

TRABAJANDO CON LA EMPRESA PERSONAL

Trabajo de Alejandro Sainz Sainz

BD-
ACTIVIDAD 3.6

MODIFICANDO LA BBDD.....	2
INSERTS EN LA TABLA.....	3
UPDATES	5
UPDATE 1	5
UPDATE 2.....	6
UPDATE 3.....	7
UPDATE 4.....	8
UPDATE 5.....	9
CONSULTAS	11
CONSULTA 1	11
CONSULTA 2	12
CONSULTA 3	13
CONSULTA 4	14
CONSULTA 5	15
CONSULTA 6	16
CONCLUSIÓN.....	17

MODIFICANDO LA BBDD

Simplemente una pequeña modificación.

Por ajustar demasiado los campos descriptivos, se quedan cortos en tamaño para algunos de los datos.

Por lo tanto, los datos de tipo varchar ven aumentada su longitud hasta 50 en unos casos o 100 en otros. Así siempre tenemos longitud de sobra para almacenar los datos requeridos.

INSERTS EN LA TABLA

Bien, no estoy seguro de el número total de inserts, pero de seguro son más de 100. Como mi base de datos contenía muchas tablas, tanto principales como intermedias, he visto un grave problema en esto, ya que necesitaría muchos más inserts para poder obtener un modelo con el que pudiese trabajar más. De todos modos, aún no contando con atributos del tipo salario en los empleados, he podido reunir la cantidad suficiente como para realizar un trabajo aceptable.

```
insert into paquetes values
('01AB','Barranquismo1','Barranquismo inicial','Alto Los Montes',350.90,12),
('01CD','Kayaks','Kayaks Principiante','Bahía Loa Ahogaos',290.50,8),
('84X4','Parapente','Parapente Guiado','CumbresVentosas',675.50,6),
('46B3','Senderismo','Rutas tranquilas','ColinasSerenas',121.16,24),
('86C1','Ruta Globo','Rutas Globo','Airesfrescos',1121.95,4),
('12ZX','Conduccion 2','Rutas en Todoterreno Zonas Abruptas','La Cuesta Empiná',520.00,16),
('6912','Paracaidismo','Saltos desde avioneta','Aerodromo Militar',945.56,4),
('01AC','Barranquismo2','Barranquismo Rutas Intermedias','Alto de la Sierra',421.75,10),
('46BK','Senderismo2','Marchas Avanzadas','Villa Olivos',195.80,18),
('84X5','Parapente2','Parapente Libre Asistido','Aires Turbulentos',721.40,5),
('R145','Rafting','Descenso Aguas Bravas','Rio Rapido',285.66,16);
```

```
insert into localidad values
('01001','Alberca Vieja','01'),
('01007','Alto de Prado','01'),
('39200','Reinosa','39'),
('39400','Los Corrales de Buelna','39'),
('39460','Cartes','39'),
('11220','ValdePrado','11'),
('23000','Jaen','23'),
('23440','Baeza','23'),
('28000','Madrid','28'),
('28801','Alcalá de Henares','28'),
('08000','Barcelona','08'),
('08860','Casteldefels','08'),
('37000','Salamanca','37');
```

```

• insert into clientes values
('12345678A', 'José Fernández', '1980-05-15', '612345678', 'Calle Sol, 12', 'jose.fernandez@correo.com', '39200'),
('87654321B', 'Laura Gómez', '1992-08-25', '623456789', 'Calle Luna, 45', 'laura.gomez@correo.com', '39460'),
('11223344C', 'Carlos Ruiz', '1987-02-18', '634567890', 'Calle Estrella, 32', 'carlos.ruiz@correo.com', '28000'),
('22334455D', 'María Díaz', '1990-10-05', '645678901', 'Calle del Sol, 56', 'maria.diaz@correo.com', '08000'),
('33445566E', 'David Martín', '1985-04-30', '656789012', 'Calle Aire, 78', 'david.martin@correo.com', '23000'),
('44556677F', 'Sara López', '1993-12-11', '667890123', 'Calle Olmo, 21', 'sara.lopez@correo.com', '01001'),
('12345679A', 'Juan Pérez', '1990-05-21', '611223344', 'Calle Mayor 1', 'juan.perez@email.com', '01001'),
('23456789B', 'Ana Gómez', '1985-11-15', '622334455', 'Avda. del Sol 3', 'ana.gomez@email.com', '01007'),
('34567890C', 'Carlos Rodríguez', '1992-03-30', '633445566', 'Calle Luna 5', 'carlos.rodriguez@email.com', '39200'),
('45678901D', 'María López', '1988-07-10', '644556677', 'Paseo del Prado 2', 'maria.lopez@email.com', '39400'),
('56789012E', 'Luis Martín', '1980-01-22', '655667788', 'Avda. Cataluña 10', 'luis.martin@email.com', '39460'),
('67890123F', 'Elena Sánchez', '1995-06-18', '666778899', 'Calle Flor 12', 'elena.sanchez@email.com', '11220'),
('78901234G', 'Pedro Ruiz', '1993-12-02', '677889900', 'Calle del Viento 8', 'pedro.ruiz@email.com', '28000'),
('89012345H', 'Laura Fernández', '1997-09-25', '688990011', 'Calle Nueva 15', 'laura.fernandez@email.com', '08860'),
('90123456I', 'David González', '1991-08-10', '699001122', 'Calle Los Olivos 4', 'david.gonzalez@email.com', '37000'),
('12345679J', 'Sara Martínez', '1996-04-20', '710112233', 'Calle Verde 7', 'sara.martinez@email.com', '23440'),
('23456780K', 'José Díaz', '1982-12-14', '721223344', 'Calle del Mar 18', 'jose.diaz@email.com', '28801'),
('34567891L', 'Ricardo Ramírez', '1994-11-01', '732334455', 'Avda. España 6', 'ricardo.ramirez@email.com', '08000'),
('45678902M', 'Marta Ruiz', '1986-03-17', '743445566', 'Paseo Robledal 21', 'marta.ruiz@email.com', '23000');

```

```

• insert into vehiculos values
('1234567', 'Furgoneta', 3, '11111111A'),
('2345678', 'Camión', 10, '22222222B'),
('3456789', 'Coche', 5, '33333333V'),
('4567890', 'Furgoneta', 4, '11111111A'),
('5678901', 'Coche', 5, '22222222B');

• insert into alquiler_vehiculos values
('A001', '2025-01-15', '2025-01-20', '1234567', 1, 'P001'),
('A002', '2025-02-10', '2025-02-15', '2345678', 2, 'P002'),
('A003', '2025-03-05', '2025-03-10', '3456789', 3, 'P003'),
('A004', '2025-04-12', '2025-04-17', '4567890', 4, 'P004'),
('A005', '2025-05-20', '2025-05-25', '5678901', 5, 'P005');

```

Aquí muestro imágenes de algunos de los inserts.

UPDATES

UPDATE 1

En este primer update, lo que busco conseguir es subir el precio de los paquetes que se ofertan. Para eso, voy a subir aquellos por debajo de 600 euros en un 10% y el resto en un 5%;

	cod_paquete	nombre_paquete	descripcion	direccion	precio	num_max_personas
	01CD	Kayaks	Kayaks Principiante	Bahía Loa Ahogaos	319.55	8
	12ZX	Conduccion 2	Rutas en Todoterreno Zonas Abruptas	La Cuesta Empiná	572.00	16
	46BJ	Senderismo	Rutas tranquilas	ColinasSerenas	133.28	24
	46BK	Senderismo2	Marchas Avanzadas	Villa Olivos	215.38	18
	6912	Paracaidismo	Salto desde avioneta	Aerodromo Militar	992.84	4
	84X4	Parapente	Parapente Guiado	CumbresVentosas	709.28	6
	84X5	Parapente2	Parapente Libre Asistido	Aires Turbulentos	757.47	5
	86C1	Ruta Globo	Rutas Globo	Airesfrescos	1178.05	4
	R145	Rafting	Descenso Aguas Bravas	Rio Rapido	314.23	16

Esto son los valores iniciales.

```
39 • update paquetes p, paquetes p1 set p.precio = p.precio * 1.1, p1.precio = p1.precio * 1.05 where p.precio <= 600.00 and p1.precio > 600;
```

En principio, el update lo hice en dos comandos. Pero probando y leyendo por ahí, si no me equivoqué, creo que eso se podía hacer sólo con 1 tal y como muestra la imagen anterior.

```
-- Este update sirve para lo mismo pero con el rango de precio de los paquetes más caros
-- select * from paquetes;
-- update paquetes set precio = precio + (precio * 0.05) where precio > 600.00;
-- select * from paquetes;
```

En la imagen previa, vemos uno de los comando alternativos, el que era para paquetes con precio menor de 600 euros.

	cod_paquete	nombre_paquete	descripcion	direccion	precio	num_max_personas
▶	01AB	Barranquismo 1	Barranquismo inicial	Alto Los Montes	424.59	12
	01AC	Barranquismo2	Barranquismo Rutas Intermedias	Alto de la Sierra	510.32	10
	01CD	Kayaks	Kayaks Principiante	Bahía Loa Ahogaos	351.51	8
	12ZX	Conduccion 2	Rutas en Todoterreno Zonas Abruptas	La Cuesta Empiná	629.20	16
	46BJ	Senderismo	Rutas tranquilas	ColinasSerenas	146.61	24
	46BK	Senderismo2	Marchas Avanzadas	Villa Olivos	236.92	18
	6912	Paracaidismo	Salto desde avioneta	Aerodromo Militar	1042.48	4
	84X4	Parapente	Parapente Guiado	CumbresVentosas	744.74	6
	84X5	Parapente2	Parapente Libre Asistido	Aires Turbulentos	795.34	5

En principio, al realizar la comprobación, creo recordar que todo estaba correcto.

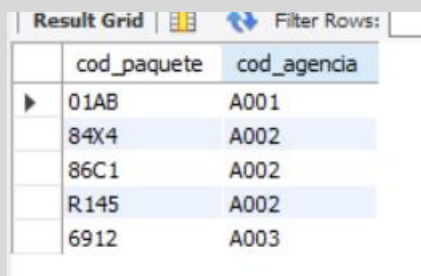
UPDATE 2

El segundo update, es también muy sencillito, ya que lo único que se pretende es cambiar un dato dependiendo de dos variables.

Este es el supuesto.

Una de las agencias colaboradoras ha decidido dejar de vender el paquete viaje en globo porque le da poca salida.

Y ha decidido cambiarlo por el otro paquete de parapente y especializarse en ello.



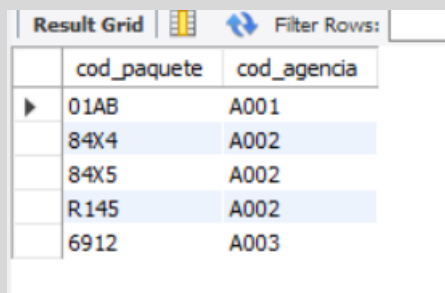
	cod_paquete	cod_agencia
▶	01AB	A001
	84X4	A002
	86C1	A002
	R145	A002
	6912	A003

Bien, el paquete que se quiere cambiar es el 86C1, que es viaje en globo, por el de parapente 2.

```
update oferta_agencia oa set oa.cod_paquete = '84X5' where oa.cod_paquete = '86C1' and oa.cod_agencia = 'A002';
```

Este es el comando que he usado.

Y el siguiente es el resultado.



	cod_paquete	cod_agencia
▶	01AB	A001
	84X4	A002
	84X5	A002
	R145	A002
	6912	A003

UPDATE 3

Cambiar el nombre de departamento de mantenimiento y la descripción en la tabla trabajadores y corregir un error en la tabla empleados por el cual un empleado de mantenimiento tenía una descripción que no concuerda.

cod_dep	departamento	num_matricula	dni	num_ss	nombre_completo	fecha_nacim	telefono	email	especialidad	cod_departamento	sede
2	Mantenimiento	4	44444444D	456789012345	María Rodríguez	1992-07-18	633456789	maria.r@viaj.es	Guía de Senderismo	2	4
2	Mantenimiento	7	77777777G	678901234567	Paco Fernandez	1987-01-01	655678901	paco.p@viaj.es	Mantenimiento	2	3
2	Mantenimiento	8	88888888H	678901234567	Jorge Jimenez	1999-01-01	655678901	jorge.j@viaj.es	Mantenimiento	2	4
2	Mantenimiento	9	99999999I	678901234567	Lola Lorenzo	2000-01-01	655678901	lola.l@viaj.es	Mantenimiento	2	6

Aquí vemos la situación antes de ejecutar el comando incluida la descripción errónea.

```

25 • update departamento d, empleados e
26   set d.departamento = 'Dep. Técnico', e.especialidad = 'Tecnico Mantenimiento'
27   where d.departamento = 'Mantenimiento' and e.especialidad in ('Mantenimiento', 'Guía de Senderismo');
28

```

Aquí vemos el comando que es un update de dos tablas a la vez, cada una con su respectiva modificación.

cod_dep	departamento	num_matricula	dni	num_ss	nombre_completo	fecha_nacim	telefono	email	especialidad	cod_departamento	sede
2	Dep. Técnico	4	44444444D	456789012345	María Rodríguez	1992-07-18	633456789	maria.r@viaj.es	Tecnico Mantenimiento	2	4
2	Dep. Técnico	7	77777777G	678901234567	Paco Fernandez	1987-01-01	655678901	paco.p@viaj.es	Tecnico Mantenimiento	2	3
2	Dep. Técnico	8	88888888H	678901234567	Jorge Jimenez	1999-01-01	655678901	jorge.j@viaj.es	Tecnico Mantenimiento	2	4
2	Dep. Técnico	9	99999999I	678901234567	Lola Lorenzo	2000-01-01	655678901	lola.l@viaj.es	Tecnico Mantenimiento	2	6

En la imagen anterior podemos apreciar el resultado.

UPDATE 4

Una de las reservas ha decidido quedarse más días de lo planeado inicialmente sobre la marcha por lo que debemos de cambiar la fecha y aumentar el precio en un 25%.

Esta en principio es otra maniobra sencillita que decidí hacerla para usar otro tipo de funciones en un update.

3	2025-03-12	2025-03-20	1035.93	1	3	P003
4	2025-04-18	2025-04-22	757.47	0	4	P004
5	2025-05-05	2025-05-10	1707.66	1	5	P005

Esta es la situación de la reserva 4 antes de realizar el update.

```

335
336 • update reservas r set r.fecha_fin_reserva = adddate(r.fecha_fin_reserva,interval 5 day), r.precio = precio * 1.25 where r.cod_reserva = 4;
337

```

Aquí el comando, usando la función adddate y un intervalo de 5 días. Además, en este update, incrementamos también el precio de la reserva de alojamiento.

	cod_reserva	fecha_reserva	fecha_fin_reserva	precio	grupo_completo	id_grupo	num_poliza
▶	1	2025-01-10	2025-01-15	705.54	1	1	P001
	2	2025-02-01	2025-02-05	992.84	0	2	P002
	3	2025-03-12	2025-03-20	1035.93	1	3	P003
	4	2025-04-18	2025-04-27	946.84	0	4	P004
	5	2025-05-05	2025-05-10	1707.66	1	5	P005

Aquí tenemos el resultado.

UPDATE 5

El último caso tiene que ver con el hecho de realizar los inserts de forma rápida y un poco sin pararme mucho a pensar, pero una vez que empezamos con los casos de los updates me vino que ni pintado.

En este caso voy a intentar realizar el update en base al resultado de una consulta.

Voy a intentar hacer un update del precio de las reservas basado en la suma de los precios de los paquetes contratados en esas reservas.

	cod_reserva	fecha_reserva	fecha_fin_reserva	precio	grupo_completo	id_grupo	num_poliza
▶	1	2025-01-10	2025-01-15	450.00	1	1	P001
	2	2025-02-01	2025-02-05	320.00	0	2	P002
	3	2025-03-12	2025-03-20	675.50	1	3	P003
	4	2025-04-18	2025-04-22	295.00	0	4	P004
	5	2025-05-05	2025-05-10	520.00	1	5	P005

Bien, este es el estado de las reservas justo después de realizar los inserts.

```

update reservas r
set r.precio =
(select sum(listado.precio) as precio_final from
(select r.cod_reserva,p.precio, p.cod_paquete
from paquetes p
join actividad a on p.cod_paquete = a.cod_paquete
join reservas r on a.cod_reserva = r.cod_reserva) as listado
where listado.cod_reserva = 1
group by listado.cod_reserva)
where r.cod_reserva = 1;

```

El resultado del update viene dado por lo obtenido de realizar una consulta anidada dentro de otra, para sumar el valor de todos los paquetes asociado a esa reserva, en este caso la reserva número 1. Uso la función sum para obtener el valor final, siendo este resultado el valor que va detrás del set.

Alejandro Sainz Sainz

Result Grid		Filter Rows:		Export:		Wrap Cell Content:	
	cod_reserva	fecha_reserva	fecha_fin_reserva	precio	grupo_completo	id_grupo	num_poliza
▶	1	2025-01-10	2025-01-15	705.54	1	1	P001
	2	2025-02-01	2025-02-05	320.00	0	2	P002
	3	2025-03-12	2025-03-20	675.50	1	3	P003
	4	2025-04-18	2025-04-22	295.00	0	4	P004
	5	2025-05-05	2025-05-10	520.00	1	5	P005

Este es el resultado después de ejecutar el comando. Mis clientes no están contentos y peor que se va a poner, ya que debo de hacerlo para las otras cuatro reservas restantes.

Result Grid		Filter Rows:		Export:		Wrap Cell Content:	
	cod_reserva	fecha_reserva	fecha_fin_reserva	precio	grupo_completo	id_grupo	num_poliza
▶	1	2025-01-10	2025-01-15	705.54	1	1	P001
	2	2025-02-01	2025-02-05	992.84	0	2	P002
	3	2025-03-12	2025-03-20	1035.93	1	3	P003
	4	2025-04-18	2025-04-22	757.47	0	4	P004
	5	2025-05-05	2025-05-10	1707.66	1	5	P005

Después de añadir los otros cuatro updates, cambiando la condición del where por el valor del cod_reserva, este es el resultado obtenido.

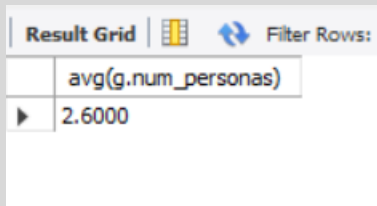
CONSULTAS

CONSULTA 1

Calcular el número medio de componentes de los grupos que reservan.

```
6  
7 • select avg(g.num_personas) from grupos g;  
8
```

Consulta sencillita para empezar. He intentado que en todas las consultas, o en gran parte de ellas, se utilicen diferentes funciones de las que hemos ido viendo.



The screenshot shows a 'Result Grid' window with a 'Filter Rows' button. The grid has one column labeled 'avg(g.num_personas)' and one row with the value '2.6000'.

avg(g.num_personas)
2.6000

Nada complicado la verdad.

Alejandro Sainz Sainz

CONSULTA 2

Contar sedes por provincia.

```
1 • select p.nombre, count(s.cod_sede) as 'Sucursales'
2   from provincia p
3  join localidad l on p.cp = l.cp
4  join sedes s on l.cp_completo = s.cp_completo
5  group by p.nombre;
```

Esta consulta es un poco más trabajada, tampoco mucho más, pero me sirve para usar algunas de las cosas vistas hasta ahora.

Result Grid			Filter Rows:
	nombre	Sucursales	
▶	Alava	2	
	Barcelona	4	
	Cadiz	1	
	Jaen	2	
	Madrid	3	
	Salamanca	1	
	Cantabria	2	

Este es su resultado.

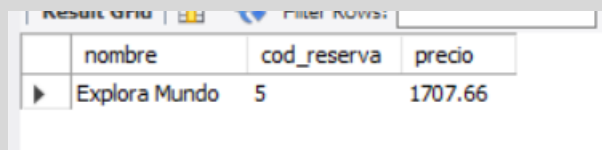
CONSULTA 3

Calcular que agencia ha facturado más.

Las agencias trabajan con reservas, así que vamos a calcular que agencia a facturado más ahora mismo de entre las reservas gestionadas por estas mismas.

```
select a.nombre, r.cod_reserva, r.precio
from agencias a
join reservas_agencias re on a.cod_agencia = re.cod_agencia
join reservas r on r.cod_reserva = re.cod_reserva
where r.precio in(select max(r.precio)
from reservas r
join reservas_agencias re on r.cod_reserva = re.cod_reserva);
```

En esta consulta usamos una consulta anidada que usa la función max para obtener el valor más alto del atributo precio de reserva apoyándonos en la tabla intermedia que relaciona reservas y agencias la cual es reservas_agencias(nada original el nombre).



	nombre	cod_reserva	precio
▶	Explora Mundo	5	1707.66

He aquí el resultado.

Como ya mencioné al iniciar este documento, necesitaría muchos más inserts para tener una buena base con la que trabajar, ya que sólo existen dos tuplas en la tabla reservas_agencia. No es una buena cantidad con la que trabajar.

CONSULTA 4

Ahora voy a volver a mostrar la ya conocida consulta que usé para realizar el update, que, aunque parezca ventajista, es una buena consulta para trabajar y probar cosas.

Calcular el precio total de todos los paquetes que se han contratado por reserva.

```
select listado.cod_reserva, sum(listado.precio) as 'Total precio paquetes' from
(select r.cod_reserva, p.precio, p.cod_paquete
from paquetes p
join actividad a on p.cod_paquete = a.cod_paquete
join reservas r on a.cod_reserva = r.cod_reserva) as listado
group by listado.cod_reserva;
```

Se realiza una consulta anidada después de relacionar de forma adecuada las tablas necesarias. Ha esa subconsulta se la llama listado, y de esa tabla intermedia llamada listado nos quedamos con lo que nos interesa, que es la suma de todos los precios agrupados por el código de la reserva.

	cod_reserva	Total precio paquetes
▶	1	705.54
	2	992.84
	3	1035.93
	4	757.47
	5	1707.66


En este caso el resultado es el que vemos en la imagen. El valor del precio es el mismo, pero hemos reducido mucho el tamaño del resultado de la consulta.

CONSULTA 5

Queremos conocer la cantidad de clientes han contratado cada paquete, ordenados de mayor a menor.

```
select listado.nombre_paquete as 'Paquete', count(listado.nombre_completo) as 'Clientes' from
(select p.nombre_paquete, c.nombre_completo from clientes c
join componentes co on c.dni = co.dni
join grupos g on co.id_grupo = g.id_grupo
join reservas r on r.id_grupo = g.id_grupo
join actividad a on r.cod_reserva = a.cod_reserva
join paquetes p on p.cod_paquete = a.cod_paquete)as listado
group by listado.nombre_paquete
order by count(listado.nombre_completo) desc;
```

De nuevo consultas anidadas, pero en este caso usando el join para relacionar una mayor cantidad de tablas. Una de las consultas nos va a dar un listado completo en el que relaciona cada cliente con el paquete contratado, luego simplemente tenemos que agruparlo por el nombre del paquete y contar el nombre o dni u otro campo identificador del cliente.



Paquete	Cientes
Paracaidismo	3
Barranquismo2	3
Conduccion 2	3
Parapente2	3
Barranquismo1	2
Kayaks	2
Senderismo2	2
Ruta Globo	2
Rafting	2

En esta imagen vemos el resultado.

CONSULTA 6

Esta última consulta viene derivada de la anterior. Ya que hemos podido ver que paquetes han sido contratados y por cuantos clientes, lo que nos interesa saber es cuales no han sido contratados nunca hasta la fecha.

Para ello voy a usar otra condición diferente a las usadas hasta ahora.

```
select p.cod_paquete, p.nombre_paquete from paquetes p
where not exists
(select a.cod_paquete
 from actividad a
 where p.cod_paquete = a.cod_paquete);
```

En este caso detrás del primer where que es el que relaciona las dos tablas voy a usar la condición not exists. Por lo tanto el resultado serán el código y nombre del paquete que no aparecen en la tabla intermedia actividad.

	cod_paquete	nombre_paquete
▶	46B3	Senderismo
	84X4	Parapente

Como vemos aquí, debemos tomar medidas con estos dos paquetes, que no se les da salida.

CONCLUSIÓN

Al final, estoy contento, aunque no haya realizada updates enormes con muchísimas condiciones, ni consultas con miles de resultados, pero he de tener en cuenta que habría necesitado el doble de datos para realizar una buena actividad. Si hubiese hecho un modelo más conservador con menos tablas, con menos volumen de datos quizá hubiese obtenido un mejor resultado.