

Proyecto final: Ascensores Aquino

Alumno: Santiago Aquino

Comisión: 34960

Índice:

1. Introducción
2. Objetivo
3. Situación problemática
4. Diagrama Entidad-Relación
5. Tablas (Script Nº 1)
6. Views (Script Nº 3)
7. Functions (Script Nº 4)
8. Stored procedures (Script Nº 5)
9. Triggers (Script Nº 6)
10. Sentencias (Script Nº 7)
11. Backup (Script Nº 9)
12. Herramientas utilizadas

1. Introducción:

El siguiente proyecto está destinado a la creación de la base de datos de la empresa de mi padre, en la cual se realizan servicios de mantenimiento, modernización de ascensores llamada *ASCENSORES AQUINO*.

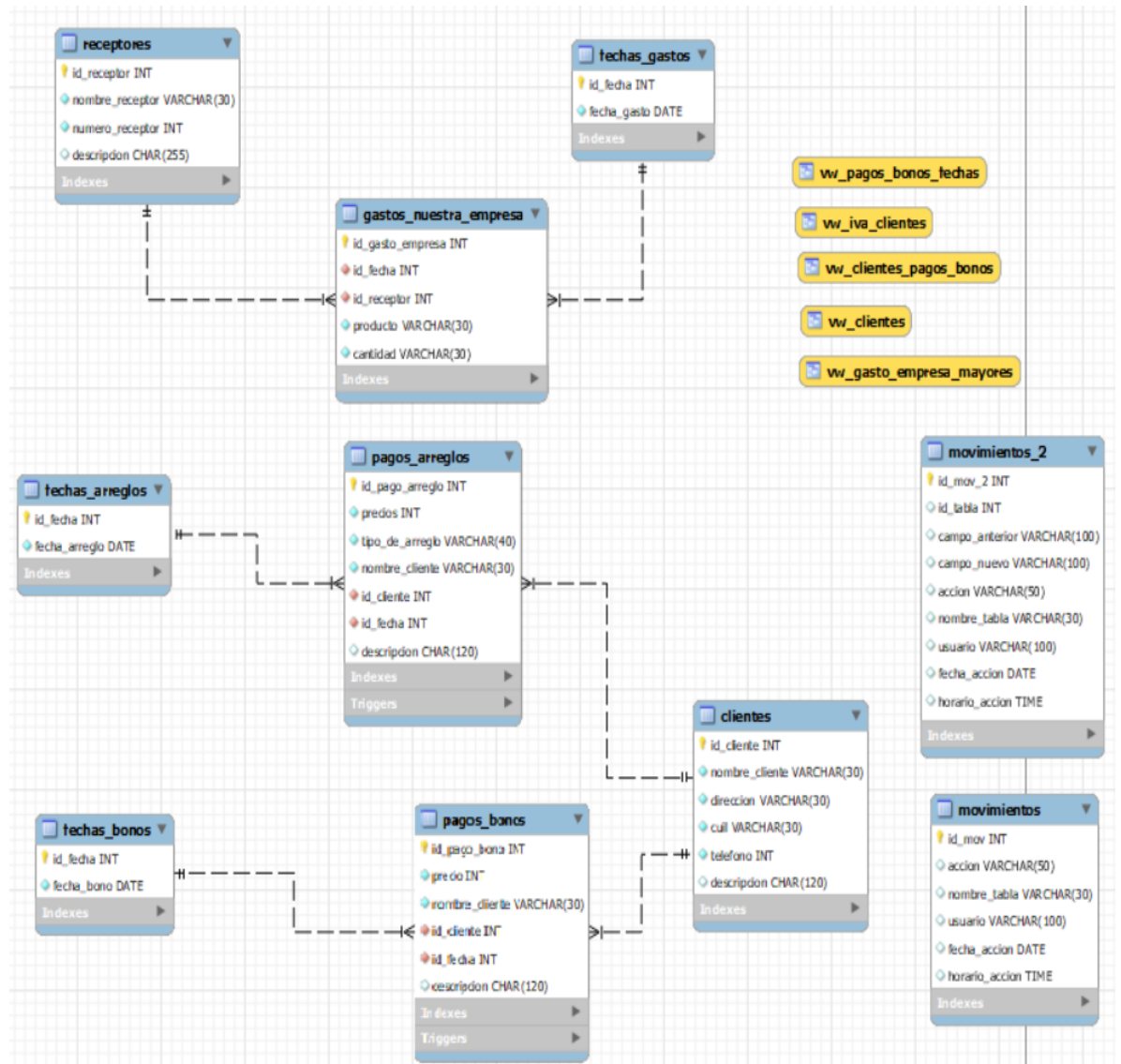
2. Objetivo:

El objetivo es llevar el control de todos los pagos que se le realizan a la empresa por sus servicios y también los gastos que la empresa realiza para sus trabajos. Contando con la información de sus fechas, quienes hicieron esos pagos, a quienes la empresa le paga para realizar sus trabajos, etc.

3. Situación problemática:

Espera resolver la desorganización de la empresa, con la información de quienes le realizan los pagos, que pagos realiza la empresa y también con la información de todas las fechas de cada pago realizado para evitar cualquier problema a futuro que pueda aparecer.

4. Diagrama Entidad-Relación:



5. Tablas:

A continuación, se va a mostrar la descripción de cada una de las tablas y cada campo de ellas:

- a) Cientes: En esta tabla se van a ver todos los clientes que tenemos hasta el día de hoy que cuentan con todos nuestros servicios. Mostrando un id (identificador de cada cliente), nombre del cliente, su dirección, su número de CUIL, su teléfono, y una descripción por cada cliente.

Field	Type	Null	Key	Default	Extra
id_cliente	int	NO	PRI	NULL	auto_increment
nombre_cliente	varchar(30)	NO		NULL	
direccion	varchar(30)	NO		NULL	
cuil	varchar(30)	NO		NULL	
telefono	int	NO		NULL	
descripcion	char(120)	YES		NULL	

- b) Fechas arreglos: En la siguiente tabla se van a mostrar todas las fechas de los pagos que nos realizan por nuestros arreglos. En la misma se detallan los id (identificador de cada fecha) y fecha_arreglo que muestra la fecha en que se realizó ese pago de arreglo.

Field	Type	Null	Key	Default	Extra
id_fecha	int	NO	PRI	NULL	auto_increment
fecha_arreglo	date	NO		NULL	

- c) Fechas bonos: En la siguiente tabla se van a mostrar todas las fechas de los pagos que nos realizan por los bonos mensuales. En la misma se detallan los id (identificador de cada fecha) y fecha_bono que muestra la fecha en que se realizó ese pago de bono.

Field	Type	Null	Key	Default	Extra
id_fecha	int	NO	PRI	NULL	auto_increment
fecha_bono	date	NO		NULL	

- d) Fechas_gastos: En la siguiente tabla se van a mostrar todas las fechas de los gastos de nuestra empresa. En la misma se detallan los id (identificador de cada fecha) y fecha_gasto que muestra la fecha en que se realizó ese pago de nuestra empresa hacia el receptor.

Field	Type	Null	Key	Default	Extra
id_fecha	int	NO	PRI	NULL	auto_increment
fecha_gasto	date	NO		NULL	

- e) Pagos_arreglos: En esta tabla se muestran los pagos que nos realizan por los arreglos que hacemos. Detallando en la misma el id (identificador de cada pago), su precio, tipo de arreglo, nombre del cliente, id_cliente que hace relación con la tabla clientes para ver que cliente realizo el pago, id_fecha que hace relación con la tabla fechas_arreglos para visualizar la fecha y una descripción de cada pago.

Field	Type	Null	Key	Default	Extra
id_pago_arreglo	int	NO	PRI	NULL	auto_increment
precios	int	NO		NULL	
tipo_de_arreglo	varchar(40)	NO		NULL	
nombre_cliente	varchar(30)	NO		NULL	
id_cliente	int	NO	MUL	NULL	
id_fecha	int	NO	MUL	NULL	
descripcion	char(120)	YES		NULL	

- f) Pagos bonos: En esta tabla se muestran los pagos de los bonos que nos realizan todos los meses por nuestros servicios. Detallando en la misma el id (identificador de cada pago), su precio, nombre del cliente, id_cliente que hace relación con la tabla clientes para ver de qué cliente recibimos el pago, id_fecha que hace relación con la tabla fechas_bonos para visualizar la fecha y una descripción de cada pago.

Field	Type	Null	Key	Default	Extra
id_pago_bono	int	NO	PRI	HULL	auto_increment
precio	int	NO		HULL	
nombre_cliente	varchar(30)	NO		HULL	
id_cliente	int	NO	MUL	HULL	
id_fecha	int	NO	MUL	HULL	
descripcion	char(120)	YES		HULL	

- g) Gastos nuestra empresa: En la siguiente tabla se muestran los gastos que realizamos para hacer nuestros trabajos. En esta tabla se detalla por cada registro su id (identificador de cada gasto), el id_fecha que hace relación a la tabla fechas_gastos para visualizar la fecha, el id_receptor que hace relación con la tabla receptores para ver quien recibe nuestro pago, y la cantidad del pago que hicimos.

Field	Type	Null	Key	Default	Extra
id_gasto_empresa	int	NO	PRI	HULL	auto_increment
id_fecha	int	NO	MUL	HULL	
id_receptor	int	NO	MUL	HULL	
producto	varchar(30)	NO		HULL	
cantidad	varchar(30)	NO		HULL	

- h) Receptores: En esta tabla podemos ver a quienes les realizamos los pagos de nuestra empresa detallando su id (identificador de cada receptor), nombre del receptor, número del receptor y una descripción breve del receptor.

Field	Type	Null	Key	Default	Extra
id_receptor	int	NO	PRI	NULL	auto_increment
nombre_receptor	varchar(30)	NO		NULL	
numero_receptor	int	NO		NULL	
descripcion	char(255)	YES		NULL	

6. Views:

Se crean 5 vistas con la información que queremos de las tablas para tenerlas almacenadas sin repetir esa búsqueda cada vez que la necesitemos.

a) VW CLIENTES:

En esta vista se muestra toda la tabla clientes.

b) VW PAGOS BONOS FECHAS:

Esta vista nos muestra la unión entre dos tablas, que trae toda la información de pagos_bonos y solo la columna fecha_bono de la tabla fechas_bonos.

c) VW CLIENTES PAGOS BONOS:

Esta vista nos muestra los nombres de los clientes donde el precio en pagos_bonos sea igual a 15000.

d) VW IVA CLIENTES:

Esta vista nos muestra la tabla pagos_bonos y se le suma una columna que se llamada suma_de_iva que lo que hace es la suma del precio más el IVA (0.21).

e) VW GASTOS EMPRESA MAYORES:

Esta vista nos muestra un conteo (de cada id de gastos_nuestra_empresa) donde en la columna cantidad el numero sea mayor a 10.000.

7. Functions:

Se crean 2 funciones para facilitar la obtención de datos de una forma automática.

FN COUNT CLIENTES:

Esta es una función en la cual le pasamos por parámetro un nombre similar o que contenga de la tabla clientes y hace un conteo de cuantos se parecen a ese nombre que le pasemos por parámetro.

FN RECEPTOR FECHA GASTO:

Esta es una función en la cual se le da una fecha de algún gasto de nuestra empresa hacia a un receptor por parámetro y te retorna el nombre del receptor al cual le hicimos ese pago en esa fecha que pasamos.

8. Stored Procedures:

Se crean 2 procedimientos y tienen un comportamiento similar al de las funciones, pero estas devuelven como resultados más columnas con la información.

SP INS FECHA BONO:

Este es un procedimiento en el cual podemos insertar un registro en la tabla fechas_bonos cada vez que lo usemos. Nos devuelve todo el registro que se ingresa por parámetro con sus respectivas columnas y después si revisamos la tabla fechas_bonos vamos a ver el registro agregado.

SP CANT TOTAL PRODUCTO:

Este es un procedimiento que está relacionado a las tablas gastos_nuestra_empresa y receptores donde le pasamos por parámetro la primera letra del nombre_receptor, la primera letra de su producto y nos devuelve para esos parámetros la cantidad total de todos los pagos que se le hizo por ese producto.

9. Triggers:

Se crea una tabla movimientos para llevar el control de todas las modificaciones que se hagan en las tablas originales, donde se le insertaran los datos de que acción, que tabla, usuario, en qué fecha y horario fueron modificados. Se crea una segunda tabla llamada movimientos_2 que cumple la misma tarea que la primera tabla sumándole la información de tres columnas id_tabla que muestra que id de esa tabla se modificó, campo_anterior que muestra el campo antes de ser modificado y campo_nuevo que muestra el campo después de ser modificado.

A continuación, se describen en detalle los 4 triggers:

TRG MOV ARREGLOS:

Este es un trigger que me avisa después de que se haga un insert en la tabla pagos_arreglos y lleva información de quien lo hizo, su fecha, etc. Hacia la tabla movimientos.

TRG MOV2 ARREGLOS:

Este es un trigger que advierte antes si se modifica el precio de la tabla pagos_arreglos y lleva la información de cual id (de pagos_arreglos) fue modificado, el precio anterior, el precio nuevo, etc. Hacia la tabla movimientos_2.

TRG MOV BONOS:

Este es un trigger que avisa antes de que se haga una eliminación de la tabla pagos_bonos e inserta con toda la información de la eliminación en la tabla movimientos.

TRG MOV2 BONOS:

Este es un trigger que después de que alguien haga una actualización de precio en la tabla pagos_bonos, nos inserta toda la información de quien lo hizo, que id (de pagos_bonos) se modificó, etc. en la tabla movimientos_2.

10. Sentencias:

Se crea un usuario dándole el permiso de solo lectura:

- Usuario: agustinrodriguez
- Contraseña: ACD651

Se crea otro usuario dándole los permisos de lectura, inserción y modificación:

- Usuario: manuelfernandez
- Contraseña: ABC481

11. Backup:

Se crea un backup de toda la base de datos en un único archivo .sql para poder exportarlo. En este archivo van a aparecer:

- 1) La creación de la base de datos.
- 2) La creación de tablas.
- 3) La inserción de las tablas.
- 4) La creación de vistas.
- 5) La creación de funciones.
- 6) La creación de procedimientos.
- 7) La creación de triggers.

12. Herramientas utilizadas:

MySQL Workbench 8.0 CE: Para la creación de la base de datos.

PDF: Para la creación de la presentación.