

Semana 9

Modelamiento de Bases de Datos (PRY2204)

Formato de respuesta

| **Nombre estudiante:** | Nataly Chacón |
| --- | --- |
| **Asignatura:** | **Carrera:** |
| **Profesor:** | **Fecha:** |

# Descripción de la actividad

La Evaluación Final Transversal de la asignatura Modelamiento de Bases de Datos, consiste en la elaboración de un encargo de manera individual donde deberás entregar, a partir de un caso, el desarrollo desde el comienzo con el Modelo conceptual, modelo lógico y finalizar con un modelo relacional para brindar solución a los requerimientos del cliente. El producto de esta evaluación se dividirá en dos partes. Para el desarrollo y cumplimiento de la parte I, tendrás que entregar el modelo desarrollado en todas sus etapas con las herramientas Oracle SQL Developer y DBMS Base de Datos Oracle (en Oracle CLOUD) para la creación y poblamiento de las tablas del Modelo Relacional (MR) y adjuntar el Script de implementación.

En la parte II, deberás realizar la presentación de tu trabajo final mediante la grabación de un video. En ella, deberás mostrar y explicar detalladamente el proceso de desarrollo de tu proyecto, cómo funciona tu implementación y las características principales de lo realizado.

## Instrucciones específicas

Para el desarrollo de la EFT, tendrás que analizar el siguiente caso estudio:

“El Ministerio de las Culturas, las Artes y el Patrimonio, una entidad del Estado de Chile, desempeña un papel crucial en el diseño, formulación e implementación de políticas, planes y programas orientados al desarrollo cultural, artístico y patrimonial. Su misión es promover una evolución armónica y equitativa en estas áreas a lo largo de todo el territorio nacional.

Recientemente, se ha observado una disminución en el interés de la población juvenil por participar y preservar las tradiciones culturales chilenas, tales como la danza, la música folclórica, el desarrollo artístico indígena y la interpretación de instrumentos nativos. Esta tendencia amenaza con debilitar la rica herencia cultural del país y disminuir la diversidad de sus expresiones artísticas.

Ante esta situación, el Ministerio ha decidido tomar medidas proactivas para revitalizar el interés y la participación de los jóvenes en estas tradiciones culturales. La estrategia incluye potenciar los proyectos concursables disponibles para las academias de arte en cada región, con especial énfasis en las disciplinas de música y danza. El objetivo es mejorar la difusión de estas artes, generar campañas y actividades en los colegios, y trabajar para aumentar la calidad e infraestructura relacionada, todo con el fin de fortalecer la cultura como parte integral del desarrollo humano.

Se espera que, a través de estos esfuerzos, se logre no solo un aumento en la participación juvenil en las actividades culturales sino también una mejora en la calidad de vida y la salud mental de los jóvenes, especialmente en tiempos desafiantes.

La convocatoria lanzada por el Ministerio tiene como fin proporcionar financiamiento a proyectos que promuevan la valorización, promoción y fomento del arte popular, originario o de las culturas tradicionales de Chile. Las academias de artes interesadas en participar deben completar y enviar formularios oficiales con información detallada sobre diversos aspectos de su funcionamiento, incluyendo costos de personal, locaciones, insumos, permisos necesarios, seguridad, producción, información general de la academia, montajistas, entre otros.

Como especialista en datos del área de tecnologías de la información, se te ha encomendado la tarea de diseñar e implementar una base de datos que permita registrar y gestionar eficientemente toda la información relevante enviada por las Academias de Artes en los formularios oficiales. Este sistema de base de datos debe ser capaz de adaptarse a las necesidades dinámicas del Ministerio y las academias, facilitando la toma de decisiones y el seguimiento del impacto de los proyectos financiados.

**Las reglas de negocio y especificaciones técnicas son las siguientes:**

* Inicialmente, el sistema se centrará en las academias de Danza, Canto, e Intérprete de instrumentos musicales, con planes de expandirse a otras artes en el futuro.
* Las municipalidades pueden ofrecer espacios a múltiples academias, que a su vez, pueden impartir diversos cursos en diferentes turnos dentro del mismo recinto.
* Los profesores contratados pueden ser de planta o a honorarios, con diferencias significativas en términos de remuneración y obligaciones legales.
* Es vital registrar los turnos de trabajo de los profesores, dado que pueden laborar en más de una academia.
* Cada academia debe tener un director responsable, cuya posición puede cambiar con el tiempo.
* Para participar en los proyectos concursables, cada academia de arte deberá completar el formulario de postulación, proporcionando detalles vitales para su identificación y la evaluación de su elegibilidad para financiamiento:

**Figura 1**

*Ejemplo de formulario de postulación*

Tabla

Descripción generada automáticamente

Este formulario sirve como un apoyo visual para ilustrar la información que las academias de artes proporcionarían durante un proceso de postulación real (no se requiere completarlo como parte de esta actividad).

Para la implementación, se requiere la creación de cuatro tablas principales en la primera etapa, considerando tanto tablas fuertes como débiles, la asignación adecuada de tipos de datos, restricciones (PK, FK, UN, NOT NULL) y el poblamiento inicial de datos para demostrar la funcionalidad del sistema”.

**Ahora que has leído el caso, deberás desarrollar lo requerido a través de los siguientes pasos:**

**Paso 1 – Diseño del Modelo conceptual**

Inicia interpretando el caso del Ministerio para diseñar un Modelo Entidad-Relación (MER). Este modelo debe representar las entidades, sus relaciones y atributos, basado en el análisis del escenario dado, donde el Ministerio busca impulsar la participación juvenil en tradiciones culturales.

**Paso 2 – Transformación del Modelo conceptual al Modelo lógico**

Transforma el MER en un Modelo Relacional Normalizado (MR), especificando cómo se organizarán los datos en la base de datos. Esto incluye la definición de claves primarias, claves foráneas y restricciones de integridad.

**Paso 3 – Desarrollo del Modelo Relacional**

Desarrolla el Modelo Relacional, considerando aspectos como la cantidad de profesores por academia.

**Paso 4 – Implementación de la base de datos**

* Utiliza Oracle SQL Developer y DBMS Base de Datos en Oracle CLOUD para llevar a cabo la creación de las tablas del Modelo Relacional (MR).

Podrás realizar la descarga de SQL a través del siguiente enlace:

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

* Implementa las tablas utilizando sentencias SQL de creación (CREATE TABLE), define las relaciones entre tablas con sentencias de clave foránea (FOREIGN KEY), y llena las tablas con datos de prueba mediante sentencias de inserción (INSERT INTO).

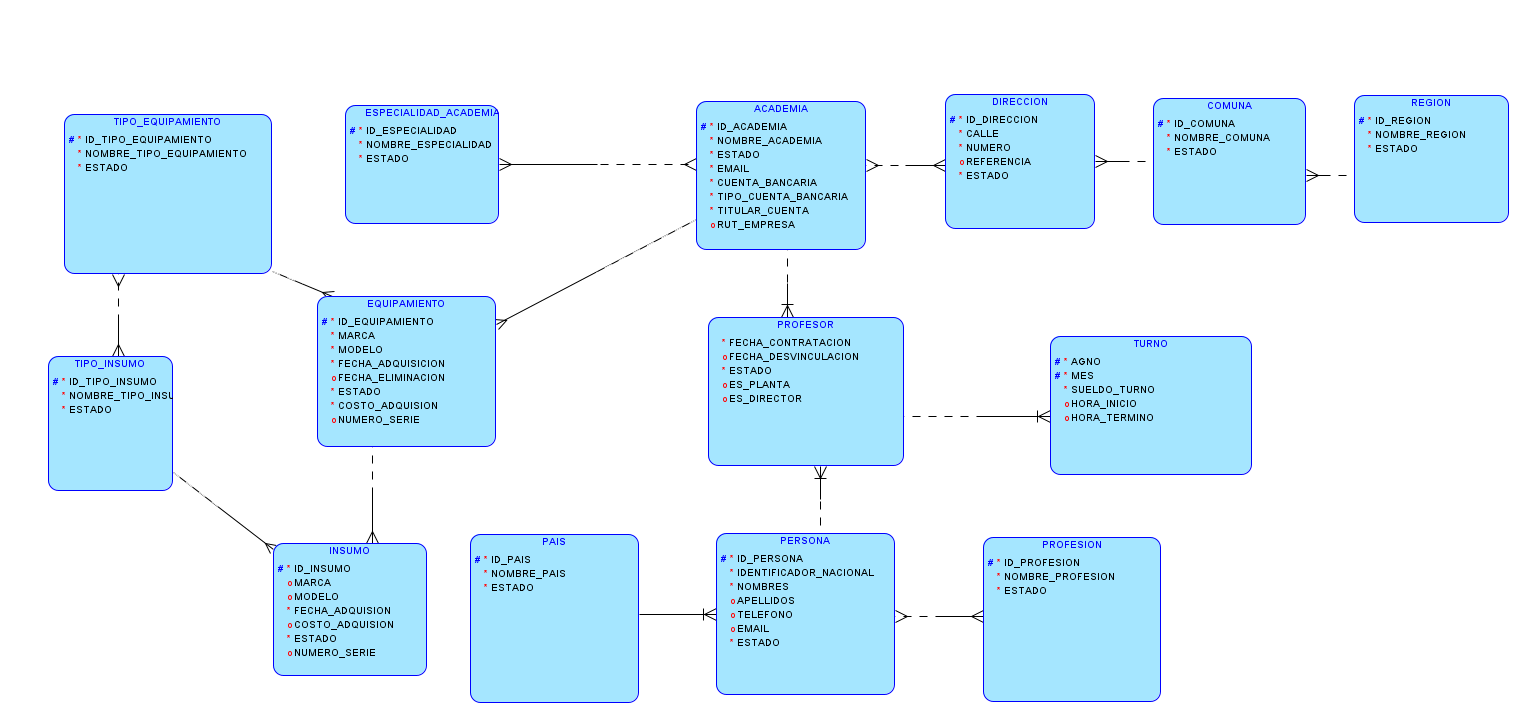
**Paso 5 – Script de implementación**

Adjunta el script de SQL utilizado para la creación y poblamiento de las tablas, asegurando que esté claro, comentado y bien organizado. Este script debe incluir:

* La construcción de cuatro tablas del modelo relacional normalizado, especificando dos tablas fuertes y dos débiles.
* Ejecución secuencial de comandos DDL en el orden adecuado.
* Asignación de tipos de datos adecuados a cada columna.
* Incorporación de restricciones de claves primarias (PK), claves foráneas (FK), únicas (UN), y not null, con nombres representativos.
* Poblamiento inicial de las tablas con al menos dos filas por tabla, utilizando objetos secuencia para asignar las llaves primarias.

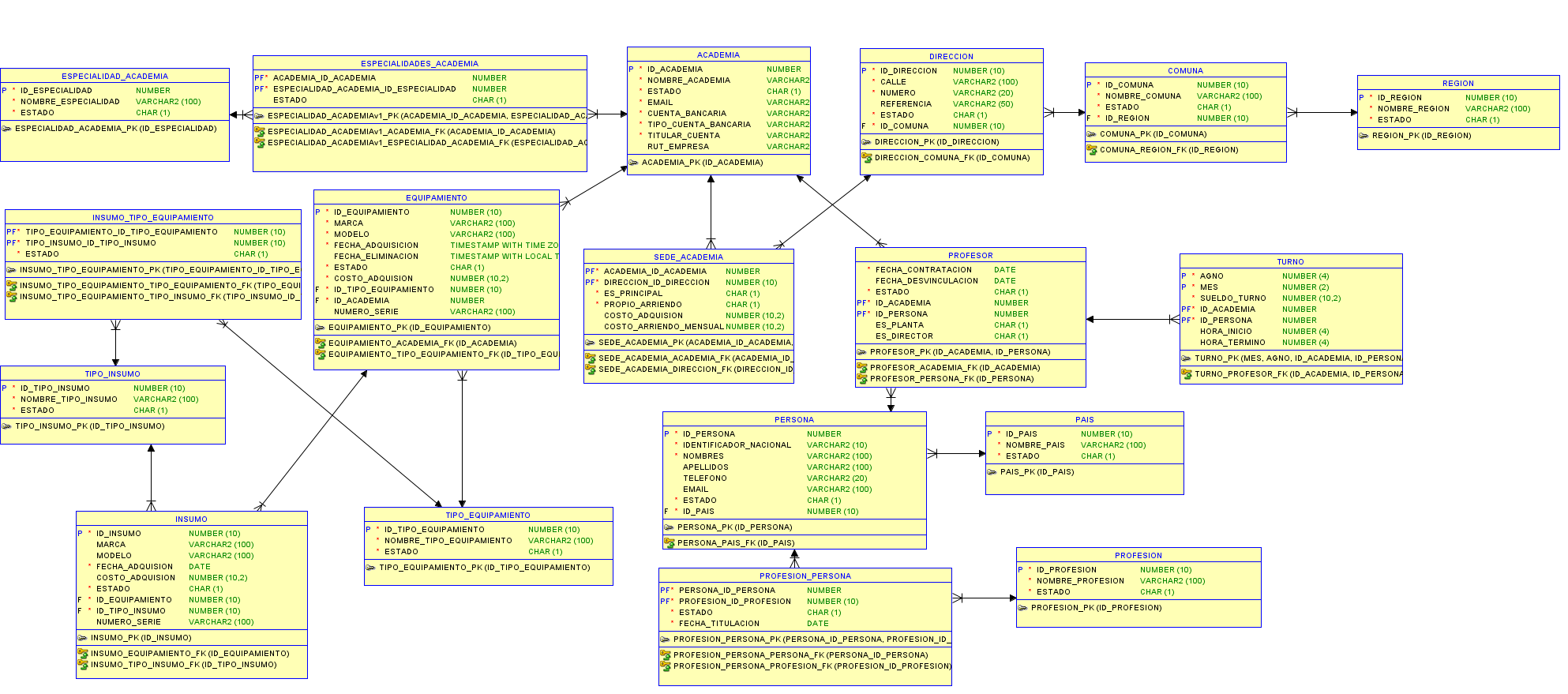
Cuando tengas lista tu solución, adjunta las siguientes evidencias:

* Modelo Entidad Relación (MER)



**(Este espacio es aproximado; ajústalo según tus necesidades)**

* Modelo Relacional Normalizado



**(Este espacio es aproximado; ajústalo según tus necesidades)**

* Script con las sentencias SQL para la creación y poblamiento de las tablas.

CREATE TABLE academia (

    id\_academia          NUMBER NOT NULL,

    nombre\_academia      VARCHAR2(100) NOT NULL,

    estado               CHAR(1) NOT NULL,

    email                VARCHAR2(100) NOT NULL,

    cuenta\_bancaria      VARCHAR2(100) NOT NULL,

    tipo\_cuenta\_bancaria VARCHAR2(10) NOT NULL,

    titular\_cuenta       VARCHAR2(100) NOT NULL,

    rut\_empresa          VARCHAR2(10)

);

ALTER TABLE academia ADD CONSTRAINT academia\_pk PRIMARY KEY ( id\_academia );

CREATE TABLE comuna (

    id\_comuna     NUMBER(10) NOT NULL,

    nombre\_comuna VARCHAR2(100) NOT NULL,

    estado        CHAR(1) NOT NULL,

    id\_region     NUMBER(10) NOT NULL

);

ALTER TABLE comuna ADD CONSTRAINT comuna\_pk PRIMARY KEY ( id\_comuna );

CREATE TABLE direccion (

    id\_direccion NUMBER(10) NOT NULL,

    calle        VARCHAR2(100) NOT NULL,

    numero       VARCHAR2(20) NOT NULL,

    referencia   VARCHAR2(50),

    estado       CHAR(1) NOT NULL,

    id\_comuna    NUMBER(10) NOT NULL

);

ALTER TABLE direccion ADD CONSTRAINT direccion\_pk PRIMARY KEY ( id\_direccion );

CREATE TABLE equipamiento (

    id\_equipamiento      NUMBER(10) NOT NULL,

    marca                VARCHAR2(100) NOT NULL,

    modelo               VARCHAR2(100) NOT NULL,

    fecha\_adquisicion    TIMESTAMP WITH TIME ZONE NOT NULL,

    fecha\_eliminacion    TIMESTAMP WITH LOCAL TIME ZONE,

    estado               CHAR(1) NOT NULL,

    costo\_adquision      NUMBER(10, 2) NOT NULL,

    id\_tipo\_equipamiento NUMBER(10) NOT NULL,

    id\_academia          NUMBER NOT NULL,

    numero\_serie         VARCHAR2(100)

);

ALTER TABLE equipamiento ADD CONSTRAINT equipamiento\_pk PRIMARY KEY ( id\_equipamiento );

CREATE TABLE especialidad\_academia (

    id\_especialidad     NUMBER NOT NULL,

    nombre\_especialidad VARCHAR2(100) NOT NULL,

    estado              CHAR(1) NOT NULL

);

ALTER TABLE especialidad\_academia ADD CONSTRAINT especialidad\_academia\_pk PRIMARY KEY ( id\_especialidad );

CREATE TABLE especialidades\_academia (

    academia\_id\_academia                  NUMBER NOT NULL,

--  ERROR: Column name length exceeds maximum allowed length(30)

    especialidad\_academia\_id\_especialidad NUMBER NOT NULL,

    estado                                CHAR(1)

);

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 .dmd*



*Nota.* Ejemplo de guardado de archivo .dmd. Oracle. (s.f.). Oracle SQL Developer Data

ALTER TABLE especialidades\_academia ADD CONSTRAINT especialidad\_academiav1\_pk PRIMARY KEY ( academia\_id\_academia,

                                                                                            especialidad\_academia\_id\_especialidad );

CREATE TABLE insumo (

    id\_insumo       NUMBER(10) NOT NULL,

    marca           VARCHAR2(100),

    modelo          VARCHAR2(100),

    fecha\_adquision DATE NOT NULL,

    costo\_adquision NUMBER(10, 2),

    estado          CHAR(1) NOT NULL,

    id\_equipamiento NUMBER(10) NOT NULL,

    id\_tipo\_insumo  NUMBER(10) NOT NULL,

    numero\_serie    VARCHAR2(100)

);

ALTER TABLE insumo ADD CONSTRAINT insumo\_pk PRIMARY KEY ( id\_insumo );

CREATE TABLE insumo\_tipo\_equipamiento (

--  ERROR: Column name length exceeds maximum allowed length(30)

    tipo\_equipamiento\_id\_tipo\_equipamiento NUMBER(10) NOT NULL,

    tipo\_insumo\_id\_tipo\_insumo             NUMBER(10) NOT NULL,

    estado                                 CHAR(1) NOT NULL

);

ALTER TABLE insumo\_tipo\_equipamiento ADD CONSTRAINT insumo\_tipo\_equipamiento\_pk PRIMARY KEY ( tipo\_equipamiento\_id\_tipo\_equipamiento,

                                                                                              tipo\_insumo\_id\_tipo\_insumo );

CREATE TABLE pais (

    id\_pais     NUMBER(10) NOT NULL,

    nombre\_pais VARCHAR2(100) NOT NULL,

    estado      CHAR(1) NOT NULL

);

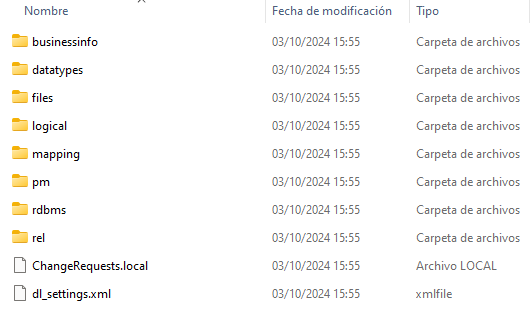
ALTER TABLE pais ADD CONSTRAINT pais\_pk PRIMARY KEY ( id\_pais );

Modeler [Software]. Oracle. [https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/](https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/%E2%80%AF)

**Figura 3**

*Archivo .dmd y subcarpetas*





*Nota.* Ejemplo de guardado de archivo .dmd. Oracle. (s.f.). Oracle SQL Developer Data Modeler [Software]. Oracle. [https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/](https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/%E2%80%AF)

También tendrás que guardar el archivo .ddl solicitado:

**Figura 4**

CREATE TABLE persona (

    id\_persona             NUMBER NOT NULL,

    identificador\_nacional VARCHAR2(10) NOT NULL,

    nombres                VARCHAR2(100) NOT NULL,

    apellidos              VARCHAR2(100),

    telefono               VARCHAR2(20),

    email                  VARCHAR2(100),

    estado                 CHAR(1) NOT NULL,

    id\_pais                NUMBER(10) NOT NULL

);

ALTER TABLE persona ADD CONSTRAINT persona\_pk PRIMARY KEY ( id\_persona );

CREATE TABLE profesion (

    id\_profesion     NUMBER(10) NOT NULL,

    nombre\_profesion VARCHAR2(100) NOT NULL,

    estado           CHAR(1) NOT NULL

);

ALTER TABLE profesion ADD CONSTRAINT profesion\_pk PRIMARY KEY ( id\_profesion );

CREATE TABLE profesion\_persona (

    persona\_id\_persona     NUMBER NOT NULL,

    profesion\_id\_profesion NUMBER(10) NOT NULL,

    estado                 CHAR(1) NOT NULL,

    fecha\_titulacion       DATE NOT NULL

);

ALTER TABLE profesion\_persona ADD CONSTRAINT profesion\_persona\_pk PRIMARY KEY ( persona\_id\_persona,

                                                                                profesion\_id\_profesion );

CREATE TABLE profesor (

    fecha\_contratacion   DATE NOT NULL,

    fecha\_desvinculacion DATE,

    estado               CHAR(1) NOT NULL,

    id\_academia          NUMBER NOT NULL,

    id\_persona           NUMBER NOT NULL,

    es\_planta            CHAR(1),

    es\_director          CHAR(1)

);

ALTER TABLE profesor ADD CONSTRAINT profesor\_pk PRIMARY KEY ( id\_academia,

                                                              id\_persona );

CREATE TABLE region (

    id\_region     NUMBER(10) NOT NULL,

    nombre\_region VARCHAR2(100) NOT NULL,

    estado        CHAR(1) NOT NULL

);

ALTER TABLE region ADD CONSTRAINT region\_pk PRIMARY KEY ( id\_region );

CREATE TABLE sede\_academia (

    academia\_id\_academia   NUMBER NOT NULL,

    direccion\_id\_direccion NUMBER(10) NOT NULL,

    es\_principal           CHAR(1) NOT NULL,

    propio\_arriendo        CHAR(1) NOT NULL,

    costo\_adquision        NUMBER(10, 2),

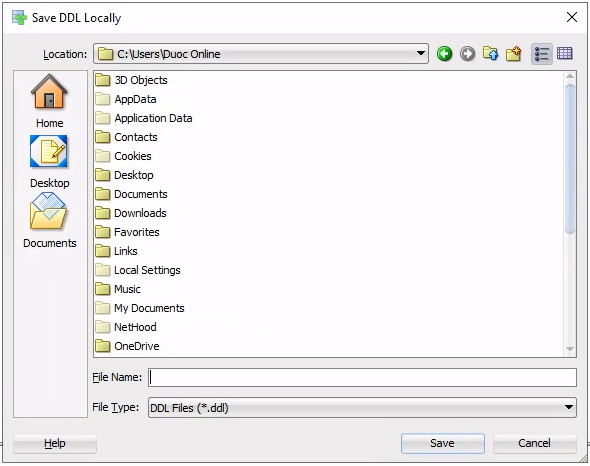
    costo\_arriendo\_mensual NUMBER(10, 2)

);

ALTER TABLE sede\_academia ADD CONSTRAINT sede\_academia\_pk PRIMARY KEY ( academia\_id\_academia,

                                                                        direccion\_id\_direccion );

Cómo guardar archivo .ddl



*Nota.* Ejemplo de guardado de archivo .ddl. Oracle. (s.f.). Oracle SQL Developer [Software]. Oracle. [https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/](https://www.oracle.com/cl/database/sqldeveloper/technologies/sql-data-modeler/%E2%80%AF)

También tendrás que guardar el archivo .sql:

**Figura 5**

ALTER TABLE profesor ADD CONSTRAINT profesor\_pk PRIMARY KEY ( id\_academia,

                                                              id\_persona );

CREATE TABLE region (

    id\_region     NUMBER(10) NOT NULL,

    nombre\_region VARCHAR2(100) NOT NULL,

    estado        CHAR(1) NOT NULL

);

ALTER TABLE region ADD CONSTRAINT region\_pk PRIMARY KEY ( id\_region );

CREATE TABLE sede\_academia (

    academia\_id\_academia   NUMBER NOT NULL,

    direccion\_id\_direccion NUMBER(10) NOT NULL,

    es\_principal           CHAR(1) NOT NULL,

    propio\_arriendo        CHAR(1) NOT NULL,

    costo\_adquision        NUMBER(10, 2),

    costo\_arriendo\_mensual NUMBER(10, 2)

);

ALTER TABLE sede\_academia ADD CONSTRAINT sede\_academia\_pk PRIMARY KEY ( academia\_id\_academia,

                                                                        direccion\_id\_direccion );

CREATE TABLE tipo\_equipamiento (

    id\_tipo\_equipamiento     NUMBER(10) NOT NULL,

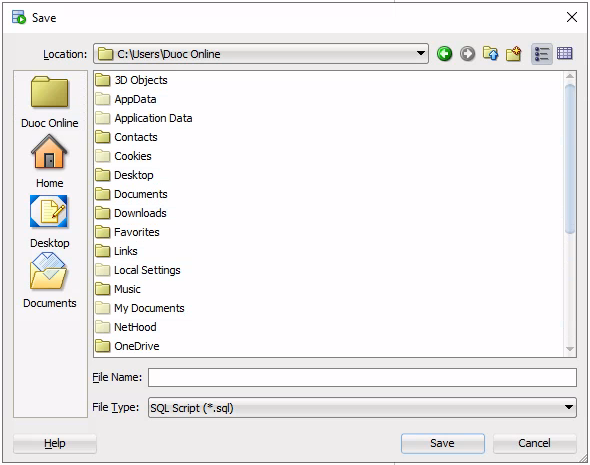
    nombre\_tipo\_equipamiento VARCHAR2(100) NOT NULL,

    estado                   CHAR(1) NOT NULL

);

ALTER TABLE tipo\_equipamiento ADD CONSTRAINT tipo\_equipamiento\_pk PRIMARY KEY ( id\_tipo\_equipamiento );

*Cómo guardar un archivo .sql*



*Nota.* Ejemplo de guardado de archivo .sql. Oracle. (s.f.). *Oracle SQL Developer* [Software]. Oracle. <https://www.oracle.com/cl/database/sqldeveloper/>

CREATE TABLE tipo\_insumo (

    id\_tipo\_insumo     NUMBER(10) NOT NULL,

    nombre\_tipo\_insumo VARCHAR2(100) NOT NULL,

    estado             CHAR(1) NOT NULL

);

ALTER TABLE tipo\_insumo ADD CONSTRAINT tipo\_insumo\_pk PRIMARY KEY ( id\_tipo\_insumo );

CREATE TABLE turno (

    agno         NUMBER(4) NOT NULL,

    mes          NUMBER(2) NOT NULL,

    sueldo\_turno NUMBER(10, 2) NOT NULL,

    id\_academia  NUMBER NOT NULL,

    id\_persona   NUMBER NOT NULL,

    hora\_inicio  NUMBER(4),

    hora\_termino NUMBER(4)

);

ALTER TABLE turno

    ADD CONSTRAINT turno\_pk PRIMARY KEY ( mes,

                                          agno,

                                          id\_academia,

                                          id\_persona );

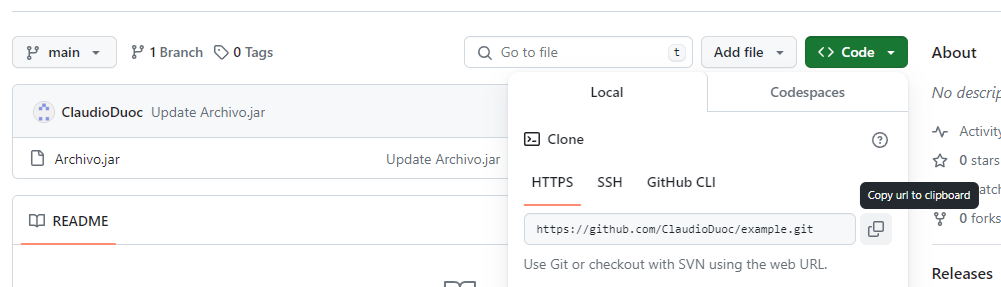
**Paso 6:** Los archivos creados en SQL Developer Data Modeler (.dmd y .ddl), con sus carpetas y subcarpetas y SQL Developer (.sql), deberás subirlos al repositorio GitHub.

* Si no has creado tu cuenta aún, puedes hacerlo a través del siguiente enlace: <https://github.com/>

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

**Figura 4**

*Enlace de proyecto GitHub*





*Nota.* Ejemplo genérico 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:

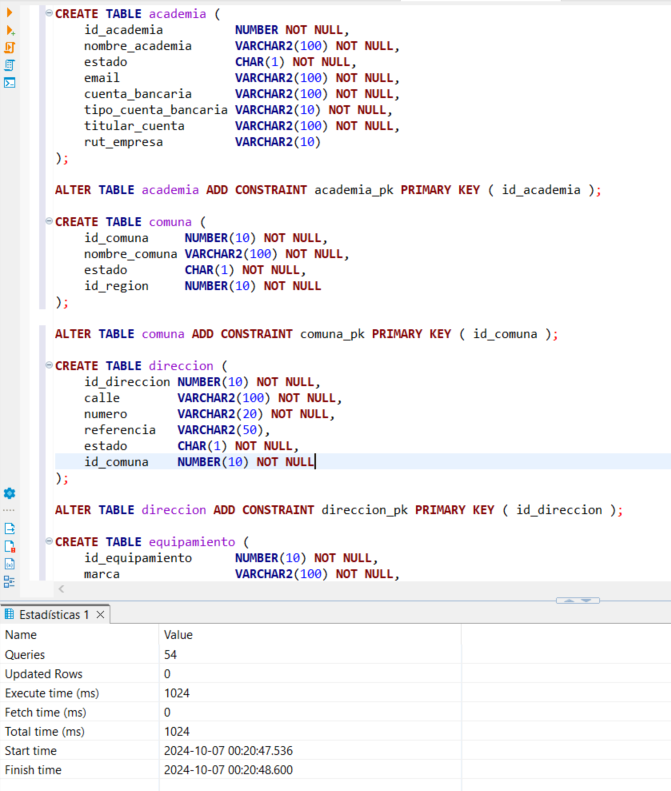


**Paso 7: Presentación**

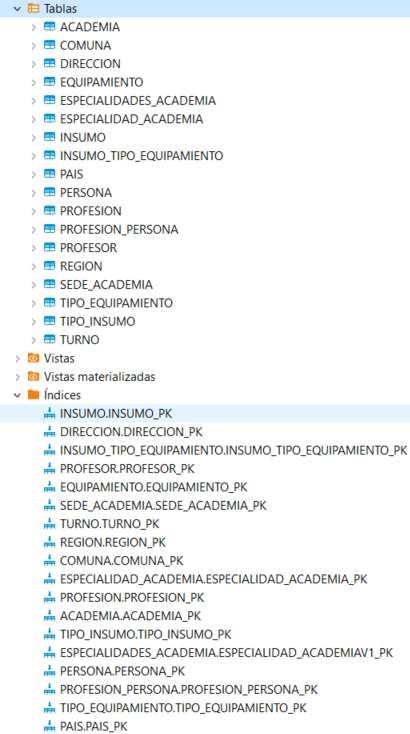
Para este paso, tendrás que realizar una presentación en video de tu proyecto con una duración máxima de 5 minutos. En este video, esperamos que compartas el proceso completo de los modelos y scripts desarrollados acorde con el caso planteado. **Adjunta el video al momento de subir tu actividad.**

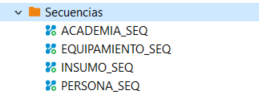
Puedes descargar el instructivo de Kaltura en el siguiente enlace: [https://ava.duoc.cl/bbcswebdav/xid-3242677\_1](https://ava.duoc.cl/bbcswebdav/xid-3242677_1 )

**Evidencia ejecución correcta del script:**

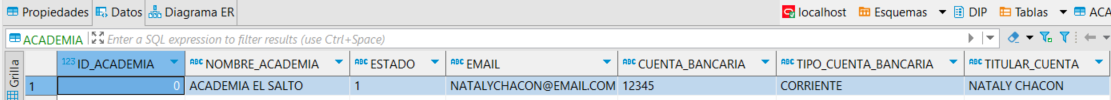


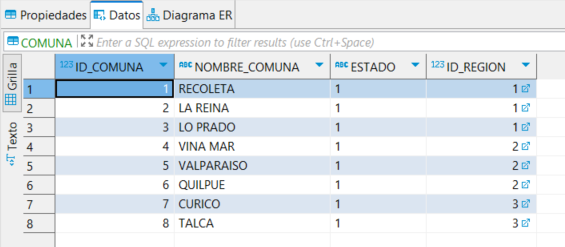
**Y se crean todas las tablas y objetos:**

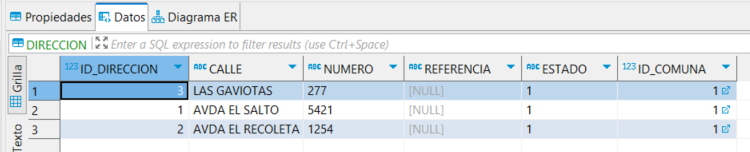


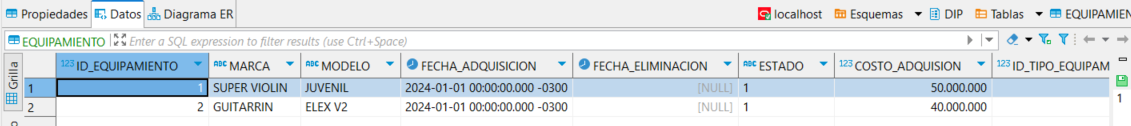


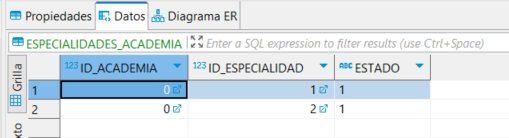
**Evidencia poblado de tablas:**

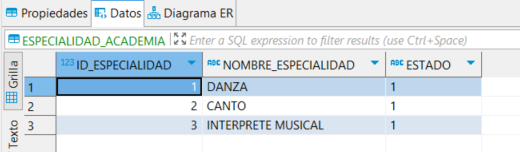


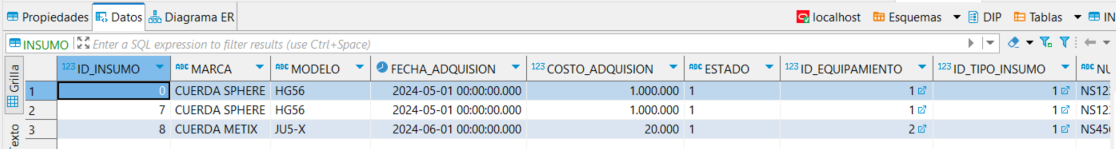


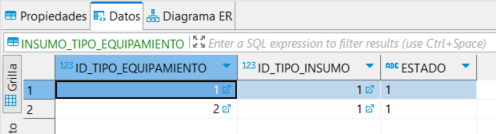




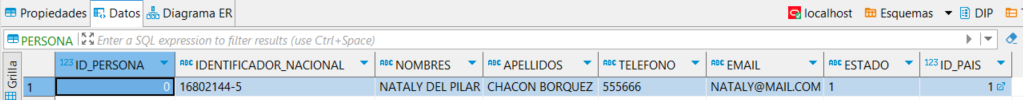


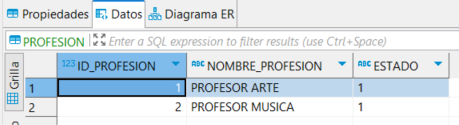


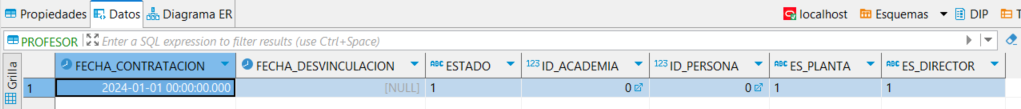


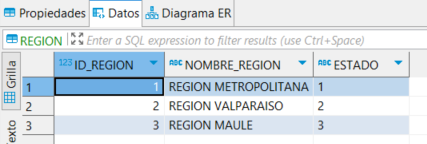


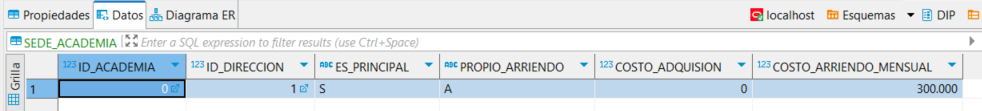


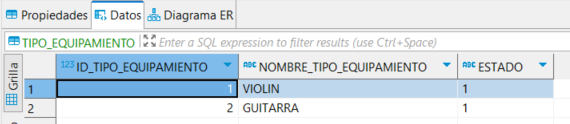


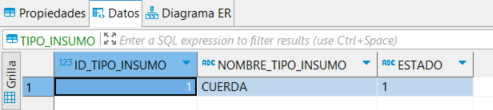


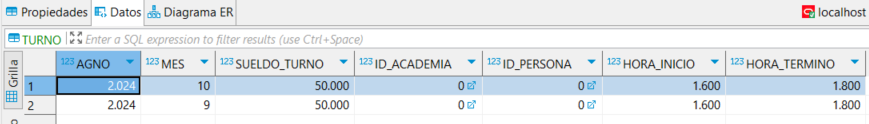












**Link a repositorio GIT:**

[**https://github.com/natchacon/EFT\_BASEDATOS\_NATALY\_CHACON.git**](https://github.com/natchacon/EFT_BASEDATOS_NATALY_CHACON.git)

**Paso 8:** para finalizar, sube este documento, los archivos .dmd, .ddl y .sql comprimidos en formato .zip o .rar.

En definitiva, el archivo comprimido debe contener:

* Modelo Entidad Relación (MER).
* Modelo Relacional Normalizado (MR).
* Script con las sentencias SQL para la creación de las tablas y poblamiento de las tablas.



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.