



Tecnológico de Monterrey Campus Querétaro

Construcción de software

**Lab 20: Creación de Consultas Utilizando SQL con Funciones
Agregadas y Sub-consultas**

Maestro/a:

Enrique Alfonso Calderon Balderas

Presenta:

Jorge Manuel Oyoqui Aguilera / A01711783

En esta actividad me comprometo a aplicar mis conocimientos, a esforzarme en su desarrollo y a no servirse de medios no autorizados o ilícitos para realizarla.

Fecha de entrega: 12 de marzo de 2025

1. Con base en lo que se explica en la lectura sobre funciones agregadas, plantea y ejecuta las siguientes consultas, agregando los alias de columna necesarios para que los resultados resulten legibles:

- La suma de las cantidades e importe total de todas las entregas realizadas durante el 97.

SELECT

SUM(cantidad) AS Total_Cantidad,

SUM(cantidad * m.precio * (1 + m.PorcentajeImpuesto/100)) AS Importe_Total

FROM entregan e

JOIN materiales m ON e.clave = m.clave

The screenshot shows a SQL IDE with a dark theme. The top panel displays the following SQL query:

```
use tc2005b_db;  
  
SELECT  
    SUM(cantidad) AS Total_Cantidad,  
    SUM(cantidad * m.precio * (1 + m.PorcentajeImpuesto/100)) AS Importe_Total  
FROM entregan e  
JOIN materiales m ON e.clave = m.clave  
WHERE YEAR(e.fecha) = 1997;
```

The bottom panel shows the results in a grid view. The first row contains the following data:

	Total_Cantidad	Importe_Total
1	2.860	2.428.448,0999999996

- Para cada proveedor, obtener la razón social del proveedor, número de entregas e importe total de las entregas realizadas.

```

SELECT

    p.razonsocial AS Razon_Social,

    COUNT(*) AS Numero_Entregas,

    SUM(e.cantidad * m.precio * (1 + m.PorcentajeImpuesto/100)) AS Importe_Total

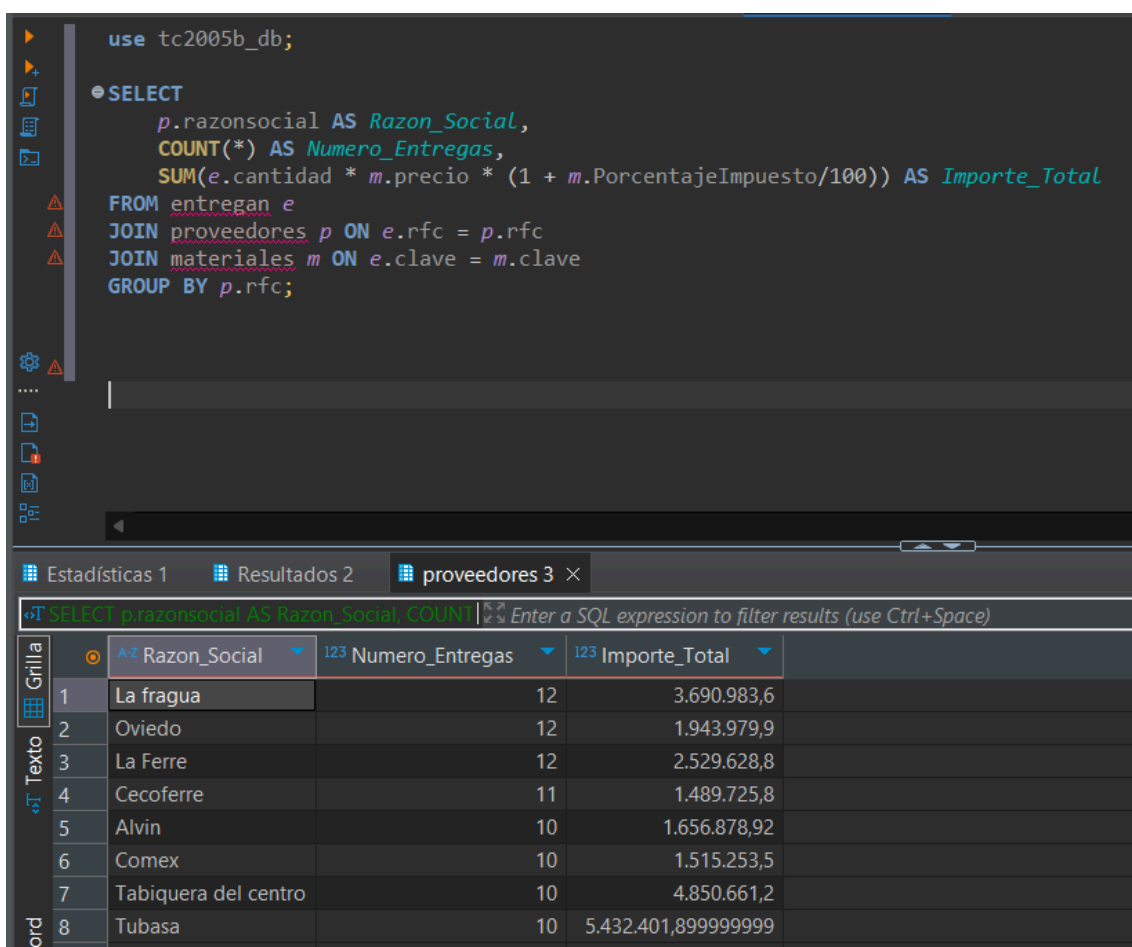
FROM entregan e

JOIN proveedores p ON e.rfc = p.rfc

JOIN materiales m ON e.clave = m.clave

GROUP BY p.rfc;

```



The screenshot shows a SQL IDE interface. The top pane displays the following SQL query:

```

use tc2005b_db;

SELECT
    p.razonsocial AS Razon_Social,
    COUNT(*) AS Numero_Entregas,
    SUM(e.cantidad * m.precio * (1 + m.PorcentajeImpuesto/100)) AS Importe_Total
FROM entregan e
JOIN proveedores p ON e.rfc = p.rfc
JOIN materiales m ON e.clave = m.clave
GROUP BY p.rfc;

```

The bottom pane shows the results in a table view. The table has three columns: Razon_Social, Numero_Entregas, and Importe_Total. The results are as follows:

	Razon_Social	Numero_Entregas	Importe_Total
1	La fragua	12	3.690.983,6
2	Oviedo	12	1.943.979,9
3	La Ferre	12	2.529.628,8
4	Cecoferre	11	1.489.725,8
5	Alvin	10	1.656.878,92
6	Comex	10	1.515.253,5
7	Tabiquera del centro	10	4.850.661,2
8	Tubasa	10	5.432.401,899999999

- Por cada material obtener la clave y descripción del material, la cantidad total entregada, la mínima cantidad entregada, la máxima cantidad entregada, el importe total de las entregas de aquellos materiales en los que la cantidad promedio entregada sea mayor a 400.

SELECT

m.clave,

m.descripcion,

SUM(e.cantidad) AS Total_Entregado,

MIN(e.cantidad) AS Minima_Cantidad,

MAX(e.cantidad) AS Maxima_Cantidad,

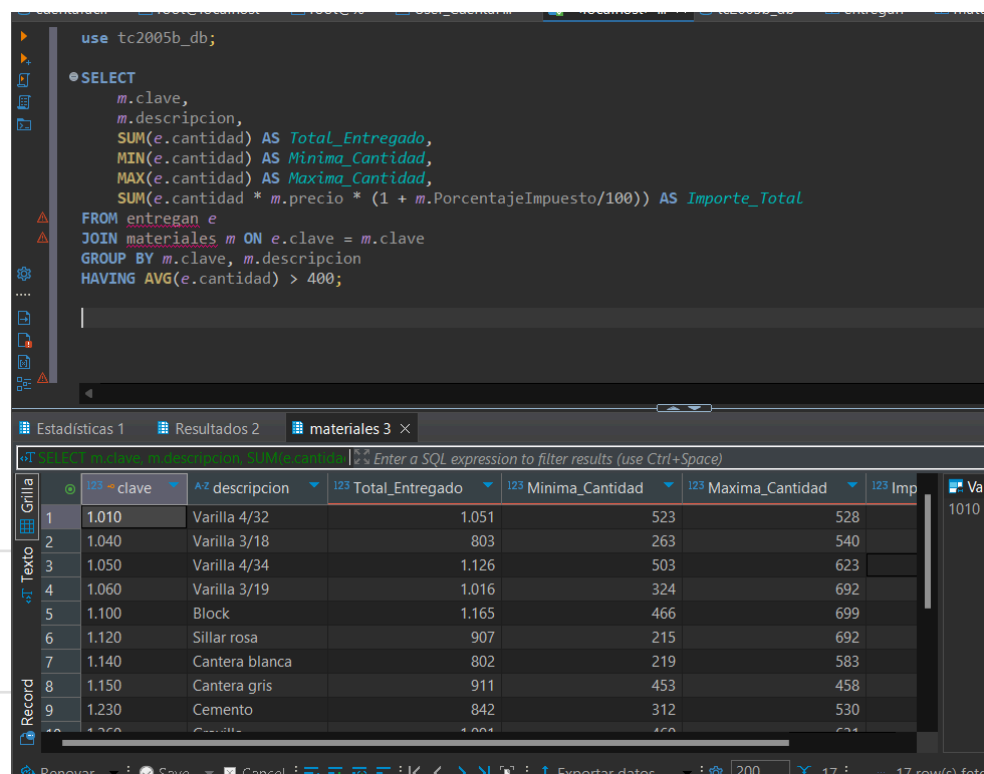
SUM(e.cantidad * m.precio * (1 + m.PorcentajeImpuesto/100)) AS Importe_Total

FROM entregan e

JOIN materiales m ON e.clave = m.clave

GROUP BY m.clave, m.descripcion

HAVING AVG(e.cantidad) > 400;



The screenshot shows a SQL IDE with a query editor and a results grid. The query in the editor is:

```
use tc2005b_db;

SELECT
    m.clave,
    m.descripcion,
    SUM(e.cantidad) AS Total_Entregado,
    MIN(e.cantidad) AS Minima_Cantidad,
    MAX(e.cantidad) AS Maxima_Cantidad,
    SUM(e.cantidad * m.precio * (1 + m.PorcentajeImpuesto/100)) AS Importe_Total
FROM entregan e
JOIN materiales m ON e.clave = m.clave
GROUP BY m.clave, m.descripcion
HAVING AVG(e.cantidad) > 400;
```

The results grid shows 17 rows of data. The columns are: clave, descripcion, Total_Entregado, Minima_Cantidad, Maxima_Cantidad, and Importe_Total. The data is as follows:

	clave	descripcion	Total_Entregado	Minima_Cantidad	Maxima_Cantidad	Importe_Total
1	1.010	Varilla 4/32	1.051	523	528	1010
2	1.040	Varilla 3/18	803	263	540	
3	1.050	Varilla 4/34	1.126	503	623	
4	1.060	Varilla 3/19	1.016	324	692	
5	1.100	Block	1.165	466	699	
6	1.120	Sillar rosa	907	215	692	
7	1.140	Cantera blanca	802	219	583	
8	1.150	Cantera gris	911	453	458	
9	1.230	Cemento	842	312	530	
10	1.260	Gravilla	1.001	450	634	
11	1.270	Gravilla	1.001	450	634	
12	1.280	Gravilla	1.001	450	634	
13	1.290	Gravilla	1.001	450	634	
14	1.300	Gravilla	1.001	450	634	
15	1.310	Gravilla	1.001	450	634	
16	1.320	Gravilla	1.001	450	634	
17	1.330	Gravilla	1.001	450	634	

- Para cada proveedor, indicar su razón social y mostrar la cantidad promedio de cada material entregado, detallando la clave y descripción del material, excluyendo aquellos proveedores para los que la cantidad promedio sea menor a 500.

SELECT

```
p.razonsocial AS Razon_Social,  
m.clave,  
m.descripcion,  
AVG(e.cantidad) AS Promedio_Cantidad
```

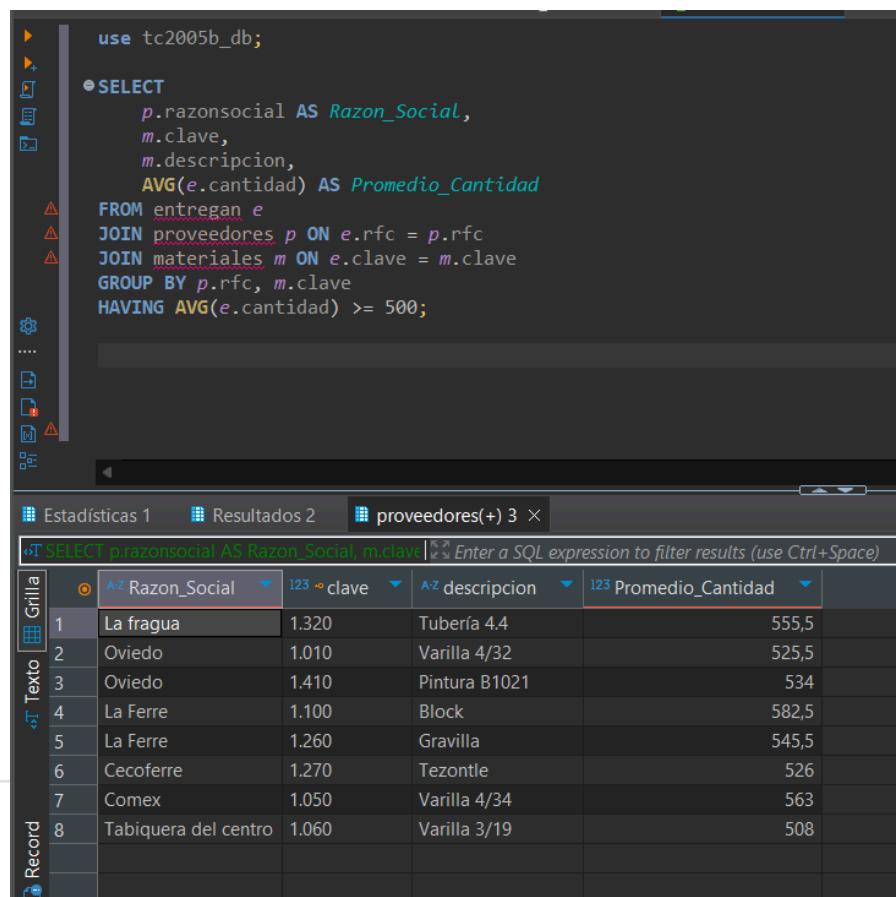
FROM entregan e

JOIN proveedores p ON e.rfc = p.rfc

JOIN materiales m ON e.clave = m.clave

GROUP BY p.rfc, m.clave

HAVING AVG(e.cantidad) >= 500;



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

```
use tc2005b_db;  
  
SELECT  
    p.razonsocial AS Razon_Social,  
    m.clave,  
    m.descripcion,  
    AVG(e.cantidad) AS Promedio_Cantidad  
FROM entregan e  
JOIN proveedores p ON e.rfc = p.rfc  
JOIN materiales m ON e.clave = m.clave  
GROUP BY p.rfc, m.clave  
HAVING AVG(e.cantidad) >= 500;
```

The results pane shows a table with 5 columns: Razon_Social, clave, descripcion, and Promedio_Cantidad. The table contains 8 rows of data.

	Razon_Social	clave	descripcion	Promedio_Cantidad
1	La fragua	1.320	Tubería 4.4	555,5
2	Oviedo	1.010	Varilla 4/32	525,5
3	Oviedo	1.410	Pintura B1021	534
4	La Ferre	1.100	Block	582,5
5	La Ferre	1.260	Gravilla	545,5
6	Cecoferre	1.270	Tezontle	526
7	Comex	1.050	Varilla 4/34	563
8	Tabiquera del centro	1.060	Varilla 3/19	508

- Mostrar en una sola consulta los mismos datos que en la consulta anterior pero para dos grupos de proveedores: aquellos para los que la cantidad promedio entregada es menor a 370 y aquellos para los que la cantidad promedio entregada sea mayor a 450.

```

SELECT
  p.razonsocial AS Razon_Social,
  m.clave,
  m.descripcion,
  AVG(e.cantidad) AS Promedio_Cantidad,
  CASE
    WHEN AVG(e.cantidad) < 370 THEN 'Grupo: <370'
    WHEN AVG(e.cantidad) > 450 THEN 'Grupo: >450'
    ELSE 'Otro'
  END AS Grupo
FROM entregan e
JOIN proveedores p ON e.rfc = p.rfc
JOIN materiales m ON e.clave = m.clave
GROUP BY p.rfc, m.clave
HAVING AVG(e.cantidad) < 370 OR AVG(e.cantidad) > 450;

```

The screenshot shows a SQL IDE interface with a query editor at the top and a results pane at the bottom. The query editor contains the following SQL code:

```

use tc2005b_db;

SELECT
  p.razonsocial AS Razon_Social,
  m.clave,
  m.descripcion,
  AVG(e.cantidad) AS Promedio_Cantidad,
  CASE
    WHEN AVG(e.cantidad) < 370 THEN 'Grupo: <370'
    WHEN AVG(e.cantidad) > 450 THEN 'Grupo: >450'
    ELSE 'Otro'
  END AS Grupo
FROM entregan e
JOIN proveedores p ON e.rfc = p.rfc
JOIN materiales m ON e.clave = m.clave
GROUP BY p.rfc, m.clave
HAVING AVG(e.cantidad) < 370 OR AVG(e.cantidad) > 450;

```

The results pane shows a table with 6 columns: Razon_Social, clave, descripcion, Promedio_Cantidad, and Grupo. The table contains 10 rows of data, grouped by provider (Oviedo and La fragua) and material (various types of bricks and paint).

	Razon_Social	clave	descripcion	Promedio_Cantidad	Grupo
1	La fragua	1.000	Varilla 3/16	209,5	Grupo: <370
2	La fragua	1.160	Cantera rosa	203	Grupo: <370
3	La fragua	1.240	Arena	259	Grupo: <370
4	La fragua	1.320	Tubería 4.4	555,5	Grupo: >450
5	La fragua	1.400	Pintura C1011	249	Grupo: <370
6	Oviedo	1.010	Varilla 4/32	525,5	Grupo: >450
7	Oviedo	1.090	Ladrillos grises	247	Grupo: <370
8	Oviedo	1.170	Cantera amarilla	116,5	Grupo: <370
9	Oviedo	1.330	Tubería 3.7	323,5	Grupo: <370
10	Oviedo	1.410	Pintura B1021	534	Grupo: >450

2. Utilizando la sentencia

INSERT INTO tabla VALUES (valorcolumna1, valorcolumna2, [...], valorcolumnan) ;

Considerando que los valores de tipos CHAR y VARCHAR deben ir encerrados entre apóstrofes, los valores numéricos se escriben directamente y los de fecha, como '1-JAN-00' para 1o. de enero del 2000, inserta cinco nuevos materiales.

```
use tc2005b_db;
INSERT INTO materiales VALUES (2010, 'Malla electrosoldada', 350, 35, 240);
INSERT INTO materiales VALUES (2020, 'Pintura esmalte rojo', 180, 18, 240);
INSERT INTO materiales VALUES (2030, 'Mortero seco', 95, 9.5, 240);
INSERT INTO materiales VALUES (2040, 'Pegazulejo', 150, 15, 240);
INSERT INTO materiales VALUES (2050, 'Polietileno expandido', 80, 8, 240);
SELECT * FROM materiales;
```

The screenshot shows a MySQL IDE interface with a SQL editor and a results grid. The SQL editor contains the following code:

```
use tc2005b_db;
INSERT INTO materiales VALUES (2010, 'Malla electrosoldada', 350, 35, 240);
INSERT INTO materiales VALUES (2020, 'Pintura esmalte rojo', 180, 18, 240);
INSERT INTO materiales VALUES (2030, 'Mortero seco', 95, 9.5, 240);
INSERT INTO materiales VALUES (2040, 'Pegazulejo', 150, 15, 240);
INSERT INTO materiales VALUES (2050, 'Polietileno expandido', 80, 8, 240);
SELECT * FROM materiales;
```

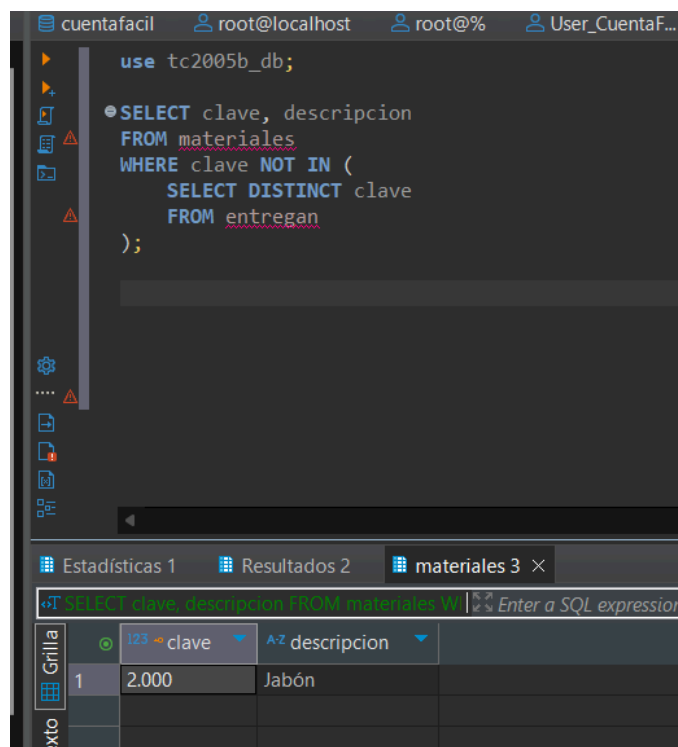
The results grid shows the output of the SQL query. The columns are: clave, descripcion, precio, impuesto, and PorcentajeImpuesto. The data is as follows:

	clave	descripcion	precio	impuesto	PorcentajeImpuesto
40	1.390	Pintura B1021	125	12,5	278
41	1.400	Pintura C1011	125	12,5	280
42	1.410	Pintura B1021	125	12,5	282
43	1.420	Pintura C1012	125	12,5	284
44	1.430	Pintura B1022	125	12,5	286
45	2.000	Jabón	125	12,5	400
46	2.010	Malla electrosoldada	350	35	240
47	2.020	Pintura esmalte rojo	180	18	240
48	2.030	Mortero seco	95	9,5	240
49	2.040	Pegazulejo	150	15	240
50	2.050	Polietileno expandido	80	8	240

3. Con base en lo que se explica en la lectura sobre consultas con roles y subconsultas, plantea y ejecuta las siguientes consultas:

- Clave y descripción de los materiales que nunca han sido entregados.

```
SELECT clave, descripcion
FROM materiales
WHERE clave NOT IN (
    SELECT DISTINCT clave
    FROM entregan
);
```



The screenshot shows a MySQL IDE interface. The top bar displays the database name 'cuentaFacil' and the user 'root@localhost'. The main editor area contains the following SQL query:

```
use tc2005b_db;

SELECT clave, descripcion
FROM materiales
WHERE clave NOT IN (
    SELECT DISTINCT clave
    FROM entregan
);
```

Below the editor, the 'Results' tab is active, showing a table with the following data:

	clave	descripcion
1	2.000	Jabón

- Razón social de los proveedores que han realizado entregas tanto al proyecto 'Vamos México' como al proyecto 'Querétaro Limpio'.

```
SELECT razonsocial
FROM proveedores p
WHERE EXISTS (
    SELECT 1 FROM entregan e
    JOIN proyectos pr ON e.numero = pr.numero
    WHERE e.rfc = p.rfc AND pr.denominacion = 'Vamos México'
)
AND EXISTS (
    SELECT 1 FROM entregan e
    JOIN proyectos pr ON e.numero = pr.numero
    WHERE e.rfc = p.rfc AND pr.denominacion = 'Querétaro Limpio'
);
```

The screenshot shows a MySQL IDE interface. The top toolbar includes icons for database management and user accounts. The main editor displays the following SQL query:

```
use tc2005b_db;

SELECT razonsocial
FROM proveedores p
WHERE EXISTS (
    SELECT 1 FROM entregan e
    JOIN proyectos pr ON e.numero = pr.numero
    WHERE e.rfc = p.rfc AND pr.denominacion = 'Vamos México'
)
AND EXISTS (
    SELECT 1 FROM entregan e
    JOIN proyectos pr ON e.numero = pr.numero
    WHERE e.rfc = p.rfc AND pr.denominacion = 'Querétaro Limpio'
);
```

Below the editor, the 'proveedores 3' tab is active, showing the results of the query. The results are displayed in a table with the following structure:

Grilla	A-Z razonsocial
1	La fragua

The interface also shows a sidebar with icons for various database operations and a bottom status bar with the text: "Enter a SQL expression to filter results (use Ctrl+Space)".

- Descripción de los materiales que nunca han sido entregados al proyecto 'CIT Yucatán'.

```
SELECT clave, descripcion
FROM materiales
WHERE clave NOT IN (
    SELECT DISTINCT e.clave
    FROM entregan e
    JOIN proyectos p ON e.numero = p.numero
    WHERE p.denominacion = 'CIT Yucatán'
);
```

The screenshot shows a database management interface with the following components:

- Top Bar:** Displays the database name 'cuentaafacil', the user 'root@localhost', and the current database 'tc2005b_db'.
- SQL Editor:** Contains the query:


```
use tc2005b_db;
SELECT clave, descripcion
FROM materiales
WHERE clave NOT IN (
    SELECT DISTINCT e.clave
    FROM entregan e
    JOIN proyectos p ON e.numero = p.numero
    WHERE p.denominacion = 'CIT Yucatán'
);
```
- Results Panel:** Shows the results of the query in a grid view. The columns are 'clave' and 'descripcion'. The results are:

	clave	descripcion
1	1.000	Varilla 3/16
2	1.010	Varilla 4/32
3	1.020	Varilla 3/17
4	1.030	Varilla 4/33
5	1.050	Varilla 4/34
6	1.060	Varilla 3/19
7	1.070	Varilla 4/35
8	1.080	Ladrillos rojos
9	1.090	Ladrillos grises
10	1.100	Block
11	1.110	Megablock
12	1.120	Sillar rosa
13	1.130	Sillar gris
14	1.140	Cantera blanca
15	1.160	Cantera rosa
16	1.170	Cantera amarilla
17	1.180	Recubrimiento P100
- Bottom Bar:** Includes buttons for 'Renovar', 'Save', 'Cancel', and 'Exportar datos ...'.

- Razón social y promedio de cantidad entregada de los proveedores cuyo promedio de cantidad entregada es mayor al promedio de la cantidad entregada por el proveedor con el RFC 'VAGO780901'.

```
SELECT
    p.razonsocial,
    AVG(e.cantidad) AS Promedio_Cantidad
FROM entregan e
JOIN proveedores p ON e.rfc = p.rfc
GROUP BY p.rfc
HAVING AVG(e.cantidad) > (
    SELECT AVG(cantidad)
    FROM entregan
    WHERE rfc = 'VAGO780901'
);
```

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
use tc2005b_db;

SELECT
    p.razonsocial,
    AVG(e.cantidad) AS Promedio_Cantidad
FROM entregan e
JOIN proveedores p ON e.rfc = p.rfc
GROUP BY p.rfc
HAVING AVG(e.cantidad) > (
    SELECT AVG(cantidad)
    FROM entregan
    WHERE rfc = 'VAGO780901'
);
```

The results pane shows a table with two columns: 'razonsocial' and 'Promedio_Cantidad'. The table is currently empty, indicating that the query has not yet been executed or that there are no results matching the criteria.

razonsocial	Promedio_Cantidad
-------------	-------------------

- RFC, razón social de los proveedores que participaron en el proyecto 'Infonavit Durango' y cuyas cantidades totales entregadas en el 2000 fueron mayores a las cantidades totales entregadas en el 2001.

```

SELECT
    p.rfc,
    p.razonsocial
FROM proveedores p
WHERE p.rfc IN (
    SELECT e.rfc
    FROM entregan e
    JOIN proyectos pr ON e.numero = pr.numero
    WHERE pr.denominacion = 'Infonavit Durango'
    GROUP BY e.rfc
    HAVING SUM(CASE WHEN YEAR(e.fecha) = 2000 THEN e.cantidad ELSE 0 END) >
           SUM(CASE WHEN YEAR(e.fecha) = 2001 THEN e.cantidad ELSE 0 END)
);

```

The screenshot shows a MySQL IDE interface. The top toolbar includes icons for file operations and a status bar at the bottom. The main editor displays the following SQL query:

```

use tc2005b_db;

SELECT
    p.rfc,
    p.razonsocial
FROM proveedores p
WHERE p.rfc IN (
    SELECT e.rfc
    FROM entregan e
    JOIN proyectos pr ON e.numero = pr.numero
    WHERE pr.denominacion = 'Infonavit Durango'
    GROUP BY e.rfc
    HAVING SUM(CASE WHEN YEAR(e.fecha) = 2000 THEN e.cantidad ELSE 0 END) >
           SUM(CASE WHEN YEAR(e.fecha) = 2001 THEN e.cantidad ELSE 0 END)
);

```

Below the editor, the results are displayed in a table with the title "proveedores 3". The table has three columns: "rfc", "razonsocial", and an empty column. The first row shows the results for the provider "GGGG800101".

	rfc	razonsocial	
1	GGGG800101	Tabiquera del centro	