

Proyecto Final

Base de Datos

SQL

Matias Christello

Comisión N°53.170

Contenido

DESCRIPCIÓN:	3
OBJETIVOS:	3
POSIBLES MEJORAS:	3
DIAGRAMA ENTIDAD - RELACIÓN:	4
DETALLE DE TABLAS:	5
VISTAS	8
FUNCIONES	9
STORED PROCEDURES	10
TRIGGERS	11
DCL	12

DESCRIPCIÓN:

Se realiza una base de datos para el ámbito de “seguros”. La idea es tener acceso a todos los clientes, sus respectivas pólizas, coberturas, los vehículos, siniestros y los pagos realizados.

OBJETIVOS:

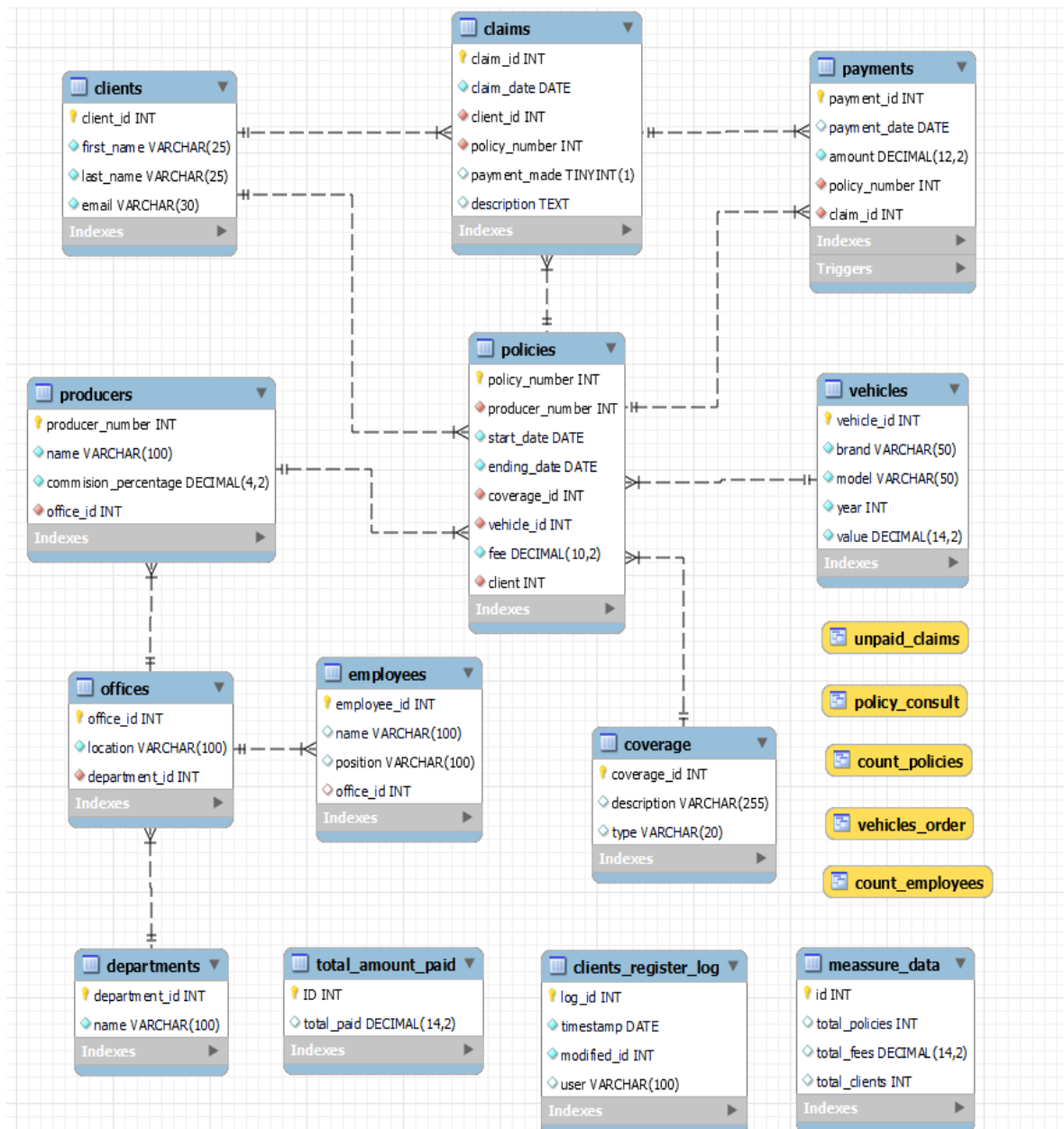
Teniendo en cuenta que es un rubro donde el volumen de operaciones puede llegar a ser muy grande, y también incluye muchas variables (vehículos, coberturas, siniestros, pagos, etc), se busca una manera de poder tener los datos ordenados para poder acceder a ellos de distintas maneras, y lo más sencillo posible.

El principal objetivo es tener ordenado y poder contabilizar los clientes, pudiendo, cada productor, filtrarlos por su código personal.

POSIBLES MEJORAS:

Se evalúa la posibilidad de mejorar las tablas para poder incluir distintas compañías de seguros. Así, un productor, podrá tener al alcance, en un solo lugar, todos los datos de sus asegurados, sin importar en que compañía tengan su póliza, así como también la información de las oficinas de dichas compañías, reclamos hechos en cada una, y estado del siniestro, con pago o no.

DIAGRAMA ENTIDAD - RELACIÓN:



DETALLE DE TABLAS:

Clients:

Tabla para almacenar los datos de los asegurados.

- `client_id` - identificador del cliente, valor numérico
- `first_name` - Almacena caracteres, hasta 25
- `last_name` - Almacena caracteres, hasta 25
- `email` - Almacena caracteres, hasta 30

Coverage:

Tabla donde se almacenan las distintas coberturas, con su nombre y descripción.

- `coverage_id` - identificador único de la cobertura, valor numérico.
- `Description` - Almacena cualquier dato de texto.
- `type` - Almacena caracteres, hasta 20.

Policies:

Tabla donde se almacenan las pólizas. Se relaciona con la tabla de Clientes, a través del `client_id`, y con la tabla de coberturas, a través del id de la cobertura.

- `policy_number` - identificador único de la póliza valor numérico.
- `start_date` - Almacena datos en modo fecha.
- `ending_date` - Almacena los datos en modo fecha.
- `Fee` - Almacena número hasta 10 caracteres, con 2 decimales
- `client` - valor numérico, hace referencia a la tabla Clientes
- `coverage_id` - valor numérico, hace referencia a la tabla Coverage
- `producer_number` - valor numérico, hace referencia a la tabla producers

Vehicles:

Tabla donde se puede consultar todos los vehículos.

- `vehicle_id` - identificador único del vehículo, valor numérico.
- `brand` - Almacena caracteres, hasta 50.
- `model` - Almacena caracteres, hasta 50.
- `year` - Almacena en valor numérico.
- `value` - Almacena número hasta 14 caracteres, con 2 decimales.

Claims:

Tabla de almacenamiento de todos los reclamos. Se relaciona con la tabla de pólizas a través del número de dicha póliza.

- `claim_id` - identificador del reclamo, valor numérico.
- `claim_date` - Almacena los datos en modo fecha.
- `policy_number` - Almacena datos en valor numérico.
- `description` - Almacena cualquier dato de texto.
- `policy_number` - valor numérico, hace referencia a la tabla Policies

Payment:

Tabla donde se registran los pagos hechos luego de los reclamos. Se relaciona, también, con la tabla de polizas, a través del número de éstas.

- `payment_id` - identificador del pago, valor numérico.
- `payment_date` - Almacena en modo fecha
- `amount` - Almacena número hasta 12 caracteres, con 2 decimales
- `policy_number` - Almacena valor numérico
- `policy_number` - valor numérico, hace referencia a la tabla Policies

Departments:

Tabla para almacenar los distintos departamentos de la compañía

- `department_id` - identificador del departamento, valor numérico.
- `name` - Almacena hasta 100 caracteres

Employees:

Tabla para almacenar los datos de los empleados. Se relaciona con la tabla Departments a través del ID.

- `employees_id` - identificador único del empleado, valor numérico.
- `name` - Almacena hasta 100 caracteres
- `position` - Almacena hasta 100 caracteres
- `office_id` - valor numérico, hace referencia a la tabla offices

Offices:

Tabla para almacenar los datos de las oficinas. Se relaciona con la tabla Departments a través del ID, y con la tabla employes

- office_id - identificador de la oficina, valor numérico.
- location - Almacena hasta 100 caracteres
- department_id - valor numérico que hacer referencia a departments

Producers:

Tabla para almacenar los datos de los empleados. Se relaciona con la tabla Departments a través del ID, y con la tabla pólices.

- Producer_number - valor numérico, identifica a cada productor
- name - Almacena hasta 100 caracteres
- office_id - valor numérico, hace referencia a la tabla offices

Clients_register_log

Tabla de uso interno, donde se registra qué usuario realizado cambios en la tabla Clients

- log_id - identificador único del log, valor numérico
- timestamp - Clave tipo fecha
- modified_id - Clave numérica, hace referencia al ID del cliente modificado
- user - Almacena hasta 100 caracteres, hace referencia al usuario que realizó la modificación

Total_ammount_paid

Tabla de uso interno, donde se va sumando el valor de los pagos realizados a clientes

- ID - valor numérico
- total_paid - valor numérico que permite hasta 14 número, incluidos dos decimales.

Meassure_data

Tabla de uso interno que almacena valores cuando hay cambios en las tablas CLIENTS o POLICIES

- id - valor numérico
- total_policies - valor numérico
- total_feed - valor numérico que permite hasta 14 número, incluidos dos decimales.
- Total_clients - valor numérico

VISTAS

COUNT_POLICIES

- Se usan las tablas POLICIES y PRODUCERS. La funcionalidad de esta vista es mostrar la cantidad de pólizas total que tiene cada productor.

COUNT_EMPLOYES

- Utilización de las tablas OFFICES y EMPLOYES. Tiene la función de mostrar la cantidad de empleados que hay, agrupados por oficina.

VEHICLES_ORDER

- Solo utiliza la tabla de VEHICLES, y los ordena los mismos, de forma descendente, por su valor

UNPAID_CLAIMS

- Utiliza las tablas de CLAIMS y PAYMENTS. Solo muestra los reclamos que NO han sido pagados, y su respectivo importe.

POLICY_CONSULT

- Utiliza las tablas de POLICIES, CLIENTS, VEHICLES y COVERAGE. Filtra los resultados mostrando sola las pólizas que tiene cobertura 'D3', y la vista trae el nombre del cliente, marca y molde del vehículo, número de póliza y tipo de cobertura.

FUNCIONES

fn_producer_info.

- Se utiliza para leer los datos del productor requerido. Para su uso es necesario proporcionar el número de productor.

fn_get_location

- Su utilidad radica en mostrar la ubicación de una oficina específica. Se debe proporcionar el ID de la oficina.

STORED PROCEDURES

sp_NewClient

- Se utiliza para la creación de un nuevo CLIENTE en la base de datos.
- Para el uso correcto es necesario brindar los datos de nombre del cliente, apellido y el email.

sp_UpdateClaim

- Se usa para actualizar la tabla CLAIMS, a partir de un pago realizado en la tabla PAYMENTS.
- Para el uso es hace falta proporcionar el ID del reclamo pagado, y un valor TRUE o FALSE, según se quiera actualizar.

TRIGGERS

TRG_clients_update_log

- Se activa al hacer un UPDATE en algún cliente en la Base de Datos. Para cada línea actualizada, crea un registro en la tabla “clients_register_log”, donde ingresa la fecha y hora de la actualización, el id del cliente actualizado, y el user que realizó la actualización.

TRG_insert_new_payment

- Se activa al hacer un INSERT en la tabla PAYMENTS. Cuando se registra un nuevo pago, este trigger se encarga de tomar el valor del nuevo pago registrado, y sumarlo al valor actual que figure en la tabla “total_amount_paid”, lo cual mantiene un registro del importe total pagado en siniestros.

TRG_meassure_data_clients

- Se activa después de hacer un INSERT en la tabla CLIENTS. Su función es actualizar la tabla MEASURE_DATA, en la columna “total_clients” y suma una unidad al valor actual de dicha columna.

TRG_meassure_data_policies

- Se activa después de hacer un INSERT en la tabla POLICIES. Al igual que el trigger anterior, actualiza la tabla MEASURE_DATA. Actualiza dos columnas, suma en una unidad al valor de la columna “total_policies”, y suma el valor insertado en la columna “fee” al valor actual de la columna “total_fees”

DATA CONTROL LANGUAGE

`coder@localhost:`

Se crea el user solo con permisos de lectura sobre la base de datos. Se identifica con el password 'coderhouse'

`mchristello@localhost:`

Se crea el user con permisos de lectura, inserción y actualización de datos sobre todas las tablas de la base de datos. Se identifica con el password 'christello'