



**IPN, ESCOM, ING. EN SISTEMAS COMPUTACIONALES**

**MATERIA: DISTRIBUTED DATA BASES**

**GRUPO: 3CM10**

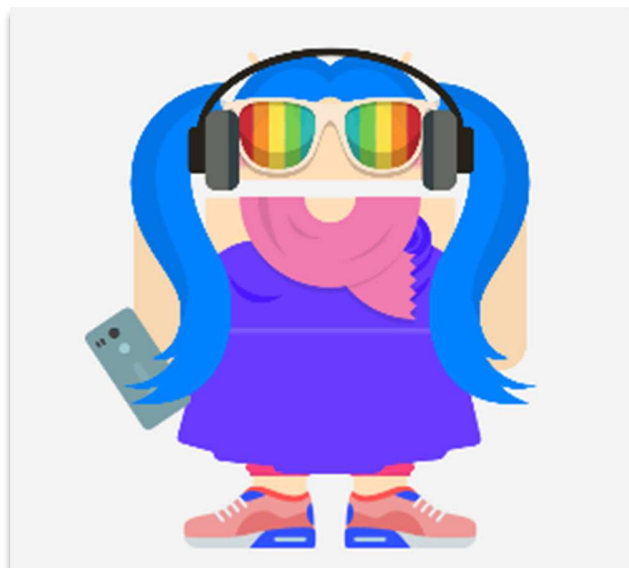
**PROFESOR: HERNÁNDEZ CONTRERAS EULER**

**NOMBRE: SALDAÑA AGUILAR GABRIELA**

**NO. DE PRÁCTICA: 2**

**PRACTICA: CONSULTAS DE SELECCIÓN**

**FECHA DE ENTREGA: 22/02/2016**



# INDICE

|                                |    |
|--------------------------------|----|
| MARCO TEÓRICO.....             | 3  |
| DESARROLLO.....                | 4  |
| CONCLUSIÓN Y BIBLIOGRAFÍA..... | 11 |

## MARCO TEORICO

### SELECT

Las consultas de selección se utilizan para indicar al motor de datos que devuelva información de las bases de datos, esta información es devuelta en forma de conjunto de registros que se pueden almacenar en un objeto recordset

Este conjunto de registros puede ser modificable.

Consultas básicas

La sintaxis básica de una consulta de selección es la siguiente:

SELECT

Campos

FROM

Tabla

### DEVOLVER LITERALES

En determinadas ocasiones nos puede interesar incluir una columna con un texto fijo en una consulta de selección:

SELECT

Empleados.Nombre, 'Tarifa semanal: ', Empleados.TarifaHora \* 40

FROM

Empleados

WHERE

Empleados.Cargo = 'Electricista'

### ORDENAR LOS REGISTROS

Adicionalmente se puede especificar el orden en que se desean recuperar los registros de las tablas mediante la cláusula ORDER BY Lista de Campos. En donde Lista de campos representa los campos a ordenar. Ejemplo:

```
SELECT
```

```
   CodigoPostal, Nombre, Telefono
```

```
FROM
```

```
   Clientes
```

```
ORDER BY
```

```
   Nombre
```

Incluso se puede especificar el orden de los registros: ascendente mediante la cláusula (ASC - se toma este valor por defecto) ó descendente (DESC)

## ALTER TABLE

statement is used to add, delete, or modify columns in an existing table.

### SYNTAXIS

To add a column in a table, use the following syntax:

```
ALTER TABLE table_name
```

```
ADD column_name datatype
```

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):

```
ALTER TABLE table_name
```

```
DROP COLUMN column_name
```

To change the data type of a column in a table, use the following syntax:

```
ALTER TABLE table_name
```

```
ALTER COLUMN column_name datatype
```

## CREATE TABLE

The CREATE TABLE statement is used to create a table in a database.

Tables are organized into rows and columns; and each table must have a name.

SQL CREATE TABLE Syntax

```
CREATE TABLE table_name
(
column_name1 data_type(size),
column_name2 data_type(size),
column_name3 data_type(size),
....
);
```

The column\_name parameters specify the names of the columns of the table. The data\_type parameter specifies what type of data the column can hold (e.g. varchar, integer, decimal, date, etc.).

The size parameter specifies the maximum length of the column of the table.

## INSTRUCCIONES

### PRACTICA 2:/

source C:/Users/alumno/Desktop/elektra.sql

1\*Mostrar el nombre completo del cliente y su monto de crédito

```
SELECT c.nombre, c.apPaterno, c.apMaterno, cre.monto
FROM cliente c, pago p, credito cre
WHERE c.idcliente=p.idcliente
AND p.idcredito=cre.idcredito
ORDER BY c.appaterno, cre.monto; //el primero no lleva alias porque solo existe un appaterno
```

```
mysql> SELECT c.nombre, c.apPaterno, c.apMaterno, cre.monto
-> FROM cliente c, pago p, credito cre
-> WHERE c.idcliente=p.idcliente
-> AND p.idcredito=cre.idcredito
-> ORDER BY c.appaterno, cre.monto;
```

| nombre         | apPaterno | apMaterno | monto |
|----------------|-----------|-----------|-------|
| JOSE EDUARDO   | ACEVES    | RESENDIZ  | 9000  |
| ALEJANDRO      | ACOSTA    | GOMEZ     | 5000  |
| GANDHI         | AGUIRRE   | ENCISO    | 18000 |
| MAURICIO       | ALONSO    | MARES     | 18000 |
| MARIANA        | ANGELES   | GONZALEZ  | 9000  |
| DANIEL         | ARROYO    | MENDOZA   | 5000  |
| AGUSTIN URIEL  | ARVIZU    | MENDOZA   | 18000 |
| GABRIELA       | BASURTO   | SEGOVIA   | 9000  |
| Yael           | BASURTO   | ESQUIVEL  | 13000 |
| LUIS ALBERTO   | BAUTISTA  | MORA      | 13000 |
| ALBERTO        | BERNAL    | SALGADO   | 18000 |
| BRENDA         | BLANCO    | ALVAREZ   | 5000  |
| ARTURO         | BLANCO    | GONZALEZ  | 18000 |
| EDGAR          | BRAVO     | ALVAREZ   | 9000  |
| JESUS          | BRIZUELA  | ALQUICIRA | 5000  |
| ANDREA         | BURGOS    | MADRIGAL  | 13000 |
| GUSTAVO        | CALDERON  | JUAREZ    | 5000  |
| PASTOR ESTEBAN | CAMARGO   | IGLESIAS  | 9000  |

2\*Mostrar el nombre del producto , su marca y su precioUnitario de aquellos productos que pertenecen a la categoria transporte.

```

SELECT p.nombre, p.marca, p.precioUnitario
FROM producto p, subcategoria sub, categoria c
WHERE p.idsubcategoria=sub.idsubcategoria
AND sub.idcategoria=c.idcategoria
AND c.nombre like "%transpo%"
ORDER BY p.preciounitario;

```

```

SELECT p.nombre, p.marca, p.precioUnitario
FROM producto p, subcategoria sub, categoria c
WHERE p.idsubcategoria=sub.idsubcategoria
AND sub.idcategoria=c.idcategoria
AND c.nombre like "%Deport%"
ORDER BY p.preciounitario;

```

```

--+
30 rows in set (0.00 sec)

mysql> SELECT p.nombre, p.marca, p.precioUnitario
-> FROM producto p, subcategoria sub, categoria c
-> WHERE p.idsubcategoria=sub.idsubcategoria
-> AND sub.idcategoria=c.idcategoria
-> AND c.nombre like "%transpo%"
-> ORDER BY p.preciounitario;
Empty set (0.02 sec)

mysql>

```

```

mysql>
mysql> SELECT p.nombre, p.marca, p.precioUnitario
-> FROM producto p, subcategoria sub, categoria c
-> WHERE p.idsubcategoria=sub.idsubcategoria
-> AND sub.idcategoria=c.idcategoria
-> AND c.nombre like "%Deport%"
-> ORDER BY p.preciounitario;
+-----+-----+-----+
| nombre                                     | marca | precioUnitario |
+-----+-----+-----+
| Eliptica B.H. BRISA                       | BH    | 4999            |
| Caminadora BH modelo PIONEER CLASSIC     | BH    | 7999            |
+-----+-----+-----+
2 rows in set (0.00 sec)

```

3\*Mostrar los datos correspondientes de las sucursales existentes en el estado de colima

```

SELECT *
FROM tienda
WHERE estado="Colima"
ORDER BY nombre;

```

```
mysql> SELECT *
mysql> -> FROM tienda
mysql> -> WHERE estado="Colima"
mysql> -> ORDER BY nombre;
```

| idtienda | nombre                      | estado | cp    | tel        | direccion   |
|----------|-----------------------------|--------|-------|------------|---|
| 1768     | APA QUESERIA                | Colima | 28500 | 3123952189 | 20 DE NOVIEMBRE 16 CUAUHTEMOC                         |
| 6117     | EKT COLIMA VILLA DE ALVAREZ | Colima | 28970 | 3215504590 | AV MARIA AHUMADA S/N VILLA DE ALVAREZ CENTRO          |
| 2273     | EKT TECOMAN                 | Colima | 28100 | 3133240178 | MEDELLIN 336 TECOMAN CENTRO                           |
| 7689     | MEGA COLIMA                 | Colima | 28000 | 3123304093 | NIGROMANTE 29 COLIMA CENTRO                           |
| 175      | MEGA MANZANILLO SALA        | Colima | 28869 | 3143332222 | BLVD MIGUEL DE LA MADRID 11500 OBISPO SERGIO MENDEZ A |
| 3052     | TT COLIMA                   | Colima | 28000 | 3123146360 | MADERO 365 COLIMA CENTRO                              |

6 rows in set (0.00 sec)

4\*Que clientes hicieron pagos el día 10 de marzo del 2010?  
Mostrar su nombre, apellido paterno y el monto de crédito correspondiente.

```
SELECT c.nombre, c.appaterno, cre.monto
FROM pago p, cliente c, credito cre
WHERE c.idcliente=p.idcliente
AND p.idcredito=cre.idcredito
AND p.fechapago like "%2010-03-10%"
ORDER BY cre.monto, appaterno;
```

```
mysql>
mysql> SELECT c.nombre, c.appaterno, cre.monto
mysql> -> FROM pago p, cliente c, credito cre
mysql> -> WHERE c.idcliente=p.idcliente
mysql> -> AND p.idcredito=cre.idcredito
mysql> -> AND p.fechapago like "%2010-03-10%"
mysql> -> ORDER BY cre.monto, appaterno;
```

| nombre     | appaterno | monto |
|------------|-----------|-------|
| FAUSTO NOE | LIMA      | 9000  |
| ARTURO     | PACHECO   | 9000  |
| IVAN       | MARTINEZ  | 18000 |
| RAFAEL     | PEREZ     | 18000 |

4 rows in set, 1 warning (0.00 sec)

5\*Modificar la definición de la relación cliente para almacenar correos

Hay que crear la relación de correo para poder almacenar un atributo multivalor  
, PRIMERO ANTES DE BORRAR HAY QUE HACER UNA COPIA DE LOS CORREOS, la foreign key sera la de la tabla sobre la que estamos extrayendo. y se hace una unión de llave primaria A)

```
CREATE table correo(
idcliente int not null,
```

email varchar(50) not null,  
primary key(idcliente,email),  
foreign key (idcliente) references cliente(idcliente)  
on delete cascade on update cascade  
);

## B) LLENAR DE DATOS LA RELACION

```
INSERT INTO correo  
SELECT idcliente, email  
FROM cliente  
WHERE EMAIL IS NOT NULL;
```

## C) ELIMINAR EL CAMPO EMAIL DE LA RELACION CLIENTE

```
alter table CLIENTE DROP COLUMN EMAIL;
```

```
mysql> CREATE table correo(  
-> idcliente int not null,  
-> email varchar(50) not null,  
-> primary key(idcliente,email),  
-> foreign key (idcliente) references cliente(idcliente)  
-> on delete cascade on update cascade  
-> );  
Query OK, 0 rows affected (0.26 sec)  
  
mysql> INSERT INTO correo  
-> SELECT idcliente, email  
-> FROM cliente;  
ERROR 1048 (23000): Column 'email' cannot be null  
mysql>  
mysql> INSERT INTO correo  
-> SELECT idcliente, email  
-> FROM cliente  
-> WHERE EMAIL IS NOT NULL;  
Query OK, 155 rows affected (0.16 sec)  
Records: 155 Duplicates: 0 Warnings: 0  
  
mysql> SELECT * FROM CORREO;  
+-----+-----+  
| idcliente | email |  
+-----+-----+  
| 1 | nuria_12314@hotmail.com |  
| 2 | brendaloka_1990@hotmail.com |  
| 3 | colin_erick_vz@hotmail.com |  
| 4 | perico33@hotmail.com |  
| 5 | yamildelgado@hotmail.com |  
| 6 | chitilis7@yahoo.com.mx |  
| 7 | fuenlin@hotmail.com |  
| 8 | jimang7@gmail.com |  
| 9 | haga87@gmail.com |  
| 10 | edward_grenader@live.com.mx |
```

6\*Mostrar el nombre de la sucursal, así como el estado en se ubican las sucursales que están en las sig cp... 20010,30470,23450

```
SELECT nombre,estado
FROM tienda
WHERE cp in(2010,30470,23450);
```

```
mysql>
mysql> SELECT nombre,estado
-> FROM tienda
-> WHERE cp in(2010,30470,23450);
```

| nombre              | estado              |
|---------------------|---------------------|
| MEGA VILLAFLORES    | Chiapas             |
| MEGA CABO SAN LUCAS | Baja California Sur |
| EKT CABO SAN LUCAS  | Baja California Sur |

3 rows in set (0.00 sec)

7\*Mostrar el nombre de los productos, marca y subcategoría de aquellos productos que son suministrados en la sucursal ekt tecate

```
SELECT p.nombre, p.marca, sub.nombre
FROM producto p, subcategoria sub, tiendaproducto tp, tienda t
WHERE t.idtienda=tp.idtienda
AND tp.idproducto=p.idproducto
AND p.idsubcategoria=sub.idsubcategoria
AND T.NOMBRE LIKE "%ekt tecate%"
ORDER BY p.marca, p.nombre desc, sub.nombre;
```

```
mysql> SELECT p.nombre, p.marca, sub.nombre
-> FROM producto p, subcategoria sub, tiendaproducto tp, tienda t
-> WHERE t.idtienda=tp.idtienda
-> AND tp.idproducto=p.idproducto
-> AND p.idsubcategoria=sub.idsubcategoria
-> AND T.NOMBRE LIKE "%ekt tecate%";
```

| nombre                               | marca     | nombre                |
|--------------------------------------|-----------|-----------------------|
| Reproductor Mp4 8GB EKT              | EKT       | Audio Portatil        |
| Reproductor Mp3 4GB SONY             | SONY      | Audio Portatil        |
| Refrigerador 25 pies WHIRLPOOL       | WHIRLPOOL | Refrigeradores        |
| Caminadora BH modelo PIONEER CLASSIC | BH        | Aparatos de Ejercicio |

4 rows in set (0.00 sec)

```
mysql> SELECT p.nombre, p.marca, sub.nombre
-> FROM producto p, subcategoria sub, tiendaproducto tp, tienda t
-> WHERE t.idtienda=tp.idtienda
-> AND tp.idproducto=p.idproducto
-> AND p.idsubcategoria=sub.idsubcategoria
-> AND T.NOMBRE LIKE "%ekt tecate%"
-> ORDER BY p.marca, p.nombre desc, sub.nombre;
```

| nombre                               | marca     | nombre                |
|--------------------------------------|-----------|-----------------------|
| Caminadora BH modelo PIONEER CLASSIC | BH        | Aparatos de Ejercicio |
| Reproductor Mp4 8GB EKT              | EKT       | Audio Portatil        |
| Reproductor Mp3 4GB SONY             | SONY      | Audio Portatil        |
| Refrigerador 25 pies WHIRLPOOL       | WHIRLPOOL | Refrigeradores        |

4 rows in set (0.00 sec)



## REALIZAR LAS SIGUIENTES MODIFICACIONES

1\* renombrar el campo nombre en la relacion producto y llamarle producto..

```
ALTER TABLE producto  
CHANGE COLUMN nombre// renombrar un campo o cambiar el y tamaño o tipo de datos  
producto varchar(100);
```

2\*eliminar el sexo en cliente...

```
ALTER TABLE cliente  
DROP COLUMN sexo;
```

```
mysql> ALTER TABLE producto  
-> CHANGE COLUMN nombre  
-> producto varchar(100);  
Query OK, 0 rows affected (0.07 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> ALTER TABLE cliente  
-> DROP COLUMN sexo;  
Query OK, 0 rows affected (0.34 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> desc cliente;
```

| Field     | Type        | Null | Key | Default | Extra |
|-----------|-------------|------|-----|---------|-------|
| idcliente | int(11)     | NO   | PRI | NULL    |       |
| nombre    | varchar(45) | YES  |     | NULL    |       |
| apPaterno | varchar(45) | YES  |     | NULL    |       |
| apMaterno | varchar(45) | YES  |     | NULL    |       |
| salario   | double      | YES  |     | NULL    |       |

```
5 rows in set (0.03 sec)
```

3\*Agregar una relacion para almacenar informacion de los empleados para las sucursales existentes asociar dicha relacion con tienda o sucursal

empleado(idempleado PK, nombre, salario, email, edad, dir, idtienda Fk);

```
CREATE TABLE empleado(  
    idempleado int not null primary key,  
    nombre varchar(80),  
    salario double,  
    email varchar(50),  
    edad int,  
    dir varchar(300),  
    idtienda int,  
    foreign key(idtienda) references tienda(idtienda)  
    on delete cascade on update cascade  
);
```

```

mysql> CREATE TABLE empleado(
-> idempleado int not null primary key,
-> nombre varchar(80),
-> salario double,
-> email varchar(50),
-> edad int,
-> dir varchar(300),
-> idtienda int,
-> foreign key(idtienda) references tienda(idtienda)
-> on delete cascade on update cascade
-> );
Query OK, 0 rows affected (0.37 sec)

mysql> desc empleado
-> 1
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idempleado | int(11) | NO | PRI | NULL | |
| nombre | varchar(80) | YES | | NULL | |
| salario | double | YES | | NULL | |
| email | varchar(50) | YES | | NULL | |
| edad | int(11) | YES | | NULL | |
| dir | varchar(300) | YES | | NULL | |
| idtienda | int(11) | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> show create table empleado;
+-----+-----+
| Table | Create Table |
+-----+-----+
| empleado | CREATE TABLE `empleado` (
  `idempleado` int(11) NOT NULL,
  `nombre` varchar(80) DEFAULT NULL,
  `salario` double DEFAULT NULL,
  `email` varchar(50) DEFAULT NULL,
  `edad` int(11) DEFAULT NULL,
  `dir` varchar(300) DEFAULT NULL,
  `idtienda` int(11) DEFAULT NULL,
  PRIMARY KEY (`idempleado`),
  KEY `idtienda` (`idtienda`),
  CONSTRAINT `empleado_ibfk_1` FOREIGN KEY (`idtienda`) REFERENCES `tienda` (`idtienda`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.02 sec)

```

4\*eliminar la asociacion existente entre la relacion pago y la relacion credito

show create table pago;//ver los constraints

//para eliminar la llave foranea de una tabla aun queda el campo en la tabla

ALTER TABLE pago DROP FOREIGN KEY  
pago\_ibfk\_1;

```

mysql> ALTER TABLE pago DROP FOREIGN KEY
-> pago_ibfk_1;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

5\*AUmentar el unmero de caracterers sobre el campo descripcion en la relacion producto, almacenar 450 caracteres

ALTER TABLE producto  
MODIFY COLUMN  
descripcion varchar(450);

```
mysql> ALTER TABLE producto
-> MODIFY COLUMN
-> descripcion varchar(450);
Query OK, 30 rows affected (0.85 sec)
Records: 30 Duplicates: 0 Warnings: 0

mysql> desc producto
-> ;
```

| Field          | Type         | Null | Key | Default | Extra |
|----------------|--------------|------|-----|---------|-------|
| idproducto     | int(11)      | NO   | PRI | NULL    |       |
| producto       | varchar(100) | YES  |     | NULL    |       |
| descripcion    | varchar(450) | YES  |     | NULL    |       |
| precioUnitario | double       | YES  |     | NULL    |       |
| marca          | varchar(45)  | YES  |     | NULL    |       |
| idsubcategoria | int(11)      | YES  | MUL | NULL    |       |

```
6 rows in set (0.02 sec)
```

## CONCLUSIÓN

Se pudo dar repaso a temas antes vistos en bases de datos, en esta práctica especialmente se tomó el repaso de cómo hacer consultas de selección básicas, se retomaron ciertos conocimientos como :

\*Interpretar un E-R para poder sacar las consultas

\*Atributos de reunión

\*Ordenamiento

\*Wild Card

\*Alias

\*Condiciones WHERE

\*Cargar un script de una BD

Entre otros. El repaso fue de mucha ayuda porque nos servirá tanto para prácticas posteriores como para el examen, también porque es conveniente tratar de generar las consultas lo mejor posible en el menor tiempo.

Se retomaron comandos como lo son el ALTER TABLE, CREATE TABLE entre otras operaciones que se realizan para modificar campos específicos de una tabla o para crear una tabla en si como en el caso que nos piden generar una nueva relación, o cuando es necesario normalizar una ya existente.

## BIBLIOGRAFÍA

<http://www.desarrolloweb.com/articulos/2257.php>

<https://dev.mysql.com/doc/refman/5.5/en/select.html>