

#A)

```
SELECT *
FROM Oficina INNER JOIN Direccion ON Oficina.idDireccion = Direccion.id
      INNER JOIN Municipio ON Direccion.nombreMunicipio =
Municipio.nombre
WHERE nombreCalle LIKE "Paseo de %"
AND nombreMunicipio = "Alcobendas"
AND nombreProvincia = "Madrid"
;
```

#B)

```
SELECT Cartero.nombre, Cartero.apellidos, Reparto.matriculaCoche, Paquete.peso
FROM Cartero, Reparto, Paquete
WHERE Cartero.dni=Reparto.dniCartero
AND Reparto.id=Paquete.idReparto
AND Paquete.peso=(SELECT MIN(Paquete.peso)
      FROM Paquete);
```

#C)

```
SELECT nombre, apellido, "CARTA" AS TIPO FROM UsuarioGenerico
WHERE idDireccion IN(SELECT Direccion.id FROM Direccion
      INNER JOIN (SELECT nombreCalle, numInicio,
numFinal FROM SegmentoDeCalle
      INNER JOIN AreaIncluyeSegmento
      ON
SegmentoDeCalle.numSegmento=AreaIncluyeSegmento.numSegmento
      WHERE idArea="AR-MAD-03")TABLA_2
      ON Direccion.nombreCalle=TABLA_2.nombreCalle
      WHERE Direccion.numero BETWEEN
TABLA_2.numInicio AND TABLA_2.numFinal)
      AND UsuarioGenerico.id IN(SELECT idReceptor FROM Carta)
UNION
SELECT nombre, apellidos, "CARTA CERTIFICADA" FROM UsuarioCertificado
WHERE idDireccion IN(SELECT Direccion.id FROM Direccion
      INNER JOIN (SELECT nombreCalle, numInicio,
numFinal FROM SegmentoDeCalle
      INNER JOIN AreaIncluyeSegmento
      ON
SegmentoDeCalle.numSegmento=AreaIncluyeSegmento.numSegmento
      WHERE idArea="AR-MAD-03")TABLA_2
      ON Direccion.nombreCalle=TABLA_2.nombreCalle
      WHERE Direccion.numero BETWEEN
TABLA_2.numInicio AND TABLA_2.numFinal)
      AND dni IN(SELECT dniReceptor FROM CartaCertificada);
```

#D)

```
SELECT COUNT(distinct Reparto.id) AS numRepartos, codOficina, Cartero.dni,  
Cartero.nombre, Cartero.apellidos  
FROM Trabaja INNER JOIN Cartero on Cartero.dni=Trabaja.dniCartero INNER JOIN  
Reparto ON Cartero.dni=Reparto.dniCartero  
Group by codOficina, dni, Cartero.nombre, Cartero.apellidos  
Order by COUNT(distinct Reparto.id) DESC  
;
```

#E)

```
SELECT SUM(Paquete.peso) AS PesoTotal,  
MAX(CartaCertificada.nivelUrgencia) AS MaximoNivelUrgencia  
FROM Reparto  
    INNER JOIN Paquete ON Paquete.idReparto = Reparto.id  
        INNER JOIN CartaCertificada ON CartaCertificada.idReparto = Reparto.id  
            INNER JOIN Coche ON Coche.matricula = Reparto.matriculaCoche  
WHERE Coche.codOficina = "OF-MAD-01"  
AND Reparto.fecha BETWEEN '2022-11-13' AND '2022-11-20'  
#F)
```

DELIMITER \$\$

```
CREATE PROCEDURE aumCoches()  
BEGIN
```

```
UPDATE Coche SET capacidad = capacidad * 1.1  
WHERE Coche.matricula IN (SELECT Reparto.matriculaCoche
```

```
FROM Reparto
```

```
WHERE Reparto.id IN ( SELECT DISTINCT idReparto
```

```
FROM Carta)
```

```
AND Reparto.id NOT IN ( SELECT DISTINCT idReparto
```

```
FROM CartaCertificada)
```

```
AND Reparto.id NOT IN ( SELECT DISTINCT idReparto
```

```
FROM Paquete));
```

```
END $$
```

```
DELIMITER ;
```

```
CALL aumCoches();
```

#G)

```
SELECT *
FROM Cartero C
WHERE NOT EXISTS (SELECT Recogida.idDireccion, sum(Recogida.dniCartero)
                  FROM Recogida
                  WHERE C.dni = Recogida.dniCartero
                  group by Recogida.idDireccion
                  HAVING COUNT(*) > 1)
;
```

#H)

```
SELECT id AS AREA, codOficina
FROM AreaDeEnvio
WHERE AreaDeEnvio.codOficina IN(SELECT Oficina.cod
                                FROM Oficina
                                WHERE Oficina.cod IN (SELECT
Turno.codOficina
FROM Turno
WHERE Turno.jornada NOT IN (SELECT Trabaja.jornadaTurno
FROM Trabaja)))
```

#I)

```
SELECT idRuta AS id
FROM Reparto
WHERE idRuta IN ( SELECT idRuta
                  FROM RutaIncluyeSegmento
                  WHERE nombreCalle="Avenida de América" AND
nombreMunicipio="Alcorcón"
                  group by idRuta
                  HAVING COUNT(idRuta)=( SELECT COUNT(*)
                                          FROM
SegmentoDeCalle
                                          WHERE
nombreCalle="Avenida de América" AND nombreMunicipio="Alcorcón"))
GROUP BY idRuta
HAVING COUNT(idRuta)>2;
```

#FUNCIONES Y PROCEDURES

#a)

DELIMITER \$\$

CREATE PROCEDURE devDNI(IN ids VARCHAR(13))

BEGIN

SELECT CONCAT(EntregaCertificada.dniUsuario, "#", UsuarioCertificado.dniAutorizado)
AS "dniR#dniA"

FROM UsuarioCertificado INNER JOIN EntregaCertificada ON

EntregaCertificada.dniUsuario = UsuarioCertificado.dni

WHERE EntregaCertificada.idCarta = ids;

END \$\$

DELIMITER ;

CALL devDNI("CE01234567890");

#B)

DROP FUNCTION IF EXISTS mejorRuta ;

DELIMITER \$\$

CREATE FUNCTION mejorRuta(reparto_id INTEGER)

RETURNS VARCHAR(9)

DETERMINISTIC

BEGIN

DECLARE terminado_rutas INT DEFAULT 0;

DECLARE b VARCHAR(9);

DECLARE maxRepartos INTEGER DEFAULT 0;

DECLARE numRepartos INTEGER DEFAULT 0;

DECLARE rutaOptima VARCHAR(9);

DECLARE rutas CURSOR FOR SELECT id FROM Ruta;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET terminado_rutas = 1;

OPEN rutas;

read_loop: LOOP

FETCH rutas INTO b;

IF terminado_rutas = 1 THEN

LEAVE read_loop;

END IF;

SET numRepartos = (

select Sum(AMOUNT)

from(SELECT numSegmento, COUNT(*) AS AMOUNT

FROM (SELECT numSegmento, calle

FROM SegmentoDeCalle , (select Direccion.nombreCalle,

Direccion.nombreMunicipio, calle from Direccion, (

```

                                select Carta.id, UsuarioGenerico.idDireccion as calle
from Carta inner join UsuarioGenerico on UsuarioGenerico.id = Carta.idReceptor
                                where idReparto = reparto_id
                                union
                                select Paquete.id, UsuarioGenerico.idDireccion as calle
from Paquete inner join UsuarioGenerico on UsuarioGenerico.id = Paquete.idReceptor
                                where idReparto = reparto_id
                                union
                                select id, UsuarioCertificado.idDireccion as calle from
CartaCertificada inner join
                                UsuarioCertificado on UsuarioCertificado.dni =
CartaCertificada.dniReceptor
                                where idReparto = reparto_id) as tabla2 where
Direccion.id = tabla2.calle) as table3
                                WHERE SegmentoDeCalle.nombreCalle = table3.nombreCalle AND
SegmentoDeCalle.nombreMunicipio = table3.nombreMunicipio
                                ) AS tabla2
                                WHERE numSegmento IN (
                                SELECT numSegmento
                                FROM RutaIncluyeSegmento
                                WHERE idRuta = b
                                )
                                group by numSegmento
                                ) AS tabla3
                                );

                                IF numRepartos > maxRepartos THEN
                                SET maxRepartos = numRepartos;
                                SET rutaOptima = b;
                                END IF;

                                END LOOP read_loop;
                                CLOSE rutas;

                                RETURN(rutaOptima);
                                END $$
                                DELIMITER ;

                                SELECT mejorRuta(1);

```

#TRIGGERS

#A)

```
DROP TRIGGER inCheck;
DELIMITER //
CREATE TRIGGER inCheck BEFORE INSERT ON Paquete
FOR EACH ROW
BEGIN
    DECLARE numerosID DOUBLE;
    SET numerosID = SUBSTRING(NEW.id, 3, 12);
    IF(NEW.id NOT LIKE 'PQ%')
        THEN SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT="El id del paquete es
erroneo";
    END IF;
    IF(numerosID REGEXP '[0-9]+$' = 0) OR LENGTH(numerosID) <> 10
        THEN SIGNAL SQLSTATE '02500' SET MESSAGE_TEXT="Los numeros del id del
paquete es erroneo";
    END IF;
    IF(NEW.ancho NOT BETWEEN 5 AND 25)
        THEN SIGNAL SQLSTATE '03000' SET MESSAGE_TEXT="El ancho del paquete son
erroneo";
    END IF;
    IF(NEW.alto NOT BETWEEN 5 AND 25)
        THEN SIGNAL SQLSTATE '03500' SET MESSAGE_TEXT="El alto del paquete son
erroneo";
    END IF;
    IF(NEW.peso NOT BETWEEN 1 AND 100)
        THEN SIGNAL SQLSTATE '04000' SET MESSAGE_TEXT="El peso del paquete es
erroneo";
    END IF;
END //
DELIMITER ;
```

#B)

```
DELIMITER $$
CREATE TRIGGER up_check_PESO AFTER INSERT ON Paquete FOR EACH ROW
BEGIN
    DECLARE DNI VARCHAR(9);
    IF(SELECT SUM(Paquete.peso) FROM Paquete WHERE NEW.idReparto =
Paquete.idReparto)>= 500
    THEN

        SET DNI = (SELECT Trabaja.dniCartero FROM Trabaja
                    INNER JOIN Oficina ON Trabaja.codOficina = Oficina.cod
                    INNER JOIN Coche ON Oficina.cod = Coche.codOficina
                    INNER JOIN Reparto ON Coche.matricula = Reparto.matriculaCoche
                    WHERE Reparto.id = NEW.idReparto
```

```
And Trabaja.dniCartero NOT IN (SELECT dniCartero FROM
Reparto WHERE Reparto.fecha != (SELECT Reparto.fecha FROM Reparto WHERE
Reparto.id = NEW.idReparto))
```

```
AND Trabaja.dniCartero IN (SELECT dniCartero FROM Trabaja INNER JOIN
Turno ON Trabaja.jornadaTurno = Turno.jornada
```

```
WHERE horarioEntrada <= CURRENT_TIME AND horarioSalida >=
CURRENT_TIME));
```

```
UPDATE Reparto SET dniCartero = DNI WHERE Reparto.id =
NEW.idReparto;
```

```
END if;
```

```
END $$
```

```
DELIMITER ;
```