

PLANTEAMIENTOS Y LOS SCRIPTS QUE LO RESUELVEN PARA AMBOS DBMS

1. Generar el script que crea cada una de las tablas que conforman la base de datos propuesta por el Comité Olímpico.

NOTA: Debe cumplir con todas las restricciones correspondientes (primary key, Foreign Key, Unique, Null, Not null).

SCRIPT SQL SERVER

```
create table PROFESION (  
    cod_prof integer not null,  
    nombre varchar(50) constraint pro_un_nombre unique not null,  
    constraint PK_profesion primary key (cod_prof)  
);
```

```
create table PAIS (  
    cod_pais integer not null,  
    nombre varchar(50) constraint pai_un_nombre unique not null,  
    constraint PK_pais primary key (cod_pais)  
);
```

```
create table PUESTO (  
    cod_puesto integer not null,  
    nombre varchar(50) constraint pue_un_nombre unique not null,  
    constraint PK_puesto primary key (cod_puesto)  
);
```

```
create table DEPARTAMENTO (  
    cod_depto integer not null,  
    nombre varchar(50) constraint dep_un_nombre unique not null,  
    constraint PK_depto primary key (cod_depto)  
);
```

```
create table MIEMBRO (  
    cod_miembro integer not null,  
    nombre varchar(100) not null,  
    apellido varchar(100) not null,  
    edad integer not null,  
    telefono integer,  
    residencia varchar(100),  
    PAIS_cod_pais integer not null,  
    PROFESION_cod_prof integer not null,  
    constraint PK_miembro primary key (cod_miembro),
```

```

        constraint FK_miembro_cod_pais foreign key (PAIS_cod_pais) references
PAIS (cod_pais),
        constraint FK_miembro_cod_prof foreign key (PROFESION_cod_prof)
references PROFESION (cod_prof)
);

```

```

create table PUESTO_MIEMBRO (
    MIEMBRO_cod_miembro integer not null,
    PUESTO_cod_puesto integer not null,
    DEPARTAMENTO_cod_depto integer not null,
    fecha_inicio date not null,
    fecha_fin date,
    constraint PK_puesto_miembro primary key (MIEMBRO_cod_miembro,
PUESTO_cod_puesto, DEPARTAMENTO_cod_depto),
    constraint FK_pm_cod_miembro foreign key (MIEMBRO_cod_miembro)
references MIEMBRO (cod_miembro),
    constraint FK_pm_cod_puesto foreign key (PUESTO_cod_puesto)
references PUESTO (cod_puesto),
    constraint FK_pm_cod_depto foreign key (DEPARTAMENTO_cod_depto)
references DEPARTAMENTO (cod_depto)
);

```

```

create table TIPO_MEDALLA (
    cod_tipo integer not null,
    medalla varchar(20) constraint tme_un_medalla unique not null,
    constraint PK_tipo_medalla primary key (cod_tipo)
);

```

```

create table MEDALLERO (
    PAIS_cod_pais integer not null,
    cantidad_medallas integer not null,
    TIPO_MEDALLA_cod_tipo integer not null,
    constraint PK_medallero primary key (PAIS_cod_pais,
TIPO_MEDALLA_cod_tipo),
    constraint FK_medallero_cod_pais foreign key (PAIS_cod_pais) references
PAIS (cod_pais),
    constraint FK_medallero_cod_tipo foreign key (TIPO_MEDALLA_cod_tipo)
references TIPO_MEDALLA (cod_tipo)
);

```

```

create table DISCIPLINA (
    cod_disciplina integer not null,
    nombre varchar(50) not null,
    descripcion varchar(150),

```

```
constraint PK_cod_disciplina primary key (cod_disciplina)
);
```

```
create table ATLETA (
    cod_atleta integer not null,
    nombre varchar(50) not null,
    apellido varchar(50) not null,
    edad integer not null,
    participaciones varchar(100) not null,
    DISCIPLINA_cod_disciplina integer not null,
    PAIS_cod_pais integer not null,
    constraint PK_atleta primary key (cod_atleta),
    constraint FK_atleta_cod_disciplina foreign key
(DISCIPLINA_cod_disciplina) references DISCIPLINA (cod_disciplina),
    constraint FK_atleta_cod_pais foreign key (PAIS_cod_pais) references
PAIS (cod_pais)
);
```

```
create table CATEGORIA (
    cod_categoria integer not null,
    categoria varchar(50) not null,
    constraint PK_categoria primary key (cod_categoria)
);
```

```
create table TIPO_PARTICIPACION (
    cod_participacion integer not null,
    tipo_participacion varchar(100) not null,
    constraint PK_tipo_participacion primary key (cod_participacion)
);
```

```
create table EVENTO (
    cod_evento integer not null,
    fecha date not null,
    ubicacion varchar(50) not null,
    hora date not null,
    DISCIPLINA_cod_disciplina integer not null,
    TIPO_PARTICIPACION_cod_participacion integer not null,
    CATEGORIA_cod_categoria integer not null,
    constraint PK_evento primary key (cod_evento),
    constraint FK_evento_cod_disciplina foreign key
(DISCIPLINA_cod_disciplina) references DISCIPLINA (cod_disciplina),
    constraint FK_evento_cod_participacion foreign key
(TIPO_PARTICIPACION_cod_participacion) references TIPO_PARTICIPACION
(cod_participacion),
```

```
        constraint FK_evento_cod_categoria foreign key  
(CATEGORIA_cod_categoria) references CATEGORIA (cod_categoria)  
);
```

```
create table EVENTO_ATLETA (  
    ATLETA_cod_atleta integer not null,  
    EVENTO_cod_evento integer not null,  
    constraint PK_evento_atleta primary key (ATLETA_cod_atleta,  
EVENTO_cod_evento),  
    constraint FK_evento_cod_atleta foreign key (ATLETA_cod_atleta)  
references ATLETA (cod_atleta),  
    constraint FK_evento_cod_evento foreign key (EVENTO_cod_evento)  
references EVENTO (cod_evento)  
);
```

```
create table TELEVISORA (  
    cod_televisora integer not null,  
    nombre varchar(50) not null,  
    constraint PK_televisora primary key (cod_televisora)  
);
```

```
create table COSTO_EVENTO (  
    EVENTO_cod_evento integer not null,  
    TELEVISORA_cod_televisora integer not null,  
    tarifa numeric not null,  
    constraint PK_costo_evento primary key (EVENTO_cod_evento,  
TELEVISORA_cod_televisora),  
    constraint FK_ce_cod_evento foreign key (EVENTO_cod_evento)  
references EVENTO (cod_evento),  
    constraint FK_ce_cod_televisora foreign key (TELEVISORA_cod_televisora)  
references EVENTO (cod_evento)  
);
```

SCRIPT ORACLE

```
create table PROFESION (  
    cod_prof integer not null,  
    nombre varchar(50) constraint pro_un_nombre unique not null,  
    constraint PK_profesion primary key (cod_prof)  
);
```

```
create table PAIS (  
    cod_pais integer not null,  
    nombre varchar(50) constraint pai_un_nombre unique not null,  
    constraint PK_pais primary key (cod_pais)
```

);

```
create table PUESTO (  
    cod_puesto integer not null,  
    nombre varchar(50) constraint pue_un_nombre unique not null,  
    constraint PK_puesto primary key (cod_puesto)  
);
```

```
create table DEPARTAMENTO (  
    cod_depto integer not null,  
    nombre varchar(50) constraint dep_un_nombre unique not null,  
    constraint PK_depto primary key (cod_depto)  
);
```

```
create table MIEMBRO (  
    cod_miembro integer not null,  
    nombre varchar(100) not null,  
    apellido varchar(100) not null,  
    edad integer not null,  
    telefono integer,  
    residencia varchar(100),  
    PAIS_cod_pais integer not null,  
    PROFESION_cod_prof integer not null,  
    constraint PK_miembro primary key (cod_miembro),  
    constraint FK_miembro_cod_pais foreign key (PAIS_cod_pais) references  
PAIS (cod_pais),  
    constraint FK_miembro_cod_prof foreign key (PROFESION_cod_prof)  
references PROFESION (cod_prof)  
);
```

```
create table PUESTO_MIEMBRO (  
    MIEMBRO_cod_miembro integer not null,  
    PUESTO_cod_puesto integer not null,  
    DEPARTAMENTO_cod_depto integer not null,  
    fecha_inicio date not null,  
    fecha_fin date,  
    constraint PK_puesto_miembro primary key (MIEMBRO_cod_miembro,  
PUESTO_cod_puesto, DEPARTAMENTO_cod_depto),  
    constraint FK_pm_cod_miembro foreign key (MIEMBRO_cod_miembro)  
references MIEMBRO (cod_miembro),  
    constraint FK_pm_cod_puesto foreign key (PUESTO_cod_puesto)  
references PUESTO (cod_puesto),  
    constraint FK_pm_cod_depto foreign key (DEPARTAMENTO_cod_depto)  
references DEPARTAMENTO (cod_depto)
```

);

```
create table TIPO_MEDALLA (  
    cod_tipo integer not null,  
    medalla varchar(20) constraint tme_un_medalla unique not null,  
    constraint PK_tipo_medalla primary key (cod_tipo)  
);
```

```
create table MEDALLERO (  
    PAIS_cod_pais integer not null,  
    cantidad_medallas integer not null,  
    TIPO_MEDALLA_cod_tipo integer not null,  
    constraint PK_medallero primary key (PAIS_cod_pais,  
TIPO_MEDALLA_cod_tipo),  
    constraint FK_medallero_cod_pais foreign key (PAIS_cod_pais) references  
PAIS (cod_pais),  
    constraint FK_medallero_cod_tipo foreign key (TIPO_MEDALLA_cod_tipo)  
references TIPO_MEDALLA (cod_tipo)  
);
```

```
create table DISCIPLINA (  
    cod_disciplina integer not null,  
    nombre varchar(50) not null,  
    descripcion varchar(150),  
    constraint PK_cod_disciplina primary key (cod_disciplina)  
);
```

```
create table ATLETA (  
    cod_atleta integer not null,  
    nombre varchar(50) not null,  
    apellido varchar(50) not null,  
    edad integer not null,  
    participaciones varchar(100) not null,  
    DISCIPLINA_cod_disciplina integer not null,  
    PAIS_cod_pais integer not null,  
    constraint PK_atleta primary key (cod_atleta),  
    constraint FK_atleta_cod_disciplina foreign key  
(DISCIPLINA_cod_disciplina) references DISCIPLINA (cod_disciplina),  
    constraint FK_atleta_cod_pais foreign key (PAIS_cod_pais) references  
PAIS (cod_pais)  
);
```

```
create table CATEGORIA (  
    cod_categoria integer not null,
```

```

        categoria varchar(50) not null,
        constraint PK_categoria primary key (cod_categoria)
    );

create table TIPO_PARTICIPACION (
    cod_participacion integer not null,
    tipo_participacion varchar(100) not null,
    constraint PK_tipo_participacion primary key (cod_participacion)
);

create table EVENTO (
    cod_evento integer not null,
    fecha date not null,
    ubicacion varchar(50) not null,
    hora date not null,
    D_cod_disciplina integer not null,
    TP_cod_participacion integer not null,
    CATEGORIA_cod_categoria integer not null,
    constraint PK_evento primary key (cod_evento),
    constraint FK_evento_cod_disciplina foreign key (D_cod_disciplina)
references DISCIPLINA (cod_disciplina),
    constraint FK_evento_cod_participacion foreign key
(TP_cod_participacion) references TIPO_PARTICIPACION (cod_participacion),
    constraint FK_evento_cod_categoria foreign key
(CATEGORIA_cod_categoria) references CATEGORIA (cod_categoria)
);

create table EVENTO_ATLETA (
    ATLETA_cod_atleta integer not null,
    EVENTO_cod_evento integer not null,
    constraint PK_evento_atleta primary key (ATLETA_cod_atleta,
EVENTO_cod_evento),
    constraint FK_evento_cod_atleta foreign key (ATLETA_cod_atleta)
references ATLETA (cod_atleta),
    constraint FK_evento_cod_evento foreign key (EVENTO_cod_evento)
references EVENTO (cod_evento)
);

create table TELEVISORA (
    cod_televisora integer not null,
    nombre varchar(50) not null,
    constraint PK_televisora primary key (cod_televisora)
);

```

```
create table COSTO_EVENTO (
    EVENTO_cod_evento integer not null,
    TELEVISORA_cod_televisora integer not null,
    tarifa numeric not null,
    constraint PK_costo_evento primary key (EVENTO_cod_evento,
    TELEVISORA_cod_televisora),
    constraint FK_ce_cod_evento foreign key (EVENTO_cod_evento)
references EVENTO (cod_evento),
    constraint FK_ce_cod_televisora foreign key (TELEVISORA_cod_televisora)
references EVENTO (cod_evento)
);
```

2. En la tabla “Evento” se decidió que la fecha y hora se trabajaría en una sola columna.
 - Eliminar las columnas fecha y hora.
 - Crear una columna llamada “fecha_hora” con el tipo de dato que corresponda según el DBMS.

SCRIPT SQL SERVER

```
alter table EVENTO drop column fecha;
alter table EVENTO drop column hora;
alter table EVENTO add fecha_hora datetime not null;
```

SCRIPT ORACLE

```
alter table EVENTO drop column fecha;
alter table EVENTO drop column hora;
alter table EVENTO add fecha_hora date not null;
```

3. Todos los eventos de las olimpiadas deben ser programados del 24 de julio de 2020 a partir de las 9:00:00 hasta el 09 de agosto de 2020 hasta las 20:00:00.

Generar el Script que únicamente permita registrar los eventos entre estas fechas y horarios.

SCRIPT SQLSERVER

```
alter table EVENTO
add constraint evento_fecha_hora
check (fecha_hora between '2020-07-24T09:00:00' and '2020-08-09T20:00:00');
```

SCRIPT ORACLE

```
alter table EVENTO
add constraint evento_fecha_hora
```


check (fecha_hora between TO_DATE('2020-07-24 09:00:00', 'YYYY-MM-DD HH:MI:SS') and TO_DATE('2020-08-09 20:00:00', 'YYYY-MM-DD HH24:MI:SS'));

4. Se decidió que las ubicación de los eventos se registrarán previamente en una tabla y que en la tabla “Evento” sólo se almacenara la llave foránea según el código del registro de la ubicación, para esto debe realizar lo siguiente:
 - a. Crear la tabla llamada “Sede” que tendrá los campos:
 - i. **Código:** será tipo entero y será la llave primaria.
 - ii. **Sede:** será tipo varchar(50) y será obligatoria.
 - b. Cambiar el tipo de dato de la columna Ubicación de la tabla Evento por un tipo entero.
 - c. Crear una llave foránea en la columna *Ubicación* de la tabla *Evento* y referenciarla a la columna *código* de la tabla *Sede*, la que fue creada en el paso anterior.

SCRIPT SQLSERVER

```
create table SEDE (  
    codigo integer not null,  
    sede varchar(50) not null,  
    constraint PK_sede primary key (codigo)  
);
```

```
alter table EVENTO alter column ubicacion integer not null;
```

```
alter table EVENTO add constraint FK_evento_ubicacion foreign key (ubicacion)  
references SEDE (codigo);
```

SCRIPT ORACLE

```
create table SEDE (  
    codigo integer not null,  
    sede varchar(50) not null,  
    constraint PK_sede primary key (codigo)  
);
```

```
alter table EVENTO modify ( ubicacion integer );
```

```
alter table EVENTO add constraint FK_evento_ubicacion foreign key (ubicacion)  
references SEDE (codigo);
```

5. Se revisó la información de los miembros que se tienen actualmente y antes de que se ingresen a la base de datos el Comité desea que a los miembros que no tengan número telefónico se le ingrese el número por Default 0 al momento de ser cargados a la base de datos.

SCRIPT SQLSERVER

alter table MIEMBRO add constraint default_miembro default 0 for telefono;

SCRIPT ORACLE

alter table MIEMBRO modify (telefono default 0 not null);

6. Generar el script necesario para hacer la inserción de datos a las tablas requeridas.

Revisar el documento “Insercion.pdf” compartido junto a este enunciado, ahí se encuentran las tablas y los datos que hay que insertar.

SCRIPT SQLSERVER

```
insert into PAIS (cod_pais, nombre) values (1, 'Guatemala');
insert into PAIS (cod_pais, nombre) values (2, 'Francia');
insert into PAIS (cod_pais, nombre) values (3, 'Argentina');
insert into PAIS (cod_pais, nombre) values (4, 'Alemania');
insert into PAIS (cod_pais, nombre) values (5, 'Italia');
insert into PAIS (cod_pais, nombre) values (6, 'Brasil');
insert into PAIS (cod_pais, nombre) values (7, 'Estados Unidos');
```

```
insert into PROFESION (cod_prof, nombre) values (1, 'Medico');
insert into PROFESION (cod_prof, nombre) values (2, 'Arquitecto');
insert into PROFESION (cod_prof, nombre) values (3, 'Ingeniero');
insert into PROFESION (cod_prof, nombre) values (4, 'Secretaria');
insert into PROFESION (cod_prof, nombre) values (5, 'Auditor');
```

```
insert into MIEMBRO (cod_miembro, nombre, apellido, edad, telefono, residencia,
PAIS_cod_pais, PROFESION_cod_prof)
values (1, 'Scott', 'Mitchell', 32, '', '1092 Highland Drive Manitowoc, WI 54220', 7, 3);
insert into MIEMBRO (cod_miembro, nombre, apellido, edad, telefono, residencia,
PAIS_cod_pais, PROFESION_cod_prof)
values (2, 'Fanette', 'Poulin', 25, 25075853, '49, boulevard Aristide Briand 76120 LE
GRAND-QUEVILLY', 2, 4);
insert into MIEMBRO (cod_miembro, nombre, apellido, edad, telefono, residencia,
PAIS_cod_pais, PROFESION_cod_prof)
values (3, 'Laura', 'Cunha Silva', 55, '', 'Rua Onze, 86 Uberaba-MG', 6, 5);
insert into MIEMBRO (cod_miembro, nombre, apellido, edad, telefono, residencia,
PAIS_cod_pais, PROFESION_cod_prof)
values (4, 'Juan José', 'López', 38, 36985247, '26 calle 4-10 zona 11', 1, 2);
insert into MIEMBRO (cod_miembro, nombre, apellido, edad, telefono, residencia,
PAIS_cod_pais, PROFESION_cod_prof)
values (5, 'Arcangela', 'Panicucci', 39, 391664921, 'Via Santa Teresa, 114 90010-Geraci
Siculo PA', 5, 1);
```

```
insert into MIEMBRO (cod_miembro, nombre, apellido, edad, telefono, residencia,
PAIS_cod_pais, PROFESION_cod_prof)
values (6, 'Jeuel', 'Villalpando', 31, '', 'Acuña de Figueroa 6106 80101 Playa Pascual', 3, 5);
```

```
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
values (1, 'Atletismo', 'Saltos de longitud y triples, de altura y con pértiga o garrocha;
las pruebas de lanzamiento de martillo, jabalina y disco');
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
values (2, 'Bádminton', '');
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
values (3, 'Ciclismo', '');
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
values (4, 'Judo', 'Es un arte marcial que se originó en japon alrededor de 1880');
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
values (5, 'Lucha', '');
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
values (6, 'Tenis de Mesa', '');
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
values (7, 'Boxeo', '');
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
values (8, 'Natación', 'Está presente como deporte en los juegos desde la primera
edición de la era moderna, en Atenas, Grecia, en 1896, donde se disputo en aguas
abiertas');
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
values (9, 'Esgrima', '');
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
values (10, 'Vela', '' );
```

```
insert into TIPO_MEDALLA (cod_tipo, medalla) values (1, 'Oro');
insert into TIPO_MEDALLA (cod_tipo, medalla) values (2, 'Plata');
insert into TIPO_MEDALLA (cod_tipo, medalla) values (3, 'Bronce');
insert into TIPO_MEDALLA (cod_tipo, medalla) values (4, 'Platino');
```

```
insert into CATEGORIA (cod_categoria, categoria) values (1, 'Clasificatorio');
insert into CATEGORIA (cod_categoria, categoria) values (2, 'Eliminatorio');
insert into CATEGORIA (cod_categoria, categoria) values (3, 'Final');
```

```
insert into TIPO_PARTICIPACION (cod_participacion, tipo_participacion) values (1,
'Individual');
insert into TIPO_PARTICIPACION (cod_participacion, tipo_participacion) values (2,
'Parejas');
insert into TIPO_PARTICIPACION (cod_participacion, tipo_participacion) values (3,
'Equipos');
```

```
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (5,1,3);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (2,1,5);
```

```

insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (6,3,4);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (4,4,3);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (7,3,10);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (3,2,8);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (1,1,2);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (1,4,5);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (5,2,7);

```

```

insert into SEDE (codigo, sede) values (1, 'Gimnasio Metropolitano de Tokio');
insert into SEDE (codigo, sede) values (2, 'Jardín del Palacio Imperial de Tokio');
insert into SEDE (codigo, sede) values (3, 'Gimnasio Nacional Yoyogi');
insert into SEDE (codigo, sede) values (4, 'Nippon Budokan');
insert into SEDE (codigo, sede) values (5, 'Estadio Olímpico');

```

```

insert into EVENTO (cod_evento, fecha_hora, ubicacion, DISCIPLINA_cod_disciplina,
TIPO_PARTICIPACION_cod_participacion, CATEGORIA_cod_categoria)
values (1, '2020-07-24T09:00:00', 3, 2, 2, 1);
insert into EVENTO (cod_evento, fecha_hora, ubicacion, DISCIPLINA_cod_disciplina,
TIPO_PARTICIPACION_cod_participacion, CATEGORIA_cod_categoria)
values (2, '2020-07-26T10:30:00', 1, 6, 1, 3);
insert into EVENTO (cod_evento, fecha_hora, ubicacion, DISCIPLINA_cod_disciplina,
TIPO_PARTICIPACION_cod_participacion, CATEGORIA_cod_categoria)
values (3, '2020-07-30T18:45:00', 5, 7, 1, 2);
insert into EVENTO (cod_evento, fecha_hora, ubicacion, DISCIPLINA_cod_disciplina,
TIPO_PARTICIPACION_cod_participacion, CATEGORIA_cod_categoria)
values (4, '2020-08-01T12:15:00', 2, 1, 1, 1);
insert into EVENTO (cod_evento, fecha_hora, ubicacion, DISCIPLINA_cod_disciplina,
TIPO_PARTICIPACION_cod_participacion, CATEGORIA_cod_categoria)
values (5, '2020-08-08T19:35:00', 4, 10, 3, 1);

```

SCRIPT ORACLE

```

insert into PAIS (cod_pais, nombre) values (1, 'Guatemala');
insert into PAIS (cod_pais, nombre) values (2, 'Francia');
insert into PAIS (cod_pais, nombre) values (3, 'Argentina');
insert into PAIS (cod_pais, nombre) values (4, 'Alemania');
insert into PAIS (cod_pais, nombre) values (5, 'Italia');
insert into PAIS (cod_pais, nombre) values (6, 'Brasil');
insert into PAIS (cod_pais, nombre) values (7, 'Estados Unidos');

```

```

insert into PROFESION (cod_prof, nombre) values (1, 'Medico');
insert into PROFESION (cod_prof, nombre) values (2, 'Arquitecto');
insert into PROFESION (cod_prof, nombre) values (3, 'Ingeniero');

```

```
insert into PROFESION (cod_prof, nombre) values (4, 'Secretaria');
```

```
insert into PROFESION (cod_prof, nombre) values (5, 'Auditor');
```

```
insert into MIEMBRO (cod_miembro, nombre, apellido, edad, residencia,
```

```
PAIS_cod_pais, PROFESION_cod_prof)
```

```
values (1, 'Scott', 'Mitchell', 32, '1092 Highland Drive Manitowoc, WI 54220', 7, 3);
```

```
insert into MIEMBRO (cod_miembro, nombre, apellido, edad, telefono, residencia,
```

```
PAIS_cod_pais, PROFESION_cod_prof)
```

```
values (2, 'Fanette', 'Poulin', 25, 25075853, '49, boulevard Aristide Briand 76120 LE  
GRAND-QUEVILLY', 2, 4);
```

```
insert into MIEMBRO (cod_miembro, nombre, apellido, edad, residencia,
```

```
PAIS_cod_pais, PROFESION_cod_prof)
```

```
values (3, 'Laura', 'Cunha Silva', 55, 'Rua Onze, 86 Uberaba-MG', 6, 5);
```

```
insert into MIEMBRO (cod_miembro, nombre, apellido, edad, telefono, residencia,
```

```
PAIS_cod_pais, PROFESION_cod_prof)
```

```
values (4, 'Juan José', 'López', 38, 36985247, '26 calle 4-10 zona 11', 1, 2);
```

```
insert into MIEMBRO (cod_miembro, nombre, apellido, edad, telefono, residencia,
```

```
PAIS_cod_pais, PROFESION_cod_prof)
```

```
values (5, 'Arcangela', 'Panicucci', 39, 391664921, 'Via Santa Teresa, 114 90010-Geraci  
Siculo PA', 5, 1);
```

```
insert into MIEMBRO (cod_miembro, nombre, apellido, edad, residencia,
```

```
PAIS_cod_pais, PROFESION_cod_prof)
```

```
values (6, 'Jeuel', 'Villalpando', 31, 'Acuña de Figueroa 6106 80101 Playa Pascual', 3, 5);
```

```
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
```

```
values (1, 'Atletismo', 'Saltos de longitud y triples, de altura y con pértiga o garrocha;  
las pruebas de lanzamiento de martillo, jabalina y disco');
```

```
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
```

```
values (2, 'Bádminton', '');
```

```
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
```

```
values (3, 'Ciclismo', '');
```

```
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
```

```
values (4, 'Judo', 'Es un arte marcial que se originó en japon alrededor de 1880');
```

```
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
```

```
values (5, 'Lucha', '');
```

```
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
```

```
values (6, 'Tenis de Mesa', '');
```

```
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
```

```
values (7, 'Boxeo', '');
```

```
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
```

```
values (8, 'Natación', 'Está presente como deporte en los juegos desde la primera  
edición de la era moderna, en Atenas, Grecia, en 1896, donde se disputo en aguas  
abiertas');
```

```
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
```

```
values (9, 'Esgrima', '');
```

```
insert into DISCIPLINA (cod_disciplina, nombre, descripcion)
```

```
values (10, 'Vela', '');
```

```
insert into TIPO_MEDALLA (cod_tipo, medalla) values (1, 'Oro');
insert into TIPO_MEDALLA (cod_tipo, medalla) values (2, 'Plata');
insert into TIPO_MEDALLA (cod_tipo, medalla) values (3, 'Bronce');
insert into TIPO_MEDALLA (cod_tipo, medalla) values (4, 'Platino');
```

```
insert into CATEGORIA (cod_categoria, categoria) values (1, 'Clasificatorio');
insert into CATEGORIA (cod_categoria, categoria) values (2, 'Eliminatorio');
insert into CATEGORIA (cod_categoria, categoria) values (3, 'Final');
```

```
insert into TIPO_PARTICIPACION (cod_participacion, tipo_participacion) values (1,
'Individual');
insert into TIPO_PARTICIPACION (cod_participacion, tipo_participacion) values (2,
'Parejas');
insert into TIPO_PARTICIPACION (cod_participacion, tipo_participacion) values (3,
'Equipos');
```

```
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (5,1,3);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (2,1,5);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (6,3,4);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (4,4,3);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (7,3,10);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (3,2,8);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (1,1,2);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (1,4,5);
insert into MEDALLERO (PAIS_cod_pais, TIPO_MEDALLA_cod_tipo, cantidad_medallas)
values (5,2,7);
```

```
insert into SEDE (codigo, sede) values (1, 'Gimnasio Metropolitano de Tokio');
insert into SEDE (codigo, sede) values (2, 'Jardín del Palacio Imperial de Tokio');
insert into SEDE (codigo, sede) values (3, 'Gimnasio Nacional Yoyogi');
insert into SEDE (codigo, sede) values (4, 'Nippon Budokan');
insert into SEDE (codigo, sede) values (5, 'Estadio Olímpico');
```

```
insert into EVENTO (cod_evento, fecha_hora, ubicacion, D_cod_disciplina,
TP_cod_participacion, CATEGORIA_cod_categoria)
values (1, TO_DATE('2020-07-24 09:00:00', 'YYYY-MM-DD HH24:MI:SS'), 3, 2, 2, 1);
insert into EVENTO (cod_evento, fecha_hora, ubicacion, D_cod_disciplina,
TP_cod_participacion, CATEGORIA_cod_categoria)
values (2, TO_DATE('2020-07-26 10:30:00', 'YYYY-MM-DD HH24:MI:SS'), 1, 6, 1, 3);
```

```

insert into EVENTO (cod_evento, fecha_hora, ubicacion, D_cod_disciplina,
TP_cod_participacion, CATEGORIA_cod_categoria)
values (3, TO_DATE('2020-07-30 18:45:00', 'YYYY-MM-DD HH24:MI:SS'), 5, 7, 1, 2);
insert into EVENTO (cod_evento, fecha_hora, ubicacion, D_cod_disciplina,
TP_cod_participacion, CATEGORIA_cod_categoria)
values (4, TO_DATE('2020-08-01 12:15:00', 'YYYY-MM-DD HH24:MI:SS'), 2, 1, 1, 1);
insert into EVENTO (cod_evento, fecha_hora, ubicacion, D_cod_disciplina,
TP_cod_participacion, CATEGORIA_cod_categoria)
values (5, TO_DATE('2020-08-08 19:35:00', 'YYYY-MM-DD HH24:MI:SS'), 4, 10, 3, 1);

```

7. Después de que se implementó el script el cuál creó todas las tablas de las bases de datos, el Comité Olímpico Internacional tomó la decisión de eliminar la restricción “UNIQUE”.

Elabore el script que elimine las restricciones “UNIQUE” de las columnas antes mencionadas.

SCRIPT SQLSERVER

```

alter table PAIS drop constraint pai_un_nombre;
alter table TIPO_MEDALLA drop constraint tme_un_medalla;
alter table DEPARTAMENTO drop constraint dep_un_nombre;

```

SCRIPT ORACLE

```

alter table PAIS drop constraint pai_un_nombre;
alter table TIPO_MEDALLA drop constraint tme_un_medalla;
alter table DEPARTAMENTO drop constraint dep_un_nombre;

```

8. Después de un análisis más profundo se decidió que los Atletas pueden participar en varias disciplinas y no sólo en una como está reflejado actualmente en las tablas, por lo que se pide que realice lo siguiente.

- a. Script que elimine la llave foránea de “cod_disciplina” que se encuentra en la tabla “Atleta”.
- b. Script que cree una tabla con el nombre “Disciplina_Atleta” que contendrá los siguiente campos:
 - i. Cod_atleta (llave foránea de la tabla Atleta)
 - ii. Cod_disciplina (llave foránea de la tabla Disciplina)

La llave primaria será la unión de las llaves foráneas “cod_atleta” y “cod_disciplina”.

SCRIPT SQLSERVER

```

alter table ATLETA drop constraint FK_atleta_cod_disciplina;

```

```

create table DISCIPLINA_ATLETA (

```

```

        ATLETA_cod_atleta integer not null,
        DISCIPLINA_cod_disciplina integer not null,
        constraint PK_disciplina_atleta primary key (ATLETA_cod_atleta,
        DISCIPLINA_cod_disciplina),
        constraint FK_da_cod_atleta foreign key (ATLETA_cod_atleta) references
        ATLETA (cod_atleta),
        constraint FK_da_cod_disciplina foreign key (DISCIPLINA_cod_disciplina)
        references DISCIPLINA (cod_disciplina)
    );

```

SCRIPT ORACLE

```

alter table ATLETA drop constraint FK_atleta_cod_disciplina;

```

```

create table DISCIPLINA_ATLETA (
    ATLETA_cod_atleta integer not null,
    DISCIPLINA_cod_disciplina integer not null,
    constraint PK_disciplina_atleta primary key (ATLETA_cod_atleta,
    DISCIPLINA_cod_disciplina),
    constraint FK_da_cod_atleta foreign key (ATLETA_cod_atleta) references
    ATLETA (cod_atleta),
    constraint FK_da_cod_disciplina foreign key (DISCIPLINA_cod_disciplina)
    references DISCIPLINA (cod_disciplina)
);

```

9. En la tabla “Costo_Evento” se determinó que la columna “tarifa” no debe ser entero sino un decimal con 2 cifras de precisión.

Generar el script correspondiente para modificar el tipo de dato que se le pide.

SCRIPT SQLSERVER

```

alter table COSTO_EVENTO alter column tarifa numeric(10, 2) not null;

```

SCRIPT ORACLE

```

alter table COSTO_EVENTO MODIFY ( tarifa numeric(10, 2) );

```

10. Generar el Script que borre de la tabla “Tipo_Medalla”

SCRIPT SQLSERVER

```

delete from MEDALLERO where TIPO_MEDALLA_cod_tipo = 4;
delete from TIPO_MEDALLA where TIPO_MEDALLA.cod_tipo = 4 and
TIPO_MEDALLA.medalla = 'Platino';

```

SCRIPT ORACLE

```

delete from MEDALLERO where TIPO_MEDALLA_cod_tipo = 4;
delete from TIPO_MEDALLA where TIPO_MEDALLA.cod_tipo = 4 and
TIPO_MEDALLA.medalla = 'Platino';

```


11. La fecha de las olimpiadas está cerca y los preparativos siguen, pero de último momento se dieron problemas con las televisoras encargadas de transmitir los eventos, ya que no hay tiempo de solucionar los problemas que se dieron, se decidió no transmitir el evento a través de las televisoras por lo que el Comité Olímpico pide generar el script que elimine la tabla "TELEVISORAS" y "COSTO_EVENTO".

SCRIPT SQLSERVER

```
drop table COSTO_EVENTO;  
drop table TELEVISORA;
```

SCRIPT ORACLE

```
drop table COSTO_EVENTO;  
drop table TELEVISORA;
```

12. El comité olímpico quiere replantear las disciplinas que van a llevarse a cabo, por lo cual pide generar el script que elimine todos los registros contenidos en la tabla "DISCIPLINA".

SCRIPT SQLSERVER

```
delete EVENTO from EVENTO E inner join DISCIPLINA D  
on E.DISCIPLINA_cod_disciplina = D.cod_disciplina;
```

```
delete DISCIPLINA_ATLETA from DISCIPLINA_ATLETA DA inner join DISCIPLINA D  
on DA.DISCIPLINA_cod_disciplina = D.cod_disciplina;
```

```
delete from DISCIPLINA;
```

SCRIPT ORACLE

```
delete from (  
  select E.* from EVENTO E INNER JOIN DISCIPLINA D  
  on E.d_cod_disciplina = D.cod_disciplina  
);
```

```
delete from (  
  select DA.* from DISCIPLINA_ATLETA DA INNER JOIN DISCIPLINA D  
  on DA.DISCIPLINA_cod_disciplina = D.cod_disciplina  
);
```

```
delete from DISCIPLINA;
```

13. Los miembros que no tenían registrado su número de teléfono en sus perfiles fueron notificados, por lo que se acercaron a las instalaciones de Comité para actualizar sus datos.

Genere el script que permita actualizar los números telefónicos de los miembros anteriores.

SCRIPT SQLSERVER

```
update MIEMBRO set telefono = 55464601 where nombre = 'Laura' and apellido = 'Cunha Silva';  
update MIEMBRO set telefono = 91514243 where nombre = 'Jeuel' and apellido = 'Villalpando';  
update MIEMBRO set telefono = 920686670 where nombre = 'Scott' and apellido = 'Mitchell';
```

SCRIPT ORACLE

```
update MIEMBRO set telefono = 55464601 where nombre = 'Laura' and apellido = 'Cunha Silva';  
update MIEMBRO set telefono = 91514243 where nombre = 'Jeuel' and apellido = 'Villalpando';  
update MIEMBRO set telefono = 920686670 where nombre = 'Scott' and apellido = 'Mitchell';
```

14. El Comité decidió que necesita la fotografía en la información de los atletas para su perfil, por lo que se debe agregar la columna “Fotografía” a la tabla Atleta, debido a que es un cambio de última hora este campo deberá ser opcional.

Utilice el tipo de dato que crea conveniente según el DBMS y explique el por qué utilizó este tipo de dato.

SCRIPT SQLSERVER

```
alter table ATLETA add fotografia varchar(100);
```

SCRIPT ORACLE

```
alter table ATLETA add fotografia varchar(100);
```

15. Todos los atletas que se registren deben cumplir con ser menores a 25 años. De lo contrario no se debe poder registrar a un atleta en la base de datos.

SCRIPT SQLSERVER

```
alter table ATLETA add constraint ch_atleta_edad check (edad < 25);
```

SCRIPT ORACLE

```
alter table ATLETA add constraint ch_atleta_edad check (edad < 25);
```