres máximo.
Telefono	Varchar		SI	NO	10 caracteres máximo que guardara el teléfono del pasajero.
Código_postal	Integer		SI	NO	
Num_pasaporte	Integer		SI	SI	Pasaporte que se validara para comprar un boleto.

PAGO

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Correlativo	Integer	Primaria	SI	SI	Código de identificación.
Num_tarjeta	Integer		NO	NO	Número de tarjeta del pasajero.
Tipo_pago	Varchar		SI	NO	Puede ser tarjeta de crédito, débito, etc.
Monto	Number		SI	NO	Monto que deberá de pagar por el boleto que compro.
PASAJERO_cui	Integer	Foránea	SI	NO	Pasajero que compro el boleto.
BOLETO_num_boleto	Integer	Foránea	SI	NO	Boleto en que se realizó el pago.

DESCRIPCION DE RELACIONES ENTRE ENTIDADES

Entidad1	Entidad2	Relación N:N	Descripción
IDIOMA	LENGUAJE	1:M	Un idioma esta en varios lenguajes de los empleados.
EMPLEADO	LABOR	1:M	Un empleado habla uno o varios idiomas.
AREA_LABOR	EMPLEADO	1:M	En un área de la aerolínea puede haber uno o varios empleados.
EMPLEADO	EMPLEADO	1:M	Un empleado puede ser jefe de otros empleados.
TIPO_EMPLEADO	EMPLEADO	1:M	Un empleado tiene una labor en la organización por lo que el tipo empleado es como la profesión que

			tiene. Un empleado tiene un solo labor en la organización. Mientras que un tipo empleado lo puede tener uno o varios empleados.
EMPLEADO	ASISTENCIA	1:M	Un empleado se le registra la asistencia por lo tanto un empleado se registra uno o varias asistencias.
JORNADA	ASISTENCIA	1:M	En una jornada hay uno o varias asistencias de empleados.
EMPLEADO	MECANICO	1:M	Es la asignación de un mecánico con un mantenimiento. Un empleado dependiendo del tipo de empleado puede ser asignado a uno o varios mecánicos.
EMPLEADO	TRIPULACION	1:M	Es la asignación de un tripulante con un viaje. Un empleado puede ser asignado a uno o varios viajes pero no en el mismo tiempo.
MANTENIMIENTO	MECANICO	1:M	Para un mantenimiento se les asignan a varios mecánicos.
AVION	MANTENIMIENTO	1:M	A un avión se le puede programar varios mantenimientos.
AVION	VIAJE	1:M	A un avión se le puede asignar a uno o varios viajes, no a la vez si no que en un determinado tiempo.
VIAJE	TRIPULACION	1:M	A un viaje se le asigna tripulantes.
VIAJE	VUELO	1:M	En un viaje hay vuelos para diferentes tipos de boletos y destinos que puede ser continuos.
DESTINO	VUELO	1:M	En un destino llegan varios vuelos.
DESTINO	DESTINO	1:M	En un destino puede haber otros destinos que lleguen antes que este.
BOLETO	VUELO	1:M	En boleto se puede realizar más viajes, suponiendo que el boleto sea para ida y vuelta.
BOLETO	PAGO	1:M	Un boleto se le puede pagar en varios pagos.
PASAJERO	BOLETO	1:M	Un pasajero puede comprar varios boletos.
PASAJERO	PAGO	1:M	Un pasajero realiza el pago de uno o varios boletos que desea comprar.

DESCRIPCION DE LAS RESTRICCIONES

Unique: La restricción unique identifica de manera única a cada fila de una tabla.

Primary Key: Identifica de manera única cada fila de una tabla.

Foreign key: Es una referencia a otra tabla. Puede aceptar valores nulos.

CONSIDERACIONES QUE SE TOMARON AL MOMENTO DE DISEÑAR EL DIAGRAMA

1. La matrícula para el convenio sobre aviación civil internacional se le pone un prefijo TG seguido de una combinación de letras diferente seria con una restricción a la columna matricula.
2. Para el mantenimiento de los aviones se consideró que debe haber mantenimiento cada cierto tiempo que son programados.
3. La asignación de los 8 mecánicos se le colocara un trigger para que valide cuantos mecánicos hay asignados al mantenimiento.
4. Se tomó en cuenta que para cada empleado puede existir un jefe.
5. Para saber cuántas horas acumuladas lleva un tripulante se puede hacer una consulta sumando la diferencia de la hora de salida y entrada de la asistencia.
6. El peso del equipaje no se lleva registro del peso, se tomó como una regla del negocio que los empleados llevan.
7. Para la tripulación se realiza un trigger para que solo 10 tripulantes puedan estar en un viaje.

SCRIPT PARA EL DBMS DE ORACLE

```
create table AREA_LABOR (
```

```
    codigo integer not null,
```

```
    nombre varchar(30) constraint AREA_LABOR_nombre_UN unique not null,
```

```
    constraint AREA_LABOR_PK primary key (codigo)
```

```
);
```

```
create table ASISTENCIA (
```

```
    hora_entrada date not null,
```

```
    hora_salida date not null,
```

```
    tipo_asistencia varchar(30) not null,
```

```
    descripcion varchar(100),
```

```
    JORNADA_codigo integer not null,
```

```
    EMPLEADO_cui integer not null,
```

```
    constraint ASISTENCIA_EMPLEADO_FK foreign key (EMPLEADO_cui) references EMPLEADO (cui),
```

```
    constraint ASISTENCIA_JORNADA_FK foreign key (JORNADA_codigo) references JORNADA (codigo)
```

```
);
```

```
create table AVION (  
    codigo    integer not null,  
    modelo    varchar(30),  
    matricula  varchar(30) not null,  
    num_primera integer not null,  
    num_economica integer not null,  
    num_ejecutivo integer not null,  
    altura_max integer not null,  
    constraint AVION_PK primary key (codigo)  
);
```

```
create table BOLETO (  
    num_boleto    integer not null,  
    tipo_vuelo    varchar(30) not null,  
    clase         varchar(30) not null,  
    fecha_validacion date not null,  
    fecha_viaje   date not null,  
    fecha_regreso date not null,  
    PASAJERO_cui  integer not null,  
    constraint BOLETO_PK primary key (num_boleto),  
    constraint BOLETO_PASAJERO_FK foreign key (PASAJERO_cui) references PASAJERO (cui)  
);
```

```
create table DESTINO (  
    codigo    integer not null,  
    nombre    varchar(30) not null,  
    DESTINO_codigo integer,  
    constraint DESTINO_PK primary key (codigo),  
    constraint DESTINO_DESTINO_FK foreign key (DESTINO_codigo) references DESTINO (codigo)  
);
```

```

create table EMPLEADO (
    cui            integer not null,
    AREA_LABOR_codigo integer not null,
    nombre        varchar(50) not null,
    apellido       varchar(50) not null,
    direccion      varchar(100),
    fecha_nac      date not null,
    correo         varchar(100),
    puesto         varchar(30) not null,
    fecha_ini      date not null,
    fecha_fin      date,
    TIPO_EMPLEADO_codigo integer not null,
    EMPLEADO_cui   integer,
    constraint EMPLEADO_PK primary key (cui),
    constraint EMPLEADO_AREA_LABOR_FK foreign key (AREA_LABOR_codigo) references
AREA_LABOR (codigo),
    constraint EMPLEADO_TIPO_EMPLEADO_FK foreign key (TIPO_EMPLEADO_codigo)
references TIPO_EMPLEADO (codigo),
    constraint EMPLEADO_EMPLEADO_FK foreign key (EMPLEADO_cui) references EMPLEADO
(cui)
);

```

```

create table IDIOMA (
    codigo integer not null,
    nombre varchar(30) constraint IDIOMA_nombre_UN unique not null,
    constraint IDIOMA_PK primary key (codigo)
);

```

```

create table JORNADA (
    codigo integer not null,
    nombre varchar(30) constraint JORNADA_nombre_UN unique not null,
    hora_ini date not null,

```

```
hora_fin date not null,  
constraint JORNADA_PK primary key (codigo)  
);
```

```
create table LENGUAJE (  
    IDIOMA_codigo integer not null,  
    EMPLEADO_cui integer not null,  
    constraint LENGUAJE_PK primary key (IDIOMA_codigo, EMPLEADO_cui),  
    constraint LENGUAJE_EMPLEADO_FK foreign key (EMPLEADO_cui) references EMPLEADO  
(cui),  
    constraint LENGUAJE_IDIOMA_FK foreign key (IDIOMA_codigo) references IDIOMA (codigo)  
);
```

```
create table MANTENIMIENTO (  
    codigo integer not null,  
    fecha_programado date not null,  
    AVION_codigo integer not null,  
    constraint MANTENIMIENTO_PK primary key (codigo),  
    constraint MANTENIMIENTO_AVION_FK foreign key (AVION_codigo) references AVION  
(codigo)  
);
```

```
create table MECANICO (  
    hora date not null,  
    observacion varchar(100),  
    EMPLEADO_cui integer not null,  
    MANTENIMIENTO_codigo integer not null,  
    constraint MECANICO_PK primary key (EMPLEADO_cui, MANTENIMIENTO_codigo),  
    constraint MECANICO_EMPLEADO_FK foreign key (EMPLEADO_cui) references EMPLEADO  
(cui),  
    constraint MECANICO_MANTENIMIENTO_FK foreign key (MANTENIMIENTO_codigo)  
references MANTENIMIENTO (codigo)
```

);

```
create table PAGO (  
    correlativo    integer not null,  
    num_tarjeta    integer,  
    tipo_pago      varchar(30) not null,  
    monto          NUMBER (10,2) not null,  
    PASAJERO_cui   integer not null,  
    BOLETO_num_boleto integer not null,  
    constraint PAGO_PK primary key (correlativo),  
    constraint PAGO_BOLETO_FK foreign key (BOLETO_num_boleto) references BOLETO  
(num_boleto),  
    constraint PAGO_PASAJERO_FK foreign key (PASAJERO_cui) references PASAJERO (cui)  
);
```

```
create table PASAJERO (  
    cui           integer not null,  
    nombre        varchar(50) not null,  
    apellido      varchar(50) not null,  
    fecha_nac     date not null,  
    correo        varchar(100),  
    direccion     varchar(100),  
    telefono      varchar(10) not null,  
    codigo_postal integer not null,  
    num_pasaporte integer constraint Pa_num_pasa_un unique not null,  
    constraint PASAJERO_PK primary key (cui)  
);
```

```
create table TIPO_EMPLEADO (  
    codigo integer not null,  
    nombre varchar(30) constraint TIPO_EMPLEADO__UN unique not null,
```

```
constraint TIPO_EMPLEADO_PK primary key (codigo)
);
```

```
create table TRIPULACION (
    EMPLEADO_cui integer not null,
    VIAJE_no_vuelo integer not null,
    constraint TRIPULACION_PK primary key (EMPLEADO_cui, VIAJE_no_vuelo),
    constraint TRIPULACION_EMPLEADO_FK foreign key (EMPLEADO_cui) references EMPLEADO
(cui),
    constraint TRIPULACION_VIAJE_FK foreign key (VIAJE_no_vuelo) references VIAJE (no_vuelo)
);
```

```
create table VIAJE (
    no_vuelo integer not null,
    no_galones integer not null,
    distancia_max integer,
    fecha_despegue date not null,
    fecha_aterrizaje date,
    AVION_codigo integer not null,
    constraint VIAJE_PK primary key (no_vuelo),
    constraint VIAJE_AVION_FK foreign key (AVION_codigo) references AVION (codigo)
);
```

```
create table VUELO (
    VIAJE_no_vuelo integer not null,
    BOLETO_num_boleto integer not null,
    DESTINO_codigo integer not null,
    constraint VUELO_PK primary key (VIAJE_no_vuelo, BOLETO_num_boleto, DESTINO_codigo)
    constraint VUELO_BOLETO_FK foreign key (BOLETO_num_boleto) references BOLETO
(num_boleto),
    constraint VUELO_DESTINO_FK foreign key (DESTINO_codigo) references DESTINO (codigo),
    constraint VUELO_VIAJE_FK foreign key (VIAJE_no_vuelo) references VIAJE (no_vuelo)
);
```


);

ENUNCIADO 3

DIAGRAMA ENTIDAD RELACION

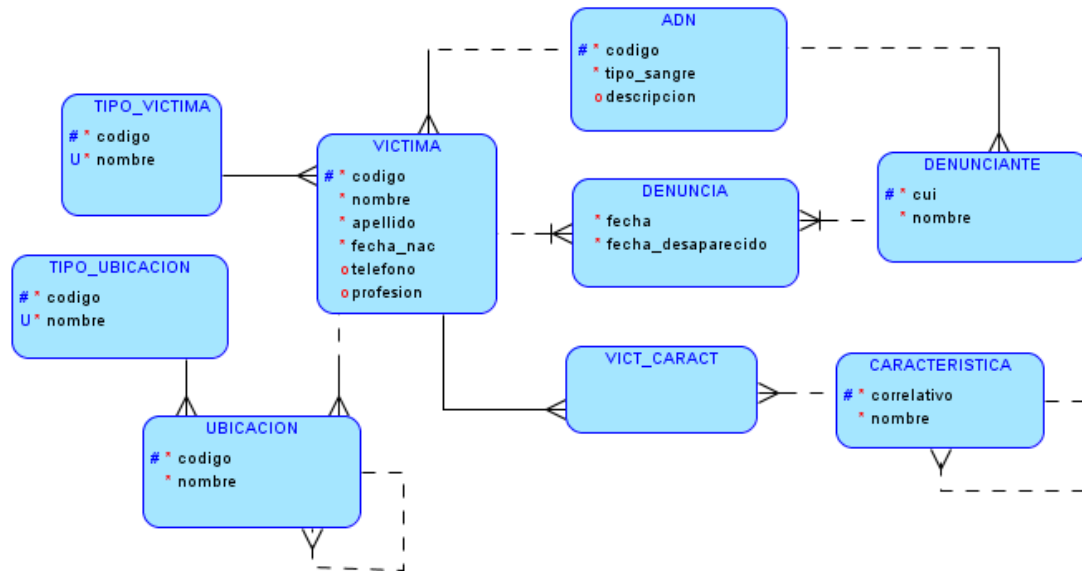
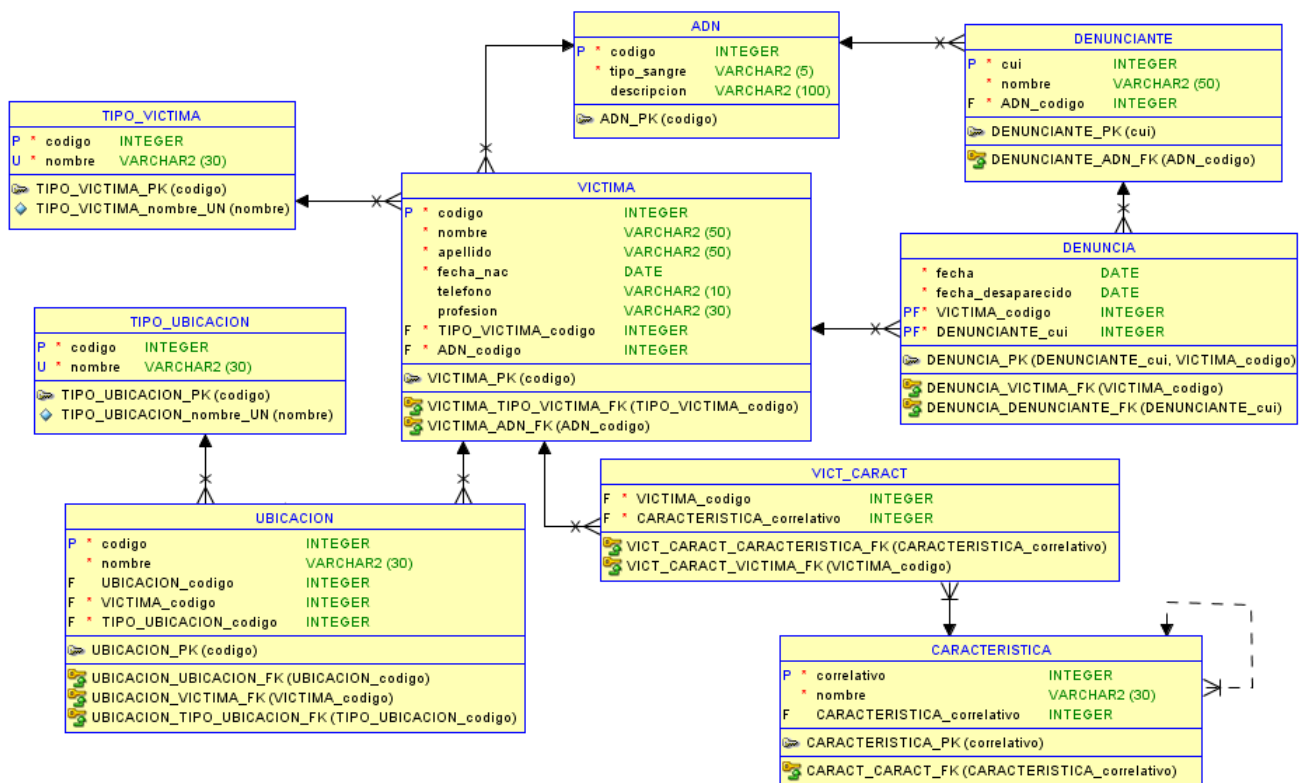


DIAGRAMA MODELO RELACIONAL



DESCRIPCION DE ENTIDADES

Nombre Entidad	Descripción
ADN	Guarda el registro de ADN de las víctimas y denunciantes.
TIPO_VICTIMA	Puede ser que sean personas vivas o muertas.
DENUNCIA	Son registros que se almacena para identificar a victimas desaparecidas.
DENUNCIANTE	Persona que reporta cuando una víctima está desaparecida.
VICTIMA	Persona que fue víctima del conflicto armado.
TIPO_UBICACION	Puede ser la ubicación de una residencia de una víctima viva o puede ser el lugar de hallazgo de una victima muerta.
UBICACIÓN	Es la ubicación de un lugar donde se encuentra la víctima.
CARACTERISTICA	Es la característica que puede tener la víctima o persona desaparecida. Ejemplo que lleve puesto camisa azul, tenía una gorra, etc.
VICT_CARACT	Una víctima se le puede registrar varias características.

DESCRIPCION DE ATRIBUTOS POR ENTIDAD

ADN

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Tipo_sangre	Varchar		SI	NO	5 caracteres máximo.
Descripción	Varchar		NO	NO	100 caracteres máximo. Descripción del tipo de sangre.

DENUNCIANTE

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Cui	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	NO	50 caracteres máximo.
ADN_codigo	Integer	Foránea	SI	NO	Registra el tipo de sangre del denunciante.

DENUNCIA

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Fecha	Date		SI	NO	Fecha que se reporto una denuncia.
Fecha_desaparecido	Date		SI	NO	Fecha que la víctima está desaparecido.
VICTIMA_codigo	Integer	Primaria, Foránea	SI	NO	Victima que se está reportando.
DENUNCIANTE_cui	Integer	Primaria, Foránea	SI	NO	Denunciante que realiza la denuncia.

VICTIMA

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
----------	------	------------	-------------	-------	-------------

Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	NO	50 caracteres máximo.
Apellido	Varchar		SI	NO	50 caracteres máximo.
Fecha_nac	Date		SI	NO	Fecha de nacimiento de la víctima.
Teléfono	Varchar		NO	NO	Teléfono que tenía o tiene.
Profesión	Varchar		NO	NO	La profesión que se dedica o dedicaba.
TIPO_VICTIMA_codigo	Integer	Foránea	SI	NO	Identifica si la persona está viva o no.
ADN_codigo	Integer	Foránea	SI	NO	Es el ADN de la víctima.

TIPO_VICTIMA

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	SI	30 caracteres máximo.

TIPO_UBICACION

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	SI	30 caracteres máximo.

UBICACIÓN

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Código	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	NO	30 caracteres máximo.
UBICACIÓN_codigo	Integer	Foránea	SI	NO	Registra otra ubicación que este está por ejemplo ubicación 7 avenida 13 calle zona 1.
VICTIMA_codigo	Integer	Foránea	SI	NO	Es la ubicación de la víctima.
TIPO_UBICACION_codigo	Integer	Foránea	SI	NO	Especifica si es residencial o no.

CARACTERISTICA

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
Correlativo	Integer	Primaria	SI	SI	Código de identificación.
Nombre	Varchar		SI	NO	30 caracteres máximo.
CARACTERITICA_correlativo	Integer	Foránea	SI	NO	Puede ser que esta característica lo defina otra característica.

VICT_CARACT

Atributo	Tipo	Tipo Llave	Obligatorio	Único	Descripción
VICTIMA_codigo	Integer	Foránea	SI	NO	Victima al que se le define varias características.
CARACTERISTICA_correlativo	Integer	Foránea	SI	NO	Característica de la víctima.
ADN_codigo	Integer	Foránea	SI	NO	Registra el tipo de sangre del denunciante.

DESCRIPCION DE RELACIONES ENTRE ENTIDADES

Entidad1	Entidad2	Relación N:N	Descripción
ADN	VICTIMA	1:M	Un examen de ADN lo realiza varias victimas.
ADN	DENUNCIANTE	1:M	Un denunciante también puede realizar pruebas de ADN por lo que se quedan registrados.
DENUNCIANTE	DENUNCIA	1:M	Un denunciante realiza una o varias denuncias.
VICTIMA	DENUNCIA	1:M	Una víctima puede estar en uno o varias denuncias. Por el motivo varios denunciantes pueden denunciar a la víctima.
VICTIMA	UBICACIÓN	1:M	Una víctima tiene una o varias ubicaciones. Ya sea varias residencias o el peor de los casos que sea descuartizado.
VICTIMA	VICT_CARACT	1:M	Una víctima tiene varias características.
TIPO_UBICACION	UBICACIÓN	1:M	Un tipo de ubicación esta en varias ubicaciones.
CARACTERISTICA	VICT_CARACT	1:M	Una característica está presente en uno o varias VICT_CARACT o sea en varias víctimas.

CARACTERISTICA	CARACTERISTICA	1:M	Una característica puede tener más características.
----------------	----------------	-----	---

DESCRIPCION DE LAS RESTRICCIONES

Unique: La restricción unique identifica de manera única a cada fila de una tabla.

Primary Key: Identifica de manera única cada fila de una tabla.

Foreign key: Es una referencia a otra tabla. Puede aceptar valores nulos.

CONSIDERACIONES QUE SE TOMARON AL MOMENTO DE DISEÑAR EL DIAGRAMA

1. Los registros de ADN son pruebas que se hicieron para cada persona ya sea denunciante o víctima.
2. Las desapariciones reportadas son las denuncias que hace un denunciante.
3. La edad y altura se tomaron parte de las características que pueden tener las víctimas también la complexión y etnia a la que pertenece, al igual si poseía un objeto de valor o algún distintivo.
4. Las victimas sin reconocer son las que están registrados en la tabla víctimas.
5. Se tomó como tipo de victima a las víctimas que puede estar vivas o muertas.
6. Características es recursiva porque dentro de las mismas características puede haber otras.
7. Ubicación es recursiva porque en ubicación pueden haber más ubicaciones como por ejemplo Departamento de Guatemala zona 5 7ma calle 12 calle 14-34.

SCRIPT PARA EL DBMS DE ORACLE

```
create table ADN (
```

```
    codigo    integer not null,
```

```
    tipo_sangre varchar(5) not null,
```

```
    descripcion varchar(100),
```

```
    constraint ADN_PK primary key (codigo)
```

```
);
```

```
create table CARACTERISTICA (
```

```
    correlativo    integer not null,
```

```
    nombre          varchar(30) not null,
```

```
    CARACTERISTICA_correlativo integer,
```

```
    constraint CARACTERISTICA_PK primary key (correlativo),
```

```
constraint CARACT_CARACT_FK foreign key (CARACTERISTICA_correlativo) references  
CARACTERISTICA (correlativo)  
);
```

```
create table DENUNCIA (  
    fecha          date not null,  
    fecha_desaparecido date not null,  
    VICTIMA_codigo integer not null,  
    DENUNCIANTE_cui integer not null,  
    constraint DENUNCIA_PK primary key (DENUNCIANTE_cui, VICTIMA_codigo),  
    constraint DENUNCIA_DENUNCIANTE_FK foreign key (DENUNCIANTE_cui) references  
DENUNCIANTE (cui),  
    constraint DENUNCIA_VICTIMA_FK foreign key (VICTIMA_codigo) references VICTIMA  
(codigo)  
);
```

```
create table DENUNCIANTE (  
    cui          integer not null,  
    nombre       varchar(50) not null,  
    ADN_codigo integer not null,  
    constraint DENUNCIANTE_PK primary key (cui),  
    constraint DENUNCIANTE_ADN_FK foreign key (ADN_codigo) references ADN (codigo)  
);
```

```
create table TIPO_UBICACION (  
    codigo integer not null,  
    nombre varchar(30) constraint TIPO_UBICACION_nombre_UN unique not null,  
    constraint TIPO_UBICACION_PK primary key (codigo)  
);
```

```
create table TIPO_VICTIMA (  
    codigo integer not null,
```

```
    nombre varchar(30) constraint TIPO_VICTIMA_nombre_UN unique not null,  
    constraint TIPO_VICTIMA_PK primary key (codigo)  
);
```

```
create table UBICACION (  
    codigo          integer not null,  
    nombre          varchar(30) not null,  
    UBICACION_codigo integer,  
    VICTIMA_codigo  integer not null,  
    TIPO_UBICACION_codigo integer not null,  
    constraint UBICACION_PK primary key (codigo),  
    constraint UBICACION_TIPO_UBICACION_FK foreign key (TIPO_UBICACION_codigo)  
references TIPO_UBICACION (codigo),  
    constraint UBICACION_UBICACION_FK foreign key (UBICACION_codigo) references  
UBICACION (codigo),  
    constraint UBICACION_VICTIMA_FK foreign key (VICTIMA_codigo) references VICTIMA  
(codigo)  
);
```

```
create table VICTIMA (  
    codigo          integer not null,  
    nombre          varchar(50) not null,  
    apellido        varchar(50) not null,  
    fecha_nac       date not null,  
    telefono        varchar(10),  
    profesion        varchar(30),  
    TIPO_VICTIMA_codigo integer not null,  
    ADN_codigo      integer not null,  
    constraint VICTIMA_PK primary key (codigo),  
    constraint VICTIMA_ADN_FK foreign key (ADN_codigo) references ADN (codigo),  
    constraint VICTIMA_TIPO_VICTIMA_FK foreign key (TIPO_VICTIMA_codigo) references  
TIPO_VICTIMA (codigo)
```

);

create table VICT_CARACT (

VICTIMA_codigo integer not null,

CARACTERISTICA_correlativo integer not null,

constraint VICT_CARACT_CARACTERISTICA_FK foreign key (CARACTERISTICA_correlativo)
references CHARACTERISTICA (correlativo),

constraint VICT_CARACT_VICTIMA_FK foreign key (VICTIMA_codigo) references VICTIMA
(codigo)

);