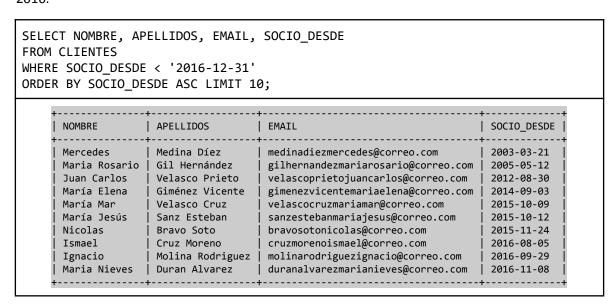
Sentencias SQL

Sentencia 1

El departamento de márketing ha decidido premiar a los primeros clientes que confiaron en la empresa para adquirir sus libros.

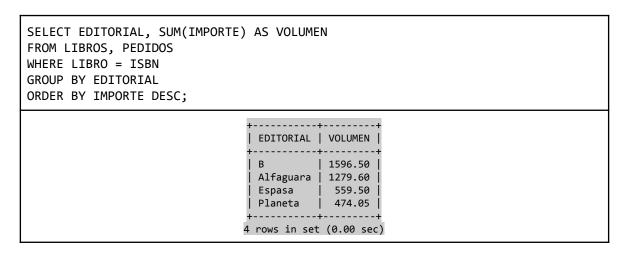
Nos solicitan una lista con los nombres, apellidos y correos electrónicos (se les enviará un cupón) de los primeros 10 clientes que se hicieron socios. Pero que su fecha de asociación no sea posterior a 2016.



Sentencia 2

El departamento de Administación desea cerrar el año de 2020 y tiene que saber cuánto pagar a las editoriales.

Se nos solicita obtener el detalle ordenado por el volumen de las ventas, desglosado por editorial de todas las realizadas en el ejercicio del año 2020.



El departamento de Administación tiene una fuerza laboral limitada (tanto como para no poder aplicar un simple porcentaje). Por ello, se nos pide, además del volumen, que calculemos el 60% y el 40% de este. Siendo el 60% lo que se debe abonar a la editorial y el 40% el margen para la empresa. Para facilitarlo se creará una **vista** y operaremos sobre esta.

CREATE VIEW VENTAS_2020 AS SELECT EDITORIAL, SUM(IMPORTE) AS VOLUMEN FROM LIBROS, PEDIDOS WHERE LIBRO = ISBN GROUP BY EDITORIAL ORDER BY VOLUMEN DESC;	SELECT EDITORIAL, VOLUMEN, VOLUMEN*0.60 AS 'A ABONAR', VOLUMEN*0.40 AS MARGEN FROM VENTAS_2020;
++ EDITORIAL IMPORTE ++ B	EDITORIAL VOLUMEN A ABONAR MARGEN +

Sentencia 4

El departamento de Administración no deja de darnos trabajo (por algo nos pagan, ¿no?). Desean saber qué cliente ha hecho más compras y cuál ha gastado más en la empresa.

SELECT CLIENTE, COUNT(*) AS 'CANTIDAD PEDIDOS' FROM CLIENTES, PEDIDOS WHERE PEDIDOS.CLIENTE = CLIENTES.DNI_CLI GROUP BY CLIENTE ORDER BY 2 DESC LIMIT 1;	SELECT CLIENTE, SUM(IMPORTE) AS 'IMPORTE TOTAL PEDIDOS' FROM CLIENTES, PEDIDOS WHERE PEDIDOS.CLIENTE = CLIENTES.DNI_CLI GROUP BY CLIENTE ORDER BY 2 DESC LIMIT 1;
CLIENTE CANTIDAD PEDIDOS 	++ CLIENTE IMPORTE TOTAL PEDIDOS ++ 90692199W 355.85 ++ 1 row in set (0.00 sec)

Nos informan que no desean saber los DNIs de los clientes sino el resto de sus datos.

Para la persona que ha realizado más pedidos:

Para la persona que ha gastado más en la tienda:

Sentencia 6

Se desea tener en la tabla TIENDAS la suma de las ventas que han realizado todos los trabajadores de dicha tienda.

En el departamento de IT estamos hasta arriba de trabajo y la empresa está en crecimiento continuo. Por lo que no podemos realizar cambios en consultas de forma manual.

```
DELIMITER $$
DROP PROCEDURE IF EXISTS ACTUALIZA_VENTAS;
CREATE PROCEDURE ACTUALIZA_VENTAS()
BEGIN

DECLARE contador INT DEFAULT 0;
DECLARE N_T INT DEFAULT 0;
SELECT COUNT(*) FROM TIENDAS INTO N_T;
```

```
WHILE (contador < N_T) DO
      SET contador = contador + 1;
      UPDATE TIENDAS
             SET VENTAS = (SELECT SUM(VENTAS) FROM TRABAJADORES WHERE OFICINA = contador)
             WHERE ID_OFI = contador;
      END WHILE;
END $$
SELECT * FROM TIENDAS;
                            ID_OFI | CIUDAD | REGION | VENTAS |
                                 1 | Madrid | Centro | 10308 |
2 | Lugo | Oeste | 4581 |
3 | Madrid | Centro | 3897 |
                                  4 | Avila
                                                | Centro |
                                                             7256
                                                            7978
                                  5 | Burgos
                                               | Norte |
                                                           2952
                                  6 | Aviles
                                                | Norte |
                                  7 | Madrid
                                                | Centro | 5707
                                  8 | Barcelona | Este |
                                                             4717
                                  9 | Barcelona | Este |
                                                             2237
                                 ---+-----
                                   9 rows in set (0.00 sec)
```

Indicar las veces que ha asistido cada cliente a eventos. En la tabla se deben incluir aquellos que no han acudido a ningún evento.

Nótese que se han eliminado los registros de dos clientes para la correcta ejemplificación del LEFT JOIN: DELETE FROM ASISTENCIA WHERE CLIENTE = '32409106J'; DELETE FROM ASISTENCIA WHERE CLIENTE = '91914347E'; SELECT C.DNI_CLI, COUNT(A.EVENTO) FROM CLIENTES AS C LEFT JOIN ASISTENCIA AS A ON CLIENTE = DNI CLI GROUP BY C.DNI_CLI; 26729071N | 8 | 29055113H | 11 | 76500479H | DNI_CLI | COUNT(A.EVENTO) | 77557848P 9 11 23485027A 57814348F | 11 91914347E 06579831Z 9 10304826K 12 32409106J 0 | 62353070R 90692199W | 12 11894766V 12 6 08306868G 18744587R 21103158Z 6 | 86695737C 00716407A | 12 10 | 85390221F 12 6 72978864W 56050769E 10 12 15676363T 37645572T 7 77622435B 40310832C 10 81678063A 13 20972038V 7 55820220W 10 | 63365370W | 13 | 7 | 10 05883847X 14 04825243G 25693246S 67414610T 8 26536647Y 10 61638432L | 14 11326330A 8 09707776M 10 62580495W 89716389J | 10 | 40 rows in set (0.00 sec)

Se nos ha indicado que en el evento del pasado 2020-12-25 se ha confirmado un caso positivo en los asistentes al evento. El cliente positivo tiene el DNI 81678063A. Obtener su asiento y sus datos de contacto.

Localizar los asistentes cercanos (+-3 puestos) y obtener también sus datos.

```
DELIMITER $$
      DROP PROCEDURE IF EXISTS CONTACTOS_COVID;
/*Creamos el procedimiento, solicitando los datos:
      - Cliente declarado positivo (DNI)
      - Fecha del evento al que asiste*/
      CREATE PROCEDURE CONTACTOS_COVID(VAR_contacto VARCHAR(10), VAR_fecha DATE)
      BEGIN
/*Declaramos variables a utilizar*/
             DECLARE VAR_id_evento INT DEFAULT 0;
             DECLARE VAR_butaca_confinadx INT DEFAULT 0;
/*Como me dan fecha, necesito saber el ID del evento que se ha celebrado en esa fecha*/
             SELECT ID_EVENTO FROM EVENTOS WHERE FECHA = VAR_fecha INTO VAR_id_evento;
/*Con el ID del evento y el DNI del cliente/asistente sacamos su butaca para ese evento. */
             SELECT BUTACA FROM ASISTENCIA WHERE CLIENTE = VAR_contacto AND EVENTO = VAR_id_evento
INTO VAR_butaca_confinadx;
/*Con los datos introducucidos y los obtenidos en los select anteriores, nos ponemos al lío*/
             SELECT A.CLIENTE, C.NOMBRE, C.APELLIDOS, C.EMAIL
             FROM CLIENTES AS C, ASISTENCIA AS A
             WHERE A.EVENTO = VAR_id_evento
/*¿Cogerá esto el cacharro?*/
             AND BUTACA IN ((VAR_butaca_confinadx - 3),
                          (VAR butaca confinadx - 2),
                          (VAR_butaca_confinadx - 1),
                          VAR_butaca_confinadx,
                          (VAR_butaca_confinadx + 1),
                          (VAR_butaca_confinadx + 2),
                          (VAR_butaca_confinadx + 3))
/*Esto es para unir las tablas, si no se nos rompe todo.*/
             AND C.DNI_CLI = A.CLIENTE;
      END$$
```

CALL CONTACTOS_COVID('55820220W','2020-12-25')\$\$

CLIENTE	NOMBRE	APELLIDOS	EMAIL
05883847X	Felix	Medina Vargas	medinavargasfelix@correo.com saezcastrojosecarlos@correo.com velascoprietojuancarlos@correo.com molinarodriguezignacio@correo.com rodriguezcanoencarnacion@correo.com saezcarmonafranciscojose@correo.com navarromartinezjoan@correo.com
15676363T	Jose Carlos	Saez Castro	
26536647Y	Juan Carlos	Velasco Prieto	
55820220W	Ignacio	Molina Rodriguez	
77557848P	Encarnacion	Rodriguez Cano	
86695737C	Francisco Jose	Saez Carmona	
89716389J	Joan	Navarro Martinez	

7 rows in set (0.05 sec)

En la tabla que contiene la información de los libros, existe una columna llamada *RESUMEN*, que contiene información relativa al contenido del libro. Se desea facilitar la agilizar al máximo las operaciones de búsqueda sobre esta columna de cara a que empleados y clientes/as puedan encontrar libros en base a sus intereses.

CREATE FULLTEXT INDEX i_libros_resumen ON LIBROS(RESUMEN);											
SHOW KEYS FROM LIBROS;											
++	++										
Table	Non_unique Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
LIBROS	0 PRIMARY	1	ISBN	A	10	NULL	NULL		BTREE		
LIBROS	1 AUTOR	1	AUTOR	A	4	NULL	NULL		BTREE	j i	İ
LIBROS	1 i_libros_resumen	1	RESUMEN	NULL	10	NULL	NULL	YES	FULLTEXT		
+		+		+	+				+	++	+
3 rows in set (0.00 sec)											

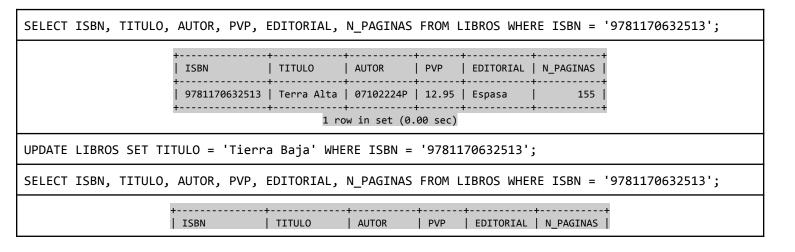
Sentencia 10

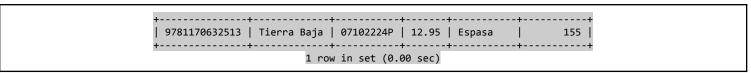
También se desea optimizar la búsqueda sobre los títulos de los libros, con el mismo motivo que para la consulta previa.

CREATE INDEX i_libros_titulo ON LIBROS(TITULO);											
SHOW KEYS FROM LIBROS;											
IOW RETS											
+											
Table No	on_unique Key_name	Seq_in_index	Column_name	Collation	Cardinality				Index_type		
LIBROS	0 PRIMARY	1	ISBN	A	10				BTREE		
LIBROS	1 AUTOR	1	AUTOR	A	4	NULL	NULL		BTREE	i i	į
LIBROS	1 i_libros_titulo	1	TITULO	A	10	NULL	NULL		BTREE	i i	İ
LIBROS	1 i_libros_resumen	1	RESUMEN	NULL	10	NULL	NULL	YES	FULLTEXT		
		+		rows in set	++					+	

Sentencia 11

Desde la editorial Espasa nos indican que había un error en el ISBN del libro llamado *Terra Alta*, pasa de tener este nombre a *Tierra Baja*, su ISBN es el 9781170632513





Por chanchullos con el cliente con el correo garridopascualsandra@correo.com, se debe aumentar en 5€ cada pedido realizado por este.

SELECT * FROM PEDIDOS WHERE CLIENTE = (SELECT DNI_CLI FROM CLIENTES WHERE EMAIL =
'garridopascualsandra@correo.com');

+	CLIENTE	VENDEDOR	LIBRO	FECHA	++ IMPORTE ++
•	37645572T 37645572T		9780566079677 9783757346857		20.90 19.90

SELECT * FROM PEDIDOS WHERE CLIENTE = (SELECT DNI_CLI FROM CLIENTES WHERE EMAIL =
'garridopascualsandra@correo.com');

N_PEDIDO	+	VENDEDOR		•	IMPORTE
34	37645572T 37645572T	12 7	9780566079677 9783757346857	2016-12-22 2017-02-16	25.90