

Alumno: \_\_\_\_\_

Apellido y Nombre: \_\_\_\_\_

## BBDD1 Parcial

2021-2022 27/11/2024

Uso interno de la cátedra

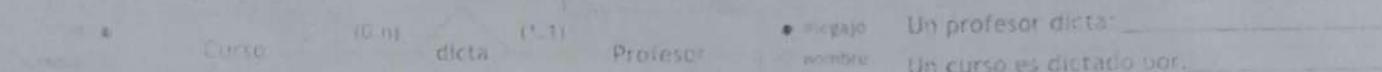
ER	Norm	AR	MySQL	Visualización

Corigió: \_\_\_\_\_

### Entidad/Relación

Indique cómo interpreta las cardinalidades del siguiente modelo

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



### 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; 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 pueda 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 debe saber de cuál de los viñedos fue extraído 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 reseñado 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.

```

Res = x
Mientras Res cambia
  Para i= 1 to cant_de_ particiones_realizadas
    Res = Res U ((Res ∩ Ri) * ∩ Ri)
  
```

### 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
  
```

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 categorías 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, #oferente, #subasta, fecha\_inscripción)  
**Oferente** (#oferente, nombre, apellido, cuil, localidad)  
**Oferta** (#oferta, #subasta, monto, fecha\_oferta)  
**Oferta\_Historial** (#oferta\_historial, #oferente, #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 eliminadora.