

Justificaciones

A la hora de crear las entidades, buscamos que cada una almacene información sobre un solo concepto, eliminando la redundancia y protegiendo la integridad de los datos mediante claves foráneas.

A continuación, se justifica la creación de cada entidad principal dentro del esquema DROPDATABASE, agrupándolas por su rol en el modelo.

Entidades Básicas

Estas tablas representan las "bases" fundamentales del negocio. Almacenan entidades que existen de forma independiente y son referenciadas por múltiples procesos, es decir, no tienen ninguna FK

- Profesor: El enunciado requiere gestionar los datos del profesor. En lugar de repetir esta información en cada curso que dicta, se decidió crear una entidad maestra. Esto asegura que si un profesor cambia su teléfono, solo se deba actualizar un registro en esta tabla. El DNI (prof_dni) se eligió como clave de negocio UNIQUE para el proceso de MERGE
- Alumno: Contiene los datos de los alumnos. De forma similar al profesor, alum_dni y alum_legajo se definen como UNIQUE, ya que la Maestra presentaba inconsistencias y ambos campos eran necesarios para identificar a un mismo alumno.
- Sede: El enunciado indica que los cursos se dictan por sede. Entonces pensábamos que almacenar los nombres de las sedes como texto en la tabla Curso sería propenso a errores de tipeo e inconsistencias. Entonces decidimos crear Sede como una tabla de catálogo para garantizar la integridad referencial, usando un ID numérico para vincularla a Curso.
- Pregunta: Se decidió crear una tabla Pregunta para almacenar el texto de cada pregunta. Esto nos permite que la institución añada o elimine preguntas en el futuro, brindándonos mayor escalabilidad y evitando la necesidad de alterar la estructura de la base de datos.

Primeras Entidades con relaciones

Una vez que ya cargamos las primeras 4 entidades bases, podemos empezar a cargar otras entidades más complejas y empezar a crear las distintas relaciones entre nuestras entidades

- Curso: Esta tabla es la encargada de guardar toda la información acerca de los cursos, desde su nombre y descripción hasta el turno y el precio por mes, por lo que podría ser considerada la "tabla principal" ya que posee la mayor cantidad de relaciones en todo nuestro modelo. Las primeras relaciones que

podemos crear son con la sede y con el profesor, y como ambas relaciones son de Muchos a Uno, se le agregó una FK dentro de curso para la sede y para el profesor

- Factura: Otra tabla que podemos crear con las 4 primeras es a la Factura, la cual posee una fecha de emisión y vencimiento, un total, y además se asocia al alumno que la creó a través de una relación Muchos a Uno y una FK en la factura

Segunda Generación de Entidades con relaciones con la Tabla Curso

Como ya dijimos, la tabla curso es nuestra “tabla principal” por lo que decidimos crear todas las distintas tablas que necesitan al curso para existir

- Encuesta: En la consigna se habla de un sistema de encuestas el cual es anónimo, por lo que no se relaciona con el alumno, y en el cual, dichas encuestas son realizadas en base a un curso en específico, por lo que ahí, hay otra relación de Muchos a Uno. Por ahora la tabla encuesta solo tiene como atributos propios la fecha de la encuesta, las observaciones de la misma, y obviamente el id del curso
- Curso Dias: Esta tabla fue creada a partir de la posibilidad de que un curso fuera impartido en varios días de la semana, por lo que se optó usar una tabla aparte la cual no tiene un id propio, sino que su PK esta formada por el id del curso y el día de la semana, esto para evitar que un curso tenga como días 2 Lunes o 2 Martes, de esta forma evitas dicha repetición sin sentido
- Inscripción Curso: Esta tabla tiene la función de registrar el historial y el estado de la solicitud de cada alumno en cada curso. Para esto se tiene un estado (PENDIENTE/CONFIRMADA/RECHAZADA) y una fecha de creación y de respuesta. Además, al ser la inscripción de UN alumno a UN curso, esta tabla posee el id tanto del curso como del alumno
- Cursada: Esta tabla existe con el único propósito de romper la relación Muchos a Muchos que existe entre el Curso y el Alumno, esto pasa porque un Alumno puede estar inscripto a muchos cursos y un Curso posee muchos alumnos
- Módulo: A su vez, se habla también de cómo está dividido un Curso, y es a través de Módulos, y como otra vez hay una Relación de Muchos a Uno, esta tabla además de tener una descripción, posee el id del curso al que pertenece
- Final: Y para terminar con esta sección, pero no con las relaciones, está la tabla Final, la cual, al poseer un único curso al que este final está asignado, posee una FK que apunta a la tabla curso a través del curso id

Segunda Generación de Entidades con relaciones con la Tabla Factura

Una vez que terminamos de crear las muchas tablas que se relacionan con Curso, pasamos a la otra tabla que creamos anteriormente y es la Factura

- Factura Detalle: Como con una factura se pueden pagar multiples cursos a los que pertenece un alumno, se necesita alguna forma de guardar dichos

cursos con sus distintos precios para cada uno. Por lo que nosotros optamos crear una tabla detalle la cual como PK tiene al nro de factura y el id del curso, esto se hizo con el mismo motivo de Curso Días. Además de la PK, esta tabla tiene una fecha de referencia y además del total para ese curso en específico

- Pago: Llegando la última tabla de esta llamada segunda generación está Pago, y se utiliza para registrar los distintos pagos que los distintos alumnos realizaron, para eso esta tabla guarda la fecha del pago, el importe, el medio de pago y la factura la cual se está pagando (Obviamente se referencia a través de una FK hacia la tabla Factura)

Última Generación de Entidades

Ahora si, llegamos a las últimas 5 tablas las cuales necesitaron que se crearan todas las tablas anteriores para poder existir

- Inscripción Final: Esta tabla funciona igual que la de Inscripción Curso, solo que en este caso, se saltea el estado de inscripción y la fecha de respuesta
- Final Alumno: Esta tabla no solo sirve para romper la relación Muchos a Muchos entre el Final y el Alumno, sino que además se utiliza para guardar la nota que tuvo dicho alumno en el final, si estuvo presente y además, qué profesor lo corrige (Como el final solo puede corregido por un único profesor, se utiliza una FK que hace referencia a la tabla Profesor)
- Tp: Esta tabla existe para guardar la información acerca de los distintos tp que un alumno rinde en cada cursada, por lo que este se relaciona con la tabla Cursada, teniendo como PK, la misma que Cursada, y esto se hace con el mismo motivo que ella. Además, tiene la fecha del tp junto con su nota
- Evaluación: Esta tabla es igual que el tp, sólo que en este caso, se busca guardar las evaluaciones del alumno en cada módulo, no en la cursada, por lo que en este caso, la PK está conformada por el id del alumno y por el id del módulo el cual se está evaluando
- Pregunta Encuesta: Como última tabla que se crea es la de pregunta encuesta, esta se utiliza para modelar el Muchos a Muchos que hay entre la Pregunta y la Encuesta, además de tener un atributo propio el cual es la respuesta que el alumno le puso a esa pregunta

Una vez hecha la creación de las distintas tablas, para la parte del emigrado de datos optamos por un enfoque similar al anterior, en el que creamos un procedure por cada tabla a migrar y luego estos se ejecutan en el mismo orden en el que se crearon las tablas para evitar que se rompan las distintas relaciones que existen entre las tablas