# Instituto Tecnológico y Estudios Superiores de Monterrey

Maria Elizabeth Orduña Toledo

A01424302

Profesor: Ricardo Cortés Espinoza

Materia: Construcción de software y toma de decisiones

Fecha de entrega: 13 de marzo del 2025



## Laboratorio 20: Consultas en SQL

## Consulta de un tabla completa

## Algebra relacional

materiales

**SQL** 

SELECT \* FROM materiales;

✓ Mostrando filas 0 - 24 (total de 45, La consulta tardó 0,0005 segundos.)

←Τ	_→		~	clave	descripcion	precio	impuesto
	🥜 Editar	<b>≩</b> Copiar	Borrar	1000	Varilla 3/16	100	10
		<b>≩</b> Copiar	Borrar	1010	Varilla 4/32	115	11.5
	🥟 Editar	<b>≩</b> Copiar	Borrar	1020	Varilla 3/17	130	13

## Selección

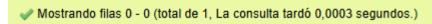
## Algebra relacional

SL{clave=1000}(materiales)

**SQL** 

SELECT \* FROM materiales

WHERE clave=1000





## Proyección

## Algebra relacional

PR{clave,rfc,fecha} (entregan)

#### SQL

## SELECT clave,rfc,fecha FROM entregan

✓ Mostrando filas 0 - 24 (total de 87, La consulta tardó 0,0003 segundos.)

clave	rfc	fecha
1000	AAAA800101	2001-12-13
1000	AAAA800101	1999-07-13
1010	BBBB800101	1998-07-28

#### Reunión Natural

## Algebra relacional

entregan JN materiales

## **SQL**

SELECT \* FROM materiales, entregan

WHERE materiales.clave = entregan.clave

✓ Mostrando filas 0 - 24 (87 en total, 0 en la consulta, La consulta tardó 0,0004 segundos.)

clave	descripcion	precio	impuesto	clave	rfc	numero	fecha	cantidad
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, si algún material no se ha entregado, no aparecería en el resultado de esta consulta. Si hay un material que nunca ha sido entregado, este no tendrá ningún registro correspondiente en la tabla "entregan". Por lo tanto, no habrá ninguna coincidencia para ese material específico y no aparecerá en los resultados de esta consulta.

## Reunión con criterio específico

## Algebra relacional

entregan JN{entregan.numero <= proyectos.numero} proyectos

#### **SQL**

SELECT \* FROM entregan, proyectos

WHERE entregan.numero <= proyectos.numero

## ✓ Mostrando filas 0 - 24 (45 en total, 0 en la consulta, La consulta tardó 0,0006 segundos.)

clave	rfc	numero	fecha	cantidad	numero	denominacion
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)

## Algebra relacional

SL{clave=1450}(entregan) UN SL{clave=1300}(entregan)

#### **SQL**

(SELECT \* FROM entregan where clave=1450)

union

(SELECT \* from entregan where clave=1300)

✓ Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0006 segundos.)

clave	rfc	numero	fecha	cantidad
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? Compruébalo.

SELECT \*

FROM entregan

WHERE clave = 1450 OR clave = 1300

## Intersección (se ilustra junto con selección y proyección)

## Algebra relacional

PR{clave}(SL{numero=5001}(entregan)) IN PR{clave}(SL{numero=5018}(entregan))

## **SQL**

(SELECT clave FROM entregan WHERE numero=5001)

intersect

(SELECT clave FROM entregan WHERE numero=5018)

-- Sin intersect

SELECT clave

FROM entregan

WHERE numero IN (5001, 5018)

**GROUP BY clave** 

HAVING COUNT(DISTINCT numero) = 2

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0006 segundos.)

clave

1010

## Diferencia (se ilustra con selección )

## Algebra relacional

entregan - SL{clave=1000}(entregan)

## SQL

(SELECT \* FROM entregan)

minus

(SELECT \* FROM entregan where clave=1000)

-- Sin minus

SELECT \*

FROM entregan

WHERE clave NOT IN (1000)

✓ Mostrando filas 0 - 24 (total de 85, La consulta tardó 0,0004 segundos.)

clave	rfc	numero	fecha	cantidad
1010	BBBB800101	5001	1998-07-28	528
1010	BBBB800101	5018	1997-02-09	523
1020	CCCC800101	5002	2003-12-16	582

## Producto cartesiano

## Algebra relacional

entregan X materiales

## **SQL**

SELECT \* FROM entregan, materiales

## ✓ Mostrando filas 0 - 3914 (total de 3915, La consulta tardó 0,0039 segundos.)

clave	rfc	numero	fecha	cantidad	clave	descripcion	precio	impuesto
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
1000	AAAA800101	5000	2001-12-13	165	1030	Varilla 4/33	145	14.5

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

El número de tuplas en el resultado de un producto cartesiano está definido como el producto del número de tuplas de cada tabla participante.

Si la tabla "entregan" tiene M tuplas y la tabla "materiales" tiene N tuplas, entonces el resultado del producto cartesiano (entregan × materiales) tendrá M × N tuplas.

## Construcción de consultas a partir de una especificación

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

Recuerda que la fecha puede indicarse como '01-JAN-2000' o '01/01/00'.

Importante: Recuerda que cuando vayas a trabajar con fechas, antes de que realices tus consultas debes ejecutar la instrucción "set dateformat dmy". Basta con que la ejecutes una sola vez para que el manejador sepa que vas a trabajar con ese formato de fechas.

#### **SQL**

SELECT M.descripcion

FROM entregan E, materiales M

WHERE E.clave = M.clave

AND E.fecha BETWEEN '2000-01-01' AND '2000-12-31'

## ¿Por qué aparecen varias veces algunas descripciones de material?

Es porque el mismo material se pudo entregar más de una vez al mismo o diferentes proyectos.

#### 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.

#### **SQL**

SELECT DISTINCT (M.descripcion)

FROM entregan E, materiales M

WHERE E.clave = M.clave

AND E.fecha BETWEEN '2000-01-01' AND '2000-12-31'

## ¿Qué resultado obtienes en esta ocasión?

Los mismos materiales pero sin las repeticiones.

✓ Mostrando filas 0 - 9 (total de 10, La consulta tardó 0,0002 segundos.)

descripcion

Varilla 3/17

Varilla 4/34

Block

Sillar gris

Cantera blanca

Recubrimiento P1028

Tubería 3.6

Pintura C1010

Pintura B1021

## **Ordenamientos**

Pintura B1022

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.

## **SQL**

SELECT P.numero, P.denominacion, E.fecha, E.cantidad FROM proyectos P, entregan E WHERE P.numero = E.numero

#### ORDER BY P.numero, E.fecha DESC;

Mostrando filas 0 - 24 (total de 87, La consulta tardó 0,0007 segundos.) [numero: 5000... - 5007...] [fecha: 2003-03-15... - 2005-06-06...]

numero 🔺 1	denominacion	fecha	cantidad
5000	Vamos Mexico	2003-03-15	177
5000	Vamos Mexico	2001-12-13	165
5000	Vamos Mexico	1999-04-07	382
5001	Aztecon	2000-05-21	43
5001	Aztecon	2000-05-18	601

## Uso de expresiones

#### **SQL**

SELECT clave, precio \* impuesto AS "precio con IVA" FROM materiales;

✓ Mostrando filas 0 - 24 (total de 45, La consulta tardó 0,0001 segundos.)

clave	precio con IVA
1000	1000
1010	1322.5
1020	1690
1030	2102.5

## Operadores de cadena

## SQL

SELECT \* FROM productos where Descripcion LIKE 'Si%'

✓ Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0001 segundos.)

clave	descripcion	precio	impuesto
1120	Sillar rosa	100	10
1130	Sillar gris	110	11

## Explica que hace el símbolo '%'.

El símbolo '%' es un comodín que representa cero o más caracteres en cualquier posición.

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

Esta consulta solo devolvería registros donde "Descripcion" sea exactamente igual a "Si"

## ¿Qué resultado obtienes?

#### **SQL**

SELECT \* FROM materiales where descripcion LIKE 'Si'

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0001 segundos.)

#### Explica a qué se debe este comportamiento.

El comportamiento es diferente porque sin el comodín %, LIKE funciona de manera similar a un operador de igualdad (=) para cadenas de texto.

## **SQL**

```
SELECT (Apellido + ', ' + Nombre) as Nombre FROM Personas;
```

```
DECLARE @foo varchar(40);

DECLARE @bar varchar(40);

SET @foo = '¿Que resultado';

SET @bar = '¿¿¿???' '

SET @foo += ' obtienes?';

PRINT @foo + @bar;
```

#### ¿Para qué sirve DECLARE?

DECLARE se utiliza para declarar variables locales en SQL, especificando el nombre y tipo de datos.

#### ¿Cuál es la función de @foo?

@foo es una variable local (identificada por el prefijo @) que almacena valores temporales durante la ejecución del script.

## ¿Que realiza el operador SET?

El operador SET asigna un valor a una variable previamente declarada.

#### **SQL**

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

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0001 segundos.)
SELECT RFC FROM Entregan WHERE RFC LIKE '[A-D]%';
```

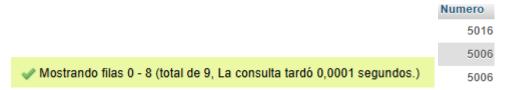
Esta consulta devuelve los RFC de la tabla "Entregan" donde el RFC comienza con cualquier letra entre A y D (A, B, C o D) seguida de cualquier cantidad de caracteres. El operador [A-D] especifica un rango de caracteres permitidos.

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

```
✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0001 segundos.)
SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%';
```

Esta consulta devuelve los RFC que NO comienzan con la letra A. El operador [^A] indica que se excluye el carácter A al inicio del RFC.

SELECT Numero FROM Entregan WHERE Numero LIKE '\_\_\_6';



Esta consulta devuelve los números de la tabla "Entregan" que tienen exactamente 4 caracteres y terminan en 6.

## **Operadores Lógicos**

Para filtrar rangos de fechas con BETWEEN, puedes usar:

#### **SQL**

SELECT clave, rfc, numero, fecha, cantidad

FROM entregan

WHERE fecha BETWEEN '2000-01-01' AND '2000-12-31'

✓ Mostrando filas 0 - 11 (total de 12, La consulta tardó 0,0001 segundos.)

clave	rfc	numero	fecha	cantidad
1020	CCCC800101	5017	2000-03-29	8
1050	FFFF800101	5014	2000-04-18	623
1100	CCCC800101	5009	2000-12-07	466
1130	FFFF800101	5006	2000-04-13	562

#### **SQL**

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])

#### ¿Qué hace la consulta?

Selecciona datos de entregas con números entre 5000 y 5010. Y solo incluye registros donde el proveedor (RFC) tiene una razón social que comienza con "La"

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

El paréntesis después de EXISTS contiene una subconsulta que funciona como una prueba de condición. La subconsulta verifica si existe al menos un registro en la tabla Proveedores que cumpla ambas condiciones: que su razón social comience con "La" y que su RFC coincida con el RFC del registro actual en la tabla Entregan.

#### SQL

#### Misma consulta usando 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%')

#### Misma consulta usando NOT IN

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%')

## Ejemplo con ALL, SOME o ANY

SELECT Clave, Descripcion, Precio
FROM Materiales
WHERE Precio > ALL (SELECT Precio
FROM Materiales
WHERE Clave BETWEEN 1000 AND 2000)

#### ¿Qué hace la siguiente sentencia?

#### **SQL**

SELECT TOP 2 \* FROM Proyectos

Esta consulta devuelve las primeras 2 filas de la tabla Proyectos.

#### ¿Qué hace la siguiente sentencia?

#### **SQL**

SELECT TOP Numero FROM proyectos

Esta consulta no funcionaría correctamente porque TOP espera una constante (número o porcentaje) y no una columna de la tabla.

#### Modificando la estructura de un tabla existente

#### **SQL**

¿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 E.Clave, E.RFC, E.Numero, E.Fecha, E.Cantidad,

M.Precio, M.Impuesto,

E.Cantidad \* M.Precio AS Subtotal,

E.Cantidad \* M.Precio \* (M.Impuesto/100) AS ImporteImpuesto,

E.Cantidad \* M.Precio \* (1 + M.Impuesto/100) AS ImporteTotal

FROM Entregan E

JOIN Materiales M ON E.Clave = M.Clave

Mostrando filas 0 - 24 (total de 87, La consulta tardó 0,0003 segundos.)

Clave	RFC	Numero	Fecha	Cantidad	Precio	Impuesto	Subtotal	ImporteImpuesto	ImporteTotal
1000	AAAA800101	5000	2001-12-13	165	100	10	16500	1650	18150
1000	AAAA800101	5019	1999-07-13	254	100	10	25400	2540	27940.000000000004
1010	BBBB800101	5001	1998-07-28	528	115	11.5	60720	6982.8	67702.8
1010	BBBB800101	5018	1997-02-09	523	115	11.5	60145	6916.675	67061.675
1020	CCCC800101	5002	2003-12-16	582	130	13	75660	9835.800000000001	85495.79999999999

#### Creación de vistas

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

CREATE VIEW MaterialesProyectoMexico AS

SELECT M.clave, M.descripcion

FROM materiales M, entregan E, proyectos P

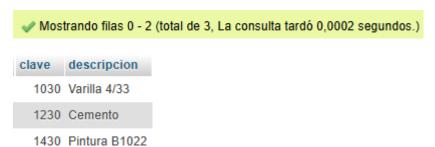
WHERE M.clave = E.clave

AND E.numero = P.numero

AND P.denominacion = 'Mexico sin ti no estamos completos'

#### Verificación

SELECT \* FROM MaterialesProyectoMexico;



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

SQL

CREATE VIEW MaterialesAcmeTools AS
SELECT M.clave, M.descripcion
FROM materiales M, entregan E, proveedores P
WHERE M.clave = E.clave

AND E.rfc = P.rfc

AND P.razonsocial = 'Acme tools';

#### Verificación

SELECT \* FROM MaterialesAcmeTools;

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

#### **SQL**

CREATE VIEW ProveedoresPromedio300 AS

SELECT E.rfc

FROM entregan E

WHERE E.fecha BETWEEN '2000-01-01' AND '2000-12-31

**GROUP BY E.rfc** 

HAVING AVG(E.cantidad) >= 300;

#### Verificación

SELECT \* FROM ProveedoresPromedio300;



## El Total entregado por cada material en el año 2000.

#### SQL

CREATE VIEW TotalPorMaterial2000 AS

SELECT M.clave, M.descripcion, SUM(E.cantidad) AS total\_entregado

FROM materiales M, entregan E

WHERE M.clave = E.clave

AND E.fecha BETWEEN '2000-01-01' AND '2000-12-31'

GROUP BY M.clave, M.descripcion;

#### Verificación

SELECT \* FROM TotalPorMaterial2000;

✓ Mostrando filas 0 - 10 (total de 11, La consulta tardó 0,0007 segundos.)

clave	descripcion	total_entregado
1020	Varilla 3/17	8
1050	Varilla 4/34	623
1100	Block	466
1130	Sillar gris	625
1140	Cantera blanca	583
1210	Recubrimiento P1028	43
1310	Tubería 3.6	72
1360	Pintura C1010	265
1390	Pintura B1021	107
1410	Pintura B1021	601
1430	Pintura B1022	13

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

#### SOL

CREATE VIEW TotalPorMaterial2001 AS

SELECT M.clave, SUM(E.cantidad) AS total\_vendido

FROM materiales M, entregan E

WHERE M.clave = E.clave

AND E.fecha BETWEEN '2000-01-01' AND '2000-12-31'

GROGROUP BY M.clave;

CREATE VIEW MaterialMasVendido2001 AS

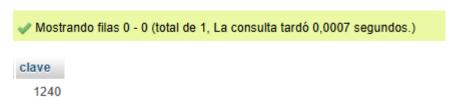
SELECT clave

FROM TotalPorMaterial2001

WHERE total\_vendido = (SELECT MAX(total\_vendido) FROM TotalPorMaterial2001);

#### Verificación

SELECT \* FROM MaterialMasVendido2001;

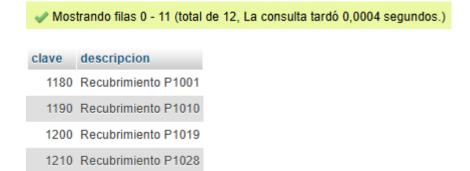


## Productos que contienen el patrón 'ub' en su nombre.

#### **SQL**

CREATE VIEW ProductosConUb AS SELECT clave, descripcion FROM materiales WHERE descripcion LIKE '%ub%';

#### SELECT \* FROM ProductosConUb;



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).

#### **SQL**

CREATE VIEW TotalPorProyecto AS

SELECT P.denominacion, SUM(E.cantidad \* M.precio) AS total\_a\_pagar

FROM proyectos P, entregan E, materiales M

WHERE P.numero = E.numero

AND E.clave = M.clave

GROUP BY P.denominacion;

## SELECT \* FROM TotalPorProyecto;

## 

denominacion	total_a_pagar
Ampliación de la carretera a la huasteca	565135
Aztecon	146595
CIT Campeche	157755
CIT Yucatan	851700

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).

#### **SQL**

);

```
CREATE VIEW Proveedores Televisa Sin Coahuila AS

SELECT P. denominacion, Prov. rfc, Prov. razonsocial

FROM proyectos P, proveedores Prov, entregan E

WHERE P. numero = E. numero

AND Prov. rfc = E. rfc

AND P. denominacion = 'Televisa en acción'

AND Prov. rfc NOT IN (

SELECT DISTINCT Prov2. rfc

FROM proveedores Prov2, entregan E2, proyectos P2

WHERE Prov2. rfc = E2. rfc

AND E2. numero = P2. numero

AND P2. denominacion = 'Educando en Coahuila'
```

## SELECT \* FROM ProveedoresTelevisaSinCoahuila;

✓ Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0009 segundos.)

denominacion	rfc	razonsocial
Televisa en acción	CCCC800101	La Ferre
Televisa en acción	DDDD800101	Cecoferre
Televisa en acción	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.

#### SOL

```
CREATE VIEW MaterialesProveedoresComunes AS

SELECT M.clave, M.descripcion, M.precio, E.cantidad, (M.precio * E.cantidad) AS costo_total

FROM materiales M, entregan E, proyectos P

WHERE M.clave = E.clave

AND E.numero = P.numero

AND P.denominacion = 'Televisa en acción'

AND E.rfc IN (

SELECT DISTINCT E2.rfc

FROM entregan E2, proyectos P2

WHERE E2.numero = P2.numero

AND P2.denominacion = 'Educando en Coahuila'
);
```

#### SELECT \* FROM MaterialesProveedoresComunes;

```
    ✓ Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0005 segundos.)

    clave
    descripcion
    precio
    cantidad
    costo_total

    1080
    Ladrillos rojos
    50
    86
    4300

    1280
    Tepetate
    34
    107
    3638
```

Reto: Usa solo el operador NOT IN en la consulta anterior (No es parte de la entrega).

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

## SQL

```
CREATE VIEW ResumenMateriales AS

SELECT M.descripcion,

COUNT(E.clave) AS veces_entregado,

SUM(E.cantidad * M.precio) AS costo_total

FROM materiales M, entregan E

WHERE M.clave = E.clave

GROUP BY M.descripcion;
```

#### SELECT \* FROM ResumenMateriales;

✓ Mostrando filas 0 - 41 (total de 42, La consulta tardó 0,0005 segundos.)

descripcion	veces_entregado	costo_total
Arena	2	103600
Block	2	34950
Cantera amarilla	2	53590
Cantera blanca	2	160400