

Esquema

Materiales(Clave, Descripción, Precio, impuesto, PorcentajeImpuesto)

Proveedores(RFC, RazonSocial)

Proyectos(Numero, Denominacion)

Entregan(Clave, RFC, Numero, Fecha, Cantidad)

Consultas naturales

Consulta de un tabla completa

```
select * from materiales;
```

45 rows returned.

```
'1000', 'Varilla 3/16', '100', '10'
```

```
'1010', 'Varilla 4/32', '115', '11.5'
```

```
'1020', 'Varilla 3/17', '130', '13'
```

Selección

```
select * from materiales where clave=1000;
```

1 row returned.

```
'1000', 'Varilla 3/16', '100', '10'
```

Proyección

```
select clave,rfc,fecha from entregan;
```

87 rows returned.

```
'1000', 'AAAA800101', '2001-12-13'
```

Alfonso Imanol Macias Marañon - A01808099 - Laboratorio 20: Consultas en SQL

'1000', 'AAAA800101', '1999-07-13'

'1010', 'BBBB800101', '1998-07-28'}

Reunión Natural

```
select * from materiales,entregan where materiales.clave = entregan.clave;
```

87 rows returned.

'1000', 'Varilla 3/16', '100', '10', '1000', 'AAAA800101', '5000', '2001-12-13', '165'

'1000', 'Varilla 3/16', '100', '10', '1000', 'AAAA800101', '5019', '1999-07-13', '254'

'1010', 'Varilla 4/32', '115', '11.5', '1010', 'BBBB800101', '5001', '1998-07-28', '528'

Si algún material no se ha entregado ¿Aparecería en el resultado de esta consulta?

No, porque este no tendría fecha.

Reunión con criterio específico

```
select * from entregan,proyectos where entregan.numero <= proyectos.numero;
```

836 rows returned.

'1000', 'AAAA800101', '5000', '2001-12-13', '165', '5000', 'Vamos Mexico'

'1200', 'EEEE800101', '5000', '2003-03-15', '177', '5000', 'Vamos Mexico'

'1400', 'AAAA800101', '5000', '1999-04-07', '382', '5000', 'Vamos Mexico'

Unión (se ilustra junto con selección)

```
(select * from entregan where clave=1450)
```

union

```
(select * from entregan where clave=1300)
```

2 rows returned.

'1300', 'GGGG800101', '5005', '2004-02-28', '521'

'1300', 'GGGG800101', '5010', '2001-02-10', '119'

¿Cuál sería una consulta que obtuviera el mismo resultado sin usar el operador Unión?

```
select * from entregan where clave=1450;
```

```
select * from entregan where clave=1300;
```

Compruébalo.

'1300', 'GGGG800101', '5005', '2004-02-28', '521'

'1300', 'GGGG800101', '5010', '2001-02-10', '119'

Sería la misma, pero son dos consultas como tal porque son dos SELECT.

Intersección (se ilustra junto con selección y proyección)

```
select clave from entregan where numero = 5001 and clave in (select clave from entregan  
where numero = 5018);
```

En mysql workbench no existe intersect, tuve que usar una subconsulta.

1 rows returned.

'1010'

Diferencia (se ilustra con selección)

```
select * from entregan where clave not in (select clave from entregan where clave = 1000);
```

Igual, subconsultas.

85 rows returned.

'1010', 'BBBB800101', '5001', '1998-07-28', '528'

Alfonso Imanol Macias Marañon - A01808099 - Laboratorio 20: Consultas en SQL

'1010', 'BBBB800101', '5018', '1997-02-09', '523'

'1020', 'CCCC800101', '5002', '2003-12-16', '582'

Producto cartesiano

```
select * from entregan,materiales
```

1000 rows returned.

'1000', 'AAAA800101', '5000', '2001-12-13', '165', '1000', 'Varilla 3/16', '100', '10'

'1000', 'AAAA800101', '5000', '2001-12-13', '165', '1010', 'Varilla 4/32', '115', '11.5'

'1000', 'AAAA800101', '5000', '2001-12-13', '165', '1020', 'Varilla 3/17', '130', '13'

¿Cómo está definido el número de tuplas de este resultado en términos del número de tuplas de entregan y de materiales?

Está definido como el mismo número de tuplas. Ocurre precisamente ya que cada fila en la tabla entregan se repite una vez por cada fila en la tabla materiales que tiene la misma clave que en entregan correspondientemente.

Construcción de consultas a partir de una especificación

Plantea ahora una consulta para obtener las descripciones de los materiales entregados en el año 2000.

```
select descripcion from materiales join entregan on materiales.clave = entregan.clave where  
year(fecha) = 2000;
```

Join sencillo con las claves primarias.

12 rows returned.

'Varilla 3/17'

'Varilla 4/34'

'Block'

'Sillar gris'

'Sillar gris'

'Cantera blanca'

'Recubrimiento P1028'

'Tubería 3.6'

'Pintura C1010'

'Pintura B1021'

'Pintura B1021'

'Pintura B1022'

¿Por qué aparecen varias veces algunas descripciones de material?

Puede que se hayan entregado el mismo material varias veces el mismo año.

Uso del calificador distinct

En el resultado anterior, observamos que una misma descripción de material aparece varias veces. Agrega la palabra distinct inmediatamente después de la palabra select a la consulta que planteaste antes. ¿Qué resultado obtienes en esta ocasión?

'Varilla 3/17'

'Varilla 4/34'

'Block'

'Sillar gris'

'Cantera blanca'

'Recubrimiento P1028'

'Tubería 3.6'

'Pintura C1010'

'Pintura B1021'

'Pintura B1022'

Se eliminaron los duplicados.

Ordenamientos.

Obtén los números y denominaciones de los proyectos con las fechas y cantidades de sus entregas, ordenadas por número de proyecto, presentando las fechas de la más reciente a la más antigua.

```
select proyectos.*, fecha, cantidad from entregan join proyectos on entregan.numero =  
proyectos.numero order by fecha;
```

87 rows returned.

'5004', 'Educando en Coahuila', '0000-00-00', '152'

'5015', 'CIT Yucatan', '1997-01-06', '272'

'5018', 'Tu cambio por la educación', '1997-02-09', '523'

Operadores de cadena

El operador LIKE se aplica a datos de tipo cadena y se usa para buscar registros, es capaz de hallar coincidencias dentro de una cadena bajo un patrón dado. También contamos con el operador comodín (%), que coincide con cualquier cadena que tenga cero o más caracteres. Este puede usarse tanto de prefijo como sufijo.

```
SELECT * FROM materiales where descripcion LIKE 'Si%'
```

¿Qué resultado obtienes?

2 rows returned.

'1120', 'Sillar rosa', '100', '10'

'1130', 'Sillar gris', '110', '11'

Explica que hace el símbolo '%'.

Básicamente incluye la búsqueda de string comodín. Indica que toda descripción debe empezar por "Si" y luego puede seguir por cualquier otro string posible.

¿Qué sucede si la consulta fuera : LIKE 'Si' ?

Estaría buscando absolutamente "Si" como se observó previamente. Pero si no se incluye el comodín indica entonces que toda descripción que se muestre deberá tener absolutamente la descripción "Si".

¿Qué resultado obtienes?

NULL

Funciones #1

¿Qué resultado obtienes de ejecutar el siguiente código?

```
DECLARE @foo varchar(40);
```

```
DECLARE @bar varchar(40);
```

```
SET @foo = '¿Que resultado';
```

```
SET @bar = '¿¿¿???'
```

```
SET @foo += ' obtienes?';
```

```
PRINT @foo + @bar;
```

No logré correr ese código. Pero parece fácil de intuir. Declare es para declarar una serie de variables, en este caso se utiliza arroba para indicar el nombre y luego el tipo de variable.

Después, se le asigna un valor que en este caso si corresponde con lo que puede almacenar que son unos cuantos caracteres. Se le concatena unos cuantos caracteres extra a foo.

Por último está print, donde se imprime la concatenación de ambas strings. Lo que debería dar: ¿Que resultado obtienes? ¿¿¿???

Funciones #2

Ahora explica el comportamiento, funci  n y resultado de cada una de las siguientes consultas:

```
SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%';
```

Pienso que deber  a haberme dado todo RFC que tenga una A,B,C o D de inicio en su string. Pero realmente me arroj   NULL.

```
SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%';
```

Pienso que deber  a haberme dado todo RFC que no tenga una A de inicio en su string. Pero realmente me arroj   NULL.

```
SELECT Numero FROM Entregan WHERE Numero LIKE '___6';
```

Este si funciona, me arroja todo RFC que terminara en 6, esto se debe a que el gui  n bajo indica comod  n para un solo car  cter en LIKE.

Operadores L  gicos.

```
SELECT Clave,RFC,Numero,Fecha,Cantidad
```

```
FROM Entregan
```

```
WHERE Numero Between 5000 and 5010;
```

43 rows returned.

```
'1000', 'AAAA800101', '5000', '2001-12-13', '165'
```

```
'1010', 'BBBB800101', '5001', '1998-07-28', '528'
```

```
'1020', 'CCCC800101', '5002', '2003-12-16', '582'
```

  C  mo filtrar  as rangos de fechas?

```
SELECT Clave,RFC,Numero,Fecha,Cantidad
```

```
FROM Entregan
```


WHERE year(Fecha) Between 1997 and 2000;

39 rows returned.

'1000', 'AAAA800101', '5019', '1999-07-13', '254'

'1010', 'BBBB800101', '5001', '1998-07-28', '528'

'1010', 'BBBB800101', '5018', '1997-02-09', '523'

Exists

Se utiliza para especificar dentro de una subconsulta la existencia de ciertas filas.

¿Qué hace la consulta?

SELECT RFC, Cantidad, Fecha, Numero

FROM Entregan

WHERE Numero BETWEEN 5000 AND 5010

AND EXISTS (

SELECT RFC

FROM Proveedores

WHERE RazonSocial LIKE 'La%'

AND Entregan.RFC = Proveedores.RFC

)

Busca todas las entregas entre los números de entrega 5000 al 5010. Esto ocurre con la subconsulta que dice que las condiciones son que la RazonSocial comience con 'La%' y que el RFC en la tabla Entregan coincida con el RFC en la tabla Proveedores.

¿Qué función tiene el paréntesis () después de EXISTS?

Agrupar la subconsulta.

Alfonso Imanol Macias Marañon - A01808099 - Laboratorio 20: Consultas en SQL

Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador IN.

```
SELECT RFC, Cantidad, Fecha, Numero  
  
FROM Entregan  
  
WHERE Numero BETWEEN 5000 AND 5010  
  
AND RFC IN (  
  
    SELECT RFC  
  
    FROM Proveedores  
  
    WHERE RazonSocial LIKE 'La%'  
  
)
```

Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador NOT IN Realiza un ejemplo donde apliques algún operador : ALL, SOME o ANY.

```
SELECT RFC, Cantidad, Fecha, Numero  
  
FROM Entregan  
  
WHERE Numero BETWEEN 5000 AND 5010  
  
AND RFC NOT IN (  
  
    SELECT RFC  
  
    FROM Proveedores  
  
    WHERE RazonSocial NOT LIKE 'La%'  
  
)
```

Top

¿Qué hace la siguiente sentencia? Explica por qué.

```
SELECT TOP 2 * FROM Proyectos
```

*La ajuste a mysql workbench: SELECT * FROM Proyectos LIMIT 2; y esencialmente son lo mismo. Tanto TOP 2 en SQL Server como LIMIT 2 en MySQL devolverán las dos primeras filas según el orden de la tabla. Por lo tanto, en términos de funcionalidad, ambos hacen lo mismo: limitan el resultado a las dos primeras filas.*

Modificando la estructura de un tabla existente.

```
ALTER TABLE materiales ADD PorcentajeImpuesto NUMERIC(6,2);
```

```
UPDATE materiales SET PorcentajeImpuesto = 2*clave/1000;
```

¿Qué consulta usarías para obtener el importe de las entregas es decir, el total en dinero de lo entregado, basado en la cantidad de la entrega y el precio del material y el impuesto asignado?

```
SELECT
```

```
    Entregan.Clave,
```

```
    Entregan.RFC,
```

```
    Entregan.Numero,
```

```
    Entregan.Fecha,
```

```
    Entregan.Cantidad,
```

```
    Materiales.Descripcion,
```

```
    Materiales.Precio,
```

```
    Materiales.Precio * Entregan.Cantidad,
```

```
    Materiales.Precio * Entregan.Cantidad * (1 + (Materiales.PorcentajeImpuesto / 100)) AS  
ImporteTotal
```

FROM

Entregan

INNER JOIN

Materiales ON Entregan.Clave = Materiales.Clave;

Creación de vistas

Los materiales (clave y descripción) entregados al proyecto "México sin ti no estamos completos".

SELECT Materiales.Clave, Materiales.Descripcion

FROM Materiales

INNER JOIN Entregan ON Materiales.Clave = Entregan.Clave

INNER JOIN Proyectos ON Entregan.Numero = Proyectos.Numero

WHERE Proyectos.Denominacion = 'México sin ti no estamos completos';

3 rows returned.

'1030', 'Varilla 4/33'

'1230', 'Cemento '

'1430', 'Pintura B1022'

Los materiales (clave y descripción) que han sido proporcionados por el proveedor "Acme tools".

SELECT Materiales.Clave, Materiales.Descripcion

FROM Materiales

INNER JOIN Entregan ON Materiales.Clave = Entregan.Clave

INNER JOIN Proveedores ON Entregan.RFC = Proveedores.RFC

Alfonso Imanol Macias Marañon - A01808099 - Laboratorio 20: Consultas en SQL

WHERE Proveedores.RazonSocial = 'Acme tools';

0 rows returned.

El RFC de los proveedores que durante el 2000 entregaron en promedio cuando menos 300 materiales.

SELECT Entregan.RFC

FROM Entregan

WHERE YEAR(Entregan.Fecha) = 2000

GROUP BY Entregan.RFC

HAVING AVG(Entregan.Cantidad) >= 300;

3 rows returned.

'BBBB800101'

'FFFF800101'

'GGGG800101'

El Total entregado por cada material en el año 2000.

SELECT Clave, SUM(Cantidad) AS Total_Entregado

FROM Entregan

WHERE YEAR.Fecha) = 2000

GROUP BY Clave;

11 rows returned.

'1020', '8'

'1050', '623'

'1100', '466'

La Clave del material más vendido durante el 2001. (se recomienda usar una vista intermedia para su solución)

```
SELECT Clave  
  
FROM (  
  
    SELECT Clave, SUM(Cantidad) AS Total_Entregado  
  
    FROM Entregan  
  
    WHERE YEAR(Fecha) = 2001  
  
    GROUP BY Clave  
  
    ORDER BY Total_Entregado DESC  
  
    LIMIT 1  
  
) AS Material_Más_Vendido;
```

1 row returned.

'1260'

Productos que contienen el patrón 'ub' en su nombre.

```
SELECT Denominacion  
  
FROM Proyectos  
  
WHERE Denominacion LIKE '%ub%';
```

0 rows returned.

Denominación y suma del total a pagar para todos los proyectos.

```
SELECT Proyectos.Denominacion, SUM(Materiales.Precio * Entregan.Cantidad) AS  
Total_Pagar
```

Alfonso Imanol Macias Mara  n - A01808099 - Laboratorio 20: Consultas en SQL

FROM Proyectos

INNER JOIN Entregan ON Proyectos.Numero = Entregan.Numero

INNER JOIN Materiales ON Entregan.Clave = Materiales.Clave

GROUP BY Proyectos.Denominacion;

20 rows returned.

'Ampliaci  n de la carretera a la huasteca', '565135'

'Aztecon', '146595'

'CIT Campeche', '157755'

Denominaci  n, RFC y RazonSocial de los proveedores que se suministran materiales al proyecto Televisa en acci  n que no se encuentran apoyando al proyecto Educando en Coahuila (Solo usando vistas).

CREATE VIEW Proveedores_Televisa AS

SELECT DISTINCT P.RFC

FROM Proyectos Pr

JOIN Entregan E ON Pr.Numero = E.Numero

JOIN Proveedores P ON E.RFC = P.RFC

WHERE Pr.Denominacion = 'Televisa en acci  n';

CREATE VIEW Proveedores_Coahuila AS

SELECT DISTINCT P.RFC

FROM Proyectos Pr

JOIN Entregan E ON Pr.Numero = E.Numero

JOIN Proveedores P ON E.RFC = P.RFC

Alfonso Imanol Macias Marañon - A01808099 - Laboratorio 20: Consultas en SQL

WHERE Pr.Denominacion = 'Educando en Coahuila';

SELECT DISTINCT P.RFC, P.RazonSocial

FROM Proveedores P

JOIN Entregan E ON P.RFC = E.RFC

JOIN Proyectos Pr ON E.Numero = Pr.Numero

WHERE Pr.Denominacion = 'Televisa en acción' AND P.RFC NOT IN (SELECT * FROM Proveedores_Coahuila);

2 rows returned.

'CCCC800101', 'La Ferre'

'DDDD800101', 'Cecoferre'

Denominación, RFC y RazonSocial de los proveedores que se suministran materiales al proyecto Televisa en acción que no se encuentran apoyando al proyecto Educando en Coahuila (Sin usar vistas, utiliza not in, in o exists).

SELECT DISTINCT P.RFC, P.RazonSocial

FROM Proveedores P

JOIN Entregan E ON P.RFC = E.RFC

JOIN Proyectos Pr ON E.Numero = Pr.Numero

WHERE Pr.Denominacion = 'Televisa en acción' AND P.RFC NOT IN (

SELECT DISTINCT P2.RFC

FROM Proveedores P2

JOIN Entregan E2 ON P2.RFC = E2.RFC

JOIN Proyectos Pr2 ON E2.Numero = Pr2.Numero

Alfonso Imanol Macias Marañon - A01808099 - Laboratorio 20: Consultas en SQL

WHERE Pr2.Denominacion = 'Educando en Coahuila'

);

2 rows returned.

'CCCC800101', 'La Ferre'

'DDDD800101', 'Cecoferre'

Costo de los materiales y los Materiales que son entregados al proyecto Televisa en acción
cuyos proveedores también suministran materiales al proyecto Educando en Coahuila.

SELECT M.Clave, M.Descripción, M.Precio * E.Cantidad AS Costo_Material

FROM Materiales M

JOIN Entregan E ON M.Clave = E.Clave

JOIN Proyectos P ON E.Numero = P.Numero

JOIN Proveedores Pr ON E.RFC = Pr.RFC

WHERE P.Denominacion = 'Televisa en acción' AND Pr.RFC IN (

SELECT DISTINCT Pr2.RFC

FROM Proveedores Pr2

JOIN Entregan E2 ON Pr2.RFC = E2.RFC

JOIN Proyectos P2 ON E2.Numero = P2.Numero

WHERE P2.Denominacion = 'Educando en Coahuila'

);

2 rows returned.

'1080', 'Ladrillos rojos', '4300'

'1280', 'Tepetate', '3638'

Alfonso Imanol Macias Marañon - A01808099 - Laboratorio 20: Consultas en SQL

Nombre del material, cantidad de veces entregados y total del costo de dichas entregas por material de todos los proyectos.

```
SELECT M.Descripcion, COUNT(*) AS Veces_Entregado, SUM(M.Precio * E.Cantidad)  
AS Total_Costo
```

```
FROM Materiales M
```

```
JOIN Entregan E ON M.Clave = E.Clave
```

```
GROUP BY M.Descripcion;
```

3 rows returned.

'Arena', '2', '103600'

'Block', '2', '34950'

'Cantera amarilla', '2', '53590'