

**Primer Fecha - 13/11/2024**

Nº de Alumno: _____	Apellido y Nombre: _____			
Uso interno de la cátedra				
ER	Norm	AR	MySQL	Visualización
Corrigió: _____				

**BBDD1 Parcial**  
1ra fecha 13/11/2024

**Entidad/Relación**

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

```

    graph LR
      Curso[Curso] -- "0..n" -->|dicta| Profesor[Profesor]
      subgraph Curso_attributes [ ]
        direction TB
        C1((Id)) --- C1_label["Id"]
        C2((nombre)) --- C2_label["nombre"]
      end
      subgraph Profesor_attributes [ ]
        direction TB
        P1((#legajo)) --- P1_label["#legajo"]
        P2((nombre)) --- P2_label["nombre"]
      end
  
```

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/CIUT, 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 uno 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 cui, 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.

**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, descripción_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, descripción\_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:

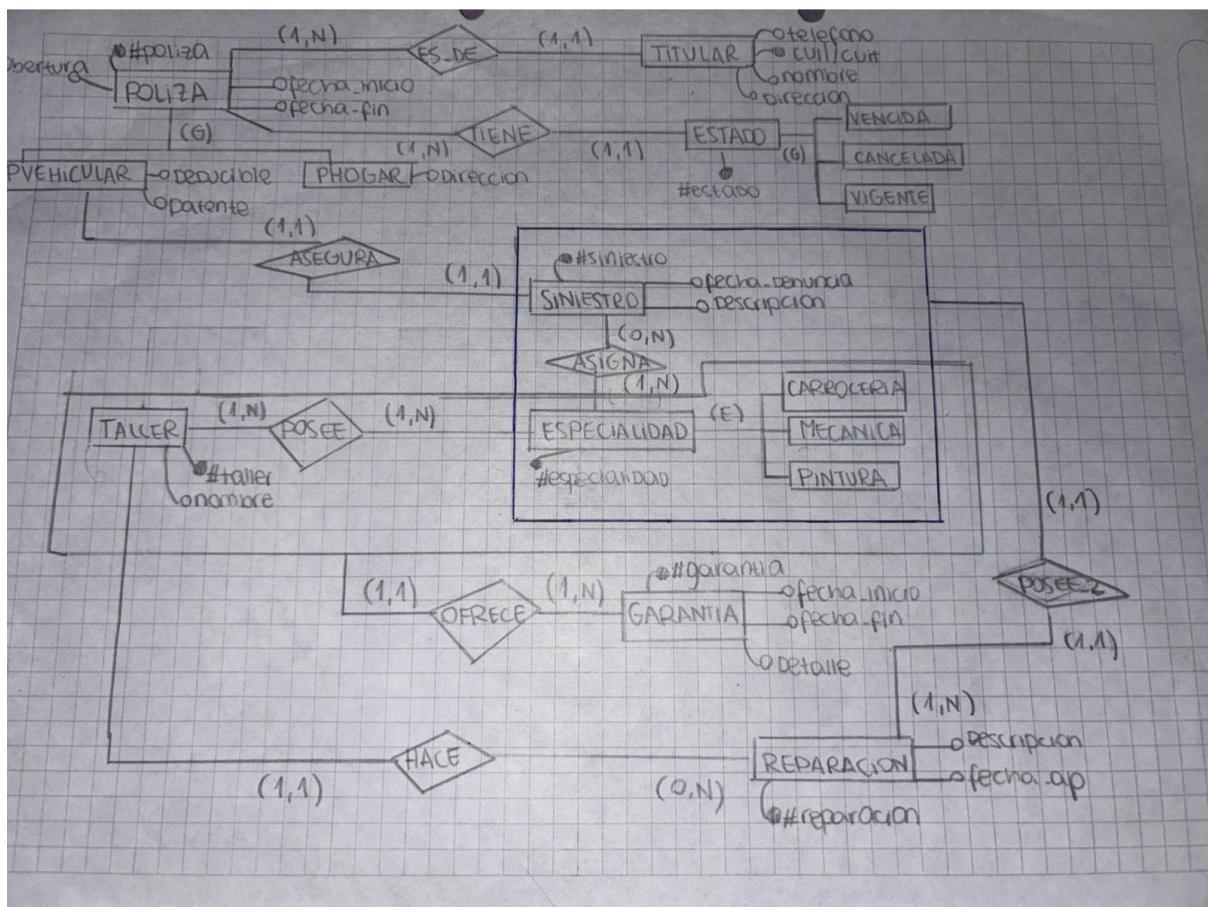
- La cantidad de productos vendidos de cada categoría para todas las sucursales, para conocer que tipo de producto es el que más se vende.
- El total ingresos de la sucursal número 10, mes a mes, durante los últimos 12 meses, para determinar si hubo o no un incremento de los ingresos.

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, categoría)
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

**ER**

TITULAR = (CUIL/CUIT, teléfono, nombre, dirección)  
 POLIZA = (#póliza, cobertura, fecha\_inicio, fecha\_fin)  
 ES\_DE (#póliza, CUIL\_CUIT)  
 ESTADO (#Estado, tipo\_Estado)  
 TIENE (#póliza, #Estado)  
 PHOGAR (#póliza, dirección)  
 PVEHICULAR (#póliza, vehículo, patente)  
 SINIESTRO (#siniestro, fecha\_denuncia, descripción)  
 ASEGURA (#póliza, #siniestro)  
 ESPECIALIDAD (#especialidad, tipoEspecialidad)  
 ASIGNA (#siniestro, #especialidad)  
 TALLER (#taller, nombre)  
 POSEE (#taller, #especialidad)  
 GARANTIA (#garantia, fecha\_inicio, fecha\_fin, detalle)  
 OFRECE (#garantia, #taller, #especialidad)  
 REPARACION (#reparacion, descripción, fecha\_ap)  
 HACE (#reparacion, #taller)  
 POSEE 2 (#siniestro, #especialidad, #reparacion)

### Normalización

DF1: #póliza -> tipo\_seguro, fecha\_inicio, fecha\_fin, #cliente

DF2: #póliza -> tipo\_seguro, fecha\_inicio, fecha\_fin, cuil

DF3: #cliente -> cuil, nombre, dirección, teléfono

DF4: cuil -> #cliente, nombre, dirección, teléfono

DF5: #siniestro -> fecha\_denuncia, estado\_siniestro

CC: {#poliza, #siniestro, #perito, #cuota, detalle\_reparacion}

- POLIZAS no está en BCNF ya que existe al menos la DF5 tal que {#siniestro} que es el determinante de la DF no es superclave en POLIZAS. Particiono para llevar a BCNF considerando DF5.

P1 (#siniestro, fecha\_denuncia, estado\_siniestro)

P2 (#póliza, #cliente, cuil, nombre, dirección, teléfono, tipo\_seguro, fecha\_inicio, fecha\_fin, #siniestro, #perito, #cuota, detalle\_reparacion)

P1 ∩ P2 = clave en P1 {#siniestro}. No se pierde información.

En P1 vale DF5.

En P2 vale DF1, DF2, DF3, DF4.

P1 está en BCNF porque vale la DF5 tal que {#siniestro} que es el determinante de la DF es superclave en P1.

- P2 no está en BCNF ya que existe al menos la DF3 tal que {#cliente} que es el determinante de la DF no es superclave en P2. Particiono para llevar a BCNF considerando P2.

P3 (#cliente, cuil, nombre, dirección, teléfono)

P4 (#póliza, #cliente, tipo\_seguro, fecha\_inicio, fecha\_fin, #siniestro, #perito, #cuota, detalle\_reparacion)

P3 ∩ P4 = clave en P3 {#cliente}. No se pierde información.

En P3 vale DF3 y DF4.

En P4 vale DF1.

¿Qué pasa con DF2? No se pierde por el siguiente razonamiento:

- #póliza -> tipo\_seguro, fecha\_inicio, fecha\_fin, #cliente

- #cliente -> cuil, nombre, dirección, teléfono

Vale DF1 tal que: #póliza -> tipo\_seguro, fecha\_inicio, fecha\_fin, #cliente

Con DF3: #cliente -> cuil, nombre, dirección, teléfono

Entonces en P4 tengo #cliente, a partir del cual recupero cuil. Puedo decir por el razonamiento seguido que tengo una equivalencia entre DFs.

No se perdieron DFs por validación simple.

P3 está en BCNF porque vale la DF3 tal que {#cliente} que es el determinante de la DF es superclave en P3, y además vale la DF4 tal que {cuil} que es el determinante de la DF es superclave en P3.

- P4 no está en BCNF ya que existe al menos la DF1 tal que {#póliza} que es el determinante de la DF no es superclave en P4. Particiono para llevar a BCNF considerando P4.

P5 (#póliza, tipo\_seguro, fecha\_inicio, fecha\_fin, #cliente)

P6 (#póliza, #siniestro, #perito, #cuota, detalle\_reparacion)

P5  $\cap$  P6 = clave en P5 {#póliza}. No se pierde información.

En P5 vale DF1.

No se perdieron DFs por validación simple.

P5 está en BCNF porque vale la DF1 tal que {#póliza} que es el determinante de la DF es superclave en P5. P6 está en BCNF porque cualquier dependencia funcional que valga es trivial.

Particiones en BCNF:

P1 (#siniestro, fecha\_denuncia, estado\_siniestro)

P3 (#cliente, cuil, nombre, dirección, teléfono)

P5 (#póliza, tipo\_seguro, fecha\_inicio, fecha\_fin, #cliente)

P6 (#póliza, #siniestro, #perito, #cuota, detalle\_reparacion)

CP: {#poliza, #siniestro, #perito, #cuota, detalle\_reparacion}

Dependencias multivaluadas válidas en P6:

DM1) #póliza  $\rightarrow\!\!\!>$  #cuota

DM2) #póliza, #siniestro  $\rightarrow\!\!\!>$  detalle\_reparacion

DM3) #siniestro  $\rightarrow\!\!\!>$  #perito

- Dado que en P6 vale al menos la DM1 que no es trivial (una dependencia multivaluada  $X \Rightarrow Y$  en un esquema  $R$  es trivial cuando la unión de los atributos  $X \cup Y$  abarca todos los atributos de  $R$ ), P6 no cumple la definición de 4FN, entonces particiono P6 considerando la DM1.

P7 (#póliza, #cuota)

P8 (#póliza, #siniestro, #perito, detalle\_reparacion)

P7 está en 4FN porque vale la DM1 tal que DM1 es trivial en P7.

- Dado que en P8 vale al menos la DM2 que no es trivial, P8 no cumple la definición de 4FN, entonces particiono P8 considerando DM2.

P9 (#póliza, #siniestro, detalle\_reparacion)

P10 (#póliza, #siniestro, #perito)

P9 está en 4FN porque vale la DM2 tal que DM2 es trivial en P9.

- Dado que en P10 vale al menos la DM3 que no es trivial, P10 no cumple la definición de 4FN, entonces particiono P10 considerando DM3.

P11 (#siniestro, #perito)

P12 (#póliza, #siniestro)

P11 está en 4FN porque vale la DM3 tal que DM3 es trivial P11.

¿Qué pasa con P12? ¿Está en 4FN? P12 no está en 4FN. Aparece una nueva DM (sólo es válida en P12):

DM4) #póliza ->> #siniestro

P12 está en 4FN porque vale la DM4 tal que DM4 es trivial en P12.

#### Esquemas resultantes en 4FN:

P1 (#siniestro, fecha\_denuncia, estado\_siniestro)

P3 (#cliente, cuil, nombre, dirección, teléfono)

P5 (#póliza, tipo\_seguro, fecha\_inicio, fecha\_fin, #cliente)

P7 (#póliza, #cuota)

P9 (#póliza, #siniestro, detalle\_reparacion)

P11 (#siniestro, #perito)

Se elimina el esquema P12 ya que es proyección de P9.

CP: {#poliza, #siniestro, #perito, #cuota, detalle\_reparacion}

#### AR

$\text{PERTIPO} \leftarrow \Pi_{\#perito, \#tiposeguro}(\text{PERITAJE} | X | \text{SINIESTRO} | X | \text{POLIZA\_TIPO\_SEGURO})$

$\text{TIPOSEG} \leftarrow \Pi_{\#tiposeguro}(\text{TIPO\_SEGURO})$

$\Pi_{\text{nombre\_perito}, \text{apellido\_perito}}((\text{PERTIPO} \% \text{TIPOSEG}) | X | \text{PERITO})$

#### MySQL

```
DELIMITER //

CREATE PROCEDURE `agregar_siniestro` (IN póliza INTEGER, IN
descripción_siniestro VARCHAR(45), IN taller INTEGER, IN perito
INTEGER, IN nombre_perito VARCHAR(30), IN apellido_perito VARCHAR(30),
IN evaluación VARCHAR(30))

BEGIN
    DECLARE current_date_only DATE;
    DECLARE last_id INTEGER;

    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
    END;

    START TRANSACTION;
    SET current_date_only = CURDATE();
```

```

    INSERT INTO siniestro(fecha_denuncia, descripcion_siniestro,
#taller, #póliza)
    VALUES(current_date_only, descripcion_siniestro, taller, póliza);

    SET last_id = LAST_INSERT_ID();

    -- Como me mandan #perito, nombre_perito y apellido_perito entiendo
que el perito no existe y necesito insertarlo.
    INSERT INTO perito(#perito, nombre_perito, apellido_perito)
    VALUES(perito, nombre_perito, apellido_perito);

    INSERT INTO peritaje(fecha_peritaje, #perito, #siniestro,
evaluación)
    VALUES(current_date_only, perito, last_id, evaluación);

    COMMIT;
END //

DELIMITER ;

```

## Visualización

Para el primer punto utilizaría un gráfico de barras ya que permite comparar qué productos se venden más por categoría en la sucursal. En el eje X se representarían las categorías de productos, y en el eje Y la cantidad vendida correspondiente a cada categoría.

Para el segundo punto utilizaría un gráfico de líneas ya que muestra las relaciones de los cambios en los datos en un período de tiempo, en este caso la evolución mensual de los ingresos, viendo los ascensos y descensos en las ventas. En el eje X se representarían los meses, y en el eje Y el total de ingresos correspondiente a cada mes.

Tablas importantes para el primer punto:

- **Producto** (id\_producto, nombre\_producto, desc\_producto, precio\_unit, categoria)
- **Item\_Venta** (id\_venta, id\_producto, cant)

Las tablas Producto e Item\_Venta son relevantes para el análisis, ya que la primera (Producto) identifica las categorías de los productos (atributo categoria), mientras que la segunda (Item\_Venta) proporciona los datos numéricos (atributo cant) necesarios para calcular la cantidad total de productos vendidos por cada categoría. Ambas se relacionan mediante el campo id\_producto, que permite vincular cada ítem vendido con su categoría correspondiente.

Tablas importantes para el segundo punto:

- **Sucursal** (id\_sucursal, ubicación, cant\_empleados)
- **Venta** (id\_venta, fecha\_venta, id\_sucursal, monto\_total)

Las tablas Venta y Sucursal son relevantes para el análisis, ya que la primera (Venta) contiene los datos económicos de cada operación, como el monto\_total y la fecha\_venta, necesarios para calcular los ingresos mensuales, mientras que la segunda (Sucursal) permite identificar la sucursal específica (atributo id\_sucursal) sobre la cual se realizará el análisis. Ambas se relacionan a través del campo id\_sucursal, que vincula cada venta con la sucursal donde se realizó.

**Segunda Fecha - 27/11/2024**

Apellido del Alumno: _____	Apellido y Nombre: _____										
BBDD1 Parcial	Uso interno de la cátedra										
2da Fecha 27/11/2024	<table border="1"> <thead> <tr> <th>ER</th> <th>Norm</th> <th>AR</th> <th>MySQL</th> <th>Visualización</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>	ER	Norm	AR	MySQL	Visualización					
ER	Norm	AR	MySQL	Visualización							
	Corrigió: _____										

**Entidad/Relación**

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

•	Curso	dicta	(1..1)	Profesor	• legajo nombre	Un profesor dicta:
						Un curso es dictado por:

**Enunciado:**

Una vinoteca quiere modelar una base de datos para almacenar los productos que ofrece y las muestras a realizar sobre estos productos.

De cada vino que la vinoteca ofrece se sabe su nombre, bodega y los distintos varietales (tipos de uva) que lo componen. Por ejemplo: Malbec, Merlot, Moscatel y Tempranillo entre otras) que componen el vino, además del porcentaje de dicho varietal que lo compone. Por ejemplo: un vino puede estar compuesto por un 70% Malbec y 30% Merlot. Si bien cada vino puede tener varios varietales, cada uno de ellos puede aparecer una única vez por vino.

Adicionalmente, los diferentes varietales se cultivan en viñedos. Un viñedo puede tener diferentes varietales en sus tierras así como un varietal puede ser objeto de cultivo de diversos viñedos. A partir de lo anterior, se da la saber de cuál de los viñedos fue extraido el varietal que compone cada vino. De cada varietal se sabe su nombre y su tipo. De los viñedos se sabe su nombre, dirección y altura.

La vinoteca organiza diferentes eventos para que clientes o expertos en vino puedan probar sus diferentes vinos. De cada evento se registra un nombre, la fecha en la que se realiza y los vinos que se eligen para ser presentados en ese evento. Un mismo vino puede ser presentado en diferentes eventos. De cada experto que haya probado un vino presentado en un evento se registra una reseña y el puntaje que éste le otorgó al vino (un valor de 0 a 100). De cada experto se sabe su nombre, apellido, cuit, fecha de nacimiento, nacionalidad y experiencia.

Es importante para la vinoteca registrar la opinión de los expertos que han probado y resenizado el vino presentado, información que servirá luego para poder promocionar sus vinos.

**Actividades:**

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

**Normalización**

Dado el siguiente esquema que representa la actividad en un parque de diversiones.

**PARQUEDIVERSIONES** (id\_atractivo, nombre\_atractivo, descripción\_atractivo, id\_categoria\_atractivo, nombre\_categoria, cuit\_empleado, nombre\_apellido\_empleado, nro\_legajo\_empleado, dia\_semana, id\_agente\_mantenimiento)

- De cada atractivo (identificado por `id_atractivo`), se conoce su nombre (`nombre_atractivo`), una descripción (`descripción_atractivo`) y las categorías a la que pertenece. Uno o más atractivos pueden tener el mismo nombre y/o descripción. Puede haber diferentes atractivos por categoría.
- El id de la categoría es un valor único (`id_categoria_atractivo`). De cada categoría se conoce su nombre. Un nombre de categoría puede repetirse para diferentes id de categoría.
- De cada empleado que trabaja en el parque de diversiones, se conocen además de sus datos personales (`cuil_empleado`, `nombre_apellido_empleado`), un número de legajo (`nro_legajo_empleado`) el cual, al igual que el cuil, es único en el sistema.
- En cada atractivo hay asignados diversos empleados, pero dado un atractivo y un empleado asignado se conoce el día de la semana que le corresponde.
- Un atractivo puede ser mantenido por distintos empleados en un día determinado.
- Un empleado puede tener asignado más de un atractivo por día
- El parque de diversiones también dispone de personal de mantenimiento, que están disponibles para atender a los diversos atractivos del parque.

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

### Álgebra Relacional

Algoritmo para analizar la pérdida de dfs.	Algoritmo para encontrar $X^+$
<pre> Res = x Mientras Res cambia   Para i= 1 to cant_de_particiones_realizadas     Res = Res U((Res ∩ R<sub>i</sub>)<sup>*</sup> ∩ R<sub>i</sub>)   </pre>	<pre> Result:= X While (hay cambios en result) do   For (cada dependencia funcional Y-&gt;Z en F ) do     if (Y ⊆ result) then result := result U Z   </pre>

Dado el siguiente esquema:

PRODUCTO (#producto, #categoria, descripción\_producto)  
CATEGORIA (#categoria, descripción\_categoria)  
MAYORISTA (#mayorista, cuil, razón\_social)  
PRECIO\_MAYORISTA (#producto, #mayorista, precio)  
VENTA (#venta, #producto, #mayorista, #cliente, cantidad, fecha)  
CLIENTE (#cliente, cuil, nombre\_cliente, apellido\_cliente)

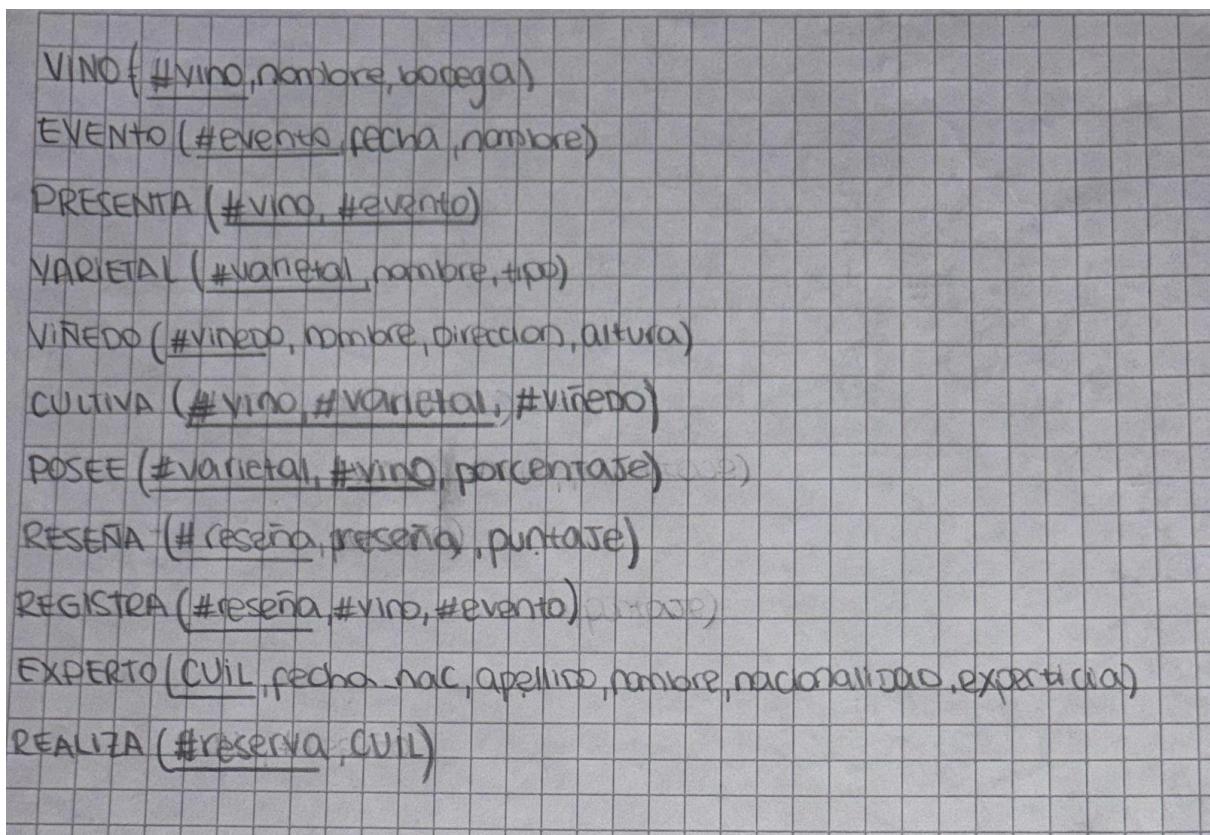
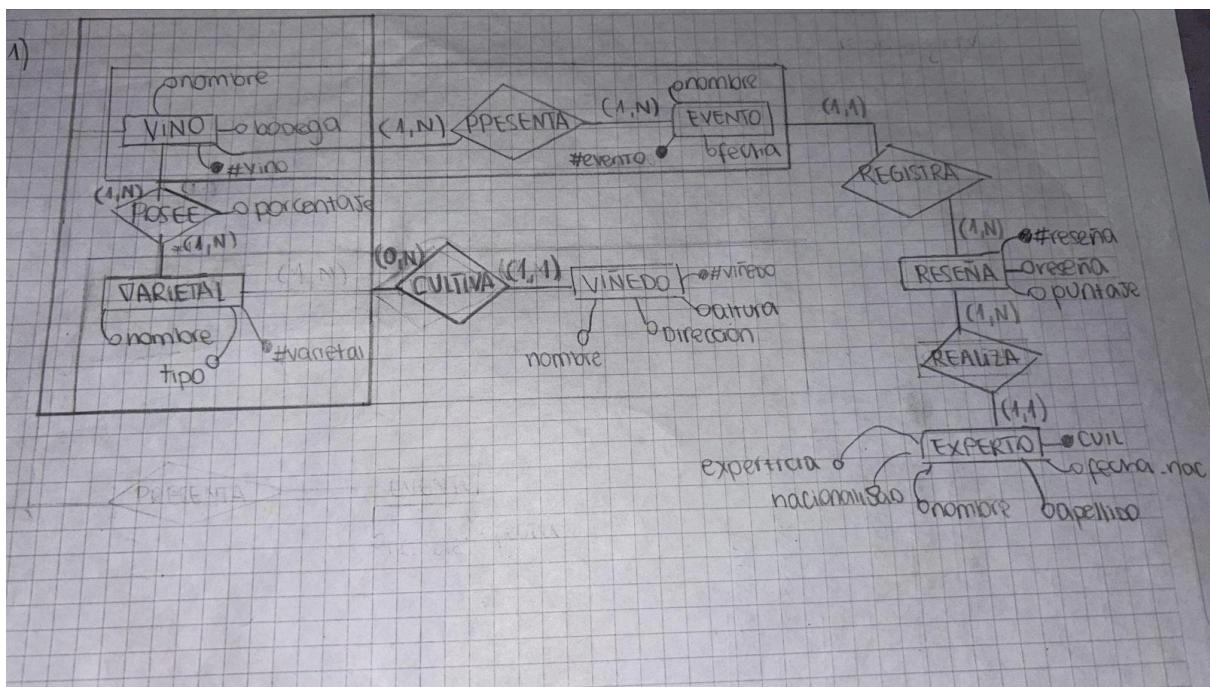
A) Listar el id de los mayoristas que han vendido productos de todas las categorias de las cuales ofrecen productos.

### MySQL

Se dispone de una base de datos para la gestión de subastas judiciales, permitiendo registrar la participación de oferentes, previamente inscritos, en estas. Para cada subasta se poseen diferentes ofertas, cada una con un monto superior a la oferta previa, de estas se registra, además del monto, el oferente correspondiente y la fecha. En la tabla "Oferta" siempre se guarda la última oferta válida. Es necesario registrar un historial de cada oferta realizada.

Subasta\_Judicial (#subasta, fecha\_comienzo, fecha\_fin)  
Inscripto\_Subasta(#inscripto\_subasta, #oferedente, #subasta, fecha\_inscripción)  
Oferente (#oferedente, nombre, apellido, cuil, localidad)  
Oferta (#oferta, #subasta, monto, fecha\_oferta)  
Oferta\_Historial (#oferta\_historial, #oferedente, #subasta, monto, fecha\_oferta, fecha\_auditoria)

1) Generar un trigger que guarda la última oferta en la tabla "Oferta\_Historial" antes de grabar la nueva oferta generadora.

**ER**

### Normalización

DF1: `id_atractivo -> nombre_atractivo, descripcion_atractivo`

DF2: `id_categoria_atractivo -> nombre_categoria`

DF3: `cuil_empleado -> nombre_apellido_empleado, nro_legajo_empleado`

DF4: `nro_legajo_empleado -> nombre_apellido_empleado, cuil_empleado`

DF5: id\_atractivo, cuil\_empleado → dia\_semana

DF6: id\_atractivo, nro\_legajo\_empleado → dia\_semana

CC1: {id\_atractivo, id\_categoria\_atractivo, cuil\_empleado, id\_agente\_mantenimiento}

CC2: {id\_atractivo, id\_categoria\_atractivo, nro\_legajo\_empleado, id\_agente\_mantenimiento}

- PARQUEDIVERSIONES no está en BCNF ya que existe al menos la DF1 tal que {id\_atractivo} que es el determinante de la DF no es superclave en PARQUEDIVERSIONES. Particiono para llevar a BCNF considerando DF1.

P1 (id\_atractivo, nombre\_atractivo, descripcion\_atractivo)

P2 (id\_atractivo, id\_categoria\_atractivo, nombre\_categoria, cuil\_empleado, nombre\_apellido\_empleado, nro\_legajo\_empleado, dia\_semana, id\_agente\_mantenimiento)

P1 ∩ P2 = clave en P1 {id\_atractivo}. No se pierde información.

En P1 vale DF1.

En P2 vale DF2, DF3, DF4, DF5 y DF6.

No se perdieron DFs por validación simple.

P1 está en BCNF porque vale la DF1 tal que {id\_atractivo} que es el determinante de la DF es superclave en P1.

- P2 no está en BCNF ya que existe al menos la DF2 tal que {id\_categoria\_atractivo} que es el determinante de la DF no es superclave en P2.

Particiono para llevar a BCNF considerando DF2.

P3 (id\_categoria\_atractivo, nombre\_categoria)

P4 (id\_atractivo, id\_categoria\_atractivo, cuil\_empleado, nombre\_apellido\_empleado, nro\_legajo\_empleado, dia\_semana, id\_agente\_mantenimiento)

P3 ∩ P4 = clave en P3 {id\_categoria\_atractivo}. No se pierde información.

En P3 vale DF2.

En P4 vale DF3, DF4, DF5 y DF6.

P3 está en BCNF porque vale la DF2 tal que {id\_categoria\_atractivo} que es el determinante de la DF es superclave en P3.

- P4 no está en BCNF ya que existe al menos la DF3 tal que {cuil\_empleado} que es el determinante de la DF no es superclave en P4. Particiono para llevar a BCNF considerando DF3.

P5 (cuil\_empleado, nombre\_apellido\_empleado, nro\_legajo\_empleado)

P6 (id\_atractivo, id\_categoria\_atractivo, cuil\_empleado, dia\_semana, id\_agente\_mantenimiento)

P5 ∩ P6 = clave en P5 {cuil\_empleado}. No se pierde información.

En P5 vale DF3 y DF4.

En P6 vale DF5.

¿Qué pasa con DF6? No se pierde por el siguiente razonamiento:

- cuil\_empleado  $\rightarrow$  nombre\_apellido\_empleado, nro\_legajo\_empleado

- id\_atractivo, nro\_legajo\_empleado  $\rightarrow$  dia\_semana

Con DF3: cuil\_empleado  $\rightarrow$  nombre\_apellido\_empleado, nro\_legajo\_empleado

Entonces en P6 tengo cuil\_empleado, a partir del cual recupero

nro\_legajo\_empleado. Puedo decir por el razonamiento seguido que tengo una equivalencia entre DFs.

No se perdieron DFs por validación simple.

P5 está en BCNF porque vale la DF3 tal que {cuil\_empleado} que es el determinante de la DF es superclave en P5, y además vale la DF4 tal que {nro\_legajo\_empleado} que es el determinante de la DF es superclave en P5.

- P6 no está en BCNF ya que existe al menos la DF5 tal que {id\_atractivo, cuil\_empleado} que es el determinante de la DF no es superclave en P6.

Particiono para llevar a BCNF considerando DF5.

P7 (id\_atractivo, cuil\_empleado, dia\_semana)

P8 (id\_atractivo, id\_categoria\_atractivo, cuil\_empleado, id\_agente\_mantenimiento)

$P7 \cap P8 = \text{clave en } P7 \{id\_atractivo, cuil\_empleado\}$ . No se pierde información.

En P7 vale DF5.

No se perdieron DFs por validación simple.

P7 está en BCNF porque vale la DF5 tal que {id\_atractivo, cuil\_empleado} que es el determinante de la DF es superclave en P7. P8 está en BCNF porque cualquier dependencia funcional que valga es trivial.

#### Particiones en BCNF:

P1 (id\_atractivo, nombre\_atractivo, descripcion\_atractivo)

P3 (id\_categoria\_atractivo, nombre\_categoria)

P5 (cuil\_empleado, nombre\_apellido\_empleado, nro\_legajo\_empleado)

P7 (id\_atractivo, cuil\_empleado, dia\_semana)

P8 (id\_atractivo, id\_categoria\_atractivo, cuil\_empleado, id\_agente\_mantenimiento)

CP: {id\_atractivo, id\_categoria\_atractivo, cuil\_empleado, id\_agente\_mantenimiento}

#### Dependencias multivaluadas válidas en P8:

DM1) id\_atractivo  $\rightarrow\!\!>$  id\_categoria\_atractivo

DM2) id\_atractivo  $\rightarrow\!\!>$  cuil\_empleado

DM3) {}  $\rightarrow\!\!>$  id\_agente\_mantenimiento

- Dado que en P8 vale al menos la DM3 que no es trivial (una dependencia multivaluada  $X \Rightarrow Y$  en un esquema  $R$  es trivial cuando la unión de los atributos  $X \cup Y$  abarca todos los atributos de  $R$ ), P8 no cumple la definición de 4FN, entonces particiono P8 considerando DM3.

P9 (id\_agente\_mantenimiento)

P10 (id\_atractivo, id\_categoria\_atractivo, cuil\_empleado)

P9 está en 4FN porque vale DM3 tal que DM3 es trivial en P9.

- Dado que en P10 vale al menos la DM1 que no es trivial, P10 no cumple la definición de 4FN, entonces particiono P10 considerando DM1.

P11 (id\_atractivo, id\_categoria\_atractivo)

P12 (id\_atractivo, cuil\_empleado)

P11 está en 4FN porque vale DM1 tal que DM1 es trivial en P11.

P12 está en 4FN porque vale DM2 tal que DM2 es trivial en P12.

Esquemas resultantes en 4FN:

P1 (id\_atractivo, nombre\_atractivo, descripcion\_atractivo)

P3 (id\_categoria\_atractivo, nombre\_categoria)

P5 (cuil\_empleado, nombre\_apellido\_empleado, nro\_legajo\_empleado)

P7 (id\_atractivo, cuil\_empleado, dia\_semana)

P9 (id\_agente\_mantenimiento)

P11 (id\_atractivo, id\_categoria\_atractivo)

Se eliminó el esquema P12 por ser proyección de P7.

CP: {id\_atractivo, id\_categoria\_atractivo, cuil\_empleado,  
id\_agente\_mantenimiento}

## AR

MAYCAT  $\leftarrow \Pi_{\#mayorista, categoria} (PRECIO MAYORISTA | X | PRODUCTO)$

MAYCATVEND  $\leftarrow \Pi_{\#mayorista, categoria} (VENTA | X | PRODUCTO)$

CATSINVEND  $\leftarrow \Pi_{\#mayorista, categoria} (MAYCAT - MAYCATVEND)$

$\Pi_{\#mayorista} (MAYCAT) - \Pi_{\#mayorista} (CATSINVEND)$

## MySQL

```
DELIMITER //
```

```
CREATE TRIGGER `add_oferta_historial`
```

```
AFTER UPDATE ON oferta
```

```
FOR EACH ROW
```

```
BEGIN
    INSERT INTO oferta_historial(#oferente, #subasta, monto,
fecha_oferta, fecha_auditoria)
    VALUES (OLD.#oferente, OLD.#subasta, OLD.monto, OLD.fecha_oferta,
CURDATE());
END; //

DELIMITER ;
```

SF - 2024



**MySQL**

Se dispone de una base de datos que registra los envíos a usuarios por parte de una empresa de correos. De cada envío, se sabe el usuario que lo solicita además de la información concreta del envío. Además, la empresa dispone de un sistema de puntos para los usuarios, que luego permite otorgarle beneficios. La estructura de la base de datos es la siguiente:

**Usuario** (#usuario, nombre, email, telefono, puntaje)  
**Envio** (#envio, #usuario, direccion\_entrega, fecha\_solicitud, informacion\_destinatario, estado\_envio)  
**Beneficio** (#beneficio, descripcion, puntos\_requeridos)  
**Canje** (#canje, #usuario, #beneficio, fecha\_canje)

1) Generar un trigger que aumente en 5 puntos el puntaje del usuario luego de que este registre un nuevo envío.

**Visualización**

Se han almacenado datos de registro de la temperatura promedio del mar, a partir de mediciones diarias frente a la costa de Las Toninas, un kilómetro mar adentro. Esta medición se viene realizando durante los últimos 7 años para estudios del comportamiento de la fauna del lugar.  
Dado el siguiente esquema de la base de datos:

**TipoGradoTemp** (#tipoGradoTemp, descripcionTipoGrado)  
**TemperaturaRegistrada** (#registroTemp, fecha, hora, valorTemp, #tipoGradoTemp)  
**TemperaturasPromedio** (#registroProm, fecha, valorProm)

a) ¿Qué tipo de gráfico utilizará para mostrar los cambios de la temperatura promedio a lo largo del tiempo?  
b) ¿Cuáles tablas son relevantes para presentar el análisis?

Además, el dataset muestra la cantidad de especies que se identificaron en el mar en las 5 distintas categorías (Moluscos, Artrópodos, Cnidarios, Peces y Mamíferos marinos).  
Dado el siguiente esquema de la base de datos:

**CategoríaEspecie** (#cat\_especie, cat\_nombre )  
**EspecieIdentificada** (#especie, #cat\_especie, nombre\_especie, cantidad)

c) ¿Qué gráfico utilizaría si se quiere visualizar la proporción de especies de cada categoría?

**MySQL**

```

DELIMITER //

CREATE TRIGGER `update_puntaje_usuario`
AFTER INSERT ON envio
FOR EACH ROW
BEGIN
    UPDATE usuario
    SET puntaje = puntaje + 5
    WHERE #usuario = NEW.#usuario;
END; //

DELIMITER ;

```

**Visualización**

Se han almacenado datos de registro de la temperatura promedio del mar, a partir de mediciones diarias frente a la costa de Las Toninas, un kilómetro mar adentro. Esta medición se viene realizando durante los últimos 7 años para estudios del comportamiento de la fauna del lugar.

Dado el siguiente esquema de la base de datos:

**TipoGradoTemp** (#tipoGradoTemp, descripcionTipoGrado)  
**TemperaturaRegistrada** (#registroTemp, fecha, hora, valorTemp, #tipoGradoTemp)  
**TemperaturasPromedio** (#registroProm, fecha, valorProm)

a) ¿Qué tipo de gráfico utilizará para mostrar los cambios de la temperatura promedio a lo largo del tiempo?

Utilizaría el gráfico de línea ya que este muestra las relaciones de los cambios en los datos en un período de tiempo, facilitando la identificación de tendencias. Se podría ver el aumento, disminución o estabilidad de las temperaturas a lo largo de los años. En el eje X se colocará la fecha y en el eje Y el valor promedio de temperatura en cada fecha.

b) ¿Cuáles tablas son relevantes para presentar el análisis?

Para el análisis es relevante la tabla TemperaturasPromedio ya que con la misma obtenemos la fecha y el valor promedio de temperatura en dicha fecha.

Además, el dataset muestra la cantidad de especies que se identificaron en el mar en las 5 distintas categorías (Moluscos, Artrópodos, Cnidarios, Peces y Mamíferos marinos).

Dado el siguiente esquema de la base de datos:

**CategoríaEspecie (#cat\_especie, cat\_nombre )**

**EspecieIdentificada (#especie, #cat\_especie, nombre\_especie, cantidad)**

- c) ¿Qué gráfico utilizaría si se quiere visualizar la proporción de especies de cada categoría?

Utilizaría un gráfico circular (torta) ya que se podría mostrar cómo esas cinco categorías se comparan en porcentaje entre ellas y el total. Cómo son cinco las categorías se entendería el gráfico.

- d) De todas las tablas propuestas, indicar cuál o cuáles son relevantes para presentar el análisis

Las dos tablas (CategoríaEspecie y EspecieIdentificada) son relevantes para el análisis, ya que la primera identifica las categorías y la segunda proporciona los datos numéricos necesarios para representar la proporción en el gráfico circular.

**Primer Fecha - 15/11/2023****BBDD1 Parcial**

1ra fecha 15/11/2023

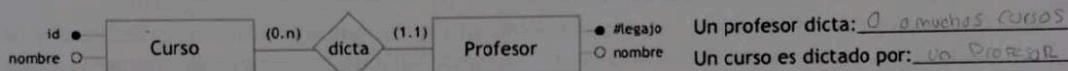
Uso interno de la cátedra

ER	Norm	AR	MySQL

Ayudante: \_\_\_\_\_

**Entidad/Relación****Indique cómo interpreta las cardinalidades del siguiente modelo**

(esto se usará para leer las cardinalidades de la solución que usted proponga)

**Enunciado:**

Se requiere de un sistema para la elección de proyectos con presupuesto participativo para un municipio. Cada proyecto tiene un nombre, descripción, una fecha de creación y un responsable (que puede ser una persona física o bien, jurídica).

De las personas físicas se conoce el nombre, dni, domicilio, CUIL y teléfono. De las personas jurídicas, se conoce su denominación, CUIT, domicilio legal y teléfono de contacto.

Cada proyecto está compuesto por un conjunto de actividades. Cada actividad tiene un nombre y un conjunto de parámetros. Algunos de estos parámetros suelen repetirse para todas las actividades, como por ejemplo, el "costo" en pesos y el "plazo de ejecución en días".

De los parámetros de las actividades se conoce un nombre y una descripción.

Una actividad puede tener diversos parámetros y un parámetro puede estar en varias actividades.

A modo de ejemplo, supongamos que "Juana Laprida" crea un proyecto participativo para la "colocación de reductores de velocidad" en la esquina de San Juan y Laprida, de su ciudad. Para este proyecto, Juana define 2 actividades, por un lado, la colocación de los reductores junto con la señalización de los mismos (a esta actividad le define dos parámetros: el "costo" y el "plazo de ejecución en días"). Decide cargar el "costo" con \$5.000.000 y el "plazo de ejecución en días" con 10 y por otro lado, la actividad de colocación de cartelería sobre la concientización de las velocidades máximas permitidas en calle/avenida, con los parámetros, "costo" cargado en \$2.300.000 y "plazo de ejecución en días" cargado con el número 7.

Notar que cuando se carga un proyecto, para cada parámetro de cada actividad, se sabe qué valor toma.

Otro aspecto a considerar es la votación de los proyectos cargados.

La votación se lleva a cabo durante una semana en la que los ciudadanos pueden acercarse a la sede barrial y votar en las pccs dispuestas allí (voto presencial) o bien, votar desde su casa por medio de su dispositivo móvil (voto remoto).

Un ciudadano, emite un voto por cada parámetro de cada actividad de un proyecto. Este voto por parámetro, puede ser positivo o negativo. De cada voto, se registra qué ciudadano lo emite y si vota de manera presencial o remota. En caso de votar de manera remota, se registra el número de ip y la ubicación desde donde fue realizado el voto. Si el voto se realiza de manera presencial en la sede barrial (desde una pc), se solicita el dni del ciudadano y se registra su dirección.

Al finalizar la votación de un proyecto, cada parámetro de una actividad deberá contar con un resultado de la elección.

**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 de una inmobiliaria con sucursales, donde se registran pagos de alquileres de departamentos.

PAGOS (#empleado, dni, nombre, fecha\_ingreso, #sucursal, ciudad, telefono, #departamento, #pago, monto\_pago, fecha\_pago, #honorario, descripcion\_h, monto\_h)

- La inmobiliaria dispone de varias sucursales, identificada por #sucursal, de las cuales se conoce la ciudad donde está ubicada y un teléfono de contacto *SUCURSAL (#sucursal, Ciudad, telefono)*
- De cada empleado, que trabaja únicamente en una sucursal, se conoce su número interno (no se repite para diferentes empleados de la inmobiliaria), dni, nombre y fecha de ingreso.
- Los #departamento identifican un departamento en alquiler que administra la inmobiliaria y no puede repetirse en las diferentes sucursales. La inmobiliaria asigna a cada departamento varios empleados para que los administren.
- Los #pagos son secuenciales para cada departamento que posee la inmobiliaria (no pueden repetirse para el mismo departamento) y se almacena el monto y la fecha de dicho pago. No se registra la sucursal donde se realizó el pago.
- La inmobiliaria registra todos los honorarios que percibe, de estos se conoce su #honorario (único en el sistema), el monto y una descripción.

**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:

USUARIO (dni, email, nombre)  
 ACTIVIDAD (#actividad, # proyecto, nombre, detalle\_actividad, condiciones\_actividad)  
 PROYECTO (# proyecto, nombre, presupuesto, cantidad\_días)  
 COMENTARIO (#comentario, dni, # actividad, valoración, detalle\_comentario)

Hallar el # proyecto y nombre de los proyectos en los que todas sus actividades tienen al menos un comentario.

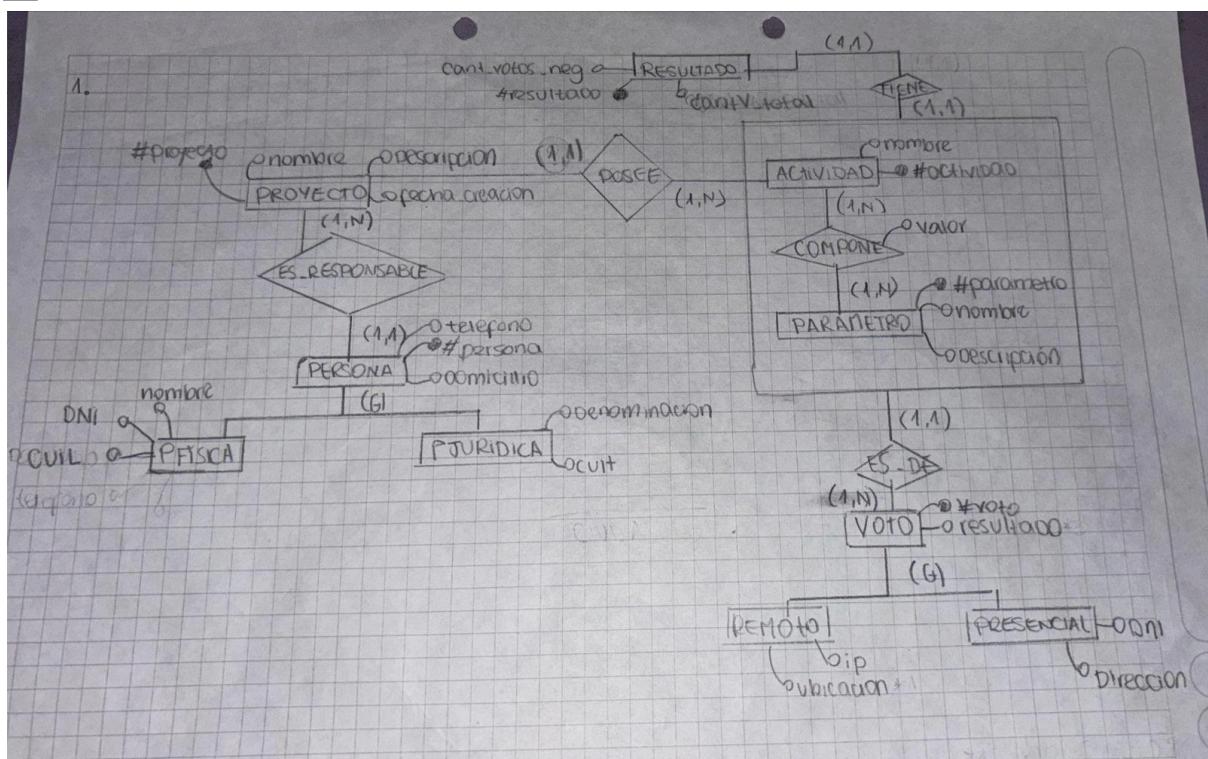
**MySQL**

1. Se modificó el esquema anterior, presentado en el ejercicio de Álgebra Relacional, para agregar a la tabla ACTIVIDAD, una nueva columna llamada "cantidad\_comentarios", resultando:

ACTIVIDAD (#actividad, # proyecto, nombre, detalle\_actividad, condiciones\_actividad, cantidad\_comentarios)

Escriba un Trigger para que con cada nuevo comentario registrado se actualice la nueva columna de la actividad (cantidad\_comentarios).

2. Enuncie y explique dos motivos por los cuales necesitaría definir una vista.

**ER**

PERSONA (#persona, telefono, domicilio)
PJURIDICA (#persona, CUIT, denominacion)
PFISICA (#persona, DNI, CUIL, nombre)
PROYECTO (#proyecto, nombre, descripcion, fecha_creacion)
ES_RESPONSABLE (#proyecto, #persona)
ACTIVIDAD (#actividad, nombre)
POSEE (#actividad, #proyecto)
PARAMETRO (#parametro, nombre, descripcion)
COMPONE (#actividad, #parametro, valor)
VOTO (#voto, resultado)
REMOTO (#voto, ip, ubicacion)
PRESENCIAL (#voto, dni, direccion)
ES_DE (#voto, #actividad, #parametro)
RESULTADO (#resultado, cant_votos_neg, cantv_total)
TIENE (#resultado, #actividad, #parametro)

### Normalización

DF1: #sucursal -> ciudad, telefono

DF2: #empleado -> dni, nombre, fecha\_ingreso, #sucursal

DF3: dni -> #empleado, nombre, fecha\_ingreso, #sucursal

DF4: #departamento, #pago -> monto\_pago, fecha\_pago

DF5: #honorario -> descripcion\_h, monto\_h

CC1: {#empleado, #departamento, #pago, #honorario}

CC2: {dni, #departamento, #pago, #honorario}

- PAGOS no está en BCNF ya que existe al menos la DF5 tal que  $\{\#honorario\}$  que es el determinante de la DF no es superclave en PAGOS. Particiono para llevar a BCNF considerando DF5.

P1 (#honorario, descripcion\_h, monto\_h)

P2 (#empleado, dni, nombre, fecha\_ingreso, #sucursal, ciudad, telefono,

#departamento, #pago, monto\_pago, fecha\_pago, #honorario)

$P1 \cap P2 = \text{clave en } P1 \{\#honorario\}$ . No se pierde información.

En P1 vale DF5.

En P2 vale DF1, DF2, DF3 y DF4.

No se perdieron DFs por validación simple.

P1 está en BCNF porque vale la DF5 tal que  $\{\#honorario\}$  que es el determinante de la DF es superclave en P1.

- P2 no está en BCNF ya que existe al menos la DF4 tal que  $\{\#departamento, \#pago\}$  que es el determinante de la DF no es superclave en P2. Particiono para llevar a BCNF considerando DF4.

P3 (#departamento, #pago, monto\_pago, fecha\_pago)

P4 (#empleado, dni, nombre, fecha\_ingreso, #sucursal, ciudad, telefono,

#departamento, #pago, #honorario)

$P3 \cap P4 = \text{clave en } P3 \{\#departamento, \#pago\}$ . No se pierde información.

En P3 vale DF4.

En P4 vale DF1, DF2 y DF3.

No se perdieron DFs por validación simple.

P3 está en BCNF porque vale la DF4 tal que  $\{\#departamento, \#pago\}$  que es el determinante de la DF es superclave en P3.

- P4 no está en BCNF ya que existe al menos la DF1 tal que  $\{\#sucursal\}$  que es el determinante de la DF no es superclave en P4. Particiono para llevar a BCNF considerando DF1.

P5 (#sucursal, ciudad, telefono)

P6 (#empleado, dni, nombre, fecha\_ingreso, #sucursal, #departamento, #pago, #honorario)

$P5 \cap P6 = \text{clave en } P5 \{\#sucursal\}$ . No se pierde información.

En P5 vale DF1.

En P6 vale DF2 y DF3.

No se perdieron DFs por validación simple.

P5 está en BCNF porque vale la DF1 tal que  $\{\#sucursal\}$  que es el determinante de la DF es superclave en P5.

- P6 no está en BCNF ya que existe al menos la DF2 tal que  $\{\#empleado\}$  que es el determinante de la DF no es superclave en P6. Particiono para llevar a BCNF considerando DF2.

P7 (#empleado, dni, nombre, fecha\_ingreso, #sucursal)

P8 (#empleado, #departamento, #pago, #honorario)

P7  $\cap$  P8 = clave en P7 {#empleado}. No se pierde información.

En P7 vale DF1 y DF2.

No se perdieron DFs por validación simple.

En P7 está en BCNF porque vale la DF2 tal que {#empleado} que es el determinante de la DF es superclave en P7, y además vale la DF3 tal que {dni} que es el determinante de la DF es superclave en P7. P8 está en BCNF porque cualquier dependencia funcional que valga es trivial.

#### Particiones en BCNF:

P1 (#honorario, descripcion\_h, monto\_h)

P3 (#departamento, #pago, monto\_pago, fecha\_pago)

P5 (#sucursal, ciudad, telefono)

P7 (#empleado, dni, nombre, fecha\_ingreso, #sucursal)

P8 (#empleado, #departamento, #pago, #honorario)

CP: {#empleado, #departamento, #pago, #honorario}

#### Dependencias multivaluadas válidas en P8:

DM1) #departamento  $\rightarrow\!\!\!>$  #empleado  $\rightarrow$  La inmobiliaria asigna a cada departamento varios empleados

DM2) #departamento  $\rightarrow\!\!\!>$  #pago:  $\rightarrow$  Para un mismo departamento tengo una lista de pagos, que es independiente del empleado asignado

DM3) {}  $\rightarrow\!\!\!>$  #honorario

- Dado que en P8 vale al menos la DM3 que no es trivial (una dependencia multivaluada  $X \twoheadrightarrow Y$  en un esquema  $R$  es trivial cuando la unión de los atributos  $X$   $\cup$   $Y$  abarca todos los atributos de  $R$ ), P8 no cumple la definición de 4FN, entonces particiono P8 considerando DM3.

P9 (#honorario)

P10 (#empleado, #departamento, #pago)

P9 está en 4FN porque vale DM3 tal que DM3 es trivial en P9.

- Dado que en P10 vale al menos la DM1 que no es trivial, P10 no cumple la definición de 4FN, entonces particiono P10 considerando DM1.

P11 (#departamento, #empleado)

P12 (#departamento, #pago)

P11 está en 4FN porque vale DM1 tal que DM1 es trivial en P11.

P12 está en 4FN porque vale DM2 tal que DM2 es trivial en P12.

#### Esquemas resultantes en 4FN:

P1 (#honorario, descripción\_h, monto\_h)  
P3 (#departamento, #pago, monto\_pago, fecha\_pago)  
P5 (#sucursal, ciudad, teléfono)  
P7 (#empleado, dni, nombre, fecha\_ingreso, #sucursal)  
P11 (#departamento, #empleado)

Se eliminaron los esquemas P9 y P12. P9 es proyección de P1 y P12 de P3.

CP: {#empleado, #departamento, #pago, #honorario}

## **AR**

ACTCONCOM  $\leftarrow \Pi_{\#actividad} (COMENTARIO)$   
ACTSINCOM  $\leftarrow \Pi_{\#actividad} (ACTIVIDAD) - ACTCONCOM$   
PROYSINCOM  $\leftarrow \Pi_{\#proyecto} (ACTSINCOM | X | ACTIVIDAD)$   
 $\Pi_{\#proyecto, nombre} (PROYECTO) - \Pi_{\#proyecto, nombre} (PROYECTO | X | PROYSINCOM)$

## **MySQL**

1.

```
DELIMITER //

CREATE TRIGGER `update_actividad_comentarios`
AFTER INSERT ON comentario
FOR EACH ROW
BEGIN
    UPDATE actividad
    SET cantidad_comentarios = cantidad_comentarios + 1;
    WHERE #actividad = NEW.#actividad;
END; //

DELIMITER ;
```

2. - Abstracción y simplificación de consultas complejas: Usar una vista para ocultar la complejidad de consultas frecuentes (joins, agregaciones, filtros) y presentar una tabla virtual fácil de usar.

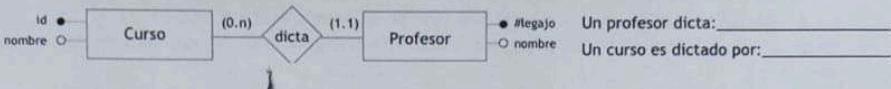
- Seguridad / control de acceso a los datos: Usar una vista para exponer solo las columnas o filas permitidas a ciertos usuarios, restringiendo el acceso a datos sensibles.

**Segunda Fecha - 29/11/2023**

Nº de Alumno: _____	Apellido y Nombre: _____								
<b>BBDD1 Parcial</b> 2da fecha 29/11/2023									
<b>Uso interno de la cátedra</b> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>ER</th> <th>Norm</th> <th>AR</th> <th>MySQL</th> </tr> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </table>		ER	Norm	AR	MySQL				
ER	Norm	AR	MySQL						
Ayudante: _____									

**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:**

Se requiere una aplicación para manejar las reservas de las canchas de voley de un complejo recientemente inaugurado.

El complejo cuenta con varias canchas de las cuales se conoce el número y una descripción. De las reservas sobre estas canchas se conoce la fecha y la hora, registrándose además el usuario de la aplicación que la realizó y los dos equipos que participarán del partido. De los equipos se conoce un nombre, fecha de alta y los jugadores que conforman el equipo. Notar que un usuario puede reservar la misma cancha en diferentes fechas.

Por cada equipo participante de una reserva, se registra su resultado (si ganó o perdió). Tener en cuenta que en toda reserva se realiza un único partido.

Todos los usuarios deben registrarse con su dni, nombre, edad, correo electrónico y contraseña. Todos los jugadores son usuarios del sistema, de los que además se registra su grupo sanguíneo. Como la reserva solo puede realizarse por mayores de edad, tener en cuenta que quien hace la reserva no necesariamente es un jugador.

De los jugadores además se guarda una categoría. Esta categoría puede ir cambiando a lo largo del tiempo, por ello se quiere guardar un historial de fechas en que el jugador perteneció a un categoría específica. Tener en cuenta que podría volver a una categoría que haya poseido con anterioridad.

El pago por el uso de la cancha se realiza después del partido y la aplicación permite que cada equipo registrado realice el pago de su parte independientemente del otro. Del pago se debe guardar la fecha, el monto final (por equipo) y el medio de pago utilizado: tarjeta o efectivo. Si el pago es mediante tarjeta se debe registrar si fue mediante crédito o débito; si el pago es en efectivo se registra un descuento.

**Actividades:**

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

**Normalización**

<b>Algoritmo para analizar la pérdida de dfs.</b> Res = x Mientras Res cambia Para i= 1 to cant_de_particiones_realizadas Res = Res U((Res ∩ Ri)* ∩ Ri)	<b>Algoritmo para encontrar X<sup>+</sup></b> Result:= X While (hay cambios en result) do For (cada dependencia funcional Y->Z en F) do if (Y ⊆ result) then result := result U Z
---	---

Se dispone del siguiente esquema que representa los resultados de diferentes torneos de fútbol.

TORNEOS (#torneo, nombre\_torneo, año, #equipo, nombre\_equipo, estadio\_equipo, puesto, #reglamentacion, descripción, #auspiciante)

- De cada torneo, se conoce su identificador (#torneo, único en el sistema) y un nombre. Un mismo torneo tiene diferentes ediciones, cada edición se realiza en un año determinado y el mismo torneo no puede repetirse el mismo año. En un año pueden realizarse varios torneos.
- Cada edición de un torneo tiene diferentes auspiciantes, identificados por #auspiciante (único en el sistema).
- En cada edición de un torneo participan varios equipos. De cada equipo se conoce su nombre, su estadio y su #equipo, que no se repite para diferentes equipos.
- Cada equipo finaliza una edición de un torneo en un puesto. Dos o más equipos no pueden finalizar en un mismo puesto.
- Además, se conoce un conjunto de reglamentaciones, identificadas por #reglamentación, aplicables a estos torneos.

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:

- EMPLEADO (#empleado, email, nombre, años\_antiguedad)
- PREMIO (#premio, descripción\_premio)
- RECIBE\_PREMIO (#premio, #empleado, fecha\_premio)
- SANCION (#sancion, descripción\_sancion)
- RECIBE\_SANCION (#sancion, #empleado, fecha\_sancion, gravedad)

Hallar el #empleado, nombre y email de los empleados con más 10 años de antigüedad que hayan recibido todos los premios y que además no hayan recibido sanciones de ningún tipo.

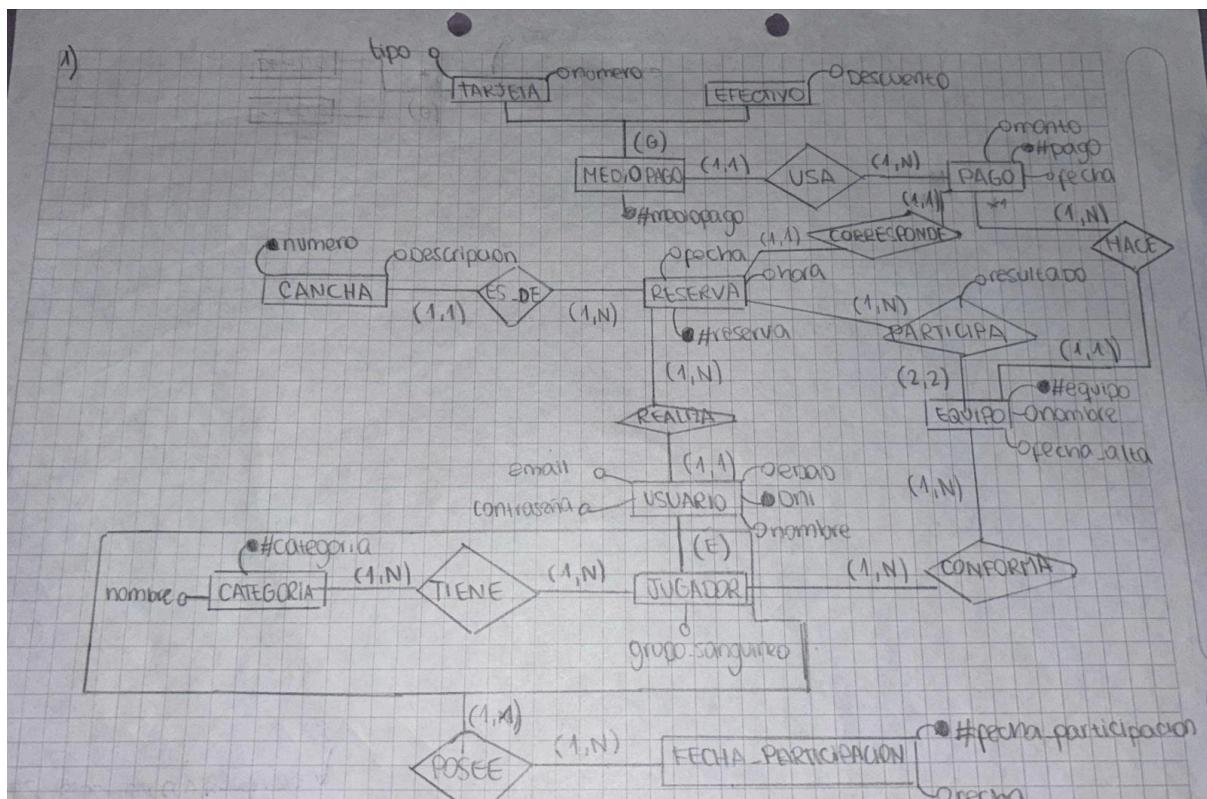
**MySQL**

Se modificó el esquema anterior, presentado en el ejercicio de Álgebra Relacional, para agregar la siguiente tabla:

AUDITORÍA (#auditoría, #empleado, #sanción, prioridad)

Escriba un trigger para que con cada nueva sanción registrada a un empleado, si esta es de gravedad "Alta" se agregue una nueva tupla a la tabla AUDITORIA donde se registre la correspondiente información con la prioridad "Inmediata".

ER



CANCHAS ( <u>número</u> , descripción)
RESERVA (#reserva, fecha, hora)
ES_DEF (#reserva, número)
USUARIO (DNI, email, contraseña, edad, nombre)
REALIZA (#reserva, DNI)
JUGADOR (DNI, grupo-sanguíneo)
CATEGORIA (#categoria, nombre)
TIENE (DNI, #categoria)
FECHA PARTICIPACION (#fecha-participacion, fecha)
ROSEE (#categoria, DNI, #fecha-participacion)
EQUIPO (#equipo, nombre, fecha-alta)
CONFIRMA (DNI, #equipo)
PARTICIPA (#reserva, #equipo, resultado)
PAGO (#pago, monto, fecha)
HACE (#pago, #equipo)
CORRESPONDE (#reserva, #pago)
MEDIO_PAGO (#medio_pago)
TARJETA (#medio_pago, tipo, número)
EFFECTIVO (#medio_pago, descuento)
USA (#pago, #medio_pago)

### Normalización

DF1: #torneo → nombre\_torneo

DF2: #equipo → nombre\_equipo, estadio\_equipo

DF3: #reglamentacion → descripcion

DF4: #torneo, año, #equipo -> puesto

DF5: #torneo, año, puesto -> #equipo

CC1: {#torneo, año, #equipo, #reglamentacion, #auspiciante}

CC2: {#torneo, año, puesto, #reglamentacion, #auspiciante}

- TORNEOS no está en BCNF ya que existe al menos la DF1 tal que {#torneo} que es el determinante de la DF no es superclave en TORNEOS. Particiono para llevar a BCNF considerando DF1.

T1 (#torneo, nombre\_torneo)

T2 (#torneo, año, #equipo, nombre\_equipo, estadio\_equipo, puesto, #reglamentacion, descripcion, #auspiciante)

T1 ∩ T2 = clave en T1 {#torneo}. No se pierde información.

En T1 vale DF1.

En T2 vale la DF2, DF3, DF4 y DF5.

No se perdieron DFs por validación simple.

T1 está en BCNF porque vale la DF1 tal que {#torneo} que es el determinante de la DF es superclave en T1.

- T2 no está en BCNF ya que existe al menos la DF2 tal que {#equipo} que es el determinante de la DF no es superclave en T2. Particiono para llevar a BCNF considerando DF2.

T3 (#equipo, nombre\_equipo, estadio\_equipo)

T4 (#torneo, año, #equipo, puesto, #reglamentacion, descripcion, #auspiciante)

T3 ∩ T4 = clave en T3 {#equipo}. No se pierde información.

En T3 vale DF2.

En T4 vale la DF3, la DF4 y DF5.

No se perdieron DFs por validación simple.

T3 está en BCNF porque vale la DF2 tal que {#equipo} que es el determinante de la DF es superclave en T3.

- T4 no está en BCNF ya que existe al menos la DF3 tal que {#reglamentacion} que es el determinante de la DF no es superclave en T4. Particiono para llevar a BCNF considerando DF3.

T5 (#reglamentacion, descripcion)

T6 (#torneo, año, #equipo, puesto, #reglamentacion, #auspiciante)

T5 ∩ T6 = clave en T5 {#reglamentacion}. No se pierde información.

En T5 vale la DF3.

En T6 vale la DF4 y DF5.

No se perdieron DFs por validación simple.

T5 está en BCNF porque vale la DF3 tal que {#reglamentacion} que es el determinante de la DF es superclave en T5.

- T6 no está en BCNF ya que existe al menos la DF4 tal que  $\{\#torneo, año, \#equipo\}$  que es el determinante de la DF no es superclave en T6. Particiono para llevar a BCNF considerando DF4.

T7 (#torneo, año, #equipo, puesto)

T8 (#torneo, año, #equipo, #reglamentacion, #auspiciante)

$T7 \cap T8 = \text{clave en } T7 \{\#torneo, año, \#equipo\}$ . No se pierde información.

En T7 vale la DF4 y DF5.

No se perdieron DFs por validación simple.

T7 está en BCNF porque vale la DF4 tal que  $\{\#torneo, año, \#equipo\}$  que es el determinante de la DF es superclave en T7, y además vale la DF5 tal que  $\{\#torneo, año, \text{puesto}\}$  que es el determinante de la DF es superclave en T7. T8 está en BCNF porque cualquier dependencia funcional que valga es trivial.

#### Particiones en BCNF:

T1 (#torneo, nombre\_torneo)

T3 (#equipo, nombre\_equipo, estadio\_equipo)

T5 (#reglamentacion, descripcion)

T7 (#torneo, año, #equipo, puesto)

T8 (#torneo, año, #equipo, #reglamentacion, #auspiciante)

CP:  $\{\#torneo, año, \#equipo, \#reglamentacion, \#auspiciante\}$

#### Dependencias multivaluadas válidas en T8:

DM1)  $\#torneo \rightarrow\!\!\!> \#equipo$

DM2)  $\#torneo \rightarrow\!\!\!> \#auspiciante$

DM3)  $\#torneo \rightarrow\!\!\!> año$

DM4)  $\{\} \rightarrow\!\!\!> \#reglamentacion$

- Dado que en T8 vale al menos DM4 que no es trivial (una dependencia multivaluada  $X \rightarrow Y$  en un esquema  $R$  es trivial cuando la unión de los atributos  $X \cup Y$  abarca todos los atributos de  $R$ ), T8 no cumple la definición de 4FN, entonces particiono T8 considerando la DM4.

T9 (#reglamentacion)

T10 (#torneo, año, #equipo, #auspiciante)

T9 está en 4FN porque vale DM4 tal que DM4 es trivial en T9.

- Dado que en T10 vale al menos DM3 que no es trivial, T10 no cumple la definición de 4FN, entonces particiono T10 considerando la DM3.

T11 (#torneo, año)

T12 (#torneo, #equipo, #auspiciante)

T11 está en 4FN porque vale DM3 tal que DM3 es trivial en T11.

- Dado que en T12 vale al menos DM2 que no es trivial, T12 no cumple la definición de 4FN, entonces particono T12 considerando la DM2.

T13 (#torneo, #auspiciante)

T14 (#torneo, #equipo)

T13 está en 4FN porque vale DM2 tal que DM2 es trivial en T13.

T14 está en 4FN porque vale DM1 tal que DM1 es trivial en T14.

Esquemas resultantes en 4FN:

T1 (#torneo, nombre\_torneo)

T3 (#equipo, nombre\_equipo, estadio\_equipo)

T5 (#reglamentacion, descripcion)

T7 (#torneo, año, #equipo, puesto)

T13 (#torneo, #auspiciante)

Se eliminaron los esquemas T9, T11, T14 por ser proyecciones. T9 es proyección de T5, T11 de T7 y T14 de T7.

CP: {#torneo, año, #equipo, #reglamentacion, #auspiciante}

## AR

EMPCONSACION  $\leftarrow \Pi_{\#empleado}(\text{EMPLEADO } |X| \text{ RECIBESANCION})$

EMPCUMPLE  $\leftarrow (\Pi_{\#empleado}(\sigma_{\text{años_antiguedad} > 10}(\text{EMPLEADO}))) - \text{EMPCONSACION}$

EMPTODOSPREMIOS $\leftarrow$

$(\Pi_{\#empleado, \#premio}(\text{EMPLEADO } |X| \text{ RECIBE_PREMIO})) \% (\Pi_{\#premio}(\text{PREMIO}))$

$\Pi_{\#empleado, email, nombre}(\text{EMPCUMPLE } |X| \text{ EMPTODOSPREMIOS } |X| \text{ EMPLEADO})$

## MySQL

```
DELIMITER //
CREATE TRIGGER `add_auditoria`
AFTER INSERT ON recibe_sancion
FOR EACH ROW
BEGIN
    IF NEW.gravedad = 'Alta' THEN
        BEGIN
            INSERT INTO auditoria (#empleado, #sanción, prioridad)
            VALUES (NEW.#empleado, NEW.#sancion, 'Inmediata');
        END IF;
    END;
DELIMITER ;
```

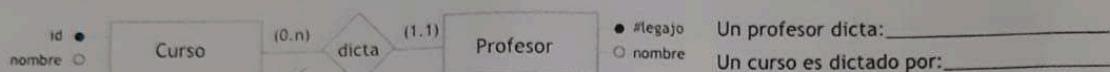
**Tercera Fecha - 13/12/2023**

Nº de Alumno: _____	Apellido y Nombre: _____		
Uso interno de la cátedra			
ER	Norm	AR	MySQL
Ayudante: _____			

BBDD1 Parcial  
3era fecha 13/12/2023

**Entidad/Relación**

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

**Enunciado:**

Se desea modelar un sistema para un estudio de grabación y su registro de músicos, grupos musicales y temas musicales.

Sobre los músicos se tiene información de su dni, nombre y apellido, año de nacimiento y qué instrumentos toca (al menos toca un instrumento). De una banda (o grupo) de música se conoce su nombre, género y los integrantes que la componen. Además de los músicos, habrá cantantes en muchos de los grupos (no en todos). De estos se sabe, además de sus datos personales, el tipo de voz que tiene (alto, bajo, medio). Pueden pertenecer, tanto músicos como cantantes a más de un grupo musical. De cada instrumento de música se conoce su nombre. Se requiere distinguir tres tipos de instrumento: de cuerdas (de los cuales se conoce su cantidad de cuerdas), de viento (de los cuales se conoce su clasificación, que puede ser "metal" o "madera") y de percusión (de los cuales se conoce su altura, que puede ser "definida" o "indefinida").

Si bien un músico puede tocar varios instrumentos de forma particular, un músico en una banda determinada sólo toca un instrumento. Sin embargo el mismo instrumento, dentro de una banda, puede ser tocado por más de un músico (por ejemplo dos guitarristas).

Se sabe que cada banda hace varios temas musicales. Un tema musical puede ser interpretado por varias bandas (o grupos) musicales. Cuando un tema queda bien pulido, la banda solicita una grabación del tema y el estudio almacena la misma con la fecha y la hora y el técnico que participó de la grabación. Si la banda lo considera necesario, podría volver a grabar un mismo tema.

**Actividades:**

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

**Normalización**

Algoritmo para analizar la pérdida de dfs.	Algoritmo para encontrar $X^+$
<pre>Res = x Mientras Res cambia   Para i= 1 to cant_de_particiones_realizadas     Res = Res U((Res ∩ Ri)* ∩ Ri)</pre>	<pre>Result:= X While (hay cambios en result) do   For (cada dependencia funcional Y-&gt;Z en F ) do     if (Y ⊑ result) then result := result U Z</pre>

Se dispone del siguiente esquema que representa los préstamos solicitados por clientes de un banco.

PRESTAMOS (dni, nombre, fecha\_nacimiento, #opción\_prestamo, título, condiciones, #prestamo, fecha\_prestamo, monto\_total, #cuota, monto\_cuota, fecha\_cuota, #condición, #plInternac)

- De cada cliente del banco que registra algún préstamo se conoce su dni, nombre y fecha de nacimiento.
- El banco dispone de un conjunto de opciones de préstamo que los clientes pueden optar (cada cliente tiene diversas opciones asociadas) identificadas por #opción\_prestamo (no se repite) y de las cuales se conoce un título y una descripción de condiciones
- Los #prestamo, identifican un préstamo. Este identificador es secuencial por cada cliente, es decir, por ejemplo, todos los primeros préstamos de cada cliente tendrán como #prestamo el 1. De estos se registra la fecha de solicitud, el monto total y la opción de préstamo elegida.
- Además, por cada préstamo de un cliente se registran una serie de condiciones particulares de ese préstamo (identificadas por #condición).
- Cada préstamo de un cliente se abona en varias cuotas. De cada cuota abonada se registra el número de cuota (#cuota), el monto y la fecha en que se abonó.
- #plInternac son los códigos de las entidades internacionales prestatarias con las que se relaciona el banco y que hace posible, entre otras cosas, efectivizar los préstamos.

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

## Álgebra Relacional

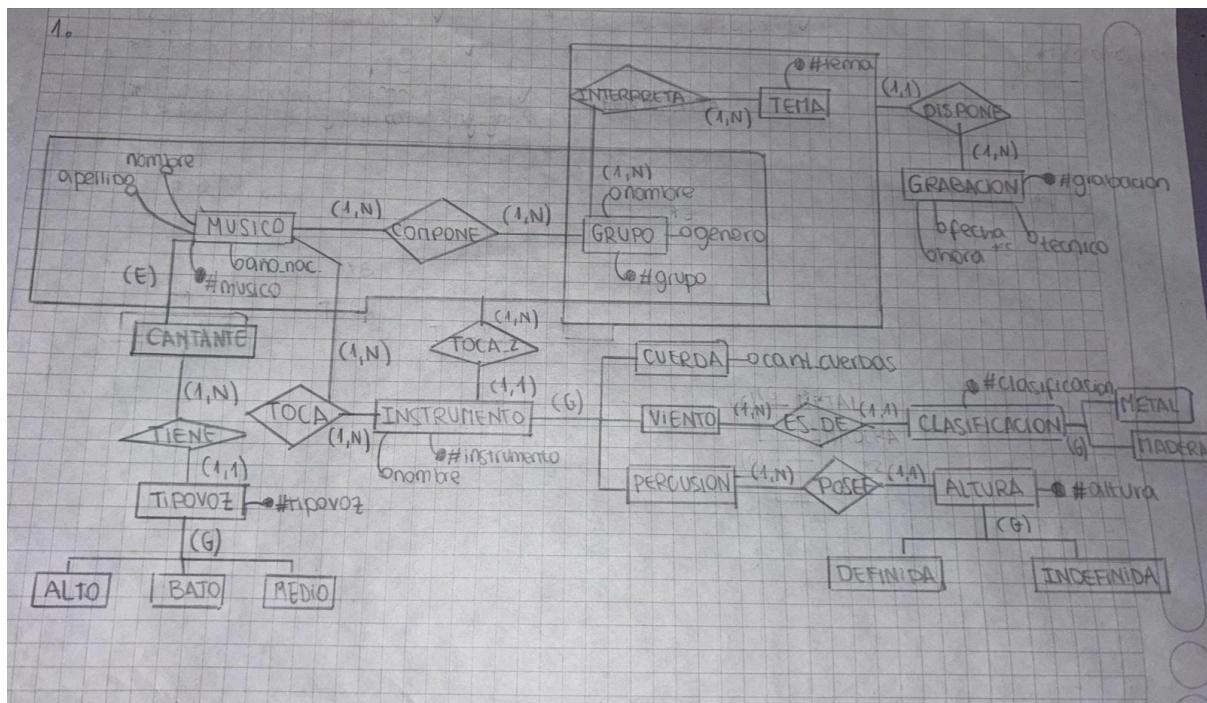
Dado el siguiente esquema:

DUEÑO (id\_dueño, nombre, teléfono, dirección, dni)  
CHOFER (id\_chofer, nombre, teléfono, dirección, fecha\_licencia\_desde, fecha\_licencia\_hasta, dni)  
AUTO (patente, id\_dueño, id\_chofer, marca, modelo, año)

**Hallar el DNI, nombre y teléfono de todos los dueños -que no sean choferes- y que solamente poseen vehículos marca "FIAT".**

## MySQL

Utilizando el esquema presentado en el ejercicio de Álgebra Relacional, escriba un Store Procedure en MySQL llamado "nuevo\_chofer". Este debe declarar como parámetro un dni y dos fechas (licencia\_desde y licencia\_hasta), y debe generar, dentro de una transacción, un nuevo chofer con los datos recuperados del dueño con dni igual al recibido como parámetro. Asuma que este dni ya existe en DUEÑO (esquema del ej .de AR).



MUSICO (#musico, año.nac, nombre, apellido)  
GRUPO (#grupo, nombre, genero)  
COMPONE (#musico, #grupo)  
CANTANTE (#musico)  
TIPOVOZ (#tipovoz, tipovoz)  
TIENE (#musico, #tipovoz)  
INSTRUMENTO (#instrumento, nombre)  
TOCA (#musico, #instrumento)  
TOCA\_2 (#musico, #grupo, #instrumento)  
CUERDA (#instrumento, cant\_cuerdas)  
VIENTO (#instrumento)  
PERCUSION (#instrumento)  
ES\_DE (#instrumento, #clasificacion)  
POSEE (#instrumento, #altura)  
CLASIFICACION (#clasificacion, +poclasificacion)  
ALTURA (#altura, tipoAltura)  
TEMA (#tema)  
INTERPRETA (#grupo, #tema)  
GRABACION (#grabacion, tecnico, fecha, hora)  
DISPONE (#grabacion, #grupo, #tema)

### Normalización

DF1: dni → nombre, fecha\_nacimiento

DF2: #opción\_prestamo → titulo, condiciones

DF3: dni, #prestamo → fecha\_prestamo, monto\_total, #opción\_prestamo

DF4: dni, #prestamo, #cuota → monto\_cuota, fecha\_cuota

CC: {dni, #prestamo, #cuota, #condicion, #plInternac}

- PRESTAMOS no está en BCNF ya que existe al menos la DF1 tal que {dni} que es el determinante de la DF no es superclave en PRESTAMOS. Particione para llevar a BCNF considerando DF1.

P1 (dni, nombre, fecha\_nacimiento)

P2 (dni, #opción\_prestamo, titulo, condiciones, #prestamo, fecha\_prestamo, monto\_total, #cuota, monto\_cuota, fecha\_cuota, #condición, #plInternac)

P1 ∩ P2 = clave en P1 {dni}. No se pierde información.

En P1 vale DF1.

En P2 vale DF2, DF3 y DF4.

No se perdieron DFs por validación simple.

P1 está en BCNF porque vale la DF1 tal que {dni} que es el determinante de la DF es superclave en P1.

- P2 no está en BCNF ya que existe al menos la DF2 tal que {#opción\_prestamo} que es el determinante de la DF no es superclave en P2. Particione para llevar a BCNF considerando DF2.

P3 (#opción\_prestamo, titulo, condiciones)

P4 (dni, #opción\_prestamo, #prestamo, fecha\_prestamo, monto\_total, #cuota, monto\_cuota, fecha\_cuota, #condición, #plInternac)

P3 ∩ P4 = clave en P3 {#opción\_prestamo}. No se pierde información.

En P3 vale DF2.

En P4 vale DF3 y DF4.

No se perdieron DFs por validación simple.

P3 está en BCNF porque vale la DF2 tal que {#opción\_prestamo} que es el determinante de la DF es superclave en P3.

- P4 no está en BCNF ya que existe al menos la DF3 tal que {dni, #prestamo} que es el determinante de la DF no es superclave en P3. Particione para llevar a BCNF considerando DF3.

P5 (dni, #prestamo, fecha\_prestamo, monto\_total, #opción\_prestamo)

P6 (dni, #prestamo, #cuota, monto\_cuota, fecha\_cuota, #condición, #plInternac)

P5 ∩ P6 = clave en P5 {dni, #prestamo}. No se pierde información.

En P5 vale DF3.

En P6 vale DF4.

No se perdieron DFs por validación simple.

P5 está en BCNF porque vale la DF3 tal que {dni, #prestamo} que es el determinante de la DF es superclave en P5.

- P6 no está en BCNF ya que existe al menos la DF4 tal que {dni, #prestamo, #cuota} que es el determinante de la DF no es superclave en P6. Particiono para llevar a BCNF considerando DF4.

P7 (dni, #prestamo, #cuota, monto\_cuota, fecha\_cuota)

P8 (dni, #prestamo, #cuota, #condición, #plInternac)

$P7 \cap P8 = \text{clave en } P7 \{ \text{dni, #prestamo, #cuota} \}$ . No se pierde información.

En P7 vale DF4.

No se perdieron DFs por validación simple.

P7 está en BCNF porque vale la DF4 tal que {dni, #prestamo, #cuota} que es el determinante de la DF es superclave en P7. P8 está en BCNF porque cualquier dependencia funcional que valga es trivial.

#### Particiones en BCNF:

P1 (dni, nombre, fecha\_nacimiento)

P3 (#opción\_prestamo, titulo, condiciones)

P5 (dni, #prestamo, fecha\_prestamo, monto\_total, #opción\_prestamo)

P7 (dni, #prestamo, #cuota, monto\_cuota, fecha\_cuota)

P8 (dni, #prestamo, #cuota, #condición, #plInternac)

CP: {dni, #prestamo, #cuota, #condicion, #plInternac}

#### Dependencias multivaluadas válidas en P8:

DM1) dni, #prestamo  $\rightarrow\!\!\!>$  #cuota

DM2) dni, #prestamo  $\rightarrow\!\!\!>$  #condicion

DM3) {}  $\rightarrow\!\!\!>$  #plInternac

- Dado que en P8 vale al menos DM3 que no es trivial (una dependencia multivaluada  $X \rightarrow Y$  en un esquema  $R$  es trivial cuando la unión de los atributos  $X \cup Y$  abarca todos los atributos de  $R$ ), P8 no cumple la definición de 4FN, entonces particiono P8 considerando la DM3.

P9 (#plInternac)

P10 (dni, #prestamo, #cuota, #condición)

P9 está en 4FN porque vale DM3 tal que DM3 es trivial en P9.

- Dado que en P10 vale al menos DM1 que no es trivial, P10 no cumple la definición de 4FN, entonces particiono P10 considerando la DM1.

P11 (dni, #prestamo, #cuota)

P12 (dni, #prestamo, #condición)

P11 está en 4FN porque vale DM1 tal que DM1 es trivial en P11.

P12 está en 4FN porque vale DM2 tal que DM2 es trivial en P12.

#### Esquemas resultantes en 4FN:

P1 (dni, nombre, fecha\_nacimiento)

P3 (#opción\_prestamo, titulo, condiciones)

P5 (dni, #prestamo, fecha\_prestamo, monto\_total, #opción\_prestamo)

P7 (dni, #prestamo, #cuota, monto\_cuota, fecha\_cuota)

P9 (#plInternac)

P12 (dni, #prestamo, #condición)

Se eliminó el esquema P11 por ser proyección de P7.

CP: {dni, #prestamo, #cuota, #condicion, #plInternac}

#### AR

DUEÑOSFIAT  $\leftarrow \Pi_{dni}(\sigma_{marca = "FIAT"}(DUEÑO | X | AUTO))$

DUEÑOSNOFIAT  $\leftarrow \Pi_{dni}(\sigma_{marca <> "FIAT"}(DUEÑO | X | AUTO))$

$\Pi_{dni, nombre, teléfono}(((DUEÑOSFIAT - DUEÑOSNOFIAT) - (\Pi_{dni}(CHOFER))) | X | DUEÑO)$

#### MySQL

```
DELIMITER //

CREATE PROCEDURE `nuevo_chofer`(IN DNI INTEGER, IN licencia_desde DATE,
licencia_hasta DATE)
BEGIN
    DECLARE nombre_chofer VARCHAR(50);
    DECLARE teléfono_chofer VARCHAR(50);
    DECLARE dirección_chofer VARCHAR(50);

    SELECT nombre, teléfono, dirección
    INTO nombre_chofer, teléfono_chofer, dirección_chofer
    FROM dueño d
    WHERE d.dni = DNI;

    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
    END;

    START TRANSACTION;
```

```
INSERT INTO chofer(nombre, teléfono, dirección,  
fecha_licencia_desde, fecha_licencia_hasta)  
VALUES (nombre_chofer, teléfono_chofer, dirección_chofer,  
licencia_desde, licencia_hasta);  
  
COMMIT;  
END //  
  
DELIMITER ;
```

## Segunda Fecha - 30/11/22

**BBDD1 Parcial**  
2da fecha 30/11/22

**Entidad/Relación**

Uso Interno		
ER	BCNF	4NF

Indique cómo interpreta las cardinalidades del siguiente modelo

```

    graph LR
        Profesor["Profesor"] -->|1..n| Curso["Curso"]
        Profesor -->|1..n| Reclamo["Reclamo"]
        Reclamo -->|1..n| Envio["Envio"]
        Envio -->|1..n| Domicilio["#domicilio"]
        Domicilio -->|1..n| Localidad["localidad"]
        Localidad -->|1..n| Calle["calle"]
        Calle -->|1..n| Numero["numero"]
        Numero -->|1..n| Producto["#producto"]
        Producto -->|1..n| Galletita["Galletita"]
        Galletita -->|1..n| Detalle["Detalle"]
        Detalle -->|1..n| Tienda["Tienda"]
        Tienda -->|1..n| Deposito["deposito"]
        Deposito -->|1..n| Localidad["localidad"]
        Localidad -->|1..n| Calle["calle"]
        Calle -->|1..n| Numero["numero"]
        Numero -->|1..n| Cliente["cuil_cliente"]
        Cliente -->|1..n| Reclamo["Reclamo"]
    
```

Mientras Res cambia  
Para i = 1 to cant\_de\_particiones\_realizadas  
Res = Res U (Res f1 R0) f1 R0  
While (hay cambios en result) do  
For cada dependencia funcional: t->Z en F1 R0  
if (Y <= result) then result := result U Z

**Normalización**

Dado el siguiente esquema:  
LOGISTICA (#envio, fecha\_creacion, fecha\_entrega, costo\_envio, #domicilio, localidad, calle, numero, cuil\_cliente, nombre\_apellido, fecha\_nacimiento, #producto, nombre\_producto, marca, f\_inicio\_precio, precio\_producto, patente\_transporte)

- De los clientes se conoce su CUIL, nombre y apellido, y fecha de nacimiento.
- De los productos se conoce su nombre, marca y se lleva registro de la variación de precio, registrando la fecha de inicio del precio y el valor. Un producto no puede cambiar de precio más de una vez en una fecha determinada.
- De cada cliente se registran los domicilios a los que se le han enviado productos. De cada domicilio se conoce calle, número y localidad. No hay garantías de que no existan dos domicilios con el mismo número, calle y localidad. Además, varios clientes pueden acusar un mismo domicilio.
- De cada envío se registra su fecha de creación, fecha de entrega, costo, el cliente y el domicilio en que debe entregarse. El envío es único en el sistema.
- En un envío pueden distribuirse uno o más productos.
- Se conocen las patentes de los transportes con los que opera la empresa.

Aplicar y explicar el proceso de normalización (el esquema ya está en 1FN).

**4NF**

Aplicar y explicar el proceso de normalización a 4NF del esquema (ya en BCNF):  
LOGISTICA (#envio, #reclamo, #centro\_operaciones, patente\_transporte, fecha\_mantenimiento, #area\_mantenimiento, #conductor\_habilitado)

- Cada centro de operaciones maneja internamente el número de envío de los paquetes que despacha y el #envio podría repetirse para distintos centros.
- Los clientes pueden ingresar reclamos para un envío, por si llega tarde, se extravía, etc. Cada envío define su propia numeración para sus reclamos, es decir que el #reclamo podría repetirse para diferentes envíos.
- De cada transporte se conocen los conductores habilitados para conducirlo.
- Para cada transporte se registran además las tareas de mantenimiento realizadas y en qué fechas se hicieron.

### Normalización - BCNF

DF1: cuil\_cliente → nombre\_apellido, fecha\_nacimiento

DF2: #producto → nombre\_producto, marca

DF3: #producto, f\_inicio\_precio → precio\_producto

DF4: #domicilio → localidad, calle, numero

DF5: #envio → fecha\_creacion, fecha\_entrega, cuil\_cliente, costo\_envio,

#domicilio

CC: {#envio, #producto, f\_inicio\_precio, patente\_transporte}

- LOGISTICA no está en BCNF ya que existe al menos la DF1 tal que {cuil\_cliente} que es el determinante de la DF no es superclave en LOGISTICA. Particiono para llevar a BCNF considerando DF1.

L1 (cuil\_cliente, nombre\_apellido, fecha\_nacimiento)

L2 (#envio, fecha\_creacion, fecha\_entrega, costo\_envio, #domicilio, localidad, calle, numero, cuil\_cliente, #producto, nombre\_producto, marca, f\_inicio\_precio, precio\_producto, patente\_transporte)

L1 ∩ L2 = clave en L1 {cuil\_cliente}. No se pierde información.

En L1 vale DF1.

En L2 vale DF2, DF3, DF4 y DF5.

L1 está en BCNF porque vale la DF1 tal que {cuil\_cliente} que es el determinante de la DF es superclave en L1.

- L2 no está en BCNF ya que existe al menos la DF2 tal que {#producto} que es el determinante de la DF no es superclave en L2. Particiono para llevar a BCNF considerando DF2.

L3 (#producto, nombre\_producto, marca)

L4 (#envio, fecha\_creacion, fecha\_entrega, costo\_envio, #domicilio, localidad, calle, numero, cuil\_cliente, #producto, f\_inicio\_precio, precio\_producto, patente\_transporte)

$L3 \cap L4 = \text{clave en } L3 \{\#producto\}$ . No se pierde información.

En L3 vale DF2.

En L4 vale DF3, DF4 y DF5.

L3 está en BCNF porque vale la DF2 tal que  $\{\#producto\}$  que es el determinante de la DF es superclave en L3.

- L4 no está en BCNF ya que existe al menos la DF4 tal que  $\{\#domicilio\}$  que es el determinante de la DF no es superclave en L4. Particiono para llevar a BCNF considerando DF4.

L5 (#domicilio, localidad, calle, numero)

L6 (#envio, fecha\_creacion, fecha\_entrega, costo\_envio, #domicilio, cuil\_cliente, #producto, f\_inicio\_precio, precio\_producto, patente\_transporte)

$L5 \cap L6 = \text{clave en } L5 \{\#domicilio\}$ . No se pierde información.

En L5 vale DF4.

En L6 vale DF3 y DF5.

L5 está en BCNF porque vale la DF4 tal que  $\{\#domicilio\}$  que es el determinante de la DF es superclave en L5.

- L6 no está en BCNF ya que existe al menos la DF5 tal que  $\{\#envio\}$  que es el determinante de la DF no es superclave en L6. Particiono para llevar a BCNF considerando DF5.

L7 (#envio, fecha\_creacion, fecha\_entrega, cuil\_cliente, costo\_envio, #domicilio)

L8 (#envio, #producto, f\_inicio\_precio, precio\_producto, patente\_transporte)

$L7 \cap L8 = \text{clave en } L7 \{\#envio\}$ . No se pierde información.

En L7 vale DF5.

En L8 vale DF3.

L7 está en BCNF porque vale la DF5 tal que  $\{\#envio\}$  que es el determinante de la DF es superclave en L7.

- L8 no está en BCNF ya que existe al menos la DF3 tal que  $\{\#producto, f_inicio_precio\}$  que es el determinante de la DF no es superclave en L8.

Particiono para llevar a BCNF considerando DF3.

L9 (#producto, f\_inicio\_precio, precio\_producto)

L10 (#envio, #producto, f\_inicio\_precio, patente\_transporte)

$L9 \cap L10 = \text{clave en } L9 \{\#producto\}$ . No se pierde información.

En L9 vale DF3.

L9 está en BCNF porque vale la DF3 tal que  $\{\#producto, f\_inicio\_precio\}$  que es el determinante de la DF es superclave en L9. L10 está en BCNF porque cualquier dependencia funcional que valga es trivial.

#### Particiones en BCNF:

- L1 (cuil\_cliente, nombre\_apellido, fecha\_nacimiento)
- L3 (#producto, nombre\_producto, marca)
- L5 (#domicilio, localidad, calle, numero)
- L7 (#envio, fecha\_creacion, fecha\_entrega, cuil\_cliente, costo\_envio, #domicilio)
- L9 (#producto, f\_inicio\_precio, precio\_producto)
- L10 (#envio, #producto, f\_inicio\_precio, patente\_transporte)

CP:  $\{\#envio, \#producto, f\_inicio\_precio, patente\_transporte\}$

#### Normalización - 4FN:

##### Dependencias multivaluadas válidas en LOGISTICA:

- DM1)  $\#centro\_operaciones, \#envio \rightarrow\!\!> \#reclamo$
- DM2)  $\text{patente\_transporte} \rightarrow \#conductor\_habilitado$
- DM3)  $\text{patente\_transporte} \rightarrow \text{fecha\_mantenimiento}, \#tarea\_mantenimiento$

- Dado que en LOGISTICA vale al menos la DM3 que no es trivial (una dependencia multivaluada  $X \twoheadrightarrow Y$  en un esquema  $R$  es trivial cuando la unión de los atributos  $X \cup Y$  abarca todos los atributos de  $R$ ), LOGISTICA no cumple la definición de 4FN, entonces particiono LOGISTICA considerando la DM3.

- L1(patente\_transporte, fecha\_mantenimiento, #tarea\_mantenimiento)
- L2(#envio, #reclamo, #centro\_operaciones, patente\_transporte, #conductor\_habilitado)

L1 está en 4FN porque vale la DM3 tal que DM3 es trivial en L1.

- Dado que en L2 vale al menos la DM2 que no es trivial, L2 no cumple la definición de 4FN, entonces particiono L2 considerando la DM2.

- L3(patente\_transporte, #conductor\_habilitado)
- L4(#envio, #reclamo, #centro\_operaciones, patente\_transporte)

L3 está en 4FN porque vale la DM2 tal que DM2 es trivial en L3.

- Dado que en L4 vale al menos la DM1 que no es trivial, L4 no cumple la definición de 4FN, entonces particiono L4 considerando la DM1.

- L5(#centro\_operaciones, #envio, #reclamo)
- L6(#envio, #centro\_operaciones, patente\_transporte)

L5 está en 4FN porque vale la DM1 tal que DM1 es trivial en L5.

¿Qué pasa con L6? ¿Está en 4FN? L6 no está en 4FN. Aparecen dos nuevas DM (sólo son válidas en L6):

DM4) {} ->> patente\_transporte  
DM5) #centro\_operaciones ->> #envio

- Dado que en L6 vale al menos la DM4 que no es trivial, L6 no cumple la definición de 4FN, entonces particiono L6 considerando la DM4.

L7 (patente\_transporte)

L8 (#envio, #centro\_operaciones)

L7 está en 4FN porque vale la DM4 tal que DM4 es trivial en L7.

L8 está en 4FN porque vale la DM5 tal que DM5 es trivial en L8.

Esquemas resultantes en 4FN:

L1(patente\_transporte, fecha\_mantenimiento, #tarea\_mantenimiento)

L3(patente\_transporte, #conductor\_habilitado)

L5(#centro\_operaciones, #envio, #reclamo)

Se eliminaron los esquemas L7 y L8 por ser proyecciones. L7 es proyección de L1 y L8 de L5.

## SF1

Nº de Alum: \_\_\_\_\_  
Apellido y Nombre: \_\_\_\_\_

BBDD1 Parcial  
1ra fecha 16/11

Uso Interno		
ER	Norm	Norm 4NF

**Entidad/Relación**  
Indique cómo interpreta las cardinalidades del siguiente modelo

Un profesor dicta \_\_\_\_\_  
Un curso es dictado por \_\_\_\_\_

Se quiere realizar un sistema para el seguimiento de pagos de tarjetas de crédito.

Una tarjeta pertenece a un titular, de quien se sabe el nombre completo, CUIL, teléfono y fecha de nacimiento. El titular puede ser cliente de varios bancos, y en cada banco puede tener una o más tarjetas. De la tarjeta en sí se conoce el número, fechas de emisión y vencimiento y CVV (código validador), el titular y el banco que la emitió.

De los bancos se conoce su CUIT y la razón social.

Con una tarjeta se pueden realizar compras, de las cuales se sabe el monto, la fecha y el número de autorización. Las compras pueden ser de dos tipos: en un pago o en cuotas. De las compras en cuotas se sabe además el interés y el número de cuotas. Además, de cada cuota se guarda el vencimiento y el monto.

Los pagos se registran una vez por mes, y de cada pago de una tarjeta se sabe qué cuotas se pagaron, y en el caso de las compras en cuotas, se sabe también qué cuota se pagó.

Normalización

Dado el siguiente esquema:

```
CUOTAS(#producto, nombre_prod, desc_prod, #plan_cuotas, cft, nro_cuotas,
#tipo_tarjeta, nombre_tarjeta, #banco, nombre_banco, #premio, desc_premio,
puntos, #categoria_prod)
```

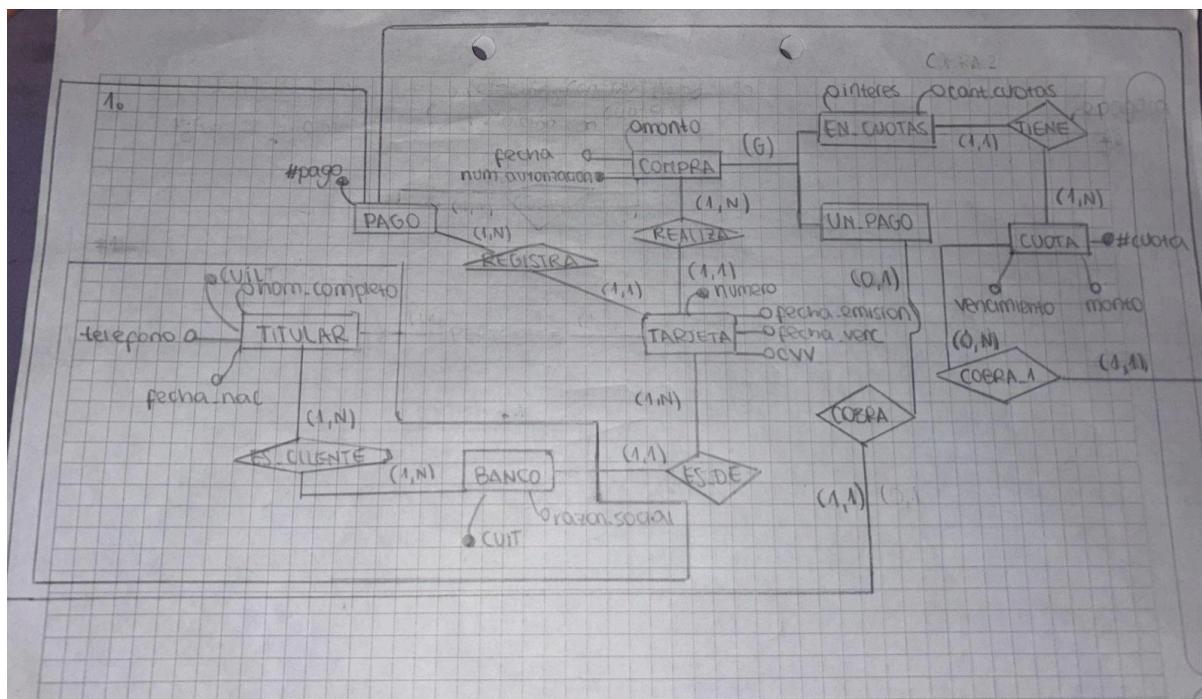
Donde:

- cada **#tipo\_tarjeta**, único en el sistema, identifica un tipo de tarjeta (es decir, VISA, MasterCard, Amex, etc). Para cada tipo se conoce el nombre.
- #banco es único en el sistema. De cada banco se conoce el nombre.
- cada plan de pago en cuotas está identificado por **#plan\_cuotas** único en el sistema y tiene un número de cuotas (**nro\_cuotas**), y un costo financiero total (**cft**). De cada plan se sabe para qué tipo de tarjeta de qué banco es válido, por ejemplo *VISA del Banco Provincia*.
- cada producto ofrecido tiene un nombre y descripción, y puede tener varios planes de pago en cuotas (**#plan\_cuotas**). Un mismo plan de cuotas podría ofrecerse para múltiples productos.
- cada producto tiene asociadas varias categorías (**#categoria\_prod**)
- Además existen premios disponibles a los cuales pueden acceder los clientes canjeando los puntos acumulados en sus tarjetas. De cada premio se conoce una descripción y los puntos requeridos.

Aplicar y explicar el proceso de normalización hasta 4NF o la máxima forma normal posible. Considerar que el esquema ya se encuentra en 1FN.

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

## ER



TITULAR ( <u>CUIL</u> , nom_completo, telefono, fecha_nac)
BANCO ( <u>CUIT</u> , razon_social)
ES_CLIENTE ( <u>CUIL</u> , <u>CUIT</u> )
TARJETA ( <u>numero</u> , <u>fecha_emision</u> , <u>fecha_venc</u> , <u>CVN</u> )
ES_DE ( <u>numero</u> , <u>CUIL</u> , <u>CUIT</u> )
COMPRA ( <u>num_authorized</u> , <u>fecha</u> , <u>monto</u> )
REALIZA ( <u>num_authorized</u> , <u>numero</u> )
EN CUOTAS ( <u>num_authorized</u> , <u>interes</u> , <u>cant_cuotas</u> )
CUOTA ( <u>#cuota</u> , <u>vencimiento</u> , <u>monto</u> )
TIENE ( <u>#cuota</u> , <u>num_authorized</u> )
PAGO ( <u>#pago</u> )
REGISTRA ( <u>#pago</u> , <u>numero</u> )
COBRA ( <u>#pago</u> , <u>num_authorized</u> )
COBRA_1 ( <u>#cuota</u> , <u>#pago</u> )

### Normalización

DF1: #tipo\_tarjeta → nombre\_tarjeta

DF2: #banco → nombre\_banco

DF3: #plan\_cuotas → nro\_cuotas, cft, #tipo\_tarjeta, #banco

DF4: #producto → nombre\_prod, desc\_prod

DF5: #premio → desc\_premio, puntos

CC: {#producto, #plan\_cuotas, #premio, #categoria\_prod}

- CUOTAS no está en BCNF ya que existe al menos la DF1 tal que {#tipo\_tarjeta} que es el determinante la DF no es superclave en CUOTAS. Particiono para llevar a BCNF considerando DF1.

C1 (#tipo\_tarjeta, nombre\_tarjeta)

C2 (#producto, nombre\_prod, desc\_prod, #plan\_cuotas, cft, nro\_cuotas, #tipo\_tarjeta, #banco, nombre\_banco, #premio, desc\_premio, puntos, #categoria\_prod)

C1  $\cap$  C2 = clave en C1 {#tipo\_tarjeta}. No se pierde información.

En C1 vale DF1.

En C2 vale DF2, DF3, DF4 y DF5.

No se perdieron DFs por validación simple.

C1 está en BCNF porque vale la DF1 tal que {#tipo\_tarjeta} que es el determinante de la DF es superclave en C1.

- C2 no está en BCNF ya que existe al menos la DF2 tal que {#banco} que es el determinante de la DF no es superclave en C2. Particiono para llevar a BCNF considerando DF2.

C3 (#banco, nombre\_banco)

C4 (#producto, nombre\_prod, desc\_prod, #plan\_cuotas, cft, nro\_cuotas, #tipo\_tarjeta, #banco, #premio, desc\_premio, puntos, #categoria\_prod)

C3  $\cap$  C4 = clave en C3 {#banco}. No se pierde información.

En C3 vale DF2.

En C4 vale DF3, DF4 y DF5.

No se perdieron DFs por validación simple.

C3 está en BCNF porque vale la DF2 tal que {#banco} que es el determinante de la DF es superclave en C3.

- C4 no está en BCNF ya que existe al menos la DF3 tal que {#plan\_cuotas} que es el determinante de la DF no es superclave en C4. Particiono para llevar a BCNF considerando DF3.

C5 (#plan\_cuotas, nro\_cuotas, cft, #tipo\_tarjeta, #banco)

C6 (#producto, nombre\_prod, desc\_prod, #plan\_cuotas, #premio, desc\_premio, puntos, #categoria\_prod)

C5  $\cap$  C6 = clave en C5 {#plan\_cuotas}. No se pierde información.

En C5 vale DF3.

En C6 vale DF4 y DF5.

No se perdieron DFs por validación simple.

C5 está en BCNF porque vale la DF3 tal que {#plan\_cuotas} que es el determinante de la DF es superclave en C5.

- C6 no está en BCNF ya que existe al menos la DF4 tal que {#producto} que es el determinante de la DF no es superclave en C6. Particiono para llevar a BCNF considerando DF4.

C7 (#producto, nombre\_prod, desc\_prod)

C8 (#producto, #plan\_cuotas, #premio, desc\_premio, puntos,  
#categoria\_prod)

$C7 \cap C8 = \text{clave en } C7 \{ \#producto \}$ . No se pierde información.

En C7 vale DF4.

En C8 vale DF5.

No se perdieron DFs por validación simple.

C7 está en BCNF porque vale la DF4 tal que  $\{ \#producto \}$  que es el determinante de la DF es superclave en C7.

- C8 no está en BCNF ya que existe al menos la DF5 tal que  $\{ \#premio \}$  que es el determinante de la DF no es superclave en C8. Particiono para llevar a BCNF considerando DF5.

C9 (#premio, desc\_premio, puntos)

C10 (#producto, #plan\_cuotas, #premio, #categoria\_prod)

$C9 \cap C10 = \text{clave en } C9 \{ \#premio \}$ . No se pierde información.

En C9 vale DF5.

No se perdieron DFs por validación simple.

C9 está en BCNF ya que existe al menos la DF5 tal que  $\{ \#premio \}$  que es el determinante de la DF es superclave en C9. C10 está en BCNF ya que cualquier dependencia funcional que valga es trivial.

#### Particiones en BCNF:

C1 (#tipo\_tarjeta, nombre\_tarjeta)

C3 (#banco, nombre\_banco)

C5 (#plan\_cuotas, nro\_cuotas, cft, #tipo\_tarjeta, #banco)

C7 (#producto, nombre\_prod, desc\_prod)

C9 (#premio, desc\_premio, puntos)

C10 (#producto, #plan\_cuotas, #premio, #categoria\_prod)

CP:  $\{ \#producto, \#plan_cuotas, \#premio, \#categoria_prod \}$

#### Dependencias multivaluadas válidas en C10:

DM1)  $\{ \} \rightarrow\!\!\!> \#premio$

DM2)  $\#producto \rightarrow\!\!\!> \#categoria_prod$

DM3)  $\#producto \rightarrow\!\!\!> \#plan_cuotas$

- Dado que en C10 vale al menos la DM1 que no es trivial (una dependencia multivaluada  $X \rightarrow Y$  en un esquema  $R$  es trivial cuando la unión de los atributos  $X \cup Y$  abarca todos los atributos de  $R$ ), C10 no cumple la definición de 4FN, entonces particiono C10 considerando la DM1.

C11 (#premio)

C12 (#producto, #plan\_cuotas, #categoria\_prod)

C11 está en 4FN porque vale la DM1 tal que DM1 es trivial en C11.

- Dado que en C12 vale al menos la DM2 que no es trivial, C12 no cumple la definición de 4FN, entonces particiono C12 considerando la DM2.

C13 (#producto, #cateogria\_prod)

C14 (#producto, #plan\_cuotas)

C13 está en 4FN porque vale la DM2 tal que DM2 es trivial en C13.

C14 está en 4FN porque vale la DM3 tal que DM3 es trivial en C14.

### Esquemas resultantes en 4FN:

C1 (#tipo\_tarjeta, nombre\_tarjeta)

C3 (#banco, nombre\_banco)

C5 (#plan\_cuotas, nro\_cuotas, cft, #tipo\_tarjeta, #banco)

C7 (#producto, nombre\_prod, desc\_prod)

C9 (#premio, desc\_premio, puntos)

C13 (#producto, #cateogria\_prod)

C14 (#producto, #plan\_cuotas)

Se eliminó el esquema C11 por ser proyección de C9.

CP: {#producto, #plan\_cuotas, #premio, #categoria\_prod}

## SF2

parcial segunda fecha.jpg

Nº de Alumno: <u>15303/4</u>	Apellido y Nombre: <u>Sáenz Abril</u>
BBDD1 Parcial	Uso interno
2da fecha 4/12	ER      AR      Norm      MySQL
Entidad/Relación	CORREGIDO POR: <u>Julián</u>
<p>Indique cómo interpreta las cardinalidades del siguiente modelo</p> <pre>     graph TD       Curso -- "1,n" --&gt; Profesor       Profesor -- "1" --&gt; Alumno       Curso -- "n,m" --&gt; Alumno   </pre> <p>Un servicio de TV permite a sus usuarios organizar sus suscripciones a plataformas de contenidos digitales (ej. Netflix, Flow). De cada usuario se conoce su nombre y teléfono. Los contenidos ofrecidos por las plataformas consisten en películas y capítulos de series, pero pueden surgir otros tipos en el futuro. Cada contenido es exclusivo de una plataforma y tiene un nombre, duración y puntaje. Las películas además tienen un género, y de los capítulos de serie se conoce la serie a la que pertenecen. Cada serie a su vez tiene un nombre y un puntaje propio (independiente de los de sus capítulos). De cada plataforma se conoce el nombre y la url. Para cada suscripción que tiene un usuario con cada plataforma, se guardan el nombre de usuario y contraseña utilizados, y la fecha de vencimiento de la misma. También se conoce, para cada suscripción, cuáles son los contenidos vistos y los marcados como favoritos por el usuario en la plataforma (puede marcar como favorito un contenido de una plataforma aunque no lo haya visto aún). Además, de cada contenido visto por un usuario, se conoce el número de veces que se reprodujo y puede haber o no una reseña que incluye un comentario y puntaje otorgado. Un usuario no puede tener más de una suscripción con una misma plataforma.</p> <p>Realizar el modelo E/R y la transformación al modelo relacional.</p> <p><b>Álgebra Relacional</b></p> <p>Dados los siguientes esquemas:</p> <pre> CAPITULO(#capítulo, #serie, nombre, puntaje_promedio) SERIE(#serie, nombre, puntaje_promedio) RESEÑA(#reseña, #usuario, #capítulo, puntaje, comentario) USUARIO(#usuario, email, nombre)   </pre> <p>Hallar el #serie de las series que tienen todos sus capítulos con al menos una reseña.</p>	

Algoritmo para analizar la pérdida de dfs.

```

Res = x
Mientras Res cambia
  Para i = 1 to cant_de_particiones_realizadas
    Res = Res U(Res ∩ R)i ∩ R
  
```

Algoritmo para encontrar X\*

```

Result := X
While (hay cambios en result) do
  For (cada dependencia funcional Y->Z en F) do
    if (Y ⊂ result) then result := result U Z
  
```

**Normalización**

Dado el siguiente esquema:

```

SUSCRIPCION(#suscripción, email, nombre_usuario, #plan, nombre_plan,
            texto_condiciones, precio, email_adicional, nombre_adicional, #contenido,
            título, sinopsis, duración, fecha_adicional)
  
```

Donde:

- Cada suscripción es realizada por un único usuario (identificado por el email) y un plan, pero además hay usuarios adicionales que la utilizan (email\_adicional).
- De cada usuario adicional que se suma a la suscripción, se guarda la fecha.
- Un plan de suscripción tiene un nombre (que no puede garantizarse que sea único en el sistema), condiciones, y un precio mensual.
- Cada contenido tiene un título, sinopsis y duración. El #contenido es único en el sistema, pero del título no puede garantizarse que lo sea.
- De cada suscripción se sabe qué contenidos fueron reproducidos, sin distinción sobre qué usuario (titular o adicionales) reprodujo cada uno.

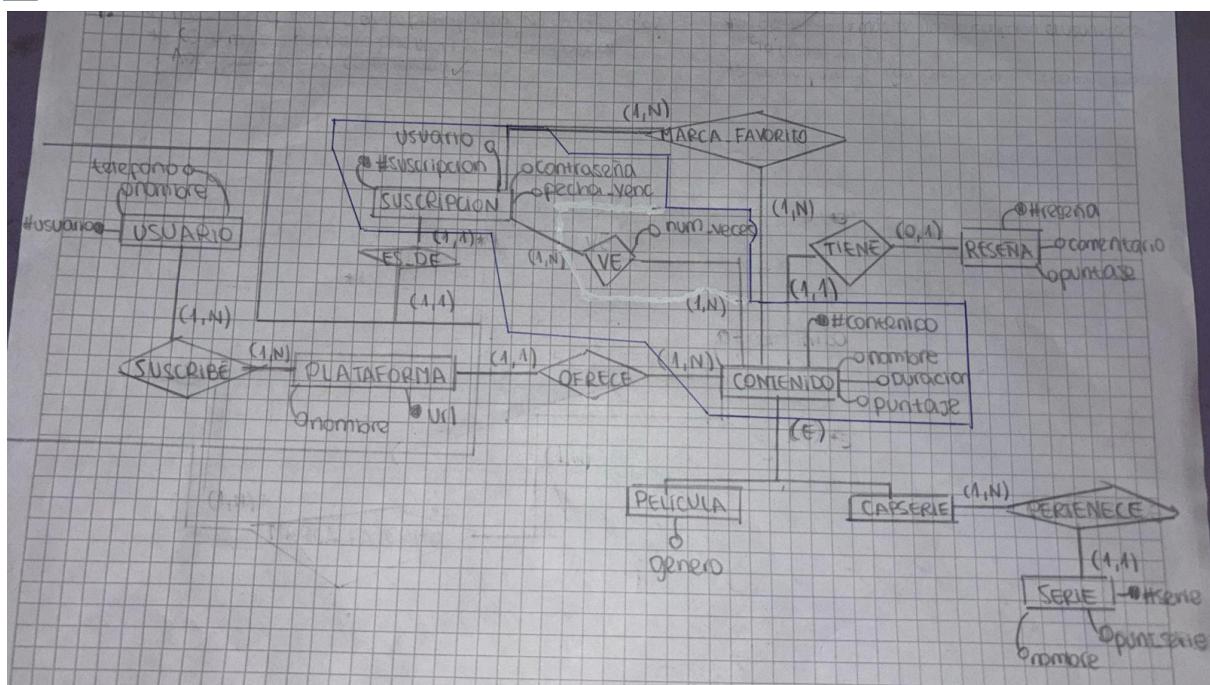
Aplicar y explicar el proceso de normalización. Considerar que el esquema ya se encuentra en 1FN.

**MySQL**

- Considerando las propiedades ACID de las transacciones y priorizando que esto se cumpla, ¿es necesario usar transacciones cuando se ejecuta una sentencia UPDATE? Justifique su respuesta.
- Explique 2 motivos por los cuales necesitaría definir una vista.
- En una base de datos con las siguientes tablas

CONTENIDO(#id\_contenido, título, sinopsis)  
 VISUALIZACION(#id, timestamp, #id\_contenido)

Hace falta crear un usuario para el frontend de la aplicación que sea capaz de obtener información sobre los contenidos, y registrar visualizaciones (pero no consultarlas). ¿Qué permisos le asignaría a cada tabla, para maximizar la seguridad de la BD?

**ER**

USUARIO (#usuario, telefono, nombre)  
 PLATAFORMA (url, nombre)  
 SUSCRIBE (#usuario, url)  
 CONTENIDO (#contenido, nombre, duracion, puntaje)  
 OFRECE (#contenido, url)  
 PELICULA (#contenido, genero)  
 CAPSERIE (#contenido)  
 SERIE (#serie, punt\_serie, nombre)  
 SUSCRIPCION (#suscripcion, usuario, contraseña, fecha\_venc)  
 ES\_DE (#suscripcion, #usuario, url)  
 VE (#suscripcion, #contenido, num\_veces)  
 MARCA\_FAVORITO (#suscripcion, #contenido)  
 RESEÑA (#resena, contenido, puntaje)  
 TIENE (#resena, #suscripcion, #contenido)

### Normalización

- DF1. #suscripcion -> email, #plan
- DF2. #plan -> nombre\_plan, texto\_condiciones, precio
- DF3. email -> nombre\_usuario
- DF4. #contenido -> titulo, sinopsis, duracion
- DF5. #suscripcion, email\_adicional -> fecha\_adicional
- DF6. email\_adicional -> nombre\_adicional

CC: {#suscripcion, #contenido, email\_adicional}

- SUSCRIPCION no está en BCNF ya que existe al menos la DF2 tal que  $\{\#plan\}$  que es el determinante de la DF no es superclave en SUSCRIPCION. Particiono para llevar a BCNF considerando DF2.

S1 (#plan, nombre\_plan, texto\_condiciones, precio)

S2 (#suscripcion, email, nombre\_usuario, #plan, email\_adicional,

nombre\_adicional, #contenido, titulo, sinopsis, duracion, fecha\_adicional)

S1  $\cap$  S2 = clave en S1  $\{\#plan\}$ . No se pierde información.

En S1 vale la DF2.

En S2 vale la DF1, DF3, DF4, DF5 y DF6.

No se perdieron DFs por validación simple.

S1 está en BCNF porque vale DF2 tal que  $\{\#plan\}$  que es el determinante de la DF es superclave en S1.

- S2 no está BCNF ya que existe al menos DF3 tal que  $\{email\}$  que es el determinante de la DF no es superclave en S2. Particiono para llevar a BCNF considerando DF3.

S3 (email, nombre\_usuario)

S4 (#suscripcion, email, #plan, email\_adicional, nombre\_adicional, #contenido, titulo, sinopsis, duracion, fecha\_adicional)

S3  $\cap$  S4 = clave en S3  $\{email\}$ . No se pierde información.

En S3 vale la DF3.

En S4 vale la DF1, DF4, DF5 y DF6.

No se perdieron DFs por validación simple..

S3 está en BCNF porque vale DF3 tal que  $\{email\}$  que es el determinante de la DF es superclave en S3.

- S4 no está BCNF ya que existe al menos DF6 tal que  $\{email\_aditional\}$  que es el determinante de la DF no es superclave en S4. Particiono para llevar a BCNF considerando DF6.

S5 (email\_adicional, nombre\_adicional)

S6 (#suscripcion, email, #plan, email\_adicional, #contenido, titulo, sinopsis, duracion, fecha\_adicional)

S5  $\cap$  S6 = clave en S5  $\{email\_aditional\}$ . No se pierde información.

En S5 vale la DF6.

En S6 vale la DF1, DF4 y DF5.

No se perdieron DFs por validación simple..

S5 está en BCNF porque vale DF6 tal que  $\{email\_aditional\}$  que es el determinante de la DF es superclave en S5.

- S6 no está BCNF ya que existe al menos DF1 tal que  $\{\#suscripcion\}$  que es el determinante de la DF no es superclave en S6. Particiono para llevar a BCNF considerando DF1.

S7 (#suscripcion, email, #plan)

S8 (#suscripcion,email\_adicional,#contenido, titulo, sinopsis, duracion, fecha\_adicional)

$S7 \cap S8 = \text{clave en } S7 \{\#suscripcion\}$ . No se pierde información.

En S7 vale DF1.

En S8 vale DF4 y DF5.

No se perdieron DFs por validación simple..

S7 está en BCNF porque vale DF1 tal que  $\{\#suscripcion\}$  que es el determinante de la DF es superclave en S7.

- S8 no está BCNF porque existe al menos la DF4 tal que  $\{\#contenido\}$  que es el determinante de la DF no es superclave en S8. Particiono para llevar a BCNF considerando DF4.

S9 (#contenido, titulo, sinopsis, duración)

S10 (#suscripcion,email\_adicional,#contenido, fecha\_adicional)

$S9 \cap S10 = \text{clave en } S9 \{\#contenido\}$ . No se pierde información.

En S9 vale DF4.

En S10 vale DF5.

No se perdieron DFs por validación simple..

S9 está en BCNF porque vale DF4 tal que  $\{\#contenido\}$  que es el determinante de la DF es superclave en S9.

- S10 no está BCNF porque existe al menos la DF5 tal que  $\{\#suscripcion, email_adicional\}$  que es el determinante de la DF no es superclave en S10.

Particiono para llevar a BCNF considerando DF5.

S11 (#suscripcion,email\_adicional, fecha\_adicional)

S12 (#suscripcion,email\_adicional,#contenido)

$S11 \cap S12 = \text{clave en } S11 \{\#suscripcion, email_adicional\}$ . No se pierde información.

En S11 vale DF5.

No se perdieron DFs por validación simple..

S11 está en BCNF porque vale DF5 tal que  $\{\#suscripcion, email_adicional\}$  que es el determinante de la DF es superclave en S11. S12 está BCNF porque cualquier dependencia funcional que valga es trivial.

#### Particiones en BCNF:

S1 (#plan,nombre\_plan, texto\_condiciones, precio)

S3 (email,nombre\_usuario)

S5 (email\_adicional,nombre\_adicional)

S7 (#suscripcion,email,#plan)

S9 (#contenido, titulo, sinopsis, duración)  
 S11 (#suscripcion, email\_adicional, fecha\_adicional)  
 S12 (#suscripcion, email\_adicional, #contenido)

CP: {#suscripcion, #contenido, email\_adicional}

Dependencias multivaluadas válidas en S12:

- DM1) #suscripcion ->> email\_adicional
- DM2) #suscripcion ->> contenido

Dado que en S12 vale al menos la DM1 que no es trivial (una dependencia multivaluada  $X \rightarrow Y$  en un esquema  $R$  es trivial cuando la unión de los atributos  $X \cup Y$  abarca todos los atributos de  $R$ ), S12 no cumple la definición de 4FN, entonces particiono I12 considerando la DM1.

S13 (#suscripcion, email\_adicional)  
 S14 (#suscripcion, #contenido)

S13 está en 4FN porque vale la DM1 tal que DM1 es trivial en S13.

S14 está en 4FN porque vale la DM2 tal que DM2 es trivial en S14.

Esquemas resultantes en 4FN:

S1 (#plan\_nombre\_plan, texto\_condiciones, precio)  
 S3 (email, nombre\_usuario)  
 S5 (email\_adicional, nombre\_adicional)  
 S7 (#suscripcion, email, #plan)  
 S9 (#contenido, titulo, sinopsis, duración)  
 S11 (#suscripcion, email\_adicional, fecha\_adicional)  
 S14 (#suscripcion, #contenido)

Se eliminó el esquema S13 por ser proyección de S12.

CP: {#suscripcion, #contenido, email\_adicional}

## **AR**

CAPCONRES  $\leftarrow \Pi_{\#capítulo}(RESEÑA)$   
 CAPSINRES  $\leftarrow \Pi_{\#capítulo}(CAPITULO) - CAPCONRES$   
 SERIESINRES  $\leftarrow \Pi_{\#serie}(SERIE | X | CAPSINRES)$   
 $\Pi_{\#serie}(SERIE) - CAPSINRES$

## **MySQL**

1. No es necesario usar transacciones explícitas para una única sentencia UPDATE, ya que por sí misma cumple las propiedades ACID.

Sin embargo, sí forma parte de un conjunto de operaciones que deben mantenerse coherentes entre sí, entonces sí debe incluirse dentro de una transacción explícita.

2. - Abstracción y simplificación de consultas complejas: Usar una vista para ocultar la complejidad de consultas frecuentes (joins, agregaciones, filtros) y presentar una tabla virtual fácil de usar.

- Seguridad / control de acceso a los datos: Usar una vista para exponer solo las columnas o filas permitidas a ciertos usuarios, restringiendo el acceso a datos sensibles.

3.

```
-- Quitar todos los permisos
REVOKE ALL PRIVILEGES ON plataforma.VISUALIZACION FROM
'frontend_user'@'localhost';
REVOKE ALL PRIVILEGES ON plataforma.CONTENIDO FROM
'frontend_user'@'localhost';

-- Contenido -> SELECT | Visualizacion -> INSERT
GRANT SELECT ON plataforma.CONTENIDO TO 'frontend_user'@'localhost';
GRANT INSERT ON plataforma.VISUALIZACION TO
'frontend_user'@'localhost';
```

**IA | AR y SQL:** **Ejercicio 1 – Álgebra Relacional**

Contexto: Una empresa de envíos administra la entrega de paquetes por diferentes transportistas.

Esquemas:

Envio(#envio, fecha\_envio, #cliente, #transporte, destino, costo)

Cliente(#cliente, nombre, ciudad)

Transporte(#transporte, nombre\_transporte, tipo)

Paquete(#envio, #producto, peso)

Producto(#producto, nombre\_producto, categoria)

Consultas en álgebra relacional:

1. Obtener los nombres de los clientes que hicieron envíos de productos de la categoría “Electrónica”.

CATELECTRONICA  $\leftarrow \Pi_{\#producto}(\sigma_{categoria = 'Electrónica'}(PRODUCTO))$

$\Pi_{nombre}(CLIENTE |X| ENVIO |X| PAQUETE |X| CATELECTRONICA)$

2. Listar los transportes que no realizaron ningún envío.

$\Pi_{\#transporte, nombre\_transporte}(TRANSPORTE) - \Pi_{\#transporte, nombre\_transporte}(TRANSPORTE |X| ENVIO)$

3. Mostrar los clientes que realizaron envíos con todos los transportes de tipo “Tercerizado”.

TERCERIZADO  $\leftarrow \Pi_{\#transporte}(\sigma_{tipo = 'Tercerizado'}(TRANSPORTE))$

$\Pi_{\#cliente}((CLIENTE |X| ENVIO) \% TERCERIZADO)$

 **Ejercicio 2 – SQL**

Contexto: Una librería online gestiona las compras de sus clientes.

Esquema:

Cliente(#cliente, nombre, email)

Libro(#libro, titulo, autor, precio)

Compra(#compra, fecha\_compra, #cliente)

DetalleCompra(#compra, #libro, cantidad)

#### A. Consultas SQL

1. Mostrar los nombres y correos de los clientes que compraron libros del autor 'Borges'.
2. Listar los libros que nunca fueron comprados.
3. Calcular el importe total gastado por cada cliente (nombre y total).
4. Obtener el autor que más ventas tuvo (en cantidad de ejemplares vendidos).

```
-- 1.
SELECT DISTINCT c.nombre, c.email
FROM cliente c
INNER JOIN compra cr ON c.#cliente = cr.#cliente
INNER JOIN detallecompra dc ON cr.#compra = dc.#compra
INNER JOIN libro l ON dc.#libro = l.#libro
WHERE l.titulo = 'Borges';

-- 2.
SELECT l.titulo
FROM libro l
WHERE l.#libro NOT IN (
    SELECT dc.#libro
    FROM detallecompra dc
);

-- 3.
SELECT c.nombre, SUM(dc.cantidad * l.precio)
FROM cliente c
INNER JOIN compra cr ON c.#cliente = cr.#cliente
INNER JOIN detallecompra dc ON cr.#compra = dc.#compra
INNER JOIN libro l ON dc.#libro = l.#libro
GROUP BY c.#cliente, c.nombre;

-- 4.
SELECT l.autor, SUM (dc.cantidad) as cantTotal
FROM libro l
INNER JOIN detallecompra dc ON l.#libro = dc.#libro
GROUP BY l.autor
ORDER BY cantTotal DESC
LIMIT 1;
```

## B. Procedimiento almacenado

- Crear un procedimiento registrar\_compra que:
- Reciba: p\_cliente, p\_libro, p\_cantidad.
- Inserte una nueva compra en la tabla Compra con la fecha actual.
- Inserte el detalle en DetalleCompra.
- Use START TRANSACTION y COMMIT.
- Si ocurre un error, haga ROLLBACK.

```
DELIMITER //
```

```

CREATE PROCEDURE `registrar_compra` (IN p_cliente INTEGER, IN p_libro
INTEGER, IN p_cantidad INTEGER)
BEGIN
    DECLARE last_id INTEGER;

    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
    END;

    START TRANSACTION;

    INSERT INTO compra(fecha_compra, #cliente)
VALUES (CURDATE(), p_cliente);

    SET last_id = LAST_INSERT_ID();

    INSERT INTO detallecompra(#compra, #libro, #cantidad)
VALUES(last_id, p_libro, p_cantidad);

    COMMIT;
END //

DELIMITER ;

```

### C. Trigger

Crear un trigger llamado update\_precio\_promedio que:

- Se ejecute después de cada inserción en DetalleCompra.
- Actualice en una tabla EstadisticaLibro(#libro, cant\_vendida, total\_recaudado):
  - Si el libro ya existe, sume la cantidad y el monto.
  - Si no existe, lo inserte con los valores iniciales.

```

DELIMITER //

CREATE TRIGGER `update_precio_promedio`
AFTER INSERT ON detallecompra
FOR EACH ROW
BEGIN
    IF EXISTS (SELECT 1 FROM estadisticaLibro WHERE #libro =
NEW.#libro) THEN
        UPDATE estadisticaLibro
        SET cant_vendida = cant_vendida + NEW.cantidad,

```

```

        total_recaudado = total_recaudado + (NEW.cantidad * (SELECT
precio FROM libro WHERE #libro = NEW.#libro))
        WHERE #libro = NEW.#libro;
    ELSE
        INSERT INTO estadisticalibro (#libro, cant_vendida,
total_recaudado)
        VALUES (NEW.#libro, NEW.cantidad, (NEW.cantidad * (SELECT
precio FROM libro WHERE #libro = NEW.#libro)));
    END IF;
END; //

DELIMITER ;

```

## IA | Visualización

### Ejercicio 1 – Ventas mensuales de productos

Contexto: Una cadena de tiendas desea analizar sus ventas mensuales por categoría de producto.

Esquema:

**Sucursal** (#sucursal, ubicación)

**Producto**(#producto, nombre\_prod, categoria, precio\_unit)

**Venta**(#venta, fecha\_venta, #sucursal, monto\_total)

**ItemVenta**(#venta, #producto, cantidad)

Consultas a visualizar:

- La cantidad total de productos vendidos por categoría, durante el último año.
- Los ingresos mensuales totales por sucursal.

Para el primer punto usaría un gráfico de barras ya que permite comparar qué productos se venden más por categoría durante el último año. En el eje X se representarían las categorías de productos, y en el eje Y la cantidad vendida correspondiente a cada categoría.

Para el segundo punto usaría un gráfico de líneas para cada sucursal ya que muestra las relaciones de los cambios en los datos en un período de tiempo, en este caso la evolución mensual de los ingresos, viendo los ascensos y descensos en las ventas. En el eje X se representarían los meses, y en el eje Y el total de ingresos correspondiente a cada mes.

Tablas importantes para el primer punto:

**Producto**(#producto, nombre\_prod, categoria, precio\_unit)

**ItemVenta**(#venta, #producto, cantidad)

Las tablas Producto e ItemVenta son relevantes para el análisis, ya que la primera (Producto) identifica las categorías de los productos (atributo categoria), mientras que la segunda (ItemVenta) proporciona los datos numéricos (atributo cantidad) necesarios para calcular la cantidad total de productos vendidos por cada categoría. Ambas se relacionan mediante el campo #producto, que permite vincular cada ítem vendido con su categoría correspondiente.

Tablas importantes para el segundo punto:

**Venta**(#venta, fecha\_venta, #sucursal, monto\_total)

**Sucursal** (#sucursal, ubicación)

Las tablas Venta y Sucursal son relevantes para el análisis, ya que la primera (Venta) contiene los datos económicos de cada operación, como el monto\_total y la fecha\_venta, necesarios para calcular los ingresos mensuales, mientras que la segunda (Sucursal) permite identificar la sucursal específica (atributo #sucursal) sobre la cual se realizará el análisis. Ambas se relacionan a través del campo #sucursal, que vincula cada venta con la sucursal donde se realizó.

 *Ejercicio 2 – Control de temperatura y humedad*

Contexto: Un laboratorio registra la temperatura y humedad diaria en distintos invernaderos.

Esquema:

**Medicion**(#medicion, fecha, temperatura, humedad, #invernadero)

**Invernadero**(#invernadero, nombre, ubicacion)

Consultas a visualizar:

- La variación diaria de la temperatura promedio de cada invernadero durante una semana.
- La relación entre temperatura y humedad en un mismo día.

Para el primer punto usaría un gráfico de múltiples líneas ya que este muestra las relaciones de los cambios en los datos en un período de tiempo, facilitando la identificación de tendencias y comparar cada invernadero. Se podría ver el aumento, disminución o estabilidad de las temperaturas durante un año. En el eje X se colocaría los días y en el Y el valor promedio de temperatura en cada fecha. Además habría una línea por invernadero.

Para el segundo punto usaría un gráfico de dispersión porque este es útil para mostrar la correlación entre dos variables, en este caso la temperatura y la humedad. En el eje X se colocaría la temperatura y sobre el eje Y la humedad.

Tablas importantes para el primer punto:

**Medicion**(#medicion, fecha, temperatura, humedad, #invernadero)

**Invernadero**(#invernadero, nombre, ubicacion)

Las tablas Medicion e Invernadero son relevantes para el análisis, ya que la primera (Medicion) contiene los datos necesarios (atributos fecha y temperatura) para calcular la temperatura promedio por día de cada semana, mientras que la segunda (Invernadero) permite identificar el invernadero específico (atributos #invernadero y nombre) sobre el cual se realizará el análisis. Ambas se relacionan a través del campo #invernadero, que vincula cada medición con el invernadero donde se realizó.

Tablas importantes para el segundo punto:

**Medicion**(#medicion, fecha, temperatura, humedad, #invernadero)

Sólo la tabla Medición es importante para el análisis ya que proporciona los datos numéricos (atributos fecha, temperatura y humedad) necesarios para calcular la relación entre temperatura y humedad en un mismo día.