

SQL

SEGUNDA ENTREGA DEL PROYECTO FINAL

Estudiante: Ramirez Sanabria, Federico

Profesor: Redondo, Camilo

Tutor: Goldfeld, Jennifer

INTRODUCCIÓN

Este proyecto apunta a la creación de una base de datos destinada a optimizar la gestión de una escuela secundaria argentina. El fin es llevar a cabo un registro y control eficiente de los estudiantes, docentes y otra información relevante asociada a estos actores. Con esta herramienta, se busca agilizar los procesos administrativos escolares y facilitar el acceso a la información para la mejora de las tareas que la escuela lleva a cabo en colaboración con toda la comunidad educativa.

OBJETIVOS

Entre los objetivos de esta base de datos se pueden listar los siguientes puntos importantes:

- 1) Almacenar en forma segura la información personal de los estudiantes de la institución.
- 2) Registrar información académica sobre los estudiantes en lo concerniente a calificaciones, asistencia y rendimiento en las asignaturas curriculares pertenecientes a cada curso.
- 3) Llevar un registro sobre el historial académico de los estudiantes para tener una noción de su trayecto académico en términos de fechas de inicio y finalización del ciclo secundario.
- 4) Registrar información sobre los docentes de la institución, asignación de cursos y materias.
- 5) Llevar un registro de todos los cursos de la institución, sus divisiones y sus respectivas asignaturas.
- 6) Gestionar los datos de los pagos de matrículas en relación a la inscripción de las cursadas e informar sobre el estado de pago de la matrícula por cada estudiante de la institución.
- 7) Gestionar los datos de los pagos de las cuotas mensuales, llevar un registro de las cuotas becas y los montos adeudados para cada estudiante de la institución educativa.

SITUACIÓN

La implementación de esta base de datos es fundamental para agilizar los procesos administrativos de una escuela. En el mundo actual, la tecnología interfiere cada vez más en nuestras vidas, las comunicaciones entre la institución y las familias ya no se desarrollan tanto en forma personal, si no que también se aprovechan los espacios virtuales de comunicación para mantener un vínculo fluido con las familias. Es por ello que, es importante el desarrollo de esta base de datos para tener a mano toda la información esencial correspondiente a un estudiante u otro personal escolar. Por ejemplo, es mucho más ágil y eficaz enviar las calificaciones y datos de asistencia en formato digital a las familias que tener que realizar boletines a mano. Lo mismo para la comunicación con el personal docente, enviarles datos sobre cursos y materias asignadas en formato digital gracias a los registros le agilizaría bastante su trabajo a la hora de considerar calificaciones, preparar el cronograma para un curso, datos de antigüedad a tener en cuenta al momento de abonar sueldos y múltiples situaciones más. En síntesis, esta base de datos permitirá una comunicación y un vínculo fluido entre la institución y toda la comunidad educativa optimizando así el trabajo escolar y llevando registros de manera constante sobre el desarrollo académico fructífero o no de cada estudiante.

MODELO DE NEGOCIO

La base de datos escolar está enfocada en proporcionar un acceso fácil y ágil a la información. La base no solamente cumplirá con el rol de almacenar información, sino que se utilizará esa decisión para la toma de decisiones. Por ejemplo, poder acercarse a los estudiantes cuyos registros demuestren bajo desempeño, poder entablar comunicaciones con las familias ante sucesivas inasistencias de estudiantes, tener datos de participación de estudiantes en eventos académicos para decidir si continuar fomentándolos o cambiar las temáticas de dichos eventos, planificar el cronograma para cada ciclo lectivo, etc. La base de datos apunta a optimizar el trabajo escolar en la institución en todos sus aspectos, ofreciendo un servicio de calidad a la institución, mejorando la gestión de sus recursos y fomentando el trabajo ágil y de calidad.

LISTADO DE TABLAS

Tabla	Descripción	Campos
Students	Registra a todos los estudiantes de la institución	id_student id_course first_name last_name birth_date address email
Courses	Registra todos los cursos y divisiones de estos mismos	id_course course_name course_division
Teachers	Registra a todos los docentes que trabajan en la institución, así como también las materias que dictan	id_teacher id_subject first_name last_name email profesional_degree admission_date
Subject	Contiene los datos de las materias correspondientes al ciclo secundario argentino	id_subject subject_name
Marks	Contiene las calificaciones obtenidas de cada estudiante de la escuela	id_mark id_student id_subject mark mark_date
Attendance	Registra las asistencias e inasistencias de los estudiantes	id_attendance id_student id_subject

		absence_number total_attendance
Academic_record	Registra el historial académico de los estudiantes, para gestionar las fechas de ingreso y egreso del ciclo secundario	id_record id_student start_date finish_date school_diploma
Academic_events	Registra los eventos académicos en los que participan estudiantes de la institución, ya sea internos o en colaboración con otras escuelas	id_event id_student event_type event_description event_date
Enrolments	Contiene los datos de matriculación a los cursos de los estudiantes, de manera de gestionar estados de pago e inscripciones	id_enrolment id_student id_course enrolment_date enrolment_status
Monthly_fees	Almacena los datos de pago de las cuotas mensuales y deudas de los estudiantes	id_fee id_student monthly_fee scholarship debt

INSERCIÓN DE DATOS

Esta base de datos consta de las 10 tablas explicadas en el apartado anterior. A las primeras cinco tablas se les realizó la inserción de datos mediante la sintaxis SQL pertinente y a las cinco restantes se les realizó la inserción de datos mediante importación CSV. Este procedimiento de importación consta de los siguientes pasos:

1. Realizo una consulta a los datos de la tabla, que claramente estará vacía.
2. Presionó el botón para realizar la importación.
3. Navegó por mi equipo hasta encontrar el archivo CSV con los datos de la tabla.

4. Procedo a elegir la opción de utilizar una tabla existente para importar los datos.
5. Verifico que las columnas de la tabla de la base de datos sean coincidentes con las de la tabla del archivo CSV.
6. Realizo la importación y finalizo el proceso.

VISTAS

El proyecto cuenta con las siguientes vistas:

- **VW_YOUNGEST_STUDENTS:** Esta vista trabaja con las tablas Courses, Students y Academic_record. Su objetivo es mostrar el listado de estudiantes que terminarán su educación secundaria posteriormente al año 2025. Se utiliza el JOIN para asociar las tablas mencionadas en función de las claves primarias y foráneas.
- **VW_EVENT_TYPE:** Esta vista trabaja con las tablas Students y Academic_events. Su objetivo es mostrar la cantidad de estudiantes agrupados (o inscriptos) en eventos académicos, pero solamente para aquellos eventos cuya categoría este repetida más de una vez. Se utiliza el JOIN para vincular las tablas mencionadas y se da uso también a otros comandos como HAVING y GROUP BY.
- **VW_TEACHERS_LIMIT:** Esta vista solo trabaja con la tabla Teachers. Su objetivo es mostrar los primeros 8 registros de esta tabla utilizando el comando LIMIT.
- **VW_MARKS_ORDER:** Esta vista solo trabaja con la tabla Marks. Su objetivo es mostrar las calificaciones de los estudiantes de mayor a menor, de manera tal de diferenciar bien los aprobados de los desaprobados.
- **VW_SUBJECTS_ATTENDANCE:** Esta vista trabaja con las tablas Students, Subjects y Attendance. Su objetivo es mostrar el listado de estudiantes que tengan un número de ausencias superior a 10 en una asignatura. Se destaca por el uso del JOIN para relacionar las tablas mediante sus respectivas llaves primarias y foráneas, y a su vez, se da uso al comando ORDER BY para ordenar los estudiantes en forma ascendente. Es decir, se mostrará el listado de menor a mayor tomando como base el número de ausencias.

FUNCIONES

Las funciones desarrolladas en esta base de datos son las siguientes:

- **getStudentLastName:** Esta función recibe por parámetro el ID del estudiante y devuelve el apellido del estudiante asociado a dicho ID. Sin proporcionar el ID la función no podrá cumplir su objetivo.
- **calculateFeeWithScholarship:** Esta función recibe dos parámetros, el valor de la cuota mensual y el porcentaje de beca en formato decimal. Su objetivo es calcular el costo final de las cuotas mensuales becadas. Si no se proporcionan esos datos como parámetros la función no logrará alcanzar su objetivo.

STORED PROCEDURES

Los stored procedures desarrollados en esta base de datos son los siguientes:

- **sp_OrderTable:** este stored procedure realiza un ordenamiento personalizado de una tabla a elección. Mediante los parámetros adecuados se podrá ordenar una tabla específica a partir de un campo y orden personalizados. Los parámetros son tres: tableName (nombre de la tabla a ordenar), tableField (nombre del campo que se usará como condición) y orderType (tipo de orden a ejecutarse, el cual podrá ser ascendente y/o descendente).
- **Sp_InsertStudent:** este stored procedure realiza una inserción de un nuevo estudiante a la tabla Students. Se deberá respetar cada parámetro para garantizar una correcta inserción de los datos. Dentro de los parámetros podemos encontrar los siguientes: p_id_course (ID del curso), p_first_name (nombre del estudiante), p_last_name (apellido del estudiante), p_birth_date (fecha de nacimiento del estudiante), p_address (domicilio del estudiante) y p_email (correo electrónico del estudiante).

TRIGGERS

Esta base de datos cuenta con los siguientes triggers:

- **TRG_LOG_TEACHERS:** Este trigger trabaja en colaboración con la tabla LOG_TEACHER_AUDIT, la cual realiza una auditoría en la tabla TEACHERS. Este trigger se ejecuta posteriormente a una operación INSERT (por eso lleva consigo el comando AFTER) y guarda en la tabla de auditoría los siguientes datos: el nuevo ID del profesor agregado, el nombre y el apellido del profesor, el tipo de operación realizada (INSERT), la tabla que recibe la inserción (TEACHERS), el usuario que realizó la operación y la fecha en la que fue hecha. Por cada nueva inserción a la tabla TEACHERS se podrá consultar la tabla de auditoría para visualizar el trabajo del trigger.
- **TR_LOG_TEACHERS_2:** Este trigger trabaja en colaboración con la tabla LOG_TEACHER_AUDIT_2, la cual realiza una auditoría en la tabla TEACHERS. Este trigger se ejecuta previamente a una operación UPDATE (motivo por el cual lleva consigo el comando BEFORE) y guarda en esta segunda tabla de auditoría los siguientes datos: el campo nuevo de apellido actualizado en contraste con el campo viejo, el tipo de operación realizada (UPDATE), la tabla afectada por la actualización (TEACHERS), el usuario que ejecutó la operación y la fecha en la que fue hecha. Por cada nuevo UPDATE se podrá consultar la tabla de auditoría correspondiente a este trigger para verificar los cambios realizados.