**Ingreso de datos**

USE restaurantedb;

SHOW TABLES;

-- TIPO USERS

INSERT INTO tipousers VALUES ( 0,'administrador');

INSERT INTO tipousers VALUES ( 0,'vendedor' );

INSERT INTO tipousers VALUES ( 0,'cliente' );

SELECT \* FROM tipousers;

-- USUARIOS--

INSERT INTO Usuarios VALUES ( 0,'camilo@gmail.com',sha('camilo2345'),1);

INSERT INTO Usuarios VALUES ( 0,'juandavid@gmail.com',sha('345676'),1);

INSERT INTO Usuarios VALUES ( 0,'maria9080@gmail.com',sha('m12345a'),1);

INSERT INTO Usuarios VALUES ( 0,'carlos@gmail.com',sha('l458962'),1);

INSERT INTO Usuarios VALUES ( 0,'luisr@gmail.com',sha('m8965732'),1);

INSERT INTO Usuarios VALUES ( 0,'juanPerez@hotmail.com',sha('n789541'),1);

INSERT INTO Usuarios VALUES ( 0,'Eduardo5670@gmail.com',sha('c45698'),1);

INSERT INTO Usuarios VALUES ( 0,'juanD10@hotmail.com',sha('k69857'),1);

INSERT INTO Usuarios VALUES ( 0,'Ramiro35Ortiz@gmail.com',sha('p45239'),1);

INSERT INTO Usuarios VALUES ( 0,'sofiajh70@hotmail.com',sha('r87856'),1);

SELECT \* FROM usuarios;

-- ROLES--

INSERT INTO roles VALUES ( 0,1,1,'2020-09-10',1);

INSERT INTO roles VALUES ( 0,2,2,'2019-07-20',1);

INSERT INTO roles VALUES ( 0,3,3,'2018-08-22',1);

INSERT INTO roles VALUES ( 0,4,2,'2018-08-22',1);

INSERT INTO roles VALUES ( 0,4,1,'2018-08-22',1);

INSERT INTO roles VALUES ( 0,5,2,'2018-08-22',1);

INSERT INTO roles VALUES ( 0,5,3,'2018-08-22',1);

INSERT INTO roles VALUES ( 0,6,3,'2018-08-22',1);

INSERT INTO roles VALUES ( 0,7,2,'2018-08-22',1);

INSERT INTO roles VALUES ( 0,8,3,'2018-08-22',1);

INSERT INTO roles VALUES ( 0,9,2,'2018-08-22',1);

INSERT INTO roles VALUES ( 0,10,2,'2018-08-22',1);

SELECT \* FROM roles;

-- Datosuser--

INSERT INTO datosusers VALUE (0,1,'Jhon','Medina','1986-01-12', 0,'cr 255 #85-12','4444444');

INSERT INTO datosusers VALUE (0,2,'Alexis','Cardona','2000-08-01', 0,'cr 288 #85-12','333333');

INSERT INTO datosusers VALUE (0,3,'Esteban','Flores','1986-09-07', 0,'cr 277 #85-12','22222');

INSERT INTO datosusers VALUE (0,4,'ramon','perez','1978-08-12', 0,'cr198 #83-11','12345');

INSERT INTO datosusers VALUE (0,5,'luis','gomez','1985-02-01', 0,'cr85 #12-05','98625');

INSERT INTO datosusers VALUE (0,6,'juan','ortiz','1955-02-01', 0,'cr129 #12-05','8888888');

INSERT INTO datosusers VALUE (0,7,'luis','gomez','1985-02-01', 0,'cr92 #12-05','999999');

INSERT INTO datosusers VALUE (0,8,'cristian','catrillon','1986-03-01', 0,'cr98 #12-05','555555');

INSERT INTO datosusers VALUE (0,9,'ramiro','portiz','1935-03-01', 0,'cr24 #12-08','333333');

INSERT INTO datosusers VALUE (0,10,'lola','perez','1935-03-01', 1,'cr24 #12-08','333333');

SELECT \* FROM datosusers;

-- VENDEDOR--

INSERT INTO vendedor VALUE (0,1,'2012-08-14',1);

INSERT INTO vendedor VALUE (0,2,'2019-02-17',0);

INSERT INTO vendedor VALUE (0,3,'2009-07-20',1);

SELECT \* FROM vendedor;

-- TIPOPRO--

INSERT INTO tipoPro VALUE (0,'bebidas');

INSERT INTO tipoPro VALUE (0,'lacteos');

INSERT INTO tipoPro VALUE (0,'licores');

INSERT INTO tipoPro VALUE (0,'comidas');

INSERT INTO tipoPro VALUE (0,'postres');

SELECT \* FROM tipoPro;

select \* from roles;

select \* from datosusers;

-- PRODUCTOS--

INSERT INTO productos VALUE (0,1,'jugo de naranja','10','naranja-agua-azucar','2020-08-14',5000);

INSERT INTO productos VALUE (0,2,'yogur','30','leche-agua-azucar','2020-08-14',3000);

INSERT INTO productos VALUE (0,3,'ron','2','soda,hielo,ron','0000-00-00',10000);

INSERT INTO productos VALUE (0,4,'Ceviche de Chicharrón','3','ají de tomate de árbol y cebolla morada, con chips de plátano','0000-00-00',23000);

INSERT INTO productos VALUE (0,4,'Entrada Típica','2','Chicharrón, chorizo, morcilla, patacón y arepa, con hogao y limón','0000-00-00',28500);

INSERT INTO productos VALUE (0,4,'Crema de Maíz ','2','crema de maíz...','0000-00-00',14500);

INSERT INTO productos VALUE (0,4,'Cazuela Típica','3','sopa de fríjoles con carne desmechada, chorizo, maicitos y viruta de papa, acompañada de chicharroncitos, trocitos de plátano maduro, arroz y aguacate.','2020-08-14',36000);

INSERT INTO productos VALUE (0,4,'Ajiaco','2',' sopa Santafereña a base de pollo, papas capira, pastusa y criolla, mazorcas y guasca, acompañada de crema de leche y alcaparras, arroz y aguacate','0000-00-00', 35500);

INSERT INTO productos VALUE (0,4,'Lengua en salsa de Carne ','3','Sudado de Lengua servida con papa, yuca, arroz y aguacate.','0000-00-00',39500);

INSERT INTO productos VALUE (0,4,'Sancocho Antioqueño ','2','sopa Antioqueña a base de res y cerdo, con mazorca, arracacha, plátano, papa y yuca, acompañado de arroz y aguacate','0000-00-00',42500);

INSERT INTO productos VALUE (0,4,'Solomito','3','Nuestro solomito es servido con papa asada o papa rústica. Acompañadas de ensalada.','0000-00-00', 46500);

INSERT INTO productos VALUE (0,4,'Planchita Mixta','3','Planchita mixta con combinación de carnes (solomito, cañon de cerdo, pollo) y pescado blanco, acompañada con ensalada fresca con vinagreta de albahaca y papa asada con sourcream de la casa.','0000-00-00', 50500);

INSERT INTO productos VALUE (0,5,'Flan de Leche','2','El flan de leche Hatoviejo es fresco, esponjoso y está cubierto de caramelo.','0000-00-00',12500);

INSERT INTO productos VALUE (0,5,'Postre de Milo','2','Es un postre hecho a base de Milo, galleta y crema.','0000-00-00',12000);

INSERT INTO productos VALUE (0,4,'Lomito de Res encostrado en 3 pimientas','2','Lomito de Res encostrado en 3 Pimientas, con salsa de hongos, acompañado de puré de papa criolla y tomates cherry.','0000-00-00',51.500);

INSERT INTO productos VALUE (0,4,'Medallones de Solomito','3','Jugosos Medallones de Solomito en salsa de vino tinto y champiñones acompañados de papa rústica y una fresca ensalada con aguacate.','0000-00-00',48500);

INSERT INTO productos VALUE (0,5,'Postre de Milo','2','Es un postre hecho a base de Milo, galleta y crema.','0000-00-00',12000);

INSERT INTO productos VALUE (0,5,'Mousse de chocolate','35','chocolate-matequilla-huevos-azucar-sal','2020-12-17',10000);

INSERT INTO productos VALUE (0,5,'Mousse de chocolate blanco','35','chocolate blanco-matequilla-huevos-azucar-sal','2020-12-17',10000);

INSERT INTO productos VALUE (0,5,'tartas','120','harina-matequilla-huevos-azucar-sal','2020-10-17',30000);

INSERT INTO productos VALUE (0,5,'pastel','120','harina-matequilla-huevos-azucar-sal-leche','2020-10-17',20000);

INSERT INTO productos VALUE (0,5,'ARROZ CON LECHE','1','leche-leche condesada-crema de leche-coco-arroz','2020-08-17',12000);

INSERT INTO productos VALUE (0,5,'TRUFAS DE CHOCOLATE','20','harina-matequilla-huevos-azucar-chocolate','2020-12-17',8000);

INSERT INTO productos VALUE (0,5,'Biscocho','35','harina-matequilla-huevos-azucar-chocolate','2020-11-17',10000);

INSERT INTO productos VALUE (0,5,'GALLETAS','30','harina-matequilla-huevos-azucar-chocolate','2020-09-17',8000);

SELECT \* FROM productos;

-- TIPOTRAN--

INSERT INTO tipotran VALUE (0,'en proceso');

INSERT INTO tipotran VALUE (0,'cancelado');

INSERT INTO tipotran VALUE (0,'finalizado');

SELECT \* FROM tipotran;

-- FOPAGO--

INSERT INTO fopago VALUE (0,'Efectivo');

INSERT INTO fopago VALUE (0,'Targeta de credito');

INSERT INTO fopago VALUE (0,'Targeta debito');

INSERT INTO fopago VALUE (0,'Giro electronico');

SELECT \* FROM fopago;

-- TRANSACCIONES--

INSERT INTO transacciones VALUE(0,3,1,1,001,'2020-02-13',40000,1);

INSERT INTO transacciones VALUE(0,1,2,2,002,'2020-03-17',80000,0);

INSERT INTO transacciones VALUE(0,2,3,3,003,'2020-03-17',50000,1);

INSERT INTO transacciones VALUE(0,2,4,1,004,'2020-03-17',200000,1);

INSERT INTO transacciones VALUE(0,3,5,3,005,'2020-02-13',40000,1);

INSERT INTO transacciones VALUE(0,1,6,2,006,'2020-03-17',80000,0);

INSERT INTO transacciones VALUE(0,2,7,3,007,'2020-03-17',50000,1);

INSERT INTO transacciones VALUE(0,2,8,2,008,'2020-03-17',200000,1);

INSERT INTO transacciones VALUE(0,3,9,1,009,'2020-02-13',40000,1);

INSERT INTO transacciones VALUE(0,1,10,2,010,'2020-03-17',80000,0);

INSERT INTO transacciones VALUE(0,2,10,3,011,'2020-03-17',50000,1);

INSERT INTO transacciones VALUE(0,2,9,1,012,'2020-03-17',200000,1);

INSERT INTO transacciones VALUE(0,3,8,3,013,'2020-02-13',40000,1);

INSERT INTO transacciones VALUE(0,1,7,2,014,'2020-03-17',80000,0);

INSERT INTO transacciones VALUE(0,2,6,3,015,'2020-03-17',50000,1);

INSERT INTO transacciones VALUE(0,2,7,2,016,'2020-03-17',200000,1);

INSERT INTO transacciones VALUE(0,3,5,1,017,'2020-02-13',40000,1);

INSERT INTO transacciones VALUE(0,1,4,2,018,'2020-03-17',80000,0);

INSERT INTO transacciones VALUE(0,2,3,3,019,'2020-03-17',50000,1);

INSERT INTO transacciones VALUE(0,2,2,1,020,'2020-03-17',200000,1);

INSERT INTO transacciones VALUE(0,3,1,2,021,'2020-02-13',40000,1);

INSERT INTO transacciones VALUE(0,1,1,2,022,'2020-03-17',80000,0);

INSERT INTO transacciones VALUE(0,2,2,3,023,'2020-03-17',50000,1);

INSERT INTO transacciones VALUE(0,2,3,1,024,'2020-03-17',200000,1);

INSERT INTO transacciones VALUE(0,2,4,1,025,'2020-04-19',850000,1);

INSERT INTO transacciones VALUE(0,3,5,3,026,'2020-04-19',47000,1);

INSERT INTO transacciones VALUE(0,1,6,2,027,'2020-04-19',87800,0);

INSERT INTO transacciones VALUE(0,2,7,3,028,'2020-04-19',54000,1);

INSERT INTO transacciones VALUE(0,2,8,1,029,'2020-04-19',560700,1);

INSERT INTO transacciones VALUE(0,3,9,1,030,'2020-04-19',24000,1);

INSERT INTO transacciones VALUE(0,1,10,2,031,'2020-04-19',74000,0);

INSERT INTO transacciones VALUE(0,3,1,1,032,'2020-08-13',40000,1);

INSERT INTO transacciones VALUE(0,1,2,2,033,'2020-08-17',80000,0);

INSERT INTO transacciones VALUE(0,2,3,3,034,'2020-07-27',50000,1);

INSERT INTO transacciones VALUE(0,2,4,1,035,'2020-07-14',200000,1);

INSERT INTO transacciones VALUE(0,3,5,2,036,'2020-07-13',40000,1);

INSERT INTO transacciones VALUE(0,1,6,2,037,'2020-03-15',80000,0);

INSERT INTO transacciones VALUE(0,2,7,3,038,'2020-09-17',50000,1);

INSERT INTO transacciones VALUE(0,2,8,3,039,'2020-04-18',200000,1);

INSERT INTO transacciones VALUE(0,3,9,1,040,'2020-02-13',40000,1);

INSERT INTO transacciones VALUE(0,1,10,2,041,'2020-03-17',80000,0);

INSERT INTO transacciones VALUE(0,2,10,3,042,'2020-05-16',50000,1);

INSERT INTO transacciones VALUE(0,2,9,1,043,'2020-03-17',200000,1);

INSERT INTO transacciones VALUE(0,3,8,2,044,'2020-06-13',40000,1);

INSERT INTO transacciones VALUE(0,1,7,2,045,'2020-03-17',80000,0);

INSERT INTO transacciones VALUE(0,2,6,3,046,'2020-04-14',50000,1);

INSERT INTO transacciones VALUE(0,2,7,1,047,'2020-03-01',200000,1);

INSERT INTO transacciones VALUE(0,3,5,1,048,'2020-02-13',40000,1);

INSERT INTO transacciones VALUE(0,1,4,2,049,'2020-03-17',80000,0);

INSERT INTO transacciones VALUE(0,2,3,1,050,'2020-05-15',50000,1);

INSERT INTO transacciones VALUE(0,2,2,1,051,'2020-03-17',200000,1);

INSERT INTO transacciones VALUE(0,3,1,2,052,'2020-02-13',40000,1);

INSERT INTO transacciones VALUE(0,1,1,2,053,'2020-03-15',80000,0);

INSERT INTO transacciones VALUE(0,2,2,3,054,'2020-08-07',50000,1);

INSERT INTO transacciones VALUE(0,2,3,3,055,'2020-03-17',200000,1);

INSERT INTO transacciones VALUE(0,2,4,1,056,'2020-04-19',850000,1);

INSERT INTO transacciones VALUE(0,3,5,2,057,'2020-08-19',47000,1);

INSERT INTO transacciones VALUE(0,1,6,2,058,'2020-04-14',87800,0);

INSERT INTO transacciones VALUE(0,2,7,3,059,'2020-05-19',54000,1);

INSERT INTO transacciones VALUE(0,2,8,1,060,'2020-04-18',560700,1);

INSERT INTO transacciones VALUE(0,3,9,1,061,'2020-06-19',24000,1);

INSERT INTO transacciones VALUE(0,1,10,2,062,'2020-04-19',74000,0);

SELECT \* FROM transacciones;

select \* from usuarios;

-- DeTATRANSACCION--

INSERT INTO detatransaccion VALUE(0,1,3,1,3,20);

INSERT INTO detatransaccion VALUE(0,2,1,2,1,70);

INSERT INTO detatransaccion VALUE(0,3,2,3,2,10);

INSERT INTO detatransaccion VALUE(0,5,3,5,7,20);

INSERT INTO detatransaccion VALUE(0,6,1,6,17,70);

INSERT INTO detatransaccion VALUE(0,7,2,7,26,10);

INSERT INTO detatransaccion VALUE(0,8,2,8,20,50);

INSERT INTO detatransaccion VALUE(0,9,3,9,3,20);

INSERT INTO detatransaccion VALUE(0,10,1,10,1,70);

INSERT INTO detatransaccion VALUE(0,11,2,10,22,10);

INSERT INTO detatransaccion VALUE(0,12,2,9,4,50);

INSERT INTO detatransaccion VALUE(0,13,3,8,24,20);

INSERT INTO detatransaccion VALUE(0,14,1,7,7,70);

INSERT INTO detatransaccion VALUE(0,15,2,6,9,10);

INSERT INTO detatransaccion VALUE(0,16,2,7,24,50);

INSERT INTO detatransaccion VALUE(0,17,3,5,3,20);

INSERT INTO detatransaccion VALUE(0,18,1,4,1,70);

INSERT INTO detatransaccion VALUE(0,19,2,3,2,10);

INSERT INTO detatransaccion VALUE(0,20,2,2,4,50);

INSERT INTO detatransaccion VALUE(0,21,3,1,7,20);

INSERT INTO detatransaccion VALUE(0,22,1,1,17,70);

INSERT INTO detatransaccion VALUE(0,23,2,2,21,10);

INSERT INTO detatransaccion VALUE(0,24,2,3,14,50);

INSERT INTO detatransaccion VALUE(0,25,2,4,4,50);

INSERT INTO detatransaccion VALUE(0,26,3,5,7,20);

INSERT INTO detatransaccion VALUE(0,27,1,6,17,70);

INSERT INTO detatransaccion VALUE(0,28,2,7,24,10);

INSERT INTO detatransaccion VALUE(0,29,2,8,18,50);

INSERT INTO detatransaccion VALUE(0,30,3,9,3,20);

INSERT INTO detatransaccion VALUE(0,31,1,10,11,70);

INSERT INTO detatransaccion VALUE(0,32,3,1,15,70);

INSERT INTO detatransaccion VALUE(0,33,1,2,3,27);

INSERT INTO detatransaccion VALUE(0,34,2,3,1,70);

INSERT INTO detatransaccion VALUE(0,35,2,4,2,10);

INSERT INTO detatransaccion VALUE(0,36,3,5,4,50);

INSERT INTO detatransaccion VALUE(0,37,1,6,7,20);

INSERT INTO detatransaccion VALUE(0,38,2,7,17,76);

INSERT INTO detatransaccion VALUE(0,39,2,8,44,40);

INSERT INTO detatransaccion VALUE(0,40,3,9,3,20);

INSERT INTO detatransaccion VALUE(0,41,1,10,1,70);

INSERT INTO detatransaccion VALUE(0,42,2,10,22,10);

INSERT INTO detatransaccion VALUE(0,43,2,9,4,50);

INSERT INTO detatransaccion VALUE(0,44,3,8,40,20);

INSERT INTO detatransaccion VALUE(0,45,1,7,7,70);

INSERT INTO detatransaccion VALUE(0,46,2,6,38,10);

INSERT INTO detatransaccion VALUE(0,47,2,7,44,55);

INSERT INTO detatransaccion VALUE(0,48,3,5,3,24);

INSERT INTO detatransaccion VALUE(0,49,1,4,1,72);

INSERT INTO detatransaccion VALUE(0,50,2,3,2,10);

INSERT INTO detatransaccion VALUE(0,51,2,2,4,50);

INSERT INTO detatransaccion VALUE(0,52,3,1,7,25);

INSERT INTO detatransaccion VALUE(0,53,1,1,17,70);

INSERT INTO detatransaccion VALUE(0,54,2,2,28,10);

INSERT INTO detatransaccion VALUE(0,55,2,3,44,50);

INSERT INTO detatransaccion VALUE(0,56,2,4,7,28);

INSERT INTO detatransaccion VALUE(0,57,3,5,17,73);

INSERT INTO detatransaccion VALUE(0,58,1,6,28,18);

INSERT INTO detatransaccion VALUE(0,59,2,7,44,50);

INSERT INTO detatransaccion VALUE(0,60,2,8,3,20);

INSERT INTO detatransaccion VALUE(0,61,3,9,7,35);

INSERT INTO detatransaccion VALUE(0,62,1,10,7,35);

SELECT \* FROM detatransaccion;

select u.emailUser, t.NombTiUs

from usuarios u

join roles r

on(u.idUsuarios = r.idUsuarios)

join tipousers t

on(t.idTipoUsers = r.idTipoUsers);

**funciones**

-- 1 SELECCIONA LOS NOMBRES QUE EMPIEZAN POR LA LETRA L

SELECT \* FROM datosusers WHERE Nombres LIKE 'L%';

-- 2 MUESTRA LOS DATOS DEL USUARIO DESDE LA IDENTIFICACION 2 HASTA LA 4

SELECT \* FROM datosusers WHERE idUsuarios BETWEEN 2 AND 4;

-- 3 MUESTRA LAS DIRECCIONES DE LOS DATOS DE USUARIO QUE COMIENZAN POR CR

SELECT \* FROM datosusers WHERE Direccion LIKE '%cr%';

-- 4 SELECCIONA LO USUARIOS CUYAS IDENTIFICACIONES SON PARES

SELECT \* FROM usuarios WHERE MOD (idUsuarios,2)=0;

-- 5 MUESTRA LOS DATOS DE DOS TABLAS SIEMPRE Y CUANDO SUS LLAVES PRIMARIAS SEAN SEMEJANTES

SELECT \* FROM tipopro INNER JOIN productos WHERE tipopro.idTipoProductos = productos.idProductos;

-- 6 CUENTA EL TOTAL DE USUARIOS

SELECT COUNT(\*) FROM usuarios;

-- 7 MUESTRA EL DIA DE LA SEMANA AL QUE CORREPONDE ESTA FECHA

SELECT DAYNAME('2020-08-14');

-- 8 COMBINA DOS TABLAS

SELECT \* FROM usuarios JOIN tipoUsers on idUsuarios = idUsuarios;

-- 9 ORGANIZA DATOS DE MENOR A MAYOR TANTO EN LETRAS COMO NUMEROS

select \*

from productos

order by nombreP asc;

-- 10 SELECCIONA EL NUMERO DE DIA AL QUE CORRESPONDE

SELECT DAYOFYEAR('2020-08-14');

-- 11 UN RAMDON

SELECT \* FROM productos ORDER BY RAND();

-- 12 CALCULA EL TOTAL DE SEMANAS DETERMINANDO EL MES Y LA FECHA

SELECT YEAR('2020-08-14'), WEEK('2020-08-14');

-- 13 MUESTRA LOS VALORES DE PRODUCTOS MENORES DE 10.000

select \*

from productos

where valorP < 10000;

-- 14 MUESTRA EL VALOR DE LOS PRODUCTOS MAYORE A 2.500

select \*

from productos

where valorP > 2500;

-- 15 MUESTRA EL VALOR DE LOS PRODUCTOS QUE SON MAYORES O IGUALES A 2.500

select \*

from productos

where valorP >= 2500;

-- 16 MUESTRA LOS USUARIOS QUE PERTENECEN A UN GENERO DIFERENTE DE 1

select \*

from datosusers

where genero <> 1;

-- 17 MUESTRA EL ID QUE ESTA GUARDADO COMO 3 EN LA TABLA TIPO PRODUCTO

select \*

from tipopro

where idTipoProductos= 3;

-- 18 MUESTRA LAS 3 PRIMERAS LETRAS QUE SE ENCUENTRAN UBICADAS A LA DERECHA

SELECT RIGHT('PRODUCTO', 3);

-- 19 MUESTRA LAS LETRAS DE ESA PALABRAS AL REVES

SELECT REVERSE('VENDEDOR');

-- 20 SUMAR LOS VALORES DE UNA COLUMNA --

SELECT SUM(valorTran) FROM transacciones;

-- 21 MUESTRA EL NUMERO MAYOR DE LA COLUMNA--

SELECT MAX(valorTran) FROM transacciones;

-- 22 MUESTRA EL NUMERO MENOR DE LA COLUMNA--

SELECT MIN(valorTran) FROM transacciones;

-- 23 SABER EL VALOR MEDIO--

SELECT AVG(valorTran) FROM transacciones;

-- 24 devuelve los datos con la fecha actual fecha y hora actuales--

SELECT idProductos, nombreP, NOW() AS fecha

FROM productos;

-- 25 SELECCIONA LO USUARIOS CUYAS IDENTIFICACIONES SON IMPARES

SELECT \* FROM usuarios WHERE MOD (idUsuarios,2)<>0;

-- 26 MUESTRA EL PRODUCTO CON EL PRECIO MAS BAJO

SELECT MIN(valorP) FROM productos;

-- 27 MUESTRA EL VALOR DE PRODUCTO MAS ALTO

SELECT MAX(valorP) FROM productos;

-- 28 MUESTRA LA TABLA DE TRANSACCIONES CON SU NUMERO DE TRANSACIION SU ID Y FECHA EN LA QUE SE REALIZO LA TRANSACCION

SELECT idTransacciones, NumeTran, NOW() AS fechatran

FROM Transacciones;

-- 29 MUESTRA EL ID NUMERO 2 DE LA TABLA FORMA DE PAGO

select \*

from fopago

where idFopago= 2;

-- 30 MUESTRA LOS NOMBRES, APELLIDOS,ID Y FECHA DE NACIMIENTO DE LA TABLA DATOS DE USUARIO

SELECT idDatosUsers, Nombres,Apellidos, NOW() AS FechNace

FROM datosusers;

-- 31 CUENTA EL TOTAL DE PRODUCTOS

SELECT COUNT(\*) FROM productos;

-- 32 UN RAMDON PARA LOS USUARIOS

SELECT \* FROM usuarios ORDER BY RAND();

-- 33 UNE LOS DATOS DE DOS TABLAS Y LOS DATOS DE LA SEGUNDA TABLA LOS MUESTRA A LA DERECHA

SELECT \*

FROM productos RIGHT JOIN tipopro

on productos.idProductos = tipopro.idTipoProductos;

-- 34 MUESTRA LOS VALORES DE PRODUCTOS MENORES DE 200.000

select \*

from transacciones

where valorTran < 200000;

-- 35 MUESTRA LOS ESTADO DE TRANSACCION QUE SON IGUAL A 0

select \*

from transacciones

where estaTran = 0;

-- 36 ORGANIZA LAS FECHAS DE NACIMIENTO EN LOS DATOS DE USUARIO DE MENOR A MAYOR

select \*

from datosusers

order by FechNace asc;

-- 37 UNE LOS DATOS DE DOS TABLAS Y LOS DATOS DE LA SEGUNDA TABLA LOS MUESTRA A LA DERECHA SOLO MUESTRA LOS DATOS QUE TIENEN UNA SEMEJANZA EN ID (sandra)

SELECT \*

FROM productos RIGHT JOIN fopago

on productos.idProductos = fopago.idFopago;

-- 38 SELECCIONA LOS EMAIL QUE EMPIEZAN POR LA LETRA J EN LA TABLA USUARIOS

SELECT \* FROM usuarios WHERE emailUser LIKE 'J%';

-- 39 MUESTRA LOS ESTADO DE ROLES QUE SON IGUAL A 1

select \*

from roles

where EstaRol = 1;

-- 40 CUENTA EL TOTAL DE ROLES

SELECT COUNT(\*) FROM roles;

-- 41 CUENTA EL TOTAL DE VENDEDORES

SELECT COUNT(\*) FROM vendedor;

-- 42 CUENTA EL TOTAL DE FORMAS DE PAGO

SELECT COUNT(\*) FROM fopago;

-- 43 SELECCIONA LO USUARIOS CUYAS IDENTIFICACIONES SON PARES

SELECT \* FROM roles WHERE MOD (idRoles,2)=0;

-- 44 SELECCIONA LO USUARIOS CUYAS IDENTIFICACIONES IMPARES

SELECT \* FROM roles WHERE MOD (idRoles,2)<>0;

-- 45 MUESTRA LOS VALORES DE PRODUCTOS MAYORES DE 100.000

select \*

from transacciones

where valorTran > 100000;

-- 46 MUESTRA LA CADENA DE TEXTO EN BINARIOS

select BINARY('Restaurante') ;

-- 47 SELECT ABS(-32);

SELECT ABS(-32);

-- 48 Devuelve el arco coseno de X, es decir, el valor cuyo coseno es X. Devuelve NULLsi Xno está en el rango -1de 1.

SELECT ACOS(0);

-- 49 Devuelve el arco seno de X, es decir, el valor cuyo seno es X. Devuelve NULLsi Xno está en el rango -1de 1.

SELECT ASIN(0.2);

-- 50 Añade un dia a la fecha

SELECT ADDDATE('2008-01-02',1);

-- 51 MUESTRA EL AÑOS Y EL NUMERO DE LA SEMANA

SELECT YEARWEEK('1987-01-01');

-- 52 MUESTRA LA FECHA ACTUAL

SELECT CURDATE();

-- 53 EXTRAE EL DATO ASIGNADO EN EL CASO EL AÑO DE LA FECHA

SELECT EXTRACT(YEAR FROM '2019-07-02');

-- 54 MUESTRA LA HORA

SELECT TIME('2003-12-31 01:02:03');

-- 55

SELECT TIMEDIFF ( '2000:01:01 00:00:00' , '2000:01:01 00:00:00.000001' ) ;

-- 56 MUESTRA EL NUMERO MENOR

SELECT CEILING(1.23);

-- 57 MUESTRA LOS VALORES ENTEROS MAS GRANDES

SELECT FLOOR(1.23), FLOOR(-1.23);

-- 58 MUESTRA EL LOGARITMO NATURAL SIEMPRE Y CUANDO SEA MAYOR DE CERO SINO LO ES TENDREMOS COMO RESULTADO UN NULL

SELECT LOG(2); SELECT LOG(-2);

-- 59 MUESTRA LOS GRADOS

SELECT DEGREES(PI()/3);

-- 60 MUESTRA LA HORA DEL DIA

SELECT TIMESTAMP ( '2003-12-31' ) ;

-- 61 MUESTRA LOS MESES ENTRE FECHA Y FECHA

SELECT TIMESTAMPDIFF ( MONTH , '2003-02-01' , '2003-05-01' ) ;

-- 62 MUESTRA LOS AÑOS ENTRE FECHA Y FECHA

SELECT TIMESTAMPDIFF ( YEAR , '2002-05-01' , '2001-01-01' ) ;

-- 63 MUESTRA LOS MINUTOS ENTRE FECHA Y FECHA

SELECT TIMESTAMPDIFF ( MINUTE , '2003-02-01' , '2003-05-01 12:05:55' ) ;

-- 64 Devuelve la semana calendario de la fecha como un número en el rango de 1a 53.

SELECT WEEKOFYEAR('2008-02-20');

-- 65 Devuelve el índice de días de la semana para date ( 0= lunes, 1= martes,… 6= domingo).

SELECT WEEKDAY('2008-02-03 22:23:00');

-- 66 AGREGA UN MINUTO A LA FECHA

SELECT TIMESTAMPADD(MINUTE,1,'2003-01-02');

-- 67 Dada una fecha date, devuelve un número de día (el número de días desde el año 0).

SELECT TO\_DAYS(950501);

-- 68 DEVUELVE EL TIMEPO EN SEGUNDOS

SELECT TIME\_TO\_SEC('22:23:00');

-- 69 Se usa como la DATE\_FORMAT()función, pero la formatcadena puede contener especificadores de formato solo para horas, minutos, segundos y microsegundos.

SELECT TIME\_FORMAT('100:00:00', '%H %k %h %I %l');

-- 70 Dada una fecha o fecha y hora expr, devuelve el número de segundos desde el año 0.

SELECT TO\_SECONDS(950501);

-- 71 MUESTRA LA PALABRA PRODUCTO SIN ESPACIOS

SELECT TRIM(' PRODUCTO ');

-- 72 Convertir de una zona horaria a otra

SELECT CONVERT\_TZ ( '2004-01-01 12:00:00' , 'GMT' , 'MET' ) ;

-- 73 Esta función es útil para GROUP BY consultas cuando el ONLY\_FULL\_GROUP\_BYmodo SQL está habilitado, para los casos en que MySQL rechaza una consulta que usted sabe que es válida por razones que MySQL no puede determinar.

SELECT name, address, MAX(age) FROM t GROUP BY name;

-- 74 devuelve un número entero que representa el valor numérico de la dirección en el orden de bytes de la red (big endian).

SELECT INET\_ATON('10.0.5.9');

-- 75 Dada una dirección de red IPv4 numérica en orden de bytes de red, devuelve la representación de cadena de cuatro puntos con puntos de la dirección como una cadena en el conjunto de caracteres de conexión.

SELECT INET\_NTOA(167773449);

-- 76 Para convertir una dirección IPv4 expr representada en forma numérica como un INTvalor en una dirección IPv6 representada en forma numérica como un VARBINARYvalor, use esta expresión:

SELECT HEX(INET6\_ATON(INET\_NTOA(167773449)));

-- 77 interpreta cada par de caracteres del argumento como un número hexadecimal y lo convierte al byte representado por el número.

SELECT UNHEX('4D7953514C');

-- 78 CONVIERTE LA PALABRA INGRESADA A MAYUSCULA

SELECT UPPER('cliente');

-- 79 DEVUELVE LA FECHA ACTUAL COMO UN VALOR O FORMATO

SELECT UTC\_DATE(), UTC\_DATE() + 4;

-- 80 Devuelve 1 si el argumento es una dirección IPv4 válida especificada como una cadena, 0 en caso contrario.

SELECT IS\_IPV4('10.0.5.9'), IS\_IPV4('10.0.5.256');

-- 81 Esta función toma una dirección IPv6 representada en forma numérica como una cadena binaria, como lo devuelve INET6\_ATON (). Devuelve 1 si el argumento es una dirección IPv6 compatible con IPv4 válida, 0 en caso contrario. Las direcciones compatibles con IPv4 tienen el formato :: ipv4\_address.

SELECT IS\_IPV4\_COMPAT ( INET6\_ATON ( '::10.0.5.9' ) ) ;

-- 82 Esta función toma una dirección IPv6 representada en forma numérica como una cadena binaria, como lo devuelve INET6\_ATON (). Devuelve 1 si el argumento es una dirección IPv6 asignada a IPv4 válida, 0 en caso contrario. Las direcciones asignadas a IPv4 tienen el formato :: ffff: ipv4\_address.

SELECT IS\_IPV4\_MAPPED ( INET6\_ATON ( '::10.0.5.9' ) ) ;

-- 83 El UUID está diseñado como un número único a nivel mundial en el espacio y el tiempo

SELECT UUID();

-- 84 Devuelve el valor numérico del carácter situado más a la izquierda de la cadena str.

SELECT ASCII('dx');

-- 85 Devuelve una representación de cadena del valor binario de N, donde Nes un número longlong ( BIGINT).

SELECT BIN(12);

-- 86 CUENTA LA CANTIDAD DE CARACTERES

SELECT CHAR\_LENGTH ('HOLA');

-- 87 CUENTA LA CANTIDAD DE CARACTERES

SELECT CHARACTER\_LENGTH ('HOLA');

-- 88 DEVUELVE EL JUEGO DE CARACTERES DEL ARGUMENTO DE LA CADENA

SELECT CHARSET('JUAN');

-- 89 Devuelve el arco seno de X, es decir, el valor cuyo seno es X. Devuelve NULL si X no está en el rango de -1 a 1.

SELECT ASIN(0.2);

-- 90 Devuelve el arco tangente de X, es decir, el valor cuya tangente es X.

SELECT ATAN(-2,2);

-- 91 Convierte el argumento de cadena a formato codificado en base 64 y devuelve el resultado como una cadena de caracteres con el conjunto de caracteres de conexión y la colación.

SELECT TO\_BASE64('abc'),FROM\_BASE64(TO\_BASE64('abc'));

-- 92 Calcula una suma de comprobación SHA-1 de 160 bits para la cadena, como se describe en RFC 3174 (Algoritmo hash seguro). El valor se devuelve como una cadena de 40 dígitos hexadecimales o NULL si el argumento era NULL.

SELECT SHA1('abc');

-- 93 Calcula la familia de funciones hash SHA-2 (SHA-224, SHA-256, SHA-384 y SHA-512). El primer argumento es la cadena de texto sin formato que se va a codificar. El segundo argumento indica la longitud de bits deseada del resultado, que debe tener un valor de 224, 256, 384, 512 o 0 (que equivale a 256).

SELECT SHA2('abc', 224);

-- 94 Devuelve el primer no NULLvalor de la lista, o NULLsi no hay no NULLvalores.

SELECT COALESCE(NULL,NULL,7);

-- 95 Devuelve la intercalación del argumento de cadena.

SELECT COLLATION('abc');

-- 96 Comprime una cadena y devuelve el resultado como una cadena binaria.

SELECT LENGTH(COMPRESS('b'));

-- 97 SE ENCARGA DE UNIR TODAS LAS LETRAS

SELECT CONCAT('CL', 'IEN', 'TE');

-- 98 convierte datos entre diferentes juegos de caracteres. En MySQL, los nombres de transcodificación son los mismos que los nombres del juego de caracteres correspondientes.

SELECT CONVERT('abc' USING utf8);

-- 99 Convierte números entre diferentes bases numéricas.

SELECT CONV('a',16,2);

-- 100 Extrae la parte de fecha de la expresión de fecha o fecha y hora expr.

SELECT DATE('2003-12-31 01:02:03');

-- 101 MUESTRA LA DIFERENCIA DE LOS DATOS DE FECHA

SELECT DATEDIFF('2007-12-31 23:59:59','2007-12-30');

-- 102 AÑADE UN DIA A LA FECHA

SELECT DATE\_ADD('2018-05-01',INTERVAL 1 DAY);

-- 103 DA LA FECHA SEGUN EL FORMATO DADO

SELECT DATE\_FORMAT('2009-10-04 22:23:00', '%W %M %Y');

-- 104 DECREMENTA 2 AÑOS DE LO INDICADO

SELECT DATE\_SUB('2009-10-04',INTERVAL 2 YEAR);

-- 105 DA LA DIFERENCIA ENTRE FECHAS

SELECT DATEDIFF('2007-12-31 23:59:59','2007-12-30');

-- 106 DICE EL DIA DE LA FECHA

SELECT DAY('2007-12-30');

-- 107 DICE QUE MES DEL AÑO ES

SELECT DAYOFMONTH('2007-02-03');

-- 108 DEVULVE EL INDICE DE LOS DIAS DE LA SEMANA

SELECT DAYOFWEEK('2007-02-03');

-- 109 DA LA FECHA ACTUAL

SELECT CURRENT\_DATE();

-- 110 DA LA HORA ACTUAL

SELECT CURRENT\_TIME();

-- 111 DA LA FECHA Y HORA ACTUAL

SELECT CURRENT\_TIMESTAMP();

-- 112 Devuelve el coseno de X, donde Xse expresa en radianes.

SELECT COS(PI());

-- 113 Devuelve la cotangente de X.

SELECT COT(12);

-- 114 Calcula un valor de verificación de redundancia cíclica y devuelve un valor sin signo de 32 bits. El resultado es NULL si el argumento es NULL. Se espera que el argumento sea una cadena y (si es posible) se trata como tal si no lo es.

SELECT CRC32('MySQL');

-- 115 Devuelve el argumento X, convertido de grados a radianes. (Tenga en cuenta que π radianes equivale a 180 grados).

SELECT RADIANS(90);

-- 116 Redondea el argumento Xa Dposiciones decimales.

SELECT ROUND(-1.23);

-- 117 Devuelve la tangente de X, donde Xse expresa en radianes.

SELECT TAN(PI());

-- 118 Devuelve el seno de X, donde Xse expresa en radianes.

SELECT SIN(PI());

-- 119 Devuelve la raíz cuadrada de un número no negativo X.

SELECT SQRT(4);

-- 120 Devuelve el signo del argumento como -1, 0o 1, dependiendo de si Xes negativo, cero o positivo.

SELECT SIGN(-32);

-- 121 Devuelve el valor entero más pequeño no menor que X.

SELECT CEIL(1.23);

-- 122 División entera. Los descartes de la división dan como resultado cualquier parte fraccionaria a la derecha del punto decimal.

SELECT 5 DIV 2;

-- 123 Evalúa 1si todos los operandos son distintos de cero y no NULL, 0si uno o más operandos lo son 0; de lo contrario, NULLse devuelve.

SELECT 1 AND 1;

-- 124 Sustracción:

SELECT 4-5;

-- 125 Menos unario. Este operador cambia el signo del operando.

SELECT - 3;

-- 126 Desplaza un número largo ( BIGINT) a la derecha.

SELECT 7 >> 2;

-- 127 Desplaza un número largo ( BIGINT) a la izquierda.

SELECT 1 << 2;

-- ???

SELECT BENCHMARK(1000000,AES\_ENCRYPT('roles','usuarios'));

-- 128 Devuelve la cadena en el número de índice

SELECT ELT(2, 'idProductos', 'tiempoP');

-- 129 Division:

SELECT 8/2;

-- 130 Extaer el valor de una cadena

SELECT ExtractValue('<a>usuario<b>tipoUsuarios</b></a>', '/a/b') AS ejemplo;

-- 131 Cuenta en que posicion se repite

SELECT FIELD('Bbb', 'Aaa', 'Aaa', 'Bbb', 'Bbb', 'Fff');

-- 132 cuentas a cuanto caracteres se repite el indice

SELECT FIND\_IN\_SET('b','a,b,c,d');

-- 133 Define el formato en el que se muestra un numero

SELECT FORMAT(12332.123456, 1);

-- 134 devuelve una cadena en formato de fecha

SELECT FROM\_UNIXTIME(1447430881);

-- 135 Puede usarse para medir la rapidez con que MySQL procesa la expresión.

SELECT BENCHMARK(1000000,AES\_ENCRYPT('hello','goodbye'));

-- 136 La CAST()función toma una expresión de cualquier tipo y produce un valor de resultado del tipo especificado, similar a CONVERT().

-- Para obtener más información, consulte la descripción de CONVERT()

SELECT CONVERT('abc' USING utf8);

-- 137 Restringir el número de filas que el servidor devuelve al cliente

SELECT SQL\_CALC\_FOUND\_ROWS \* FROM datosusers

WHERE idDatosUsers > 2 LIMIT 3;

SELECT FOUND\_ROWS();

-- 138 Extraer la hora en un numero entero

SELECT HOUR('12:05:03');

-- 139 Muestra si se cumple la condicion

SELECT IF(1>2,'no','si');

-- 140 Si expr1no es así NULL, IFNULL()vuelve expr1; de lo contrario, vuelve expr2.

SELECT IFNULL(2,0);

-- 141 El tipo de retorno de un CASE resultado de expresión es el tipo agregado de todos los valores de resultado.

SELECT CASE 1 WHEN 1 THEN 'one'

WHEN 2 THEN 'two' ELSE 'more' END;

-- 142 El resultado es un entero de 64 bits sin signo.

SELECT 1 ^ 1;

-- 143 Devuelve 1(verdadero) si expres igual a cualquiera de los valores de la IN()lista, de lo contrario devuelve 0(falso).

SELECT 'wefwf' IN ('wee','wefwf','weg');

-- 143 Comprueba si el candado nombrado esta libre u ocupado

select IS\_FREE\_LOCK('lock1');

-- 144 Pruebas de un valor contra un valor booleano, donde boolean\_valuepuede ser TRUE, FALSEo UNKNOWN.

SELECT 1 IS NOT UNKNOWN, 0 IS NOT UNKNOWN, NULL IS NOT UNKNOWN;

-- 145 Comprueba si un valor no lo es NULL.

SELECT 1 IS NOT NULL, 0 IS NOT NULL, NULL IS NOT NULL;

-- 146 Comprueba si un valor es NULL.

SELECT 1 IS NULL, 0 IS NULL, NULL IS NULL;

-- 147 Devuelve el trimestre del año para date, en el rango 1de 4.

SELECT QUARTER('2008-07-01');

-- 148 producir un resultado que se puede utilizar como un valor de datos

SELECT QUOTE('hola\'v!');

-- 149 Devuelve la palabra con los espacios finales eliminados.

SELECT RTRIM('usuario ');

-- 150 la primer letra mezclada con la ultima con una longitud determinada por el numero del medio

SELECT RPAD('hi',5,'?');

-- 151 Devuelve la cadena strcon todas las apariciones de la cadena from\_str reemplazada por la cadena to\_str. REPLACE()realiza

-- una coincidencia que distingue entre mayúsculas y minúsculas al buscar from\_str.

SELECT REPLACE('www.mysql.com', 'w', 'Ww');

-- 152 repite una plabara el numero de veces que se le indique

SELECT REPEAT('producto-', 3);

-- 153 Libera el candado nombrado por la cadena strque se obtuvo con GET\_LOCK(). Devuelve 1si el bloqueo se liberó,

-- 0si el bloqueo no fue establecido

select RELEASE\_LOCK('lock1');

-- 154 Cuenta las filas actualizadas

SELECT ROW\_COUNT() AS 'Contar';

-- 155 Genera un valor aleatorio, que hace que su resultado no sea determinista

Select RANDOM\_BYTES(len);

-- 156 RLIKEes sinónimo de REGEXP.

-- 157 Redondea el argumento Xa Dposiciones decimales.

SELECT ROUND(-1.23);

-- 158 Devuelve la hora UTC actual como un valor 'hh:mm:ss'o hhmmssformato, dependiendo de si la función se utiliza en un contexto de cadena o numérico.

SELECT UTC\_TIME(), UTC\_TIME() + 0;

-- 159 Devuelve la fecha y hora UTC actual como un valor o formato, dependiendo de si la función se utiliza en un contexto numérico o de cadena. 'YYYY-MM-DD hh:mm:ss'YYYYMMDDhhmmss

select \* from productos limit 4;

select \* from productos where idTipoProductos in (4,5) order by nombreP, idTipoProductos;

**join**

select concat(u.emailUser,'--- ',u.EstaUser) as usuarios, r.EstaRol as roles, du.Telefonos as datosusers

from usuarios u

join datosusers du

on (du.idUsuarios= u.idUsuarios)

join roles r

on (r.idUsuarios= du.idUsuarios);

select \* from datosusers;

select concat (t.NumeTran,'---- ',t.fechaTran) as transacciones, tp.NombTipo as tipotran, v.idVendedor as vendedor, u.emailUser as usuarios

from transacciones t

join vendedor v

on (v.idVendedor= t.idVendedor)

join tipotran tp

on(tp.idTipoTran= t.idTipoTran)

join usuarios u

on(u.idUsuarios=t.idUsuarios)

order by t.fechaTran;

SELECT \* FROM tipopro INNER JOIN productos WHERE tipopro.idTipoProductos = productos.idProductos;

select \* from tipousers