

Universidad Nacional Del Nordeste



Facultad de Ciencias Exactas y Naturales y Agrimensura

Licenciatura en Sistemas de Información

Catedra: Base de Datos 1

Año: 2022

Proyecto Entrega Final: Proyecto Escuela Secundaria

Integrantes:

* Haberles, Lucas Francisco
* Luque Soler, Ignacio Gabriel
* Mulas, Vittorio Augusto
* Pasian Cominotti, Maurizio Giovanni
* Peloso, Nicolás Aníbal

ÍNDICE

[Capítulo 1 3](#_Toc117792246)

[Introducción 3](#_Toc117792247)

[Tema 3](#_Toc117792248)

[Definición o planteamiento del problema 3](#_Toc117792249)

[Objetivos del Trabajo Práctico 3](#_Toc117792250)

[Capítulo 2 4](#_Toc117792251)

[Temas estudiados 4](#_Toc117792252)

[Funciones y procedimientos almacenados 4](#_Toc117792253)

[Disparadores 4](#_Toc117792254)

[Transacciones 5](#_Toc117792255)

[Vistas 5](#_Toc117792256)

[Permisos 6](#_Toc117792257)

[Índices agrupados y no agrupados. Índices filtrados 6](#_Toc117792258)

[BackUp y restauración 7](#_Toc117792259)

[Capítulo 3 9](#_Toc117792260)

[Herramientas utilizadas 9](#_Toc117792261)

[Discord 9](#_Toc117792262)

[WhatsApp 9](#_Toc117792263)

[SQL Server Management Studio 9](#_Toc117792264)

[GitHub 9](#_Toc117792265)

[Capítulo 4 10](#_Toc117792266)

[Descripción de las diferentes entidades y sus relaciones 10](#_Toc117792267)

[Atributos de las Tablas 11](#_Toc117792268)

[Restricciones 13](#_Toc117792269)

[Capítulo 5 14](#_Toc117792270)

[Conclusiones 14](#_Toc117792271)

[Bibliografías 15](#_Toc117792272)

# Capítulo 1

## Introducción

### Tema

Dado que tenemos conocimiento del caso de nuestras antiguas escuelas secundarias de cómo se manejaban respecto a la administración de las mismas, nos proponemos como grupo de trabajo, unidos, trabajar sobre el caso práctico de la administración de una escuela del nivel secundario.

### Definición o planteamiento del problema

En este proyecto tratamos el caso práctico de una escuela secundaria, en el cual incorporamos todos los conocimientos teóricos-prácticos obtenidos durante la cursada y también investigaciones en internet.

El proyecto tiene como finalidad la gestión de la administración de una escuela del nivel secundario, el sistema presentado en la continuidad de este informe debe permitir la gestión de alumnos, tutores, profesores, cursos y materias.

## Objetivos del Trabajo Práctico

El desarrollo del modelado de datos de nuestro proyecto <<*Escuela Secundaria*>> será capaz de; administrar alumnos que puedan tener varios tutores, tutores que puedan tener varios alumnos a cago, alumnos que pertenezcan a un solo curso, este último tiene asociadas varias materias y el alumno tiene tres notas asociadas a cada una de esas materias, éstas son dictadas por uno o varios profesores, los cuales se pueden encargar de varias materias de diferentes cursos.

# Capítulo 2

## Temas estudiados

### Funciones y procedimientos almacenados

#### Funciones

Las funciones en SQL pueden devolver siempre el mismo resultado (deterministas) o pueden retornar resultados distintos (no deterministas) cuando son llamadas con un mismo conjunto de valores de entrada.

A la hora de escribir un script complejo para SQL que realiza una determinada tarea, se puede crear una función para que se encargue de ello. Así, en caso de volver a necesitar nuevamente esa tarea, no habrá que volver a crear el script, siendo suficiente volver a llamar a esa función proporcionándole los parámetros necesarios.

#### Procedimientos almacenados

Un procedimiento almacenado (STORE PROCEDURE) está formado por un conjunto de instrucciones Transact-SQL que definen un determinado proceso a ejecutar, puede aceptar parámetros de entrada y devolver un valor o conjunto de resultados. Este procedimiento se guarda en el servidor y puede ser ejecutado en cualquier momento

Los procedimientos almacenados se crean mediante la sentencia CREATE PROCEDURE y se ejecutan con EXEC (o EXECUTE).

Para ejecutarlo también se puede utilizar el nombre del procedimiento almacenado sólo, siempre que sea la primera palabra del lote. Para eliminar un procedimiento almacenado utilizamos la sentencia DROP PROCEDURE.

#### Diferencias

Por definición, las funciones son ideales para retornar valores con los que deseo operar, mientras que los procedimientos almacenados suelen ser la parte de salida final del código. Es decir que el procedimiento almacenado es el que se comunica directamente con el usuario y le presenta el resultado final.

Por eso, una función no puede invocar un procedimiento almacenado, solamente puede invocar otra función. En cambio, el procedimiento almacenado, como es código final, puede invocar funciones y procedimientos almacenados.

Así que a correcta integración de un sistema que use ambos, sería utilizar procedimientos almacenados para comunicarse y enviar datos a la aplicación; mientras que las funciones se usan para procesar datos internamente dentro del stored procedure.

### Disparadores

Un "trigger" (disparador o desencadenador) es un tipo de procedimiento almacenado que se ejecuta cuando se intenta modificar los datos de una tabla (o vista).

* Se definen para una tabla (o vista) específica.
* Se crean para conservar la integridad referencial y la coherencia entre los datos entre distintas tablas.
* Si se intenta modificar (agregar, actualizar o eliminar) datos de una tabla en la que se definió un disparador para alguna de estas acciones (inserción, actualización y eliminación), el disparador se ejecuta (se dispara) en forma automática.
* Un trigger se asocia a un evento (inserción, actualización o borrado) sobre una tabla.

La diferencia con los procedimientos almacenados del sistema es que los triggers:

* no pueden ser invocados directamente; al intentar modificar los datos de una tabla para la que se ha definido un disparador, el disparador se ejecuta automáticamente.
* No reciben y retornan parámetros.
* son apropiados para mantener la integridad de los datos, no para obtener resultados de consultas.

Los disparadores, a diferencia de las restricciones "Check", pueden hacer referencia a campos de otras tablas.

### Transacciones

Una transacción es un conjunto de operaciones que van a ser tratadas como una única unidad. Estas transacciones deben cumplir 4 propiedades fundamentales comúnmente conocidas como ACID (atomicidad, coherencia, asilamiento y durabilidad).

La transacción más simple en SQL Server es una única sentencia SQL.

Cuando enviamos este tipo de sentencia al SQL Server se escribe en el fichero de transacciones lo que va a ocurrir y a continuación realiza los cambios necesarios en la base de datos. Si hay algún tipo de problema al hacer esta operación el SQL Server puede leer en el fichero de transacciones lo que se estaba haciendo y si es necesario puede devolver la base de datos al estado en el que se encontraba antes de recibir la sentencia.

### Vistas

Una vista es una alternativa para mostrar datos de varias tablas. Una vista es como una tabla virtual que almacena una consulta. Los datos accesibles a través de la vista no están almacenados en la base de datos como un objeto.

Entonces, una vista almacena una consulta como un objeto para utilizarse posteriormente. Las tablas consultadas en una vista se llaman tablas base. En general, se puede dar un nombre a cualquier consulta y almacenarla como una vista.

Una vista suele llamarse también tabla virtual porque los resultados que retorna y la manera de referenciarlas es la misma que para una tabla.

Las vistas permiten:

* Ocultar información: permitiendo el acceso a algunos datos y manteniendo oculto el resto de la información que no se incluye en la vista. El usuario opera con los datos de una vista como si se tratara de una tabla, pudiendo modificar tales datos.
* Simplificar la administración de los permisos de usuario: se pueden dar al usuario permisos para que solamente pueda acceder a los datos a través de vistas, en lugar de concederle permisos para acceder a ciertos campos, así se protegen las tablas base de cambios en su estructura.
* Mejorar el rendimiento: se puede evitar “tipear” instrucciones repetidamente almacenando en una vista el resultado de una consulta compleja que incluya información de varias tablas.

Podemos crear vistas con: un subconjunto de registros y campos de una tabla; una unión de varias tablas; una combinación de varias tablas; un resumen estadístico de una tabla; un subconjunto de otra vista, combinación de vistas y tablas.

La sintaxis básica parcial para crear una vista es la siguiente:

CREATE VIEW NOMBREVISTA AS

--SENTENCIASSELECT

FROM TABLA;

### Permisos

Los permisos de Motor de base de datos se administran en el nivel de servidor mediante inicios de sesión y roles de servidor, y en el nivel de base de datos mediante usuarios de base de datos y roles base de datos. El modelo de SQL Database expone el mismo sistema dentro de cada base de datos, pero los permisos de nivel de servidor no están disponibles.

El número total de permisos para SQL Server 2019 (15.x) es 248. Azure SQL Database expone 254 permisos. La mayoría de los permisos se aplica a todas las plataformas, pero otros no. Por ejemplo, los permisos de nivel de servidor no se pueden conceder en Azure SQL Database y algunos permisos solo tienen sentido en Azure SQL Database. Los nuevos permisos se introducen gradualmente con nuevas versiones.

### Índices agrupados y no agrupados. Índices filtrados

#### Índice Agrupado

es similar a una guía telefónica, los registros con el mismo valor de campo se agrupan juntos. Un índice agrupado determina la secuencia de almacenamiento de los registros en una tabla. Se utilizan para campos por los que se realizan búsquedas con frecuencia o se accede siguiendo un orden.

Una tabla sólo puede tener UN índice agrupado.

El tamaño medio de un índice agrupado es aproximadamente el 5% del tamaño de la tabla.

#### Índice No Agrupado

Es como el índice de un libro, los datos se almacenan en un lugar diferente al del índice, los punteros indican el lugar de almacenamiento de los elementos indizados en la tabla. Un índice no agrupado se emplea cuando se realizan distintos tipos de búsquedas frecuentemente, con campos en los que los datos son únicos.

Una tabla puede tener hasta 249 índices no agrupados.

Si no se especifica un tipo de índice, de modo predeterminado será no agrupado.

Los campos de tipo text, ntext e image no se pueden indizar.

Es recomendable crear los índices agrupados antes que los no agrupados, porque los primeros modifican el orden físico de los registros, ordenándolos secuencialmente

#### Diferencias

La diferencia básica entre índices agrupados y no agrupados es que los registros de un índice agrupado están ordenados y almacenados de forma secuencial en función de su clave.

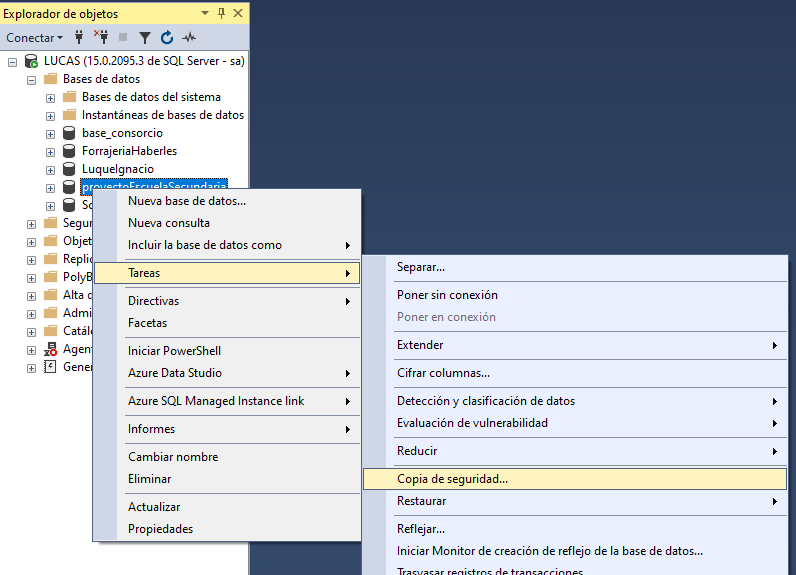
Resumiendo, los índices facilitan la recuperación de datos, permitiendo el acceso directo y acelerando las búsquedas, consultas y otras operaciones que optimizan el rendimiento general.

### BackUp y restauración

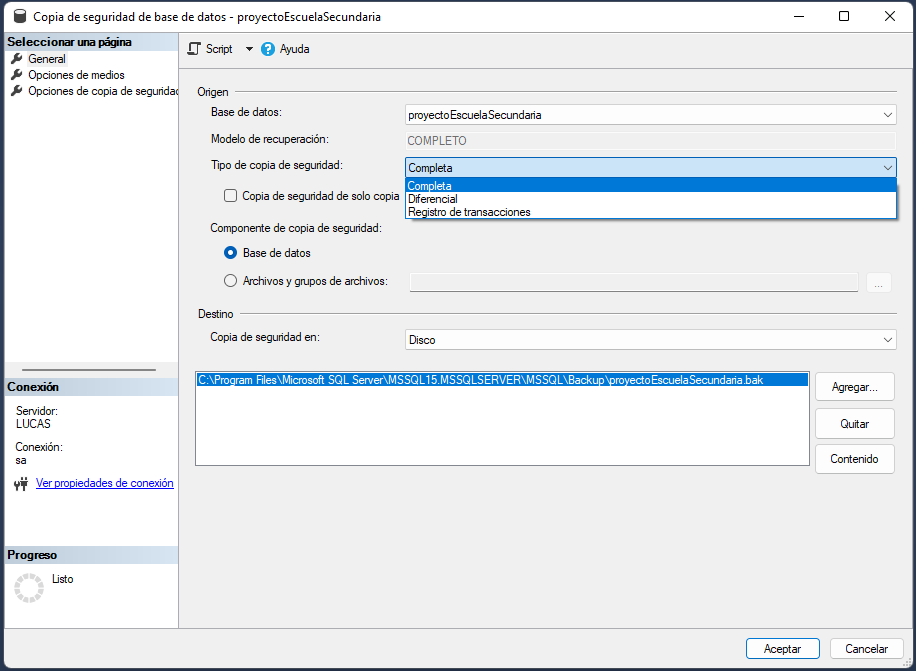
Realizar copias de seguridad de nuestras bases de datos nos permite poder tener respaldo ante posibles anomalías en nuestros proyectos, estás pueden suceder por cause del personal humano, es decir, errores cometidos por personas que administran las bases de datos, o bien por errores en el software y/o hardware del equipo donde se aloja la base de datos. Ante un caso de estos, es esencial tener un respaldo al cual acudir de manera inmediata, por esto, es de mucha importancia el hecho de tener copias de seguridad. A continuación, detallamos como realizar BackUp desde el motor de base de datos MS SQL Server.

También, el uso de las copias de seguridad de archivos puede aumentar la velocidad de recuperación ya que se pueden restaurar solo los archivos dañados sin tener que restaurar el resto de la base de datos.

Por ejemplo, si una base de datos está compuesta por varios archivos ubicados en diferentes discos y se producen errores en uno de ellos, solo debe restaurar el archivo situado en el disco en que se produjeron los errores. El archivo dañado se puede restaurar rápidamente, y la recuperación es más rápida que para una base de datos completa.

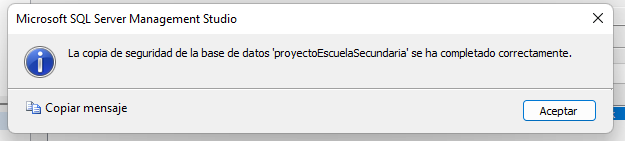
Para realizar un Backup, primeramente, debemos iniciar sesión en el sistema que aloja nuestra base de datos la cual queremos realizar la copia de seguridad. Luego seguir los pasos detallados a continuación

Hacer click derecho sobre la base de datos en cuestión e ir al menú *Tareas > Copia de Seguridad*.



Revisamos los detalles de la copia de seguridad y, a continuación, pulsamos *Aceptar* para crear la copia de seguridad.

Una vez finalizada la copia, obtenemos un mensaje de que la misma se realizó de manera efectiva.



# Capítulo 3

## Herramientas utilizadas

Las herramientas que utilizamos para la realización del proyecto fueron las detalladas a continuación

### Discord

Fue con la que realizamos videoconferencias para la distribución de tareas e informarnos de cómo íbamos yendo en cuanto a las actividades pendientes, para estas actividades realizadas en esta aplicación creamos un nuevo servidor en el cual incorporamos chats de voz y de texto, los de voz para las videoconferencias y los de texto para ir almacenando información que creíamos importante para el desarrollo del proyecto.

### WhatsApp

Fue la aplicación de mensajería para una comunicación más rápida para cuando no nos podíamos reunir todos los integrantes

### SQL Server Management Studio

Esta herramienta decidimos utilizar para realizar el desarrollo de la base de datos, ya que es una herramienta muy completa, conocida y encontramos muy buenas referencias de la misma en Internet.

Algunas de las ventajas que nos interesaron al momento de decidir SQL Server Management Studio como gestor de base de datos utilizar fueron:

* Contiene la facilidad de soporte de transacciones
* Contiene escalabilidad estabilidad y seguridad
* También puede Soportar procedimientos almacenados
* Puede Incluir también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
* También permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información
* Además, permite administrar información de otros servidores de datos.
* También Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
* Contiene también una base de datos relacional por lo cual también se puede incluir en la licencia de base de base datos multidimensional
* Es un producto muy maduro con mucha base implantada y muy buena comunidad.

### GitHub

Esta herramienta colaborativa nos permitió poder trabajar en conjunto. Además, Git nos ofrece varias ventajas frente a otros sistemas tradicionales:

* Sistema distribuido, sin un punto central de fallo, que permite el trabajo incluso sin conexión.
* Superrápido y ligero, optimizado para hacer operaciones de control muy rápidas.
* Crear ramas y mezclarlas es rápido y poco propenso a problemas, al contrario que en otros sistemas tradicionales.
* La integridad de la información está asegurada gracias a su modelo de almacenamiento, que permite predecir este tipo de problemas. En sistemas tradicionales este era un problema grave.
* Permite flujos de trabajo muy flexibles.
* El concepto de área de preparación o staging permite versionar los cambios como nos convenga, no todo o nada.
* Es gratis y de código abierto.
* Otra ventaja importante es que, en la actualidad, la mayoría de empresas al momento de tomar nuevas personas en su equipo de trabajo, solicitan que tengan conocimientos en Git, lo cual nos motivó a implementarlo en nuestro proyecto para ir adquiriendo conocimientos en esta poderosa herramienta colaborativa

# Capítulo 4

A continuación, presentamos el Diagrama Entidad Relación del caso



### Descripción de las diferentes entidades y sus relaciones

Alumno: esta tabla contendrá los datos personales y académicos del alumno, la misma se relaciona con las tablas Alumno\_Nota, Curso, Localidad, Tutor\_Alumno.

Alumno\_Nota: esta tabla pivote hace la conexión de mucho a muchos entre las tablas Alumno y Materia, la misma contendrá las notas del alumno en las diferentes materias

Materia: esta tabla posee código y nombre de la materia, la misma se relaciona con las tablas Alumno\_Nota y Profesor\_Materia

Profesor: esta tabla contiende datos personales del profesor, se relaciona con las tablas Profesor\_Materia y Localidad

Profesor\_Materia: esta tabla pivote realiza la conexión de mucho a muchos entre las tablas Profesor y Materia

Curso: esta tabla contendrá datos del curso, la misma tiene relación con las tablas Curso\_Materia y Alumno

Curso\_Materia: esta tabla pivote hace la conexión de mucho a muchos entre las tablas Curso y Materia

Localidad: esta tabla contiene descripción de una determinada localidad, tiene relación con las tablas Provincia, Profesor, Alumno y Tutor

Provincia: esta tabla contiene descripción de una determinada provincia, tiene relación con la tabla Localidad

Tutor: esta tabla contiene datos personales del tutor, se relaciona con las tablas de Localidad y Tutor\_Alumno

Tutor\_Alumno: esta tabla pivote contiene la relación de mucho a muchos entre las tablas Tutor y Alumno

### Atributos de las Tablas

|  |
| --- |
| Alumno |
| Nombre | Tipo(longitud) | Acepta Valores NULL | Descripción |
| Dni | INT | No | Dni del alumno PK |
| Cod\_provincia | INT | No | Código de provincia FK |
| Cod\_localidad | INT | No | Código de localidad FK |
| Anio | INT | Si | Año del curso FK |
| Division | CHAR (1) | Si | (1 – 2 – 3 etc.) o  (A – B – C etc.) FK |
| Nombre | VARCHAR (50) | No | Nombre/s del alumno |
| Apellido | VARCHAR (50) | No | Apellido/s del alumno |
| FechaNacimiento | DATE | No | Fecha nacimiento del alumno |
| Direccion | VARCHAR (255) | No | Calle y Altura de dirección |
| Sexo | CHAR (1) | No | M (Masculino) – F (Femenino) |

|  |
| --- |
| Alumno\_Nota |
| Nombre | Tipo(longitud) | Acepta Valores NULL | Descripción |
| Dni | INT | No | Dni del alumno PK FK |
| Cod\_materia | INT | No | Código de materia |
| Nota1 | DECIMAL (4, 2) | Si | Valor decimal de la nota |
| Nota2 | DECIMAL (4, 2) | Si | Valor decimal de la nota |
| Nota3 | DECIMAL (4, 2) | Si | Valor decimal de la nota |

|  |
| --- |
| Materia |
| Nombre | Tipo(longitud) | Acepta Valores NULL | Descripción |
| Cod\_materia | INT | No | Código de la materia PK |
| Nombre | VARCHAR (130) | No | Nombre de la materia |

|  |
| --- |
| Profesor |
| Nombre | Tipo(longitud) | Acepta Valores NULL | Descripción |
| Dni | INT | No | Dni del profesor PK |
| Cod\_provincia | INT | No | Código de provincia FK |
| Cod\_localidad | INT | No | Código de localidad FK |
| Nombre | VARCHAR (50) | No | Nombre del profesor |
| Apellido | VARCHAR (50) | No | Apellido del profesor |
| FechaNacimiento | DATE | No | Facha nacimiento del profesor |
| Direccion | VARCHAR (50) | No | Calle y Altura de dirección |
| Sexo | CHAR (1) | No | M (Masculino) – F (Femenino) |

|  |
| --- |
| Profesor\_Materia |
| Nombre | Tipo(longitud) | Acepta Valores NULL | Descripción |
| Dni | INT | No | Dni del profesor PK FK |
| Cod\_materia | INT | No | Código de la materia PK FK |

|  |
| --- |
| Curso |
| Nombre | Tipo(longitud) | Acepta Valores NULL | Descripción |
| Anio | INT | No | Año del curso PK |
| Division | CHAR (1) | No | (1 – 2 – 3 etc.) o  (A – B – C etc.) PK |

|  |
| --- |
| Curso\_Materia |
| Nombre | Tipo(longitud) | Acepta Valores NULL | Descripción |
| Anio | INT | No | Año del curso PK FK |
| Division | CHAR (1) | No | (1 – 2 – 3 etc.) o  (A – B – C etc.) PK FK |
| DniProfesor | INT | No | Dni del profesor PK FK |
| Cod\_materia | INT | No | Código de la materia PK FK |

|  |
| --- |
| Tutor |
| Nombre | Tipo(longitud) | Acepta Valores NULL | Descripción |
| Dni | INT | No | Dni del tutor PK |
| Cod\_provincia | INT | No | Código de provincia FK |
| Cod\_localidad | INT | No | Código de localidad FK |
| Nombre | VARCHAR (50) | No | Nombre/s del tutor |
| Apellido | VARCHAR (50) | No | Apellido/s del tutor |
| FechaNacimiento | DATE | No | Fecha nacimiento del tutor |
| Direccion | VARCHAR (255) | No | Calle y Altura de dirección |
| Sexo | CHAR (1) | No | M (Masculino) – F (Femenino) |

|  |
| --- |
| Tutor\_Alumno |
| Nombre | Tipo(longitud) | Acepta Valores NULL | Descripción |
| Dni\_tutor | INT | No | Dni del tutor PK FK |
| Dni\_alumno | INT | No | Dni del alumno PK FK |

|  |
| --- |
| Provincia |
| Nombre | Tipo(longitud) | Acepta Valores NULL | Descripción |
| Cod\_provincia | INT | No | Código de provincia PK |
| Nombre | VARCHAR (50) | No | Nombre de la provincia |

|  |
| --- |
| Localidad |
| Nombre | Tipo(longitud) | Acepta Valores NULL | Descripción |
| Cod\_provincia | INT | No | Código de provincia PK FK |
| Cod\_localidad | INT | No | Código de localidad PK |
| Nombre | VARCHAR (50) | No | Nombre de la localidad |

### Restricciones

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre Entidad | Tipo de Restricción | Nombre de Restricción | Columna |
| Alumno | PRIMARY KEY | PK\_Alumno | Dni |
| Alumno | FOREIGN KEY | FK\_alumno\_curso | Anio, Division |
| Alumno | FOREIGN KEY | FK\_alumno\_localidad\_provincia | Cod\_provincia, Cod\_localidad |
| Alumno | CHECK | CK\_Sexo\_Alumno | Sexo |
| Alumno\_Nota | PRIMARY KEY  FOREIGN KEY | PK\_alumno\_nota | Dni, Cod\_materia |
| Alumno\_Nota | FOREIGN KEY | FK\_alumno\_alumno | Dni |
| Alumno\_Nota | FOREIGN KEY | FK\_alumno\_materia | Cod\_materia |
| Curso | PRIMARY KEY | PK\_curso | Anio, Division |
| Curso\_Materia | PRIMARY KEY | PK\_curso\_materia | Anio, Division, DniProfesor, Cod\_materia |
| Curso\_Materia | FOREIGN KEY | FK\_curso\_materia\_curso | Anio, Division |
| Curso\_Materia | FOREIGN KEY | FP\_curso\_materia\_profesor\_materia | DniProfesor,  Cod\_materia |
| Localidad | PRIMARY KEY | PK\_localidad | Cod\_provincia,  Cod\_localidad |
| Localidad | FOREIGN KEY | FK\_cod\_provincia | Cod\_provincia |
| Materia | PRIMARY KEY | PK\_Materia | Cod\_materia |
| Profesor | PRIMARY KEY | PK\_Profesor | Dni |
| Profesor | FOREIGN KEY | FK\_profesor\_localidad\_provincia | Cod\_provincia,  Cod\_localidad |
| Profesor | CHECK | CK\_Sexo\_Profesor | Sexo |
| Profesor\_Materia | PRIMARY KEY | PK\_profesor\_materia | Dni, Cod\_materia |
| Profesor\_Materia | FOREIGN KEY | FK\_profesor\_materia\_cod\_materia | Cod\_materia |
| Profesor\_Materia | FOREIGN KEY | FK\_profesor\_materia\_dni | Dni |
| Provincia | PRIMARY KEY | PK\_Provincia | Cod\_provincia |
| Tutor | PRIMARY KEY | PK\_Tutor | Dni, Cod\_provincia,  Cod\_localidad |
| Tutor | FOREIGN KEY | FK\_tutor\_localidad\_provincia | Cod\_provincia,  Cod\_localidad |
| Tutor | CHECK | CK\_Sexo\_Tutor | Sexo |
| Tutor\_Alumno | PRIMARY KEY | PK\_tutor\_alumno | Dni\_tutor, Dni\_alumno |
| Tutor\_Alumno | FOREIGN KEY | FK\_tutor\_alumno\_alumno | Dni\_alumno |
| Tutor\_Alumno | FOREIGN KEY | FK\_tutor\_alumno\_tutor | Dni\_tutor |

# Capítulo 5

## Conclusiones

Se logró elaborar y plantear el objetivo propuesto de la creación de un modelado de datos para nuestro proyecto sobre la gestión de la administración de una escuela de nivel secundario.

En él se abordaron los temas presentados en el capítulo 2 del presente informe como son funciones, procedimientos almacenados, disparadores, transacciones, vistas, entre otros. Para el desarrollo, hemos utilizado todos los conocimientos adquiridos durante el cursado y además adquirimos nuevos conocimientos mediante la practica e implementación de los temas investigados en internet. Estos temas fueron desarrollados con éxito por la buena distribución de tareas durante la realización del proyecto y la guía del docente.

# Bibliografías

[https://learn.microsoft.com/es-es/sql/t-sql/](https://learn.microsoft.com/es-es/sql/t-sql/statements/backup-transact-sql?view=sql-server-ver16)

<https://www.ibm.com/docs/es/license-metric-tool?topic=database-restoring-ms-sql-server>

<https://www.ibm.com/docs/es/license-metric-tool?topic=database-backing-up-ms-sql-server>

<https://www.tutorialesprogramacionya.com/sqlserverya/index.php?inicio=1>

<https://codigosql.top/sql-server/>