

Evaluación: JUNIO I (eval 2)	Fecha:30-5-24	Calificación	IES SEGUNDO DE CHOMÓN
Módulo/Materia: Bases de Datos	Curso:1º DAM		
Departamento: INFORMÁTICA			
Nombre y apellidos:			

**Funciones extra que necesitaréis en alguna consulta:**

**TIMESTAMPDIFF(MINUTE, fechaInicial, fechaFinal)** devuelve la diferencia en minutos de la fechaFinal menos la fechaInicial.

Debéis entregar un documento pdf llamado **Tunombre\_Ex\_BD\_JunioI\_eval2** con capturas en las que se vea el código de la sentencia sql y la tabla resultante al ejecutarla.

1.- Pedro Pérez tiene previsto salir tarde del trabajo hoy viernes, llegará a casa a las 12 de la noche, y quiere pedir la cena a algún restaurante que atienda su zona, realiza la consulta para ver en que restaurantes puede pedir la cena (nombre y dirección completa) y la hora de cierre de los mismos (1.5 puntos)

```
mysql> select restaurantes.nombre, restaurantes.calle, restaurantes.codigo_postal, horarios.hora_cierre from restaurantes join horarios on restaurantes.codigo = horarios.codigo join areascobertura on restaurantes.codigo = areascobertura.codigo where dia_semana = "V" and hora_cierre > "00:00:00" and codigoPostal = (select codigoPostal from clientes where nombre = "pedro" and apellido = "perez");
```

nombre	calle	codigo_postal	hora_cierre
pizzahud	abascal 45	12345	03:00:00
Templo hindú	carbón 22	12567	03:00:00

2 rows in set (0.00 sec)

2. Como puedes observar en las tablas, en un mismo pedido se pueden pedir platos a más de un restaurante, por tanto, nos interesa saber: qué pedidos de los que han sido entregados han tenido menos de 30 minutos de espera, el número de variedades de platos distintos que se han pedido y a cuántos restaurantes. (2 puntos)

```
mysql> select pedidos.codigo, count(distinct contiene.plato) as variedadPlato, count(distinct contiene.restaura) from pedidos join contiene on pedidos.codigo = contiene.codigo where pedidos.estado = "entregado" group by pedidos.codigo and timestampdiff(minute, fecha_hora_pedido, fecha_hora_entrega) < 30;
```

codigo	variedadPlato	count(distinct contiene.restaura)
2	2	2
3	4	2

2 rows in set (0.01 sec)

3. Muestra un listado con el código y nombre del restaurante y el número de personas que le han realizado un pedido (no importa el estado del pedido) si algún restaurante no ha tenido clientes también debe aparecer. (USA OBLIGATORIAMENTE JOIN) (1.75 puntos)

```
mysql> select restaurantes.codigo, restaurantes.nombre, count(distinct pedidos.cliente) from restaurantes left join platos on restaurantes.codigo = platos.codigo left join contiene on platos.nombreplato = contiene.plato left join pedidos on contiene.codigo = pedidos.codigo group by restaurantes.codigo;
```

codigo	nombre	count(distinct pedidos.cliente)
1234	pizzahud	2
2345	Templo hindú	2
3456	burger-burger	3
4444	teruel	0
5677	rissoto	0
5678	telericatorta	2

6 rows in set (0.00 sec)

4- ¿Cuál es el plato más caro, del pedido más caro, del cliente con Dni '45678901Y', de los que no han sido cancelados ni rechazados? Di también su precio y que restaurante lo ha servido (1.75 puntos)

```
mysql> select pedidos.codigo, platos.nombrePlato, platos.precio, restaurantes.nombre as restaurante, pedidos.importe_total from restaurantes join platos on restaurantes.codigo = platos.rescod join contiene on platos.nombrePlato = contiene.plato and platos.rescod = contiene.restaura join pedidos on contiene.pedido = pedidos.codigo where cliente = "45678901Y" and pedidos.estado <> "cancel" and pedidos.estado <> "rechazado" group by platos.precio having importe_total >= (select max(importe_total) from pedidos where pedidos.codigo = (select codigo from pedidos where cliente = "45678901Y" and importe_total = (select max(importe_total) from pedidos where cliente = "45678901Y")) and platos.precio >= (select max(platos.precio) from platos join contiene on platos.nombrePlato = contiene.plato join pedidos on contiene.pedido = pedidos.codigo where cliente = "45678901Y" and platos.precio = (select max(platos.precio) from platos join contiene on platos.nombrePlato = contiene.plato join pedidos on contiene.pedido = pedidos.codigo and codigo = (select codigo from pedidos where cliente = "45678901Y" and importe_total = (select max(importe_total) from pedidos where cliente = "45678901Y"))));
```

codigo	nombrePlato	precio	restaurante	importe_total
9	pizza vegetal	10.50	pizzahud	45.36

1 row in set (0.00 sec)

5- Muestra una tabla con el número de pedido, su importe total, de cuantos platos totales está compuesto el pedido y su precio medio por plato de los pedidos que no han sido cancelados ni rechazados que tengan más de un plato y cuyo precio medio por plato esté entre los 8 y los 20 euros. (1.75 puntos)

```
mysql> select pedidos.codigo, pedidos.importe_total, sum(contiene.unidades) as numeroPlatos, sum(platos.precio*contiene.unidades)/sum(contiene.unidades) as mediaPrecio from pedidos join contiene on pedidos.codigo = contiene.pedido join platos on platos.nombrePlato = contiene.plato where pedidos.estado not in ("rechazado", "cancel") group by pedidos.codigo having numeroPlatos > 1 and mediaPrecio > 8 and mediaPrecio < 20;
```

codigo	importe_total	numeroPlatos	mediaPrecio
2	48.30	3	15.666667
9	45.36	5	8.800000

2 rows in set (0.00 sec)

6.- Muestra en cuantos restaurantes pueden cenar cada día los clientes de cada área de cobertura de cada restaurante. El resultado deben ser dos columnas una con el área de cobertura seguido del día de la semana y otra con el número de restaurantes. Debe estar ordenado por el área de cobertura. (1.25 punto)

areacobertura_por_dia	numrestaurantes
CP-dia	8

```
mysql> select concat(areascobertura.codigopostal, " ", horarios.dia_semana) as area_cobertura_por_dia, count(distinct restaurantes.codigo) as numRest from clientes join pedidos on clientes.dni = pedidos.cliente join contiene on pedidos.codigo = contiene.pedido join platos on contiene.plato = platos.nombrePlato join restaurantes on platos.resCod = restaurantes.codigo join horarios on restaurantes.codigo = horarios.codres join areascobertura on restaurantes.codigo = areascobertura.codigores group by area_cobertura_por_dia;
```

area_cobertura_por_dia	numRest
12345 D	1
12568 D	1
28224 D	1
12345 J	2
12346 J	1
12568 J	1
28224 J	1
12345 S	2
12346 S	1
12568 S	1
28224 S	1
12345 V	2
12346 V	1
12568 V	1
28224 V	1
12345 X	1
12346 X	1

17 rows in set (0.00 sec)