

# Lab 20: Consultas SQL

Instituto Tecnológico y de Estudios Superiores de Monterrey Campus Querétaro

Construcción de software y toma de decisiones

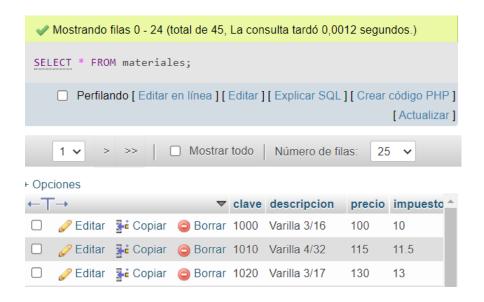
TC2005.402

Eduardo Daniel Juárez Pineda Ricardo Cortés Espinosa

Presenta:

Ricardo Andrés Cáceres Villibord | A01706972

# Consulta una tabla completa



### Selección



# <u>Proyección</u>



#### **Reunión Natural**



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

-> No se mostrarían, ya que es un constraint de que no se muestren los que tienen 0 en la tabla.

### Reunión con Criterio Específico



#### Unión



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



[Editar en línea][Editar][Crear código PHF

25 🗸

# Diferencia (se ilustra con selección )

☐ Mostrar todo Número de filas:

+ Opciones clave 1010



#### Producto cartesiano



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

-> Para cada renglón de Entregan se asignan también los renglones de Materiales.

### Construcción de consultas a partir de una especificación



¿Por qué aparecen varias veces algunas descripciones de material?

-> Porque el mismo material pudo haber sido entregado varias veces en el mismo año.

### Uso del calificador distinct

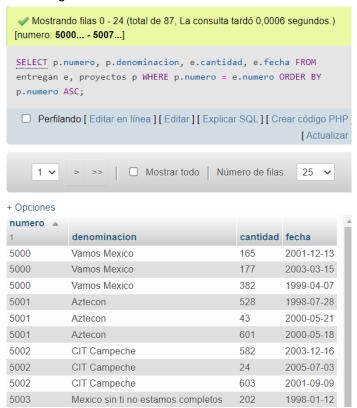


¿Qué resultado obtienes en esta ocasión?

-> Se eliminan los valores repetidos en la búsqueda anterior.

### **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





#### **Uso de Expresiones**

En álgebra relacional los argumentos de una proyección deben ser columnas. Sin embargo en una sentencia SELECT es posible incluir expresiones aritméticas o funciones que usen como argumentos de las columnas de las tablas involucradas o bien constantes. Los operadores son:

- + Suma
- Resta
- \* Producto

/ División

Las columnas con expresiones pueden renombrar escribiendo después de la expresión un alias que puede ser un nombre arbitrario; si el alias contiene caracteres que no sean números o letras (espacios, puntos etc.) debe encerrarse entre comillas dobles (" nuevo nombre" ). Para SQL Server también pueden utilizarse comillas simples.

#### Operadores de cadena



# ¿Qué resultado obtienes?

-> Los materiales en los que la descripción contiene el string "Si"

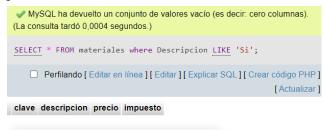
Explica que hace el símbolo '%'.

-> Buscar coincidencias con el string dado. Se pone al final cuando se quiere encontrar al principio del string, y se pone al principio cuando se quiere encontrar en el final del string.

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

-> Devolverá los resultados en donde la descripción sea "Si"

#### ¿Qué resultado obtienes?



Explica a qué se debe este comportamiento.

-> Cuando no se pone el operador "%", se busca que todo el registro sea "Si" es por esto que no devuelve ningún registro porque no hay ningún material con descripción "Si"

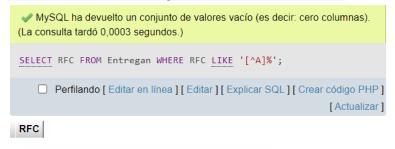
Ahora explica el comportamiento, función y resultado de cada una de las siguientes consultas:

# SELECT RFC FROM Entregan WHERE RFC LIKE [A-D]%':



Esta consulta devuelve todos los registros de RFC de la tabla Entrega donde se contenga como parte del registro letras en el intervalo de A-D. Como no se cumple, no devuelve nada.

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



Devuelve el RFC de la tabla de Entregan, donde los registros no contengan el intervalo de la letra A.

### SELECT Numero FROM Entregan WHERE Numero LIKE '\_\_\_6':



Devuelve los registros de número donde se tenga un espacio de 3 números y después seguidos siempre por un 6.

#### **Between**

### ¿Cómo filtrarías rangos de fechas?



#### **Exists**



#### ¿Qué hace la consulta?

-> Devuelve la consulta haciendo uso de una subconsulta en donde se saca el RFC de los proveedores que su razón social contenga el string "La" en su razón social y donde el RFC de entrega sea el de los proveedores para asegurarse que fueron ellos quienes lo entregaron. ¿Qué función tiene el paréntesis () después de EXISTS?

-> Sirve para diferenciar la subconsulta, ya que esta se debe de tomar como una consulta separada y después se compara con la consulta principal.

# <u>Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador IN</u>



<u>Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo</u> resultado, pero usando el operador NOT IN

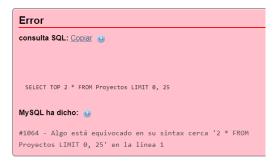


Realiza un ejemplo donde apliques algún operador : ALL, SOME o ANY.



¿Qué hace la siguiente sentencia? Explica por qué.

SELECT TOP 2 \* FROM Proyectos



En MySQL la función de Top es LIMIT.



Lo que hace el LIMIT es limitar los resultados que se obtienen a solo los primeros n registros.

## ¿Qué sucede con la siguiente consulta? Explica por qué.

SELECT TOP Numero FROM Proyectos



Nos marca un error porque número no es un valor numérico. Número en proyectos solo es un dato almacenado, por lo tanto el LIMIT no puede funcionar de manera correcta.

### Modificando la estructura de una tabla existente.

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

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0188 segundos.)

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

[Editar en línea] [Editar] [Crear código PHP]
```

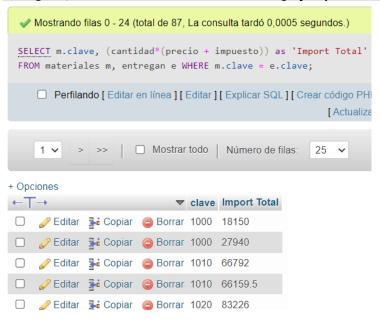
UPDATE materiales SET Porcentajelmpuesto = 2\*clave/1000;

```
✓ 45 filas afectadas. (La consulta tardó 0,0043 segundos.)

UPDATE materiales SET PorcentajeImpuesto = 2*clave/1000;

[Editar en línea] [Editar] [Crear código P
```

¿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?



Comprueba lo anterior, creando vistas para cinco de las consultas que planteaste anteriormente en la práctica. Posteriormente revisa cada vista creada para comprobar que devuelve el mismo resultado.

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0052 segundos.)

CREATE VIEW vista1 AS SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%';
```

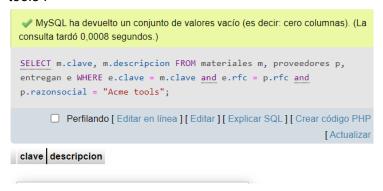


# En el reporte incluye la sentencia, una muestra de la salida (dos o tres renglones) y el número de renglones que SQL Server reporta al final de la consulta.

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



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



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



El Total entregado por cada material en el año 2000.



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



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



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



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 (Solo usando vistas).



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).

