

DAW/DAM. UD 6. MODELO FÍSICO DQL. ACTIVIDADES NO EVALUABLES. BOLETÍN A (SOLUCIONADO)

DAW/DAM. Bases de datos (BD)

UD 6. MODELO FÍSICO DQL

Boletín A. Prácticas no evaluables (solucionado)

Abelardo Martínez y Pau Miñana

Basado y modificado de Sergio Badal (www.sergiobadal.com) y Raquel Torres.

Curso 2023-2024

Aspectos a tener en cuenta

Importante

Estas actividades son opcionales y no evaluables pero es recomendable hacerlas para un mejor aprendizaje de la asignatura.

Si buscas las soluciones por Internet o preguntas al oráculo de ChatGPT, te estarás engañando a ti mismo. Ten en cuenta que **ChatGPT no es infalible ni todopoderoso.**

Es una gran herramienta para agilizar el trabajo una vez se domina una materia, pero usarlo como atajo en el momento de adquirir habilidades y conocimientos básicos perjudica gravemente tu aprendizaje. Si lo utilizas para obtener soluciones o asesoramiento respecto a las tuyas, revisa cuidadosamente las soluciones propuestas igualmente. Intenta resolver las actividades utilizando los recursos que hemos visto y la documentación extendida que encontrarás en el “Aula Virtual”.

Recomendaciones

Importante

- **No uses NUNCA tildes, ni eñes, ni espacios, ni caracteres no alfanuméricos** (salvo el guión bajo) **en los metadatos** (nombres de elementos de una base de datos).
- Sé coherente con el uso de mayúsculas/minúsculas.

1. BD Jardinería. Esquema

Disponemos del siguiente esquema de la BD o diseño físico, en el que se muestran las tablas que lo forman y cómo están relacionadas entre sí:



Dedícale unos minutos a revisar los nombres de las tablas, los campos que las forman y cómo están relacionadas entre ellas. Es fundamental conocer las tablas para realizar después las consultas de forma adecuada.

2. Creación de la BD

El siguiente paso será crear la base de datos junto con sus tablas correspondientes.

Actividad no evaluable

El SGBD a utilizar es MySQL. El archivo que contiene el *script* con la base de datos lo puedes descargar del Aula Virtual. Lo ejecutaremos y comprobaremos que se ha creado la base de datos y las tablas correctamente.

Base de datos:

```
mysql> CREATE DATABASE DBJardineria CHARACTER SET utf8 COLLATE utf8_spanish_ci;
Query OK, 1 row affected, 2 warnings (0,00 sec)

mysql>
mysql> USE DBJardineria;
Database changed
```

Tablas:

```
mysql> show tables;
+-----+
| Tables_in_DBJardineria |
+-----+
| Clientes                |
| DetallePedidos          |
| Empleados               |
| GamasProductos         |
| Oficinas                |
| Pagos                   |
| Pedidos                 |
| Productos               |
+-----+
8 rows in set (0,00 sec)
```

3. Consultas. Nivel básico

Actividad no evaluable

Realiza las siguientes consultas en **MySQL**.

3.1. Ejercicio

Mostrar las distintas ciudades en las que la empresa de jardinería tiene clientes. Fíjate en el resultado obtenido y si detectas errores de algún tipo, corrígelos (pista: hay dos errores).

Solución

```
mysql> SELECT DISTINCT ciudad  
-> FROM Clientes;
```

```
+-----+  
| ciudad |  
+-----+  
| San Francisco |  
| Miami |  
| New York |  
| Fuenlabrada |  
| Madrid |  
| San Lorenzo del Escorial |  
| Montornes del valles |  
| Santa cruz de Tenerife |  
| Barcelona |  
| Canarias |  
| Barcelona |  
| Sotogrande |  
| Humanes |  
| Getafe |  
| Fenlabrada |  
| Paris |  
| Sydney |  
| London |
```

```
+-----+
18 rows in set (0,01 sec)
```

Fíjate en el resultado, como puedes observar a pesar de haber utilizado la cláusula `distinct` aparecen dos Barcelona, pero si lo observas bien, verás que el segundo está un poco separado de la columna. Esto es un error habitual, cuando se ha tecleado el dato se ha incluido sin querer un espacio en blanco delante del nombre, con lo cual la base de datos lo está tomando como dos ciudades distintas. Normalmente estas cosas deben estar controladas por el programa que permite la introducción de los datos, pero evidentemente éstos no son perfectos y a veces ocurren fallos como éste. Si se observan anomalías de este tipo hay que corregirlas.

Para cambiar dicho registro y colocarlo adecuadamente tenemos varias opciones; nosotros vamos a realizar una de ellas, pero esperamos que a vosotros al menos se os ocurran un par de ellas más.

```
mysql> UPDATE Clientes
      -> SET Ciudad ='Barcelona'
      -> WHERE Ciudad = ' Barcelona';
Query OK, 1 row affected (0,01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Observa que en el filtro, después de la primera comilla, hemos dejado un espacio en blanco para reproducir el dato erróneo que hay en el campo. Si ahora vuelves a realizar la consulta anterior podrás observar que ya solo aparece una ciudad Barcelona.

Fíjate también que aparecen como ciudades Fuenlabrada y Fenlabrada, como es obvio, también es un error al teclear, se ha omitido una letra. Esto es más difícil de detectar en los programas si no hay una tabla de ciudades con la que comparar. Pues bien, los datos deben ser revisados de forma periódica para detectar estos errores humanos.

Vamos a cambiarlo con una sentencia *update*.

```
mysql> UPDATE Clientes
      -> SET Ciudad ='Fuenlabrada'
```



```
-> WHERE Ciudad = 'Fenlabrada';  
Query OK, 1 row affected (0,00 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```

3.2. Ejercicio

Mostrar cuántos clientes tenemos y mostrarlos en una columna denominada "Num_de_Clientes".

Solución

```
mysql> SELECT COUNT(*) AS Num_de_Clientes  
-> FROM Clientes;  
+-----+  
| Num_de_Clientes |  
+-----+  
|              36 |  
+-----+  
1 row in set (0,01 sec)
```

3.3. Ejercicio

Mostrar el nombre, la cantidad en almacén y el precio de compra (proveedor) de los productos de la gama "Herramientas" ordenado por el nombre del producto.

Solución

```
mysql> SELECT Nombre, CantidadEnStock, PrecioProveedor  
-> FROM Productos  
-> WHERE Gama = 'Herramientas'  
-> ORDER BY Nombre;
```

Nombre	CantidadEnStock	PrecioProveedor
Azadón	15	11.00
Pala	15	13.00
Rastrillo de Jardín	15	11.00
Sierra de Poda 400MM	15	11.00

```
4 rows in set (0,01 sec)
```

3.4. Ejercicio

Mostrar la valoración del almacén de cada producto de la gama "Herramientas" (en una columna denominada "Importe") ordenado por el importe obtenido y su nombre. (La valoración del almacén se realiza multiplicando los productos en *stock* por su precio de compra)

Solución

```
mysql> SELECT Nombre, (CantidadEnStock*PrecioProveedor) AS Importe
-> FROM Productos
-> WHERE Gama = 'Herramientas'
-> ORDER BY 2, Nombre;
```

```
+-----+-----+
| Nombre          | Importe |
+-----+-----+
| Azadón          | 165.00 |
| Rastrillo de Jardín | 165.00 |
| Sierra de Poda 400MM | 165.00 |
| Pala            | 195.00 |
+-----+-----+
4 rows in set (0,00 sec)
```

3.5. Ejercicio

Mostrar el beneficio obtenido en la venta de cada producto de la gama "Herramientas" (en una columna denominada "Beneficio"). (El beneficio se calculará como el precio de venta menos el precio del proveedor)

Solución

```
mysql> SELECT Nombre, (PrecioVenta-PrecioProveedor) AS Beneficio  
-> FROM Productos  
-> WHERE Gama = 'Herramientas';
```

```
+-----+-----+  
| Nombre           | Beneficio |  
+-----+-----+  
| Sierra de Poda 400MM |      3.00 |  
| Pala              |      1.00 |  
| Rastrillo de Jardín |      1.00 |  
| Azadón            |      1.00 |  
+-----+-----+  
4 rows in set (0,00 sec)
```

3.6. Ejercicio

Mostrar cuál es el beneficio máximo (en una columna denominada "Beneficio") que se puede obtener con la venta de un producto de los que tenemos en Stock (si no tiene *stock* no cuenta).

Solución

```
mysql> SELECT MAX(PrecioVenta-PrecioProveedor) AS Beneficio
-> FROM Productos
-> WHERE CantidadEnStock > 0;
+-----+
| Beneficio |
+-----+
|      93.00 |
+-----+
1 row in set (0,00 sec)
```

3.7. Ejercicio

Mostrar el código del pedido, su fecha, el código del cliente y la fecha esperada del pedido para todos aquellos cuya fecha de entrega haya sido posterior a la esperada. El resultado debe estar ordenado por la fecha de pedido.

Solución

```
mysql> SELECT CodigoPedido, FechaPedido, CodigoCliente, FechaEspera
-> FROM Pedidos
-> WHERE FechaEsperada < FechaEntrega
-> ORDER BY FechaPedido;
```

CodigoPedido	FechaPedido	CodigoCliente	FechaEsperada
32	2007-01-07	4	2007-01-19
96	2008-03-20	35	2008-04-12
49	2008-07-12	26	2008-07-22
31	2008-09-04	13	2008-09-30
113	2008-10-28	36	2008-11-09
128	2008-11-10	38	2008-12-10
115	2008-11-29	36	2009-01-26
55	2008-12-10	14	2009-01-10
9	2008-12-22	1	2008-12-27
60	2008-12-22	1	2008-12-27
18	2009-01-05	9	2009-01-06
16	2009-01-06	7	2009-01-07
17	2009-01-08	7	2009-01-09
22	2009-01-11	9	2009-01-11
13	2009-01-12	7	2009-01-14

	123		2009-01-15		30		2009-01-20	
	114		2009-01-15		36		2009-01-29	
	103		2009-01-15		30		2009-01-20	
	28		2009-02-10		3		2009-02-17	
	68		2009-02-10		3		2009-02-17	
	112		2009-03-05		36		2009-04-06	
	38		2009-03-05		19		2009-03-06	
	39		2009-03-06		19		2009-03-07	
	40		2009-03-09		19		2009-03-10	
	42		2009-03-22		19		2009-03-23	
	43		2009-03-25		23		2009-03-26	
	44		2009-03-26		23		2009-03-27	
	45		2009-04-01		23		2009-03-04	
	46		2009-04-03		23		2009-03-04	
	92		2009-04-19		27		2009-04-30	
	106		2009-05-13		30		2009-05-15	
	126		2009-05-13		30		2009-05-15	

+-----+-----+-----+-----+

32 rows in set (0,01 sec)

3.8. Ejercicio

Obtener cuántos pedidos nos ha realizado el cliente que tiene el código 30.

Solución

```
mysql> SELECT COUNT(*) AS 'Total pedidos cliente 30'
-> FROM Pedidos
-> WHERE CodigoCliente = 30;
+-----+
| Total pedidos cliente 30 |
+-----+
|                        10 |
+-----+
1 row in set (0,00 sec)
```


3.9. Ejercicio

Se ha detectado que hay errores en nuestros datos. Se han encontrado pedidos con fecha de entrega nula y estado "Entregado". Se desea encontrar esas inconsistencias y mostrarlas ordenadas por la fecha de pedido.

Solución

```
mysql> SELECT CodigoCliente, FechaPedido, Estado
-> FROM Pedidos
-> WHERE FechaEntrega IS NULL AND Estado = 'Entregado'
-> ORDER BY FechaPedido;
```

```
+-----+-----+-----+
| CodigoCliente | FechaPedido | Estado    |
+-----+-----+-----+
|           28 | 2009-01-24  | Entregado |
|           3  | 2009-02-07  | Entregado |
|           3  | 2009-02-07  | Entregado |
+-----+-----+-----+
3 rows in set (0,00 sec)
```

3.10. Ejercicio

Mostrar las líneas del pedido 10 ordenadas por el número de línea.

Solución

```
mysql> SELECT NumeroLinea, CódigoProducto, Cantidad, PrecioUnidad  
-> FROM DetallePedidos  
-> WHERE CódigoPedido = 10  
-> ORDER BY NumeroLinea;
```

NumeroLinea	CódigoProducto	Cantidad	PrecioUnidad
1	FR-91	30	75.00
2	FR-82	5	70.00
3	OR-234	5	64.00

3 rows in set (0,00 sec)

4. Bibliografía

- MySQL 8.0 Reference Manual. <https://dev.mysql.com/doc/refman/8.0/en/>
- Oracle Database Documentation. <https://docs.oracle.com/en/database/oracle/oracle-database/index.html>
- MySQL Tutorial. <https://www.w3schools.com/mysql/>
- GURU99. Tutorial de MySQL para principiantes Aprende en 7 días. <https://guru99.es/sql/>
- SQL Tutorial - Learn SQL. <https://www.sqltutorial.net/>



Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](https://creativecommons.org/licenses/by-sa/4.0/)