

Curso de SQL

(Junio a Septiembre de 2024)



Alumno: De Cicco, Martín Daniel

Docente: Blanc, Lucía Soledad

Tutor: Solorzano, Melina

- **Introducción**

Las bases de datos son utilizadas hoy en día, en todos los ámbitos, para modelar entidades y negocios de todo tipo. Tener una DB relacional bien organizada y funcional es fundamental para hacer consultas y obtener resultados rápidos y confiables. Además, es una forma segura de conservar la integridad de los datos a través de los vínculos entre las tablas, y de los permisos otorgados a los distintos usuarios.

- **Objetivo**

La presente base de datos pretende brindar a una universidad una manera óptima de organizar los registros de todas sus entidades. Tener un padrón actualizado de los estudiantes, de cómo es su distribución en las distintas carreras, así como de los profesores que dictan cada asignatura, permitirá elaborar informes y sacar conclusiones.

- **Situación problemática**

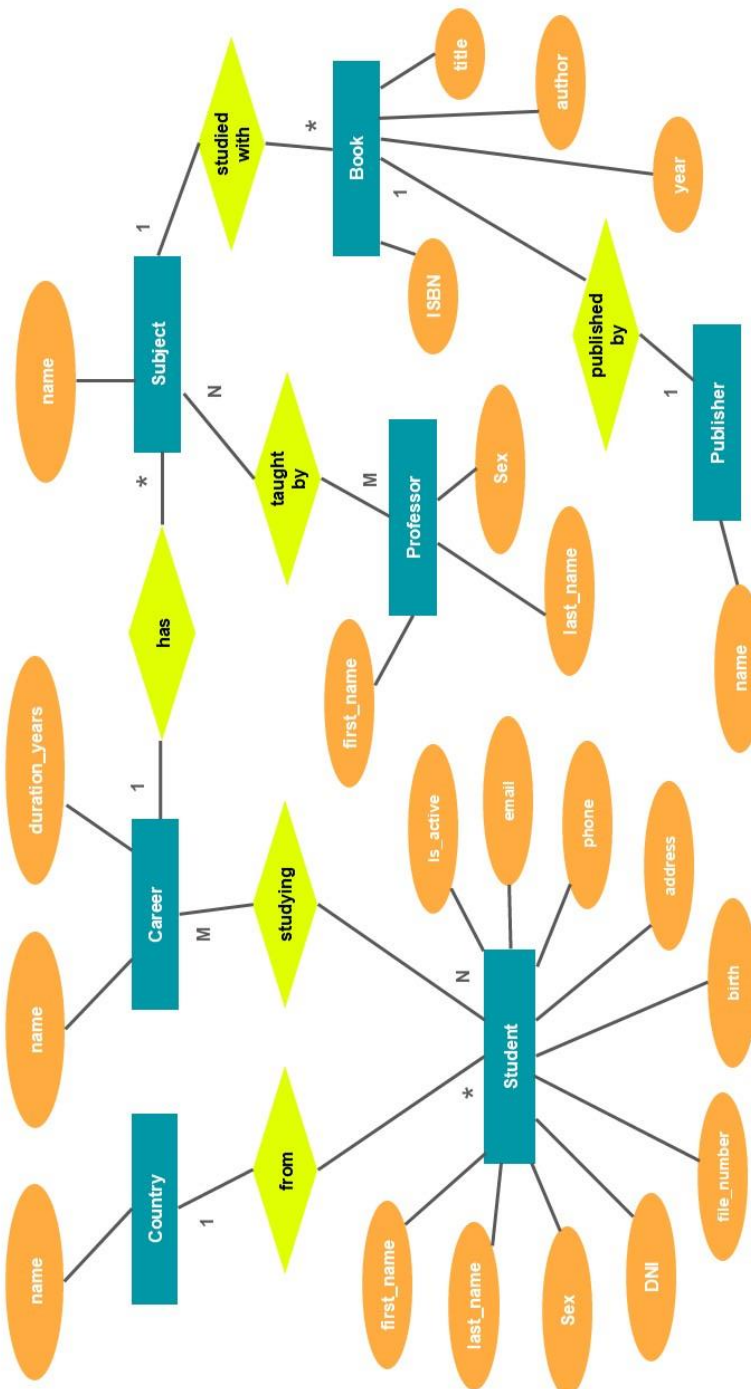
Se busca representar los registros con los que cuenta una universidad promedio. La información sobre alumnos, docentes, carreras, etc., permitirá obtener resultados y obrar de manera acorde a éstos. En el estado actual de la DB se asume que ningún alumno ha terminado de estudiar, es decir que no hay egresados. Solamente se diferenciará entre estudiantes activos (regulares) e inactivos (pasivos).

- **Modelo de negocio**

En este caso tenemos una universidad en la que se dictan muchas carreras. Cada una de éstas tiene, a su vez, una currícula con varias asignaturas. Dichas materias son dictadas por uno o más profesores, y pueden ser estudiadas a través de la bibliografía disponible en la biblioteca de la universidad. Los alumnos pueden ser de distintas nacionalidades, y pueden estudiar más de una carrera a la vez.

- **Diagrama Entidad – Relación de la DB university**

En el siguiente esquema los rectángulos simbolizan las entidades, por ejemplo, los alumnos, las asignaturas, etc. Las elipses representan los atributos o campos de cada entidad, mientras que los rombos indican las relaciones entre ellas. Estas relaciones pueden ser uno a uno (1 - 1), uno a muchos (1 - *) o muchos a muchos (N - M).



- **Listado de Tablas**

Tabla	professor						
Descripción	Tiene la información de los profesores que dictan las asignaturas.						
Key	COLUMN	TYPE	LENGTH	NOT NULL	DEFAULT	ON DELETE	NOTES
PK	id_prof	INT		TRUE			ID del profesor
	first_name	VARCHAR	50				Nombres
	last_name	VARCHAR	50				Apellido
	sex	CHAR	1				Sexo

Tabla	career						
Descripción	Tiene el listado de las carreras disponibles.						
Key	COLUMN	TYPE	LENGTH	NOT NULL	DEFAULT	ON DELETE	NOTES
PK	id_career	INT		TRUE			ID de la carrera
	name	VARCHAR	50				Nombre de la carrera
	duration_years	INT					Años de duración

Tabla	subject						
Descripción	Tiene el listado de las asignaturas.						
Key	COLUMN	TYPE	LENGTH	NOT NULL	DEFAULT	ON DELETE	NOTES
PK	id_subject	INT		TRUE			ID de la asignatura
	name	VARCHAR	50				Nombre de la asignatura
FK	prof_id	INT		TRUE			ID del profesor
FK	career_id	INT		TRUE			ID de la carrera

Tabla	publisher						
Descripción	Tiene el listado de las editoriales de los libros.						
Key	COLUMN	TYPE	LENGTH	NOT NULL	DEFAULT	ON DELETE	NOTES
PK	id_pub	INT		TRUE			ID de la editorial
	name	VARCHAR	50				Nombre de la editorial

Tabla	book						
Descripción	Tiene el listado de los libros de la biblioteca que se utilizan en las asignaturas.						
Key	COLUMN	TYPE	LENGTH	NOT NULL	DEFAULT	ON DELETE	NOTES
PK	id_book	INT		TRUE			ID del libro
	title	VARCHAR	80				Título del libro
	author	VARCHAR	80				Autor
FK	publisher_id	INT		TRUE			ID de la editorial
	year	INT		TRUE			Año de publicación
	ISBN	VARCHAR	13				ISBN del libro
FK	subject_id	INT		TRUE			ID de la asignatura

Tabla	country						
Descripción	Tiene el listado de los países de los que son oriundos los estudiantes.						
Key	COLUMN	TYPE	LENGTH	NOT NULL	DEFAULT	ON DELETE	NOTES
PK	id_country	INT		TRUE			ID del país
	name	VARCHAR	50				Nombre del país

Tabla	student						
Descripción	Tiene el listado de los estudiantes de la Universidad.						
Key	COLUMN	TYPE	LENGTH	NOT NULL	DEFAULT	ON DELETE	NOTES
PK	id_student	INT		TRUE			ID del estudiante
	first_name	VARCHAR	50				Nombres
	last_name	VARCHAR	50				Apellido
	sex	CHAR	1				Sexo
	DNI	VARCHAR	8				DNI
FK	country_id	INT		TRUE			ID del país
	file_numer	INT					Número de legajo
	birth	DATE					Fecha de nacimiento
	address	VARCHAR	30				Domicilio
	phone	VARCHAR	15				Teléfono
	email	VARCHAR	50				Correo electrónico
	is_active	INT			1		Alumno regular o pasivo.

Tabla	professor_subject						
Descripción	Tabla intermedia para la relación N-M entre profesores y asignaturas.						
Key	COLUMN	TYPE	LENGTH	NOT NULL	DEFAULT	ON DELETE	NOTES
FK, PK(1/2)	professor_id	INT		TRUE			ID del profesor
FK, PK(2/2)	subject_id	INT		TRUE			ID de la asignatura

Tabla	student_career						
Descripción	Tabla intermedia para la relación N-M entre alumnos y carreras.						
Key	COLUMN	TYPE	LENGTH	NOT NULL	DEFAULT	ON DELETE	NOTES
FK, PK(1/2)	student_id	INT		TRUE			ID del profesor
FK, PK(2/2)	career_id	INT		TRUE		CASCADE	ID de la asignatura

• Listado de Vistas

➤ Vista 1: vw_students_argentina

Descripción: Muestra toda la información de los alumnos de Argentina.

Objetivo: Ver cuántos alumnos hay de Argentina, con todos sus datos.

Tablas/Datos: student

➤ Vista 2: vw_students_career

Descripción: Lista los alumnos junto a sus carreras.

Objetivo: Mostrar qué carrera(s) está cursando cada estudiante.

Tablas/Datos: student, student_career, career

➤ **Vista 3: vw_students_per_career**

Descripción: Muestra la cantidad de alumnos en cada carrera (activos).

Objetivo: Saber cuántos alumnos activos/regulares hay por carrera.

Tablas/Datos: student, student_career, career

➤ **Vista 4: vw_total_students_per_career**

Descripción: Muestra la cantidad de alumnos en cada carrera (totales).

Objetivo: Ver cuántos alumnos totales (activos + pasivos) hay por carrera.

Tablas/Datos: student, student_career, career

➤ **Vista 5: vw_elder_students**

Descripción: Lista los alumnos activos mayores de 25 años.

Objetivo: Saber qué alumnos siguen estudiando después de los 25 años de edad.

Tablas/Datos: student

➤ **Vista 6: vw_recent_books**

Descripción: Muestra los libros publicados de 2010 en adelante.

Objetivo: Mostrar cuántos libros publicados recientemente tiene la biblioteca de la universidad.

Tablas/Datos: book

➤ **Vista 7: vw_professors_subjects**

Descripción: Lista los profesores junto a las asignaturas que dictan.

Objetivo: Mostrar qué materias enseña cada profesor.

Tablas/Datos: professor, professor_subject, subject

- **Listado de Funciones**

- **Función 1: fn_total_students**

Descripción: Muestra la cantidad de alumnos (totales).

Objetivo: Saber el número total de estudiantes en la universidad.

Tablas/Datos: student. Sin parámetros.

- **Función 2: fn_women_percentage**

Descripción: Da el porcentaje de mujeres inscriptas en la universidad.

Objetivo: Mostrar el porcentaje de estudiantes mujeres sobre el total de alumnos.

Tablas/Datos: student. Sin parámetros.

- **Función 3: fn_active_percentage**

Descripción: Muestra el porcentaje de alumnos activos en toda la universidad.

Objetivo: Ver qué porcentaje de estudiantes se encuentran en condición de regular/activo.

Tablas/Datos: student. Sin parámetros.

- **Función 4: fn_name_career (id_num INT)**

Descripción: Retorna el nombre de la carrera elegida.

Objetivo: Conocer el nombre de una carrera a partir de su número de identificación.

Tablas/Datos: career. Recibe 1 parámetro INT correspondiente al ID de la carrera.

- **Función 5: fn_career_most_active**

Descripción: Devuelve el nombre de la carrera con más alumnos activos.

Objetivo: Saber cuál es la carrera con mayor número de alumnos regulares/activos.

Tablas/Datos: student, student_career, career. Sin parámetros.

- **Listado de Procedimientos**

- **Procedimiento 1: sp_career_students_list (IN id INT, OUT message VARCHAR(40))**

Descripción: Muestra los nombres de los alumnos de una carrera dada (totales).

Objetivo: Conocer, a partir del identificador de una carrera, todos los alumnos (activos + pasivos) de la misma.

Tablas/Datos: student, student_career, career

- **Procedimiento 2: sp_career_active_list (IN id INT, OUT message VARCHAR(40))**

Descripción: Lista los nombres de los alumnos de una carrera dada (activos).

Objetivo: Saber los nombres de los estudiantes activos en una carrera, a partir del número de identificación de la misma.

Tablas/Datos: student, student_career, career

- **Procedimiento 3: sp_career_total_students (IN id INT, OUT message VARCHAR(40))**

Descripción: Obtiene la cantidad de alumnos de la carrera elegida (totales).

Objetivo: Ver cuántos alumnos totales hay en determinada carrera.

Tablas/Datos: student, student_career, career

- **Procedimiento 4: sp_career_subjects (IN id INT)**

Descripción: Da el listado de asignaturas de la carrera, a partir de su identificador.

Objetivo: Mostrar el plan de estudio de una carrera dada.

Tablas/Datos: subject, career

- **Procedimiento 5: sp_career_books (IN id INT)**

Descripción: Muestra la bibliografía utilizada en la carrera elegida.

Objetivo: Indicar todos los libros disponibles para la carrera indicada.

Tablas/Datos: book, subject, career

➤ **Procedimiento 6: sp_career_professors (IN id INT)**

Descripción: Lista los profesores que enseñan en una determinada carrera.

Objetivo: Conocer los nombres de los docentes de la carrera deseada.

Tablas/Datos: professor, professor_subject, subject, career

➤ **Procedimiento 7: sp_student_careers (IN file_num INT)**

Descripción: Indica las carreras que cursa un alumno dado.

Objetivo: Mostrar qué carreras cursa un alumno, a partir de su matrícula o número de legajo.

Tablas/Datos: student, student_career, career

➤ **Procedimiento 8: sp_check_student (IN first_name VARCHAR(50), IN last_name VARCHAR(50), IN sex CHAR(5), IN DNI VARCHAR(8), IN country_id INT, IN file_num INT, IN birth DATE, IN address VARCHAR(30), IN phone VARCHAR(15), IN email VARCHAR(50), IN is_active INT)**

Descripción: Permite ingresar alumnos luego de validar campos.

Objetivo: Validar los datos que se quieren ingresar en cada campo antes de ingresar al alumno.

Tablas/Datos: student

➤ **Procedimiento 9: sp_add_student (IN first_name VARCHAR(50), IN last_name VARCHAR(50), IN sex VARCHAR(50), IN DNI VARCHAR(50), IN country_id VARCHAR(50), IN file_num VARCHAR(50), IN birth VARCHAR(50), IN address VARCHAR(50), IN phone VARCHAR(50), IN email VARCHAR(50), IN is_active VARCHAR(50))**

Descripción: Captura errores al intentar insertar alumnos. Si no los hay llama al Procedimiento 9.

Objetivo: Manejar errores relacionados con los tipos de datos, al querer ingresar un estudiante.

Tablas/Datos: NINGUNA

➤ **Procedimiento 10: sp_add_professor (IN surname VARCHAR(50), IN name VARCHAR(50), IN sex CHAR(4))**

Descripción: PERMITE INGRESAR PROFESORES

Objetivo: Incorporar nuevos docentes al personal universitario.

Tablas/Datos: professor

- **Listado de Triggers**

- **Trigger 1: tr_audit_students**

Descripción: Guarda los ingresos de los alumnos al sistema, en una tabla de auditoría.

Objetivo: Hacer un registro de cada estudiante que se ingresa en la DB, en caso de querer consultar todos los que ha habido, y no sólo los que permanecen en la tabla de alumnos.

Tablas/Datos: student, new_students (auditoría)

- **Trigger 2: tr_audit_deleted_students**

Descripción: Conserva los alumnos eliminados del sistema, en una tabla de auditoría.

Objetivo: Llevar un historial de los alumnos eliminados, en caso de que estos registros llegaran a ser necesarios.

Tablas/Datos: student, old_students (auditoría)

- **Listado de Usuarios**

- **Usuario 1: alumno@localhost**

Descripción: Representa a un alumno cualquiera que estudia en la universidad.

Objetivo: Que puedan ver solamente la información general de la universidad, y lo concerniente a ellos mismos.

Tablas/Datos: SELECT a todas las tablas, menos a student y student_career. De la tabla professor sólo puede ver id_prof, first_name, last_name. No puede ejecutar funciones ni procedimientos.

- **Usuario 2: profesor@localhost**

Descripción: Destinado para el uso de cualquier profesor de la universidad.

Objetivo: Que puedan ver solamente la información general de la universidad, y algunos datos de los alumnos.

Tablas/Datos: SELECT a todas las tablas. De la tabla professor sólo puede ver id_prof, first_name, last_name; de la tabla student sólo puede seleccionar id_student, first_name, last_name, file_number, email, is_active. No puede ejecutar funciones ni procedimientos.

➤ **Usuario 3: secretaria@localhost**

Descripción: Representa a una persona de secretaría, que puede consultar toda la información para hacer informes, pero no puede modificar los datos.

Objetivo: Que la secretaria pueda consultar toda la información de la DB sin alterarla.

Tablas/Datos: SELECT a todas las tablas. Puede utilizar todas las funciones. Se le permite ejecutar todos los procedimientos excepto los que ingresan alumnos y profesores.

➤ **Usuario 4: division_alumnos@localhost**

Descripción: Destinado a cualquier empleado de la oficina División Alumnos, quien puede hacer libre uso de las tablas.

Objetivo: Que la persona pueda insertar, modificar y eliminar registros, y que pueda utilizar todos los procedimientos y funciones.

Tablas/Datos: Todas las operaciones sobre tablas. Funciones y procedimientos, sólo puede ejecutarlos.

- **Scripts**

➤ **Script de creación de objetos**

https://github.com/mddecicco85/Curso_SQL/blob/main/Proyecto_Final/Script_Create_objects.sql

➤ **Script de inserción de datos**

https://github.com/mddecicco85/Curso_SQL/blob/main/Proyecto_Final/Script_Insert.sql

➤ **Script de creación de usuarios**

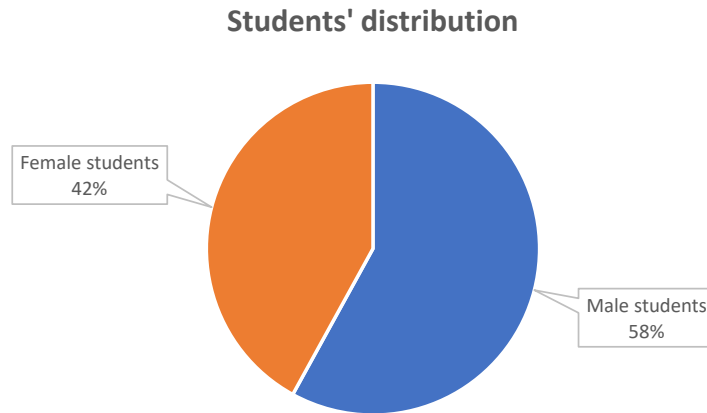
https://github.com/mddecicco85/Curso_SQL/blob/main/Proyecto_Final/Script_Create_Users.sql

➤ **Script unificado**

https://github.com/mddecicco85/Curso_SQL/blob/main/Proyecto_Final/Script_UNIFICADO.sql

- **Informes generados a partir de la información de la DB**

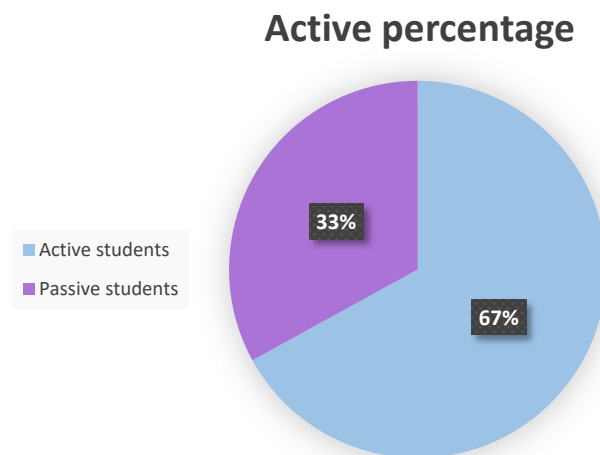
En este caso, si hacemos uso de la Función 2, `fn_women_percentage`, podemos calcular el porcentaje de alumnas en la universidad, o, lo que es lo mismo, cómo es la distribución de alumnos según el sexo.



	career	total_students
►	Astronomía	4
	Ciencias de la atmósfera	4
	Ciencias políticas	2
	Comunicación social	2
	Derecho	4
	Diseño Gráfico	1
	Diseño Industrial	4
	Farmacia	1

Aquí vemos otro ejemplo donde, utilizando la Vista 4, `vw_total_students_per_career`, podemos obtener el listado de todas las carreras que se dictan, junto al número de alumnos totales de cada una. Esto puede ser útil para saber cuál es la carrera más demandada, y evaluar si es necesario incorporar nuevos profesores.

Por último, otro de los resultados que se pueden obtener -mediante la Función 3, `fn_active_percentage`- es la proporción de estudiantes que se encuentran en condición de alumno regular (activo). Si el porcentaje de estudiantes pasivos (inactivos) supera un determinado valor, sería importante estudiar por qué ocurre este fenómeno, y qué consecuencias podría tener.



- **Herramientas y tecnologías usadas**

- MySQL Workbench
- Microsoft Power Point
- Copilot
- Microsoft Word
- Microsoft Excel
- Google
- Git Hub

- **Futuras líneas**

- Incluir un procedimiento para insertar libros en la biblioteca de la universidad.
- Establecer un Estado al alumno, como Recibido o Estudiando, y este último separarlo en Activo/Regular y Pasivo.
- Agregar las calificaciones del alumno en las asignaturas, para poder calcular promedios y hacer otras operaciones o funciones con números.
- Incorporar más información sobre los profesores, como el DNI, en caso de que haya nombres repetidos. También las fechas de nacimiento para saber quiénes tienen más de 70 años.
- Cambiar la relación entre Asignaturas y Libros, de 1 a muchos, a muchos a muchos, ya que un mismo libro puede utilizarse en más de una asignatura.
- Modificar la relación entre Carreras y Asignaturas, de 1 a muchos, a muchos a muchos, porque una misma asignatura puede estar en varias carreras.