

# UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA – UNAD ESCUELA DE CIENCIAS BÁSICAS TECNOLOGÍA E INGENIERÍA PROGRAMA: INGENIERÍA DE SISTEMAS

#### **BASE DE DATOS AVANZADA**

## PARTICIPANTE LILIANA ANDREA RINCON

GRUPO 301125\_26

TUTOR
MARCO ANTONIO LOPEZ OSPINA

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA
OCTUBRE 2016



## UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA – UNAD ESCUELA DE CIENCIAS BÁSICAS TECNOLOGÍA E INGENIERÍA PROGRAMA: INGENIERÍA DE SISTEMAS

#### INTRODUCCION

El siguiente trabajo tiene como objetivo la creación de la base de datos donde cada participante expone su base de datos creada, ademas las respuestas a los interrogantes propuestos por el tutor. El cual se encuentra estructurado de la siguiente manera, respuestas seleccionadas, documento sobre procedimientos y funciones, en el gestor que el grupo está trabajando, instructivo de creación, actualización y consulta de bases de datos con el gestor que el grupo esté trabajando.



PROGRAMA: INGENIERÍA DE SISTEMAS

#### RESPUESTAS A LAS PREGUNTAS ORIENTADORAS

#### Que es un lenguaje 4g y de algunos ejemplos.

Son lenguajes de cuarta generación, es decir de alto nivel, permitiendo realizacion de programas a corto tiempo, ademas es un poco más facil de trabajar gracias a sus interfaces. Ya que son más orientados a objetos.

Algunos ejemplos son: SQL, ORACLE.

#### **4** Que es SQL y cuáles son sus principales componentes.

Lenguaje de consultas estructuradas, tomando forma de lenguaje de comandos, permitiendo realizar diferentes actividades com insertar, actualizar entre otros, junto con una interfaz de programación.

#### Principales componentes:

- Tipos de datos: Varían de acuerdo a los tipos de datos. "Numéricos, Alfanuméricos, Fecha, Lógico, Blob".
- Operadores. Estas se dividen en aritméticos, relacionados, lógicos, concatenación.
- Palabra clave
- Funciones agregadas
- Predicados

#### Que es una llave foránea y cual su utilidad.

Es una llave externa, consiste en relacionar tablas donde la llave maestra realiza unión con otro campo de otra tabla.

Determinar en el momento de las relaciones entre las tablas en la cual la Llave Primaria se realizar la unión con el otro campo de otra tabla y automáticamente se vuelve una llave foránea.

#### **4** Que puede pasar si no se define adecuadamente la integridad referencial.

Puede haber inconsistencias en los datos, errores graves como invalides de los registros de las tablas donde se borren o cambien datos relacionados de forma accidental.

#### **♣** Que es una llave candidata y de un ejemplo.

Son claves que pueden servir como clave principal



PROGRAMA: INGENIERÍA DE SISTEMAS

Por ejemplo Tabla xy cod\_xy, nit\_xy nombre\_xy aquí el campo clave es el cod\_xy pero si no se tuviera o estuviera repetido tendríamos como candidato al campo nit\_xy que al igual que el código generaría unicidad en la tabla empresa.

**Cuáles son las operaciones que se hacen con DDL** "Lenguaje de Definición de Datos, Operaciones básicas."

Créate: crea bases de datos, vistas, usuarios, tablas

Alter: modifica tablas

Drop: borrar bases de datos, vistas, usuarios, tablas.

Trúncate: Borra los registros de una tabla.

**Que operaciones se realizan con DML.**" Lenguaje de manipulación de datos, operaciones no básicas"

Select: Permite a los usuarios de la base de datos retirar información específica que ellos desean de una base de datos operacional.

Insert: Agregar nuevos registros a una tabla existente.

Update: actualiza datos de las tablas que conforman la base datos

Delete: borrar registros específicos de una tabla.

**↓** Cuando se utiliza el DCL. controlar el acceso a datos denegando y otorgando privilegios sobre los objetos existentes.

OMMIT : Guarda los trabajos realizados en las transacciones

ROLLBACK : Restaura la base de datos al estado original desde el comando COMMIT pasado en las transacciones

SAVEPOINT : establecer un punto en que es posible un ROLLBACK.

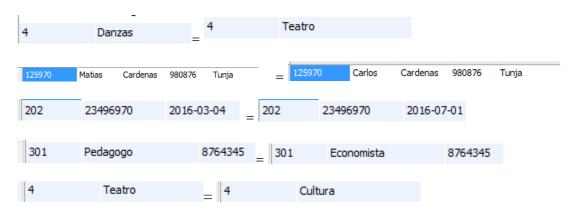
SAVE TRANSACTION: Establece un punto de almacenamiento dentro de una transacción

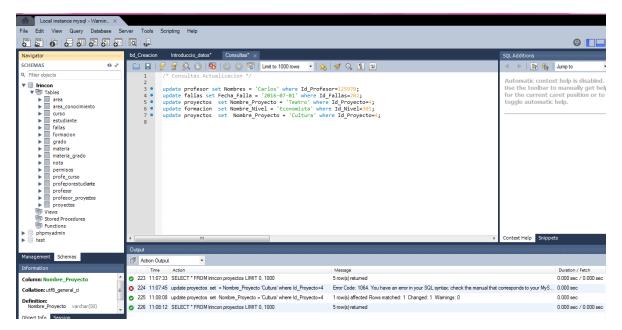


PROGRAMA: INGENIERÍA DE SISTEMAS

#### **SENTENCIAS**

**Sentencia5 Actualización de Datos:** UPDATE nombre\_de\_table SETnombre\_columna = nuevo\_valor, -WHERE....





### 4. Consultar la base de datos y extraer la información solicitada en las consultas.

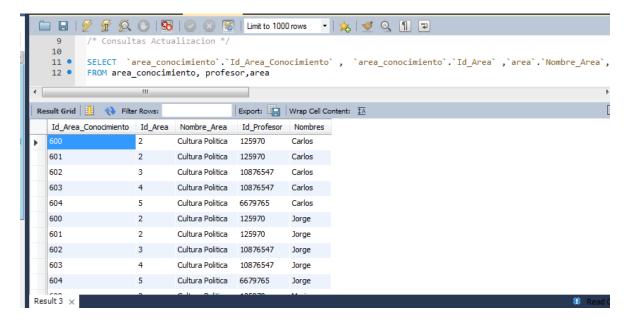
#### \* Sentencias6 de Consultas:

- Consulta que muestra Profesores por áreas de conocimiento

SELECT `area\_conocimiento`.`Id\_Area\_Conocimiento` `area\_conocimiento`.`Id\_Area`, `area`.`Nombre\_Area`, `area\_conocimiento`.`Id\_Profesor`, `profesor`.`Nombres` ROM area\_conocimiento, profesor,área



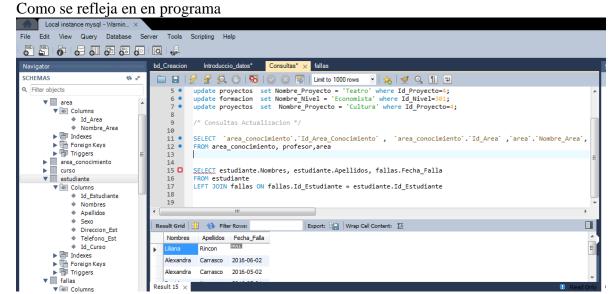
#### PROGRAMA: INGENIERÍA DE SISTEMAS



- Consulta donde se mira las fallas de los alumnos y fechas donde aparece que hay alumnos que no tuvieron fallas por esta razon aparece Null

SELECT estudiante.Nombres, estudiante.Apellidos, fallas.Fecha\_Falla FROM estudiante

LEFT JOIN fallas ON fallas.Id\_Estudiante = estudiante.Id\_Estudiante



Como se refleja en el localhost



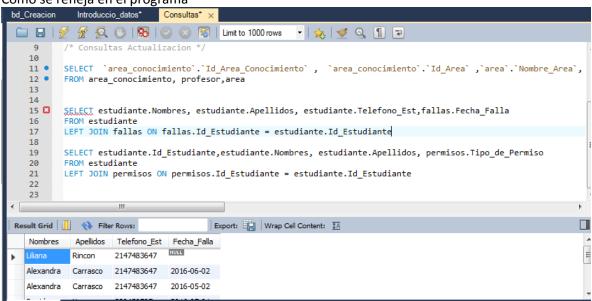
PROGRAMA: INGENIERÍA DE SISTEMAS

← <del>T</del> →				
Nombres	Apellidos	Fecha_Falla		
Liliana	Rincon	NULL		
Alexandra	Carrasco	co 2016-06-02		
Alexandra	Carrasco	2016-05-02		
David	Vargas	2016-07-01		
David	Vargas	2016-06-09		
Alexander	Arias	NULL		
Juan	Martinez	2016-11-01		

- Reporte de los permisos que ha tenido los alumno, al igual si aparece Null es porque no han tenido ningun reporte de permiso alguno

SELECT estudiante.Id\_Estudiante,estudiante.Nombres, estudiante.Apellidos, permisos.Tipo\_de\_Permiso
FROM estudiante
LEFT JOIN permisos ON permisos.Id Estudiante = estudiante.Id Estudiante

Como se refleja en el programa



Como se refleja en el programa

← <del>T</del> →				
Id_Estudiante	Nombres	Apellidos	Tipo_de_Permiso	
1033764222	Liliana	Rincon	Examen medicos	
2147483647	Alexandra	Carrasco	NULL	
23496970	David	Vargas	Familiar	
849384030	Alexander	Arias	NULL	
7834950	Juan	Martinez	Acontecimiento	
7834950	Juan	Martinez	Celebridad	
7834950	Juan	Martinez	Terapias Medicas	

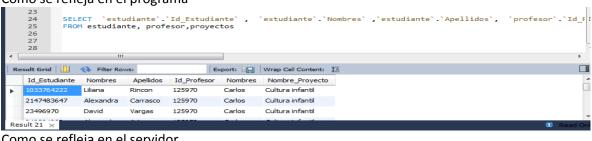


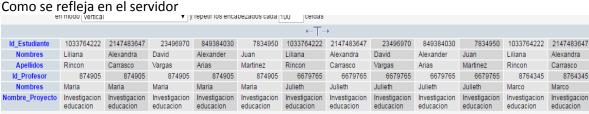
#### PROGRAMA: INGENIERÍA DE SISTEMAS

Consulta que relaciona los proyectos que manejas los alumnos y profesores.

SELECT `estudiante`.`Id\_Estudiante`, `estudiante`.`Nombres`, `estudiante`.`Apellidos`, `profesor`.`Id\_Profesor`, `profesor`.`Nombres`, `proyectos`.`Nombre\_Proyecto` FROM estudiante, profesor, proyectos

#### Como se refleja en el programa



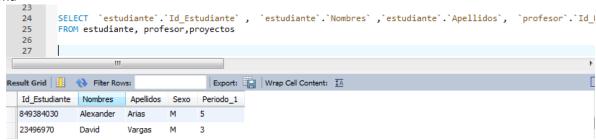


- Consulta de los resultados del primer periodo para los hombres

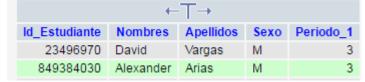
SELECT `estudiante`.`Id\_Estudiante`, `estudiante`.`Nombres`,`estudiante`.`Apellidos`, `estudiante`.`Sexo`, `nota`.`Periodo 1`

FROM estudiante, nota where ((sexo= 'M'))

Programa



Servidor



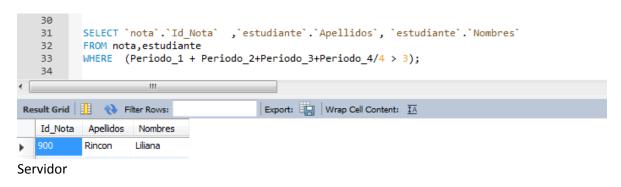
- Consulta q muestra las personas q pasaron

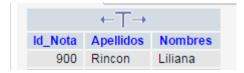
SELECT `nota`.`Id\_Nota`, `estudiante`.`Apellidos`, `estudiante`.`Nombres` FROM nota.estudiante

WHERE (Periodo\_1 + Periodo\_2+Periodo\_3+Periodo\_4/4 > 3);



#### PROGRAMA: INGENIERÍA DE SISTEMAS







PROGRAMA: INGENIERÍA DE SISTEMAS

### INSTRUCTIVO DE CREACIÓN, ACTUALIZACIÓN Y CONSULTA DE BASES DE DATOS CON EL GESTOR

**CREACION BASE DE DATOS:** Al haber creado el modelo lógico y seleccionado la herramienta o software en donde se va a crear la base de datos se procede a implementar la base de datos para ello se tiene diferentes sentencias que se utilizaran de igual manera para diferentes cosas.

Se especifica la sentencia **créate** seguido de **database** para especificar qué es lo que queremos crear seguido de **nombre** que se le dará a la base de datos así:

create database nombre\_bd;

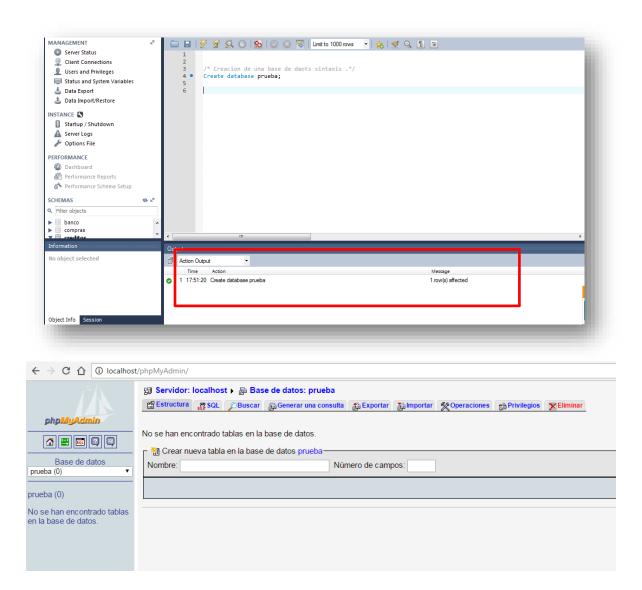
Ejemplo:

Create database prueba;

Al darle clic en refrescar en el icono de un rayo, parecerá en la parte inferior base de datos creada y ademas a ir al local host aparecerá la base de datos creada.



PROGRAMA: INGENIERÍA DE SISTEMAS



♣ <u>CREACION DE TABLAS:</u> Para crear las tablas se utilizara la misma sentencia create pero esta vez seguido de la palabra table y el nombre de la tabla a crear.

Para esto se puede de dos formas si solo estamos trabajando con una base de datos la sintaxis seria:

#### create table <a href="mailto:nombre\_tabla">nombre\_tabla</a> (campos que la conforman)

Para trabajar más de una tabla es preferible especificar en qué base de datos queremos crear la tabla asi como se indicar a continuación.

create table nombre\_bd.nombre\_tabla. (Campos que la conforman).

Ejemplo

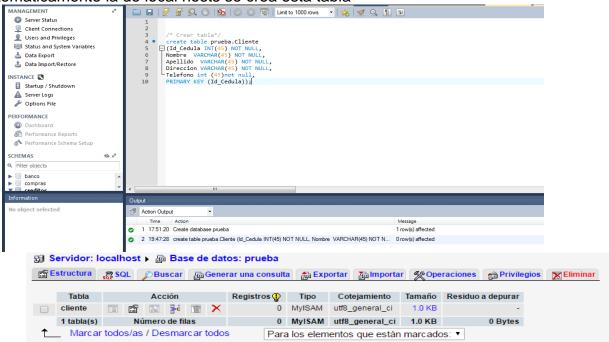


PROGRAMA: INGENIERÍA DE SISTEMAS

Crear la tabla cliente para la base de datos prueba

/\* Crear tabla\*/
create table prueba.Cliente
(Id\_Cedula INT(45) NOT NULL,
Nombre VARCHAR(45) NOT NULL,
Apellido VARCHAR(45) NOT NULL,
Direccion VARCHAR(45) NOT NULL,
Telefono int (45)not null,
PRIMARY KEY (Id\_Cedula));

Al darle refrescar el programa al final aparecer la creación exitosa yal igual automaticamente la de local hosts se crea esta tabla



En la parte izquierda aparece la base de datos ya creada y a medida q creemos las tablas aparecerá esta mismas.

**INSERTACION DE DATOS EN LAS TABLAS:** Se procede a insertar datos.

Sentencia:

Insert into nombre tabla (campo1, campo2) values (valor campo1, valor campo2);

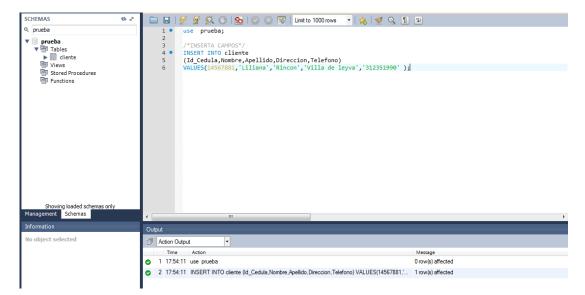
Ejemplo

/\*INSERTA CAMPOS\*/
INSERT INTO cliente



PROGRAMA: INGENIERÍA DE SISTEMAS

(Id\_Cedula,Nombre,Apellido,Direccion,Telefono) VALUES(14567881,'Liliana','Rincon','Villa de leyva','312351990'); En la parte inferior aparecerán datos ingresados con éxito.





PROGRAMA: INGENIERÍA DE SISTEMAS

Al igual en el local host donde está la base de datos se verá reflejado que esta los datos creados.



#### **LIMINAR DATOS**

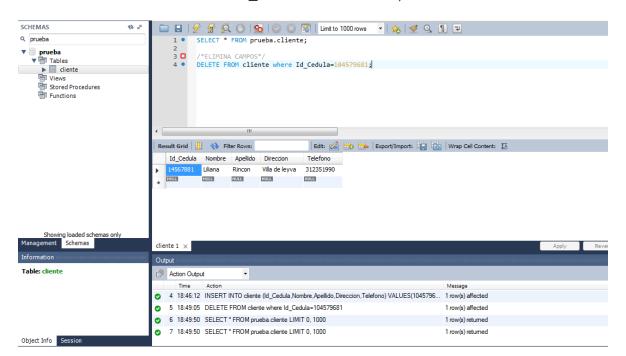
Si se desea eliminar una fila de la tabla que se esté trabajando se utiliza la sentencia **delete** cuya sintaxis es

#### delete from nombre\_tabla where campo\_clave;

Ejemplo

Indica que borre de la tabla cliente cuando id de la cedula sea 104579681 /\*ELIMINA CAMPOS\*/

DELETE FROM cliente where Id\_Cedula=104579681;



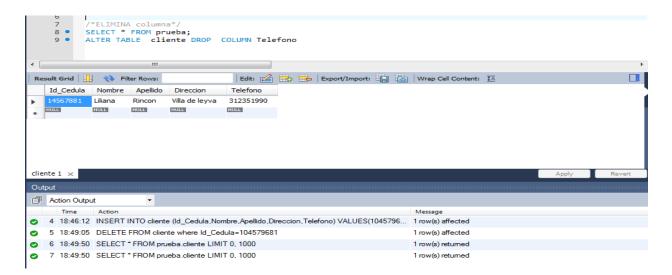


#### PROGRAMA: INGENIERÍA DE SISTEMAS

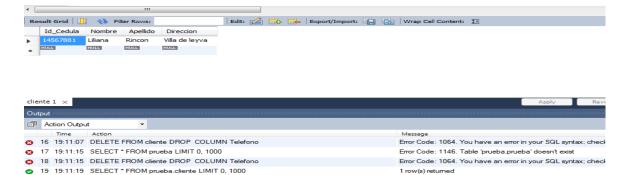


#### **LIMINAR COLUMNA**

/\*ELIMINA columna\*/
SELECT \* FROM prueba;
DELETE FROM cliente DROP COLUMN Telefono



#### Después aparece l tabla ya modificada





PROGRAMA: INGENIERÍA DE SISTEMAS

#### **MODIFICACION DE DATOS**

La sintaxis de esta orden es:

Mysql> UPDATE nombre\_de\_table SET

- -> nombre\_columna = nuevo\_valor,
- -> WHERE....

#### Ejemplo

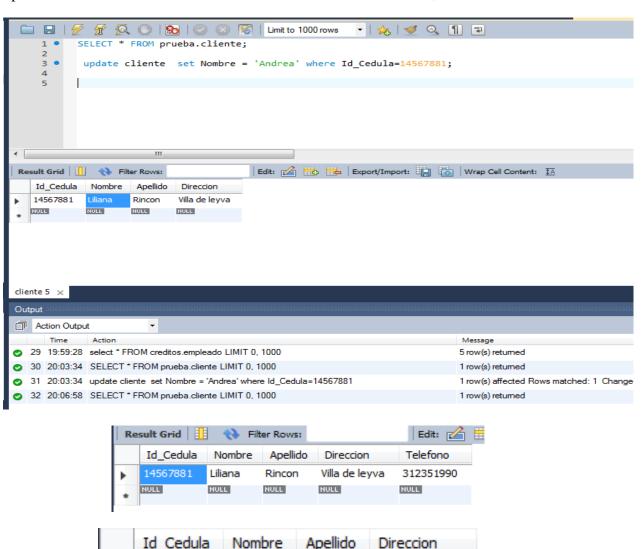
update cliente set Nombre = 'Andrea' where Id\_Cedula=14567881;

14567881

NULL

Andrea

NULL



Rincon

NULL

Villa de leyva

NULL



PROGRAMA: INGENIERÍA DE SISTEMAS

#### **4** CONSULTAS

Después de haber creado, insertado información a las tablas, se procede a realizar las respectivas consultas. Estas se pueden consultar de una tabla o de dos tablas, el **select, from, where, inner join, between, like, limit, order by, group by** aunque existen muchas con el mismo propósito.

#### Select

Permite seleccionar las columnas que se van a mostrar y en el orden en que lo van a hacer. Simplemente es la instrucción que la base de datos interpreta como que vamos a solicitar información.

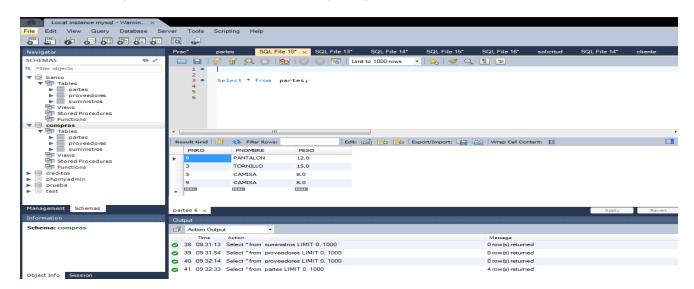
#### Select y from

From: Esta cláusula permite indicar las tablas o vistas de las cuales vamos a obtener la información. De momento veremos ejemplos para obtener información de una sola tabla.

"Sintaxis" **Select** \* **from** nombre\_tabla; Ejemplo

Select \* from partes;

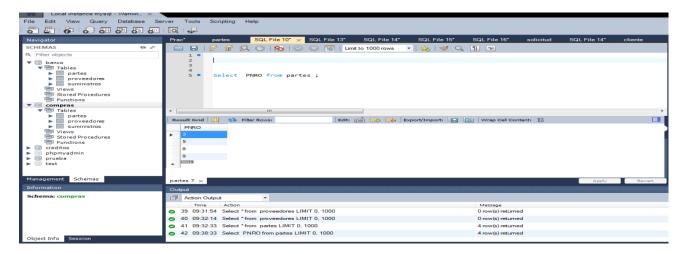
Esta línea indica que seleccione toda la informacio que hay en la tabla cliente:



Select cedula from Cliente:



#### PROGRAMA: INGENIERÍA DE SISTEMAS



#### o ALL / DISTINCT

ALL es el valor predeterminado, especifica que el conjunto de resultados puede incluir filas duplicadas. Por regla general nunca se utiliza.

DISTINCT especifica que el conjunto de resultados sólo puede incluir filas únicas. Es decir, si al realizar una consulta hay registros exactamente iguales que aparecen más de una vez, éstos se eliminan. Muy útil en muchas ocasiones.<sup>1</sup>

"Sintaxis " SELECT DISTINCT Campos where condición FROM Tabla;

Ejemplo

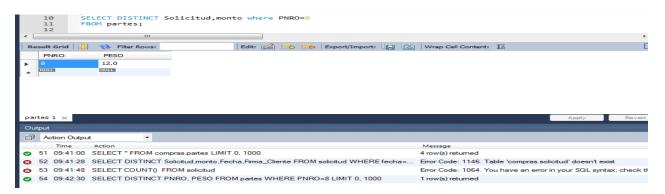
SELECT DISTINCT Solicitud,monto where PNRO=8 FROM partes;

Esta sentencia me dice que extraer la información de solicitud monto para el codigo 8 de la tabla partes. En si es traer información específica de una tabla.

<sup>&</sup>lt;sup>1</sup> http://www.campusmvp.es/recursos/post/Fundamentos-de-SQL-Como-realizar-consultas-simples-con-SELECT.aspx



#### PROGRAMA: INGENIERÍA DE SISTEMAS

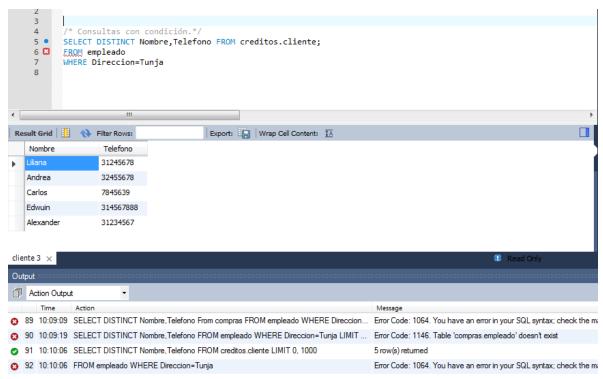




PROGRAMA: INGENIERÍA DE SISTEMAS

#### Consultas con condición

/\* Consultas con condición.\*/
SELECT DISTINCT Nombre, Telefono FROM creditos. cliente;
FROM empleado
WHERE Direccion=Tunja



### Sentencia order by

Nos indica la manera que queremos ver ordenada la información.

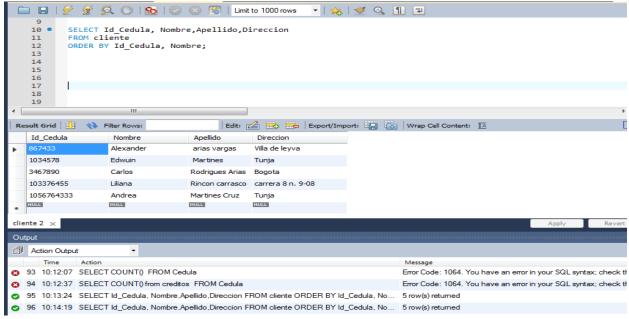
"Sintaxis" Select campo from tabla order by campo;

Ejemplo

SELECT Id\_Cedula, Nombre, Apellido, Direccion FROM cliente
ORDER BY Id\_Cedula, Nombre;

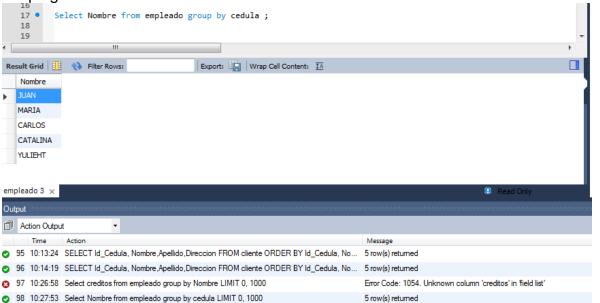


PROGRAMA: INGENIERÍA DE SISTEMAS



#### Sentencia group by

Es sentencia permite agrupar los datos de una tabla como en un menú desplegable

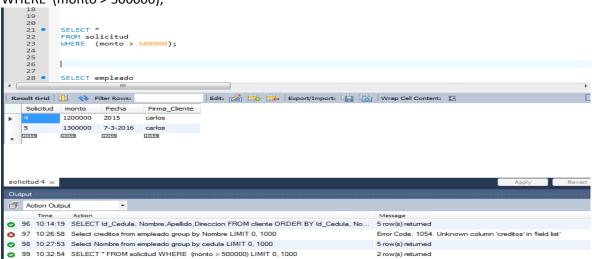




## UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA – UNAD ESCUELA DE CIENCIAS BÁSICAS TECNOLOGÍA E INGENIERÍA PROGRAMA: INGENIERÍA DE SISTEMAS

**Consultas con condiciones** 

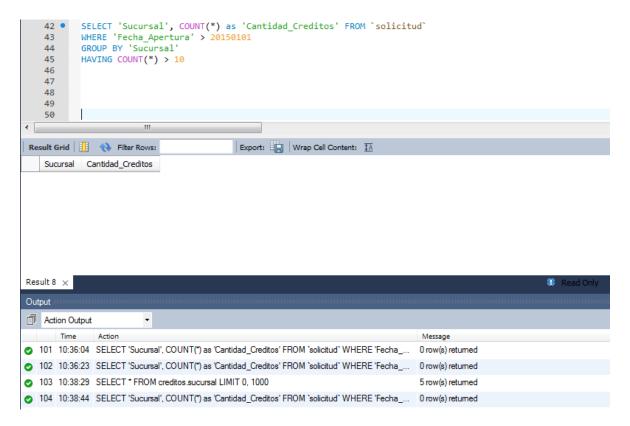
SELECT \*
FROM solicitud
WHERE (monto > 500000);



CONSULTA CON VARIAS SENTENCIAS



#### PROGRAMA: INGENIERÍA DE SISTEMAS

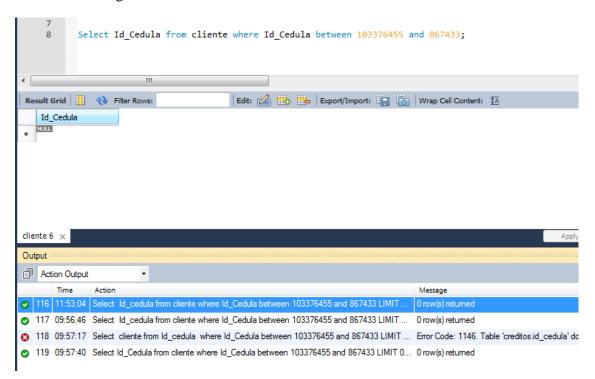




PROGRAMA: INGENIERÍA DE SISTEMAS

#### Sentencia between

#### Consulta de rango



#### Sentencia like

Este operador 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. Sintaxis

- 1° Select campo from tabla where campo like '%cadena caracteres%';
- 2° Select campo from tabla where campo like j\_\_\_\_;

#### **Ejemplo**

select \*
from cliente
where Apellido like 'R%'



#### PROGRAMA: INGENIERÍA DE SISTEMAS



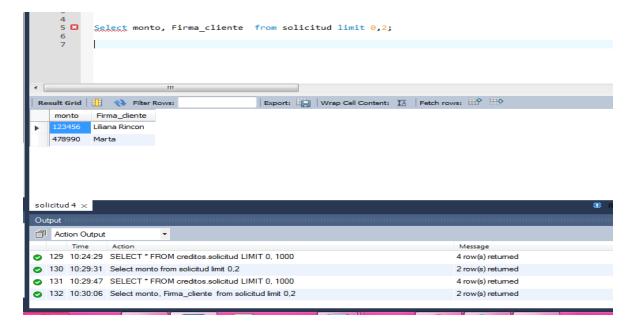
#### Sentencia limit

La cláusula "limit" se usa para restringir los registros que se retornan en una conLa cláusula "limit" se usa para restringir los registros que se retornan en una consulta "select".sulta "select".

Select campo from tabla limit 0,2;

#### Ejemplo

Select monto, Firma\_cliente from solicitud limit 0,2;





## UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA – UNAD ESCUELA DE CIENCIAS BÁSICAS TECNOLOGÍA E INGENIERÍA PROGRAMA: INGENIERÍA DE SISTEMAS

#### **CONCLUSIONES**

El siguiente trabajo fue de gran importancia ya que nos sirve para colocar en práctica los conocimientos adquiridos ademas ver las diferentes sentencias, ademas se logró evidenciar la importancia de realizar un diseño antes ya que al aplicarlo al programa es un poco menos extenso, por otra parte logra resolver dudas a traves de las realimentaciones realizadas por mis compañeros. Ademas se trabajó con compañeros de otros grupos haciendo mas dinamica la actividad y productiva. Se tuvo en cuenta las recomendación de mi compañero Rigoberto Coy "La importancia de las llaves foráneas es fundamental ya que están demarcan la relación de cada tabla, esto facilita la extracción y manipulación de datos, pero debemos ser muy cuidadosos en este manejo ya que si por alguna razón una llave foránea se borra o se actualiza se dañaran una serie de registros y toca mencionar que no es posible recuperar los de manera automática, cada cambio realizado queda de forma permanente"



PROGRAMA: INGENIERÍA DE SISTEMAS

#### REFERENCIAS BIBLIOGRAFICAS

- Sistemas de gestión de bases de datos (s.f.). Recuperado de: http://www.mcgraw-hill.es/bcv/guide/capitulo/8448148797.pdf
- ORACLE(2014). MySQL 5.0. Manual de referencia. Capítulo 2. Instalar MySQL. Recuperado de: <a href="http://datateca.unad.edu.co/contenidos/301125/2015-24/Instalacion\_de\_MySql.pdf">http://datateca.unad.edu.co/contenidos/301125/2015-24/Instalacion\_de\_MySql.pdf</a>
- http://searchdatacenter.techtarget.com/es/definicion/SQL-o-lenguaje-de-consultasestructuradas
- http://www.casdreams.com/auladeinformatica/cet/mansql2.htm
- https://rahsuarez.wordpress.com/2010/01/05/ddl-dml-dcl/