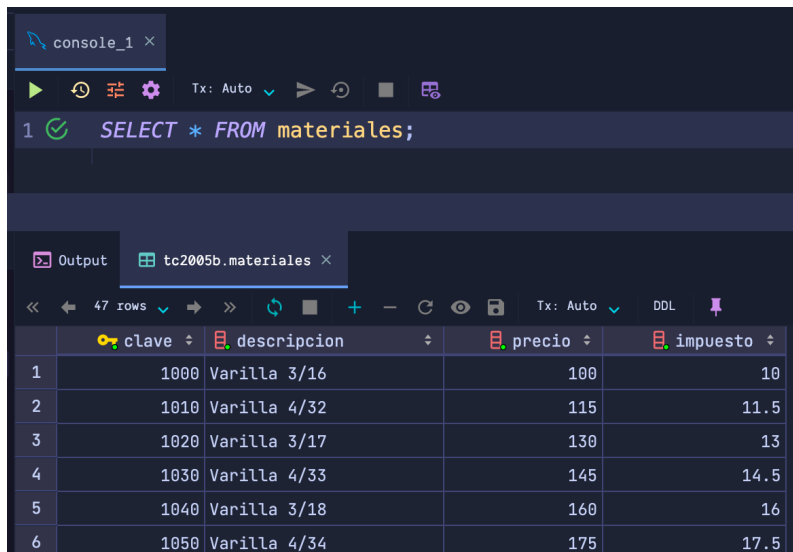


Laboratorio 20

Consultas en SQL

Materiales(Clave, Descripción, Costo)
Proveedores(RFC, RazonSocial)
Proyectos(Numero, Denominacion)
Entregan(Clave, RFC, Numero, Fecha, Cantidad)

Consulta de una tabla completa



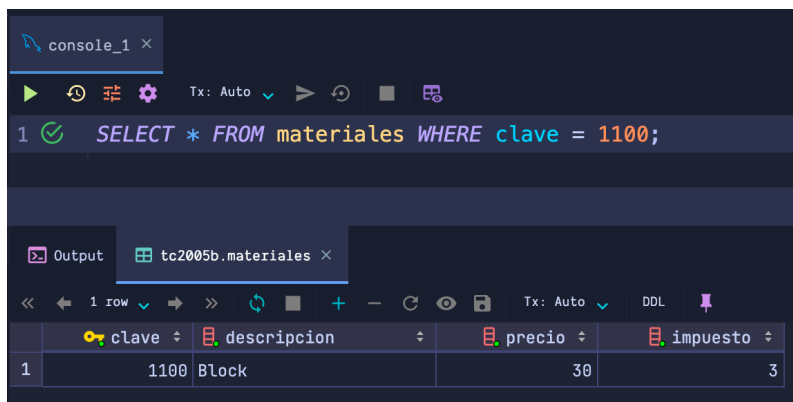
```
1 SELECT * FROM materiales;
```

Output: tc2005b.materiales

47 rows

	clave	descripcion	precio	impuesto
1	1000	Varilla 3/16	100	10
2	1010	Varilla 4/32	115	11.5
3	1020	Varilla 3/17	130	13
4	1030	Varilla 4/33	145	14.5
5	1040	Varilla 3/18	160	16
6	1050	Varilla 4/34	175	17.5

Selección



```
1 SELECT * FROM materiales WHERE clave = 1100;
```

Output: tc2005b.materiales

1 row

	clave	descripcion	precio	impuesto
1	1100	Block	30	3

Proyección

console_1 x

Tx: Auto

```
1 SELECT rfc, fecha FROM entregan;
```

Output tc2005b.entregan x

87 rows

	rfc	fecha
1	AAAA800101	2001-12-13
2	EEEE800101	2003-03-15
3	AAAA800101	1999-04-07
4	BBBB800101	1998-07-28
5	FFFF800101	2000-05-21

Reunión Natural

console_1 x

Tx: Auto

```
1 SELECT * FROM proyectos, entregan WHERE proyectos.numero = entregan.numero;
```

Output Result 15 x

87 rows

	proyectos.numero	denominacion	clave	rfc	entregan.numero	fecha	cantidad
1	5000	Vamos Mexico	1000	AAAA800101	5000	2001-12-13	165
2	5000	Vamos Mexico	1200	EEEE800101	5000	2003-03-15	177
3	5000	Vamos Mexico	1400	AAAA800101	5000	1999-04-07	382
4	5001	Aztecon	1010	BBBB800101	5001	1998-07-28	528
5	5001	Aztecon	1210	FFFF800101	5001	2000-05-21	43
6	5001	Aztecon	1410	BBBB800101	5001	2000-05-18	601
7	5002	CIT Campeche	1020	CCCC800101	5002	2003-12-16	582
8	5002	CIT Campeche	1220	GGGG800101	5002	2005-07-03	24
9	5002	CIT Campeche	1420	CCCC800101	5002	2001-09-09	603
10	5003	Mexico sin ti no estamos completos	1030	DDDD800101	5003	1998-01-12	202
11	5003	Mexico sin ti no estamos completos	1230	HHHH800101	5003	1998-09-12	530
12	5003	Mexico sin ti no estamos completos	1430	DDDD800101	5003	2005-04-30	576
13	5004	Educando en Coahuila	1040	EEEE800101	5004	1999-12-18	263
14	5004	Educando en Coahuila	1150	HHHH800101	5004	2002-11-14	453
15	5004	Educando en Coahuila	1240	AAAA800101	5004	0000-00-00	152

Reunión con criterio específico

console_1 x

Tx: Auto

```
1 SELECT * FROM proyectos, entregan WHERE proyectos.numero <> entregan.numero;
```

Output Result 16 x

1-500 of 501

	proyectos.numero	denominacion	clave	rfc	entregan.numero	fecha	cantidad
1	5000	Vamos Mexico	1000	AAAA800101	5000	2001-12-13	165
2	5000	Vamos Mexico	1200	EEEE800101	5000	2003-03-15	177
3	5000	Vamos Mexico	1400	AAAA800101	5000	1999-04-07	382
4	5001	Aztecon	1000	AAAA800101	5000	2001-12-13	165
5	5001	Aztecon	1200	EEEE800101	5000	2003-03-15	177

Unión

console_1 x

Tx: Auto

```
1 (SELECT * FROM entregan WHERE clave=1100) UNION (SELECT * FROM entregan WHERE clave=1300);
```

Output tc2005b.entregan x

4 rows

	clave	rfc	numero	fecha	cantidad
1	1100	CCCC800101	5009	2000-12-07	466
2	1100	CCCC800101	5010	2001-11-19	699
3	1300	GGGG800101	5005	2004-02-28	521
4	1300	GGGG800101	5010	2001-02-10	119

Diferencia

console_1 x

Tx: Auto

```
1 SELECT * FROM entregan as E1 WHERE clave NOT IN (SELECT clave FROM entregan as E2 WHERE E1.numero != E2.numero)
```

Output tc2005b.entregan x

1 row

	clave	rfc	numero	fecha	cantidad
1	1350	DDDD800101	5015	1997-01-06	272

Producto cartesiano

console_1 x

Tx: Auto

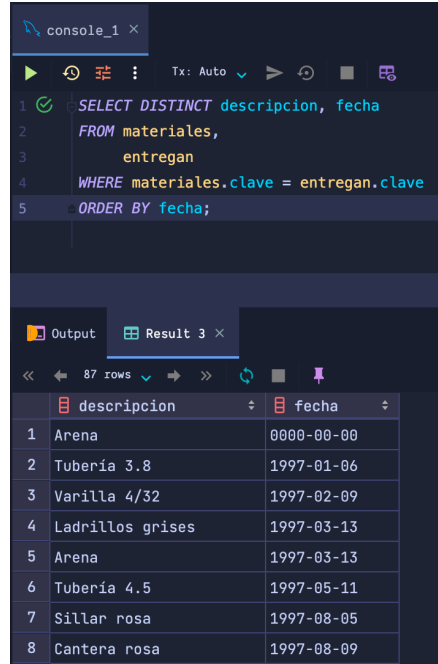
```
1 SELECT * FROM entregan, materiales;
```

Output Result 2 x

1-500 of 501

	entregan.clave	rfc	numero	fecha	cantidad	materiales.clave	descripcion	precio	impuesto
1	1000	AAAA800101	5000	2001-12-13	165	1000	Varilla 3/16	100	10
2	1000	AAAA800101	5000	2001-12-13	165	1010	Varilla 4/32	115	11.5
3	1000	AAAA800101	5000	2001-12-13	165	1020	Varilla 3/17	130	13
4	1000	AAAA800101	5000	2001-12-13	165	1030	Varilla 4/33	145	14.5
5	1000	AAAA800101	5000	2001-12-13	165	1040	Varilla 3/18	160	16
6	1000	AAAA800101	5000	2001-12-13	165	1050	Varilla 4/34	175	17.5
7	1000	AAAA800101	5000	2001-12-13	165	1060	Varilla 3/19	190	19
8	1000	AAAA800101	5000	2001-12-13	165	1070	Varilla 4/35	205	20.5
9	1000	AAAA800101	5000	2001-12-13	165	1080	Ladrillos rojos	50	5
10	1000	AAAA800101	5000	2001-12-13	165	1090	Ladrillos grises	35	3.5
11	1000	AAAA800101	5000	2001-12-13	165	1100	Block	30	3
12	1000	AAAA800101	5000	2001-12-13	165	1110	Megablock	40	4
13	1000	AAAA800101	5000	2001-12-13	165	1120	Sillar rosa	100	10

Ordenamientos.



The screenshot shows a SQL IDE with a query editor and a results pane. The query is:

```
1 SELECT DISTINCT descripcion, fecha
2 FROM materiales,
3     entregan
4 WHERE materiales.clave = entregan.clave
5 ORDER BY fecha;
```

The results pane shows 87 rows. The first 8 rows are displayed in a table with columns 'descripcion' and 'fecha'.

	descripcion	fecha
1	Arena	0000-00-00
2	Tubería 3.8	1997-01-06
3	Varilla 4/32	1997-02-09
4	Ladrillos grises	1997-03-13
5	Arena	1997-03-13
6	Tubería 4.5	1997-05-11
7	Sillar rosa	1997-08-05
8	Cantera rosa	1997-08-09

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 mas caracteres. Este puede usarse tanto de prefijo como sufijo.

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

¿Qué resultado obtienes? Las tuplas de producto donde su descripción empiece con Si
Explica que hace el símbolo '%'. Indica la existencia de cualquier string.

¿Qué sucede si la consulta fuera : LIKE 'Si' ? Desplegaría todos los productos donde su descripción es 'Si'

Explica a qué se debe este comportamiento.

No se indica que exista una cadena de caracteres antes o después del "Si"

Otro operador de cadenas es el de concatenación, (+, +=) este operador concatena dos o más cadenas de caracteres.

Su sintaxis es : Expresión + Expresión.

5 May 2022

Un ejemplo de su uso, puede ser: Un ejemplo de su uso, puede ser:
`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;
```

¿Qué resultado obtienes de ejecutar el siguiente código? ¿¿¿???

¿Para qué sirve **DECLARE**? Asigna una variable

¿Cuál es la función de **@foo**? Referencia el valor de una variable

¿Que realiza el operador **SET**? Asigna el valor de una variable

Sin embargo, tenemos otros operadores como [], [^] y _.

[] - Busca coincidencia dentro de un intervalo o conjunto dado. Estos caracteres se pueden utilizar para buscar coincidencias de patrones como sucede con LIKE.

[^] - En contra parte, este operador coincide con cualquier caracter que no se encuentre dentro del intervalo o del conjunto especificado.

_ - El operador _ o guion bajo, se utiliza para coincidir con un caracter de una comparación de cadenas.

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

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

Despliega todos los RFC de las entregas donde el RFC empieza desde la A a la D.

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

Despliega todos los RFC de las entregas donde empieza con cualquier carácter menos la A.

SELECT Numero FROM Entregan WHERE Numero LIKE '____6';

Despliega entregas cuando su número termina en 6.