

Proyecto Final SQL

Academia Baile Mambolaye

**Alumno:** Ernesto Vega

**Profesor:** Miguel Rodas

**Tutor:** Juan Rabazzi

**1.- Introducción:**

Mediante el presente documento contiene toda la información recopilada, scripts generados y todos los requerimientos que han sido solicitados para presentar el Proyecto Final SQL Comisión 31270 de CoderHouse.

El modelo de ER y Los scripts SQL que se detallan en este proyecto se encuentra en el directorio **Scripts** en el presente repositorio, al inicio de cada descripción se indica el nombre del respectivo archivo con su extensión y dirección URL, esto con el fin de facilitar su ubicación.

**2.- Objetivo:**

Crear una Bases de Datos relacional para llevar un mejor registro de alumnos, actividades, clases impartidas y demás personas involucradas en la Academia Mambolaye.

**3.- Modelo de Negocio:**

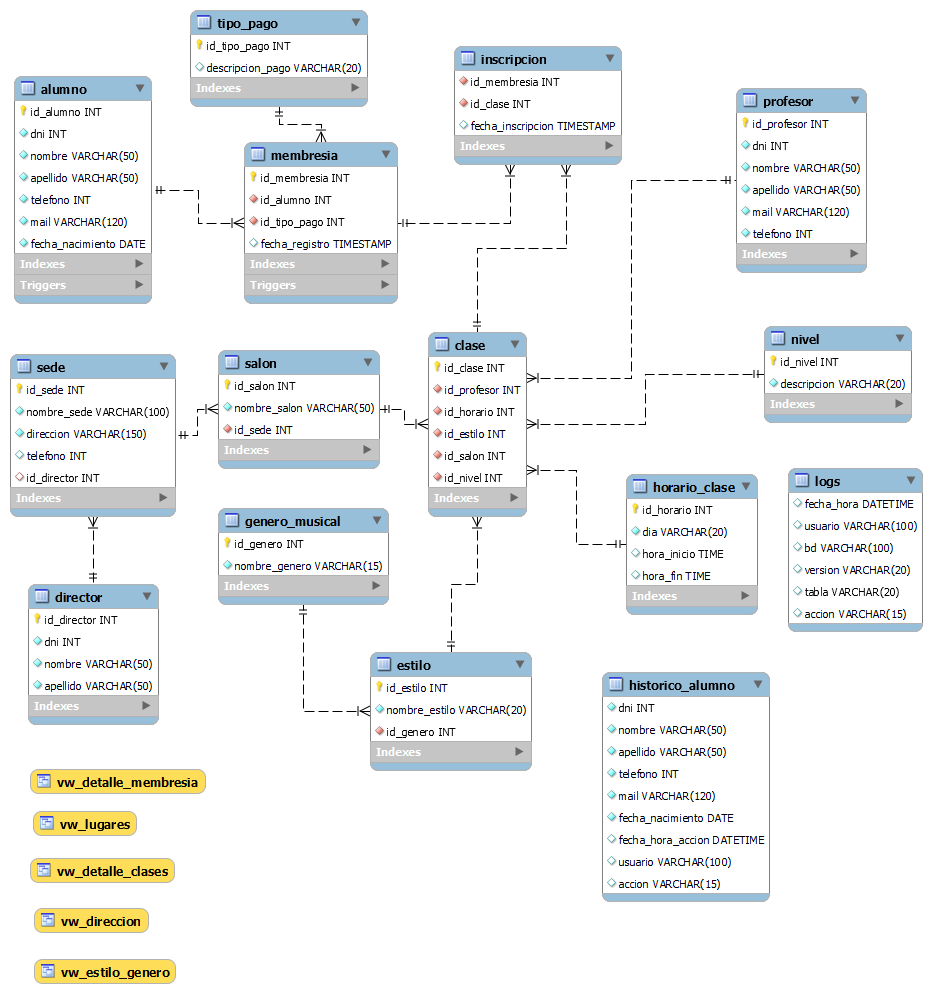
La Academia Mambolaye es una escuela de baile que se dedica a enseñar diversos estilos de baile como Mambo (Salsa on2), Mambo Lady Style, Bachata Estilo Pacheco y próximamente se irán agregando más estilos.

Mambolaye se fundó en Venezuela y actualmente cuenta con un solo espacio de enseñanza en CABA ubicado en Almagro, con miras de expansión. El nombre de la academia está compuesto por dos palabras **Mambo** que es el estilo de baile basado en la Salsa y **Laye** que significa calle, la fusión de estas dos se defino como *el estilo de baile con un toque propio de la persona que solo se obtiene fuera de salones de baile*.

**4.-** **Situación Problemática:**

Actualmente la academia no cuenta con un sistema de registros para los actores ni actividades que se realizan en ella, se maneja de manera manual y la automatización es casi nula, por ende, con el presente proyecto aplicando un modelo de Bases de Datos Relacional se busca resolver esta problemática y así poder llevar un mejor registro de todo lo que se realiza en la institución.

**5.- Diagrama E-R**



**6.- Descripción de tablas:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **alumno (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_alumno | int | autoinc | x | x |  | Datos de los alumnos de la academia |
| dni | int |  | x |  |  |
| nombre\_alumno | varchar | 50 | x |  |  |
| apellido\_alumno | varchar | 50 | x |  |  |
| telefono | int |  | x |  |  |
| mail | varchar | 120 | x |  |  |
| fecha\_nacimiento | date | 10 | x |  |  |
|  |  |  |  |  |  |  |
| **tipo\_pago (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_tipo\_pago | int | autoinc | x | x |  | Informacion de los metodos de pago |
| descripcion\_pago | varchar | 20 |  |  |  |
|  |  |  |  |  |  |  |
| **Membresia (TT)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_membresia | int | autoinc | x | x |  | Datos de la membresia de cada alumno |
| id\_alumno | int |  | x |  | x |
| id\_tipo\_pago | int |  | x |  | x |
| fecha\_registro | Timestamp |  | x |  | x |
|  |  |  |  |  |  |  |
| **profesor (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NUL** | **PK** | **FK** | **Descripcion** |
| id\_profesor | int | autoinc | x | x |  | Contiene informacion de los profesores de baile |
| dni | int |  | x |  |  |
| nombre | varchar | 50 | x |  |  |
| apellido | varchar | 50 | x |  |  |
| mail | varchar | 120 | x |  |  |
| telefono | int |  | x |  |  |
|  |  |  |  |  |  |  |
| **genero\_musical (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_genero | int | autoinc | x | x |  | Descripcion del genero musical |
| nombre\_genero | varchar | 15 | x |  |  |
|  |  |  |  |  |  |  |
| **estilo (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_estilo | int | autoinc | x | x |  | Estilos de baile relacionados con los generos musicales |
| nombre\_estilo | varchar | 20 | x |  |  |
| id\_genero | int |  | x |  | x |
|  |  |  |  |  |  |  |
| **horario\_clase (TD)** | | | | | | |
| **Campos** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_horario | int | autoinc | x | x |  | Tabla de horarios de clases |
| dia | varchar | 20 | x |  |  |
| hora\_inicio | Time |  |  |  |  |
| hora\_fin | time |  |  |  |  |
|  |  |  |  |  |  |  |
| **director(TD)** | | | | | | |
| **Campos** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_director | int | autoinc | x | x |  | Datos de los directores de las sedes |
| dni | int |  | x |  |  |
| nombre | varchar | 50 | x |  |  |
| apellido | varchar | 50 | x |  |  |
|  |  |  |  |  |  |  |
| **sede (TD)** | | | | | | |
| **Campos** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_sede | int | autoinc | x | x |  | Contiene informacion de la(s) sede(s) de la academia |
| nombre\_sede | varchar | 100 | x |  |  |
| direccion | varchar | 150 | x |  |  |
| telefono | int |  |  |  |  |
| id\_director | int |  |  |  | x |
|  |  |  |  |  |  |  |
| **nivel (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_genero | int | autoinc | x | x |  | Niveles de baile |
| descripcion | varchar | 20 | x |  |  |
|  |  |  |  |  |  |  |
| **salon (TD)** | | | | | | |
| **Campos** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_salon | int | autoinc | x | x |  | Contiene datos de salon de practica |
| nombre\_salon | varchar | 50 | x |  |  |
| id\_sede | int |  | x |  | x |
|  |  |  |  |  |  |  |
| **clase (TD)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_clase | int | autoinc | x | x |  | Contiene informacion sobre las clases que brinda la academia |
| id\_profesor | int |  | x |  | x |
| id\_horario | int |  | x |  | x |
| id\_estilo | int |  | x |  | x |
| id\_salon | int |  | x |  | x |
| id\_nivel | int |  | x |  | x |
|  |  |  |  |  |  |  |
| **inscripcion (TH)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| id\_registro | int | autoinc | x | x |  | Contiene datos de la inscripcion de cada alumno |
| fecha\_registro | datetime | 16 | x |  |  |
| tipo\_pago | varchar | 20 | x |  |  |
| id\_clase | int |  | x |  | x |
| id\_alumno | int |  | x |  | x |
|  |  |  |  |  |  |  |
| **logs (TT)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| fecha\_hora | datetime | autoinc | x | x |  | Contiene informacion sobre las clases que brinda la academia |
| usuario | varchar | 100 | x |  | x |
| bd | varchar | 100 | x |  | x |
| version | varchar | 20 | x |  | x |
| tabla | varchar | 20 | x |  | x |
| accion | varchar | 15 | x |  | x |
|  |  |  |  |  |  |  |
| **historico\_alumnos (TT)** | | | | | | |
| **Campo** | **Tipo** | **Longitud** | **NOT NULL** | **PK** | **FK** | **Descripcion** |
| dni | int |  | x |  |  | Contiene informacion sobre las clases que brinda la academia |
| nombre | varchar | 50 | x |  |  |
| apellido | varchar | 50 | x |  |  |
| telefono | int |  | x |  |  |
| mail | varchar | 120 | x |  |  |
| fecha\_nacimiento | date |  | x |  |  |
| fecha\_hora\_accion | datetime |  | x |  |  |
| usuario | varchar | 100 | x |  |  |
| accion | varchar | 15 | x |  |  |

**7.- Script Objetos BD**

***7.1.- Crear Schema y tablas*** [***Scripts/academia\_mambolaye.sql***](https://github.com/ernieve/SQL_Coder/blob/master/Scripts/academia_mambolaye.sql)

***7.2 Vistas de la BD*** [***Scripts/ScripVisVega.sql***](https://github.com/ernieve/SQL_Coder/blob/master/Scripts/ScripVisVega.sql)

Vista 1: vw\_direccion

**Objetivo**: observar las sedes de la academia y sus respectivos directores. **Tablas**: sede y director

Vista 2: vw\_ estilo\_genero

**Objetivo:** Debido que los estilos de baile pueden variar para un género musical, mediante esta vista se puede obtener el género y sus distintos tipos de estilo. **Tablas:** estilo y genero\_musical

Vista 3: vw\_detalle\_clase

**Objetivo**: La tabla clase está conformada por distintos ID que pertenecen a otras tablas, esto a la vista humana es poco entendible, por ende esta view asocia los distintos ID con su respectiva tablas trayendo la descripción de los mismos y así la información sea más comoda a la vista del usuario. **Tablas**: clase, estilo, profesor, horario\_clase, nivel y salón.

Vista 4: vw\_lugares

**Objetivo:** Observar cuales son los salones que se encuentran en cada sede. **Tablas:** salón y sede

Vista 5: vw\_membresia

**Objetivo:** Observar toda la información de los alumnos que han obtenido una membresía ordenada de manera ascendente. **Tablas:** alumno, membresia y tipo\_pago

***7.3.- Funciones*** [***Scripts/FormatoVega.sql***](https://github.com/ernieve/SQL_Coder/blob/master/Scripts/FormatoVega.sql)

Función 1: cant\_alumnos\_x\_clase

**Objetivo:** Contabilizar cuantos alumnos inscritos en una clase especifica indicando el id de la clase. Se ingresa un entero que corresponde al ID de la clase a consultar y la función realiza un SELECT COUNT de la tabla inscripción.

Función 2: ult\_alumno\_x\_clase

**Objetivo:** Retornar el nombre del último alumno que se registró a una clase en especifica. Se ingresa un entero que corresponde al ID de la clase a consultar y la función retornar una cadena de texto (VARCHAR) con el nombre de la persona.

***7.4.- Stored Procedures*** [***Scripts/* *StoredVega.sql***](https://github.com/ernieve/SQL_Coder/blob/master/Scripts/StoredVega.sql)

Procedimiento 1: sp\_orden\_x\_campo

**Objetivo:** Mostrar datos ordenados de una tabla indicada por el usuario, esto permitirá al usuario que trabaje directamente en la BD ahorrar tiempo y código de sentencia ya que al llamar al SP solo deberá ingresar el nombre de la tabla, la columna y si es en orden ascendente o descendente.

Procedimiento 2: sp\_delete\_alumno

**Objetivo:** Eliminar alumno enviando por parámetro el DNI, el SP ayuda a evitar la posibilidad de eliminar registros por equivocación y omisión de la cláusula WHERE. Cuenta con una validación de longitud por si el usuario no ingresa por completo la secuencia de números que conforman un DNI.

***7.5.- Triggers*** [***Scripts/* *TriggersVega.sql***](https://github.com/ernieve/SQL_Coder/blob/master/Scripts/TriggersVega.sql)***:***

Triggers logs

**Objetivo:** Generar un registro de acciones DML que se puedan realizar sobre las tablas membresía y alumnos.

Triggers Historico

**Objetivo:** Generar una copia de los registros que se insertan o actualizan en la tabla alumnos.

**8.- Script Inserción registros** [**Scripts/ScriptInsVega.sql**](https://github.com/ernieve/SQL_Coder/blob/master/Scripts/ScriptInsVega.sql)

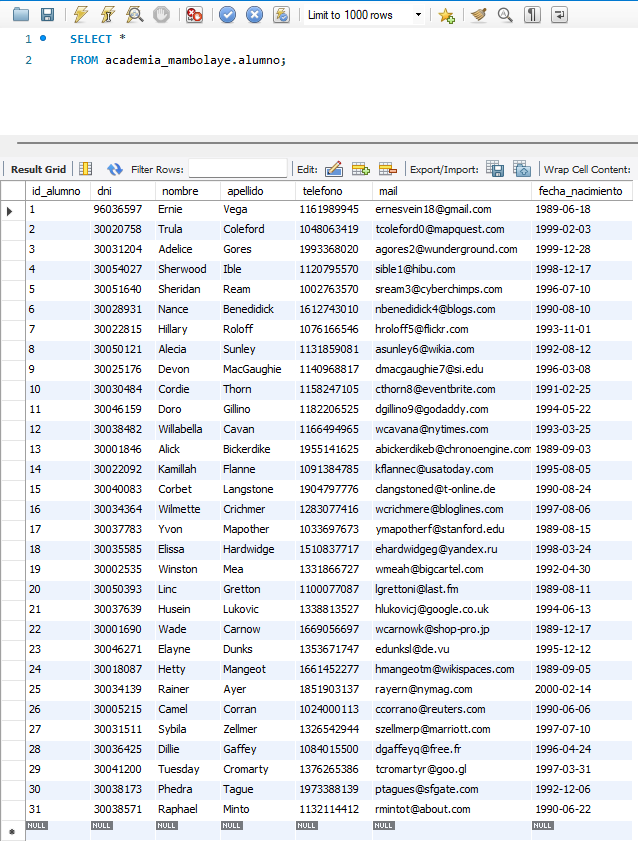
Los registros de la BD fueron cargados mediante script, no se dispuso de algún datasets, motivado que el propósito de este proyecto es hacerlo muy cercano a la realidad actual y planes a futuro de la academia.

**9.- Información de las tablas**

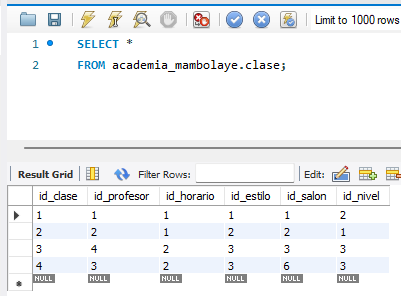
A continuación, se da una imagen de la información que es arrojada por el SGBD MySQL Workbench al consultar las distintas tablas y ejecutar cada objeto generado a lo largo de la cursada

**9.1.- Información de las tablas**

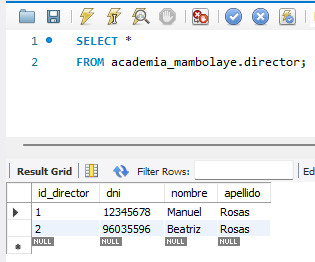
9.1.1.- Alumno



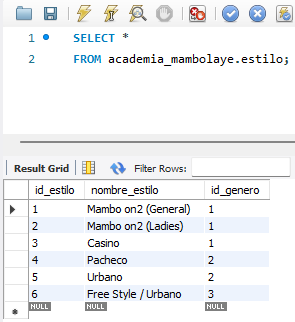
9.1.2.- Clase



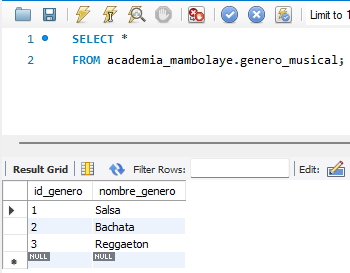
9.1.3.- Director



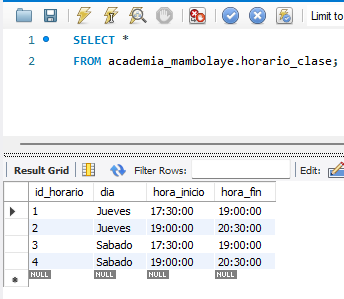
9.1.4.- Estilo



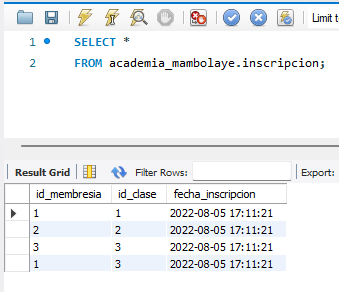
9.1.5.- Género musical



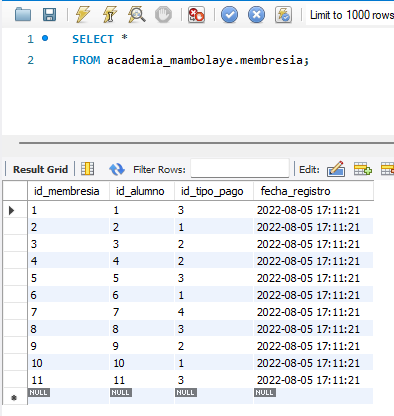
9.1.6.- Horario clase



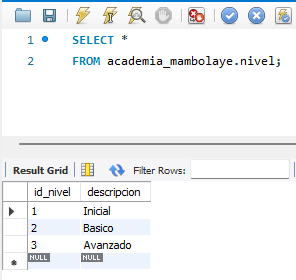
9.1.7.- Inscripción



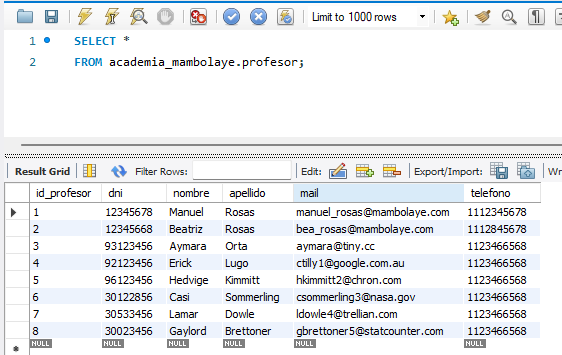
9.1.8.- Membresía



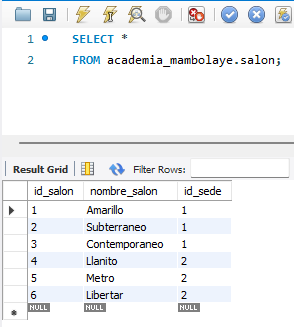
9.1.9.- Nivel



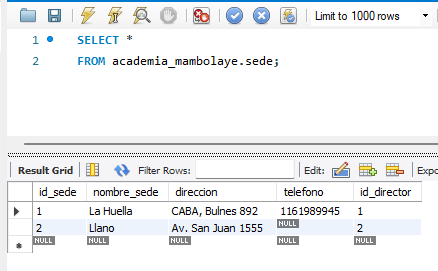
9.1.10.- Profesor



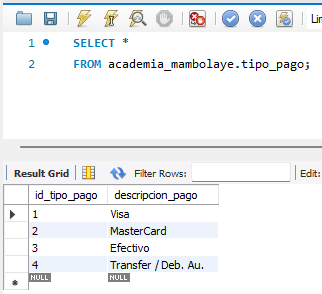
9.1.11.- Salón



9.1.12.- Sedes

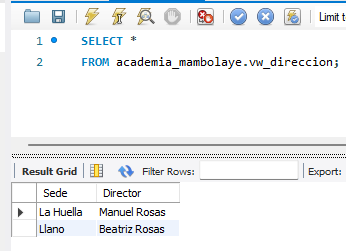


9.1.13.- Tipo pago

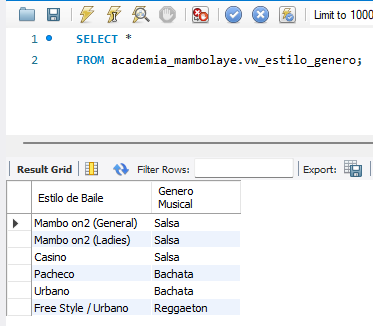


**9.2.- Uso de vistas**

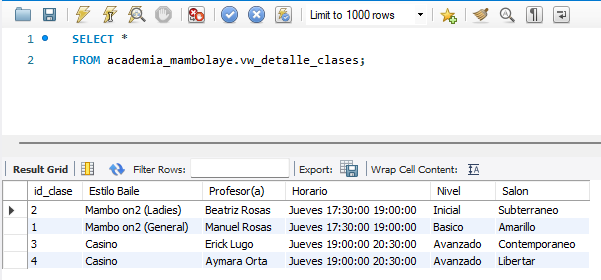
9.2.1.- Dirección



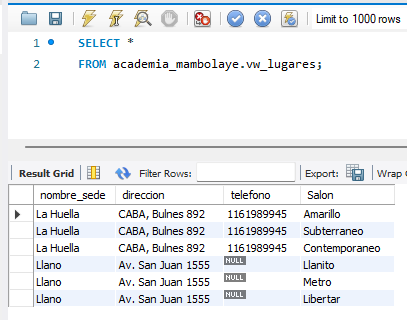
9.2.2.- Estilo Genero



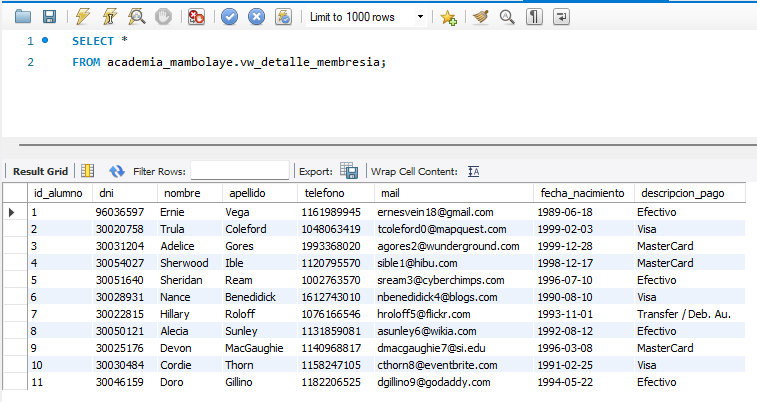
9.2.3.- Detalle Clase



9.2.4.- Lugares

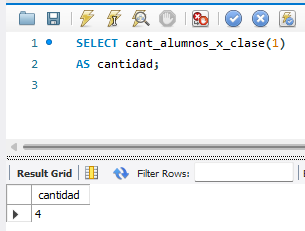


9.2.5.- Detalle Membresía

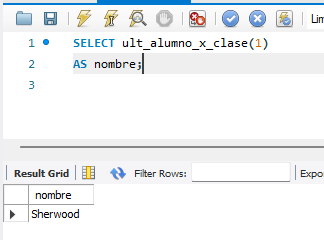


**9.3.- Funciones**

9.3.1.- Cantidad de alumnos por clase



9.3.2.- Ultimo nombre de alumno inscritos en clase especifica



**10.- Herramientas y Tecnologías usadas**

**Draw.io:** Se realizo en primera instancia para la creación y moldeado de las tablas, posibles campos que contendrían cada tabla y bosquejo de Diagrama ER.

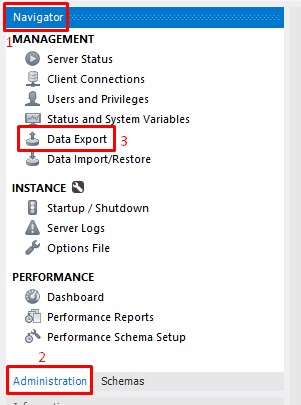
**Microsoft Excel:** Se utilizo para el armado de los detalles correspondientes a cada tabla y así poder asegurar el tipo de datos que puede contener cada campo

**MySQL Workbench:** SGBD usado para la creación de los scripts SQL de todos los objetos de la BD de este proyecto.

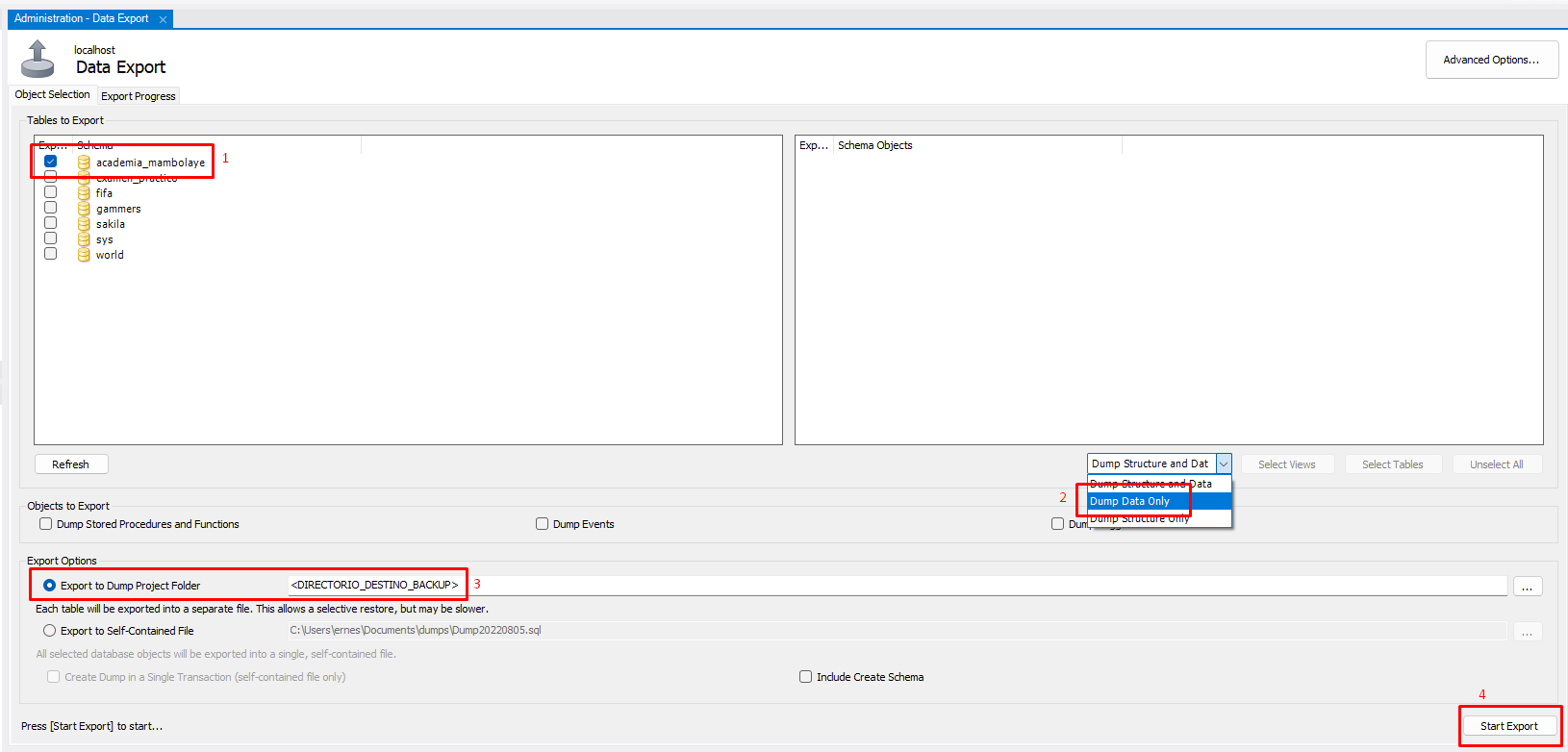
**11.- BackUp:**

Se realiza un paso a paso del procedimiento a seguir para la creación del SQL de respaldo correspondiente a la BD del proyecto:

Paso 1.- Navigator > Administration > Data Export



Paso 2.- Seleccionar Schema > Dump Data Only > Ingresar Directorio Destino > Start Export



Este proceso generará un archivo .sql con el nombre que se le haya indicado el cual contendrá toda la información de la BD seleccionado en el paso 2.

Para este proyecto se generó el archivo [BackupVega.sql](https://github.com/ernieve/SQL_Coder/blob/master/Scripts/BackupVega.sql) que contiene el respaldo de la información del proyecto.

Para la recuperación o importación de la información solo se debe seleccionar la opción **Data Import/Restore** del paso 1 y seleccionar el archivo que contiene la información que se desea recuperar.

**12.- Scripts**

**- DCL Data Control Language (**[**Scripts/SentenciasVega.sql**](https://github.com/ernieve/SQL_Coder/blob/master/Scripts/SentenciasVega.sql)**)**

**- TCL Transaction Control Language (**[**Scripts/TCLVega.sql**](https://github.com/ernieve/SQL_Coder/blob/master/Scripts/TCLVega.sql)**)**