Investigación sobre DBMS empresariales

Investiga los siguientes aspectos de los siguientes manejadores de base de datos relacionales (Oracle, SQL Server, MySql y un DBMS NO relacional)  
  
- Requerimientos Generales de Hardware

Oracle 12c

* Procesador AMD64 o Intel EM64T
* 2 Gb de RAM
* Conección a Internet
* Adaptador de video de 256 colores con resolución 1024x768
* 10,5 Gb de espacio en disco disponible (6 Gb software + 4,5 Gb DB)

SQL Server 2019

* Memoria :
  + Mínimo:Ediciones Express: 512 MB
  + Las demás ediciones: 1 GB
  + Se recomienda que use:Ediciones Express: 1 G
  + Las demás ediciones: Al menos 4 GB, que debe aumentar a medida que el tamaño de la base de datos aumente para asegurar un rendimiento óptimo.
* Velocidad del procesador Mínimo: Procesador x64: 1,4 GHz
* Se recomienda que use: 2.0 GHz o superior
* Tipo de procesador Procesador x64: AMD Opteron, AMD Athlon 64, Intel Xeon compatible con Intel EM64T, Intel Pentium IV compatible con EM64T

MySql

* 512 Mb de memoria Ram
* 1024 Mb maquina virtual
* 1 GB de espacio de disco duro
* Arquitectura del sistema 32/64 bit
* Protocolo de red TCP/IP

DBMS NO relacional

- Ambientes o plataformas en las que pueden operar.

Oracle

SQL Server 2019

MySql

DBMS NO relacional

- Costos de implementación y mantenimiento.

Oracle

El proceso de implementación de la solución de ODBS, implica cuatro pasos sencillos:

1. Proceso de suscripción y activación del servicio de nube
2. Instalación del módulo de RMAN
3. Configuración de RMAN
4. Realización de respaldo y recuperación de datos.

No existen costos de actualización y soporte de hardware de respaldo

SQL Server 2019

MySql

DBMS NO relacional

- Ventajas y desventajas de su uso.

Oracle

**VENTAJAS**

1**)Oracle es el motor de base de datos objeto-relacional más usado a nivel mundial.**  
 **2)Puede ejecutarse en todas las plataformas, desde una Pc hasta un supercomputador.**  
 **3)Oracle soporta todas las funciones que se esperan de un servidor "serio": un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos", con triggers y procedimientos almacenados, con una integridad referencial declarativa bastante potente.**  
 **4)Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.**  
 **5)El software del servidor puede ejecutarse en multitud de sistemas operativos.**  
 **6)Existe incluso una versión personal para Windows 9x, lo cual es un punto a favor para los desarrolladores que se llevan trabajo a casa.**  
 **7)Oracle es la base de datos con más orientación hacía INTERNET.**

**DESVENTAJAS**

**1)Las versiones más recientes de Oracle son la 11g, 10g, 9g, 8g, desde el lanzamiento original de la 8 se sucedieron varias versiones con correcciones, hasta alcanzar la estabilidad en la 8.0.3. El motivo de tantos fallos fue, al parecer, la remodelación del sistema de almacenamiento por causa de la introducción de extensiones orientadas a objetos.**

**2)El mayor inconveniente de Oracle es quizás su precio. Incluso las licencias de Personal Oracle son excesivamente caras, en mi opinión. Otro problema es la necesidad de ajustes. Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y enchufar directamente las aplicaciones clientes. Un Oracle mal configurado puede ser desesperantemente lento.**

**3)También es elevado el coste de la información, y sólo últimamente han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.**

SQL Server 2019

DESVENTAJAS DE SQL SERVER

1. Utiliza mucho la memoria RAM para las instalaciones y utilización de software.

2. No se puede utilizar como practicas porque se prohíben muchas cosas, tiene restricciones en lo particular.

3. La relación, calidad y el precio esta muy debajo comparado con oracle.

4. Tiene muchos bloqueos a nivel de página, un tamaño de página fijo y demasiado pequeño, una pésima implementación de los tipos de datos variables.

VENTAJAS DE SQL SERVER

Es un sistema de gestión de base de datos.

Es útil para manejar y obtener datos de la red de redes.

Nos permite olvidarnos de los ficheros que forman la base de datos.

Si trabajamos en una red social nos permite agregar otros servidores de SQL Server. Por ejemplo dos personas que trabajan con SQL Server, uno de ellos se puede conectar al servidor de su otro compañero y así se puede ver las bases de datos del otro compañero con SQL Server.

SQL permite administrar permisos a todo. También permite que alguien conecte su SQLO al nuestro pero sin embargo podemos decirle que no puede ver esta base de datos pero otro si.

MySql

DBMS NO relacional

- Porcentaje del mercado que controlan

Oracle

SQL Server 2019

MySql

DBMS NO relacional

Referencias :

<https://docs.microsoft.com/es-es/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server-ver15?view=sql-server-ver15>

<https://www.oracle.com/technetwork/es/articles/database-performance/instal-oracle19c-window-5592361-esa.html>

<http://lopez-garcia-victor.blogspot.com/2012/08/11-requerimientos-e-instalacion-del-sgbd.html>

<http://oraclebddepn.blogspot.com/2013/05/ventajas-y-desventajas.html>

**Reunión Natural**  
  
Algebra relacional.  
entregan JN materiales  
  
SQL  
select \* from materiales,entregan  
where materiales.clave = entregan.clave  
  
Si algún material no ha se ha entregado ¿Aparecería en el resultado de esta consulta?

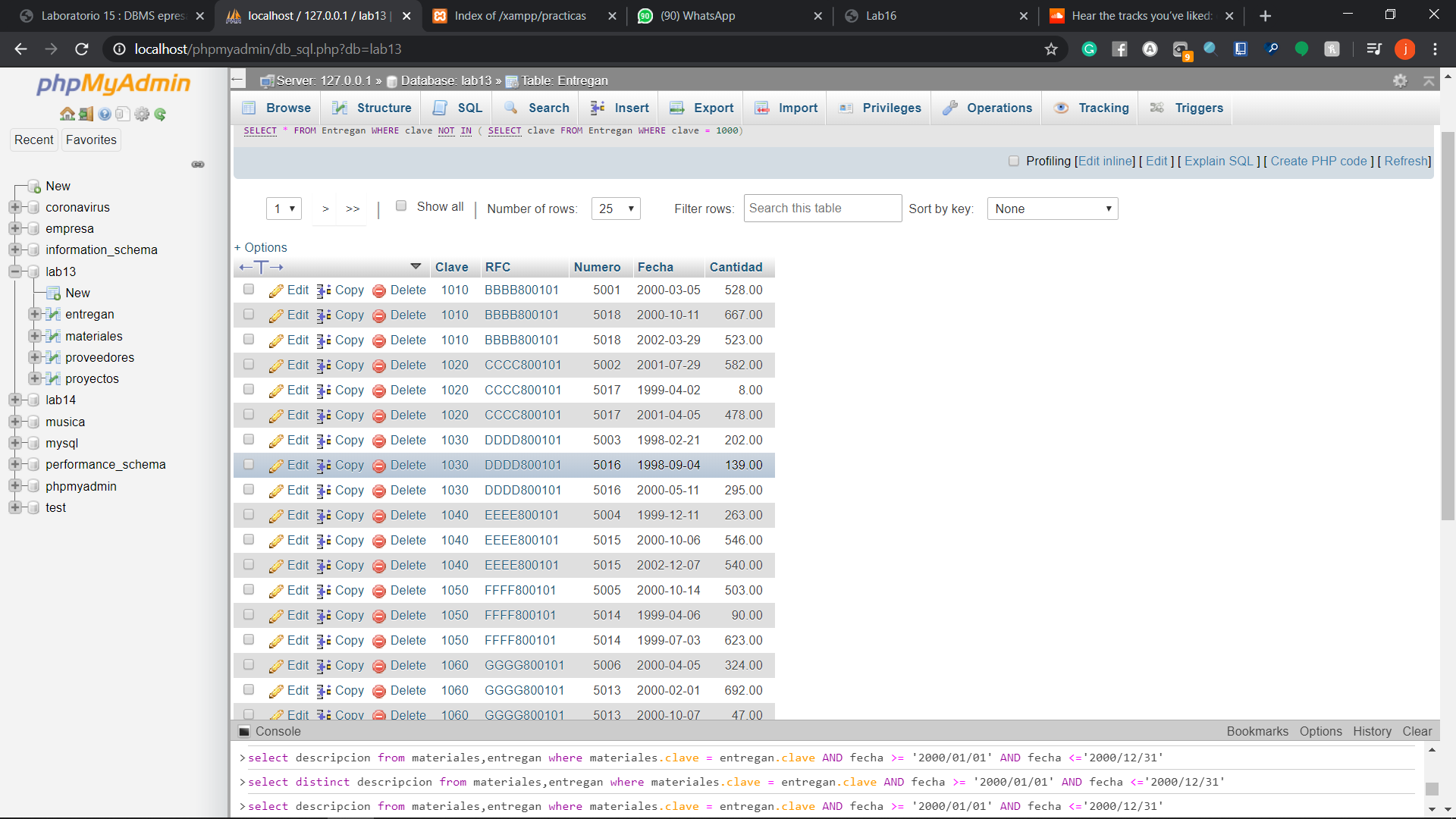
R=No, la consulta solo muestra los materiales ya entregados

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

R=select \* from entregan where clave=1450 OR clave=1300

**Diferencia (se ilustra con selección )**  
  
Algebra relacional.  
entregan - SL{clave=1000}(entregan)  
  
SQL  
(select \* from entregan)  
minus  
(select \* from entregan where clave=1000)  
  
Nuevamente, "minus" es una palabra reservada que no está definida en SQL Server, define una consulta que regrese el mismo resultado.

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM Entregan WHERE clave [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ( [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) clave FROM Entregan WHERE clave = 1000)



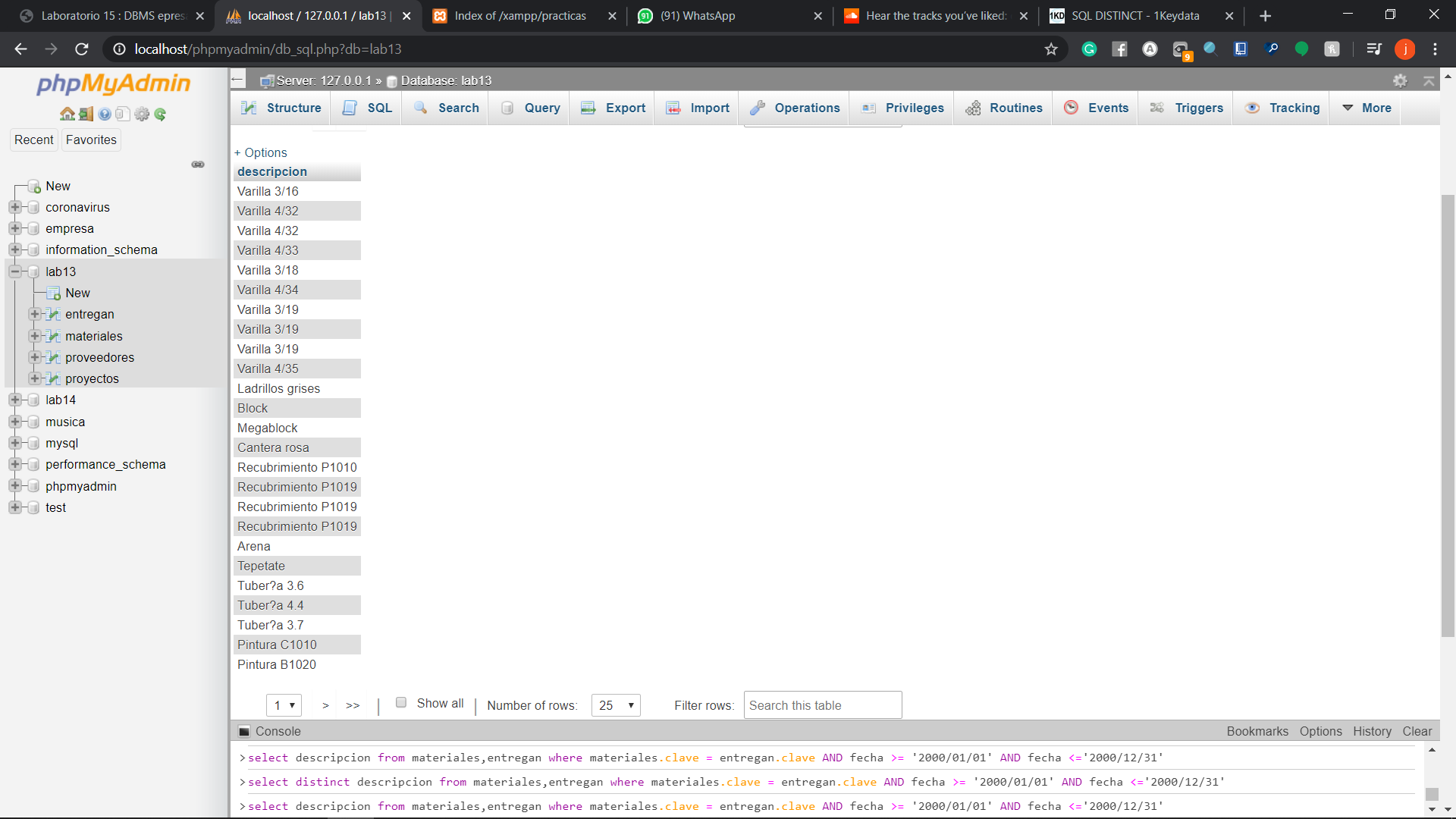
**Producto cartesiano**  
  
Algebra relacional.  
entregan X materiales  
  
SQL  
select \* from entregan,materiales  
  
¿Cómo está definido el número de tuplas de este resultado en términos del número de tuplas de entregan y de materiales?

Estas tuplas es la suma de las tuplas de cada tabla

**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.  
  
¿Por qué aparecen varias veces algunas descripciones de material?

select descripcion from materiales,entregan where materiales.clave = entregan.clave AND fecha >= '2000/01/01' AND fecha <='2000/12/31'

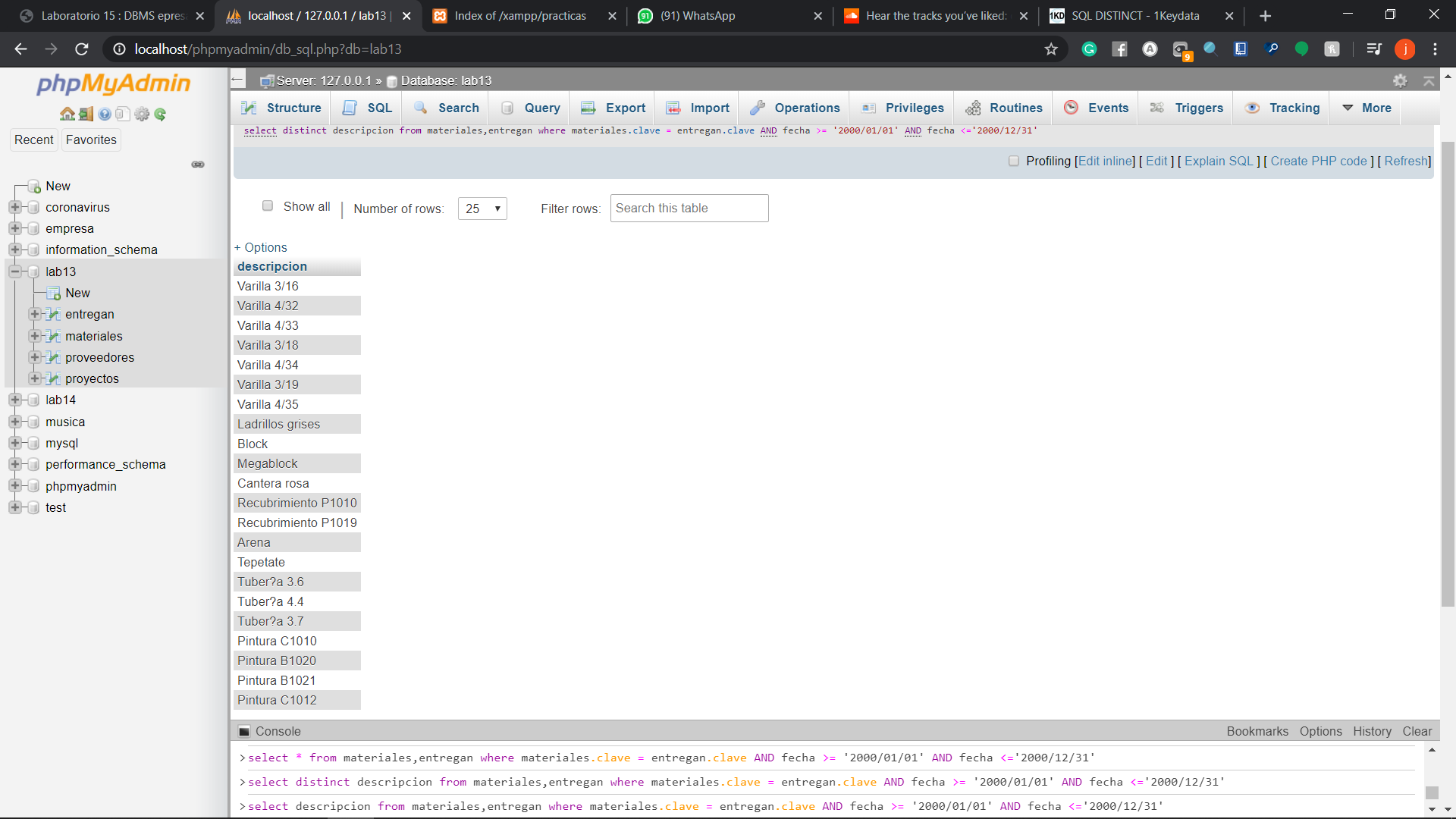
Porque fueron entregados varias veces



**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.  
  
¿Qué resultado obtienes en esta ocasión?

select distinct descripcion from materiales,entregan where materiales.clave = entregan.clave AND fecha >= '2000/01/01' AND fecha <='2000/12/31'

ahora no se repite ningúna descripción

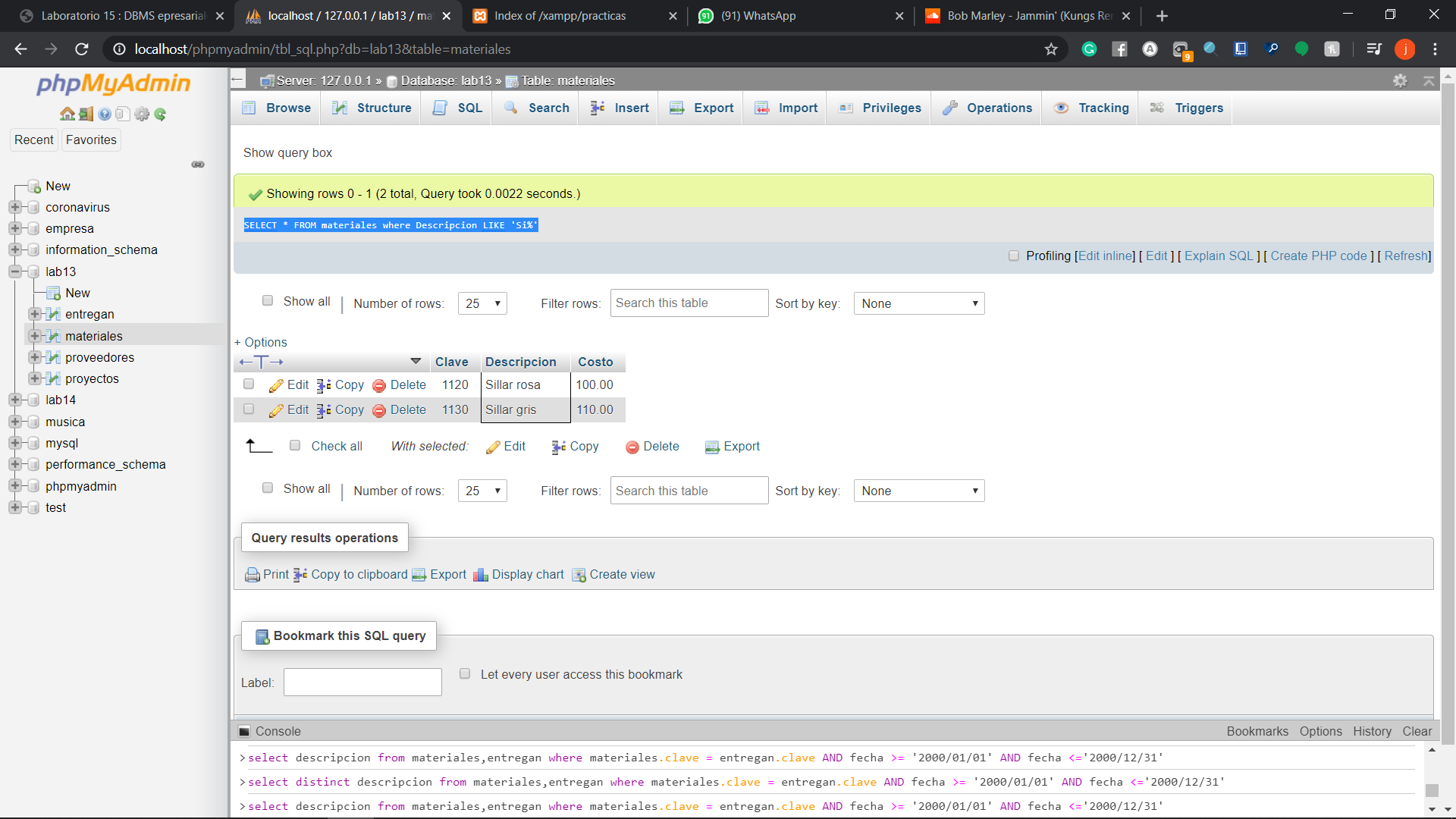


**Ordenamientos.**  
  
Si al final de una sentencia select se agrega la cláusula  
  
order by campo [desc] [,campo [desc] ...]  
  
donde las partes encerradas entre corchetes son opcionales (los corchetes no forman parte de la sintaxis), los puntos suspensivos indican que pueden incluirse varios campos y la palabra desc se refiere a descendente. Esta cláusula permite presentar los resultados en un orden específico.  
  
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.

Faltante

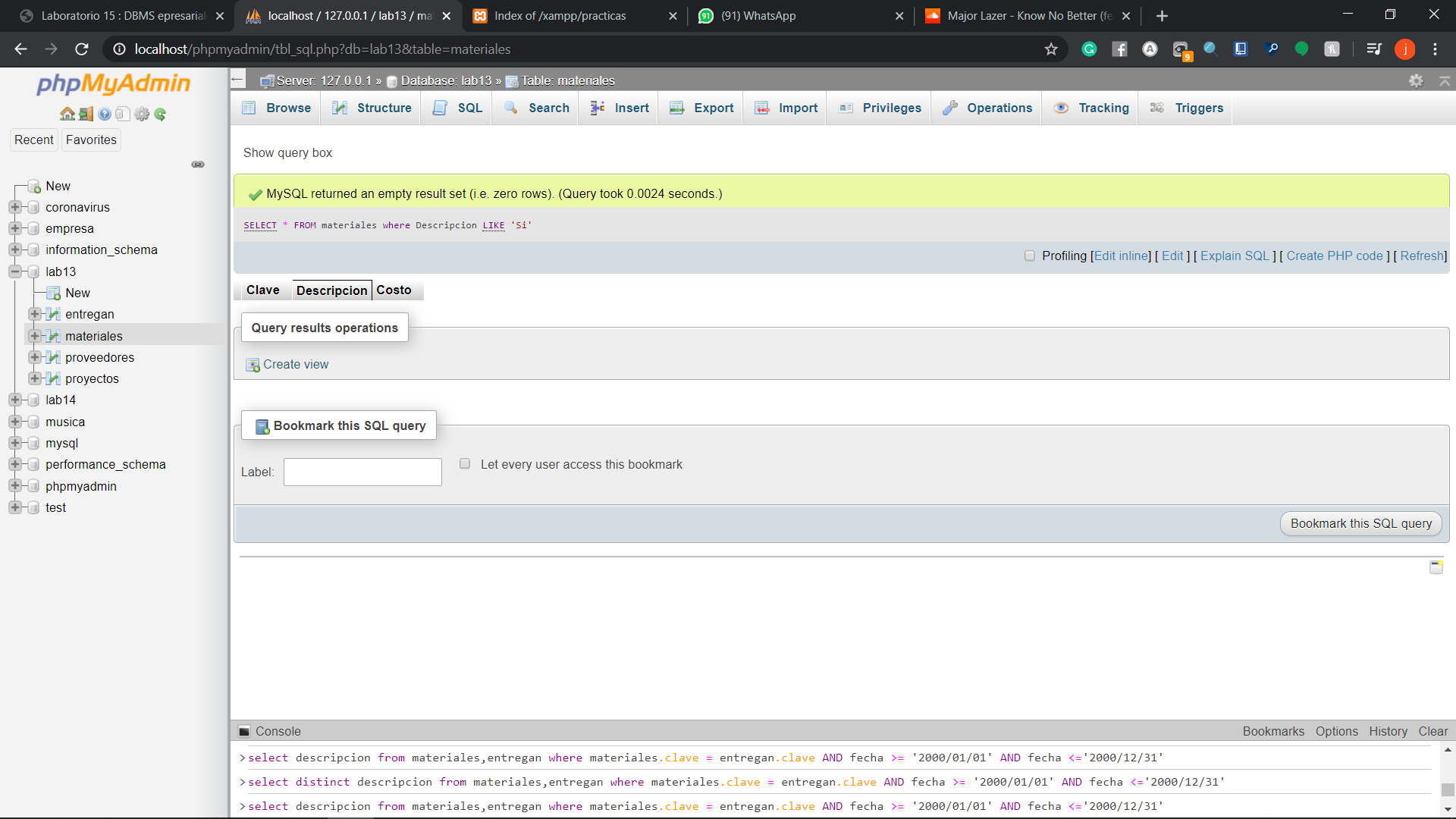
**Operadores de cadena**  
  
El operador LIKE se aplica a datos de tipo cadena y se usa para buscar registros, es capaz de hallar coincidencias dentro de una cadena bajo un patrón dado.  
  
También contamos con el operador comodín (%), que coincide con cualquier cadena que tenga cero o más caracteres. Este puede usarse tanto de prefijo como sufijo.  
  
  
[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM materiales where Descripcion [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-comparison-functions.html#operator_like) 'Si%'

¿Qué resultado obtienes?  
Explica que hace el símbolo '%'. Esto significa que esperas mas letras por delante.



¿Qué sucede si la consulta fuera : LIKE 'Si' ?  
¿Qué resultado obtienes?  
Explica a qué se debe este comportamiento.

Esto significa que estas buscando tal cual esa palabra, no esperas que lleve otras letras por delante



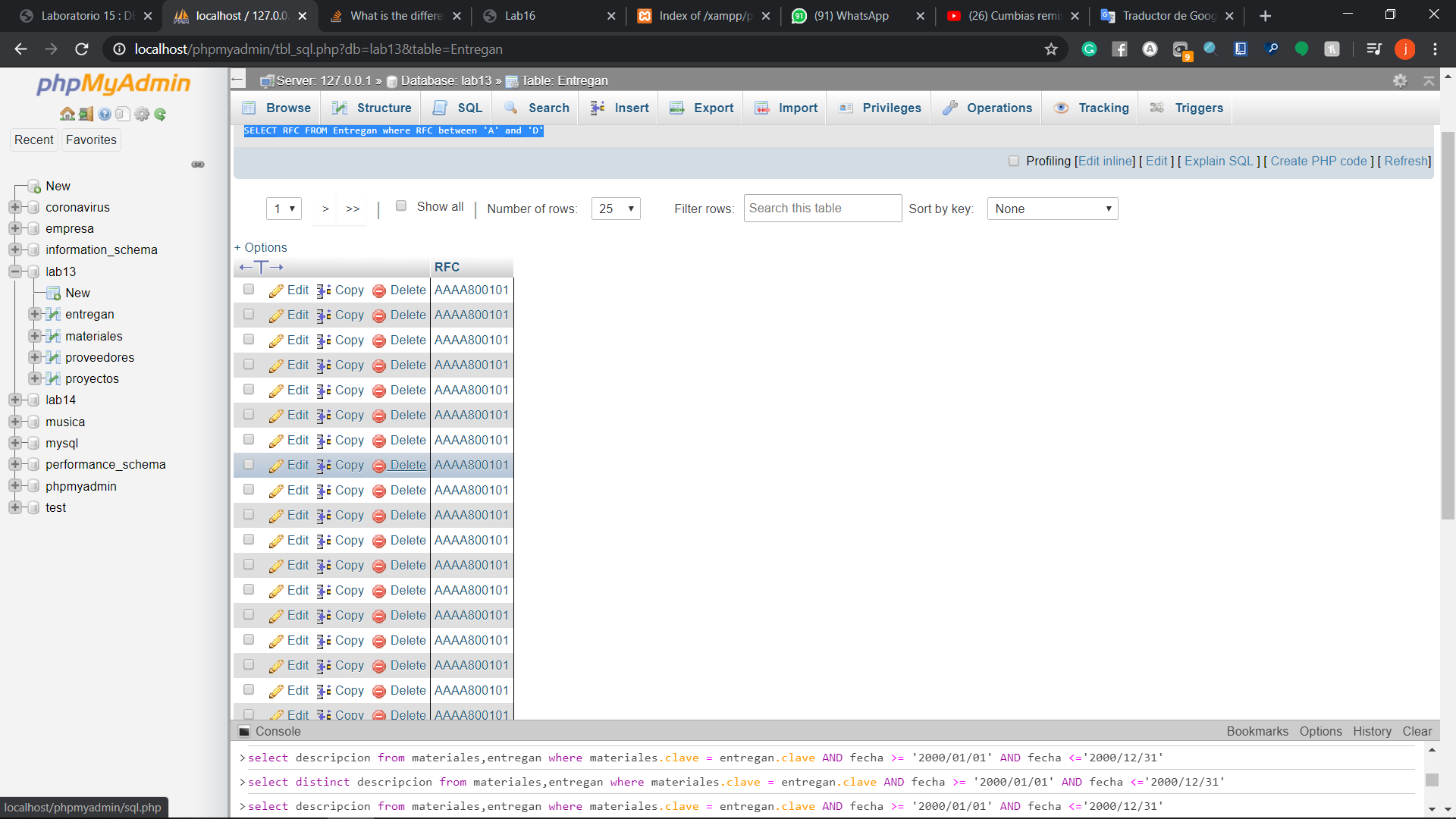
Otro operador de cadenas es el de concatenación, (+, +=) este operador concatena dos o más cadenas de caracteres.  
Su sintaxis es : Expresión + Expresión.  
Un ejemplo de su uso, puede ser: Un ejemplo de su uso, puede ser:  
SELECT (Apellido + ', ' + Nombre) as Nombre FROM Personas;  
  
DECLARE @foo 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?  
¿Cuál es la función de @foo?  
¿Que realiza el operador SET?**

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]%';

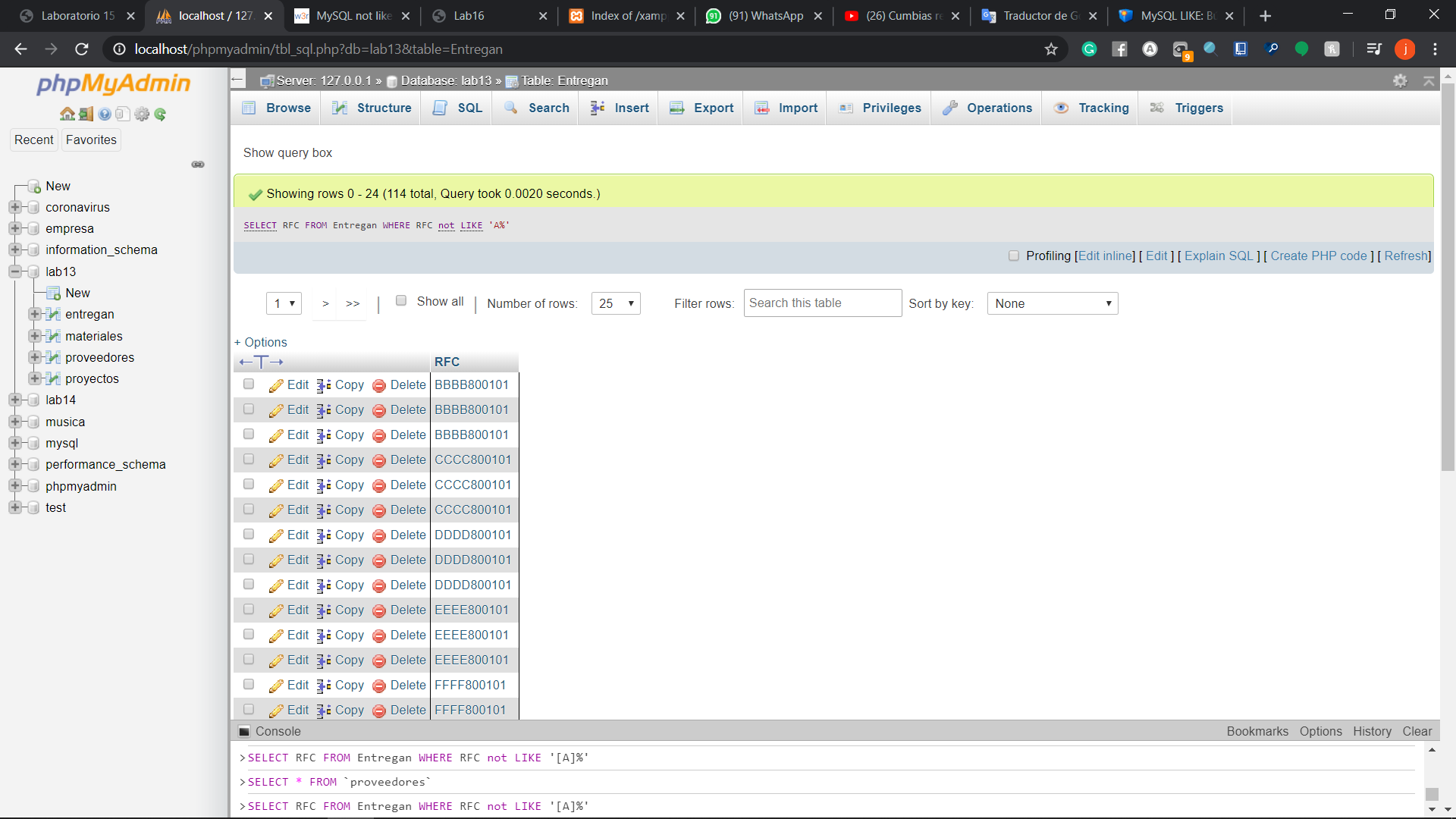
[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) RFC FROM Entregan where RFC between 'A' [and](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) 'D'

Busca el rfc que empiece entre una A y una D

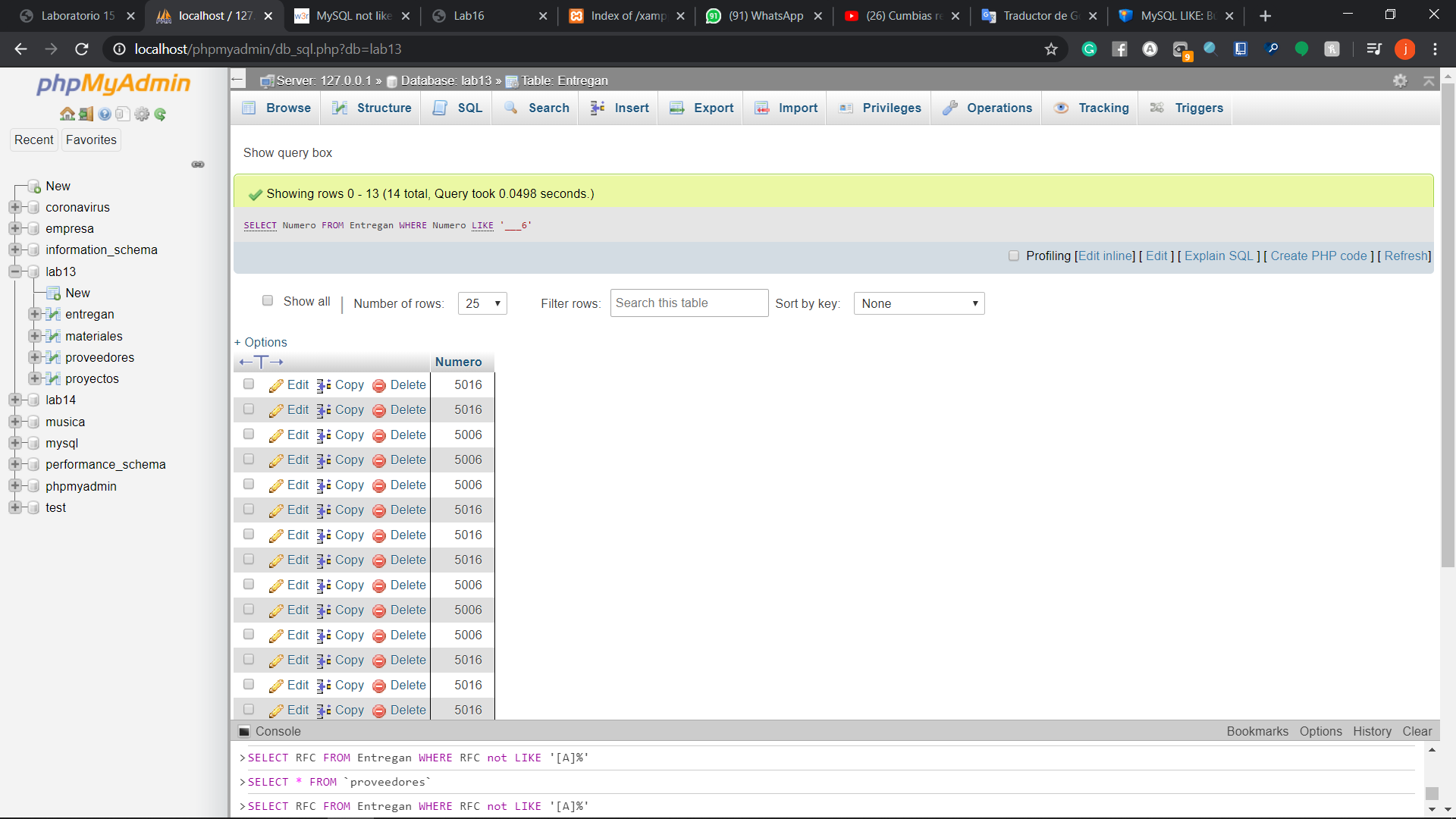
  
SELECT RFC FROM Entregan WHERE RFC LIKE '[^A]%';

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) RFC FROM Entregan WHERE RFC [not](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-comparison-functions.html#operator_like) 'A%'

Busca los rfc que no empiecen con una A

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

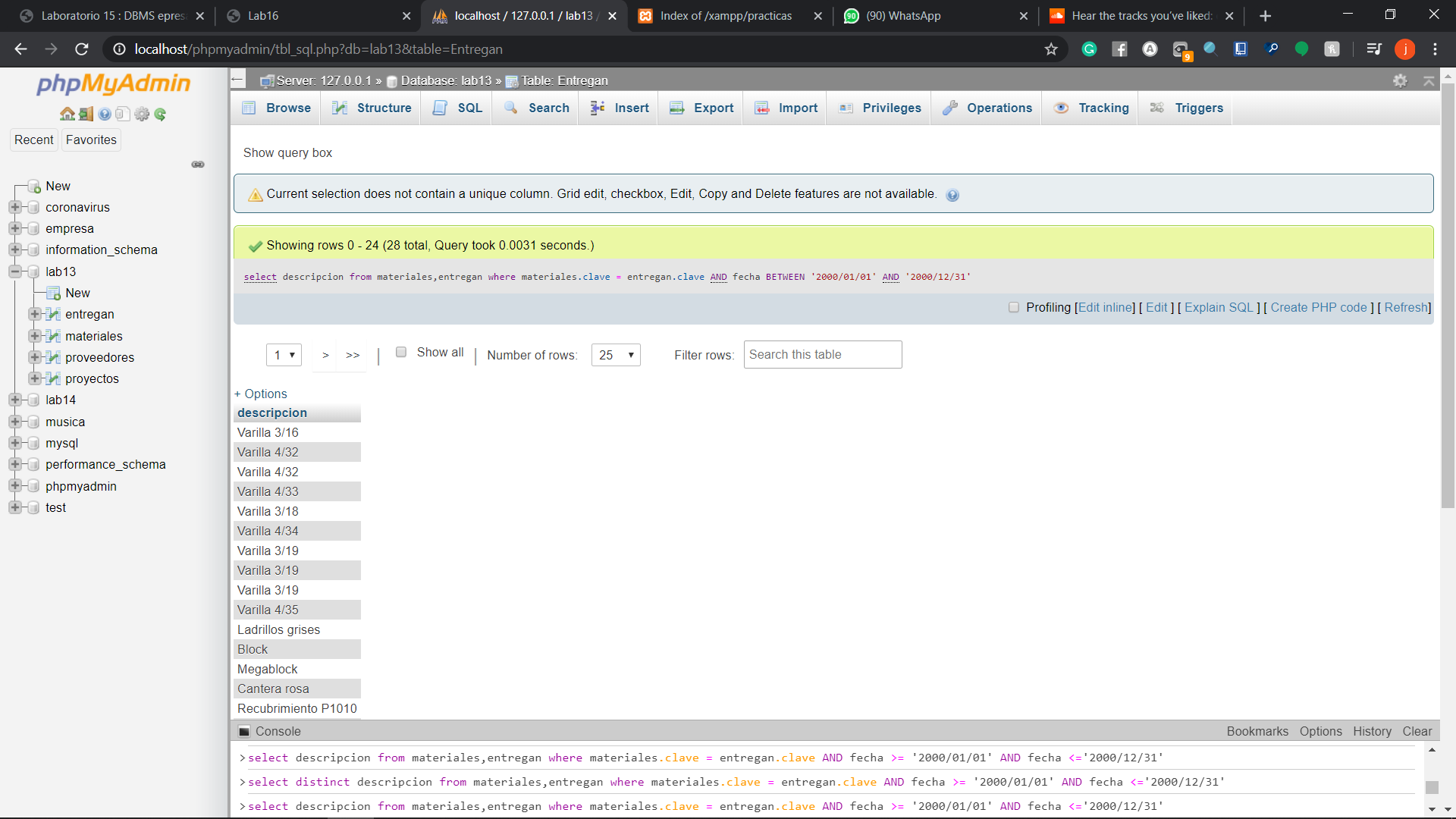
Después de 3 espacios que exista un 6 en el numero de la tabla entregan



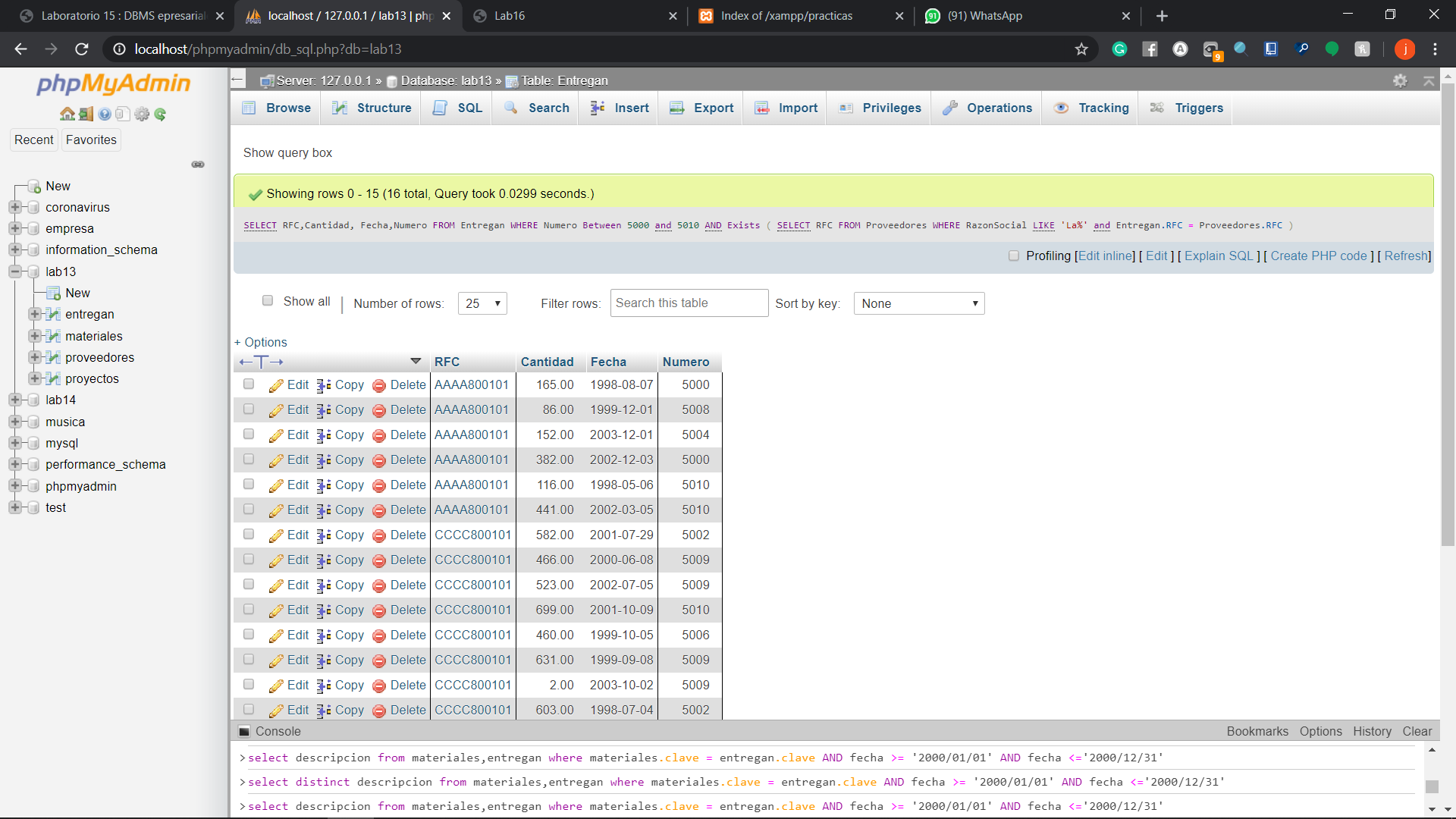
**Operadores Lógicos.**  
  
Los operadores lógicos comprueban la verdad de una condición, al igual que los operadores de comparación, devuelven un tipo de dato booleano (True, false o unknown).  
  
**ALL** Es un operador que compara un valor numérico con un conjunto de valores representados por un subquery. La condición es verdadera cuando todo el conjunto cumple la condición.  
  
**ANY o SOME** Es un operador que compara un valor numérico con un conjunto de valores. La condición es verdadera cuando al menos un dato del conjunto cumple la condición.  
  
La sintaxis para ambos es: valor\_numerico {operador de comparación} subquery  
  
**BETWEEN** Es un operador para especificar intervalos. Una aplicación muy común de dicho operador son intervalos de fechas.  
  
SELECT Clave,RFC,Numero,Fecha,Cantidad  
FROM Entregan  
WHERE Numero Between 5000 and 5010;  
  
¿Cómo filtrarías rangos de fechas?

Algo asi:

[select](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) descripcion from materiales,entregan where materiales.clave = entregan.clave [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) fecha BETWEEN '2000/01/01' [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) '2000/12/31'



**EXISTS** Se utiliza para especificar dentro de una subconsulta la existencia de ciertas filas.  
  
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?

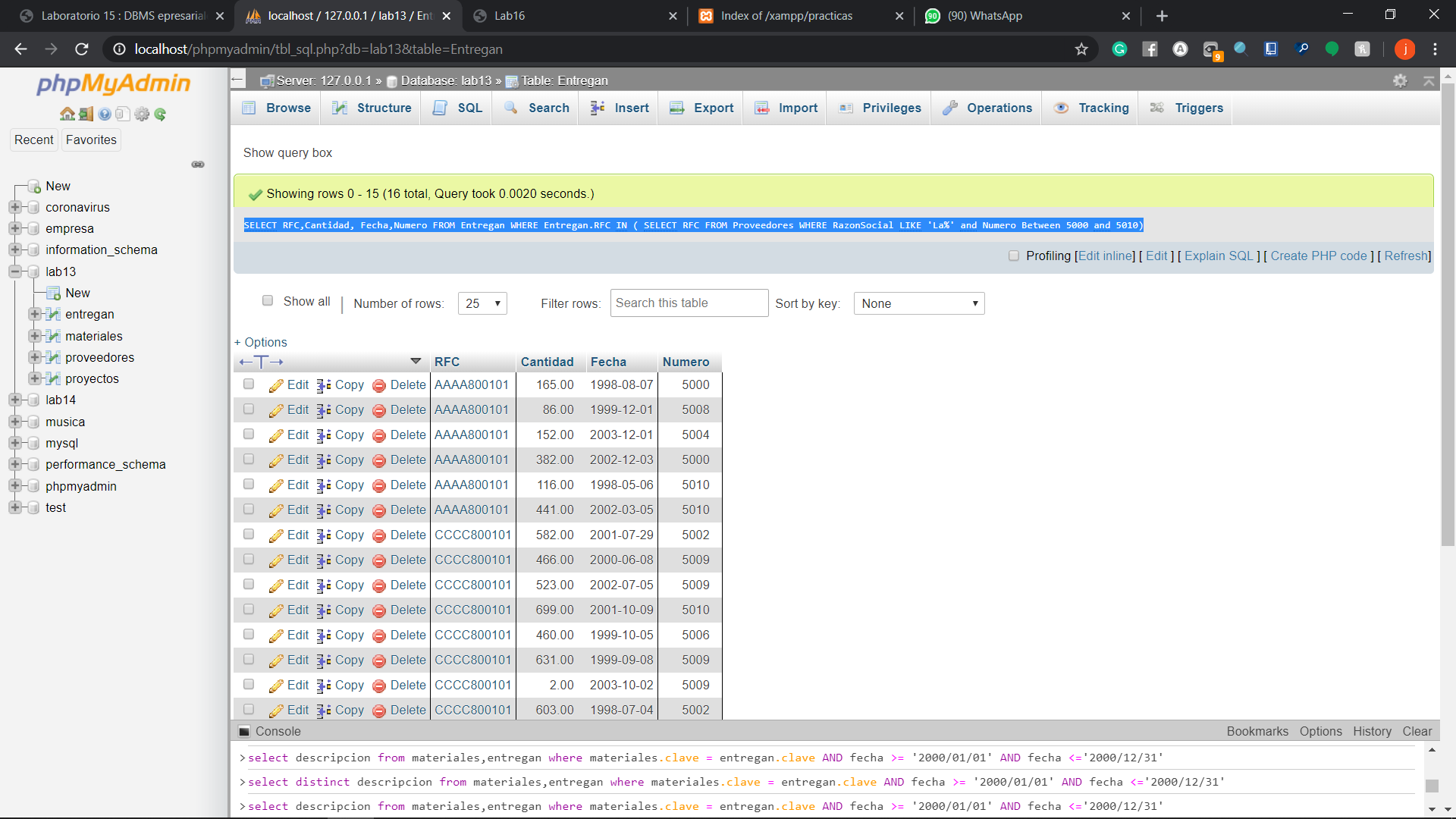


¿Qué función tiene el paréntesis ( ) después de EXISTS?

Indica lo que es la otra subconsulta

**Tomando de base la consulta anterior del EXISTS, realiza el query que devuelva el mismo resultado, pero usando el operador IN**

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) RFC,Cantidad, Fecha,Numero FROM Entregan WHERE Entregan.RFC [IN](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/comparison-operators.html#function_in) ( [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) RFC FROM Proveedores WHERE RazonSocial [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-comparison-functions.html#operator_like) 'La%' [and](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) Numero Between 5000 [and](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) 5010)



El Operador TOP, es un operador que recorre la entrada, un query, y sólo devuelve el primer número o porcentaje especifico de filas basado en un criterio de ordenación si es posible.  
  
¿Qué hace la siguiente sentencia? Explica por qué.  
  
SELECT TOP 2 \* FROM Proyectos