

EJERCICIOS AVANZADOS SQL

PROYECTOS

CodP	Descripción	Localidad	Cliente	Teléfono
1	Garaje	Arganda	Felipe Sol	600111111
2	Solado	Rivas	José Pérez	912222222
3	Garaje	Arganda	Rosa López	666999666
4	Techado	Loeches	José Pérez	913333333
5	Buhardilla	Rivas	Ana Botijo	NULL

MÁQUINAS

CodM	Nombre	PrecioHora
1	Excavadora	15000
2	Hormigonera	10000
3	Volquete	11000
4	Apisonadora	18000

CONDUCTORES

CodC	Nombre	Localidad	Categoría
1	José Sánchez	Arganda	18
2	Manuel Díaz	Arganda	15
3	Juan Pérez	Rivas	20
4	Luis Ortiz	Arganda	18
5	Javier Martín	Loeches	12
6	Carmen Pérez	Rivas	15

TRABAJOS

CodC	CodM	CodP	Fecha	Tiempo
2	3	1	2023-09-10	100
3	1	2	2023-09-10	200
5	3	2	2023-09-10	150
4	3	2	2023-09-10	90
1	2	2	2023-09-12	120
2	3	3	2023-09-13	30
3	1	4	2023-09-15	300
2	3	2	2023-09-15	NULL
1	3	4	2023-09-15	180
5	3	4	2023-09-15	90
1	2	4	2023-09-17	NULL
2	3	1	2023-09-18	NULL

1. Obtener el nombre de los conductores con categoría 15.

```
SELECT nombre
FROM conductores
WHERE categ > 15
```

2. Obtener la descripción de los proyectos en los que se haya realizado trabajos durante los días 11 al 15 de septiembre de 2023.

- Solución con subconsultas

```
SELECT descrip
FROM proyectos
WHERE codP IN (SELECT codP
                FROM trabajos
                WHERE fecha BETWEEN '11/sep/02' AND '15/sep/02');
```

- Solución con join

```
SELECT descrip
FROM proyectos, trabajos
WHERE proyectos.codP = trabajos.codP
    AND fecha BETWEEN '11/sep/02' AND '15/sep/02';
```

- Solución con Inner Join

```
SELECT descrip
FROM proyectos INNER JOIN trabajos ON proyectos.codP = trabajos.codP
WHERE fecha BETWEEN '11/sep/02' AND '15/sep/02';
```

3. Obtener el nombre de los conductores que hayan trabajado con una Hormigonera, ordenados descendientemente.

```
SELECT nombre
FROM conductores
WHERE codC IN (SELECT codC
                FROM trabajos
                WHERE CodM = 2)
```

```

WHERE codM IN (SELECT codM
                FROM maquinas
                WHERE nombre = 'Hormigonera'))
ORDER BY nombre DESC;

```

4. Obtener el nombre de los conductores que hayan trabajado con una Hormigonera en proyectos de Arganda.

- Solución con subconsultas

```

SELECT nombre
FROM conductores
WHERE codC IN (SELECT codC
                FROM trabajos
                WHERE codM IN (SELECT codM
                                FROM maquinas
                                WHERE nombre = 'Hormigonera')
                AND codP IN (SELECT codP
                              FROM proyectos
                              WHERE localidad = 'Arganda'));

```

- Solución con condición de Join en el Where

```

SELECT conductores.nombre
FROM conductores, trabajos, maquinas, proyectos
WHERE proyectos.localidad = 'Arganda' AND
    maquinas.nombre = 'Hormigonera' AND
    proyectos.codP = trabajos.codP AND
    trabajos.codM = maquinas.codM AND
    trabajos.codC = conductores.codC;

```

- Solución con Inner Join

```

SELECT conductores.nombre
FROM conductores INNER JOIN trabajos ON conductores.codC = trabajos.codC
    INNER JOIN maquinas ON trabajos.codM = maquinas.codM
    INNER JOIN proyectos ON proyectos.codP = trabajos.codP
WHERE proyectos.localidad = 'Arganda' AND
    maquinas.nombre = 'Hormigonera';

```

5. Obtener el nombre de los conductores y descripción del proyecto, para aquellos conductores que hayan trabajado con una Hormigonera en proyectos de Arganda durante los días 12 al 17 de Septiembre.

```

SELECT conductores.nombre, proyectos.descrip
FROM conductores, trabajos, maquinas, proyectos
WHERE proyectos.localidad = 'Arganda' AND
    maquinas.nombre = 'Hormigonera' AND
    trabajos.fecha BETWEEN '10/sep/02' AND '18/sep/02' AND
    proyectos.codP = trabajos.codP AND
    trabajos.codM = maquinas.codM AND
    trabajos.codC = conductores.codC;

```

6. Obtener los conductores que trabajan en los proyectos de José Pérez.

```

SELECT DISTINCT conductores.nombre
FROM conductores, trabajos, proyectos
WHERE proyectos.cliente = 'José Pérez' AND
    proyectos.codP = trabajos.codP AND
    trabajos.codC = conductores.codC;

```

7. Obtener el nombre y localidad de los conductores que NO trabajan en los proyectos de José Pérez

- Solución errónea. Selecciona conductores que no hayan trabajado en alguna ocasión en proyectos de José Pérez, ya que establece la condición por cada fila. Puede haber conductores que aparezcan en una fila en proyectos que no sean de José Pérez y en otras filas en proyectos de otros clientes.

```
SELECT DISTINCT conductores.nombre, conductores.localidad
FROM conductores, trabajos, proyectos
WHERE proyectos.cliente <> 'José Pérez' AND
      proyectos.codP = trabajos.codP AND
      trabajos.codC = conductores.codC;
```

- Solución correcta

```
SELECT conductores.nombre, conductores.localidad
FROM conductores
WHERE codC NOT IN (SELECT codC
                  FROM trabajos
                  WHERE codP IN (SELECT codP
                                FROM proyectos
                                WHERE cliente = 'José Pérez'));
```

8. Obtener todos los datos de los proyectos realizados en Rivas o que sean de un cliente llamado José.

- Solución errónea. La evaluación de la condición lógica no es la correcta. Evalúa todas las condiciones hasta el OR como primer operando y deja como segundo operando la última condición.

```
SELECT DISTINCT conductores.*
FROM conductores, trabajos, proyectos
WHERE proyectos.codP = trabajos.codP AND
      trabajos.codC = conductores.codC AND
      proyectos.localidad = 'Loeches' OR
      proyectos.cliente like '%Felipe%';
```

- Solución correcta

```
SELECT DISTINCT conductores.*
FROM conductores, trabajos, proyectos
WHERE proyectos.codP = trabajos.codP AND
      trabajos.codC = conductores.codC AND
      (proyectos.localidad = 'Loeches' OR
       proyectos.cliente like '%Felipe%');
```

9. Obtener los conductores que habiendo trabajado en algún proyecto, figuren sin horas trabajadas.

```
SELECT DISTINCT conductores.*
FROM conductores, trabajos
WHERE trabajos.codC = conductores.codC AND
      tiempo IS NULL;
```

10. Obtener los empleados que tengan como apellido Pérez y hayan trabajado en proyectos de localidades diferentes a las suyas

- Solución correcta en SqlServer e Internase.

```
SELECT conductores.*
FROM conductores, trabajos, proyectos
WHERE conductores.nombre like '%Pérez%' AND
      proyectos.localidad <> conductores.localidad AND
      proyectos.codP = trabajos.codP AND
      trabajos.codC = conductores.codC;
```

11. Obtener el nombre de los conductores y la localidad del proyecto, para aquellos conductores que hayan trabajado con máquinas con precio hora comprendido entre 10000 y 15000 ptas.

```

SELECT DISTINCT C.nombre As "Conductor", P.localidad AS "Localidad del Proyecto"
FROM conductores C, trabajos T, proyectos P, maquinas M
WHERE M.preciohora BETWEEN 10000 AND 15000 AND
      M.codM = T.codM AND
      P.codP = T.codP AND
      T.codC = C.codC;

```

12. Obtener el nombre y localidad de los conductores, y la localidad del proyecto para aquellos proyectos que sean de Rivas y en los que no se haya utilizado una máquina de tipo Excavadora o una máquina de tipo Hormigonera.

- Solución correcta. Cuidado con la expresión lógica de la subconsulta. Es necesario poner los paréntesis para indicar el orden de evaluación de la expresión lógica.

```

SELECT C.nombre As "Conductor", C.localidad As "Localidad Conductor",
      P.localidad As "Localidad del Proyecto", M.nombre As Maquina
FROM conductores C, trabajos T, proyectos P, maquinas M
WHERE P.localidad = 'Rivas' AND
      P.CodP NOT IN (SELECT codP
                     FROM trabajos T1, maquinas M1
                     WHERE (M1.nombre = 'Excavadora' OR
                           M1.nombre = 'Hormigonera') AND
                           M1.codM = T1.codM )
      AND
      M.codM = T.codM AND
      P.codP = T.codP AND T.codC = C.codC;

```

13. Obtener todos los datos de los proyectos, y para aquellos proyectos realizados el día 15 de Septiembre, además incluir el nombre y localidad de los conductores que hayan trabajado en dicho proyecto.

- En Interbase, ponemos en la segunda consulta cadenas de caracteres con el mismo dominio que las correspondientes columnas de la otra consulta.

```

SELECT DISTINCT proyectos.*, conductores.nombre, conductores.localidad
FROM conductores, trabajos, proyectos
WHERE trabajos.fecha = '15/sep/02' AND
      proyectos.codP = trabajos.codP AND
      trabajos.codC = conductores.codC
UNION
SELECT DISTINCT proyectos.*, ' ', ' ', ' '
FROM proyectos
WHERE codP NOT IN (SELECT codP FROM trabajos WHERE fecha = '15/sep/02');

```

14. Obtener el nombre de los conductores y el nombre y localidad de los clientes, en los que se haya utilizado la máquina con precio hora más elevado.

```

SELECT DISTINCT conductores.nombre, proyectos.cliente, proyectos.localidad
FROM conductores, trabajos, proyectos
WHERE trabajos.codM IN (SELECT codM
                       FROM maquinas
                       WHERE preciohora IN (SELECT MAX(preciohora)
                                             FROM maquinas) ) AND
      proyectos.codP = trabajos.codP AND
      trabajos.codC = conductores.codC;

```

15. Obtener todos los datos de los proyectos que siempre han utilizado la máquina de precio más bajo.

```

SELECT DISTINCT proyectos.*
FROM trabajos, proyectos
WHERE trabajos.codP NOT IN
      (SELECT codP
       FROM trabajos
       WHERE codM IN (SELECT codM
                     FROM maquinas

```

```

WHERE preciohora <> (SELECT MIN(preciohora)
FROM maquinas) )) AND
proyectos.codP = trabajos.codP;

```

- 16. Obtener los proyectos en los que haya trabajado el conductor de categoría más alta menos dos puntos, con la máquina de precio hora más bajo.**

```

SELECT DISTINCT proyectos.*
FROM trabajos, proyectos
WHERE codM IN (SELECT codM
FROM maquinas
WHERE preciohora IN (SELECT MIN(preciohora)
FROM maquinas) ) AND
codC IN (SELECT codC
FROM conductores
WHERE categ IN (SELECT MAX(categ) - 2
FROM conductores) ) AND
proyectos.codP = trabajos.codP;

```

- 17. Obtener por cada uno de los clientes el tiempo total empleado en sus proyectos.**

```

SELECT cliente, SUM(Tiempo)
FROM proyectos LEFT JOIN trabajos ON proyectos.codP = trabajos.codP
GROUP BY cliente;

```

- 18. Obtener por cada uno de los proyectos existentes en la BD, la descripción del proyecto, el cliente y el total a facturar en ptas y en euros. Ordenar el resultado por uno de los totales y por cliente.**

```

SELECT cliente, descrip, SUM(Tiempo*preciohora) As TotalPtas,
SUM(Tiempo*preciohora) / 166.366 AS TotalEuros
FROM proyectos LEFT JOIN trabajos ON proyectos.codP = trabajos.codP
LEFT JOIN maquinas ON trabajos.codM = maquinas.codM
GROUP BY descrip, cliente
ORDER BY 3,1;

```

- 19. Obtener para el proyecto que más se vaya a facturar la descripción del proyecto, el cliente y el total a facturar en Ptas. y en euros**

```

1.
SELECT cliente, descrip, SUM(Tiempo*preciohora) As TotalPtas,
SUM(Tiempo*preciohora) / 166.386 AS TotalEuros
FROM proyectos INNER JOIN trabajos ON proyectos.codP = trabajos.codP
INNER JOIN maquinas ON trabajos.codM = maquinas.codM
GROUP BY descrip, cliente
HAVING SUM(Tiempo*preciohora) >= ALL (SELECT SUM(Tiempo*preciohora)
FROM trabajos INNER JOIN maquinas
ON trabajos.codM = maquinas.codM
GROUP BY CodP);

```

```

2.
CREATE VIEW sumasProyecto (cliente, descrip, totalPtas, totalEuros)
AS SELECT cliente, descrip, SUM(tiempo*preciohora),
SUM(tiempo*preciohora) / 166.386
FROM proyectos INNER JOIN trabajos ON proyectos.codP = trabajos.codP
INNER JOIN maquinas ON trabajos.codM = maquinas.codM
GROUP BY descrip, cliente;

```

```

SELECT *
FROM sumasProyecto
WHERE totalEuros IN (SELECT MAX(totalEuros)
FROM sumasProyecto)

```

```

/* DROP VIEW sumasProyecto; */

```

20. Obtener los conductores que hayan trabajado en todos los proyectos de la localidad de Arganda.

```
SELECT trabajos.codc, nombre, COUNT(distinct trabajos.codp)
FROM conductores INNER JOIN trabajos ON conductores.codC = trabajos.codC
      INNER JOIN proyectos ON trabajos.codp=proyectos.codp
WHERE proyectos.localidad='Arganda'
GROUP BY trabajos.codc, nombre
HAVING COUNT(DISTINCT trabajos.codp) IN (SELECT COUNT(*)
                                         FROM proyectos
                                         WHERE localidad = 'Arganda');
```

21. Obtener el tiempo máximo dedicado a cada proyecto para aquellos proyectos en los que haya participado más de un conductor diferente.

- Solución errónea. Se pueden seleccionar proyectos que hayan tenido dos veces el mismo conductor en trabajos.

```
SELECT CodP, MAX(tiempo)
FROM trabajos
GROUP BY CodP
HAVING COUNT(CodC) > 1;
```

- Solución correcta.

```
SELECT CodP, MAX(tiempo) As MaxTiempo
FROM trabajos
GROUP BY CodP
HAVING COUNT(DISTINCT CodC) > 1;
```

22. Obtener el número de partes de trabajo, código del proyecto, descripción y cliente para aquel proyecto que figure con más partes de trabajo.

```
SELECT proyectos.CodP, descrip, cliente, COUNT(*) As "Número de Trabajos"
FROM proyectos, trabajos
WHERE proyectos.codP = trabajos.codP
GROUP BY proyectos.CodP, descrip, cliente
HAVING COUNT(*) >= ALL (SELECT COUNT(*)
                       FROM trabajos
                       GROUP BY CodP);
```

23. Obtener la localidad cuyos conductores (al menos uno) haya participado en más de dos proyectos diferentes.

```
SELECT localidad
FROM conductores
WHERE CodC IN (SELECT CodC
               FROM trabajos
               GROUP BY CodC
               HAVING COUNT(DISTINCT CodP) > 2);
```

24. Subir el precio por hora en un 10% del precio por hora más bajo para todas las máquinas excepto para aquella que tenga el valor más alto.

```
UPDATE maquinas
SET preciohora = preciohora + (SELECT MIN(preciohora)*0.1
                              FROM maquinas)
WHERE preciohora NOT IN (SELECT MAX(preciohora)
                        FROM maquinas);
```

- 25. Subir la categoría un 15% a los conductores que no hayan trabajado con Volquete y hayan trabajado en más de un proyecto distinto.**

```
UPDATE conductores
SET categ = categ*1.15
WHERE CodC NOT IN (SELECT CodC
                    FROM trabajos
                    WHERE codM NOT IN (SELECT codM
                                      FROM maquinas
                                      WHERE nombre = 'Volquete')) AND
                    CodC IN (SELECT CodC
                            FROM trabajos
                            GROUP BY CodC
                            HAVING COUNT(DISTINCT CodP) > 1);
```

- 26. Eliminar el proyecto Solado de José Pérez.**

```
1.
DELETE
FROM trabajos
WHERE CodP IN (SELECT CodP
               FROM proyectos
               WHERE descrip = 'Solado' AND
                  cliente = 'José Pérez' );
```

```
DELETE
FROM Proyectos
WHERE descrip = 'Solado' AND
      cliente = 'José Pérez' ;
```

2.

```
DELETE
FROM Proyectos
WHERE descrip = 'Solado' AND
      cliente = 'José Pérez' ;
DELETE
FROM trabajos
WHERE CodP NOT IN (SELECT CodP
                  FROM proyectos);
```

- 27. Modificar la estructura de la base de datos, añadiendo las claves foráneas, sin ninguna opción de integridad referencial.**

```
ALTER TABLE trabajos
ADD FOREIGN KEY (codC) REFERENCES conductores (codC);
```

```
ALTER TABLE trabajos
ADD FOREIGN KEY (codM) REFERENCES maquinas (codM);
```

```
ALTER TABLE trabajos
ADD FOREIGN KEY (codP) REFERENCES proyectos (codP);
```

- 28. Insertar en la tabla trabajos la fila 'C01', 'M04','P07','19/09/02',100.**

```
INSERT INTO trabajos
VALUES ('C01', 'M04','P07','19/sep/02',null);
```

Inserción errónea, por no cumplir la restricción de integridad referencial el valor 'P07'

- 29. Eliminar el conductor 'C01' de la tabla conductores.**

```
DELETE
FROM conductores
```

```
WHERE codC = 'C01';
```

Imposible eliminar la fila porque se incumpliría la restricción de integridad referencial con la tabla trabajos.

30. Modificar el código del conductor 'C01' de la tabla conductores, por el código 'C05'.

```
UPDATE conductores  
SET codC = 'C05'  
WHERE codC = 'C01'
```

Imposible modificar el valor porque crearía un valor duplicado en el índice de clave primaria.

31. Modificar el código del conductor 'C01' de la tabla conductores, por el código 'C07'.

```
UPDATE conductores  
SET codC = 'C07'  
WHERE codC = 'C01'
```

Imposible modificar la fila porque se incumpliría la restricción de integridad referencial con la tabla trabajos.

32. Modificar la estructura de la base de datos, para que las claves foráneas tengan condiciones de integridad referencial en borrado y modificación. Especificar todas las opciones de integridad referencial, y ejecutar sentencias de actualización para comprobar su funcionamiento.

```
ALTER TABLE trabajos  
ADD FOREIGN KEY (codC) REFERENCES conductores (codC)  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

```
ALTER TABLE trabajos  
ADD FOREIGN KEY (codM) REFERENCES maquinas (codM)  
ON DELETE SET DEFAULT  
ON UPDATE SET DEFAULT;
```

```
ALTER TABLE trabajos  
ADD FOREIGN KEY (codP) REFERENCES proyectos (codP)  
ON DELETE SET NULL  
ON UPDATE CASCADE;
```

33. Crear una vista que contenga el nombre del conductor, la descripción del proyecto y la media aritmética del tiempo trabajado.

```
CREATE VIEW ap33 (conductor, proyecto, tiempoMedio)  
AS SELECT nombre, descrip, avg(tiempo)  
FROM conductores, trabajos, proyectos  
WHERE conductores.codC = trabajos.codC AND  
trabajos.codP = proyectos.codP  
GROUP BY nombre, descrip;
```

34. Crear una vista sobre la tabla trabajos, para los trabajos realizados después del 15 de septiembre de 2002. Crearla sin la cláusula "With Check Option" y sin ella, comprobando su funcionamiento.

```
CREATE VIEW ap34a (conductor, maquina, proyecto, fecha, tiempo)  
AS SELECT codc, codm, codp, fecha, tiempo  
FROM trabajos  
WHERE fecha > '15/sep/02';
```

```
INSERT INTO ap34  
VALUES ('C01','M01','P01','03/sep/02',10);
```

```
CREATE VIEW ap34b (conductor, maquina, proyecto, fecha, tiempo)  
AS SELECT codc, codm, codp, fecha, tiempo  
FROM trabajos
```



```
WHERE fecha > '15/sep/02'  
WITH CHECK OPTION;
```

```
INSERT INTO ap34  
VALUES ('C01','M01','P01','04/sep/02',10);
```

35. Eliminar las tablas de la base de datos.

```
DROP VIEW ap33;  
DROP VIEW ap34;
```

```
DROP TABLE trabajos;  
DROP TABLE conductores;  
DROP TABLE maquinas;  
DROP TABLE proyectos;
```