

Practica 7

Agenda Personal

7.1. ENUNCIADO DEL PROBLEMA

Se desea desarrollar un sistema casero para el mantenimiento de la información correspondiente a una agenda personal.

En una agenda personal se mantiene información correspondiente a una serie de personas que son de interés para el propietario de la misma. Esta información contiene datos acerca de sus nombres, direcciones en donde pueden ser localizados (vivienda y/o trabajo, generalmente), número de teléfonos, fax, e-mail y toda aquella información que permita su clasificación en las entradas de la agenda y su fácil y rápida localización.

En este problema, se van a considerar los siguientes supuestos semánticos:

Supuesto 1: Cada entrada en la agenda podrá corresponder a personas físicas o jurídicas, siendo necesario representar esta característica.

Supuesto 2: Si existen entradas en la agenda de personas físicas relacionadas con entradas jurídicas, esta relación debe ser presentada.

Supuesto 3: Para cualquier entrada en la agenda, se almacenará:

- Un nombre (o nombre y apellidos) que identifique la entrada.
- El dni o nif, si éste fuera conocido.
- La dirección o direcciones completas asociadas.
- Los número de teléfono, e-mail y fax.

Supuesto 4: Además, para cada entrada, interesa almacenar ciertas fechas relacionadas con la misma. Por ejemplo, para las entradas correspondientes a personas físicas, es de interés almacenar la fecha de nacimiento, aniversario de boda, etc.

Así, los apuntes correspondientes a estas fechas deberán estar clasificados en categorías o tipos de fechas correspondientes a la entrada.

Supuesto 5: Una entrada en la agenda puede tener varias direcciones (casa en el centro, casa de campo, apartamento en la playa, trabajo, etc), debiendo todas ellas almacenarse con la identificación del tipo a que corresponda.

Supuesto 6: Cada entrada en la agenda puede tener varios números de fax, teléfono y e-mail de contacto, los cuáles pueden estar vinculados a direcciones o no.

Supuesto 7: Cada entrada en la agenda tiene asignado un tipo o característica de la entrada. Estos tipos se generalizan en: *Compañías y Particulares*, y éstos se especializan en, por ejemplo: gas, luz, agua, etc., amigo, conocido, mujer de., etc.

Supuesto 8: No existen dos entradas en la agenda repetidas; es decir, para simplificar se considera que no hay nombres repetidos ni de particulares ni de compañías.

Supuesto 9: Una dirección e-mail siempre está asociada a una entrada de la agenda y no a una dirección; mientras que un teléfono o un fax puede estar asociado a una dirección o no, ya que pueden ser móviles.

Supuesto 10: De igual forma, las fechas relacionadas con las entradas en la agenda no se encuentran vinculadas con las direcciones que pudieran estar, a su vez, relacionadas con éstas, sino únicamente con la entrada en sí.

7.2 SUPUESTOS AÑADIDOS

Supuesto 11: Para una misma persona no puede existir más de un tipo de dirección igual.

Supuesto 12: Para una misma persona no puede existir más de un tipo de fecha igual.

Supuesto 13: Una misma persona se podrá relacionar con más de una persona y está relación sólo podrá ser entre personas físicas y personas jurídicas o viceversa, pero no entre personas del mismo tipo, es decir, no se permitiría una relación de una persona física con otra física o jurídica con jurídica.

Supuesto 14: Para que una persona se incluya en la agenda no es necesario que se tenga obligatoriamente conocimiento simultaneo por un lado de su teléfono, fax, e-mail o por el otro lado de su dirección. Es decir existirán entradas en la agenda de las que no dispondremos de su dirección pero si de su teléfono fax o e-mail, o viceversa.

Supuesto 15: Podrán existir los mismos números de teléfonos, faxes y direcciones de correo electrónico para más de una entrada, ya que varias personas pueden compartir sus formas de contacto e incluso sus direcciones de correo electrónico. Idem para las direcciones.

7.3 DESCRIPCION DETALLADA DEL PROBLEMA.

7.3.1 ELEMENTOS DEL SISTEMA.

A continuación se detalla la información que desea mantenerse sobre los elementos que forman parte del sistema.

Entrada de la agenda

Datos personales: Nombre completo (servirá como identificativo), apuntes (almacenaremos la relación que tiene con nosotros), fechas y tipos de fechas asociadas a la entrada, formas de contacto (teléfono, e-mail, móviles), direcciones asociadas, mantendremos información acerca de si nuestras entradas son personas físicas o personas jurídicas y las posibles relaciones entre ellas, ubicación de teléfonos.

Direcciones: Almacenaremos información acerca de la calle, número, Código postal y localidad.

7.4 MODELO CONCEPTUAL.

Se trata de representar la información asociada a las personas que podremos ir introduciendo en nuestra agenda y representar las posibles relaciones existentes entre las personas físicas y las personas jurídicas.

7.4.1 ANALISIS DE LOS TIPOS DE ENTIDAD

Tipo de entidad **Entrada:** representa la entrada a la agenda personal. Según el supuesto 1 se considera que esta entrada puede corresponder a personas físicas o jurídicas. Además como se dice que hay que representar esta característica es necesario especializar este tipo de entidad de forma total y exclusiva en compañías y

particulares. Existe una interrelación entre las dos especializaciones que se debe al supuesto 2 ya que hay que representar que una entrada a la agenda de personas físicas puede estar relacionada con entradas jurídicas. Para cualquier tipo de persona hay que considerar unos atributos comunes (según el supuesto 3) que son: nombre (o nombre y apellidos) que es el identificador principal (supuesto 3) ya que no se repite, dni o nif si estos fueran conocidos (esto quiere decir que puede tomar valores nulos). Debemos considerar un atributo nombre indivisible, ya que a priori no se puede saber los componentes de este atributo. Además los nombres pueden incluso carecer de apellidos, ser pseudónimos, ser nombres de empresas, instituciones etc.

Consideramos también el atributo apuntes que nos permitirá almacenar a efectos identificativos del propio usuario de la Agenda Personal la relación que mantiene con la persona (Entrada) que se ha introducido en la agenda.

Tipo de entidad **Forma_contacto**: el cual representa las diferentes formas de contactar con la entrada. En el enunciado se consideran tres tipos de formas de contactar con las entradas: Fijo, móvil y e-mail.

Hemos considerado conveniente establecer un atributo común definido en el mismo dominio (alfanumérico) que llamamos alfanumérico para almacenar tanto números de teléfonos, faxes como direcciones de correo electrónico.

Especializamos Forma_contacto en MOVIL, FIJO y E-MAIL. Además MOVIL y FIJO llevarán unos atributos particulares que indicarán el tipo de dispositivo al que se refiere, es decir si se refiere a un fax o a un teléfono. Además de esto, la especialización es conveniente debido a que estos subtipos se relacionan también de forma diferente con respecto a otras entidades que participan en el problema (supuesto 6,9).

Este tipo de entidad es débil por existencia (supuesto 14), existen entradas que no tendrán asociada ninguna forma de contacto.

El identificador será el atributo alfanumérico.

Tipo de entidad **Direcciones**: este tipo de entidad representa las direcciones que pueden estar asociadas a las entradas.

Este tipo de entidad es débil por identificación, porque aunque una dirección exista por si sola, en nuestra agenda personal, la entidad direcciones no podrá ser identificada (diferenciada del resto de las entidades del mismo tipo) ya que pueden existir varias direcciones idénticas pero es la entrada la que hace diferenciar una misma dirección, (tipo de dirección).

Los atributos que se consideran para esta entidad son dirección que es la dirección completa y tipo de dirección que especifica el tipo de dirección (supuesto 5).

El atributo identificador es el agregado de los atributos tipo_direccion y nombre_completo. Este agregado es necesario debido a que pueden aparecer varias direcciones para una misma persona, por lo tanto nombre_completo no sería un identificador válido por si solo.

Tipo de entidad **Fechas**: esta entidad representa las fechas que se desean mantener y relacionar con las entradas.

Este tipo de entidad es débil por identificación con respecto al tipo de entidad Entrada ya que pueden existir para diferentes entradas el mismo tipo de fecha con su fecha. Los atributos que se consideran para este tipo de entidad son tipo_fecha y fecha (supuesto 4).

El identificador de este tipo de entidad será la agregación de los atributos tipo_de_fecha y nombre_completo

heredado del tipo de entidad ENTRADA.

7.4.2 ANALISIS DE LOS TIPOS DE INTERRELACIÓN

Los tipos de entidades antes mencionadas se encuentran relacionadas de la siguiente forma:

Tipo de interrelación **Entradas/Fechas(E-F)**: el cual relaciona los tipos de entidad ENTRADA, que participa con una cardinalidad (1,1) (una fecha es asignada a una entrada) y FECHAS, que participa con cardinalidades (0,n) (una entrada puede tener ninguna, una o varias fechas). Además FECHAS sólo se relaciona con ENTRADA (supuesto 10). Este tipo de interrelación es débil por identificación con respecto a FECHAS.

Tipo de interrelación **Entrada/FormaContacto (E-FC)**: el cual relaciona los tipos de entidad ENTRADA y FORMA_CONTACTO. El tipo de entidad FORMA_CONTACTO participa con cardinalidad (0,n) ya que una entrada puede tener asociadas ninguna o varias formas de contacto (supuesto 14). El tipo de entidad ENTRADA participa con cardinalidad (1,n) ya que una FORMA_CONTACTO debe estar relacionada al menos con una entrada pero pueden existir personas que comparten la misma FORMA_CONTACTO por diferentes razones (supuesto 15). Este tipo de interrelación es débil por existencia con respecto a FORMA_CONTACTO.

Tipo de interrelación **Entrada/Direcciones (E-D)**: el cual relaciona los tipos de entidad ENTRADA y DIRECCIONES. El tipo de entidad DIRECCIONES participa con cardinalidad (0,n) ya que una entrada puede tener asociado ninguna dirección o varias (supuesto 14). El tipo de entidad ENTRADA participa con cardinalidad (1,1) ya que una dirección si existe estará relacionada con una única entrada, porque aunque existan varias direcciones iguales el significado que tiene para una entrada no tiene porque ser el mismo para otra, por eso consideraremos la interrelación débil por identificación.

Tipo de interrelación **Fijo/Direcciones (F-D)**: el cual relaciona los tipos de entidad FIJO y DIRECCIONES. El tipo de entidad FIJO participa con cardinalidad (0,n) ya que puede haber más de un fijo en una misma dirección y DIRECCION participa con cardinalidad (0,1) ya que un fijo puede estar en una dirección como máximo y cero porque se puede desconocer la ubicación del teléfono o fax (supuesto14). Aunque en esta interrelación participan dos tipos de entidades débiles con respecto al problema, pero no entre ellas, por eso consideraremos una interrelación fuerte.

Tipo de interrelación **Persona_f/Persona_j (P_F-P_J)**: en la cual participan los subtipos PERSONA_FISICA y PERSONA_JURIDICA. El tipo de entidad PERSONA_FISICA participa con cardinalidad (0,n) ya que una persona física puede estar relacionada con más de una persona jurídica (Ej: pluriempleado). Para la entidad PERSONA JURIDICA se puede aplicar el mismo razonamiento anteriormente expuesto (Ej:empleo a más de una persona).

Nota: Se adjuntan al final de la documentación los esquemas conceptuales antes y depues de haber aplicado las prereglas y las reglas de derivación.

7.4.3 DEFINICIÓN SINTÁCTICA DEL PROBLEMA.

De acuerdo con cada uno de los tipos de entidad e interrelaciones reconocidos anteriormente y de la estructura de cada uno de estos tipos, se puede representar este problema mediante la siguiente estructura sintáctica:

AgendaPersonal				
Definición de los Tipos de Entidad(tabla-1)				
Atributos	Dominio	*	tipo	Restricciones
Entrada				

IP	Nombre	cadena(30)	1	a-z,	
<i>Apuntes</i>		cadena(10)	1	a-z	<i>Juridico, Fisico</i>
Personafisica		hereda de Entrada			
IP	Nombre	cadena(30)	1	a-z	<i>heredado</i>
<i>Dni</i>		Entero(8)	1	0-9	
Personajuridica		hereda de Entrada			
IP	Nombre	cadena(10)	1	a-z	<i>heredado</i>
<i>Cif</i>		Cadena(20)	1	a-z,0-9	
Fechas		hereda de Entrada			
IP	Nombre	cadena(30)	1	a-z	<i>heredado</i>
IP	Tipo_fecha	cadena(20)	1	a-z	
<i>fecha</i>		date	1	date	
Forma_contacto					
IP	Alfanumerico	cadena(30)	1	a-z, 0-9	
Movil		Hereda de Forma_contacto			
IP	Alfanumerico	cadena(30)	1	a-z, 0-9	<i>heredado</i>
<i>tipo</i>		cadena(10)	1	a-z	
Fijo		Hereda de Forma_contacto			
IP	Alfanumerico	cadena(30)	1	a-z, 0-9	<i>heredado</i>
<i>tipo</i>		cadena(10)	1	a-z	
E-mail		Hereda de Forma_contacto			
IP	Alfanumerico	cadena(30)	1	a-z, 0-9	<i>heredado</i>
Direcciones		hereda de Entrada			
IP	Nombre	cadena(30)	1	a-z	<i>heredado</i>
IP	Tipo_direccion	cadena(20)	1	a-z	
<i>Calle</i>		cadena(20)	1	a-z	
<i>Localidad</i>		cadena(30)	1	a-z	
<i>numero</i>		entero(4)	1	0-9	
<i>C_P</i>		entero(5)	1	0-9	

Definición de los Tipos de Interrelación(tabla-1)

Entidad	m	M	Entidad	m	M	Atributos(tabla-1)		
Es_Un	Jerárquica exclusiva de Entrada							
persona- fisica	0,1	1	persona- juridica	0,1	1	tipo = Fisico,Juridico		
Es_Un	Jerárquica exclusiva de Forma-Contacto							
Movil	0	1	Fijo	0	1	Email	0	1
						categoria= Movil,Fijo,E-mail		
E-D								
Entrada	1	1	Direcciones	0	n			
F-D								
Fijo	0	n	Direcciones	0	n			
Pf-Pj								

Persona- Fisica	0	n	Persona- Juridica	0	n
Fc-E					
Forma-contacto	0	n	Entrada Juridica	0	n
F-E					
Fechas	0	n	Entrada	1	1

7.5 MODELO RELACIONAL.

El proceso de obtención del esquema relacional consiste en la aplicación de las reglas de transformación al modelo conceptual descrito. Nuestro proceso inicial consiste en la eliminación de los atributos múltiples y posteriormente de las jerarquías.

7.5.1 CREACIÓN DE LAS TABLAS.

En este apartado describiremos tanto la eliminación de las jerarquías como de los atributos múltiples y además se va a proceder a la aplicación de las reglas **pretecar** para posteriormente aplicar las reglas correspondientes al resto de los objetos que forman parte del esquema conceptual.

Tabla **Entrada**: se forma a partir del tipo de entidad ENTRADA incorporando todos los atributos de la misma.

Entrada(nombre).

Tabla **Fechas**: se forma a partir del tipo de entidad FECHAS incorporando todos los atributos de la misma, además incorpora el atributo identificador del tipo de entidad ENTRADA con el cual participa en un tipo de interrelación débil por identificación. Según la regla **rtecar-3.1** aplicada a la interrelación ENTRADA/FECHAS el atributo nombre de la entidad ENTRADA actúa como clave foránea manteniendo de esta forma una referencia con la tabla entrada que participa con cardinalidad máxima uno.

Fechas(**nombre**, tipo_fecha, fecha).

Tabla **Direcciones**: se forma a partir del tipo de entidad DIRECCIONES incorporando todos los atributos de la misma, además incorpora el atributo identificador del tipo de entidad ENTRADA con el cual participa en un tipo de interrelación débil por identificación. Los atributos calle, número, C.P, localidad se obtienen por aplicación de la regla **PRETECAR-2** en el que se ha descompuesto el atributo dirección. Según la regla **RETECAR-3.1** aplicada a la interrelación ENTRADA/DIRECCIONES el atributo nombre de la entidad ENTRADA actúa como clave foránea manteniendo de esta forma una referencia con la tabla entrada que participa con cardinalidad máxima uno.

Direcciones(nombre, tipo_direccion, calle, número, C.P, localidad).

Tabla **Movil**: se forma a partir del tipo de entidad MOVIL incorporando todos los atributos del mismo. Esta entidad se genera del proceso de aplicación de la regla **Pretecar-3** recibiendo del supertipo FORMA_CONTACTO sus atributos.

Movil(alfanumérico, tipo).

Tabla **E-Mail**: se forma a partir del tipo de entidad E-MAIL incorporando todos los atributos del la mismo. Esta entidad se genera del proceso de aplicación de la regla **PRETECAR-3** recibiendo del supertipo FORMA_CONTACTO sus atributos.

E-MAIL(alfanumérico).

Tabla **Fijo**: se forma a partir del tipo de entidad FIJO incorporando todos los atributos del la mismo. Esta entidad se genera del proceso de aplicación de la regla **PRETECAR-3** recibiendo del supertipo FORMA_CONTACTO sus atributos.

FIJO(alfanumérico, tipo).

Tabla **Persona-Física**: se forma a partir de la entidad del mismo nombre incorporando todos los atributos de la misma, además incorpora el atributo identificador del tipo de entidad ENTRADA con el cual participa en un tipo de interrelación débil por identificación como consecuencia del proceso de aplicación de la regla **PRETECAR-5**. Además la entidad PERSONA FISICA mantiene una interrelación con ENTRADA, en principio la cardinalidades se ajustan para aplicar la **RTECAR-2.2**. Esta a su vez nos remite a usar la **RTECAR-2.1** ya que los identificadores son los mismos para ambas entidades. A efectos de procesamiento nos interesa mantener dos tablas independientes (debido a que existe a su vez una interrelación entre PERSONA FISICA y PERSONA JURIDICA).

PERSONA_FISICA(nombre, dni).

Tabla **Persona-Jurídica**: se forma a partir del tipo de entidad del mismo nombre incorporando todos los atributos de la misma, además incorpora el atributo identificador del tipo de entidad ENTRADA con el cual participa en un tipo de interrelación débil por identificación como consecuencia del proceso de aplicación de la regla **PRETECAR-5**. Además la entidad PERSONA JURIDICA mantiene una interrelación con ENTRADA, en principio la cardinalidades se ajustan para aplicar la **RTECAR-2.2**. Esta a su vez nos remite a usar la **RTECAR-2.1** ya que los identificadores son los mismos para ambas entidades. A efectos de procesamiento nos interesa mantener dos tablas independientes (debido a que existe a su vez una interrelación entre PERSONA FISICA y PERSONA JURIDICA).

PERSONA_JURIDICA(nombre, CIF).

Tabla **Apuntes**: se forma a partir del tipo de entidad del mismo nombre incorporando todos los atributos de la misma. Esta entidad se genera al aplicar la regla **PRETECAR-1**, además incorpora el atributo identificador del tipo de entidad ENTRADA con el cual participa en un tipo de interrelación débil por identificación.

APUNTES(nombre, apuntes).

Tabla **Entrada-Movil**: esta tabla se genera al aplicar la regla **RTECAR-4** al tipo de interrelación E-M, su esquema será el siguiente:

ENTRADA-MOVIL(nombre, alfanumérico).

Tabla **Entrada_Fijo**: esta tabla se genera al aplicar la regla **RTECAR-4** al tipo de interrelación E-F, su esquema será el siguiente:

ENTRADA-FIJO(nombre, alfanumérico).

Tabla **Entrada-E-Mail**: esta tabla se genera al aplicar la regla **RTECAR-4** al tipo de interrelación E-EM, su esquema será el siguiente:

ENTRADA-E-MAIL(nombre, alfanumérico).

Tabla **Direcciones-Fijo**: esta tabla se genera a partir de la aplicación de la regla **RTECAR-3.2** la cual se forma a partir de las claves identificativas principales de ambas entidades que serán clave foránea y la clave principal de la entidad FIJO será la clave principal de la nueva tabla generada.

DIRECCIONES-FIJO(alfanumérico, nombre, tipo_dirección)

Tabla **Entrada-Apuntos**: esta tabla se genera al aplicar la regla **RTECAR-4** al tipo de interrelación E-A, su esquema será el siguiente:

ENTRADA-APUNTES(nombre,apuntos).

Tabla **Persona-física-Persona-Jurídica**: esta tabla se genera al aplicar la regla **RTECAR-4** al tipo de interrelación E-A, su esquema será el siguiente:

PERSONA FISICA-PERSONA JURIDICA(nombre,nombre).

Una vez que se ha llevado a cabo el proceso anterior obtenemos el siguiente resultado:

Esquema Relacional: Agenda Personal.

Entrada(nombre).

Fechas(nombre, tipo_fecha, fecha).

Direcciones(nombre, tipo_direccion, calle, número, C.P, localidad).

Movil(alfanumérico, tipo).

E-mail(alfanumérico).

Fijo(alfanumérico, tipo).

Persona_Fisica(nombre, dni).

Persona_Juridica(nombre, CIF).

Apuntos(nombre, apuntos).

Entrada-Movil(nombre, alfanumérico).

Entrada-Fijo(nombre, alfanumérico).

Entrada-E-Mail(nombre, alfanumérico).

Direcciones-Fijo(alfanumérico, nombre, tipo_dirección)

Entrada-Apuntos(nombre,apuntos).

Persona Fisica-Persona Juridica(nombre,nombre).

7.6 NORMALIZACIÓN DEL MODELO.

En la tabla Direcciones se puede apreciar una dependencia funcional transitiva entre atributos no primos de la relación :

Direcciones.(nombre) Direcciones.(CP)

Direcciones.(nombre) Direcciones.(localidad)

Direcciones.(CP) Direcciones.(localidad)

Como consecuencia no se cumple la tercera forma normal (*FN3*) y por lo tanto descompondremos la relación en dos nuevas relaciones eliminando de esta manera la dependencia funcional transitiva que existía:

Estas relaciones quedarán de la forma:

Direcciones(nombre,tipo_dirección,calle,número,**CP**)

Localidad(CP,localidad)

De esta manera todas las relaciones estarán en FNBC.

Entrada

Fechas

Direcciones

E-Mail

Fijo

Movil

PersonaFisica

PersonaJuridica

Apuntes

EntradaMovil

EntradaFijo

Entradae-mail

DireccionesFijo

EntradaApuntes

PersonaFisicaPersonaJuridica

Localidad

7.7 SENTENCIAS DE DEFINICION Y MANIPULACION DE LA BASE DE DATOS.

7.7.1 SENTENCIAS DE DEFINICION DE LA BASE DE DATOS.

7.7.1.1 SENTENCIAS DE DEFINICIÓN SINTÁCTICA DE LAS TABLAS DE LA BASE DE DATOS.

El conjunto de sentencias que a continuación se muestran definirán o podemos decir que crearán la estructura de tablas que definirán a la base de datos de nuestro sistema.

CREATE TABLE entrada

(

nombre VARCHAR2(30) NOT NULL,

tipo VARCHAR2(10) NOT NULL,

CONSTRAINT pk_ent

PRIMARY KEY (nombre),

CONSTRAINT up_ent

check(tipo = UPPER(tipo),

CONSTRAINT ck_ent

check (tipo IN ('FISICO','JURIDICO'))

);

CREATE TABLE fechas(

nombre VARCHAR2(30) NOT NULL,

tipo_fecha VARCHAR2(20) NOT NULL,

fecha DATE NOT NULL,

CONSTRAINT pk_fech

PRIMARY KEY (nombre,tipo_fecha),

CONSTRAINT fk_fech

FOREIGN KEY (nombre)

REFERENCES entrada ON DELETE CASCADE,

CONSTRAINT up_fech

```

check(tipo_fecha = UPPER(tipo_fecha))

);

CREATE TABLE localidad(

C_P INTEGER NOT NULL,

localidad VARCHAR2(30) NULL,

CONSTRAINT pk_loc

PRIMARY KEY (C_P),

CONSTRAINT up_loc

check(localidad = UPPER(localidad))

);

CREATE TABLE direcciones(

nombre VARCHAR2(30) NOT NULL,

tipo_direccion VARCHAR2(20) NOT NULL,

calle VARCHAR2(20) NULL,

numero INTEGER NULL,

C_P INTEGER NULL,

CONSTRAINT pk_dir

PRIMARY KEY (nombre,tipo_direccion),

CONSTRAINT fk_direccion

FOREIGN KEY(nombre)

REFERENCES entrada(nombre) ON DELETE CASCADE,

CONSTRAINT fk_cpostal

FOREIGN KEY (C_P)

REFERENCES localidad(C_P));

CREATE TABLE movil(

alfanumerico VARCHAR2(30) NOT NULL,

```

```

tipo VARCHAR2(10) NULL,

CONSTRAINT pk_mov

PRIMARY KEY (alfanumerico),

CONSTRAINT ck_mov

check (tipo IN ('telefono','fax'))

);

CREATE TABLE fijo(

alfanumerico VARCHAR2(30) NOT NULL,

tipo VARCHAR2(10) NOT NULL,

CONSTRAINT pk_fij

PRIMARY KEY (alfanumerico),

CONSTRAINT ck_fijo

check (tipo IN ('telefono','fax'))

);

CREATE TABLE email(

alfanumerico VARCHAR2(30) NOT NULL,

CONSTRAINT pk_ema

PRIMARY KEY (alfanumerico));

CREATE TABLE personafisica(

nombre VARCHAR2(30) NOT NULL,

dni INTEGER NULL,

CONSTRAINT pk_pf

PRIMARY KEY (nombre),

CONSTRAINT fk_pf

FOREIGN KEY(nombre)

REFERENCES entrada ON DELETE CASCADE

```

);

```
CREATE TABLE personajuridica(  
    nombre VARCHAR2(30) NOT NULL,  
    cif VARCHAR2(20) NULL,  
    CONSTRAINT pk_pj  
    PRIMARY KEY (nombre),  
    CONSTRAINT fk_pj  
    FOREIGN KEY(nombre)  
    REFERENCES entrada ON DELETE CASCADE  
);
```

```
CREATE TABLE apuntes(  
    nombre VARCHAR2(30) NOT NULL,  
    apuntes VARCHAR2(40) NOT NULL,  
    CONSTRAINT pk_apu  
    PRIMARY KEY (nombre,apuntes),  
    CONSTRAINT fk_apu  
    FOREIGN KEY(nombre)  
    REFERENCES entrada ON DELETE CASCADE  
);
```

```
CREATE TABLE entradamovil(  
    nombre VARCHAR2(30) NOT NULL,  
    alfanumerico VARCHAR2(30) NOT NULL,  
    CONSTRAINT pk_emo  
    PRIMARY KEY (nombre,alfanumerico),  
    CONSTRAINT fk_emo_en  
    FOREIGN KEY(nombre)
```

```

REFERENCES entrada ON DELETE CASCADE,

CONSTRAINT fk_emo_mov

FOREIGN KEY(alfanumerico)

REFERENCES movil ON DELETE CASCADE

);

CREATE TABLE entradafijo(

nombre VARCHAR2(30) NOT NULL,

alfanumerico VARCHAR2(30) NOT NULL,

CONSTRAINT pk_efi

PRIMARY KEY (nombre,alfanumerico),

CONSTRAINT fk_efi_en

FOREIGN KEY(nombre)

REFERENCES entrada ON DELETE CASCADE,

CONSTRAINT fk_efi_fi

FOREIGN KEY(alfanumerico)

REFERENCES fijo ON DELETE CASCADE

);

CREATE TABLE entradaemail(

nombre VARCHAR2(30) NOT NULL,

alfanumerico VARCHAR2(30) NOT NULL,

CONSTRAINT pk_eema

PRIMARY KEY (nombre,alfanumerico),

CONSTRAINT fk_eema_en

FOREIGN KEY(nombre)

REFERENCES entrada ON DELETE CASCADE,

CONSTRAINT fk_eema_ema

```

```

FOREIGN KEY(alfanumerico)
REFERENCES email ON DELETE CASCADE
);

CREATE TABLE direccionesfijo(
alfanumerico VARCHAR2(30) NOT NULL,
nombre VARCHAR2(30) NOT NULL,
tipo_direccion VARCHAR2(20) NOT NULL,
CONSTRAINT pk_df
PRIMARY KEY (alfanumerico,nombre),
CONSTRAINT fk_df_fij
FOREIGN KEY(alfanumerico)
REFERENCES fijo ON DELETE CASCADE,
CONSTRAINT fk_df_dir
FOREIGN KEY(nombre,tipo_direccion)
REFERENCES direcciones ON DELETE CASCADE
);

CREATE TABLE entradaapuntes(
nombre VARCHAR2(30) NOT NULL,
apuntes VARCHAR2(40) NOT NULL,
CONSTRAINT pk_ea
PRIMARY KEY (nombre,apuntes),
CONSTRAINT fk_ea_en
FOREIGN KEY(nombre)
REFERENCES entrada ON DELETE CASCADE,
CONSTRAINT fk_ea_ap
FOREIGN KEY(apuntes,nombre)

```

REFERENCES apuntes(apuntes,nombre) ON DELETE CASCADE

);

CREATE TABLE personafisicapersonajuridica(

nombre VARCHAR2(30) NOT NULL,

nombre2 VARCHAR2(30) NOT NULL,

CONSTRAINT pk_pfpj

PRIMARY KEY (nombre,nombre2),

CONSTRAINT fk_pfpj_pf

FOREIGN KEY(nombre)

REFERENCES personafisica ON DELETE CASCADE,

CONSTRAINT fk_pfpj_pj

FOREIGN KEY(nombre2)

REFERENCES personajuridica(nombre) ON DELETE CASCADE

);

7.7.1.2 SENTENCIAS QUE PERMITIRÁN UN BORRADO DE TODAS LAS TABLAS DE LA BASE DE DATOS.

Las sentencias que vienen listadas a continuación realizarán el borrado de todas las tablas que fueron definidas en el apartado anterior. Si este conjunto de sentencias se ejecutan correctamente es porque se han borrado correctamente todas las tablas.

DROP TABLE fechas;

DROP TABLE direccionesfijo;

DROP TABLE direcciones;

DROP TABLE localidad;

DROP TABLE entradamovil;

DROP TABLE entradafijo;

DROP TABLE entradaemail;

DROP TABLE movil;

DROP TABLE email;

DROP TABLE ijo;

DROP TABLE entradaapuntes;

DROP TABLE personafisicapersonajuridica;

DROP TABLE personafisica;

DROP TABLE personajuridica;

DROP TABLE apuntes;

DROP TABLE entrada;

7.7.2 SENTENCIAS QUE PERMITIRÁN UNA ALTERACIÓN DE LA ESTRUCTURA DE LAS TABLAS Y SUS RESTRICCIONES.

Modificar el constraint ck_mov de la tabla movil y ck_fijo de la tabla fijo

para que además contemple los valores TELEFONO FAX

ALTER TABLE movil

DROP CONSTRAINT ck_mov;

ALTER TABLE movil

ADD CONSTRAINT ck_mov

check (tipo IN ('TELEFONO','FAX','telefono','fax'));

Como consecuencia de la modificación del constraint de la tabla móvil del ejemplo anterior también modificaremos de la misma manera la tabla fijo.

ALTER TABLE fijo

DROP CONSTRAINT ck_fijo;

ALTER TABLE fijo

ADD CONSTRAINT ck_fijo

check (tipo IN ('TELEFONO','FAX','telefono','fax'));

En la tabla direcciones queremos tener también el atributo "bloque", para localizarlas exactamente una dirección junto con los atributos ya existentes calle, numero, localidad y código postal

ALTER TABLE direcciones

ADD (bloque VARCHAR2(4) null);

En la tabla apuntes queremos tener información también acerca de la fecha en que anotamos nuestra

relación con la persona o compañía en cuestión.

ALTER TABLE apuntes

ADD (fecha_apuntes DATE null);

En la tabla e-mail queremos añadir un nuevo dominio que nos proporciona información acerca de si el e-mail es propio del domicilio o del lugar de trabajo (facultad, empresa ...), pudiendo no aparecer esta información.

ALTER TABLE email

ADD (localizacion VARCHAR2(30) null);

7.7.4 SENTENCIAS PARA LA MANIPULACIÓN DE LA BASE DE DATOS.

7.7.4.1 SENTENCIAS SELECT.

Deseamos obtener las personas que solo tienen direccion en Córdoba única y exclusivamente y mostrar su teléfono y dirección.

SELECT nombre, alfanumerico from entradafijo,

WHERE nombre NOT IN

(SELECT nombre from direcciones

WHERE C_P IN

(select C_P from localidad

where localidad NOT LIKE 'CORDOBA'

))

ORDER BY nombre;

Deseamos obtener los nombres de las personas cuyo cumpleaños sea este mes o en el siguiente y cualquier forma de contacto para felicitarlo (dos formas)

select fechas.nombre, fecha, alfanumerico

from fechas, entradafijo

where entradafijo.nombre = fechas.nombre

and

tipo_fecha = 'CUMPLEAÑOS'

AND

```
TO_NUMBER(SUBSTR(TO_CHAR(FECHA),4,2))  
  
BETWEEN  
  
TO_NUMBER(SUBSTR(TO_CHAR(SYSDATE),4,2))  
  
AND  
  
TO_NUMBER(SUBSTR(TO_CHAR(SYSDATE),4,2))+2  
  
;
```

-- segunda forma de hacerlo

```
select fechas.nombre,fecha,alfanumerico  
  
from fechas,fijo  
  
where tipo_fecha = 'CUMPLEAÑOS'
```

AND

```
TO_NUMBER(SUBSTR(TO_CHAR(FECHA),4,2))  
  
BETWEEN  
  
TO_NUMBER(SUBSTR(TO_CHAR(SYSDATE),4,2))  
  
AND  
  
TO_NUMBER(SUBSTR(TO_CHAR(SYSDATE),4,2))+2  
  
AND
```

```
fijo.alfanumerico IN
```

```
(select alfanumerico from entradafijo  
  
where fechas.nombre = entradafijo.nombre);
```

DESEAMOS obtener las personas que no sean jurídicas, que tengan alguna dirección en Córdoba y que tengan mas de un móvil.

```
select nombre,COUNT(nombre) from entradamovil  
  
where nombre NOT IN  
  
(select nombre from entrada  
  
where tipo = 'JURIDICO')  
  
AND
```

nombre **IN**

(**select** nombre **from** direcciones

where C_P **in**

(**select** C_P **from** localidad

where localidad = 'CORDOBA'))

group by nombre

having COUNT(nombre) > 1;

Deseamos obtener aquellas personas que hayan nacido antes del 1975 y sea amigo de SAUL e indicar si es una persona física o jurídica

select entrada.nombre,tipo **from** entrada

where entrada.nombre **IN**

(**select** nombre **from** fechas

where tipo_fecha = 'CUMPLEAÑOS'

AND TO_NUMBER(SUBSTR(TO_CHAR(fecha),7,2)) < 76)

and nombre **in**(

select apuntes.nombre **from** apuntes

where entrada.nombre = apuntes.nombre

and apuntes = 'AMIGO DE SAUL');

Para cada tipo de fecha mostraremos cuantas personas tienen ese tipo de fecha cumpliendo la condicion de que residan en Cordoba, tengan un telefono y sean personas fisicas

select tipo_fecha,count(tipo_fecha) **from** entrada,fechas

where tipo='FISICO'

AND

entrada.nombre **in**

(**select** nombre **from** direcciones

where C_P **IN**

(**select** C_P **from** localidad

where localidad='CORDOBA')

AND

direcciones.nombre **in**

(**select** nombre **from** entradafijo))

group by tipo_fecha;

7.7.4.2 SENTENCIAS INSERT.

Inserción de datos en la tabla entrada. La inserción de datos en esta tabla es fundamental para el funcionamiento del sistema.

INSERT INTO entrada (nombre,tipo)

VALUES('SAUL CEJUDO CORDOBA','FISICO');

INSERT INTO entrada (nombre,tipo)

VALUES('PEDRO LOPEZ','JURIDICO');

INSERT INTO entrada (nombre,tipo)

VALUES('SEVILLANA S.A.','JURIDICO');

INSERT INTO entrada (nombre,tipo)

VALUES('EMACSA S.L.','JURIDICO');

INSERT INTO entrada (nombre,tipo)

VALUES('TALLERES FIGO S.L.','JURIDICO');

INSERT INTO entrada (nombre,tipo)

VALUES('JUANITO VALDERAMA','FISICO');

INSERT INTO entrada (nombre,tipo)

VALUES('MIGUEL LEIVA SALAS','FISICO');

INSERT INTO entrada (nombre,tipo)

VALUES('MADONNA','FISICO');

INSERT INTO entrada (nombre,tipo)

VALUES('REPSOL S.A.','JURIDICO');

INSERT INTO entrada (nombre,tipo)

VALUES('MARTA SANCHEZ','FISICO');

INSERT INTO entrada (nombre,tipo)

VALUES('CAJASUR','JURIDICO');

INSERT INTO entrada (nombre,tipo)

VALUES('PEPE','JURIDICO');

INSERT INTO entrada (nombre,tipo)

VALUES('JUAN LUCIA','JURIDICO');

INSERT INTO entrada (nombre,tipo)

VALUES('JUAN GOMEZ','FISICO');

Inserción de datos en la tabla apuntes.

INSERT INTO apuntes (nombre,apuntes)

VALUES('SAUL CEJUDO CORDOBA','COMPAÑERO DE CLASE');

INSERT INTO apuntes (nombre,apuntes)

VALUES('MIGUEL LEIVA SALAS','COMPAÑERO DE CLASE');

INSERT INTO apuntes (nombre,apuntes)

VALUES('PEDRO LOPEZ','ABOGADO');

INSERT INTO apuntes (nombre,apuntes)

VALUES('JUANITO VALDERAMA','AMIGO DE SAUL');

INSERT INTO apuntes (nombre,apuntes)

VALUES('TALLERES FIGO S.L.','ASISTENCIA TECNICA DE MI MOTO');

INSERT INTO apuntes (nombre,apuntes)

VALUES('SEVILLANA S.A.','COMPAÑIA DE ELECTRICIDAD');

INSERT INTO apuntes (nombre,apuntes)

VALUES('EMACSA S.L.','COMPAÑIA DE AGUA');

INSERT INTO apuntes (nombre,apuntes)

VALUES('REPSOL S.A.','COMPAÑIA PETROLIFERA');

```
INSERT INTO apuntes (nombre,apuntes)
VALUES('MIGUEL LEIVA SALAS','ENEMIGO');

INSERT INTO apuntes (nombre,apuntes)
VALUES('PEDRO LOPEZ','ENEMIGO');
```

Inserción de datos en la tabla localidad.

```
INSERT INTO localidad (C_P,localidad)
VALUES('14001','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14002','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14003','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14004','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14005','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14006','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14007','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14008','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14009','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14114','GUADALCAZAR');

INSERT INTO localidad (C_P,localidad)
```

```

VALUES('14100','LA CARLOTA');

INSERT INTO localidad (C_P,localidad)

VALUES('12110','ALCORA');

INSERT INTO localidad (C_P,localidad)

VALUES('23004','MADRID');

INSERT INTO localidad (C_P,localidad)

VALUES('12001','CASTELLON DE LA PLANA');

INSERT INTO localidad (C_P,localidad)

VALUES('11002','CADIZ');

INSERT INTO localidad (C_P,localidad)

VALUES('16004','CORUÑA');

INSERT INTO localidad (C_P,localidad)

VALUES('22003','GRANADA');

INSERT INTO localidad (C_P,localidad)

VALUES('34009','SALAMANCA');

```

Inserción de datos en la tabla direcciones .

```

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('SAUL CEJUDO CORDOBA','CASA FAMILIAR','STO DOMINGO',3,14003);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('PEDRO LOPEZ','CASA FAMILIAR','PINTOR ESPINOSA',9,14003);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('SEVILLANA S.A.','SEDE PRINCIPAL','CRUZ CONDE',29,14001);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('EMACSA S.L.','SEDE PRINCIPAL','NOGAL',9,14004);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('TALLERES FIGO S.L.','NEGOCIO','CARLOS III',67,14007);

```

```

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('JUANITO VALDERAMA', 'CASA FAMILIAR', 'JUAN BLAS', 3, 11002);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('MIGUEL LEIVA SALAS', 'CASA FAMILIAR', 'PTDA PEDRIZA', 34, 12110);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('MADONNA', 'CASA PLAYA', 'PINTOR MONROY', 123, 22003);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('REPSOL S.A.', 'SEDE PRINCIPAL', 'CLARA DEL REY', 10, 23004);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('MARTA SANCHEZ', 'CASA FAMILIAR', 'CORAZON DE MARIA', 9, 23004);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('PEDRO LOPEZ', 'PISO ESTUDIANTES', 'VIRGEN DOLORES', 45, 14004);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('SAUL CEJUDO CORDOBA', 'PISO ESTUDIANTES', 'VIRGEN DOLORES', 45, 14004);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('SEVILLANA S.A.', 'SUCURSAL', 'MARIANO RAJOY', 6, 14008);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('MIGUEL LEIVA SALAS', 'PISO ESTUDIANTES', 'MARIANO FORTUNY', 21, 14004);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('CAJASUR', 'SEDE PRINCIPAL', 'CRUZ CONDE', 28, 14002);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('CAJASUR', 'SUCURSAL', 'STO DOMINGO', 3, 14003);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('PEDRO LOPEZ', 'CASA PLAYA', 'LAS PALMERAS', 22, 22003);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('SAUL CEJUDO CORDOBA', 'CASA PLAYA', 'RODRIGO DIAZ', 85, 16004);

```

```
INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)
VALUES('MIGUEL LEIVA SALAS','CASA SIERRA','PEDRO RUIZ',104,12110);
```

Inserción de datos en la tabla direccionesfijo .

```
INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('957280155','SAUL CEJUDO CORDOBA','CASA FAMILIAR');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('957456789','PEDRO LOPEZ','CASA FAMILIAR');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('957234567','SEVILLANA S.A.','SEDE PRINCIPAL');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('957328976','EMACSA S.L.','SEDE PRINCIPAL');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('957402098','TALLERES FIGO S.L.','NEGOCIO');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('964386998','MIGUEL LEIVA SALAS','CASA FAMILIAR');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('957488237','CAJASUR','SUCURSAL');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('955299287','JUANITO VALDERAMA','CASA FAMILIAR');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('957345671','SAUL CEJUDO CORDOBA','PISO ESTUDIANTES');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('914132939','MARTA SANCHEZ','CASA FAMILIAR');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('957345671','MIGUEL LEIVA SALAS','PISO ESTUDIANTES');
```

Inserción de datos en la tabla email.

INSERT INTO email (alfanumerico)

VALUES('i52lesam@uco.es');

INSERT INTO email (alfanumerico)

VALUES('i52cecos@uco.es');

INSERT INTO email (alfanumerico)

VALUES('peloj@hotmail.com');

INSERT INTO email (alfanumerico)

VALUES('lnavarro@ibm.com');

INSERT INTO email (alfanumerico)

VALUES('gccrab5@.repsol.com');

Inserción de datos en la tabla entradaapuntes.

INSERT INTO entradaapuntes (nombre,apuntes)

VALUES('SAUL CEJUDO CORDOBA','COMPAÑERO DE CLASE');

INSERT INTO entradaapuntes (nombre,apuntes)

VALUES('MIGUEL LEIVA SALAS','COMPAÑERO DE CLASE');

INSERT INTO entradaapuntes (nombre,apuntes)

VALUES('PEDRO LOPEZ','ABOGADO');

INSERT INTO entradaapuntes (nombre,apuntes)

VALUES('JUANITO VALDERAMA','AMIGO DE SAUL');

INSERT INTO entradaapuntes (nombre,apuntes)

VALUES('TALLERES FIGO S.L.','ASISTENCIA TECNICA DE MI MOTO');

INSERT INTO entradaapuntes (nombre,apuntes)

VALUES('SEVILLANA S.A.','COMPAÑIA DE ELECTICIDAD');

INSERT INTO entradaapuntes (nombre,apuntes)

VALUES('EMACSA S.L.','COMPAÑIA DE AGUA');

INSERT INTO entradaapuntes (nombre,apuntes)

```
VALUES('REPSOL S.A.','COMPAÑIA PETROLIFERA');
```

```
INSERT INTO entradaapuntes (nombre,apuntes)
```

```
VALUES('MIGUEL LEIVA SALAS','ENEMIGO');
```

```
INSERT INTO entradaapuntes (nombre,apuntes)
```

```
VALUES('PEDRO LOPEZ','ENEMIGO');
```

Inserción de datos en la tabla entradaemail.

```
INSERT INTO entradaemail (alfanumerico,nombre)
```

```
VALUES('i52lesam@uco.es','MIGUEL LEIVA SALAS');
```

```
INSERT INTO entradaemail (alfanumerico,nombre)
```

```
VALUES('i52cecos@uco.es','SAUL CEJUDO CORDOBA');
```

```
INSERT INTO entradaemail (alfanumerico,nombre)
```

```
VALUES('peloj@hotmail.com','PEDRO LOPEZ');
```

```
INSERT INTO entradaemail (alfanumerico,nombre)
```

```
VALUES('lnavarro@ibm.com','SEVILLANA S.A.');
```

```
INSERT INTO entradaemail (alfanumerico,nombre)
```

```
VALUES('gccrab5@.repsol.com','REPSOL S.A.');
```

Inserción de datos en la tabla entradafijo.

```
INSERT INTO entradafijo (alfanumerico,nombre)
```

```
VALUES('957280155','SAUL CEJUDO CORDOBA');
```

```
INSERT INTO entradafijo (alfanumerico,nombre)
```

```
VALUES('957456789','PEDRO LOPEZ');
```

```
INSERT INTO entradafijo (alfanumerico,nombre)
```

```
VALUES('957234567','SEVILLANA S.A.');
```

```
INSERT INTO entradafijo (alfanumerico,nombre)
```

```
VALUES('957328976','EMACSA S.L.');
```

```
INSERT INTO entradafijo (alfanumerico,nombre)
```

```

VALUES('964386998','MIGUEL LEIVA SALAS');

INSERT INTO entradafijo (alfanumerico,nombre)

VALUES('914132939','MARTA SANCHEZ');

INSERT INTO entradafijo (alfanumerico,nombre)

VALUES('957402098','TALLERES FIGO S.L.');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)

VALUES('957488237','CAJASUR');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)

VALUES('957474063','EMACSA S.L.');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)

VALUES('955299287','JUANITO VALDERAMA');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)

VALUES('957345671','SAUL CEJUDO CORDOBA');
```

Inserción de datos en la tabla entramovil.

```

INSERT INTO entramovil (nombre,alfanumerico)

VALUES('MIGUEL LEIVA SALAS','609453832');
```

```

INSERT INTO entramovil (nombre,alfanumerico)

VALUES('SAUL CEJUDO CORDOBA','639895433');
```

```

INSERT INTO entramovil (nombre,alfanumerico)

VALUES('MIGUEL LEIVA SALAS','630887654');
```

```

INSERT INTO entramovil (nombre,alfanumerico)

VALUES('SEVILLANA S.A.','630993625');
```

```

INSERT INTO entramovil (nombre,alfanumerico)

VALUES('PEDRO LOPEZ','639454533');
```

```

INSERT INTO entramovil (nombre,alfanumerico)

VALUES('MADONNA','630558987');
```

```

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('JUANITO VALDERAMA','639458796');

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('TALLERES FIGO S.L.','607565654');

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('SAUL CEJUDO CORDOBA','609541244');

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('MIGUEL LEIVA SALAS','630591204');

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('SAUL CEJUDO CORDOBA','630594455');

```

Inserción de datos en la tabla Fechas.

```

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
VALUES('SAUL CEJUDO CORDOBA','ANIVERSARIO DE
BODA',TO_DATE('05-05-1999','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
VALUES('PEDRO LOPEZ','CUMPLEAÑOS',TO_DATE('07-23-1997','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
VALUES('PEDRO LOPEZ','SANTO',TO_DATE('7-23','MM-DD'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
VALUES('SEVILLANA S.A.','FECHA DE ALTA',TO_DATE('03-04-1995','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
VALUES('EMACSA S.L.','FECHA DE ALTA',TO_DATE('11-29-1995','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
VALUES('JUANITO VALDERAMA','CUMPLEAÑOS',TO_DATE('12-24-1940','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
VALUES('MIGUEL LEIVA SALAS','CUMPLEAÑOS',TO_DATE('12-08-1975','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)

```

```
VALUES('SAUL CEJUDO CORDOBA','CUMPLEAÑOS',TO_DATE('05-29-1977','MM-DD-YYYY'));
```

```
INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
```

```
VALUES('MARTA SANCHEZ','CUMPLEAÑOS',TO_DATE('06-09-1969','MM-DD-YYYY'));
```

```
INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
```

```
VALUES('JUANITO VALDERAMA','ANIVERSARIO DE  
BODA',TO_DATE('10-25-1959','MM-DD-YYYY'));
```

```
INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
```

```
VALUES('MIGUEL LEIVA SALAS','SANTO',TO_DATE('10-29','MM-DD'));
```

```
INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
```

```
VALUES('SAUL CEJUDO CORDOBA','SANTO',TO_DATE('09-24','MM-DD'));
```

```
INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
```

```
VALUES('MARTA SANCHEZ','SANTO',TO_DATE('10-01','MM-DD'));
```

Inserción de datos en la tabla Fijo.

```
INSERT INTO fijo (alfanumerico,tipo)
```

```
VALUES('957280155','telefono');
```

```
INSERT INTO fijo (alfanumerico,tipo)
```

```
VALUES('957456789','telefono');
```

```
INSERT INTO fijo (alfanumerico,tipo)
```

```
VALUES('957234567','telefono');
```

```
INSERT INTO fijo (alfanumerico,tipo)
```

```
VALUES('957328976','telefono');
```

```
INSERT INTO fijo (alfanumerico,tipo)
```

```
VALUES('964386998','telefono');
```

```
INSERT INTO fijo (alfanumerico,tipo)
```

```
VALUES('914132939','telefono');
```

```
INSERT INTO fijo (alfanumerico,tipo)
```

```
VALUES('957402098','fax');
```

```
INSERT INTO fijo (alfanumerico,tipo)
```

```
VALUES('957488237','fax');
```

```
INSERT INTO fijo (alfanumerico,tipo)
```

```
VALUES('957474063','telefono');
```

```
INSERT INTO fijo (alfanumerico,tipo)
```

```
VALUES('955299287','telefono');
```

```
INSERT INTO fijo (alfanumerico,tipo)
```

```
VALUES('957345671','telefono');
```

Inserción de datos en la tabla movil.

```
INSERT INTO movil (alfanumerico,tipo)
```

```
VALUES('609453832','telefono');
```

```
INSERT INTO movil (alfanumerico,tipo)
```

```
VALUES('639895433','telefono');
```

```
INSERT INTO movil (alfanumerico,tipo)
```

```
VALUES('630887654','telefono');
```

```
INSERT INTO movil (alfanumerico,tipo)
```

```
VALUES('630993625','telefono');
```

```
INSERT INTO movil (alfanumerico,tipo)
```

```
VALUES('639454533','telefono');
```

```
INSERT INTO movil (alfanumerico,tipo)
```

```
VALUES('630558987','telefono');
```

```
INSERT INTO movil (alfanumerico,tipo)
```

```
VALUES('639458796','telefono');
```

```
INSERT INTO movil (alfanumerico,tipo)
```

```
VALUES('607565654','telefono');
```

```
INSERT INTO movil (alfanumerico,tipo)
```

```
VALUES('609541244','telefono');
```

```
INSERT INTO movil (alfanumerico,tipo)
```

```
VALUES('630591204','telefono');
```

```
INSERT INTO movil (alfanumerico,tipo)
```

```
VALUES('630594455','fax');
```

Inserción de datos en la tabla personafisica.

```
INSERT INTO PERSONAFISICA (nombre,dni)
```

```
VALUES ('SAUL CEJUDO CORDOBA',3094302);
```

```
INSERT INTO PERSONAFISICA (nombre,dni)
```

```
VALUES ('JUANITO VALDERAMA',4567598);
```

```
INSERT INTO PERSONAFISICA (nombre,dni)
```

```
VALUES ('MIGUEL LEIVA SALAS',30304056);
```

```
INSERT INTO PERSONAFISICA (nombre,dni)
```

```
VALUES ('MADONNA',69696969);
```

```
INSERT INTO PERSONAFISICA (nombre,dni)
```

```
VALUES ('MARTA SANCHEZ',86759465);
```

Inserción de datos en la tabla personajuridica.

```
INSERT INTO personajuridica (nombre,cif)
```

```
VALUES('PEDRO LOPEZ',NULL);
```

```
INSERT INTO personajuridica (nombre,cif)
```

```
VALUES('SEVILLANA S.A.',NULL);
```

```
INSERT INTO personajuridica (nombre,cif)
```

```
VALUES('EMACSA S.L.',NULL);
```

```
INSERT INTO personajuridica (nombre,cif)
```

```
VALUES('TALLERES FIGO S.L.',NULL);
```

```
INSERT INTO personajuridica (nombre,cif)
```

```

VALUES('REPSOL S.A.',NULL);

INSERT INTO personajuridica (nombre,cif)

VALUES('CAJASUR',NULL); INSERT INTO personajuridica (nombre,cif)

VALUES('PEDRO LOPEZ',NULL);

INSERT INTO personajuridica (nombre,cif)

VALUES('SEVILLANA S.A.',NULL);

INSERT INTO personajuridica (nombre,cif)

VALUES('EMACSA S.L.',NULL);

INSERT INTO personajuridica (nombre,cif)

VALUES('TALLERES FIGO S.L.',NULL);

INSERT INTO personajuridica (nombre,cif)

VALUES('REPSOL S.A.',NULL);

INSERT INTO personajuridica (nombre,cif)

VALUES('CAJASUR',NULL);

```

Inserción de datos en la tabla personajuridica.

```

INSERT INTO PERSONAFISICA (nombre,dni)

VALUES ('SAUL CEJUDO CORDOBA',3094302);

INSERT INTO PERSONAFISICA (nombre,dni)

VALUES ('JUANITO VALDERAMA',4567598);

INSERT INTO PERSONAFISICA (nombre,dni)

VALUES ('MIGUEL LEIVA SALAS',30304056);

INSERT INTO PERSONAFISICA (nombre,dni)

VALUES ('MADONNA',69696969);

INSERT INTO PERSONAFISICA (nombre,dni)

VALUES ('MARTA SANCHEZ',86759465);

```

Inserción de datos mediante insert más complejos.

Insertaremos en la tabla personafisica aquellas tuplas de la tabla entrada cuyo tipo sea igual a Físico y no se encuentran ya en la tabla personafisica.

```
insert into personafisica(nombre)
(select nombre from entrada
where tipo = 'FISICO' AND nombre NOT IN
(select nombre from personafisica)
);
```

Insertaremos en la tabla personajuridica aquellas tuplas de la tabla entrada cuyo tipo sea igual a Jurídico y no se encuentran ya en la tabla personajuridica.

```
insert into personajuridica(nombre)
(select nombre from entrada
where tipo = 'JURIDICO' AND nombre NOT IN
(select nombre from personajuridica)
);
```

Insertaremos en la tabla apuntes la anotación de la localidad de aquellas personas cuya localidad sea igual a Madrid.

```
insert into apuntes(nombre,apuntes)
(select nombre, 'ES DE MADRID' from direcciones
where C_P in
(select C_P from localidad
where localidad = 'MADRID')
);
```

Introduciremos en la tabla apuntes la anotación de llamadas caras para aquellas personas que tengan móvil y este empiece por 609.

```
insert into apuntes(nombre,apuntes)
(select nombre,'llamadas caras' from entradamovil
where alfanumerico like '609%');
```

7.7.4.3 SENTENCIAS UPDATE

Actualizar el campo fecha de la tabla fechas para la entrada Saúl Cejudo Córdoba cuanto el tipo de fecha sea igual a Aniversario de boda

```
UPDATE fechas set fecha = to_date('12-25-1999','MM-DD-YYYY')
```

```
where tipo_fecha='ANIVERSARIO DE BODA'
```

```
AND
```

```
nombre='SAUL CEJUDO CORDOBA';
```

Actualizar el dni de la tabla persona física al valor que se indica cuando la entrada sea igual a Saúl Cejudo Córdoba

```
UPDATE personafisica set dni = 45678976
```

```
where nombre='SAUL CEJUDO CORDOBA';
```

```
UPDATE personajuridica set cif = '87654876-E'
```

```
where nombre='REPSOL S.A.';
```

UPDATES COMPLEJOS

Queremos cambiar la dirección del piso de estudiante que SAUL CEJUDO

posee por la dirección del piso de estudiante Miguel Leiva Salas.

```
UPDATE direcciones SET (calle,C_P,numero) =
```

```
(select calle,C_P,numero from direcciones
```

```
WHERE nombre = 'MIGUEL LEIVA SALAS'
```

```
and
```

```
tipo_direccion = 'PISO ESTUDIANTES')
```

```
WHERE nombre = 'SAUL CEJUDO CORDOBA'
```

```
AND tipo_direccion= 'PISO ESTUDIANTES';
```

```
UPDATE direccionesfijo SET alfanumerico =
```

```
(select alfanumerico from direccionesfijo
```

```
WHERE nombre = 'MIGUEL LEIVA SALAS'
```

```
and
```

```
tipo_direccion = 'PISO ESTUDIANTES')
```

WHERE nombre = 'SAUL CEJUDO CORDOBA'

AND tipo_direccion= 'PISO ESTUDIANTES' ;

Deseamos actualizar los teléfonos con una supuesta reforma de telefonica que establece que todos los teléfonos móviles que tengan los primeros tres dígitos iguales a 630 son faxes.

UPDATE MOVIL SET tipo = 'fax'

where alfanumerico IN

(SELECT ALFANUMERICO FROM MOVIL

WHERE alfanumerico like '630%'

and tipo='telefono');

7.7.4.4 SENTENCIAS DELETE

Deseamos borrar la entrada Miguel Leiva Salas, el efecto será el borrado también en cualquier tabla donde se haga referencia a esta tupla.

delete from entrada where ENTRADA.nombre = 'MIGUEL LEIVA SALAS';

Deseamos borrar de la tabla fechas cualquier tupla que haga referencia a un aniversario, por eso se busca en el cuerpo de la sentencia la subcadena Aniversario.

DELETE FROM FECHAS WHERE TIPO_FECHA LIKE 'ANIVERSARIO%';

Borrar aquellos móviles que sean de tipo fax y empiecen por 630.

DELETE FROM MOVIL WHERE TIPO = 'fax' AND alfanumerico LIKE '630%';

Deseamos borrar aquella personas que nacieron antes de 1950.

DELETE FROM entrada

WHERE nombre IN

(SELECT nombre FROM fechas

WHERE TIPO_FECHA = 'CUMPLEAÑOS'

AND

TO_NUMBER(SUBSTR(TO_CHAR(fecha),7,2)) < 50

);

Deseamos borrar de nuestra agenda todas las personas que tengan como apuntes que sean ENEMIGOS o sean algún tipo de COMPAÑÍA

```
DELETE FROM entrada
```

```
WHERE nombre IN
```

```
(SELECT nombre FROM apuntes
```

```
WHERE apuntes = 'ENEMIGO'
```

```
OR
```

```
apuntes LIKE 'COMPAÑIA%'
```

```
);
```

Deseamos borrar todas las direcciones de la tabla direcciones que no sean de CORDOBA

```
DELETE FROM direcciones
```

```
WHERE C_P NOT IN
```

```
(SELECT C_P FROM localidad
```

```
WHERE localidad = 'CORDOBA');
```

0.8 PROCEDIMIENTOS DE PL/SQL.

A continuación se listarán los diferentes procedimientos que se utilizarán para realizar operaciones sobre las diferentes entidades de la Base de datos.

Si una persona no tiene ninguna forma de contacto, consideramos que no nos interesa y lo sacaremos de nuestra agenda.

SOLUCION: Primero contamos todas sus formas de contacto que posee el parametro que recibe el procedimiento y utilizando una estructura de control condicional eliminaremos aquellas entradas que no posean ninguna forma de contacto.

```
CREATE OR REPLACE PROCEDURE checknombres(p_nombre VARCHAR2)
```

```
AS
```

```
num_dir NUMBER;
```

```
num_mov NUMBER;
```

```
num_fijo NUMBER;
```

```
num_email NUMBER;
```

```
no_direccion EXCEPTION;
```

```

si_direccion EXCEPTION;

BEGIN

num_dir:=0;

num_mov:=0;

num_fijo:=0;

num_email:=0;

SELECT COUNT(nombre) INTO num_dir from direcciones

WHERE nombre = p_nombre;

SELECT COUNT(nombre) INTO num_mov from entradamovil

WHERE nombre = p_nombre;

SELECT COUNT(nombre) INTO num_fijo from entradafijo

WHERE nombre = p_nombre;

SELECT COUNT(nombre) INTO num_email from entradaemail

WHERE nombre = p_nombre;

IF num_dir=0 AND num_mov=0 AND num_email=0 AND num_fijo=0 THEN

DBMS_OUTPUT.PUT_LINE('NO ASIGNADO UNA DIRECCION');

delete from entrada where nombre= p_nombre;

RAISE no_direccion;

END IF;

RAISE si_direccion;

EXCEPTION

WHEN no_direccion THEN

DBMS_OUTPUT.PUT_LINE('NO ASIGNADO UNA DIRECCION');

WHEN si_direccion THEN

DBMS_OUTPUT.PUT_LINE('NO ASIGNADO UNA DIRECCION');

END;

```

/

Deseamos crear un procedimiento que dado un nombre lo introduzca en la tabla fechas y le de los valores 'SANTO' para tipo_fecha y la fecha del día actual para fecha. Solo si no existe una tupla para este tipo de fecha.

SOLUCION: Para resolver esta situación usaremos cursores que nos ayudarán al tratamiento de los datos.

```
CREATE OR REPLACE PROCEDURE hoy_santo_de(p_nombre VARCHAR2)
```

```
AS
```

```
CURSOR c_nombres IS
```

```
SELECT nombre,tipo
```

```
FROM entrada
```

```
WHERE substr(nombre,1,length(p_nombre)) like p_nombre
```

```
AND nombre NOT IN
```

```
(SELECT nombre from fechas
```

```
WHERE tipo_fecha = 'SANTO');
```

```
v_nombres c_nombres%ROWTYPE;
```

```
BEGIN
```

```
OPEN c_nombres;
```

```
FETCH c_nombres INTO v_nombres;
```

```
WHILE c_nombres%FOUND LOOP
```

```
INSERT INTO fechas (nombre,tipo_fecha,fecha)
```

```
VALUES(v_nombres.nombre,'SANTO',sysdate);
```

```
FETCH c_nombres INTO v_nombres;
```

```
END LOOP;
```

```
CLOSE c_nombres;
```

```
END hoy_santo_de;
```

/

Ha existido una modificación en los prefijos de los teléfonos fijos en el que el prefijo 957 se ha modificado por el 956. Se ha implementado un procedimiento de forma general que recibe como

parámetro el prefijo antiguo y el nuevo prefijo por el que se sustituirá.(los cambios serán generalizados para todas nuestras relaciones).

SOLUCIÓN: Para resolver este problema hemos utilizado tres cursores y tres tablas. Las tres tablas se irán cargando mediante sucesivas lecturas de los cursores y haciendo el tratamiento de las tablas se insertarán los nuevos valores en la agenda.

```
CREATE OR REPLACE PROCEDURE cambioprefijo(prefijoa VARCHAR2,
prefijon VARCHAR2)
AS
CURSOR c_alfanum IS
SELECT nombre,alfanumerico
FROM entradafijo
WHERE substr(alfanumerico,1,3) like prefijoa;
CURSOR c_alfanum2 IS
SELECT alfanumerico,tipo
FROM fijo
WHERE substr(alfanumerico,1,3) like prefijoa;
CURSOR c_alfanum3 IS
SELECT alfanumerico,tipo_direccion,nombre
FROM direccionesfijo
WHERE substr(alfanumerico,1,3) like prefijoa;
v_nombres c_alfanum%ROWTYPE;
v_alfanumero c_alfanum2%ROWTYPE;
v_direc c_alfanum3%ROWTYPE;
indice NUMBER:=0;
total NUMBER:=0;
total2 NUMBER:=0;
total3 NUMBER:=0;
TYPE tabla IS TABLE OF c_alfanum%ROWTYPE INDEX BY BINARY_INTEGER;
```

```

TYPE tabla2 IS TABLE OF c_alfanum2%ROWTYPE INDEX BY BINARY_INTEGER;

TYPE tabla3 IS TABLE OF c_alfanum3%ROWTYPE INDEX BY BINARY_INTEGER;

v_tabla tabla;

v_tabla2 tabla2;

v_tabla3 tabla3;

BEGIN

OPEN c_alfanum;

OPEN c_alfanum2;

OPEN c_alfanum3;

select count(alfanumerico) into total from entradafijo

where substr(alfanumerico,1,3) like prefijoa;

select count(alfanumerico) into total2 from fijo

where substr(alfanumerico,1,3) like prefijoa;

select count(alfanumerico) into total3 from fijo

where substr(alfanumerico,1,3) like prefijoa;

FOR indice IN 1..total2 LOOP

FETCH c_alfanum2 INTO v_tabla2(indice);

INSERT INTO fijo (alfanumerico,tipo)

VALUES(prefijon || substr(v_tabla2(indice).alfanumerico,4,20),

v_tabla2(indice).tipo);

END LOOP;

FOR indice IN 1..total3 LOOP

FETCH c_alfanum3 INTO v_tabla3(indice);

INSERT INTO direccionesfijo (alfanumerico,nombre,tipo_direccion)

VALUES(prefijon || substr(v_tabla3(indice).alfanumerico,4,20),

v_tabla3(indice).nombre,

```

```

v_tabla3(indice).tipo_direccion);

END LOOP;

FOR indice IN 1..total LOOP

FETCH c_alfanum INTO v_tabla(indice);

INSERT INTO entradafijo (nombre,alfanumerico)

VALUES(v_tabla(indice).nombre,prefijon || substr(v_tabla(indice).alfanumerico,4,20));

END LOOP;

delete FROM entradafijo

WHERE substr(alfanumerico,1,3) like prefijoa;

delete FROM direccionesfijo

WHERE substr(alfanumerico,1,3) like prefijoa;

delete FROM fijo

WHERE substr(alfanumerico,1,3) like prefijoa;

CLOSE c_alfanum;

CLOSE c_alfanum2;

CLOSE c_alfanum3;

END cambioprefijo;

/

```

INSTRUCCIONES UTILIZADAS

```

insert into personafisica(nombre)

(select nombre from entrada

where tipo = 'FISICO' AND nombre NOT IN

(select nombre from personafisica)

);

insert into personajuridica(nombre)

(select nombre from entrada

```

```

where tipo = 'JURIDICO' AND nombre NOT IN
(select nombre from personajuridica)
);

insert into apuntes(nombre,apuntes)
(select nombre, 'ES DE MADRID' from direcciones
where C_P in
(select C_P from localidad
where localidad = 'MADRID')
);

insert into apuntes(nombre,apuntes)
(select nombre,'llamadas caras' from entradamovil
where alfanumerico like '609%');

-- Modificar el constraint ck_mov de la tabla movil y ck_fijo de la tabla fijo
-- para que ademas contemple los valores TELEFONO FAX

ALTER TABLE movil
DROP CONSTRAINT ck_mov;

ALTER TABLE movil
ADD CONSTRAINT ck_mov
check (tipo IN ('TELEFONO','FAX','telefono','fax'));

-- Como consecuencia de la modificacion del constraint de la tabla
-- movil del ejemplo anterior tambien modificaremos de la misma ma-
-- nera la tabla fijo.

ALTER TABLE fijo
DROP CONSTRAINT ck_fijo;

ALTER TABLE fijo
ADD CONSTRAINT ck_fijo

```

check (tipo IN ('TELEFONO','FAX','telefono','fax'));

-- En la tabla direcciones queremos tener tambien el atributo "bloque", para localizar

-- mas exactamente una direccion junto con los atributos ya existentes calle,numero,

-- localidad y codigo postal

ALTER TABLE direcciones

ADD (bloque VARCHAR2(4) null);

-- En la tabla apuntes queremos tener informacion tambien acerca de la fecha en que anotamos

-- nuestra relacion con la persona o compa ia en cuestion.

ALTER TABLE apuntes

ADD (fecha_apuntes DATE null);

-- En la tabla e-mail queremos a adir un nuevo dominio que nos proporciona informacion acerca

-- de si el e-mail es propio del domicilio o del lugar de trabajo (facultad,empresa ...), pu-

-- diendo no aparecer esta informacion.

ALTER TABLE email

ADD (localizacion VARCHAR2(30) null);

CREATE TABLE entrada(

nombre VARCHAR2(30) NOT NULL,

tipo VARCHAR2(10) NOT NULL,

CONSTRAINT pk_ent

PRIMARY KEY (nombre),

CONSTRAINT up_ent

check(tipo = UPPER(tipo)),

CONSTRAINT ck_ent

check (tipo IN ('FISICO','JURIDICO'))

);

CREATE TABLE fechas(

```

nombre VARCHAR2(30) NOT NULL,

tipo_fecha VARCHAR2(20) NOT NULL,

fecha DATE NOT NULL,

CONSTRAINT pk_fech

PRIMARY KEY (nombre,tipo_fecha),

CONSTRAINT fk_fech

FOREIGN KEY (nombre)

REFERENCES entrada ON DELETE CASCADE,

CONSTRAINT up_fech

check(tipo_fecha = UPPER(tipo_fecha))

);

CREATE TABLE localidad(

C_P INTEGER NOT NULL,

localidad VARCHAR2(30) NULL,

CONSTRAINT pk_loc

PRIMARY KEY (C_P),

CONSTRAINT up_loc

check(localidad = UPPER(localidad))

);

CREATE TABLE direcciones(

nombre VARCHAR2(30) NOT NULL,

tipo_direccion VARCHAR2(20) NOT NULL,

calle VARCHAR2(20) NULL,

numero INTEGER NULL,

C_P INTEGER NULL,

CONSTRAINT pk_dir

```

```

PRIMARY KEY (nombre,tipo_direccion),

CONSTRAINT fk_direccion

FOREIGN KEY(nombre)

REFERENCES entrada(nombre) ON DELETE CASCADE,

CONSTRAINT fk_cpostal

FOREIGN KEY (C_P)

REFERENCES localidad(C_P));

CREATE TABLE movil(

alfanumerico VARCHAR2(30) NOT NULL,

tipo VARCHAR2(10) NULL,

CONSTRAINT pk_mov

PRIMARY KEY (alfanumerico),

CONSTRAINT ck_mov

check (tipo IN ('telefono','fax'))

);

CREATE TABLE fijo(

alfanumerico VARCHAR2(30) NOT NULL,

tipo VARCHAR2(10) NOT NULL,

CONSTRAINT pk_fij

PRIMARY KEY (alfanumerico),

CONSTRAINT ck_fijo

check (tipo IN ('telefono','fax'))

);

CREATE TABLE email(

alfanumerico VARCHAR2(30) NOT NULL,

CONSTRAINT pk_ema

```

```

PRIMARY KEY (alfanumerico)

);

CREATE TABLE personafisica(

nombre VARCHAR2(30) NOT NULL,

dni INTEGER NULL,

CONSTRAINT pk_pf

PRIMARY KEY (nombre),

CONSTRAINT fk_pf

FOREIGN KEY(nombre)

REFERENCES entrada ON DELETE CASCADE

);

CREATE TABLE personajuridica(

nombre VARCHAR2(30) NOT NULL,

cif VARCHAR2(20) NULL,

CONSTRAINT pk_pj

PRIMARY KEY (nombre),

CONSTRAINT fk_pj

FOREIGN KEY(nombre)

REFERENCES entrada ON DELETE CASCADE

);

CREATE TABLE apuntes(

nombre VARCHAR2(30) NOT NULL,

apuntes VARCHAR2(40) NOT NULL,

CONSTRAINT pk_apu

PRIMARY KEY (nombre,apuntes),

CONSTRAINT fk_apu

```

```

FOREIGN KEY(nombre)

REFERENCES entrada ON DELETE CASCADE

);

CREATE TABLE entradamovil(

nombre VARCHAR2(30) NOT NULL,

alfanumerico VARCHAR2(30) NOT NULL,

CONSTRAINT pk_emo

PRIMARY KEY (nombre,alfanumerico),

CONSTRAINT fk_emo_en

FOREIGN KEY(nombre)

REFERENCES entrada ON DELETE CASCADE,

CONSTRAINT fk_emo_mov

FOREIGN KEY(alfanumerico)

REFERENCES movil ON DELETE CASCADE

);

CREATE TABLE entradafijo(

nombre VARCHAR2(30) NOT NULL,

alfanumerico VARCHAR2(30) NOT NULL,

CONSTRAINT pk_efi

PRIMARY KEY (nombre,alfanumerico),

CONSTRAINT fk_efi_en

FOREIGN KEY(nombre)

REFERENCES entrada ON DELETE CASCADE,

CONSTRAINT fk_efi_fi

FOREIGN KEY(alfanumerico)

REFERENCES fijo ON DELETE CASCADE

```

```

);

CREATE TABLE entradaemail(

nombre VARCHAR2(30) NOT NULL,

alfanumerico VARCHAR2(30) NOT NULL,

CONSTRAINT pk_eema

PRIMARY KEY (nombre,alfanumerico),

CONSTRAINT fk_eema_en

FOREIGN KEY(nombre)

REFERENCES entrada ON DELETE CASCADE,

CONSTRAINT fk_eema_ema

FOREIGN KEY(alfanumerico)

REFERENCES email ON DELETE CASCADE

);

CREATE TABLE direccionesfijo(

alfanumerico VARCHAR2(30) NOT NULL,

nombre VARCHAR2(30) NOT NULL,

tipo_direccion VARCHAR2(20) NOT NULL,

CONSTRAINT pk_df

PRIMARY KEY (alfanumerico,nombre),

CONSTRAINT fk_df_fij

FOREIGN KEY(alfanumerico)

REFERENCES fijo ON DELETE CASCADE,

CONSTRAINT fk_df_dir

FOREIGN KEY(nombre,tipo_direccion)

REFERENCES direcciones ON DELETE CASCADE

);

```

```

CREATE TABLE entradaapuntes(
nombre VARCHAR2(30) NOT NULL,
apuntes VARCHAR2(40) NOT NULL,
CONSTRAINT pk_ea
PRIMARY KEY (nombre,apuntes),
CONSTRAINT fk_ea_en
FOREIGN KEY(nombre)
REFERENCES entrada ON DELETE CASCADE,
CONSTRAINT fk_ea_ap
FOREIGN KEY(apuntes,nombre)
REFERENCES apuntes(apuntes,nombre) ON DELETE CASCADE
);

```

```

CREATE TABLE personafisicapersonajuridica(
nombre VARCHAR2(30) NOT NULL,
nombre2 VARCHAR2(30) NOT NULL,
CONSTRAINT pk_pfpj
PRIMARY KEY (nombre,nombre2),
CONSTRAINT fk_pfpj_pf
FOREIGN KEY(nombre)
REFERENCES personafisica ON DELETE CASCADE,
CONSTRAINT fk_pfpj_pj
FOREIGN KEY(nombre2)
REFERENCES personajuridica(nombre) ON DELETE CASCADE
);

```

--DESEAMOS BORRAR DE LA ENTRADA MIGUEL LEIVA SALAS

--EL EFECTO SERÁ EL DEL BORRADO TAMBIÉN EN CUALQUIER TABLA LAS

-- TUPLAS QUE HAGAN REFERENCIA A ESTE NOMBRE

delete from entrada where ENTRADA.nombre = 'MIGUEL LEIVA SALAS';

-- DESEAMOS BORRAR DE LA TABLA FECHAS AQUELLAS TUPLAS QUE HAGAN REFEREN-
-- CIA A ANIVERSARIOS DE CUALQUIER TIPO, POR ELLO SE BUSCA LA SUB-
-- CADENA ANIVERSARIO.

DELETE FROM FECHAS WHERE TIPO_FECHA LIKE 'ANIVERSARIO%';

-- BORRAR AQUELLOS MOVILES QUE SEAN DE TIPO FAX y empiezen por 630

DELETE FROM MOVIL WHERE TIPO = 'fax' AND alfanumerico LIKE '630%';

-- Deseamos borrar aquella personas que nacieron antes de 1950

DELETE FROM entrada

WHERE nombre IN

(SELECT nombre FROM fechas

WHERE TIPO_FECHA = 'CUMPLEAÑOS'

AND

TO_NUMBER(SUBSTR(TO_CHAR(fecha),7,2)) < 50

);

-- Deseamos borrar de nuestra agenda todas las personas que tengan como

-- apuntes que sean ENEMIGOS o sean algún tipo de COMPAÑIA

DELETE FROM entrada

WHERE nombre IN

(SELECT nombre FROM apuntes

WHERE apuntes = 'ENEMIGO'

OR

apuntes LIKE 'COMPAÑIA%'

);

-- Deseamos borrar todas las direcciones de la tabla direcciones que no

```

-- sean de CORDOBA

DELETE FROM direcciones

WHERE C_P NOT IN

(SELECT C_P FROM localidad

WHERE localidad = 'CORDOBA');

drop table fechas;

drop table direccionesfijo;

drop table direcciones;

drop table localidad;

drop table entradamovil;

drop table entradafijo;

drop table entradaemail;

drop table movil;

drop table email;

drop table fijo;

drop table entradaapuntes;

drop table personafisicapersonajuridica;

drop table personafisica;

drop table personajuridica;

drop table apuntes;

drop table entrada;

-- SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME = entrada ;

-- SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME = "nombre";

-- SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME ="entrada(nombre)";

INSERT INTO apuntes (nombre,apuntes)

VALUES('SAUL CEJUDO CORDOBA','COMPAÑERO DE CLASE');

```

```

INSERT INTO apuntes (nombre,apuntes)

VALUES('MIGUEL LEIVA SALAS','COMPAÑERO DE CLASE');

INSERT INTO apuntes (nombre,apuntes)

VALUES('PEDRO LOPEZ','ABOGADO');

INSERT INTO apuntes (nombre,apuntes)

VALUES('JUANITO VALDERAMA','AMIGO DE SAUL');

INSERT INTO apuntes (nombre,apuntes)

VALUES('TALLERES FIGO S.L.','ASISTENCIA TECNICA DE MI MOTO');

INSERT INTO apuntes (nombre,apuntes)

VALUES('SEVILLANA S.A.','COMPAÑIA DE ELECTICIDAD');

INSERT INTO apuntes (nombre,apuntes)

VALUES('EMACSA S.L.','COMPAÑIA DE AGUA');

INSERT INTO apuntes (nombre,apuntes)

VALUES('REPSOL S.A.','COMPAÑIA PETROLIFERA');

INSERT INTO apuntes (nombre,apuntes)

VALUES('MIGUEL LEIVA SALAS','ENEMIGO');

INSERT INTO apuntes (nombre,apuntes)

VALUES('PEDRO LOPEZ','ENEMIGO');

INSERT INTO localidad (C_P,localidad)

VALUES('14001','CORDOBA');

INSERT INTO localidad (C_P,localidad)

VALUES('14002','CORDOBA');

INSERT INTO localidad (C_P,localidad)

VALUES('14003','CORDOBA');

INSERT INTO localidad (C_P,localidad)

VALUES('14004','CORDOBA');

```

```

INSERT INTO localidad (C_P,localidad)
VALUES('14005','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14006','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14007','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14008','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14009','CORDOBA');

INSERT INTO localidad (C_P,localidad)
VALUES('14114','GUADALCAZAR');

INSERT INTO localidad (C_P,localidad)
VALUES('14100','LA CARLOTA');

INSERT INTO localidad (C_P,localidad)
VALUES('12110','ALCORA');

INSERT INTO localidad (C_P,localidad)
VALUES('23004','MADRID');

INSERT INTO localidad (C_P,localidad)
VALUES('12001','CASTELLON DE LA PLANA');

INSERT INTO localidad (C_P,localidad)
VALUES('11002','CADIZ');

INSERT INTO localidad (C_P,localidad)
VALUES('16004','CORUÑA');

INSERT INTO localidad (C_P,localidad)
VALUES('22003','GRANADA');

```

```

INSERT INTO localidad (C_P,localidad)

VALUES('34009','SALAMANCA');

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('SAUL CEJUDO CORDOBA','CASA FAMILIAR','STO DOMINGO',3,14003);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('PEDRO LOPEZ','CASA FAMILIAR','PINTOR ESPINOSA',9,14003);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('SEVILLANA S.A.','SEDE PRINCIPAL','CRUZ CONDE',29,14001);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('EMACSA S.L.','SEDE PRINCIPAL','NOGAL',9,14004);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('TALLERES FIGO S.L.','NEGOCIO','CARLOS III',67,14007);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('JUANITO VALDERAMA','CASA FAMILIAR','JUAN BLAS',3,11002);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('MIGUEL LEIVA SALAS','CASA FAMILIAR','PTDA PEDRIZA',34,12110);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('MADONNA','CASA PLAYA','PINTOR MONROY',123,22003);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('REPSOL S.A.','SEDE PRINCIPAL','CLARA DEL REY',10,23004);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('MARTA SANCHEZ','CASA FAMILIAR','CORAZON DE MARIA',9,23004);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('PEDRO LOPEZ','PISO ESTUDIANTES','VIRGEN DOLORES',45,14004);

INSERT INTO direcciones(nombre,tipo_direccion,calle,numero,C_P)

VALUES('SAUL CEJUDO CORDOBA','PISO ESTUDIANTES','VIRGEN DOLORES',45,14004);

```

```

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('SEVILLANA S.A.', 'SUCURSAL', 'MARIANO RAJOY', 6, 14008);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('MIGUEL LEIVA SALAS', 'PISO ESTUDIANTES', 'MARIANO FORTUNY', 21, 14004);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('CAJASUR', 'SEDE PRINCIPAL', 'CRUZ CONDE', 28, 14002);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('CAJASUR', 'SUCURSAL', 'STO DOMINGO', 3, 14003);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('PEDRO LOPEZ', 'CASA PLAYA', 'LAS PALMERAS', 22, 22003);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('SAUL CEJUDO CORDOBA', 'CASA PLAYA', 'RODRIGO DIAZ', 85, 16004);

INSERT INTO direcciones(nombre, tipo_direccion, calle, numero, C_P)
VALUES('MIGUEL LEIVA SALAS', 'CASA SIERRA', 'PEDRO RUIZ', 104, 12110);

INSERT INTO direccionesfijo(alfanumerico, nombre, tipo_direccion)
VALUES('957280155', 'SAUL CEJUDO CORDOBA', 'CASA FAMILIAR');

INSERT INTO direccionesfijo(alfanumerico, nombre, tipo_direccion)
VALUES('957456789', 'PEDRO LOPEZ', 'CASA FAMILIAR');

INSERT INTO direccionesfijo(alfanumerico, nombre, tipo_direccion)
VALUES('957234567', 'SEVILLANA S.A.', 'SEDE PRINCIPAL');

INSERT INTO direccionesfijo(alfanumerico, nombre, tipo_direccion)
VALUES('957328976', 'EMACSA S.L.', 'SEDE PRINCIPAL');

INSERT INTO direccionesfijo(alfanumerico, nombre, tipo_direccion)
VALUES('957402098', 'TALLERES FIGO S.L.', 'NEGOCIO');

INSERT INTO direccionesfijo(alfanumerico, nombre, tipo_direccion)
VALUES('964386998', 'MIGUEL LEIVA SALAS', 'CASA FAMILIAR');

```

```

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('957488237','CAJASUR','SUCURSAL');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('955299287','JUANITO VALDERAMA','CASA FAMILIAR');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('957345671','SAUL CEJUDO CORDOBA','PISO ESTUDIANTES');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('914132939','MARTA SANCHEZ','CASA FAMILIAR');

INSERT INTO direccionesfijo(alfanumerico,nombre,tipo_direccion)
VALUES('957345671','MIGUEL LEIVA SALAS','PISO ESTUDIANTES');

INSERT INTO email (alfanumerico)
VALUES('i52lesam@uco.es');

INSERT INTO email (alfanumerico)
VALUES('i52cecos@uco.es');

INSERT INTO email (alfanumerico)
VALUES('peloj@hotmail.com');

INSERT INTO email (alfanumerico)
VALUES('lnavarro@ibm.com');

INSERT INTO email (alfanumerico)
VALUES('gccrab5@.repsol.com');

INSERT INTO entrada (nombre,tipo)
VALUES('SAUL CEJUDO CORDOBA','FISICO');

INSERT INTO entrada (nombre,tipo)
VALUES('PEDRO LOPEZ','JURIDICO');

INSERT INTO entrada (nombre,tipo)
VALUES('SEVILLANA S.A.','JURIDICO');

```

```

INSERT INTO entrada (nombre, tipo)
VALUES('EMACSA S.L.', 'JURIDICO');

INSERT INTO entrada (nombre, tipo)
VALUES('TALLERES FIGO S.L.', 'JURIDICO');

INSERT INTO entrada (nombre, tipo)
VALUES('JUANITO VALDERAMA', 'FISICO');

INSERT INTO entrada (nombre, tipo)
VALUES('MIGUEL LEIVA SALAS', 'FISICO');

INSERT INTO entrada (nombre, tipo)
VALUES('MADONNA', 'FISICO');

INSERT INTO entrada (nombre, tipo)
VALUES('REPSOL S.A.', 'JURIDICO');

INSERT INTO entrada (nombre, tipo)
VALUES('MARTA SANCHEZ', 'FISICO');

INSERT INTO entrada (nombre, tipo)
VALUES('CAJASUR', 'JURIDICO');

INSERT INTO entrada (nombre, tipo)
VALUES('PEPE', 'JURIDICO');

INSERT INTO entrada (nombre, tipo)
VALUES('JUAN LUCIA', 'JURIDICO');

INSERT INTO entrada (nombre, tipo)
VALUES('JUAN GOMEZ', 'FISICO');

INSERT INTO entradaapunte (nombre, apunte)
VALUES('SAUL CEJUDO CORDOBA', 'COMPAÑERO DE CLASE');

INSERT INTO entradaapunte (nombre, apunte)
VALUES('MIGUEL LEIVA SALAS', 'COMPAÑERO DE CLASE');

```

```

INSERT INTO entradaapuntes (nombre,apuntes)
VALUES('PEDRO LOPEZ','ABOGADO');

INSERT INTO entradaapuntes (nombre,apuntes)
VALUES('JUANITO VALDERAMA','AMIGO DE SAUL');

INSERT INTO entradaapuntes (nombre,apuntes)
VALUES('TALLERES FIGO S.L.','ASISTENCIA TECNICA DE MI MOTO');

INSERT INTO entradaapuntes (nombre,apuntes)
VALUES('SEVILLANA S.A.','COMPAÑIA DE ELECTRICIDAD');

INSERT INTO entradaapuntes (nombre,apuntes)
VALUES('EMACSA S.L.','COMPAÑIA DE AGUA');

INSERT INTO entradaapuntes (nombre,apuntes)
VALUES('REPSOL S.A.','COMPAÑIA PETROLIFERA');

INSERT INTO entradaapuntes (nombre,apuntes)
VALUES('MIGUEL LEIVA SALAS','ENEMIGO');

INSERT INTO entradaapuntes (nombre,apuntes)
VALUES('PEDRO LOPEZ','ENEMIGO');

INSERT INTO entradaemail (alfanumerico,nombre)
VALUES('i52lesam@uco.es','MIGUEL LEIVA SALAS');

INSERT INTO entradaemail (alfanumerico,nombre)
VALUES('i52cecos@uco.es','SAUL CEJUDO CORDOBA');

INSERT INTO entradaemail (alfanumerico,nombre)
VALUES('peloj@hotmail.com','PEDRO LOPEZ');

INSERT INTO entradaemail (alfanumerico,nombre)
VALUES('lnavarro@ibm.com','SEVILLANA S.A.');
```

```

INSERT INTO entradaemail (alfanumerico,nombre)
VALUES('gccrab5@.repsol.com','REPSOL S.A.');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)
VALUES('957280155','SAUL CEJUDO CORDOBA');

INSERT INTO entradafijo (alfanumerico,nombre)
VALUES('957456789','PEDRO LOPEZ');

INSERT INTO entradafijo (alfanumerico,nombre)
VALUES('957234567','SEVILLANA S.A.');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)
VALUES('957328976','EMACSA S.L.');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)
VALUES('964386998','MIGUEL LEIVA SALAS');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)
VALUES('914132939','MARTA SANCHEZ');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)
VALUES('957402098','TALLERES FIGO S.L.');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)
VALUES('957488237','CAJASUR');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)
VALUES('957474063','EMACSA S.L.');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)
VALUES('955299287','JUANITO VALDERAMA');
```

```

INSERT INTO entradafijo (alfanumerico,nombre)
VALUES('957345671','SAUL CEJUDO CORDOBA');
```

```

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('MIGUEL LEIVA SALAS','609453832');
```

```

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('SAUL CEJUDO CORDOBA','639895433');
```

```

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('MIGUEL LEIVA SALAS','630887654');

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('SEVILLANA S.A.','630993625');

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('PEDRO LOPEZ','639454533');

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('MADONNA','630558987');

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('JUANITO VALDERAMA','639458796');

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('TALLERES FIGO S.L.','607565654');

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('SAUL CEJUDO CORDOBA','609541244');

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('MIGUEL LEIVA SALAS','630591204');

INSERT INTO entradamovil (nombre,alfanumerico)
VALUES('SAUL CEJUDO CORDOBA','630594455');

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
VALUES('SAUL CEJUDO CORDOBA','ANIVERSARIO DE
BODA',TO_DATE('05-05-1999','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
VALUES('PEDRO LOPEZ','CUMPLEAÑOS',TO_DATE('07-23-1997','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
VALUES('PEDRO LOPEZ','SANTO',TO_DATE('7-23','MM-DD'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)
VALUES('SEVILLANA S.A.','FECHA DE ALTA',TO_DATE('03-04-1995','MM-DD-YYYY'));

```

```

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)

VALUES('EMACSA S.L.','FECHA DE ALTA',TO_DATE('11-29-1995','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)

VALUES('JUANITO VALDERAMA','CUMPLEAÑOS',TO_DATE('12-24-1940','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)

VALUES('MIGUEL LEIVA SALAS','CUMPLEAÑOS',TO_DATE('12-08-1975','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)

VALUES('SAUL CEJUDO CORDOBA','CUMPLEAÑOS',TO_DATE('05-29-1977','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)

VALUES('MARTA SANCHEZ','CUMPLEAÑOS',TO_DATE('06-09-1969','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)

VALUES('JUANITO VALDERAMA','ANIVERSARIO DE
BODA',TO_DATE('10-25-1959','MM-DD-YYYY'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)

VALUES('MIGUEL LEIVA SALAS','SANTO',TO_DATE('10-29','MM-DD'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)

VALUES('SAUL CEJUDO CORDOBA','SANTO',TO_DATE('09-24','MM-DD'));

INSERT INTO FECHAS(nombre,tipo_fecha,fecha)

VALUES('MARTA SANCHEZ','SANTO',TO_DATE('10-01','MM-DD'));

INSERT INTO fijo (alfanumerico,tipo)

VALUES('957280155','telefono');

INSERT INTO fijo (alfanumerico,tipo)

VALUES('957456789','telefono');

INSERT INTO fijo (alfanumerico,tipo)

VALUES('957234567','telefono');

INSERT INTO fijo (alfanumerico,tipo)

VALUES('957328976','telefono');

```

```
INSERT INTO fijo (alfanumerico, tipo)
VALUES('964386998', 'telefono');

INSERT INTO fijo (alfanumerico, tipo)
VALUES('914132939', 'telefono');

INSERT INTO fijo (alfanumerico, tipo)
VALUES('957402098', 'fax');

INSERT INTO fijo (alfanumerico, tipo)
VALUES('957488237', 'fax');

INSERT INTO fijo (alfanumerico, tipo)
VALUES('957474063', 'telefono');

INSERT INTO fijo (alfanumerico, tipo)
VALUES('955299287', 'telefono');

INSERT INTO fijo (alfanumerico, tipo)
VALUES('957345671', 'telefono');

INSERT INTO movil (alfanumerico, tipo)
VALUES('609453832', 'telefono');

INSERT INTO movil (alfanumerico, tipo)
VALUES('639895433', 'telefono');

INSERT INTO movil (alfanumerico, tipo)
VALUES('630887654', 'telefono');

INSERT INTO movil (alfanumerico, tipo)
VALUES('630993625', 'telefono');

INSERT INTO movil (alfanumerico, tipo)
VALUES('639454533', 'telefono');

INSERT INTO movil (alfanumerico, tipo)
VALUES('630558987', 'telefono');
```

```

INSERT INTO movil (alfanumerico,tipo)
VALUES('639458796','telefono');

INSERT INTO movil (alfanumerico,tipo)
VALUES('607565654','telefono');

INSERT INTO movil (alfanumerico,tipo)
VALUES('609541244','telefono');

INSERT INTO movil (alfanumerico,tipo)
VALUES('630591204','telefono');

INSERT INTO movil (alfanumerico,tipo)
VALUES('630594455','fax');

INSERT INTO PERSONAFISICA (nombre,dni)
VALUES ('SAUL CEJUDO CORDOBA',3094302);

INSERT INTO PERSONAFISICA (nombre,dni)
VALUES ('JUANITO VALDERAMA',4567598);

INSERT INTO PERSONAFISICA (nombre,dni)
VALUES ('MIGUEL LEIVA SALAS',30304056);

INSERT INTO PERSONAFISICA (nombre,dni)
VALUES ('MADONNA',69696969);

INSERT INTO PERSONAFISICA (nombre,dni)
VALUES ('MARTA SANCHEZ',86759465);

INSERT INTO personajuridica (nombre,cif)
VALUES('PEDRO LOPEZ',NULL);

INSERT INTO personajuridica (nombre,cif)
VALUES('SEVILLANA S.A.',NULL);

INSERT INTO personajuridica (nombre,cif)
VALUES('EMACSA S.L.',NULL);

```

```

INSERT INTO personajuridica (nombre,cif)
VALUES('TALLERES FIGO S.L.',NULL);

INSERT INTO personajuridica (nombre,cif)
VALUES('REPSOL S.A.',NULL);

INSERT INTO personajuridica (nombre,cif)
VALUES('CAJASUR',NULL);

INSERT INTO PERSONAFISICA (nombre,dni)
VALUES ('SAUL CEJUDO CORDOBA',3094302);

INSERT INTO PERSONAFISICA (nombre,dni)
VALUES ('JUANITO VALDERAMA',4567598);

INSERT INTO PERSONAFISICA (nombre,dni)
VALUES ('MIGUEL LEIVA SALAS',30304056);

INSERT INTO PERSONAFISICA (nombre,dni)
VALUES ('MADONNA',69696969);

INSERT INTO PERSONAFISICA (nombre,dni)
VALUES ('MARTA SANCHEZ',86759465);

-- UPDATES SIMPLES

UPDATE fechas set fecha = to_date('12-25-1999','MM-DD-YYYY')

where tipo_fecha='ANIVERSARIO DE BODA'

AND

nombre='SAUL CEJUDO CORDOBA';

UPDATE personafisica set dni = 45678976

where nombre='SAUL CEJUDO CORDOBA';

UPDATE personajuridica set cif = '87654876-E'

where nombre='REPSOL S.A.';

-- UPDATES COMPLEJOS

```

-- Queremos cambiar la dirección del piso de estudiante que SAUL CEJUDO

-- posee por la dirección del piso de estudiante Miguel Leiva Salas.

```
UPDATE direcciones SET (calle,C_P,numero) =
```

```
(select calle,C_P,numero from direcciones
```

```
WHERE nombre = 'MIGUEL LEIVA SALAS'
```

```
and
```

```
tipo_direccion = 'PISO ESTUDIANTES')
```

```
WHERE nombre = 'SAUL CEJUDO CORDOBA'
```

```
AND tipo_direccion= 'PISO ESTUDIANTES';
```

```
UPDATE direccionesfijo SET alfanumerico =
```

```
(select alfanumerico from direccionesfijo
```

```
WHERE nombre = 'MIGUEL LEIVA SALAS'
```

```
and
```

```
tipo_direccion = 'PISO ESTUDIANTES')
```

```
WHERE nombre = 'SAUL CEJUDO CORDOBA'
```

```
AND tipo_direccion= 'PISO ESTUDIANTES' ;
```

-- Deseamos actualizar los telefonos con una supuesta reforma de telefo-

-- nica que establece que todos los telefonos moviles que tengan los primeros

-- tres digitos igual a 630 son faxes.

```
UPDATE MOVIL SET tipo = 'fax'
```

```
where alfanumerico IN
```

```
(SELECT ALFANUMERICO FROM MOVIL
```

```
WHERE alfanumerico like '630%'
```

```
and tipo='telefono');
```

-- Si una persona no tiene ninguna forma de contacto, consideramos que

-- nos interesa y lo sacaremos de nuestra agenda. toma ya!!!

```

CREATE OR REPLACE PROCEDURE checknombres(p_nombre VARCHAR2)

AS

num_dir NUMBER;

num_mov NUMBER;

num_fijo NUMBER;

num_email NUMBER;

no_direccion EXCEPTION;

si_direccion EXCEPTION;

BEGIN

num_dir:=0;

num_mov:=0;

num_fijo:=0;

num_email:=0;

SELECT COUNT(nombre) INTO num_dir from direcciones

WHERE nombre = p_nombre;

SELECT COUNT(nombre) INTO num_mov from entradamovil

WHERE nombre = p_nombre;

SELECT COUNT(nombre) INTO num_fijo from entradafijo

WHERE nombre = p_nombre;

SELECT COUNT(nombre) INTO num_email from entradaemail

WHERE nombre = p_nombre;

IF num_dir=0 AND num_mov=0 AND num_email=0 AND num_fijo=0 THEN

DBMS_OUTPUT.PUT_LINE('NO ASIGNADO UNA DIRECCION');

delete from entrada where nombre= p_nombre;

RAISE no_direccion;

END IF;

```

```

RAISE si_direccion;

EXCEPTION

WHEN no_direccion THEN

DBMS_OUTPUT.PUT_LINE('NO ASIGNADO UNA DIRECCION');

WHEN si_direccion THEN

DBMS_OUTPUT.PUT_LINE('NO ASIGNADO UNA DIRECCION');

END;

/

```

-- Ejemplo de aplicación

```
execute checknombres ('PEPE');
```

```
execute checknombres ('ANTONIO');
```

-- deseamos crear un procedimiento que dado un nombre lo introduzca

-- en la tabla fechas y le de los valores 'SANTO' para tipo_fecha

-- y la fecha del dia actual para fecha. Solo si no existe una tupla para

-- este tipo de fecha.

```
CREATE OR REPLACE PROCEDURE hoy_santo_de(p_nombre VARCHAR2)
```

```
AS
```

```
CURSOR c_nombres IS
```

```
SELECT nombre,tipo
```

```
FROM entrada
```

```
WHERE substr(nombre,1,length(p_nombre)) like p_nombre
```

```
AND nombre NOT IN
```

```
(SELECT nombre from fechas
```

```
WHERE tipo_fecha = 'SANTO');
```

```
v_nombres c_nombres%ROWTYPE;
```

```
BEGIN
```

```

OPEN c_nombres;

FETCH c_nombres INTO v_nombres;

WHILE c_nombres%FOUND LOOP

INSERT INTO fechas (nombre, tipo_fecha, fecha)

VALUES(v_nombres.nombre, 'SANTO', sysdate);

FETCH c_nombres INTO v_nombres;

END LOOP;

CLOSE c_nombres;

END hoy_santo_de;

/

EXECUTE hoy_santo_de('MADONNA');

-- Ha existido una modificacion en los prefijos de los telefonos fijos
-- en el que el prefijo 957 se ha modificado por el 956. Se ha imple-
-- mentado un procedimiento de forma general que recibe como parametro
-- el prefijo antiguo y el nuevo prefijo por el que se sustituirá.
-- (los cambios serán generalizados para todas nuestras relaciones).

CREATE OR REPLACE PROCEDURE cambioprefijo(prefijoa VARCHAR2,
prefijon VARCHAR2)

AS

CURSOR c_alfanum IS

SELECT nombre, alfanumerico

FROM entradafijo

WHERE substr(alfanumerico, 1, 3) like prefijoa;

CURSOR c_alfanum2 IS

SELECT alfanumerico, tipo

FROM fijo

```

```

WHERE substr(alfanumerico,1,3) like prefijoa;

CURSOR c_alfanum3 IS

SELECT alfanumerico,tipo_direccion,nombre

FROM direccionesfijo

WHERE substr(alfanumerico,1,3) like prefijoa;

v_nombres c_alfanum%ROWTYPE;

v_alfanumero c_alfanum2%ROWTYPE;

v_direc c_alfanum3%ROWTYPE;

indice NUMBER:=0;

total NUMBER:=0;

total2 NUMBER:=0;

total3 NUMBER:=0;

TYPE tabla IS TABLE OF c_alfanum%ROWTYPE INDEX BY BINARY_INTEGER;

TYPE tabla2 IS TABLE OF c_alfanum2%ROWTYPE INDEX BY BINARY_INTEGER;

TYPE tabla3 IS TABLE OF c_alfanum3%ROWTYPE INDEX BY BINARY_INTEGER;

v_tabla tabla;

v_tabla2 tabla2;

v_tabla3 tabla3;

BEGIN

OPEN c_alfanum;

OPEN c_alfanum2;

OPEN c_alfanum3;

select count(alfanumerico) into total from entradafijo

where substr(alfanumerico,1,3) like prefijoa;

select count(alfanumerico) into total2 from fijo

where substr(alfanumerico,1,3) like prefijoa;

```

```

select count(alfanumerico) into total3 from fijo

where substr(alfanumerico,1,3) like prefijoa;

FOR indice IN 1..total2 LOOP

FETCH c_alfanum2 INTO v_tabla2(indice);

INSERT INTO fijo (alfanumerico,tipo)

VALUES(prefijon || substr(v_tabla2(indice).alfanumerico,4,20),

v_tabla2(indice).tipo);

END LOOP;

FOR indice IN 1..total3 LOOP

FETCH c_alfanum3 INTO v_tabla3(indice);

INSERT INTO direccionesfijo (alfanumerico,nombre,tipo_direccion)

VALUES(prefijon || substr(v_tabla3(indice).alfanumerico,4,20),

v_tabla3(indice).nombre,

v_tabla3(indice).tipo_direccion);

END LOOP;

FOR indice IN 1..total LOOP

FETCH c_alfanum INTO v_tabla(indice);

INSERT INTO entradafijo (nombre,alfanumerico)

VALUES(v_tabla(indice).nombre,prefijon || substr(v_tabla(indice).alfanumerico,4,20));

END LOOP;

delete FROM entradafijo

WHERE substr(alfanumerico,1,3) like prefijoa;

delete FROM direccionesfijo

WHERE substr(alfanumerico,1,3) like prefijoa;

delete FROM fijo

WHERE substr(alfanumerico,1,3) like prefijoa;

```

```

CLOSE c_alfanum;

CLOSE c_alfanum2;

CLOSE c_alfanum3;

END cambioprefijo;

/

-- Ejemplo de aplicación

EXECUTE tablas_ejemplo('957','956');

-- Si una persona no tiene ninguna forma de contacto, consideramos que

-- nos interesa y lo sacaremos de nuestra agenda. toma ya!!!

CREATE OR REPLACE PROCEDURE checknombres(p_nombre VARCHAR2)

AS

num_dir NUMBER;

num_mov NUMBER;

num_fijo NUMBER;

num_email NUMBER;

no_direccion EXCEPTION;

si_direccion EXCEPTION;

BEGIN

num_dir:=0;

num_mov:=0;

num_fijo:=0;

num_email:=0;

SELECT COUNT(nombre) INTO num_dir from direcciones

WHERE nombre = p_nombre;

SELECT COUNT(nombre) INTO num_mov from entradamovil

WHERE nombre = p_nombre;

```

```

SELECT COUNT(nombre) INTO num_fijo from entradafijo

WHERE nombre = p_nombre;

SELECT COUNT(nombre) INTO num_email from entradaemail

WHERE nombre = p_nombre;

IF num_dir=0 AND num_mov=0 AND num_email=0 AND num_fijo=0 THEN

DBMS_OUTPUT.PUT_LINE('NO ASIGNADO UNA DIRECCION');

delete from entrada where nombre= p_nombre;

RAISE no_direccion;

END IF;

RAISE si_direccion;

EXCEPTION

WHEN no_direccion THEN

DBMS_OUTPUT.PUT_LINE('NO ASIGNADO UNA DIRECCION');

WHEN si_direccion THEN

DBMS_OUTPUT.PUT_LINE('NO ASIGNADO UNA DIRECCION');

END;

/

-- Ejemplo de aplicación

execute checknombres ('PEPE');

execute checknombres ('ANTONIO');

-- deseamos crear un procedimiento que dado un nombre lo introduzca

-- en la tabla fechas y le de los valores 'SANTO' para tipo_fecha

-- y la fecha del dia actual para fecha. Solo si no existe una tupla para

-- este tipo de fecha.

CREATE OR REPLACE PROCEDURE hoy_santo_de(p_nombre VARCHAR2)

AS

```

```

CURSOR c_nombres IS

SELECT nombre,tipo

FROM entrada

WHERE substr(nombre,1,length(p_nombre)) like p_nombre

AND nombre NOT IN

(SELECT nombre from fechas

WHERE tipo_fecha = 'SANTO');

v_nombres c_nombres%ROWTYPE;

BEGIN

OPEN c_nombres;

FETCH c_nombres INTO v_nombres;

WHILE c_nombres%FOUND LOOP

INSERT INTO fechas (nombre,tipo_fecha,fecha)

VALUES(v_nombres.nombre,'SANTO',sysdate);

FETCH c_nombres INTO v_nombres;

END LOOP;

CLOSE c_nombres;

END hoy_santo_de;

/

EXECUTE hoy_santo_de('MADONNA');

-- Ha existido una modificacion en los prefijos de los telefonos fijos
-- en el que el prefijo 957 se ha modificado por el 956. Se ha imple-
-- mentado un procedimiento de forma general que recibe como parametro
-- el prefijo antiguo y el nuevo prefijo por el que se sustituirá.
-- (los cambios serán generalizados para todas nuestras relaciones).

CREATE OR REPLACE PROCEDURE cambioprefijo(prefijoa VARCHAR2,

```

```

prefijon VARCHAR2)

AS

CURSOR c_alfanum IS

SELECT nombre,alfanumerico

FROM entradafijo

WHERE substr(alfanumerico,1,3) like prefijoa;

CURSOR c_alfanum2 IS

SELECT alfanumerico,tipo

FROM fijo

WHERE substr(alfanumerico,1,3) like prefijoa;

CURSOR c_alfanum3 IS

SELECT alfanumerico,tipo_direccion,nombre

FROM direccionesfijo

WHERE substr(alfanumerico,1,3) like prefijoa;

v_nombres c_alfanum%ROWTYPE;

v_alfanumero c_alfanum2%ROWTYPE;

v_direc c_alfanum3%ROWTYPE;

indice NUMBER:=0;

total NUMBER:=0;

total2 NUMBER:=0;

total3 NUMBER:=0;

TYPE tabla IS TABLE OF c_alfanum%ROWTYPE INDEX BY BINARY_INTEGER;

TYPE tabla2 IS TABLE OF c_alfanum2%ROWTYPE INDEX BY BINARY_INTEGER;

TYPE tabla3 IS TABLE OF c_alfanum3%ROWTYPE INDEX BY BINARY_INTEGER;

v_tabla tabla;

v_tabla2 tabla2;

```

```

v_tabla3 tabla3;

BEGIN

OPEN c_alfanum;

OPEN c_alfanum2;

OPEN c_alfanum3;

select count(alfanumerico) into total from entradafijo

where substr(alfanumerico,1,3) like prefijoa;

select count(alfanumerico) into total2 from fijo

where substr(alfanumerico,1,3) like prefijoa;

select count(alfanumerico) into total3 from fijo

where substr(alfanumerico,1,3) like prefijoa;

FOR indice IN 1..total2 LOOP

FETCH c_alfanum2 INTO v_tabla2(indice);

INSERT INTO fijo (alfanumerico,tipo)

VALUES(prefijon || substr(v_tabla2(indice).alfanumerico,4,20),

v_tabla2(indice).tipo);

END LOOP;

FOR indice IN 1..total3 LOOP

FETCH c_alfanum3 INTO v_tabla3(indice);

INSERT INTO direccionesfijo (alfanumerico,nombre,tipo_direccion)

VALUES(prefijon || substr(v_tabla3(indice).alfanumerico,4,20),

v_tabla3(indice).nombre,

v_tabla3(indice).tipo_direccion);

END LOOP;

FOR indice IN 1..total LOOP

FETCH c_alfanum INTO v_tabla(indice);

```

```

INSERT INTO entradafijo (nombre,alfanumerico)

VALUES(v_tabla(indice).nombre,prefijon || substr(v_tabla(indice).alfanumerico,4,20));

END LOOP;

delete FROM entradafijo

WHERE substr(alfanumerico,1,3) like prefijoa;

delete FROM direccionesfijo

WHERE substr(alfanumerico,1,3) like prefijoa;

delete FROM fijo

WHERE substr(alfanumerico,1,3) like prefijoa;

CLOSE c_alfanum;

CLOSE c_alfanum2;

CLOSE c_alfanum3;

END cambioprefijo;

/

-- Ejemplo de aplicación

EXECUTE tablas_ejemplo('957','956');

SELECT nombre, alfanumerico from entradafijo

WHERE nombre not IN

(SELECT nombre from direcciones

WHERE C_P IN

(select C_P from localidad

where localidad NOT LIKE 'CORDOBA'

))

ORDER BY nombre;

SELECT nombre, alfanumerico from entrada,fijo

WHERE nombre not IN

```

```

(SELECT nombre from direcciones
WHERE C_P IN
(select C_P from localidad
where localidad NOT LIKE 'CORDOBA'
)
INTERSECT
select nombre,alfanumerico from entradafijo
);

select to_char(fecha) from fechas;

start c:\misdoc~1\sql\create
start c:\misdoc~1\sql\ins_ent
start c:\misdoc~1\sql\ins_cp
start c:\misdoc~1\sql\ins_apu
start c:\misdoc~1\sql\ins_mov
start c:\misdoc~1\sql\ins_fij
start c:\misdoc~1\sql\ins_ema
start c:\misdoc~1\sql\ins_entema
start c:\misdoc~1\sql\ins_entfij
start c:\misdoc~1\sql\ins_entmov
start c:\misdoc~1\sql\ins_fech
start c:\misdoc~1\sql\ins_direc
start c:\misdoc~1\sql\ins_dirfij
start c:\misdoc~1\sql\ins_pef
start c:\misdoc~1\sql\ins_pej
start c:\misdoc~1\sql\ins_entapu
start c:\misdoc~1\sql\ins_com

```

PRACTICAS

/* Hacer que sea posible introducir una duracion por clase practica de mas de 10 horas */

```
ALTER TABLE clase_practica
```

```
DISABLE CONSTRAINT ck_duracion;
```

/* Deshabilitar la restriccion de clave foranea en tramitacion de documentos a la tabla

coste_tramitacion para hacer un cambio en la extension de esta ultima */

```
ALTER TABLE tramitacion_documentos
```

```
DISABLE CONSTRAINT fk_coste_tram;
```

/* Añadir un atributo codigo_postal a la entidad alumno, de tipo NUMBER */

```
ALTER TABLE alumno
```

```
ADD (codigo_postal NUMBER(10));
```

/* Obligar a que se facilite una direccion para cada alumno */

```
ALTER TABLE alumno
```

```
ADD CONSTRAINT ck_direccion
```

```
CHECK (direccion <> NULL);
```

/* El alumno con dni igual a 1 se ha mudado a la direccion 'Puerta Real 7' y con telefono 867043 */

```
UPDATE alumno SET
```

```
direccion = 'Puerta Real 7' ,
```

```
telefono = 867043
```

```
WHERE dni_al = 1;
```

/* Cambiar el coste de matriculacion a 3500 Ptas */

```
UPDATE coste_matriculacion SET
```

```
importe = 3500
```

```
WHERE coste_mat = 'coste_mat';
```

/* Subir un 30% el lujo EXCLUSIVO y EJECUTIVO de los vehiculos */

```
UPDATE lujo SET
```

```
tarifa_adicional = tarifa_adicional + tarifa_adicional * 0.30
```

```
WHERE tipo_lujo = 'ejecutivo' OR tipo_lujo = 'exclusivo';
```

```
/* Reducir la tarifa base un 40% de los vehiculos que superan en la media el numero total de horas  
de practicas que se han realizado en la autoescuela */
```

```
UPDATE vehiculo SET
```

```
tarifa = tarifa - tarifa*0.30
```

```
WHERE matricula IN ( SELECT seleccionados.matricula
```

```
FROM (SELECT v.matricula, SUM(cp.duracion)
```

```
FROM vehiculo v,consumo c,clase_practica cp
```

```
WHERE v.matricula = c.matricula AND c.fecha_practica = cp.fecha_practica
```

```
AND c.n_matricula = cp.n_matricula
```

```
GROUP BY v.matricula
```

```
HAVING SUM(cp.duracion) > (SELECT AVG(duracion)
```

```
FROM clase_practica
```

```
)
```

```
) seleccionados
```

```
);
```

```
/* Reducir un 2% el numero la duracion de cada clase practica del alumno que mas horas haya realizado  
en la autoescuela para la licencia 'B' */
```

```
UPDATE clase_practica SET
```

```
duracion = duracion - duracion*0.02
```

```
WHERE n_matricula IN
```

```
(SELECT maximos.n_matricula
```

```
FROM (SELECT cp.n_matricula, SUM(cp.duracion)
```

```
FROM matricula m,clase_practica cp
```

```
WHERE m.n_matricula = cp.n_matricula AND m.tipo_lic = 'B'
```

```

GROUP BY cp.n_matricula

HAVING SUM(cp.duracion) = (SELECT MAX(sumas.recuento)

FROM (SELECT cp.n_matricula,

SUM(cp.duracion) recuento

FROM matricula m,clase_practica cp

WHERE m.n_matricula = cp.n_matricula AND

m.tipo_lic = 'B'

GROUP BY cp.n_matricula) sumas

)

) maximos

);

/* Aumentar un 10% los consumos realizados por el alumno con dni igual a 8 en todas las licencias

en las que se haya matriculado */

UPDATE consumo SET

litros = litros + litros*0.1,

kilometros = kilometros + kilometros*0.1

WHERE n_matricula IN(

SELECT m.n_matricula

FROM matricula m

WHERE m.dni_al = 8

);

SELECCIONADA

/* Obtener el alumno que mas errores comete en las pruebas teoricas, contando todos los errores de todos

los test */

/* Para ello primero tenemos que calcular para cada alumno todos los errores que comete en todos los test

que realiza. Esa es la mision del select mas interior, del que lo unico que hay que comentar es que hacemos

```

la reunion por el atributo n_matricula, haciendo entonces aparecer en el producto cartesiano todas las matriculas

con sus pruebas teoricas. Como es obvio, saldran varias tuplas con el dni_al iguales, y esta es la clave para realizar los grupos en funcion del dni_al y calcular la suma de errores. Vemos como calculamos el maximo del

conjunto de alumnos. Despues (en el select mas exterior) volvemos a generar la misma tabla anterior, pero solo

mostramos los grupos (clausula HAVING) que tienen una suma igual al maximo*/

```
select a.dni_al dni,a.nombre,a.apellido_1,a.apellido_2,sum(pt.errores)
```

```
from alumno a,matricula m,prueba_teorica pt
```

```
where a.dni_al = m.dni_al AND m.n_matricula = pt.n_matricula
```

```
group by a.dni_al,a.nombre,a.apellido_1,a.apellido_2
```

```
having sum(pt.errores) =
```

```
(select max(err_al.errores)
```

```
from (select m.dni_al,sum(pt.errores) errores
```

```
from matricula m,prueba_teorica pt
```

```
where pt.n_matricula = m.n_matricula
```

```
group by m.dni_al) err_al
```

```
)
```

```
;
```

SELECCIONADA

/* Obtener el total de ingresos en la autoescuela desglosada por alumnos */

La orden de consulta es sencilla. Necesitamos generar una tabla en la que para cada alumno, venga

reflejado cada pago que realiza, es decir, que por cada pago realizado por un alumno tengamos una tupla

en esa tabla. Una vez realizado esto, calculamos la suma total para cada alumno agrupando por el identificador

del alumno

```
select a.dni_al DNI,a.nombre nombre,a.apellido_1 app1,a.apellido_2 app2 ,sum(p.importe) Ptas
```

```

from alumno a,matricula m,pago p

where a.dni_al = m.dni_al AND m.n_matricula = p.n_matricula

group by a.dni_al,a.nombre,a.apellido_1,a.apellido_2

;

```

SELECCIONADA

/* Calcular lo que cada alumno adeuda a la autoescuela en concepto de clases practicas para todas las licencias de las que se ha matriculado, mostrando tambien los litros totales consumidos y los kilometros totales realizados. Como cada clase practica se realiza en un solo vehiculo, tendremos que calcular para cada alumno y para cada vehiculo que usa en las clases practicas de las licencias en que se ha matriculado, el importe que debe a la autoescuela.

La resolucio[n] de esta consulta involucra a seis tablas: alumno,matricula, clase_practica, vehiculo, lujo y consumo. Para realizar el calculo, necesitamos averiguar cuanto debe un alumno por cada clase practica. Para ello

tenemos que multiplicar el numero de horas realizadas por la suma de la tarifa base del vehiculo mas la tarifa adicional en base al lujo del mismo. Y una vez realizado esto, tenemos que sumar para cada alumno el coste de cada

clase practica. La clausula WHERE se encarga de seleccionar las tuplas de tal manera que aparezcan tantas tuplas

con el mismo dni y tipo de licencia como clases practicas haya realizado el alumno en ese tipo de licencia.

La clausula GROUP BY se encarga de hacer los grupos en base al identificador del alumno, el nombre los apellidos y el tipo de licencia del que se ha matriculado.

```

select a.dni_al dni,a.nombre,a.apellido_1,a.apellido_2,v.tipo_lic licencia,sum(cp.duracion) Horas,
sum(c.kilometros) Km, sum(c.litros) Litros, sum(cp.duracion * (v.tarifa + l.tarifa_adicional)) Ptas
from matricula m,clase_practica cp,vehiculo v,lujo l,consumo c,alumno a

where a.dni_al = m.dni_al AND m.n_matricula = cp.n_matricula AND cp.n_matricula = c.n_matricula AND
cp.fecha_practica = c.fecha_practica AND c.matricula = v.matricula AND
v.tipo_lujo = l.tipo_lujo

```

```
group by a.dni_al,a.nombre,a.apellido_1,a.apellido_2,v.tipo_lic
```

```
;
```

SELECCIONADA

/* Suma desglosada de los litros,kilometros,horas de practicas de cada alumno en cada vehiculo y con cada profesor en la autoescuela.

Simplemente observamos que un vehiculo esta asociado a una docencia, y una o mas de estas

a un profesor. Organizamos los grupos primero en base al alumno, despues en base al profesor y por

ultimo en base al vehiculo, ya que los consumos estan asociados a los vehiculos. Aunque habra que tener en cuenta

las clases practicas porque ahi viene reflejada la duracion de cada practica. Veamos

```
select m.dni_al,m.tipo_lic,m.dni_prof,c.matricula,sum(cp.duracion) horas, sum(c.kilometros) Km,  
sum(c.litros) l
```

```
from matricula m,clase_practica cp,consumo c
```

```
where m.n_matricula = cp.n_matricula AND cp.n_matricula = c.n_matricula AND
```

```
cp.fecha_practica = c.fecha_practica
```

```
group by m.dni_al,m.tipo_lic,m.dni_prof,c.matricula;
```

SELECCIONADA

/* Encontrar Los alumnos que superan la media de litros consumidos en las practicas. Mostrar ademas la media de litros gastados. */

Para esta consulta necesitamos saber los litros que cada alumno ha gastado en el total de sus clases

practicas para cualquier licencia de la que se ha matriculado, y conocer la media de litros gastados en todas

las clases practicas. Para conocer el primer conjunto de datos, usamos el primer select anidado dentro de

la clausula from que hemos etiquetado con el nombre litros_alumno. Como puede observarse, esta sentencia

agrupa todos los gastos de un alumno para cualquier licencia de la que tiene registradas practicas, y calcula

el gasto total de litros. Para conocer la media de litros gastados en las clases practicas, como sabemos que

por cada clase practica hay uno y solo un consumo, simplemente calculamos la media en base a la tabla

consumo. El segundo select de la clausula from realiza este calculo. Lo hemos etiquetado como c para

poder referenciarlo en el select superior y así poder mostrar la media en todos los alumnos. Por último, comentar que en el select más exterior se ven involucradas las tablas alumno y las dos anteriormente generadas. La cláusula WHERE de este select más externo, se encarga de relacionar a cada alumno con su gasto en litros y además de mostrar solo los alumnos que han superado la media de litros gastados

```
select a.nombre,a.apellido_1,a.apellido_2,a.dni_al,litros_alumno.suma,c.media
```

```
from alumno a,
```

```
(select a.dni_al,sum(c.litros) suma
```

```
from alumno a,matricula m,clase_practica cp,consumo c
```

```
where a.dni_al = m.dni_al AND m.n_matricula = cp.n_matricula AND cp.n_matricula = c.n_matricula
```

```
AND cp.fecha_practica = c.fecha_practica
```

```
group by a.dni_al) litros_alumno,
```

```
(select avg(consumo.litros) media from consumo) c
```

```
where litros_alumno.suma > c.media AND litros_alumno.dni_al = a.dni_al
```

```
;
```

```
select d.dni_prof,d.tipo_lic,count(n_matricula)
```

```
from matricula m,docencia d
```

```
where m.tipo_lic = d.tipo_lic AND m.dni_prof = d.dni_prof AND m.tipo_lic = 'A'
```

```
group by d.dni_prof,d.tipo_lic
```

```
;
```

```
select m.dni_al,c.litros
```

```
from matricula m,clase_practica cp,consumo c
```

```
where m.n_matricula = cp.n_matricula AND cp.n_matricula = c.n_matricula
```

```
AND cp.fecha_practica = c.fecha_practica
```

```
;
```

```
select m.dni_al,max(td.fecha_tramitacion)
```

```
from matricula m,tramitacion_documentos td
```

where m.n_matricula = td.n_matricula AND m.dni_al = 3 AND m.tipo_lic = 'A'

group by m.dni_al;

select m.dni_al,td.fecha_tramitacion,count(e.fecha_examen)

from matricula m,tramitacion_documentos td,examen e

where m.n_matricula = td.n_matricula AND td.n_matricula = e.n_matricula AND td.fecha_tramitacion = e.fecha_tramitacion

group by m.dni_al,td.fecha_tramitacion;

ALTER TABLE DOCENCIA

DROP CONSTRAINT FK_VEHICULO;

DELETE D_L;

DROP TABLE D_L;

DELETE DOCUMENTACION;

DROP TABLE DOCUMENTACION;

DELETE PRUEBA_TEORICA;

DROP TABLE PRUEBA_TEORICA;

DELETE CONSUMO;

DROP TABLE CONSUMO;

DELETE VEHICULO;

DROP TABLE VEHICULO;

DELETE CLASE_PRACTICA;

DROP TABLE CLASE_PRACTICA;

DELETE LUJO;

DROP TABLE LUJO;

DELETE EXAMEN;

DROP TABLE EXAMEN;

DELETE TRAMITACION_DOCUMENTOS;

DROP TABLE TRAMITACION_DOCUMENTOS;

```

DELETE PAGO;

DROP TABLE PAGO;

DELETE MATRICULA;

DROP TABLE MATRICULA;

DELETE COSTE_MATRICULACION;

DROP TABLE COSTE_MATRICULACION;

DELETE DOCENCIA;

DROP TABLE DOCENCIA;

DELETE PROFESOR;

DROP TABLE PROFESOR;

DELETE ALUMNO;

DROP TABLE ALUMNO;

DELETE COSTE_ENSENIANZA_TEORICA;

DROP TABLE COSTE_ENSENIANZA_TEORICA;

DELETE LICENCIA;

DROP TABLE LICENCIA;

DELETE COSTE_TRAMITACION;

DROP TABLE COSTE_TRAMITACION;

/* ULTIMA REVISION 15/6/99. SIN PROBLEMAS */

CREATE OR REPLACE PROCEDURE gastoalumno(gasto IN NUMBER ) AS

/* zona de declaracion de variables */

/* este cursor contiene el conjunto de los gastos en litros para cada alumno */

CURSOR al_litros IS

SELECT m.dni_al,c.litros

FROM matricula m,clase_practica cp, consumo c

WHERE m.n_matricula = cp.n_matricula AND

```

```

cp.n_matricula = c.n_matricula AND

cp.fecha_practica = c.fecha_practica;

/* esta tabla la usamos para recoger los resultados y agrupar los
litros gastados por cada alumno */

TYPE t_litros IS TABLE OF al_litros%ROWTYPE INDEX BY BINARY_INTEGER;

t_al_litros t_litros;

/* Esta variable recogerá el total de litros de todos los consumos */

v_total_litros NUMBER (10) := 0;

/* variable auxiliar que recoge las extracciones del cursor */

auxiliar al_litros%ROWTYPE;

encontrado BOOLEAN := FALSE;

/* esta excepcion captura el error que se produce cuando no hay
gastos de combustible en la base de datos */

no_hay_gastos EXCEPTION;

BEGIN

/* Calculamos el total */

SELECT sum(c.litros)

INTO v_total_litros

FROM consumo c ;

IF SQL%NOTFOUND THEN

RAISE no_hay_gastos;

END IF;

/* realizamos el calculo del tanto por ciento correspondiente */

v_total_litros := v_total_litros * gasto;

OPEN al_litros;

/* mientras en la iteracion anterior se saco una fila */

```

```

FETCH al_litros INTO auxiliar;

WHILE al_litros%FOUND AND NOT (al_litros%FOUND IS NULL) LOOP

IF t_al_litros.EXISTS(auxiliar.dni_al) THEN

t_al_litros(auxiliar.dni_al).litros := t_al_litros(auxiliar.dni_al).litros + auxiliar.litros;

ELSE

t_al_litros(auxiliar.dni_al) := auxiliar;

END IF;

FETCH al_litros INTO auxiliar;

END LOOP;

CLOSE al_litros;

/* ya tenemos el calculo en la variable de tabla t_al_litros */

/* mostramos en pantalla los alumnos que superan el
tanto por ciento indicado en el parametro formal */

FOR v_contador IN t_al_litros.FIRST .. t_al_litros.LAST LOOP

IF t_al_litros(v_contador).litros > v_total_litros THEN

DBMS_OUTPUT.PUT_LINE ( 'El alumno con dni ' || t_al_litros(v_contador).dni_al
|| ' ha superado el gasto indicado de'
|| v_total_litros ||' . Su gasto total ha sido de '|| t_al_litros(v.contador).litros);

END IF;

END LOOP;

/* Zona de EXCEPCIONES */

EXCEPTION

WHEN no_hay_gastos THEN

DBMS_OUTPUT.PUT_LINE('No hay gastos de combustible almacenados');

END;

/* ULTIMA REVISION 16/6/99*/

```

```

CREATE OR REPLACE PROCEDURE crearalumno( dni IN alumno.dni_al%TYPE,
nombre IN alumno.nombre%TYPE,
apellido_1 IN alumno.apellido_1%TYPE,
apellido_2 IN alumno.apellido_2%TYPE,
direccion IN alumno.direccion%TYPE,
telefono IN alumno.telefono%TYPE,
licencia_anterior IN alumno.tipo_lic%TYPE,
licencia_matriculacion IN matricula.tipo_lic%TYPE
) IS

numero_matricula BINARY_INTEGER:=0; /*recogera el numero de matricula mas alto en la tabla matricula
*/

v_docencia docencia%ROWTYPE; /*recoge a la docencia seleccionada*/

al_prev_mat EXCEPTION; /* captura el error de que ya haya un alumno igual en la base de datos*/

lic_inex EXCEPTION; /* captura el error de solicitar una licencia que no existe */

lic_ant_inex EXCEPTION; /* la licencia anterior introducida no es valida */

basura VARCHAR(50);

CURSOR c_alumno (dni_alumno alumno.dni_al%TYPE) IS

SELECT a.dni_al

FROM alumno a

WHERE a.dni_al = dni_alumno; /*este ultimo es un parametro de entrada */

CURSOR c_licencia (tipo_licencia licencia.tipo_lic%TYPE) IS

SELECT l.tipo_lic

FROM licencia l

WHERE l.tipo_lic = tipo_licencia;

CURSOR c_max_matricula IS

SELECT max(m.n_matricula)

FROM matricula m;

```

```

CURSOR c_matricula_alumno (dni_alumno alumno.dni_al%TYPE,
licencia_matriculacion licencia.tipo_lic%TYPE) IS

SELECT m.tipo_lic

FROM matricula m

WHERE m.dni_al = dni AND /* buscamos una matricula asociada al alumno */

m.tipo_lic = licencia_matriculacion; /* y en la licencia que queremos matricularlo */

/* INICIO DE LA DECLARACION DEL SUBPROCEDIMIENTO CALCULAR_DOCENCIA */

FUNCTION calcular_docencia(licencia_seleccionada IN matricula.tipo_lic%TYPE,
docencia_seleccionada OUT docencia%ROWTYPE

) RETURN BOOLEAN IS

/* Este procedimiento devuelve la docencia que tiene menos matriculados en la licencia
que se especifica en el argumento de la funcion*/

/* DECLARACION DE VARIABLES*/

/* Este cursor apunta a una sentencia select que devuelve el numero de matriculados
de todas las docencias que den clase de la licencia especificada en el argumento
de la funcion */

CURSOR c_docencia (lm matricula.tipo_lic%TYPE) IS

SELECT d.dni_prof,d.tipo_lic,d.matricula,count(m.n_matricula) recuento

FROM docencia d,matricula m

WHERE d.dni_prof = m.dni_prof AND

m.tipo_lic = d.tipo_lic AND

d.tipo_lic = lm

GROUP BY d.dni_prof,d.tipo_lic,d.matricula;

/* Esta variable contendra el minimo obtenido*/

minimo_actual c_docencia%ROWTYPE;

/* Esta variable sirve para recoger las tuplas extraidas del cursor */

```

```

auxiliar c_docencia%ROWTYPE;

/* INICIO*/

BEGIN

/* vamos a comprobar primero si el conjunto activo del cursor
tiene al menos una tupla. En caso de que no, esto quiere decir
que se introdujo una licencia de la que no se da clase en
la autoescuela */

minimo_actual.recuento := 1000;

OPEN c_docencia(licencia_seleccionada);

FETCH c_docencia INTO auxiliar;

IF c_docencia%NOTFOUND THEN /*si se realizo una extraccion y no hubo tupla extraida */

CLOSE c_docencia;

docencia_seleccionada.dni_prof := NULL;

docencia_seleccionada.tipo_lic := NULL;

docencia_seleccionada.matricula := NULL;

RETURN FALSE; /* devolvemos FALSE como respuesta */

END IF;

/* Como en este caso hay tuplas en el cursor */

CLOSE c_docencia;

OPEN c_docencia(licencia_seleccionada);

LOOP

FETCH c_docencia INTO auxiliar;

EXIT WHEN c_docencia%NOTFOUND;

/* si no encontramos datos nos salimos */

IF minimo_actual.recuento >= auxiliar.recuento THEN

/* si encontramos alguno menor */

```

```

minimo_actual := auxiliar;

/* actualizamos el minimo */

END IF;

END LOOP;

CLOSE c_docencia;

docencia_seleccionada.dni_prof := minimo_actual.dni_prof;

docencia_seleccionada.tipo_lic := minimo_actual.tipo_lic;

docencia_seleccionada.matricula := minimo_actual.matricula;

RETURN TRUE; /*hemos encontrado el minimo. La funcion ha tenido exito */

END calcular_docencia;

/* FIN DE LA DEFINICION DEL SUBPROCEDIMIENTO CALCULAR_DOCENCIA*/

/* Inicio procedimiento principal */

BEGIN

/* primero vamos a comprobar que el alumno no esta ya registrado en la base de datos*/

OPEN c_alumno(dni);

FETCH c_alumno INTO basura;

IF c_alumno%NOTFOUND THEN /* Usamos el cursor para detectar si hubo filas */

/* Estamos en el caso de que el alumno no existe */

/* calculemos ahora la docencia que se encargara de sus clases */

IF NOT calcular_docencia(licencia_matriculacion,v_docencia) THEN

/* Si la licencia no existe */

RAISE lic_inex;

ELSE

/* Si la licencia existe */

/*ahora tenemos que saber si tiene una licencia anterior y cual es esta*/

IF licencia_anterior IS NULL THEN /*no hay licencia anterior*/

```

```

/* luego tenemos que asociarle el coste de enseñanza teorica */

INSERT INTO alumno VALUES

(dni,nombre,apellido_1,apellido_2,direccion,telefono,NULL,'coste_e_t');

ELSE

/* vamos a ver si la licencia anterior es valida */

OPEN c_licencia(licencia_anterior);

FETCH c_licencia INTO basura;

IF c_licencia%NOTFOUND THEN

/* la licencia anterior no existe */

CLOSE c_licencia;

RAISE lic_ant_inex;

ELSE

/* la licencia anterior existe */

/* como tiene una licencia anterior,se la asociamos */

INSERT INTO alumno VALUES

(dni,nombre,apellido_1,apellido_2,direccion,telefono,licencia_anterior,NULL);

CLOSE c_licencia;

END IF;

END IF;

/* Una vez resulelto esto hay que realizar una entrada en la tabla matricula */

/* Vamos a calcular el numero mas alto del atributo n_matricula para asociarlo a la nueva
matricula */

OPEN c_max_matricula;

FETCH c_max_matricula INTO numero_matricula;

IF c_max_matricula%NOTFOUND THEN

/* no hay matriculas */

```

```

numero_matricula := 1;

CLOSE c_max_matricula;

ELSE

/* incrementamos el numero de matricula */

numero_matricula := numero_matricula +1;

/* creamos una ocurrencia en la tabla matricula para matricular al alumno */

INSERT INTO matricula VALUES

(numero_matricula,0,dni,v_docencia.dni_prof,

v_docencia.tipo_lic,'coste_mat');

CLOSE c_max_matricula;

END IF;

END IF;

CLOSE c_alumno;

ELSE

/* Estamos en el caso de que el alumno ya esta matriculado en alguna licencia*/

/* Habra que ver si ya esta matriculado en la licencia de la que quiere matricularse */

OPEN c_matricula_alumno(dni,licencia_matriculacion);

FETCH c_matricula_alumno INTO basura;

IF c_matricula_alumno%NOTFOUND THEN /* si no encontramos respuesta, es que no esta matriculado */

/* en la licencia de la que ahora vamos a matricularle */

IF NOT calcular_docencia(licencia_matriculacion,v_docencia) THEN

dbms_output.put_line( v_docencia.dni_prof || v_docencia.tipo_lic);

/* no existe la licencia de la que vamos a matricularlo */

RAISE lic_inex;

ELSE

/* si existe*/

```

```

/* procedemos a la matriculacion */

OPEN c_max_matricula;

FETCH c_max_matricula INTO numero_matricula;

IF c_max_matricula%NOTFOUND THEN

numero_matricula :=1;

ELSE

numero_matricula:= numero_matricula +1;

END IF;

/* matriculamos */

INSERT INTO matricula VALUES

(numero_matricula,0,dni,v_docencia.dni_prof,v_docencia.tipo_lic,'coste_mat');

END IF;

ELSE

/* Como la sentencia select ha encontrado al alumno matriculado en la licencia de la
que queriamos matricularle, levantaremos una excepcion */

RAISE al_prev_mat;

END IF;

CLOSE c_matricula_alumno;

END IF;

EXCEPTION

WHEN lic_inex THEN

DBMS_OUTPUT.PUT_LINE('La licencia introducida no tiene profesor asociado o no existe');

WHEN al_prev_mat THEN

DBMS_OUTPUT.PUT_LINE('el alumno ya esta matriculado de esa licencia');

WHEN lic_ant_inex THEN

DBMS_OUTPUT.PUT_LINE('La licencia anterior es inválida');

```

```

END crearalumno;

/* ULTIMA REVISION 16/6/99 */

CREATE OR REPLACE PROCEDURE inserta_examen(dni_alumno IN alumno.dni_al%TYPE,
licencia_examen IN licencia.tipo_lic%TYPE,
fecha_examen IN examen.fecha_examen%TYPE,
tipo_ex IN examen.tipo_examen%TYPE,
err_teorico IN examen.errores%TYPE,
dur_practico IN examen.duracion%TYPE) AS
t_a tramitacion_documentos%ROWTYPE;
no_hallada EXCEPTION;

/* Este subprocedimiento busca una tramitacion de documentos para un alumno en una
licencia concreta, devolviendo FALSE si no existe una tramitacion o si existe y ademas
tiene 3 examenes asociados, con lo que no podemos asociar un nuevo examen. En
caso de que exista una tramitacion adecuada, se devuelve en el parametro de salida
tram_correcta */

FUNCTION hallar_tramitacion(
dni IN alumno.dni_al%TYPE,
licencia_examen IN licencia.tipo_lic%TYPE,
tram_correcta OUT tramitacion_documentos %ROWTYPE
) RETURN BOOLEAN IS

/* Este cursor continene una tabla en la que se muestra para un alumno y en una licencia
concreta, y para cada tramitacion de documentos que realiza el numero de examenes
asociado. De esta manera recorriendo el cursor, podemos hallar la tramitacion mas
actual recorriendo el cursor y comprobar si ya tiene agotadas las tres convocatorias
pertinentes */

CURSOR tramitacion (dni1 alumno.dni_al%TYPE,licencia_examen1 licencia.tipo_lic%TYPE) IS

```

```

SELECT m.dni_al,m.n_matricula,td.fecha_tramitacion,count(e.fecha_examen) recuento
FROM matricula m,tramitacion_documentos td,examen e
WHERE m.dni_al = dni1 AND m.tipo_lic = licencia_examen1 AND
m.n_matricula = td.n_matricula AND
e.fecha_tramitacion = td.fecha_tramitacion AND
e.n_matricula = td.n_matricula
GROUP BY m.dni_al,m.n_matricula,td.fecha_tramitacion;

/* Aqui almacenamos las tuplas que leemos del cursor superior */

CURSOR tram_examen (dni1 alumno.dni_al%TYPE,licencia_examen1 licencia.tipo_lic%TYPE) IS
SELECT max(td.fecha_tramitacion) fecha
FROM tramitacion_documentos td,matricula m
WHERE m.dni_al = dni1 AND m.tipo_lic = licencia_examen1 AND
m.n_matricula = td.n_matricula;

r_tram_examen tram_examen%ROWTYPE;

r_tramitacion tramitacion%ROWTYPE;

tramitacion_tardia tramitacion%ROWTYPE;

BEGIN

OPEN tramitacion(dni,licencia_examen);

FETCH tramitacion INTO tramitacion_tardia;

IF tramitacion%NOTFOUND THEN

/* El cursor esta vacio. no hay tramitaciones asociadas con el alumno
en esa matricula*/

CLOSE tramitacion;

RETURN FALSE;

END IF;

/* llegados a este punto si que existen tramitaciones de documentos */

```

```

CLOSE tramitacion;

OPEN tramitacion (dni,licencia_examen);

LOOP

FETCH tramitacion INTO r_tramitacion;

IF tramitacion%NOTFOUND THEN

CLOSE tramitacion;

EXIT;

END IF;

/* vamos a procesar el cursor */

/* vamos a calcular la tramitacion mas tardia */

IF tramitacion_tardia.fecha_tramitacion <= r_tramitacion.fecha_tramitacion THEN

tramitacion_tardia := r_tramitacion;

END IF;

END LOOP;

/* ahora tenemos en tramitacion_tardia la tramitacion mas actual

ahora lo que tenemos que hacer es ver si la tramitacion tiene

asociadas tres examenes, con lo que no podria ser considerado un

nuevo examen */

OPEN tram_examen (dni,licencia_examen);

FETCH tram_examen INTO r_tram_examen;

CLOSE tram_examen;

IF tramitacion_tardia.fecha_tramitacion = r_tram_examen.fecha THEN

IF tramitacion_tardia.recuento < 3 THEN

/* Pueden asociarse mas examenes */

tram_correcta.n_matricula := tramitacion_tardia.n_matricula;

tram_correcta.fecha_tramitacion := tramitacion_tardia.fecha_tramitacion;

```

```

tram_correcta.coste_tram := 'coste_tram';

ELSE

/* No pueden asociarse mas examenes */

RETURN FALSE;

END IF;

ELSE

tram_correcta.fecha_tramitacion := r_tram_examen.fecha;

tram_correcta.n_matricula := tramitacion_tardia.n_matricula;

tram_correcta.coste_tram := 'coste_tram';

END IF;

RETURN TRUE;

END hallar_tramitacion ;

/* COMENZAMOS LA SECCION EJECUTABLE DEL BLOQUE */

BEGIN

/* Buscamos una tramitacion */

IF hallar_tramitacion(dni_alumno,licencia_examen,t_a) THEN

/* hay una tramitacion en la que podemos hacer un nuevo examen */

IF tipo_ex = 'T' THEN

/*es teorico*/

INSERT INTO examen VALUES

( fecha_examen,t_a.n_matricula,t_a.fecha_tramitacion,'T',NULL,err_teorico);

ELSE

/*es practico*/

INSERT INTO examen VALUES

( fecha_examen,t_a.n_matricula,t_a.fecha_tramitacion,'P',dur_practico,NULL);

END IF;

```

```

ELSE

/* No se hallo la tramitacion asociada */

RAISE no_hallada;

END IF;

/* SECCION DE EXCEPCIONES */

EXCEPTION

WHEN no_hallada THEN

DBMS_OUTPUT.PUT_LINE(' No se encontro una tramitacion con menos de 3 examenes ');

END inserta_examen;

CREATE TABLE Licencia

(

tipo_lic VARCHAR(3) NOT NULL,

CONSTRAINT pk_Licencia

PRIMARY KEY (tipo_lic)

);

CREATE TABLE Documentacion

(

nombre_doc VARCHAR(30) NOT NULL,

CONSTRAINT pk_Documentacion

PRIMARY KEY (nombre_doc)

);

CREATE TABLE Coste_Ensenianza_Teorica

(

coste_et VARCHAR(10) NOT NULL,

importe NUMBER(4) NOT NULL,

CONSTRAINT pk_Coste_Ensenianza_Teorica

```

```

PRIMARY KEY (coste_et)

);

CREATE TABLE Coste_Matriculacion

(

coste_mat VARCHAR(14) NOT NULL,

importe NUMBER(4) NOT NULL,

CONSTRAINT pk_Coste_Matriculacion

PRIMARY KEY (coste_mat)

);

CREATE TABLE Lujo

(

tipo_lujo VARCHAR(10),

tarifa_adicional NUMBER(4),

CONSTRAINT pk_Lujo

PRIMARY KEY (tipo_lujo)

);

CREATE TABLE Coste_Tramitacion

(

coste_tram VARCHAR(10) NOT NULL,

importe NUMBER(4),

CONSTRAINT pk_Coste_Tramitacion

PRIMARY KEY (coste_tram)

);

CREATE TABLE Profesor

(

dni_prof NUMBER(4) NOT NULL,

```

```

nombre VARCHAR(10) NOT NULL,
apellido_1 VARCHAR(10) NOT NULL,
apellido_2 VARCHAR(10) NOT NULL,
direccion VARCHAR(20),
telefono NUMBER(9),
CONSTRAINT pk_Profesor
PRIMARY KEY (dni_prof)
);

CREATE TABLE Alumno
(
dni_al NUMBER(4) NOT NULL ,
nombre VARCHAR(10) NOT NULL ,
apellido_1 VARCHAR(10) NOT NULL ,
apellido_2 VARCHAR(10) NOT NULL ,
direccion VARCHAR(20) ,
telefono NUMBER(9) ,
tipo_lic VARCHAR(3) ,
coste_et VARCHAR(10) ,
CONSTRAINT pk_Alumno
PRIMARY KEY (dni_al) ,
CONSTRAINT fk_Licencia
FOREIGN KEY (tipo_lic)
REFERENCES Licencia(tipo_lic) ,
CONSTRAINT fk_Coste_Ensenianza_Teorica
FOREIGN KEY (coste_et)
REFERENCES Coste_Ensenianza_Teorica(coste_et),

```

```

CONSTRAINT ck_exclusiva_lic_cet

CHECK (

( (NOT(coste_et IS NULL)) AND (tipo_lic IS NULL) )

OR

( (coste_et IS NULL) AND (NOT(tipo_lic IS NULL)) )

)

);

```

```

CREATE TABLE Docencia

(

dni_prof NUMBER(4) NOT NULL,

tipo_lic VARCHAR(3) NOT NULL,

CONSTRAINT pk_Docencia

PRIMARY KEY (dni_prof,tipo_lic),

CONSTRAINT fk_Profesor

FOREIGN KEY (dni_prof)

REFERENCES Profesor(dni_prof)

ON DELETE CASCADE,

CONSTRAINT fk_tipo_lic

FOREIGN KEY (tipo_lic)

REFERENCES Licencia(tipo_lic)

ON DELETE CASCADE

);

```

```

CREATE TABLE Matricula

(

n_matricula NUMBER(10) NOT NULL,

doc_presente NUMBER(1) NOT NULL,

```

```

dni_al NUMBER(4) NOT NULL,

dni_prof NUMBER(4) ,

tipo_lic VARCHAR(3) ,

coste_mat VARCHAR(14) NOT NULL,

CONSTRAINT pk_Matricula

PRIMARY KEY (n_matricula),

CONSTRAINT fk_Alumno

FOREIGN KEY (dni_al)

REFERENCES Alumno(dni_al),

CONSTRAINT fk_Docencia

FOREIGN KEY (tipo_lic,dni_prof)

REFERENCES Docencia(tipo_lic,dni_prof),

CONSTRAINT fk_Coste_Matriculacion

FOREIGN KEY (coste_mat)

REFERENCES Coste_Matriculacion(coste_mat),

CONSTRAINT ck_doc_presente

CHECK ( ((doc_presente <= 1)

OR (doc_presente > 0))

)

);

CREATE TABLE Prueba_Teorica

(

fecha_prueba DATE NOT NULL,

N_matricula NUMBER(10) NOT NULL,

N_test NUMBER(4) NOT NULL,

errores NUMBER(2) NOT NULL,

```

```

CONSTRAINT pk_Prueba_Teorica
PRIMARY KEY (fecha_prueba,N_matricula),
CONSTRAINT fk_Matricula
FOREIGN KEY (N_matricula)
REFERENCES Matricula(N_matricula)
ON DELETE CASCADE
);

```

```

CREATE TABLE Pago
(
  N_pago NUMBER(4) NOT NULL,
  N_matricula NUMBER(10) NOT NULL,
  importe NUMBER(6) NOT NULL,
CONSTRAINT pk_Pago
PRIMARY KEY (N_pago,N_matricula),
CONSTRAINT ck_importe
CHECK ( importe > 0 ),
CONSTRAINT fk_Mat
FOREIGN KEY (N_matricula)
REFERENCES Matricula(N_matricula)
ON DELETE CASCADE
);

```

```

CREATE TABLE Clase_Practica
(
  fecha_practica DATE NOT NULL,
  N_matricula NUMBER(10) NOT NULL,
  duracion NUMBER(2) NOT NULL,

```

```

objetivos VARCHAR(50),

resultados VARCHAR(50),

CONSTRAINT pk_Clase_Practica

PRIMARY KEY (fecha_practica,N_matricula),

CONSTRAINT fk_Matric

FOREIGN KEY (N_matricula)

REFERENCES Matricula(n_matricula)

ON DELETE CASCADE,

CONSTRAINT ck_duracion

CHECK ( duracion < 10)

);

CREATE TABLE Tramitacion_Documentos

(

fecha_tramitacion DATE NOT NULL,

N_matricula NUMBER(10) NOT NULL,

coste_tram VARCHAR(10) NOT NULL,

CONSTRAINT pk_Tramitacion_Documentos

PRIMARY KEY (fecha_tramitacion,N_matricula),

CONSTRAINT fk_Matr

FOREIGN KEY (N_matricula)

REFERENCES Matricula(N_matricula)

ON DELETE CASCADE,

CONSTRAINT fk_coste_tram

FOREIGN KEY (coste_tram)

REFERENCES coste_tramitacion(coste_tram)

);

```

```

CREATE TABLE Examen
(
    fecha_examen DATE NOT NULL,
    N_matricula NUMBER(10) NOT NULL,
    fecha_tramitacion DATE NOT NULL,
    tipo_examen VARCHAR(1) NOT NULL,
    duracion NUMBER(2),
    errores NUMBER(2),
    CONSTRAINT pk_Examen
PRIMARY KEY (fecha_examen,
    N_matricula, fecha_tramitacion),
    CONSTRAINT ck_tipo_examen
CHECK (tipo_examen IN ('T','P')),
    CONSTRAINT ck_valor_error_duracion
CHECK (
    ( (tipo_examen = 'T') AND (errores <> NULL)
    AND (duracion = NULL) )
    OR
    ( (tipo_examen = 'P') AND (duracion <> NULL)
    AND (errores = NULL) )
    ),
    CONSTRAINT fk_Tramitacion_Documentos
FOREIGN KEY (fecha_tramitacion, N_matricula)
REFERENCES Tramitacion_Documentos(
    fecha_tramitacion, N_matricula)
ON DELETE CASCADE

```

```

);

CREATE TABLE Vehiculo

(

matricula VARCHAR(6) NOT NULL,

marca VARCHAR(10) NOT NULL,

modelo VARCHAR(10) NOT NULL,

cilindrada NUMBER(4) NOT NULL,

anio_compra DATE NOT NULL,

tarifa NUMBER(4) NOT NULL,

tipo_lic VARCHAR(3),

dni_prof NUMBER(4),

tipo_lujo VARCHAR(10) NOT NULL,

licencia_necesaria VARCHAR(3) NOT NULL,

CONSTRAINT pk_Vehiculo

PRIMARY KEY (matricula),

CONSTRAINT ck_Licencia_Vehiculo

CHECK (tipo_lic = licencia_necesaria ),

CONSTRAINT sk_Vehiculo

UNIQUE (dni_prof,tipo_lic),

CONSTRAINT fk_Docen

FOREIGN KEY (dni_prof,tipo_lic)

REFERENCES Docencia(dni_prof,tipo_lic),

CONSTRAINT fk_Licenc

FOREIGN KEY (licencia_necesaria)

REFERENCES Licencia(tipo_lic),

CONSTRAINT fk_Lujo

```

FOREIGN KEY (tipo_lujo)

REFERENCES Lujo(tipo_lujo)

);

ALTER TABLE Docencia

ADD (matricula VARCHAR(6));

ALTER TABLE Docencia

ADD CONSTRAINT sk_docencia

UNIQUE (matricula)

;

ALTER TABLE Docencia

ADD CONSTRAINT fk_vehiculo

FOREIGN KEY (matricula)

REFERENCES vehiculo(matricula)

;

ALTER TABLE Docencia

ADD CONSTRAINT ck_docencia

check (matricula <> NULL)

;

CREATE TABLE Consumo

(

matricula VARCHAR(6) NOT NULL,

N_matricula NUMBER(10) NOT NULL ,

fecha_practica DATE NOT NULL,

kilometros NUMBER(3) NOT NULL,

litros NUMBER(2) NOT NULL,

CONSTRAINT pk_Consumo

PRIMARY KEY (matricula,N_matricula,fecha_practica),

CONSTRAINT fk_Clase_Practica

FOREIGN KEY (fecha_practica,N_matricula)

REFERENCES Clase_Practica

(fecha_practica,N_matricula)

ON DELETE CASCADE,

CONSTRAINT fk_Vehic

FOREIGN KEY (matricula)

REFERENCES Vehiculo(matricula)

ON DELETE CASCADE

);

CREATE TABLE D_L

(

tipo_lic VARCHAR(3) NOT NULL,

nombre_doc VARCHAR(30) NOT NULL,

CONSTRAINT pk_D_L

PRIMARY KEY (tipo_lic,nombre_doc),

CONSTRAINT fk_Lic

FOREIGN KEY (tipo_lic)

REFERENCES Licencia(tipo_lic)

ON DELETE CASCADE,

CONSTRAINT fk_Documentacion

FOREIGN KEY (nombre_doc)

REFERENCES Documentacion(nombre_doc)

ON DELETE CASCADE

);

INSERT INTO Licencia

VALUES

('A');

INSERT INTO Licencia

VALUES ('B');

INSERT INTO Licencia

VALUES ('C');

INSERT INTO Documentacion

VALUES ('DNI');

INSERT INTO Documentacion

VALUES ('CERT MEDICO');

INSERT INTO Documentacion

VALUES ('CERT PSICOL');

INSERT INTO Documentacion

VALUES ('LICENCIA A');

INSERT INTO Documentacion

VALUES ('LICENCIA B');

INSERT INTO Documentacion

VALUES ('LICENCIA C');

INSERT INTO Documentacion

VALUES ('CERT DGT');

INSERT INTO Profesor

VALUES (1,'Rafael','Andino','Pozo','Onieva 3',00001);

INSERT INTO Profesor

VALUES (2,'Ana','Ruiz','Romero','Garberas 16',00002);

INSERT INTO Profesor

```

VALUES (3,'Carolina','Arredondo','Ruiz','Liberacion 3',00003);

INSERT INTO Profesor

VALUES (4,'Nestor','Arredondo','Martinez','Liberacion 3',0004);

ALTER TABLE DOCENCIA

DISABLE CONSTRAINT FK_VEHICULO;

INSERT INTO Docencia

VALUES (1,'A','000005');

INSERT INTO Docencia

VALUES (2,'A','000006');

INSERT INTO Docencia

VALUES (2,'B','000001');

INSERT INTO Docencia

VALUES (3,'B','000002');

INSERT INTO Docencia

VALUES (3,'C','000003');

INSERT INTO Docencia

VALUES (4,'C','000004');

INSERT INTO Coste_Ensenianza_Teorica

VALUES ('coste_e_t',4500);

INSERT INTO Coste_Matriculacion

VALUES ('coste_mat',3000);

INSERT INTO Coste_Tramitacion

VALUES ('coste_tram',3000);

INSERT INTO Lujo

VALUES ('normal',500);

INSERT INTO Lujo

```

```

VALUES ('extra',1000);

INSERT INTO Lujo

VALUES ('ejecutivo',2000);

INSERT INTO Lujo

VALUES ('exclusivo',5000);

INSERT INTO Vehiculo

VALUES ('000001','Ford','Escort',2000,TO_DATE('21/3/99','DD/MM/YY'),

1000,'B',2,'normal','B');

INSERT INTO Vehiculo

VALUES ('000002','Wolkswagen','Golf TDI',1900,TO_DATE('31/3/95',

'DD/MM,YY'),1000,'B',3,'extra','B');

INSERT INTO Vehiculo

VALUES ('000003','Scania','Globe',4500,TO_DATE('7/7/97',

'DD/MM/YY'),3000,'C',3,'normal','C');

INSERT INTO Vehiculo

VALUES ('000004','Scania','RoadRun',6000,TO_DATE('1/6/99',

'DD/MM/YY'),3000,'C',4,'ejecutivo','C');

INSERT INTO Vehiculo

VALUES ('000005','Suzuki','MRZ',500,TO_DATE('27/8/98','DD/MM/YY'),

700,'A',1,'exclusivo','A');

INSERT INTO Vehiculo

VALUES ('000006','Honda','MRV',500,TO_DATE('5/2/95','DD/MM/YY'),

700,'A',2,'normal','A');

ALTER TABLE DOCENCIA

ENABLE CONSTRAINT FK_VEHICULO;

INSERT INTO Alumno

```

```

VALUES (1,'Jose','Garcia','Perez','Liberacion 4',1,'A',NULL);

INSERT INTO Alumno

VALUES (2,'Ana','Belen','Ruiz','Plaza 1',2,'B',NULL);

INSERT INTO Alumno

VALUES (3,'Elisa','Roca','Perez','Plaza 3',3,NULL,'coste_e_t');

INSERT INTO Alumno

VALUES (4,'Petra','Garcia','Obregon','Orgullo 1',4,'B',NULL);

INSERT INTO Alumno

VALUES (5,'Pepa','Martos','Garcia','Canto 1',5,'A',NULL);

INSERT INTO Alumno

VALUES (6,'Antonio','Molina','Mora','Campillo 7',6,'B',NULL);

INSERT INTO Alumno

VALUES (7,'Elisabeth','Ruiz','Romero','Garberas 1',7,NULL,'coste_e_t');

INSERT INTO Alumno

VALUES (8,'Juan','Martinez','Fontiveros','Amador 1',8,NULL,'coste_e_t');

INSERT INTO Alumno

VALUES (9,'Susana','Ruiz','Pozo','Amador 2',9,NULL,'coste_e_t');

INSERT INTO Alumno

VALUES (10,'Pepe','Barba','Martinez','Prado 9',10,'B',NULL);

INSERT INTO Matricula

VALUES (0,0,1,2,'B','coste_mat');

INSERT INTO Matricula

VALUES (1,0,2,3,'C','coste_mat');

INSERT INTO Matricula

VALUES (2,0,3,1,'A','coste_mat');

INSERT INTO Matricula

```

```

VALUES (3,0,4,4,'C','coste_mat');

INSERT INTO Matricula

VALUES (4,0,5,2,'B','coste_mat');

INSERT INTO Matricula

VALUES (5,0,5,4,'C','coste_mat');

INSERT INTO Matricula

VALUES (6,0,6,4,'C','coste_mat');

INSERT INTO Matricula

VALUES (7,0,7,1,'A','coste_mat');

INSERT INTO Matricula

VALUES (8,0,8,2,'A','coste_mat');

INSERT INTO Matricula

VALUES (9,0,8,3,'B','coste_mat');

INSERT INTO Matricula

VALUES (10,0,8,4,'C','coste_mat');

INSERT INTO Matricula

VALUES (11,0,9,2,'A','coste_mat');

INSERT INTO Matricula

VALUES (12,0,9,2,'B','coste_mat');

INSERT INTO Matricula

VALUES (13,0,10,4,'C','coste_mat');

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('1/2/99','DD/MM/YY'),1,1,4);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('3/4/98','DD/MM/YY'),1,2,3);

INSERT INTO Prueba_Teorica

```

```

VALUES (TO_DATE('2/4/99','DD/MM/YY'),3,4,1);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('3/4/98','DD/MM/YY'),5,1,3);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('5/5/98','DD/MM/YY'),7,5,2);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('13/3/98','DD/MM/YY'),1,5,6);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('4/4/98','DD/MM/YY'),9,9,6);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('2/3/98','DD/MM/YY'),6,7,8);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('5/7/97','DD/MM/YY'),3,4,5);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('3/6/98','DD/MM/YY'),3,5,6);

INSERT INTO Prueba_Teorica

VALUES (TO_DATE('25/6/98','DD/MM/YY'),7,2,1);

INSERT INTO Pago

VALUES (1,1,1000);

INSERT INTO Pago

VALUES (2,1,20000);

INSERT INTO Pago

VALUES (1,3,10000);

INSERT INTO Pago

VALUES (1,4,10000);

INSERT INTO Pago

```

```

VALUES (2,4,5000);

INSERT INTO Pago

VALUES (1,5,1000);

INSERT INTO Pago

VALUES (1,6,1000);

INSERT INTO Clase_Practica

VALUES (TO_DATE('2/2/99','DD/MM/YY'),1,2,'rampas','mejorar');

INSERT INTO Consumo

VALUES ('000003',1,TO_DATE('2/2/99','DD/MM/YY'),10,1);

INSERT INTO Clase_Practica

VALUES (TO_DATE('2/2/99','DD/MM/YY'),2,1,'reduccion marchas',
'mejorar');

INSERT INTO Consumo

VALUES ('000005',2,TO_DATE('2/2/99','DD/MM/YY'),5,1);

INSERT INTO Clase_Practica

VALUES (TO_DATE('1/2/99','DD/MM/YY'),1,1,'rampas','mejorar');

INSERT INTO Consumo

VALUES ('000003',1,TO_DATE('1/2/99','DD/MM/YY'),25,3);

INSERT INTO Clase_Practica

VALUES (TO_DATE('3/3/99','DD/MM/YY'),2,3,'rampas','mejorar');

INSERT INTO Consumo

VALUES ('000005',2,TO_DATE('3/3/99','DD/MM/YY'),15,3);

INSERT INTO Clase_Practica

VALUES (TO_DATE('4/12/98','DD/MM/YY'),2,2,'aparcas','bien');

INSERT INTO Consumo

VALUES ('000005',2,TO_DATE('4/12/98','DD/MM/YY'),20,4);

```

```

INSERT INTO Clase_Practica
VALUES (TO_DATE('5/12/98','DD/MM/YY'),3,1,'reduccion','bien');

INSERT INTO Consumo
VALUES ('000004',3,TO_DATE('5/12/98','DD/MM/YY'),30,10);

INSERT INTO Clase_Practica
VALUES (TO_DATE('5/11/98','DD/MM/YY'),4,1,'aparcar','mejorar');

INSERT INTO Consumo
VALUES ('000001',4,TO_DATE('5/11/98','DD/MM/YY'),40,10);

INSERT INTO Clase_Practica
VALUES (TO_DATE('6/12/98','DD/MM/YY'),4,1,'espejos','bien');

INSERT INTO Consumo
VALUES ('000001',4,TO_DATE('6/12/98','DD/MM/YY'),30,8);

INSERT INTO Clase_Practica
VALUES (TO_DATE('6/12/98','DD/MM/YY'),5,2,'giros','bien');

INSERT INTO Consumo
VALUES ('000004',5,TO_DATE('6/12/98','DD/MM/YY'),15,2);

INSERT INTO Clase_Practica
VALUES (TO_DATE('7/7/99','DD/MM/YY'),6,1,'rampas','bien');

INSERT INTO Consumo
VALUES ('000004',6,TO_DATE('7/7/99','DD/MM/YY'),6,1);

INSERT INTO Clase_Practica
VALUES (TO_DATE('7/7/99','DD/MM/YY'),7,1,'señales','mejorar');

INSERT INTO Consumo
VALUES ('000005',7,TO_DATE('7/7/99','DD/MM/YY'),7,1);

INSERT INTO Clase_Practica
VALUES (TO_DATE('10/12/99','DD/MM/YY'),8,1,'señanles','mejorar');

```

```

INSERT INTO Consumo

VALUES ('000006',8,TO_DATE('10/12/99','DD/MM/YY'),8,1);

INSERT INTO Clase_Practica

VALUES (TO_DATE('11/12/99','DD/MM/YY'),8,2,'señanles','mejorar');

INSERT INTO Consumo

VALUES ('000006',8,TO_DATE('11/12/99','DD/MM/YY'),14,2);

INSERT INTO Clase_Practica

VALUES (TO_DATE('10/1/99','DD/MM/YY'),9,1,'señanles','mejorar');

INSERT INTO Consumo

VALUES ('000002',9,TO_DATE('10/1/99','DD/MM/YY'),9,1);

INSERT INTO Clase_Practica

VALUES (TO_DATE('10/2/99','DD/MM/YY'),10,3,'señanles','mejorar');

INSERT INTO Consumo

VALUES ('000004',10,TO_DATE('10/2/99','DD/MM/YY'),10,3);

INSERT INTO Clase_Practica

VALUES (TO_DATE('11/2/99','DD/MM/YY'),10,2,'señanles','mejorar');

INSERT INTO Consumo

VALUES ('000004',10,TO_DATE('11/2/99','DD/MM/YY'),20,2);

INSERT INTO Clase_Practica

VALUES (TO_DATE('10/12/99','DD/MM/YY'),11,1,'señanles','mejorar');

INSERT INTO Consumo

VALUES ('000006',11,TO_DATE('10/12/99','DD/MM/YY'),11,1);

INSERT INTO Clase_Practica

VALUES (TO_DATE('10/12/99','DD/MM/YY'),12,2,'señanles','mejorar');

INSERT INTO Consumo

VALUES ('000001',12,TO_DATE('10/12/99','DD/MM/YY'),12,2);

```

```

INSERT INTO Clase_Practica

VALUES (TO_DATE('10/12/99','DD/MM/YY'),13,3,'señales','mejorar');

INSERT INTO Consumo

VALUES ('000004',13,TO_DATE('10/12/99','DD/MM/YY'),13,3);

INSERT INTO Tramitacion_Documentos

VALUES (TO_DATE('2/2/99','DD/MM/YY'),2,'coste_tram');

INSERT INTO Examen

VALUES (TO_DATE('12/5/99','DD/MM/YY'),2,TO_DATE('2/2/99',
'DD/MM/YY'),'T',NULL,6);

INSERT INTO Examen

VALUES (TO_DATE('17/5/99','DD/MM/YY'),2,TO_DATE('2/2/99',
'DD/MM/YY'),'T',NULL,9);

INSERT INTO Tramitacion_Documentos

VALUES (TO_DATE('11/10/99','DD/MM/YY'),1,'coste_tram');

INSERT INTO Examen

VALUES (TO_DATE('1/6/99','DD/MM/YY'),1,TO_DATE('11/10/99',
'DD/MM/YY'),'T',NULL,0);

INSERT INTO Examen

VALUES (TO_DATE('3/6/99','DD/MM/YY'),1,TO_DATE('11/10/99',
'DD/MM/YY'),'P',1,NULL);

INSERT INTO Tramitacion_Documentos

VALUES (TO_DATE('19/5/99','DD/MM/YY'),6,'coste_tram');

INSERT INTO Examen

VALUES (TO_DATE('27/5/99','DD/MM/YY'),6,TO_DATE('19/5/99',
'DD/MM/YY'),'T',NULL,1);

INSERT INTO D_L

```

```

VALUES ('A','DNI');

INSERT INTO D_L

VALUES ('A','CERT MEDICO');

INSERT INTO D_L

VALUES ('B','LICENCIA A');

INSERT INTO D_L

VALUES ('B','DNI');

INSERT INTO D_L

VALUES ('B','CERT MEDICO');

INSERT INTO D_L

VALUES ('C','LICENCIA B');

INSERT INTO D_L

VALUES ('C','DNI');

INSERT INTO D_L

VALUES ('C','CERT MEDICO');

INSERT INTO D_L

VALUES ('C','CERT PSICOL');

```

Agenda Personal

tipo

Nombre

fecha

Nombre,tipo_fecha

Nombre,tipo_fecha

Calle,numero,C_P

alfanumerico

Tipo

alfanumerico

Tipo

Alfanumerico

Dni

Nombre

Cif

Nombre

Apuntes

Nombre

Nombre,alfanumerico

Nombre, alfanumerico

Nombre,alfanumerico

Alfanumerico,nombre,tipo_direccion

Apuntes

Nombre

Nonmbre,Nombre

C_P

localidad