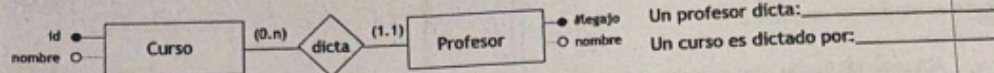


Primera fecha 2024

Entidad/Relación

Indique cómo interpreta las cardinalidades del siguiente modelo
(esto se usará para leer las cardinalidades de la solución que usted proponga)



Un profesor dicta: _____

Un curso es dictado por: _____

Enunciado:

Una empresa de seguros necesita una base de datos para la gestión de pólizas, siniestros, clientes y la relación con talleres asociados para reparaciones de vehículos.

De las pólizas se registra un número único, fecha de inicio, fecha de fin y estado (vigente, cancelada, vencida). De las pólizas vehiculares se registra el número de patente del vehículo, un deducible específico y la cobertura de daños; para las pólizas de hogar se guarda la dirección de la vivienda y la cobertura (una descripción de las condiciones acordadas)

De la pólizas se conoce los datos de su titular, estos son CUIL/CUIT, nombre completo, dirección y teléfono. Cada titular puede tener varias pólizas y una póliza solo le pertenece a un titular.

Además, la empresa de seguros trabaja con ciertos talleres especializados para las reparaciones de vehículos asegurados. De cada taller se registra un ID único, nombre, y la o las especialidades que tiene (carrocería, mecánica, pintura). Para cada especialidad de un taller, este ofrece garantías, de la cual se guarda un ID, los detalles de la garantía y las fechas de inicio y fin de la misma, estas son generales y no dependen del vehículo que se atiende.

Se deben registrar únicamente los siniestros de automóviles. Estos siniestros son registrados con una fecha de denuncia y una descripción del mismo, junto con la póliza de cobertura. A un siniestro vehicular se le pueden asignar una o más reparaciones (cada una con su descripción y fecha de aprobación) que estarán asociadas a los talleres y especialidades registrados por la compañía. Tenga en cuenta que a un siniestro solo se le puede asignar, como máximo, una reparación de cada especialidad (no podría haber dos reparaciones de una misma especialidad para un mismo siniestro).

Actividades:

1. Realizar el modelo E/R
2. Realizar la transformación del modelo de E/R al modelo relacional.

Normalización

Se dispone del siguiente esquema del dominio de las empresas de seguros y registro de siniestros.

POLIZAS(#póliza, #cliente, cuil, nombre, dirección, teléfono,
tipo_seguro, fecha_inicio, fecha_fin, #siniestro, fecha_denuncia, estado_siniestro,
#perito, #cuota, detalle_reparacion)

- De las pólizas se conoce un ID único (#póliza), un tipo de seguro, y las fechas de inicio y fin para el asegurado titular de la misma. Además se registran las cuotas pagas que le corresponden a las pólizas, cada una identificada con #cuota, este es un número secuencial a partir de 1 por cada póliza.
- De los clientes se conoce un ID único (#cliente), su cuil, nombre, dirección y teléfono. Un cliente puede poseer varias pólizas. Una póliza pertenece a un único cliente
- De un siniestro se conoce un ID único (#siniestro), una fecha de denuncia y el estado (pendiente, en proceso, resuelto). Debe registrarse a su vez las pólizas asociadas al siniestro, ya que pueden estar involucrados más de un vehículo.
- Sobre cada siniestro distintos peritos (expertos en investigar el accidente) realizan una inspección, cada perito es identificado con un #perito.
- Sobre cada póliza involucrada en un siniestro se registran un conjunto de posibles reparaciones (detalle_reparación), los cuales son predefinidos, por lo que podrían repetirse.

PARCIAL_2024_1ERA_FECHA.excalidraw

▼ Normalizacion

POLIZAS(#póliza, #cliente, cuil, nombre, dirección, teléfono, tipo_seguro, fecha_inicio, fecha_fin, #siniestro, fecha_denuncia, estado_siniestro, #perito, #cuota, detalle_reparacion)

Dependencias funcionales

df1: #poliza → tipo_seguro, fecha_inicio, fecha_fin, #cliente ✗

df2: #poliza → tipo_seguro, fecha_inicio, fecha_fin, cuil

df3: #cliente → cuil, nombre, direccion, telefono ✗

df4: cuil → #cliente, nombre, direccion, telefono ✗

df5: #siniestro → fecha_denuncia, estado_siniestro ✗

cc1: {#poliza, #siniestro, #perito, #cuota, detalle_operacion}

Proceso de normalizacion

Analizamos el esquema Polizas, vemos que no cumple con BCNF ya que existe al menos una dependencia funcional donde el determinante no es superclave del esquema. Como lo es {#siniestro} de la df5. Particionamos en base a esta.

P1(#siniestro, fecha_denuncia, estado_siniestro)

P2(#póliza, #cliente, cuil, nombre, dirección, teléfono, tipo_seguro, fecha_inicio, fecha_fin, #siniestro, #perito, #cuota, detalle_reparacion)

Como P1 n P2 es {#siniestro}, clave en P1, no perdimos informacion. No perdimos DFs ya que en P1 vale df5 y en P2 df1, df2, df3, df4 por validacion simple.

P1 esta en BCNF ya que el determinante de la df5 {#siniestro} es superclave en P1.

Vemos que P2 no cumple con BCNF ya que existe al menos una dependencia funcional donde el determinante no es superclave del

esquema. Como lo es {#cliente} de la df3. Particionamos en base a esta.

P3(#cliente, cuil, nombre, direccion, telefono)

P4(#póliza, #cliente, tipo_seguro, fecha_inicio, fecha_fin, #siniestro,
#perito, #cuota, detalle_reparacion)

Como P3 n P4 es {#cliente}, clave en P3, no perdimos informacion.
No perdimos DFs ya que en P3 valen df3 y df4. Y en P4 vale df1. Que paso con df2? No se pierde ya que es posible acceder con su determinante a todos sus determinados, ya sea directa o indirectamente. cuil no esta en P4, sin embargo la dependencia funcional no se pierde ya que existe una relacion indirecta a traves de la dependencia funcional

df3: #cliente → cuil, nombre, direccion, telefono

Donde recuperamos cuil. Se tiene un equivalencia entre dfs.

P3 esta en BCNF ya que los determinantes de la df3 y df4, {#cliente} y {cuil}, son claves del esquema.

Vemos que P4 no cumple con BCNF ya que existe al menos una dependencia funcional donde el determinante no es superclave del esquema. Como lo es {#poliza} de la df1. Particionamos en base a esta.

P5(#poliza, tipo_seguro, fecha_inicio, fecha_fin, #cliente)

P6(#póliza, #siniestro, #perito, #cuota, detalle_reparacion)

Como P5 n P6 es {#poliza}, clave en P5, no perdimos informacion.
No perdimos DFs ya que en P5 vale df1 por validacion simple.
P5 esta en BCNF ya que el determinante de la df5 {#poliza} es superclave en P5.

Notamos que en P6 quedaron todos los atributos de la clave, por lo que toda dependencia funcional que valga sera trivial.

CP: {#póliza, #siniestro, #perito, #cuota, detalle_reparacion}

Esquemas en BCNF: P1, P3, P5, P6.

Buscamos las dependencias multivaluadas en la ultima particion.

Dependencias multivaluadas

dm1: #poliza →> #cuota

dm2: #siniestro →> #perito

dm3: #poliza, #siniestro → > detalle_operacion

Vemos que P6 no esta en 4FN ya que valen DMs que no son triviales.
Como lo es dm1. Particionamos por esta.

P7(#poliza, #cuota)

P8(#póliza, #siniestro, #perito, detalle_reparacion)

P7 esta en 4FN ya que la dm1 que vale en ella es trivial.

P8 no esta en 4FN ya que valen DMs que no son triviales. Como lo es dm2.
Particionamos por esta.

P9(#siniestro, #perito)

P10((#póliza, #siniestro, detalle_reparacion))

P9 esta en 4FN ya que la dm2 que vale en ella es trivial.

P10 esta en 4FN ya que la dm3 que vale en ella es trivial.

Esquemas en 4FN

P1(#siniestro, fecha_denuncia, estado_siniestro)

P3(#cliente, cuil, nombre, direccion, telefono)

P5(#poliza, tipo_seguro, fecha_inicio, fecha_fin, #cliente)

P7(#poliza, #cuota)

P9(#siniestro, #perito)

P10((#póliza, #siniestro, detalle_reparacion))

Aplicar y explicar el proceso de normalización visto en la materia. Considerar que el esquema ya se encuentra en 1FN.

Álgebra Relacional

Dado el siguiente esquema:

Cliente (#cliente, dni, nombre, dirección, teléfono)
Perito (#perito, nombre_perito, apellido_perito)
Siniestro (#siniestro, fecha_denuncia, descripcion_siniestro, #taller, #póliza)
Póliza (#póliza, fecha_inicio, fecha_fin, #cliente)
Peritaje (fecha_peritaje, #perito, #siniestro, evaluación)
Tipo_Seguro (#tipo_seguro, nombre_tipo_seguro)
Póliza_Tipo_Seguro (#póliza, #tipo_seguro)

Listar el nombre y apellido de todos los peritos que hayan tratado con todos los tipos de seguros.

MySQL

Se ha creado una base de datos siguiendo el esquema definido en el ejercicio de Álgebra Relacional, con una relación uno a uno para las tablas y atributos. En esta base de datos, crear un stored procedure que sirva para agregar, de forma transaccional, un nuevo siniestro, junto al registro del perito, que realiza el peritaje. El stored procedure debe recibir los siguientes parámetros: #póliza, descripcion_siniestro, #taller, #perito, nombre_perito, apellido_perito, evaluación. El #siniestro es un autoincremental. La fecha_denuncia será la fecha actual.

Visualización de datos

Una cadena de supermercados quiere poder visualizar:

- La cantidad de productos vendidos de cada categoría para todas las sucursales, para conocer que tipo de producto es el que más se vende.
- El total ingresos de la sucursal número 10, mes a mes, durante los últimos 12 meses, para determinar si hubo o no un incremento de los ingresos.

Para ello dispone de una base de datos con el siguiente esquema:

Venta (id_venta, fecha_venta, id_sucursal, monto_total)
Item_Venta (id_venta, id_producto, cant)
Sucursal (id_sucursal, ubicación, cant_empleados)
Producto (id_producto, nombre_producto, desc_producto, precio_unit, categoria)
Cliente (id_cliente, nombre_cliente, apellido_cliente, tipo_cliente)

Determine qué tipo de gráfico de los vistos en la materia podría utilizar y justifique su elección. Con el esquemas proporcionado, elegir cuáles tablas son relevantes para presentar el análisis visual propuesto anteriormente

Álgebra

PERITOS_SEGUROS \leftarrow h #perito, #tipo_seguro (PERITO \bowtie PERITAJE \bowtie SINIESTRO
 \bowtie POLI
 ZA_TIPO_SEGURO)

PERITOS_CUMPLEN \leftarrow (PERITOS_SEGUROS % h #tipo_seguro TIPO_SEGURO)

```
RESULTADO ← h nombre_perito, apellido_perito (PERITOS_CUMPLEN |X| P  
ERITO)
```

SQL

```
DELIMITER ??
```

```
CREATE PROCEDURE agregar_siniestro_perito(IN p_poliza INT, IN descripci  
on_siniestro STRING,
```

```
IN taller INT, IN perito IN
```

```
T, IN nombre_perito STRING,
```

```
IN apellido_perito STRIN
```

```
G, IN evaluacion STRING)
```

```
BEGIN
```

```
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
```

```
    BEGIN
```

```
        ROLLBACK;
```

```
    END;
```

```
    DECLARE fecha DATE;
```

```
    DECLARE ultimo_siniestro INT;
```

```
    START TRANSACTION
```

```
        SET fecha = NOW();
```

```
        INSERT INTO SINIESTRO(fecha_denuncia, descripcion_siniestro, t  
aller, poliza)
```

```
        VALUES(fecha, descripcion_siniestro, taller, poliza);
```

```
        SET ultimo_siniestro = LAST INSERT ID();
```

```
        INSERT INTO PERITO(perito, nombre_perito, apellido_perito)
```

```
        VALUES(perito, nombre_perito, apellido_perito)
```

```
INSERT INTO PERITAJE (fecha_peritaje, perito, siniestro, evaluacion)
VALUES (fecha, perito, ultimo_siniestro, evaluacion)

COMMIT

END

DELIMITER ;
```

Visualizacion de datos

Para el primer caso usaria un grafico de barras, donde en el eje X irian las distintas categorias y en el eje Y la cantidad de ventas de productos de esa categoria. Elijo este formato ya que se puede visualizar bien la diferencia de ventas entre una categoria y otra, en un grafico de torta por ejemplo, no se cuantas sucursales son y que % de ventas tendre, por lo que no me puedo asegurar que el grafico de tortas sea eficiente para este caso. Las tablas relevantes son Producto (id_producto y categoria) e ItemVenta (id_venta, id_producto, cant)

Para el segundo caso utilizaria un grafico de lineas, ya que me permite analizar los ingresos en relacion con el tiempo. En el eje Y va el tiempo en meses y en el eje X los ingresos. Este grafico me permite ver como evoluciona los ingresos a traves del tiempo de forma clara. Las tablas que necesitaria seria Venta (fecha_venta, id_sucursal, monto_total)