Diagrama

Descripción generada automáticamente**Caso 2:** : **Perú Service Summit**

**LINK:** [**https://my.vertabelo.com/public-model-view/kxjj9Tz644Mg72Gr5JoxurSFXPMg5jncuEf6ToLGkSp2M6nVTbvXiKFZzuvZB7Lx?x=1956&y=2369&zoom=0.3885**](https://my.vertabelo.com/public-model-view/kxjj9Tz644Mg72Gr5JoxurSFXPMg5jncuEf6ToLGkSp2M6nVTbvXiKFZzuvZB7Lx?x=1956&y=2369&zoom=0.3885)

**Relación Empresa-Disponibilidad**

Cada Empresa cuenta con una disponibilidad, para poderse realizar las reuniones, y tienen diferentes fechas disponibles, por ello la relación es de 1 a 0 o muchos, porque puede tener disponibilidad o no tenerla.

**Relación Reunión-OportunidadNegocio**

Por cada reunión se tiene oportunidades de negocio, pero no siempre es así, porque ambas partes pueden tener desacuerdos y no lograrse ni una negociación, por ello la relación es de 1 reunión a 0 o muchas oportunidades de negocio.

**Tabla Reunión**

La tabla Reunión es sencilla, porque cuenta con el dato realizacion que nos permitirá saber si se realizo la reunión o no se realizó la reunión.

**Caso 2: Películas**

Pregunta 2 (2 p.).

Crear un procedimiento almacenado o función que retorne la cantidad de actores que participaron en películas de un determinado género (ingresado como parámetro) para un determinado año (ingresado como parámetro).

create function count\_actor\_por\_genreanio(@genero varchar(20), @anio int) returns int

begin

declare @quantity int

select @quantity = count(a.id)

from actors a

join movie\_cast mc on a.id = mc.actor\_id

join movies m on mc.movie\_id = m.id

join genres g on m.id= g.id

where description=@genero and year=@anio

return @quantity

end;

select dbo.count\_actor\_por\_genreanio('Ciencia ficción', 2000) as 'Cantidad Actores' from actors

Pregunta 3 (2 p.).

Crear un procedimiento almacenado o función que retorne el nombre del actor (o actores) que participó más veces en películas de un determinado género (ingresado como parámetro) para un determinado año (ingresado como parámetro).

create function f\_actor\_per\_movie\_genre\_anio(@genero varchar(20), @anio int) returns table

as return

(select name, count(a.id) as cantidad

from actors a

join movie\_cast mc on a.id=mc.actor\_id

join movies m on mc.movie\_id=m.id

join genres g on m.id=g.id

where description=@genero and year=@anio

group by name

);

create function f\_max\_actor\_per\_genre\_anio(@genero varchar(20), @anio int) returns table

as return

(

select name from dbo.f\_actor\_per\_movie\_genre\_anio(@genero, @anio)

where cantidad = (

select max(cantidad) from dbo.f\_actor\_per\_movie\_genre\_anio(@genero, @anio)))

select \* from dbo.f\_max\_actor\_per\_genre\_anio('Ciencia ficción', 2000)

**Caso 3: Airbnb**

db.createCollection("Apartamento",{

validator:{

$jsonSchema:

{

bsonType: "object",

description: "Information about the Apartaments",

required: ["nombre\_anfitrión","Elementos","nombre\_apartamento","Valoración","Nro\_Reseñas"],

properties:{

nombre\_anfitrión:{

bsonType:"string",

description:"El nombre del anfitrión es un campo string y es obligatorio"

},

Elementos:{

bsonType: "array",

description: "Los elementos que cuenta el apartamento es un array y es obligatorio",

minItems: 3,

items:{

bsonType: "string",

description: "Cada elemento que cuenta el apartamento es un campo string y es obligatorio"

}

},

nombre\_apartamento:{

bsonType:"string",

description:"El nombre del apartamento es un campo string y es obligatorio"

},

Valoración:{

bsonType:"number",

minimum:0.0,

maximum:5.0,

description:"La valoración debe ser un decimal y es obligatorio"

},

Nro\_Reseñas:{

bsonType:"int",

description:"El número de reseñas es un entero y es obligatorio"

}

}

}

}

}

)

**Caso 4: Inspecciones**

Pregunta 5 (2 p.).

Escribir una consulta que permita mostrar la cantidad de inspecciones realizadas por cada ciudad.

db.Inspecciones.aggregate(

[{$group:{"address.city":{}, quantity: {$count:{}}}}]

)

Pregunta 6 (2 p.).

Escribir una consulta que permita mostrar la cantidad de inspecciones realizadas por cada resultado obtenido solo para la ciudad de NEW YORK.

db.Inspecciones.find({"address.city":"NEW YORK"}).count()