



Semana 8

## Modelamiento de Bases de Datos (PRY2204)

Formato de respuesta

Nombre estudiante: Victor Vasquez	
Asignatura: PRY2204	Carrera: Desarrollo de Aplicaciones
Profesor: Josué Oteiza	Fecha: 2024-09-30

## Descripción de la actividad

En esta octava semana, realizarás de forma individual una actividad sumativa llamada "Realizando el Poblamiento y Consultas en la base de datos con sentencias SQL" donde tendrás que insertar y seleccionar datos desde la base de datos, así como también generar los scripts de inserción de datos y construir consultas simples para mostrar los datos almacenados.

### Instrucciones específicas

Para poder realizar la actividad de la semana, primero lee el caso planteado:

“El Instituto Nacional del Deporte (IND) de Chile está en proceso de fortalecer la infraestructura deportiva y la participación en actividades físicas a nivel nacional. Con el fin de promover un estilo de vida saludable y después de la pandemia del Covid-19, el IND se ha dado cuenta de la necesidad de un sistema integral que maneje eficientemente la información de todas las escuelas deportivas del país.

Este sistema permitirá al IND gestionar y ofrecer apoyo financiero a las escuelas deportivas que fomentan la práctica del deporte entre los jóvenes. Para lograr este fin, el sistema debe ser capaz de almacenar datos sobre el personal contratado, los costos operacionales, y las instalaciones deportivas, así como facilitar el proceso de solicitud de fondos”.

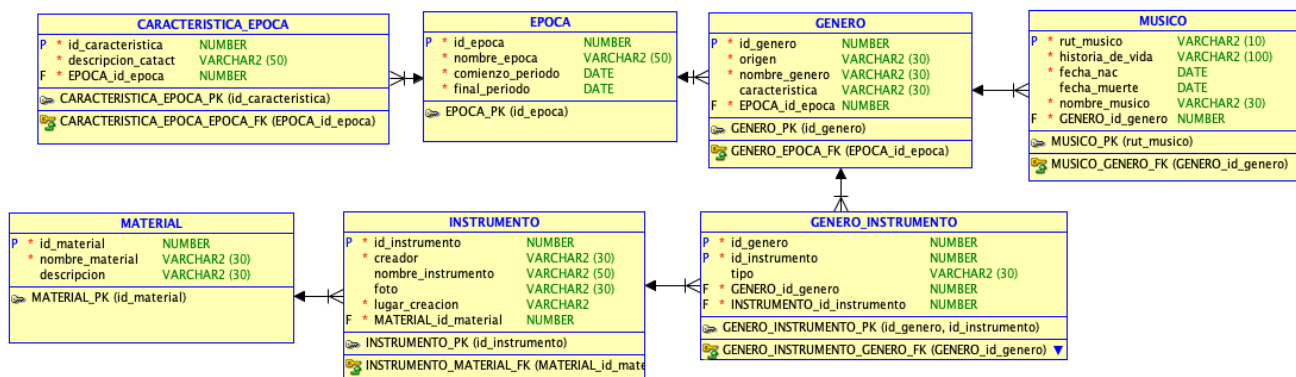
Tu tarea como especialista en tecnologías de la información es diseñar e implementar una base de datos relacional que sirva como el esqueleto de este sistema. Deberás asegurarte de que la base de datos sea capaz de:

- Almacenar la información detallada de cada escuela deportiva, incluyendo el tipo de escuela (fútbol, baloncesto, etc.), la información de contacto, y los detalles de la ubicación.
- Registrar los datos del personal a cargo de cada escuela, incluyendo su profesión, nacionalidad y otros datos personales relevantes.
- Gestionar los costos asociados al funcionamiento de cada escuela, tales como el pago a entrenadores y la compra de equipos e insumos.

Antes de comenzar con la creación de tu base de datos, es útil observar un ejemplo de lo que podrías aspirar a diseñar. A continuación se presenta una figura que muestra un modelo de datos relacional. Este es un ejemplo de cómo se pueden organizar las tablas, las relaciones entre ellas y las diversas claves primarias y foráneas.

**Figura 1**

### Modelo de datos relacional



N

ota. Ejemplo de un modelo de datos relacional. Oracle. (s.f.). Oracle Data Modeler [Software]. Oracle. <https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/>

**Nota:** este diagrama es solo un ejemplo y no refleja las entidades específicas ni las relaciones que encontrarás en el caso del Instituto Nacional del Deporte (IND) de Chile. Tu tarea será desarrollar un modelo que esté alineado con los requisitos del caso presentado, siguiendo las estructuras de datos y relaciones que apoyen las funciones operacionales del IND.

Ahora, veamos los pasos a seguir para poder realizar la actividad:

### Paso 1: Elaboración del Script de creación de las tablas del modelo

- Examina el modelo de base de datos proporcionado y crea un script SQL para la creación de las tablas.

- Define las restricciones (constraints) de Clave Primaria (PK), Clave Foránea (FK) y Clave Única (UN) para todas las tablas, asegurándote de que cada restricción tenga un nombre representativo.
- Asigna los tipos de datos y tamaños adecuados a las columnas de acuerdo con el modelo y las necesidades del caso ficticio.
- Identifica si faltan tablas que sean necesarias para normalizar completamente el modelo y créalas según sea necesario.

### **Paso 2: Implementación de autoincremento**

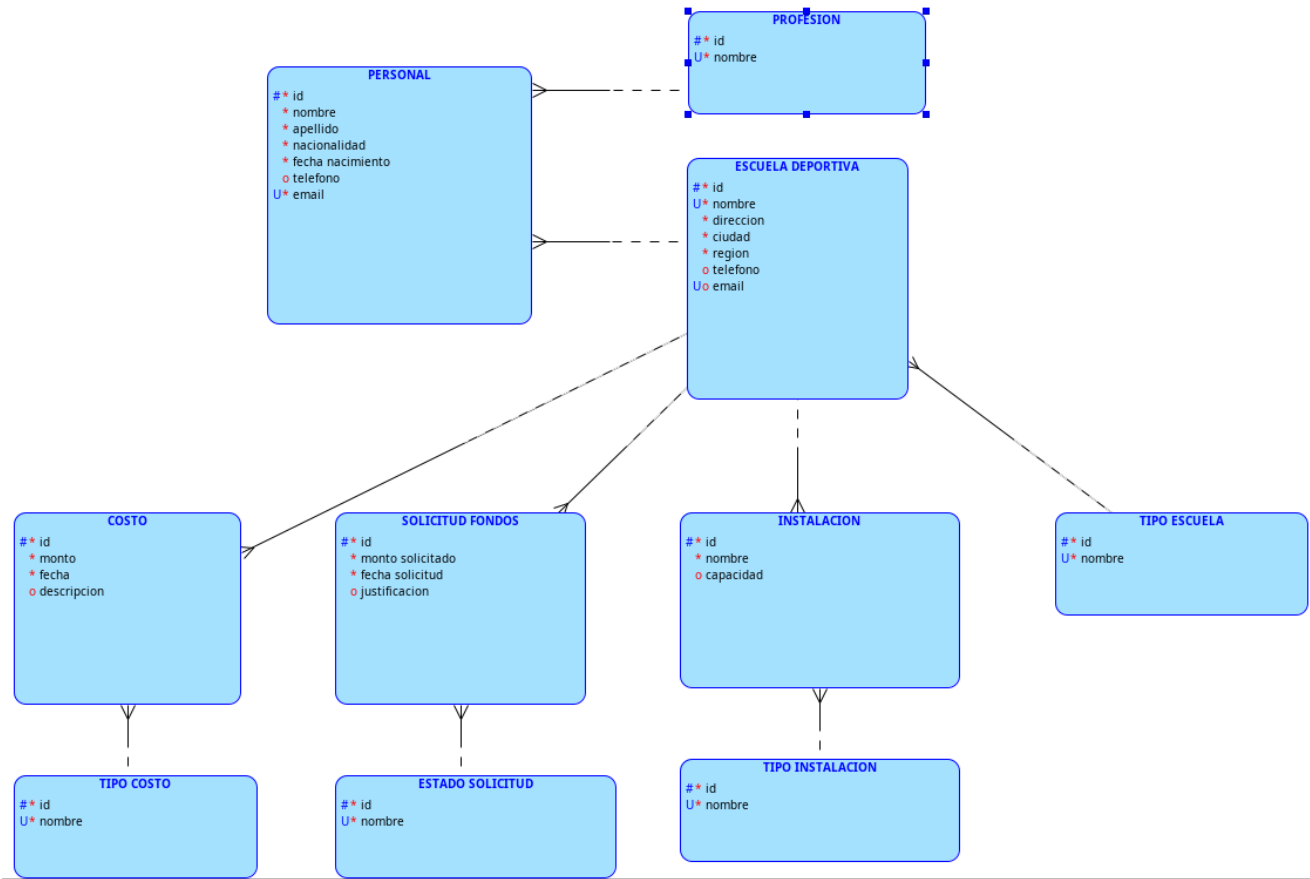
Modifica al menos dos claves primarias en el modelo para que sean autoincrementables, facilitando así la inserción de registros.

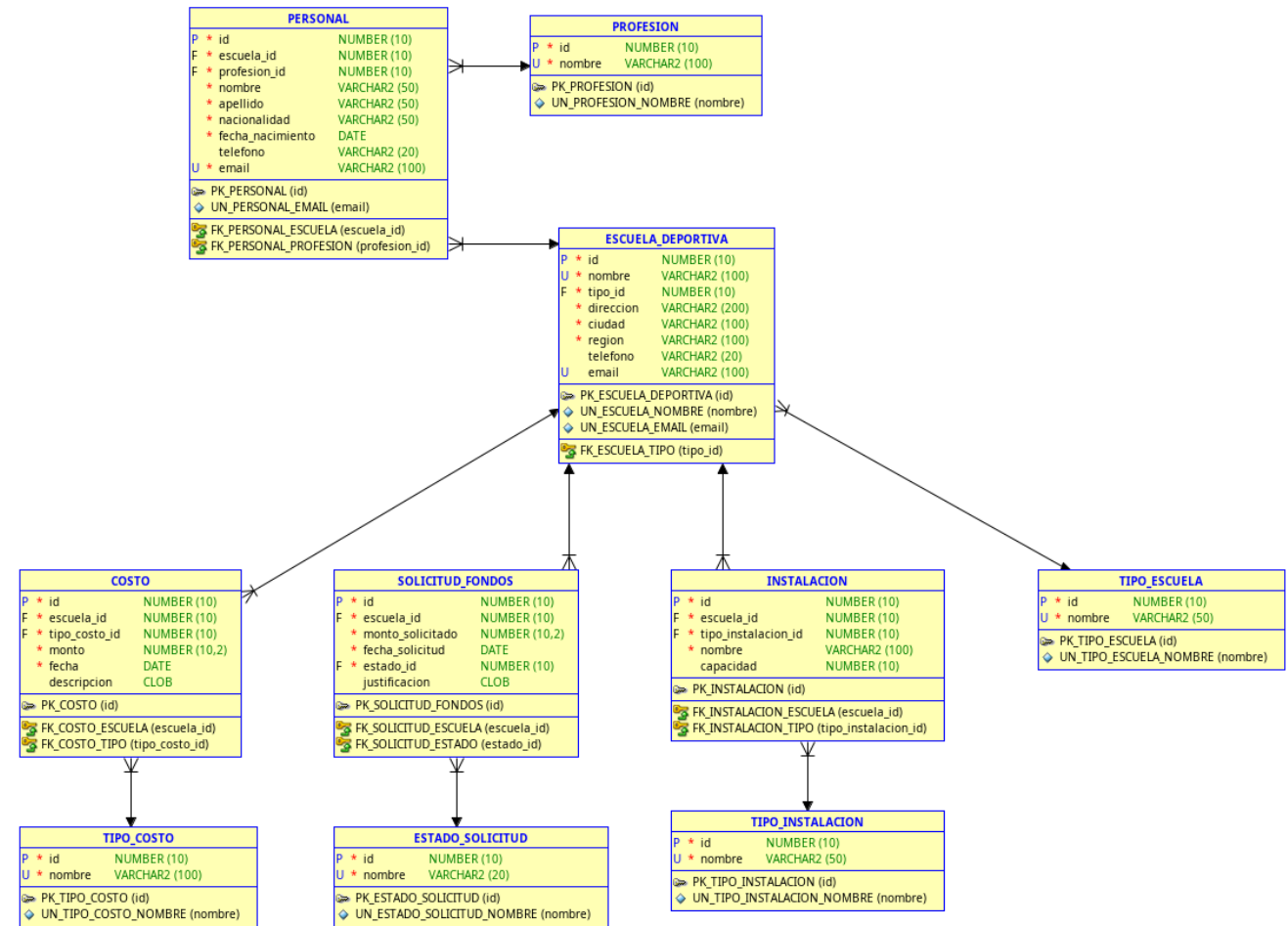
### **Paso 3: Poblado de las tablas**

- Inserta al menos 4 registros en cada tabla utilizando el script de creación, aplicando la secuencia autoincrementable en al menos dos tablas.
- Considera la relevancia y coherencia de los datos.

### **Paso 4: Consultas simples para demostrar poblado**

- Crea y ejecuta consultas SQL para demostrar que las tablas han sido pobladas adecuadamente.
- Las consultas deben ser capaces de mostrar todos los registros, filtrar según condiciones específicas y realizar cálculos simples si es necesario.





## Paso 1 - Scripts y Autoincrementable

```
CREATE SEQUENCE seq_costo START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_escuela_deportiva START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_estado_solicitud START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_instalacion START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_personal START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_profesion START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_solicitud_fondos START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_tipo_costo START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_tipo_escuela START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE seq_tipo_instalacion START WITH 1 INCREMENT BY 1;
```

```
CREATE TABLE costo (
    id            NUMBER(10) DEFAULT seq_costo.NEXTVAL NOT NULL,
    escuela_id    NUMBER(10) NOT NULL,
    tipo_costo_id NUMBER(10) NOT NULL,
    monto         NUMBER(10, 2) NOT NULL,
    fecha         DATE NOT NULL,
    descripcion   CLOB
)
LOGGING;
```

```
ALTER TABLE costo ADD CONSTRAINT pk_costo PRIMARY KEY ( id );
```

```
CREATE TABLE escuela_deportiva (
    id            NUMBER(10) DEFAULT seq_escuela_deportiva.NEXTVAL NOT NULL,
    nombre        VARCHAR2(100) NOT NULL,
    tipo_id       NUMBER(10) NOT NULL,
    direccion     VARCHAR2(200) NOT NULL,
    ciudad        VARCHAR2(100) NOT NULL,
    region        VARCHAR2(100) NOT NULL,
    telefono      VARCHAR2(20),
    email         VARCHAR2(100)
)
LOGGING;
```

```
ALTER TABLE escuela_deportiva ADD CONSTRAINT pk_escuela_deportiva PRIMARY KEY ( id );
ALTER TABLE escuela_deportiva ADD CONSTRAINT un_escuela_email UNIQUE ( email );
ALTER TABLE escuela_deportiva ADD CONSTRAINT un_escuela_nombre UNIQUE ( nombre );
```

```
CREATE TABLE estado_solicitud (
    id            NUMBER(10) DEFAULT seq_estado_solicitud.NEXTVAL NOT NULL,
    nombre        VARCHAR2(20) NOT NULL
)
LOGGING;
```

```
ALTER TABLE estado_solicitud ADD CONSTRAINT pk_estado_solicitud PRIMARY KEY ( id );
ALTER TABLE estado_solicitud ADD CONSTRAINT un_estado_solicitud_nombre UNIQUE ( nombre );
```

```

CREATE TABLE instalacion (
    id                NUMBER(10) DEFAULT seq_instalacion.NEXTVAL NOT NULL,
    escuela_id        NUMBER(10) NOT NULL,
    tipo_instalacion_id NUMBER(10) NOT NULL,
    nombre            VARCHAR2(100) NOT NULL,
    capacidad          NUMBER(10)
)
LOGGING;

ALTER TABLE instalacion ADD CONSTRAINT pk_instalacion PRIMARY KEY ( id );

CREATE TABLE personal (
    id                NUMBER(10) DEFAULT seq_personal.NEXTVAL NOT NULL,
    escuela_id        NUMBER(10) NOT NULL,
    profesion_id       NUMBER(10) NOT NULL,
    nombre            VARCHAR2(50) NOT NULL,
    apellido          VARCHAR2(50) NOT NULL,
    nacionalidad       VARCHAR2(50) NOT NULL,
    fecha_nacimiento  DATE NOT NULL,
    telefono          VARCHAR2(20),
    email             VARCHAR2(100) NOT NULL
)
LOGGING;

ALTER TABLE personal ADD CONSTRAINT pk_personal PRIMARY KEY ( id );
ALTER TABLE personal ADD CONSTRAINT un_personal_email UNIQUE ( email );

CREATE TABLE profesion (
    id                NUMBER(10) DEFAULT seq_profesion.NEXTVAL NOT NULL,
    nombre            VARCHAR2(100) NOT NULL
)
LOGGING;

ALTER TABLE profesion ADD CONSTRAINT pk_profesion PRIMARY KEY ( id );
ALTER TABLE profesion ADD CONSTRAINT un_profesion_nombre UNIQUE ( nombre );

CREATE TABLE solicitud_fondos (
    id                NUMBER(10) DEFAULT seq_solicitud_fondos.NEXTVAL NOT NULL,
    escuela_id        NUMBER(10) NOT NULL,
    monto_solicitado  NUMBER(10, 2) NOT NULL,
    fecha_solicitud   DATE NOT NULL,
    estado_id         NUMBER(10) NOT NULL,
    justificacion     CLOB
)
LOGGING;

ALTER TABLE solicitud_fondos ADD CONSTRAINT pk_solicitud_fondos PRIMARY KEY ( id );

CREATE TABLE tipo_costo (
    id                NUMBER(10) DEFAULT seq_tipo_costo.NEXTVAL NOT NULL,
    nombre            VARCHAR2(100) NOT NULL
)
LOGGING;

ALTER TABLE tipo_costo ADD CONSTRAINT pk_tipo_costo PRIMARY KEY ( id );
ALTER TABLE tipo_costo ADD CONSTRAINT un_tipo_costo_nombre UNIQUE ( nombre );

```



```

CREATE TABLE tipo_escuela (
    id      NUMBER(10) DEFAULT seq_tipo_escuela.NEXTVAL NOT NULL,
    nombre VARCHAR2(50) NOT NULL
)
LOGGING;

ALTER TABLE tipo_escuela ADD CONSTRAINT pk_tipo_escuela PRIMARY KEY ( id );
ALTER TABLE tipo_escuela ADD CONSTRAINT un_tipo_escuela_nombre UNIQUE ( nombre );

CREATE TABLE tipo_instalacion (
    id      NUMBER(10) DEFAULT seq_tipo_instalacion.NEXTVAL NOT NULL,
    nombre VARCHAR2(50) NOT NULL
)
LOGGING;

ALTER TABLE tipo_instalacion ADD CONSTRAINT pk_tipo_instalacion PRIMARY KEY ( id );
ALTER TABLE tipo_instalacion ADD CONSTRAINT un_tipo_instalacion_nombre UNIQUE ( nombre );

ALTER TABLE costo
    ADD CONSTRAINT fk_costo_escuela FOREIGN KEY ( escuela_id )
        REFERENCES escuela_deportiva ( id )
    NOT DEFERRABLE;

ALTER TABLE costo
    ADD CONSTRAINT fk_costo_tipo FOREIGN KEY ( tipo_costo_id )
        REFERENCES tipo_costo ( id )
    NOT DEFERRABLE;

ALTER TABLE escuela_deportiva
    ADD CONSTRAINT fk_escuela_tipo FOREIGN KEY ( tipo_id )
        REFERENCES tipo_escuela ( id )
    NOT DEFERRABLE;

ALTER TABLE instalacion
    ADD CONSTRAINT fk_instalacion_escuela FOREIGN KEY ( escuela_id )
        REFERENCES escuela_deportiva ( id )
    NOT DEFERRABLE;

ALTER TABLE instalacion
    ADD CONSTRAINT fk_instalacion_tipo FOREIGN KEY ( tipo_instalacion_id )
        REFERENCES tipo_instalacion ( id )
    NOT DEFERRABLE;

ALTER TABLE personal
    ADD CONSTRAINT fk_personal_escuela FOREIGN KEY ( escuela_id )
        REFERENCES escuela_deportiva ( id )
    NOT DEFERRABLE;

ALTER TABLE personal
    ADD CONSTRAINT fk_personal_profesion FOREIGN KEY ( profesion_id )
        REFERENCES profesion ( id )
    NOT DEFERRABLE;

ALTER TABLE solicitud_fondos
    ADD CONSTRAINT fk_solicitud_escuela FOREIGN KEY ( escuela_id )
        REFERENCES escuela_deportiva ( id )
    NOT DEFERRABLE;

```

```
ALTER TABLE solicitud_fondos
ADD CONSTRAINT fk_solicitud_estado FOREIGN KEY ( estado_id )
REFERENCES estado_solicitud ( id )
NOT DEFERRABLE;
```

### Paso 3 - Poblado de tablas

```
INSERT INTO TIPO_ESCUELA (nombre) VALUES ('Fútbol');
INSERT INTO TIPO_ESCUELA (nombre) VALUES ('Tenis');
INSERT INTO TIPO_ESCUELA (nombre) VALUES ('Natación');
INSERT INTO TIPO_ESCUELA (nombre) VALUES ('Atletismo');
```

```
INSERT INTO ESCUELA_DEPORTIVA (nombre, tipo_id, direccion, ciudad, region, telefono, email) VALUES ('Escuela de Fútbol Santiago', 1, 'Av. Grecia 2001', 'Santiago', 'Región Metropolitana', '223545700', 'futbol.santiago@escuela.cl');
INSERT INTO ESCUELA_DEPORTIVA (nombre, tipo_id, direccion, ciudad, region, telefono, email) VALUES ('Club de Tenis Viña del Mar', 2, 'Av. Borgoño 16180', 'Viña del Mar', 'Valparaíso', '322658000', 'tenis.vina@escuela.cl');
INSERT INTO ESCUELA_DEPORTIVA (nombre, tipo_id, direccion, ciudad, region, telefono, email) VALUES ('Centro de Natación Concepción', 3, 'Av. Chacabuco 1085', 'Concepción', 'Biobío', '412621900', 'natacion.conce@escuela.cl');
INSERT INTO ESCUELA_DEPORTIVA (nombre, tipo_id, direccion, ciudad, region, telefono, email) VALUES ('Escuela de Atletismo Antofagasta', 4, 'Av. Angamos 0610', 'Antofagasta', 'Antofagasta', '552637000', 'atletismo.anto@escuela.cl');
```

```
INSERT INTO PROFESION (nombre) VALUES ('Entrenador');
INSERT INTO PROFESION (nombre) VALUES ('Preparador Físico');
INSERT INTO PROFESION (nombre) VALUES ('Fisioterapeuta');
INSERT INTO PROFESION (nombre) VALUES ('Nutricionista');
```

```
INSERT INTO PERSONAL (escuela_id, profesion_id, nombre, apellido, nacionalidad, fecha_nacimiento, telefono, email) VALUES (1, 1, 'Juan', 'Pérez', 'Chilena', TO_DATE('1980-05-15', 'YYYY-MM-DD'), '956781234', 'juan.perez@escuela.cl');
INSERT INTO PERSONAL (escuela_id, profesion_id, nombre, apellido, nacionalidad, fecha_nacimiento, telefono, email) VALUES (2, 2, 'María', 'González', 'Chilena', TO_DATE('1985-09-22', 'YYYY-MM-DD'), '987654321', 'maria.gonzalez@escuela.cl');
INSERT INTO PERSONAL (escuela_id, profesion_id, nombre, apellido, nacionalidad, fecha_nacimiento, telefono, email) VALUES (3, 3, 'Carlos', 'Soto', 'Chilena', TO_DATE('1982-11-30', 'YYYY-MM-DD'), '912345678', 'carlos.soto@escuela.cl');
INSERT INTO PERSONAL (escuela_id, profesion_id, nombre, apellido, nacionalidad, fecha_nacimiento, telefono, email) VALUES (4, 4, 'Ana', 'Muñoz', 'Chilena', TO_DATE('1988-07-12', 'YYYY-MM-DD'), '965432109', 'ana.munoz@escuela.cl');
```

```
INSERT INTO TIPO_COSTO (nombre) VALUES ('Salarios');
INSERT INTO TIPO_COSTO (nombre) VALUES ('Equipamiento');
INSERT INTO TIPO_COSTO (nombre) VALUES ('Mantenimiento');
INSERT INTO TIPO_COSTO (nombre) VALUES ('Servicios Básicos');
```

```
INSERT INTO COSTO (escuela_id, tipo_costo_id, monto, fecha, descripcion) VALUES (1, 1, 5000000, TO_DATE('2023-01-15', 'YYYY-MM-DD'), 'Pago mensual de salarios');
INSERT INTO COSTO (escuela_id, tipo_costo_id, monto, fecha, descripcion) VALUES (2, 2, 2000000, TO_DATE('2023-02-01', 'YYYY-MM-DD'), 'Compra de equipamiento deportivo');
INSERT INTO COSTO (escuela_id, tipo_costo_id, monto, fecha, descripcion) VALUES (3, 3, 800000, TO_DATE('2023-03-10', 'YYYY-MM-DD'), 'Mantenimiento de piscina');
INSERT INTO COSTO (escuela_id, tipo_costo_id, monto, fecha, descripcion) VALUES (4, 4, 500000, TO_DATE('2023-04-05', 'YYYY-MM-DD'), 'Pago de servicios básicos');
```

```
INSERT INTO TIPO_INSTALACION (nombre) VALUES ('Cancha de Fútbol');
INSERT INTO TIPO_INSTALACION (nombre) VALUES ('Cancha de Tenis');
INSERT INTO TIPO_INSTALACION (nombre) VALUES ('Piscina Olímpica');
INSERT INTO TIPO_INSTALACION (nombre) VALUES ('Pista de Atletismo');
```

```

INSERT INTO INSTALACION (escuela_id, tipo_instalacion_id, nombre, capacidad) VALUES (1, 1, 'Cancha Principal
Estadio Nacional', 500);
INSERT INTO INSTALACION (escuela_id, tipo_instalacion_id, nombre, capacidad) VALUES (2, 2, 'Cancha Central Club de
Tenis', 200);
INSERT INTO INSTALACION (escuela_id, tipo_instalacion_id, nombre, capacidad) VALUES (3, 3, 'Piscina Olímpica
Municipal', 300);
INSERT INTO INSTALACION (escuela_id, tipo_instalacion_id, nombre, capacidad) VALUES (4, 4, 'Pista Atlética Estadio
Regional', 400);

INSERT INTO ESTADO_SOLICITUD (nombre) VALUES ('Pendiente');
INSERT INTO ESTADO_SOLICITUD (nombre) VALUES ('Aprobada');
INSERT INTO ESTADO_SOLICITUD (nombre) VALUES ('Rechazada');
INSERT INTO ESTADO_SOLICITUD (nombre) VALUES ('En revisión');

INSERT INTO SOLICITUD_FONDOS (escuela_id, monto_solicitado, fecha_solicitud, estado_id, justificacion) VALUES (1,
10000000, TO_DATE('2023-01-10', 'YYYY-MM-DD'), 1, 'Mejora de instalaciones de fútbol');
INSERT INTO SOLICITUD_FONDOS (escuela_id, monto_solicitado, fecha_solicitud, estado_id, justificacion) VALUES (2,
5000000, TO_DATE('2023-02-15', 'YYYY-MM-DD'), 2, 'Compra de equipamiento de tenis');
INSERT INTO SOLICITUD_FONDOS (escuela_id, monto_solicitado, fecha_solicitud, estado_id, justificacion) VALUES (3,
8000000, TO_DATE('2023-03-20', 'YYYY-MM-DD'), 3, 'Renovación del sistema de filtración de la piscina');
INSERT INTO SOLICITUD_FONDOS (escuela_id, monto_solicitado, fecha_solicitud, estado_id, justificacion) VALUES (4,
6000000, TO_DATE('2023-04-25', 'YYYY-MM-DD'), 4, 'Actualización de equipos de medición para atletismo');

```

#### Paso 4: Consultas simples para demostrar poblado

Mostrar todos los registros de la tabla escuela\_deportiva

```
SELECT * FROM escuela_deportiva;
```

Mostrar todas las solicitudes de fondos junto con el nombre de la escuela y el estado de la solicitud

```

SELECT sf.id, ed.nombre AS escuela, sf.monto_solicitado, sf.fecha_solicitud, es.nombre AS estado
FROM solicitud_fondos sf
JOIN escuela_deportiva ed ON sf.escuela_id = ed.id
JOIN estado_solicitud es ON sf.estado_id = es.id;

```

Filtrar las instalaciones por una escuela específica (ejemplo: escuela con ID 1)

```

SELECT i.id, i.nombre, i.capacidad, te.nombre AS tipo_instalacion
FROM instalacion i
JOIN tipo_instalacion te ON i.tipo_instalacion_id = te.id
WHERE i.escuela_id = 1;

```

Mostrar todos los costos de una escuela deportiva y su tipo (ejemplo: escuela con ID 1)

```
SELECT c.id, c.monto, c.fecha, tc.nombre AS tipo_costo
FROM costo c
JOIN tipo_costo tc ON c.tipo_costo_id = tc.id
WHERE c.escuela_id = 1;
```

Contar cuántos empleados (personal) hay en una escuela específica (ejemplo: escuela con ID 1)

```
SELECT COUNT(*) AS total_empleados
FROM personal
WHERE escuela_id = 1;
```

Mostrar el nombre, profesión y email de todos los empleados (personal) de una escuela específica

```
SELECT p.nombre, p.apellido, pr.nombre AS profesion, p.email
FROM personal p
JOIN profesion pr ON p.profesion_id = pr.id
WHERE p.escuela_id = 1;
```

Mostrar todos los tipos de costos y el total gastado por tipo de costo

```
SELECT tc.nombre AS tipo_costo, SUM(c.monto) AS total_gastado
FROM costo c
JOIN tipo_costo tc ON c.tipo_costo_id = tc.id
GROUP BY tc.nombre;
```

Filtrar las solicitudes de fondos que están en un estado específico (ejemplo: "Pendiente")

```
SELECT sf.id, ed.nombre AS escuela, sf.monto_solicitado, sf.fecha_solicitud, es.nombre AS estado
FROM solicitud_fondos sf
JOIN escuela_deportiva ed ON sf.escuela_id = ed.id
JOIN estado_solicitud es ON sf.estado_id = es.id
WHERE es.nombre = 'Pendiente';
```

Calcular el costo promedio de todas las instalaciones de una escuela deportiva específica (ejemplo: escuela con ID 1)

```
SELECT AVG(c.monto) AS costo_promedio
FROM costo c
WHERE c.escuela_id = 1;
```

Mostrar todas las escuelas deportivas junto con el número de instalaciones que poseen

```
SELECT ed.nombre AS escuela, COUNT(i.id) AS total_instalaciones
```

FROM escuela\_deportiva ed

```
LEFT JOIN instalacion i ON ed.id = i.escuela id
```

GROUP BY ed.nombre;

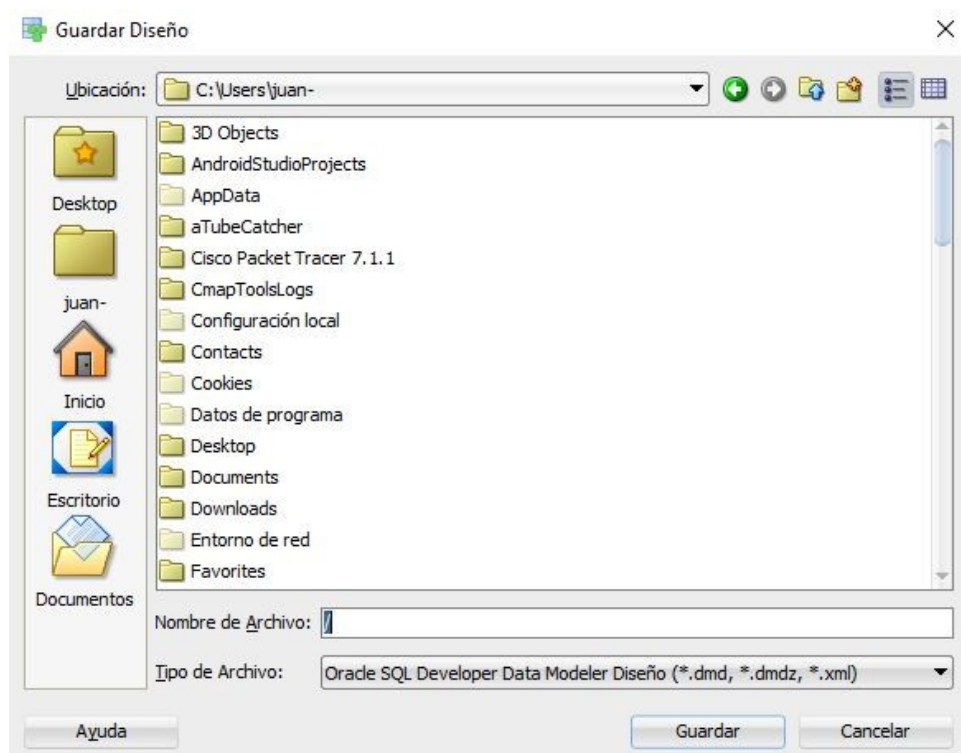
**Paso 5:** para realizar el ejercicio, tendrás que utilizar la herramienta Oracle SQL Developer, disponible de descarga a través del siguiente enlace:

<https://www.oracle.com/database/sqldeveloper/technologies/download/>

Además, tendrás que descargar el resultado. Para ello, deberás hacer clic en la opción Guardar como... del menú Archivo, esto despliega el submenú que se ilustra en la siguiente figura:

## Figura 2

## Cómo guardar un archivo en SQL



Nota. Ejemplo de guardado de archivo SQL. Oracle. (s.f.). *SQL Developer* [Software].

Oracle. <https://www.oracle.com/cloud/database/sqldeveloper/technologies/sql-data-modeler/>

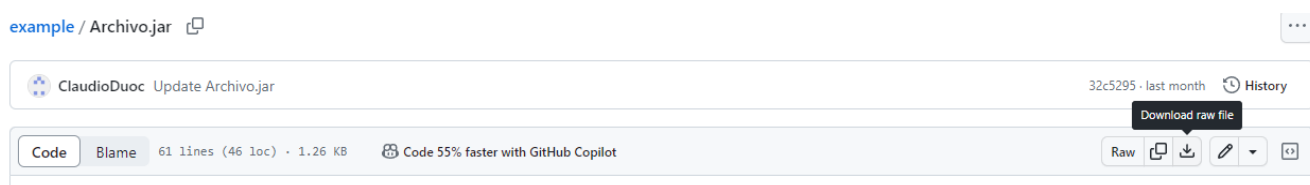
**Paso 6:** el archivo descargado desde SQL deberás subirlo al repositorio GitHub. Si no has creado tu cuenta aún, puedes hacerlo a través del siguiente enlace:

<https://github.com/>

Una vez subido el archivo a GitHub, deberás descargar el archivo comprimido .raw desde tu repositorio, tal como se muestra en la imagen:

### Figura 3

#### Archivo .raw en GitHub



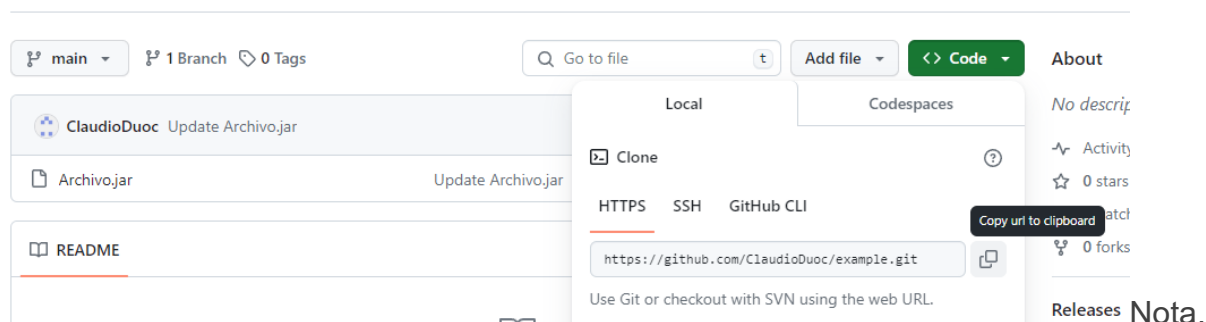
Nota. Descarga de archivo desde repositorio GitHub. GitHub (s.f.). *GitHub*.

<https://github.com/>

Posteriormente, desde el repositorio, deberás generar un enlace de tu proyecto:

### Figura 4

#### Enlace de proyecto GitHub



Ejemplo de dónde se extrae un enlace en GitHub. GitHub (s.f.). *GitHub*. <https://github.com/>

Deja en este apartado el enlace de tu repositorio GitHub:

**Paso 7:** una vez adjunta tu respuesta y enlace, no olvides comprimir este documento y el archivo SQL en un archivo .ZIP o .RAR, el cual deberás subir al AVA.

Reservados todos los derechos Fundación Instituto Profesional Duoc UC. No se permite copiar, reproducir, reeditar, descargar, publicar, emitir, difundir, de forma total o parcial la presente obra, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio (electrónico, mecánico, fotocopia, grabación u otros) sin autorización previa y por escrito de Fundación Instituto Profesional Duoc UC. La infracción de dichos derechos puede constituir un delito contra la propiedad intelectual.