José Diego Llaca Castro A01704793

Iván Ricardo Paredes Avilez A01705083

### **Laboratorio 21: Creación de Consultas Utilizando SQL con Funciones Agregadas y Sub-consultas**

A modo de referencia, incluimos los esquemas de las tablas que creaste en la práctica anterior y que serán con las que trabajaremos en esta práctica:

**Materiales(Clave, Descripción, Costo, PorcentajeImpuesto)**

**Proveedores(RFC, RazonSocial)**

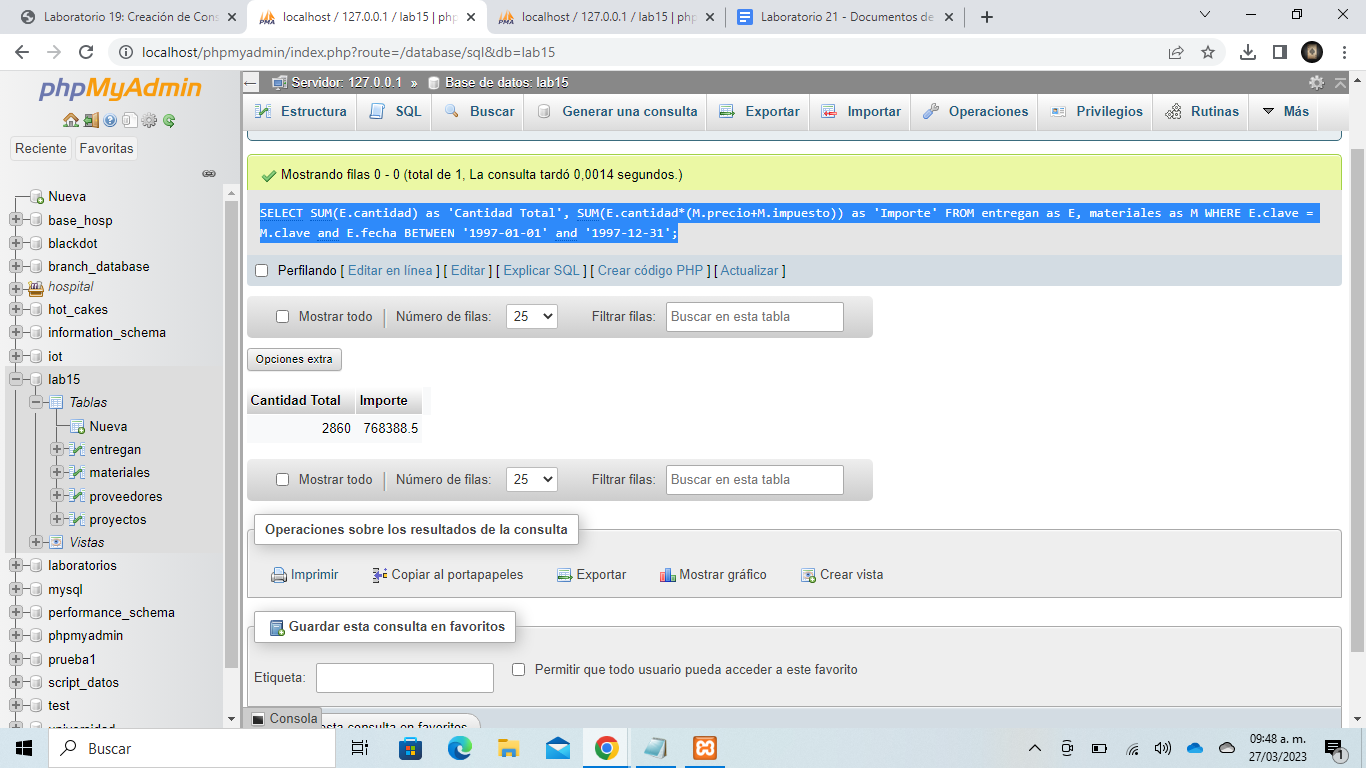
**Proyectos(Numero, Denominacion)**

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

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(E.cantidad) as 'Cantidad Total', SUM(E.cantidad\*(M.precio+M.impuesto)) as 'Importe' FROM entregan as E, materiales as M WHERE E.clave = M.clave and E.fecha BETWEEN '1997-01-01' and '1997-12-31';



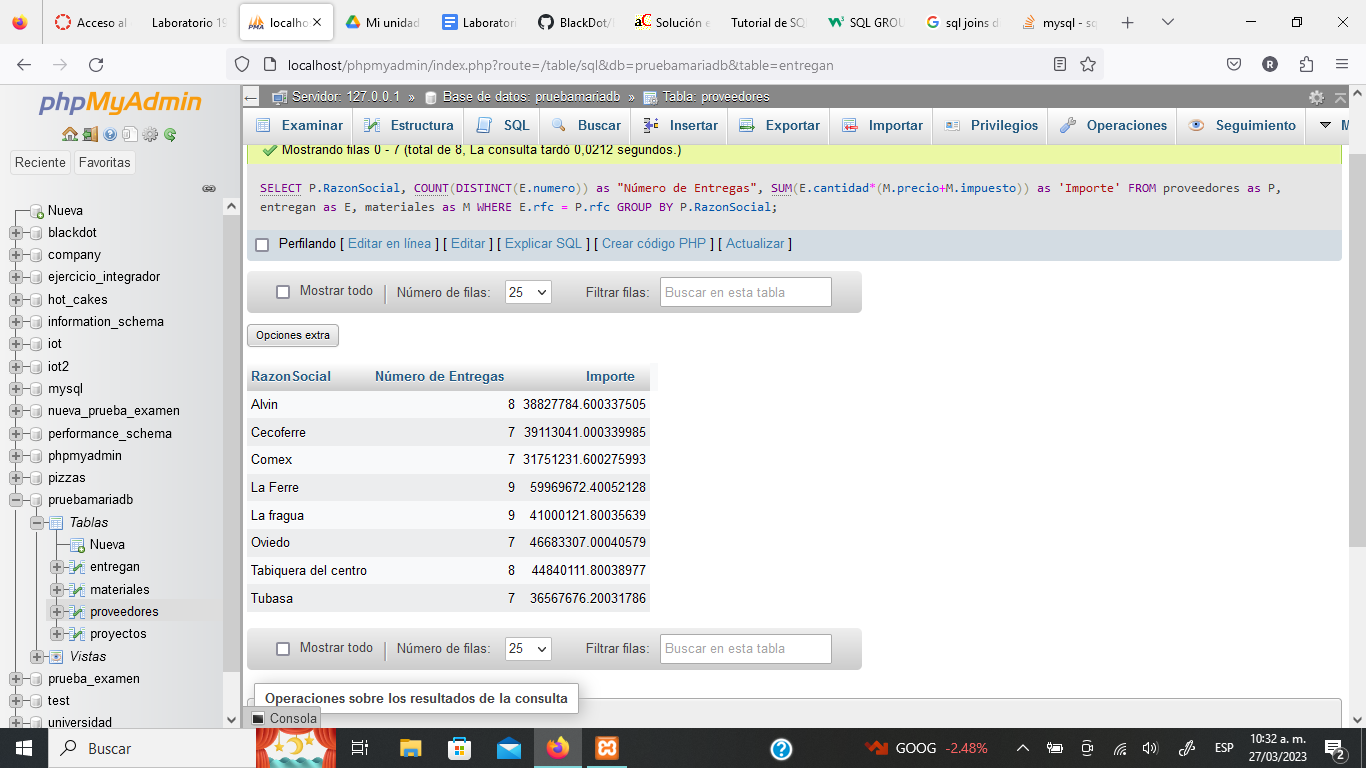
* Para cada proveedor, obtener la razón social del proveedor, número de entregas e importe total de las entregas realizadas.

SELECT P.RazonSocial, COUNT(DISTINCT(E.numero)) as "Número de Entregas", SUM(E.cantidad\*(M.precio+M.impuesto)) as 'Importe'

FROM proveedores as P, entregan as E, materiales as M

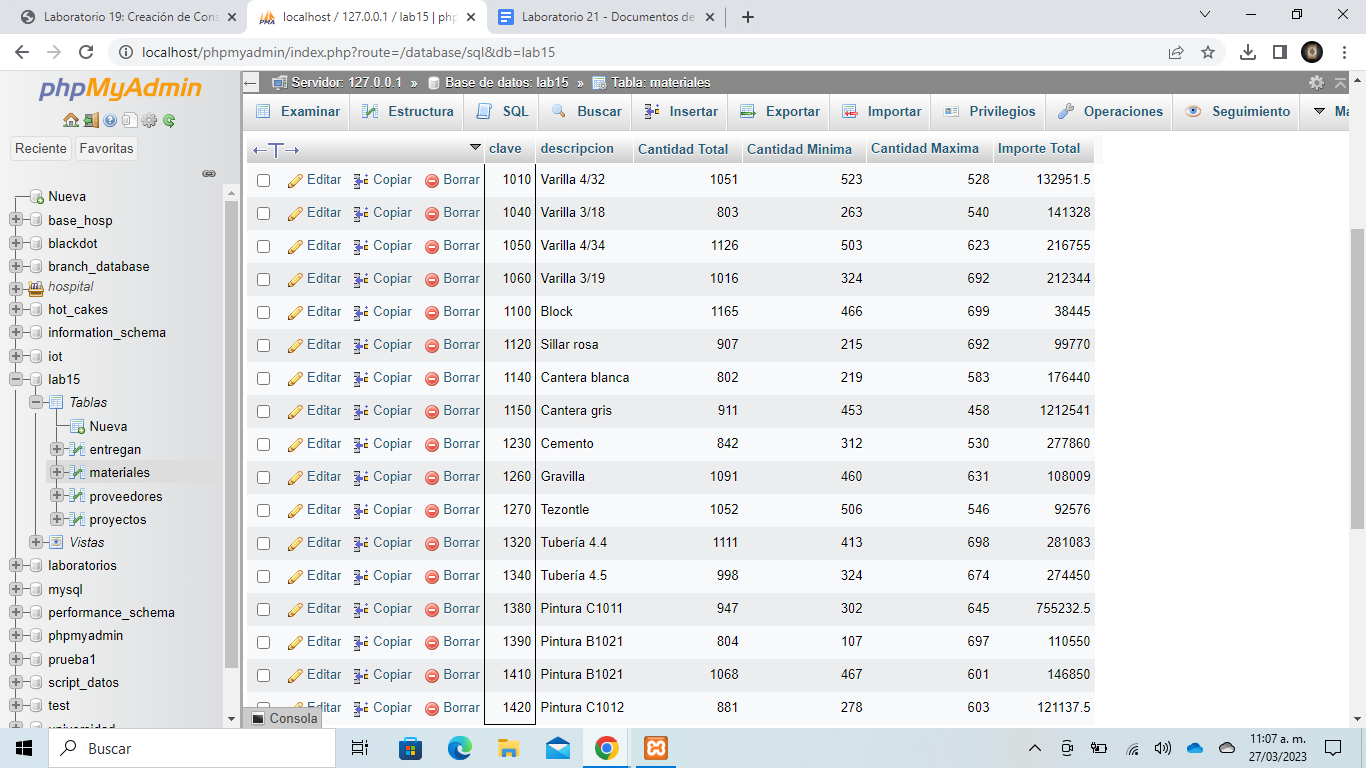
WHERE E.rfc = P.rfc

GROUP BY P.RazonSocial



* 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 'Cantidad Total', MIN(E.cantidad) as 'Cantidad Minima', MAX(E.cantidad) as 'Cantidad Maxima', SUM(E.cantidad\*(M.precio+M.impuesto)) as 'Importe Total' FROM materiales as M, entregan as E WHERE M.clave = E.clave GROUP BY M.clave, M.descripcion HAVING AVG(E.cantidad) > 400;



* 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.

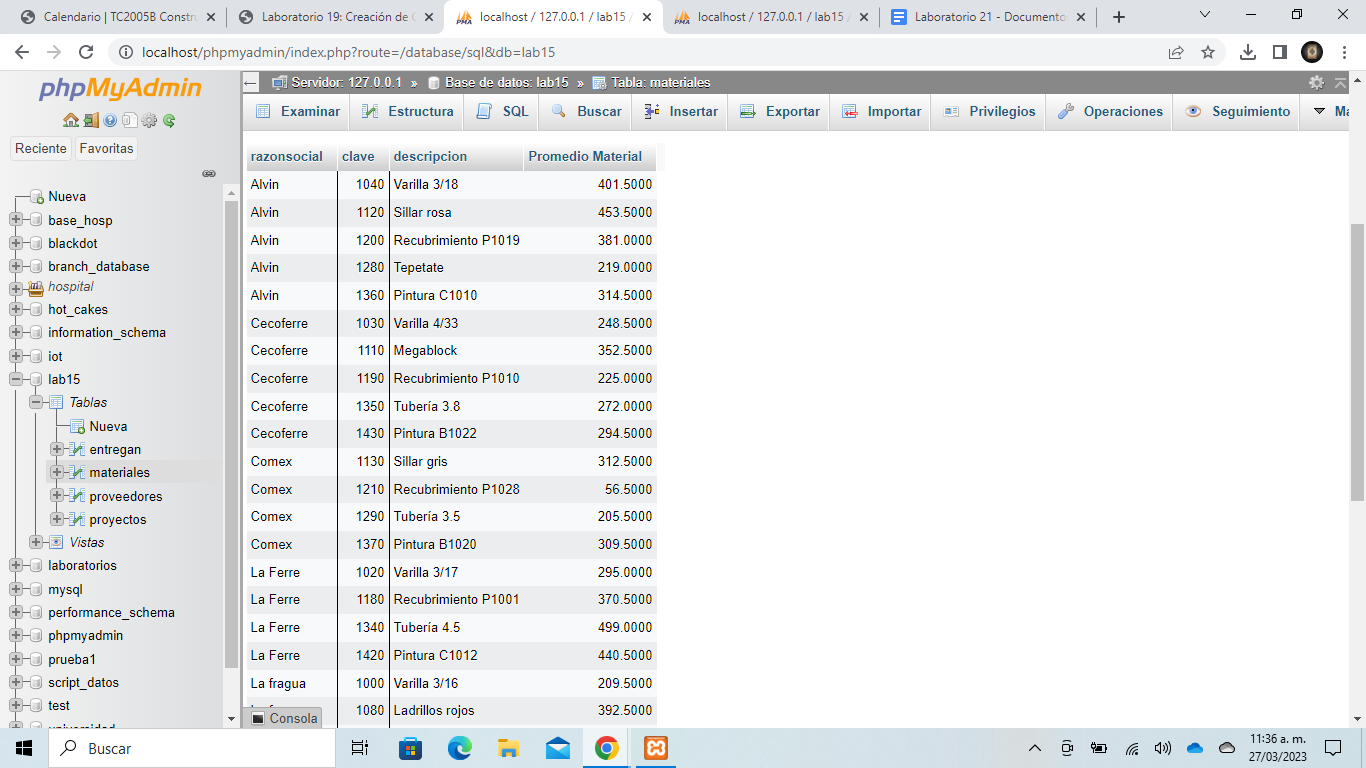
SELECT P.razonsocial, M.clave, M.descripcion, AVG(E.cantidad) as 'Promedio Material'

FROM proveedores as P, materiales as M, entregan as E

WHERE P.rfc = E.rfc and M.clave = E.clave

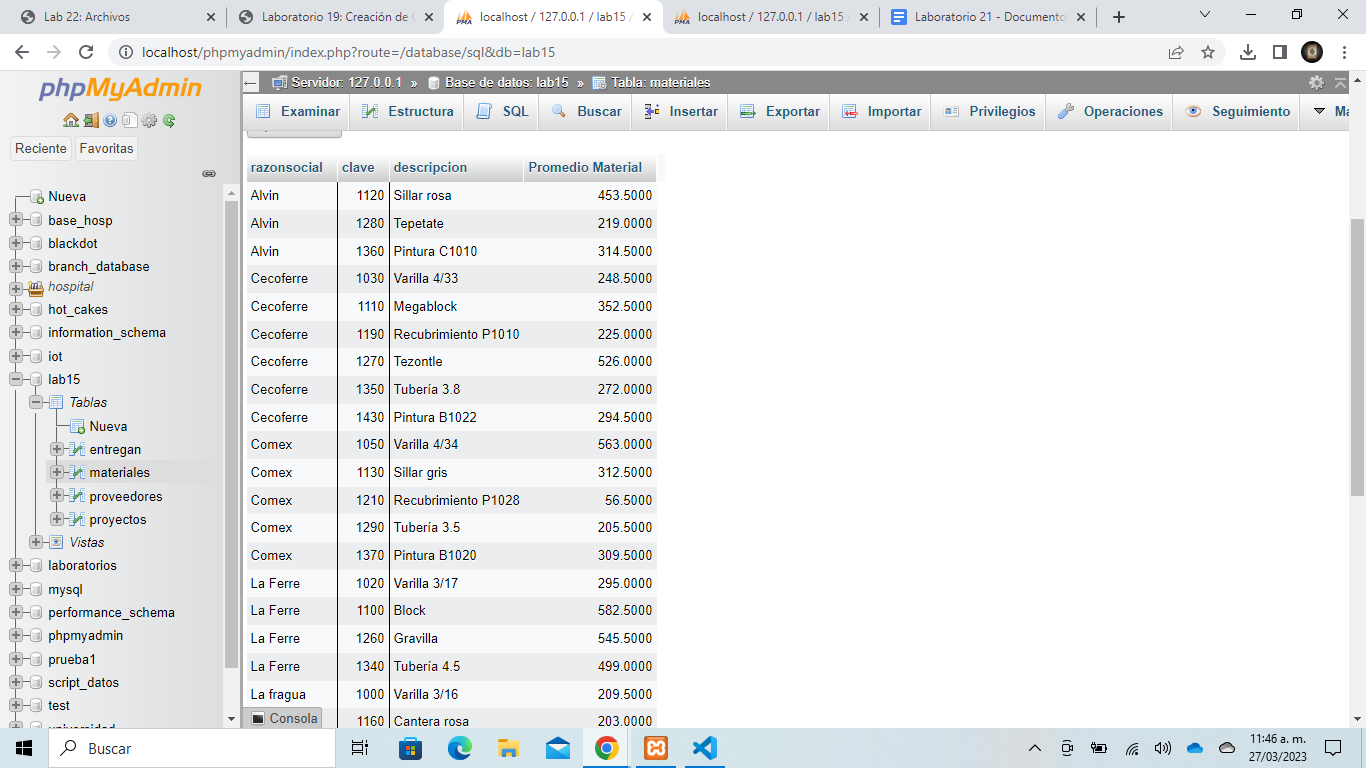
GROUP BY P.razonsocial, M.clave, M.descripcion

HAVING AVG(E.cantidad) < 500;



* Mostrar en una solo 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, M.clave, M.descripcion, AVG(E.cantidad) as 'Promedio Material' FROM proveedores as P, materiales as M, entregan as E WHERE P.rfc = E.rfc and M.clave = E.clave GROUP BY P.razonsocial, M.clave, M.descripcion HAVING AVG(E.cantidad) < 370 OR AVG(E.cantidad) > 450;



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.

INSERT INTO materiales (clave, descripcion, precio, impuesto, PorcentajeImpuesto)

VALUES

(3000, 'Granito', 500, 50.0, 4.02),

(3010, 'Mármol', 1000, 100, 4.04),

(3020, 'Caliza', 700, 70, 4.06),

(3030, 'Pizarra', 800, 80, 4.08),

(3040, 'Arenisca', 300, 30, 4.10);

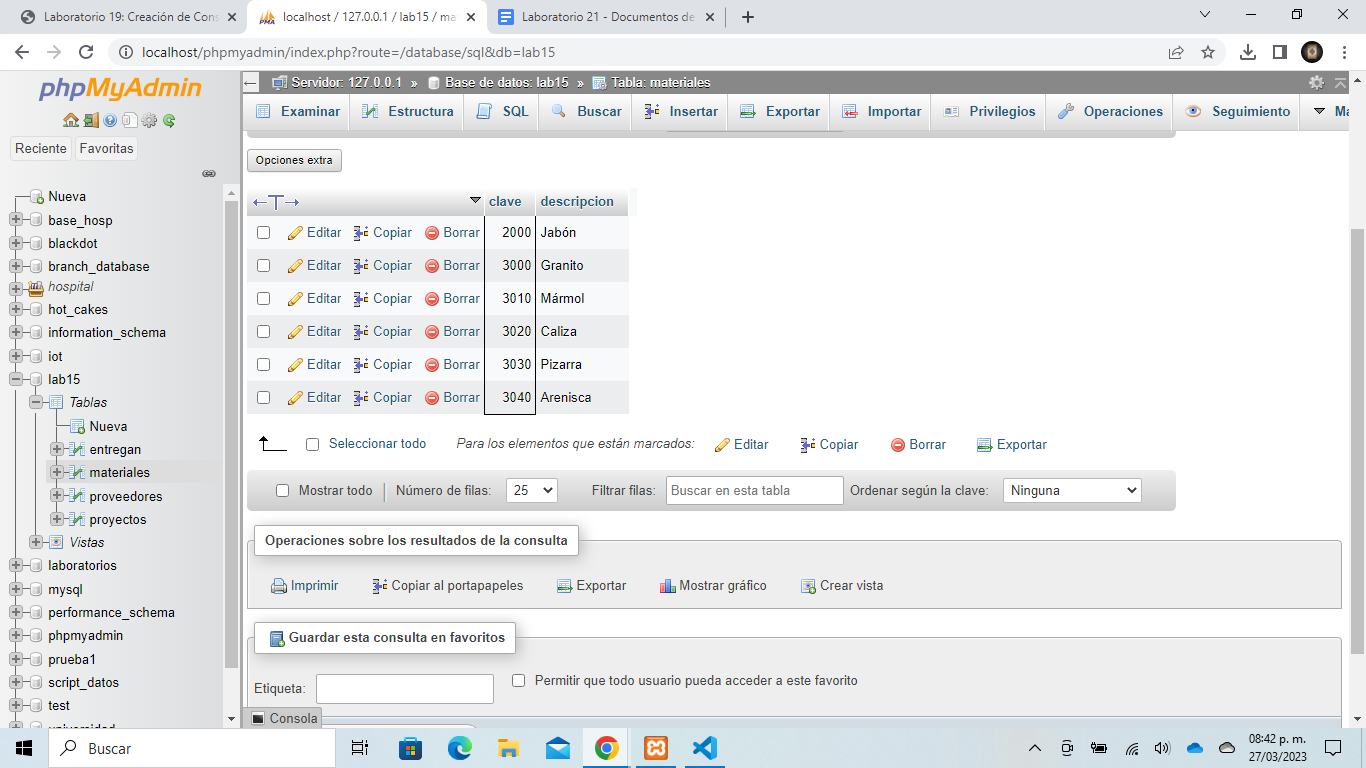
Con base en lo que se explica en la lectura sobre consultas con roles y subconsultas, plantea y ejecuta las siguientes consultas:

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

SELECT M.clave, M.descripcion

FROM materiales as M

WHERE M.clave NOT IN (SELECT E.clave FROM entregan as E);

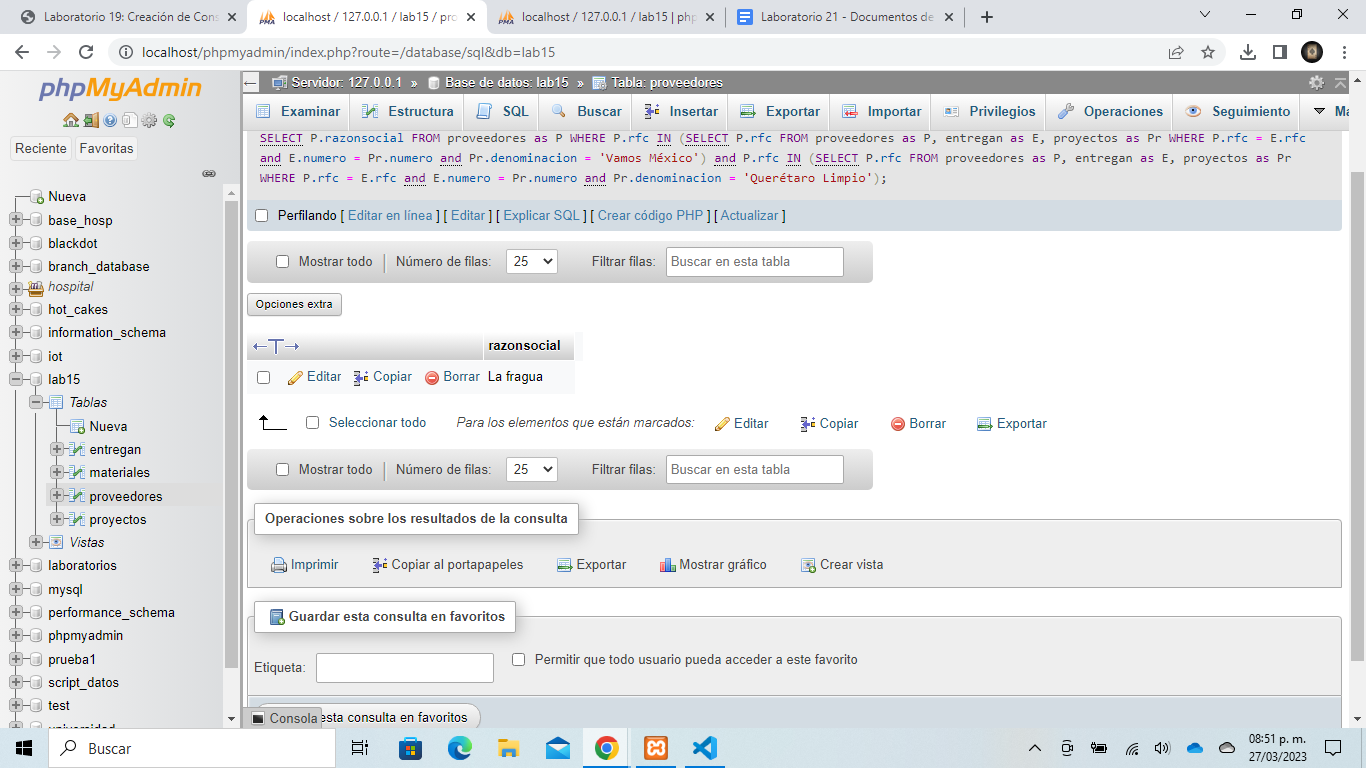


* Razón social de los proveedores que han realizado entregas tanto al proyecto 'Vamos México' como al proyecto 'Querétaro Limpio'.

SELECT P.razonsocial

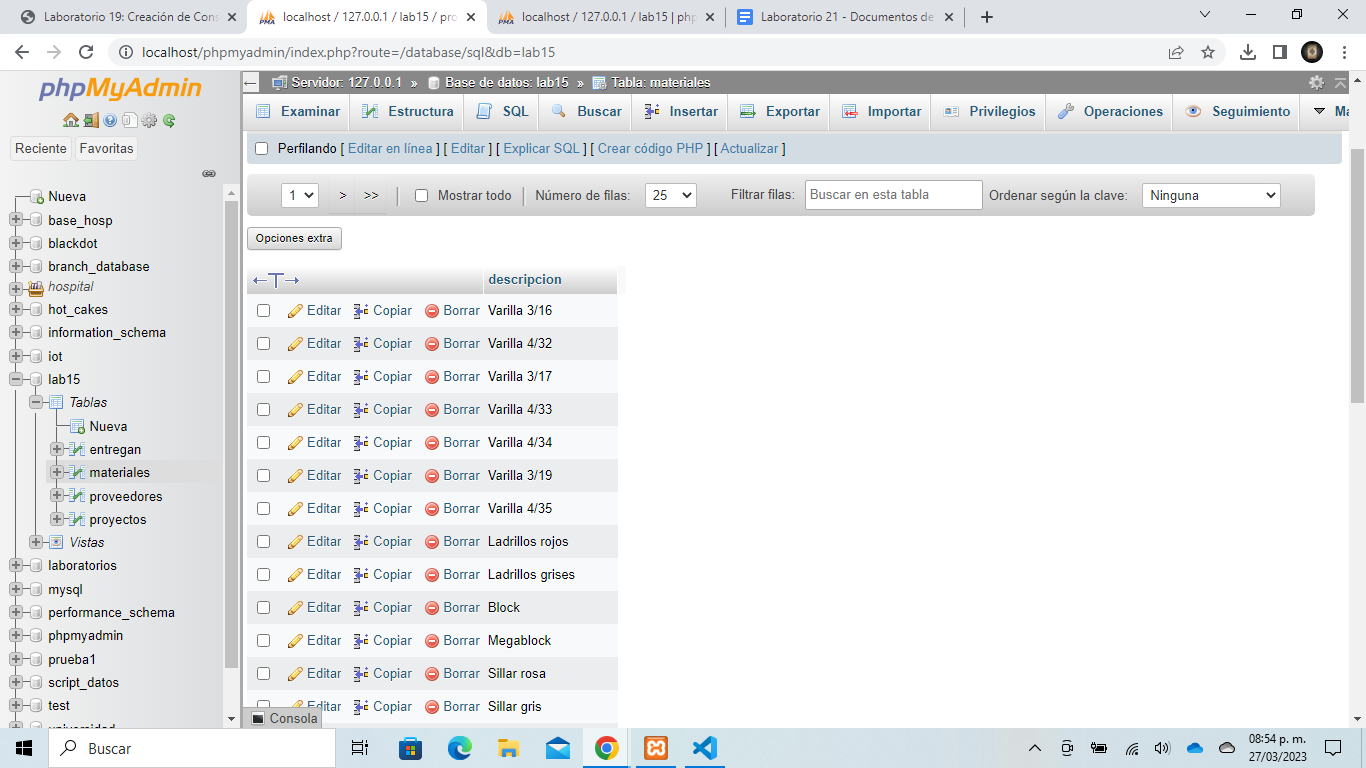
FROM proveedores as P

WHERE P.rfc IN (SELECT P.rfc FROM proveedores as P, entregan as E, proyectos as Pr WHERE P.rfc = E.rfc and E.numero = Pr.numero and Pr.denominacion = 'Vamos México') and P.rfc IN (SELECT P.rfc FROM proveedores as P, entregan as E, proyectos as Pr WHERE P.rfc = E.rfc and E.numero = Pr.numero and Pr.denominacion = 'Querétaro Limpio');



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

SELECT M.descripcion FROM materiales as M WHERE M.clave NOT IN (SELECT E.clave FROM entregan as E, proyectos as P WHERE E.numero = P.numero and P.denominacion = 'CIT Yucatán');



* 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'.

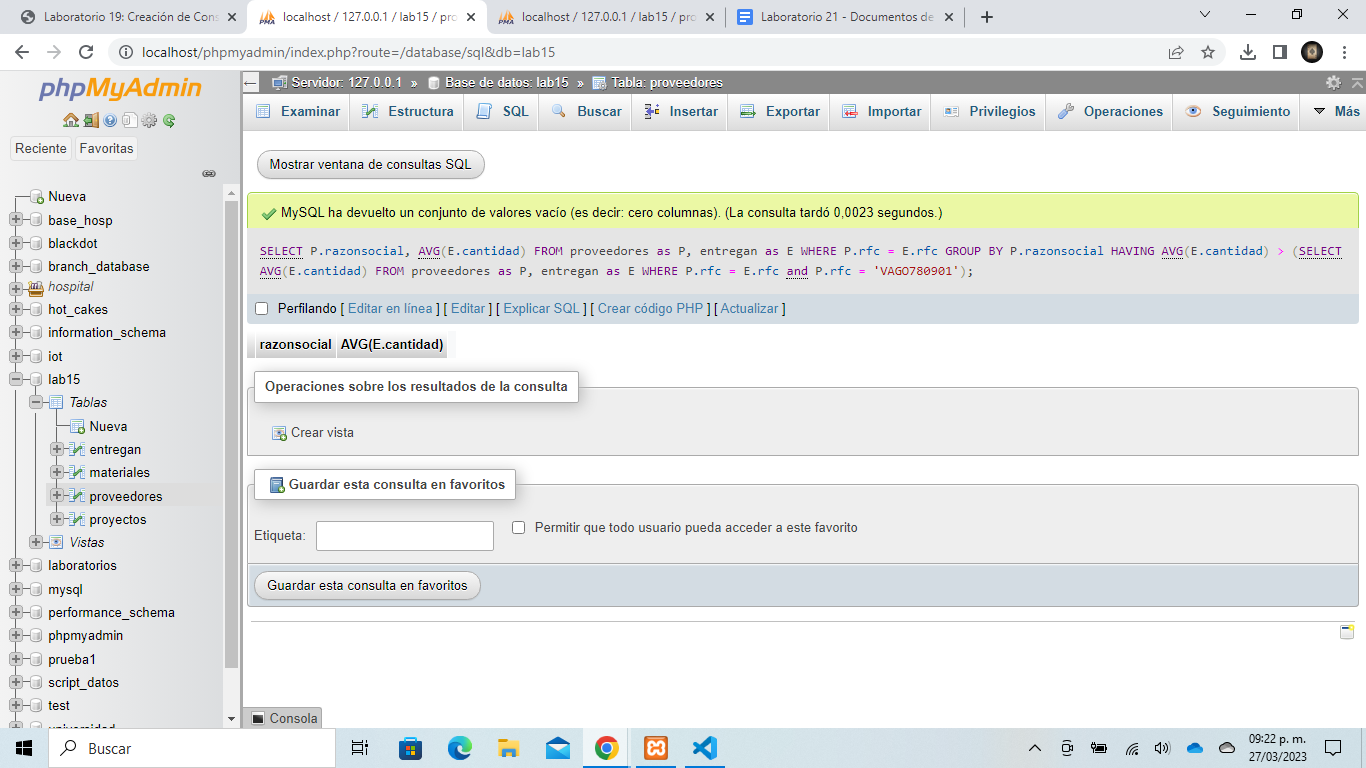
SELECT P.razonsocial, AVG(E.cantidad)

FROM proveedores as P, entregan as E

WHERE P.rfc = E.rfc

GROUP BY P.razonsocial

HAVING AVG(E.cantidad) > (SELECT AVG(E.cantidad) FROM proveedores as P, entregan as E WHERE P.rfc = E.rfc and P.rfc = 'VAGO780901');



* 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.

CREATE VIEW CantidadTotal2001 (rfc, razonSocial, cantidadTotal) as SELECT P.rfc, P.razonsocial, SUM(E.cantidad) FROM proveedores as P, entregan as E, proyectos as Pr WHERE P.rfc = E.rfc and E.numero = Pr.numero and Pr.denominacion = 'Infonavit Durango' and E.fecha BETWEEN '2001-01-01' and '2001-12-31' GROUP BY P.rfc, P.razonsocial;

SELECT P.rfc, P.razonsocial, SUM(E.cantidad) FROM proveedores as P, entregan as E, proyectos as Pr WHERE P.rfc = E.rfc and E.numero = Pr.numero and Pr.denominacion = 'Infonavit Durango' and E.fecha BETWEEN '2000-01-01' and '2000-12-31' GROUP BY P.rfc, P.razonsocial HAVING SUM(E.cantidad) > (SELECT cantidadTotal FROM CantidadTotal2001);

