



Informe técnico de base de datos activa - Gestión de notas

Gestión de base de datos

Integrantes:

- Manuel González
- Matías Parra

1. Introducción

En éste proyecto se tuvo que implementar una base de datos activa que fuese consumida por una página web y que solucione el problema de gestión de notas de alumnos en la universidad. Ésta base de datos está basada en el sistema de gestión de notas de la Universidad de Talca. Para implementar nuestra solución se tuvo que crear varias tablas relacionadas entre sí, crear scripts de procedimientos almacenados, funciones, triggers y vistas, finalizando con la implementación de una página web para consumir la base de datos, con ésta página solo implementando las llamadas y conexiones a la base de datos y dejando toda o casi toda la lógica en los scripts de la base de datos.

2. Explicación de la base de datos

La base de datos está compuesta de 10 tablas, siendo éstas las siguientes:

- **alumno:** es la tabla que almacena la entidad alumno en el sistema y es el alumno el que interactúa con varias otras tablas, ya que se almacenan las notas de éste alumno, se inscribe a cursos, etc.
- **alumno_seguridad:** es la tabla encargada de guardar las contraseñas de cada alumno en la base de datos.
- **profesor:** almacena los datos de cada profesor en la universidad, dejando su rut como PK.
- **profesor_seguridad:** se basa en la tabla alumno_seguridad, pero ésta almacenará las contraseñas de cada profesor.
- **curso:** se encarga de almacenar el curso “padre” del sistema, esto quiere decir que es el curso general sin sus secciones, alumnos y semestres.
- **instancia_curso:** éste se relaciona con la tabla curso y tiene los detalles de un curso, es decir, sus secciones, el periodo, semestre y año en que se realiza y el porcentaje que manejará la creación de las evaluaciones en las tablas siguientes.
- **matricula:** es la relación que tiene el alumno con la instancia curso y sus notas finales de cada curso. También tiene la situación del alumno que puede ser aprobado, reprobado o cursando.
- **evaluacion:** muy parecida a la tabla curso porque ésta tabla es la general de una evaluación, es decir, tiene el tipo de evaluación, el porcentaje que le

pertenece, si es exigible o no y está relacionada con el profesor que crea la evaluación y la instancia del curso.

- **instancia_evaluacion:** contiene la nota de la evaluación por cada alumno. Ésta tabla se llena a través de un cursor y un trigger después de crear una evaluación.
- **log:** ésta tabla no tiene referencias a ninguna tabla, pero contiene todas las acciones (INSERT, UPDATE, DELETE) que se le hacen a todas las tablas y almacena la hora en que se realizó, qué usuario lo hizo, qué hizo y los datos originales y nuevos dependiendo de la acción que se realizó.

3. Funcionalidades de la base de datos

Las funcionalidades de la base de datos se divide en varios tipos de scripts (de insert, cursores, etc) y a continuación se explicarán:

- **Procedimientos de insert:** son 8 procedimientos que insertan datos nuevos en cada tabla, así la página web llama a estos procedimientos y le pasa los datos. Varios de estos procedimientos tienen IFs de verificación de los datos para que estos estén dentro del rango definido, otros activan triggers después de ser usados, etc.
- **Procedimientos de update:** son 10 procedimientos que se encargan de actualizar algunos datos de las tablas. No modifica los datos importantes como las referencias a otras tablas o las PK.
- **Procedimientos de delete:** solo son 3 procedimientos, dos de los cuales no borra pero deshabilita alumno y profesor. El procedimiento que si borra es para la instancia curso, siempre y cuando no tenga estudiantes.
- **Funciones de select:** hay 19 funciones que utilizan select y devuelve un valor entero (para validaciones o cantidades) o tablas completas que entregan datos de varias tablas (join) o entregan éstos datos ordenados o representados de otra manera (el promedio más texto, etc), hay funciones para todas las tablas que muestran sus valores a través de su PK, etc.
- **Cursores:** los cursores que tenemos son 5, dos de los cuales son triggers functions (funciones que retornan un trigger), éstos dos cursores se encargan de agregar las evaluaciones automáticamente por cada alumno inscrito en el curso al cual se le crea la evaluación “padre”, y el segundo es relacionado a lo mismo pero cuando se hace un update. Los otros cursores se encargan de verificar la situación del alumno y el porcentaje de la instancia_curso.

- **Procedimiento calcular_nota_final:** como el nombre lo dice, este procedimiento calcula la nota final de un alumno en particular inscrito en un curso respectivo. Este procedimiento es el más largo y contiene las llamadas de varios otros procedimientos y/o funciones porque para poder calcular el promedio final de un alumno debe verificar si la instancia_evaluación existe, si el porcentaje de todas las evaluaciones de esa instancia_curso sumen 100% y se encarga de poner la situación del alumno al final de calcular el promedio final, verificando si tiene nota roja en una evaluación exigible o no y poner el promedio que se conoce en el sistema de la universidad de Talca.
- **Triggers:** hay varios triggers y se encargan de actualizar el promedio cuando se actualiza una nota, registrar en la tabla log cuando se hace un insert, update o delete en cualquier tabla, y como se mencionó anteriormente, ejecutar cursores después de un insert o update en una evaluación.
- **Vista materializada de alumno:** la vista muestra todos los cursos inscritos por un alumno para poder mostrar a dicho alumno sólo los datos de esos cursos. Esta vista cuenta con dos funciones, una función trigger y un trigger para hacer que funcione. Una función es para crear la vista materializada (se ejecuta solo una vez en la vida de la base de datos), la otra función es para llamar la vista y mostrar su contenido al usuario, la función trigger y el trigger son para actualizar los datos de la vista materializada de manera automáticamente una vez que se haga un insert, update o delete en la tabla matrícula, esto hará que la vista siempre tenga los datos actualizados.
- **Vista materializada de profesor:** esta vista muestra los cursos que el profesor imparte y utiliza también dos funciones, una función trigger y un trigger. Las primeras dos funciones son parecidas a las de la vista alumno que son para crear y consultar la vista, la función trigger y el trigger actualizan la vista materializada con los datos nuevos después de un insert, delete o update en la tabla instancia_curso.
- **Reportes:** se cuenta con 11 reportes que verifican los siguientes requisitos:
 1. **Promedio de una sección:** muestra el promedio de todo el curso.
 2. **Porcentaje de alumnos aprobados y reprobados en una sección:** muestra cada sección o instancia de un curso con el porcentaje de alumnos aprobados versus los alumnos reprobados.
 3. **Porcentaje de alumnos matriculados:** muestra el porcentaje de los alumnos que están matriculados comparados con todos los alumnos del sistema.
 4. **Lista de alumnos con mejor a peor promedio:** muestra una lista de todos los alumnos ordenados de manera descendente sobre su

promedio final de la carrera, destacando los primeros que son los con mejor promedios.

5. **Lista de alumnos con peor a mejor promedio:** muestra una lista de todos los alumnos ordenados de manera ascendente sobre su promedio final de la carrera, destacando los primeros que son los con peor promedios.
6. **Alumnos con mayor a menor número de cursos reprobados:** muestra una lista descendente de los alumnos que han reprobado más cursos en su vida universitaria.
7. **Alumnos con mayor a menor número de cursos aprobados:** muestra una lista descendente de los alumnos que han aprobado más cursos en su vida universitaria.
8. **Número de cursos dictados por cada profesor:** muestra una lista de los profesores y la cantidad de cursos que a dictado cada uno.
9. **Profesores y la cantidad de alumnos reprobados en sus cursos:** muestra una lista de los profesores junto al número de alumnos reprobados por el total de los cursos dictados por este.
10. **Profesores y la cantidad de alumnos aprobados en sus cursos:** muestra una lista de los profesores junto al número de alumnos aprobados por el total de los cursos dictados por este.
11. **Lista de cursos con su porcentaje de reprobados:** muestra una lista de todos los cursos junto con el porcentaje de alumnos reprobados en esta.

4. Creación de la página web

Para almacenar la base de datos activa se usó PostgreSQL y para consumirla se creó una página web con Angular y Express.js. En esta página web se consideran tres usuarios: el administrador, los profesores y los alumnos. El administrador es el encargado de ver y modificar todo de la base de datos, éste usuario puede agregar y actualizar alumnos, profesores, cursos, instancias, evaluaciones, etc. Los usuarios de tipo profesor pueden crear evaluaciones, modificar sus datos y modificar las notas de los alumnos. Finalmente, los usuarios de tipo alumno solo pueden ver sus cursos, evaluaciones y notas, lo único que pueden modificar son sus propios datos (teléfono).

5. Conclusión

Para solucionar el problema de un gestor de notas creamos una base de datos activa con pocas clases pero bastantes procedimientos almacenados y funciones para poder administrar todos los requisitos del sistema. Los procedimientos almacenados son de varios tipos comenzando por los más básicos de inserción, actualización y algunos de borrado o deshabilitado. Las funciones más importantes en el sistema son el cálculo de nota final y el manejo de las inscripciones entre el alumno y sus notas, y para esto se tuvo que realizar varias validaciones sobre los porcentajes de la nota, la situación de aprobado o reprobado del alumno, etc. Finalmente, la creación de la página web resultó y permitió un fácil y más cómodo manejo de los datos en la base de datos, ya que al tener varias funciones y procedimientos puede ser bastante abrumador.