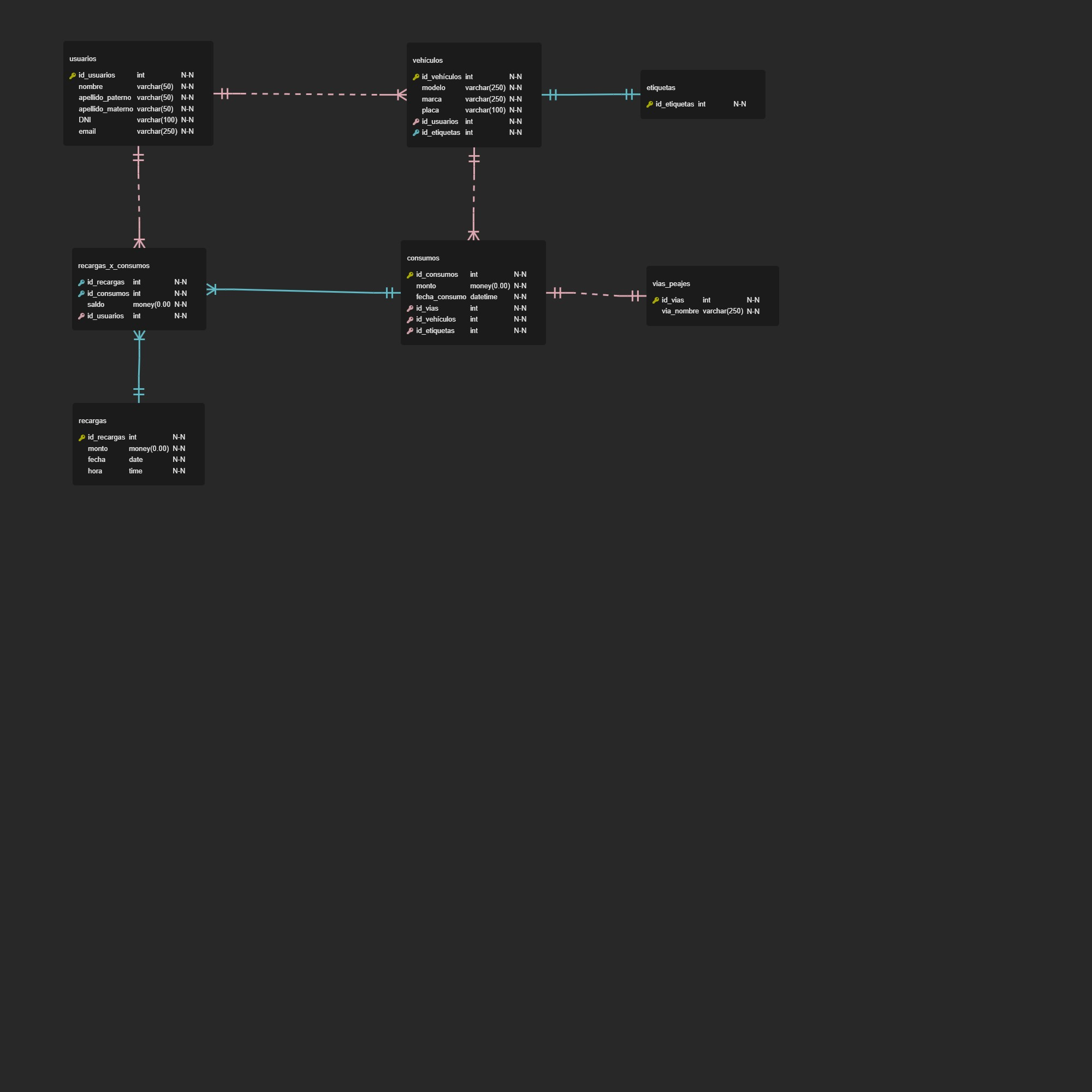
**Caso1: PexPerú**

--pregunta 2

CREATE FUNCTION mostrar\_datos\_usuario()

RETURNS TABLE

AS

RETURN(

SELECT

e.apellido\_paterno, e.apellido\_materno as 'nombre'

c.carrera\_nombre as 'carrera'

FROM estudiantes e

JOIN carreras c ON e.codigo\_carrera=c.codigo

);

SELECT \* FROM mostrar\_datos\_usuario();

--pregunta 3

CREATE FUNCTION mostrar\_cantidad\_estudiantes\_x\_carrera()

RETURNS TABLE

AS

RETURN(

SELECT

c.nombre as 'carrera', COUNT(\*) as cantidad

FROM estudiantes e

JOIN carreras c ON e.codigo\_carrera=c.codigo

GROUP BY c.nombre

);

SELECT \* FROM mostrar\_cantidad\_estudiantes\_x\_carrera();

--pregunta 4

create procedure calcular\_destinarios

as

begin

SELECT

c.codigo\_conversacion,

p.nombre as destinatario,

COUNT(p.nombre) as cantidad\_destinatarios

FROM destinatarios\_correos dc

JOIN correos c ON dc.codigo\_correo = c.codigo

JOIN personas p ON dc.codigo\_destinatario = p.codigo

GROUP BY

c.codigo\_conversacion,

p.nombre

ORDER BY c.codigo\_conversacion;

END;

exec calcular\_destinarios

--pregunta 5

CREATE PROCEDURE spammers

AS

BEGIN

SELECT

p.codigo as codigo\_spammer,

p.nombre as nombre\_spammer

FROM personas p

JOIN correos c ON p.codigo=c.codigo\_remitente

JOIN destinatarios\_correos dc ON c.codigo = dc.codigo\_correo

GROUP BY p.codigo, p.nombre

HAVING

COUNT(c.codigo\_conversacion)=

(

SELECT max(total\_conversaciones)

FROM (

SELECT p.codigo, COUNT(c.codigo\_conversacion) as total\_conversaciones

FROM personas p

JOIN correos c ON p.codigo=c.codigo\_remitente

JOIN destinatarios\_correos dc ON c.codigo dc.codigo\_correo

GROUP BY p.codigo

) as conversaciones\_por\_persona

)

end;

exec spammers

--pregunta 6

create function mas\_correos ()

returns table

return

( Select p.codigo

from ( SELECT c.codigo, count(c.codigo) as cantidad\_correos

FROM correos c

join destinarios\_correos dc on dc.codigo\_correo = c.codigo\_conversacion

group by c.codigo ) w

where w.cantidad = (select max(cantidad) from

( SELECT dc.codigo, count(c.codigo\_remitente) as cantidad

FROM destinatarios\_correos dc

join codigo\_correo dc on dc.codigo\_destiantario = c.codigo

where dc.codigo\_destinario

group by c.codigo ) w)

)

select \* from mas\_correos

pregunta7

db.createCollection("Reserva\_Espacio\_Estacionamiento", {

validator:

{

$jsonSchema: {

bsonType: 'object',

title: 'Coleccion Reserva\_Espacio\_Estacionamiento',

required: [

'reserva\_espacio\_estacionamiento\_id',

'usuario',

'espacio\_estacionamiento',

'fecha\_reserva',

'fecha\_inicio',

'hora\_reserva',

'hora\_inicio',

'tipo\_vehiculo',

'placa',

'marca',

'modelo',

],

properties: {

reserva\_espacio\_estacionamiento\_id: {

bsonType: 'int',

description: 'Debe ser un int es obligatorio'

},

usuario: {

bsonType: 'object',

required: [“nombre”, “apellido\_paterno”, “apellido\_materno”, “email”],

properties: {

nombre: {

bsonType: “string”

}

}

},

fecha\_reserva: {

bsonType: 'date',

description: 'Debe ser un date y es obligatorio'

},

fecha\_inicio: {

bsonType: 'date',

description: 'Debe ser un date y es obligatorio'

},

hora\_reserva: {

bsonType: 'time',

description: 'Debe ser un time y es obligatorio'

},

tipo\_vehiculo: {

bsonType: 'string',

description: 'Debe ser un string y es obligatorio'

},

hora\_inicio: {

bsonType: 'time',

description: 'Debe ser un time y es obligatorio'

},

placa: {

bsonType: 'string',

description: 'Debe ser un string es obligatorio'

},

marca: {

bsonType: 'string',

description: 'Debe ser un string y es obligatorio'

},

modelo: {

bsonType: 'date',

description: 'Debe ser un date y es obligatorio'

},

}

}

}

}

})

Pregunta 8

--Model one-to-one relationships with embedded documents

--Solo un usuario puede realizar una reserva

-- Documento embebido: En el mismo documento que representa una reserva de estacionamiento se tiene el campo [usuario] el cual es un objeto que representa a un usuario.

--Model one-to-one relationships with embedded documents

--Una reserva solo se puede hacer en un estacionamiento

-- Documento embebido: En el mismo documento que representa una reserva de estacionamiento se tiene el campo [espacio\_estacionamiento] el cual es un objeto que representa a un espacio de un estacionamiento.

Pregunta 9

db.restaurant.aggregate([{ $group: { \_id: "city", quantity: {$count :{}} }}])



Pregunta 10

db.restaurant.aggregate([{$match:{"cuisine":"Bakery"}},{$group: { \_id: "city", quantity: {$count :{}} }}])

