Trabajo Practico integrador

NOMBRE DEL TP

INTEGRANTES: CALABRESE, NEMI, MIRANDA, PEREZ, RUIZ

PROFESORA: VERONICA PINEYRO

MATERIA: BASE DE DATOS

FECHA DE ENTREGA: 16/07/2021

Enunciado:

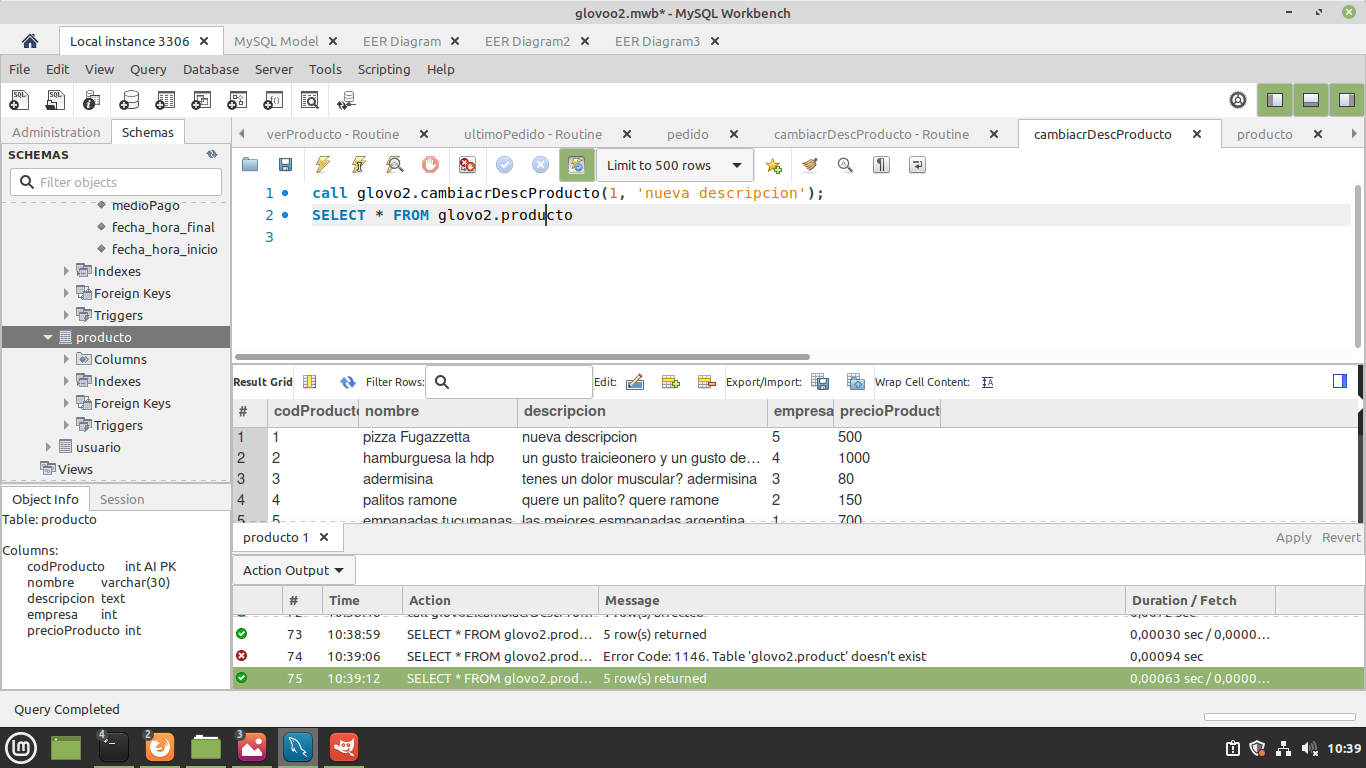
GloBo es una novedosa APP (aplicación para teléfonos celulares) que promete llevar cualquier cosa pequeña (aprox. 40x40cm) en menos de 60 minutos al lugar que vos indiques. La App GloBo, funciona con el formato “multi delivery on Demand” y basa su negocio en la economía colaborativa. El consumo colaborativo o economía colaborativa se define como una interacción entre dos o más sujetos, a través de medios digitalizados o no, que satisface una necesidad (no necesariamente real), a una o más personas. Las plataformas digitales establecen un marco, donde los usuarios pueden interactuar entre ellos y/o con la misma plataforma. Los usuarios seleccionan el rol que desean en cada momento,o varios roles simultáneamente (por ejemplo: vendedor y comprador) es un sistema abierto y dinámico. Normalmente, existe un sistema de evaluación entre usuarios, mediante el cual, adquieren una reputación, y con ella, la confianza necesaria para seguir llevando a cabo la actividad que deseen. Cuanto mayor sea el número de usuarios, que exista en la plataforma digital, más valor tendrá la misma, los usuarios tendrán más posibilidades de elección y/o desarrollo, serán mejor evaluados, y la confianza estará más contrastada. El movimiento del consumo colaborativo supone un cambio cultural y económico en los hábitos de consumo marcado por la migración de un escenario de consumismo individualizado hacia nuevos modelos potenciados por los medios sociales y las plataformas de tipo peer-to-peer(red-entre-pares o red-entre-iguales).En los servicios de consumo colaborativo las barreras de desconfianza se ven minimizadas gracias al uso de perfiles de usuarios con valoraciones y referencias añadidas por otros usuarios, lo que da origen a nuevas maneras de relacionarse, intercambiar, y monetizar habilidades y/o bienes económicos, lo que era impensable hace unos años. Tal como dicen desde la empresa,“en GloBo compramos, recogemos y enviamos cualquier cosa que necesites de tu ciudad. Úsanos para hacer envíos exprés (documentos, paquetes, llaves, etc), pedir cualquier cosa (farmacia, comida, electrónica, flores, etc) o hacer tus gestiones... ¡lo que se te ocurra!¡Y lo hacemos en minutos!”

Store procedure:

-realizarPedido: inserta una fila en la tabla pedido

|  |  |
| --- | --- |
| CREATE DEFINER=`root`@`localhost` PROCEDURE `realizarPedido` (in codPedido int , in codGlober int, codigoUsuario int, in direc varchar(30), in fecha date, in estadopedido varchar(40), estadoglober varchar(40), in mediopago varchar(45), fechainicio varchar(45) ) | |
| BEGIN |
| insert into pedido values (codPedido, codglober,codigoUsuario, direc, fecha, estadopedido, estadoglober, mediopago, fechainicio); | |

END



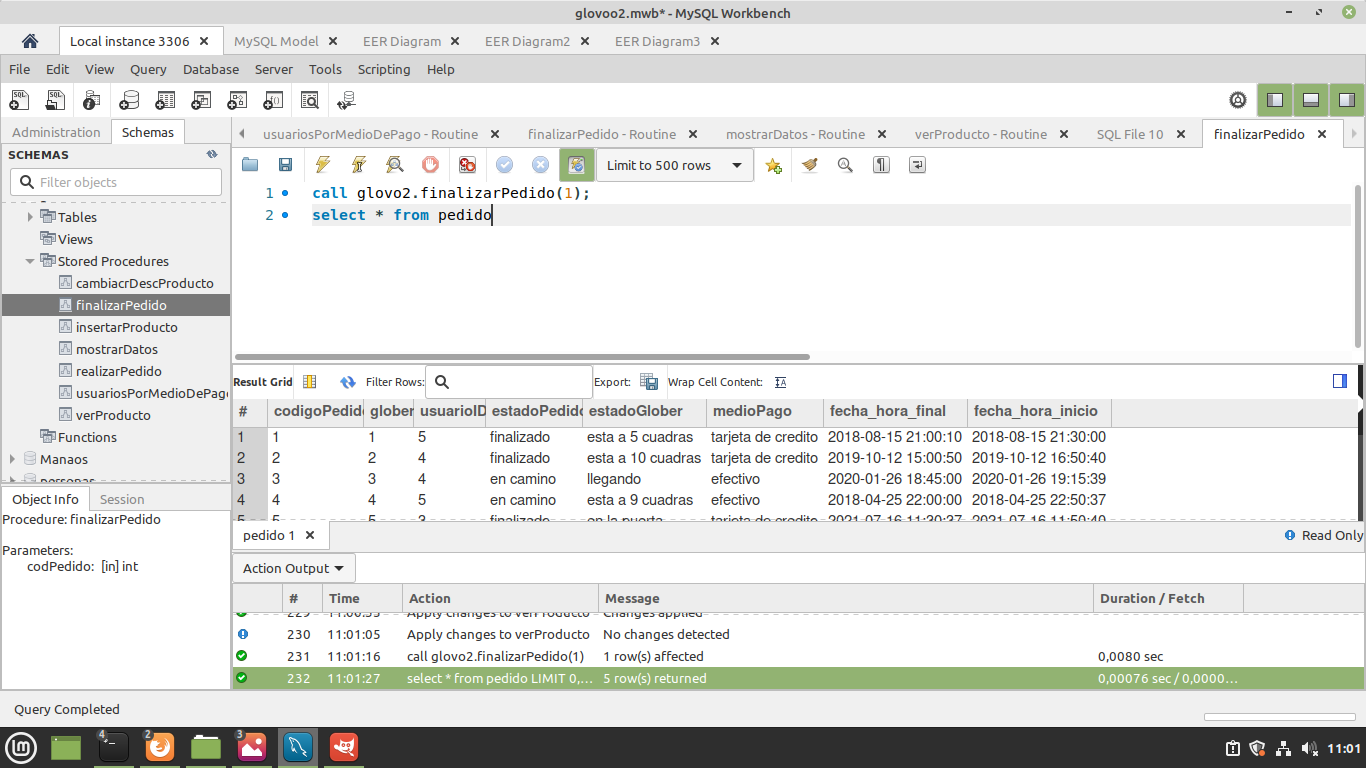
-insertarProducto: inserta una fila en la tabla producto

CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarProducto`(in codProducto int , in nombre varchar(30),in descripcion text,in empresa int, in precioProducto int )

BEGIN

insert into produo values (codProducto, nombre, descripcion, empresa, precioProducto);

END



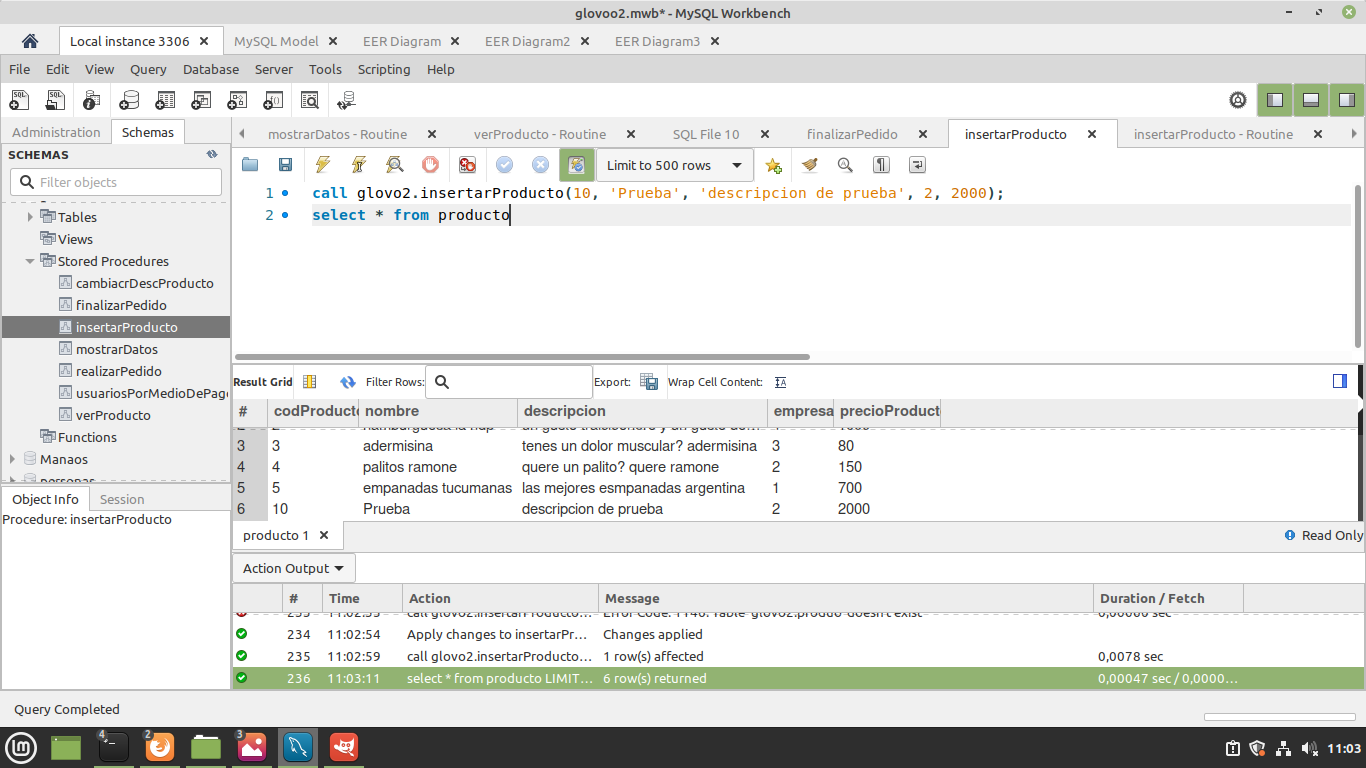
-cambiarDescProducto: de aquellos productos que no tengan descripcion les cambia esta por el precio del producto

CREATE DEFINER=`root`@`localhost` PROCEDURE `cambiacrDescProducto`(in codigo INT, in ponerDescripcion TEXT)

BEGIN

update producto set descripcion = ponerDescripcion where codProducto = codigo;

END



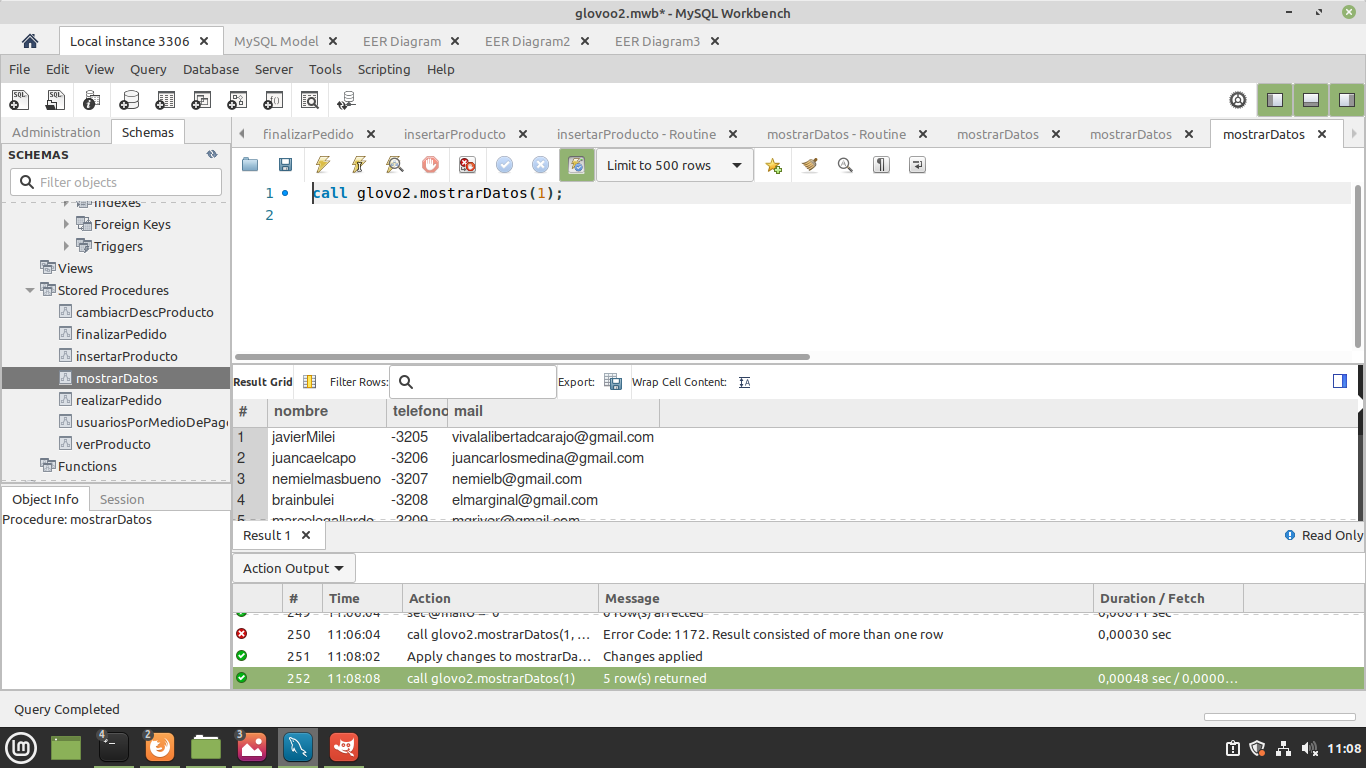
-usuariosPorMediosdepago: devuelve la cantidad de usuarios que utilizan un medio de pago

CREATE DEFINER=`root`@`localhost` PROCEDURE `usuariosPorMedioDePago`(in mediopago varchar(15), out cantidad int )

BEGIN

SELECT count(usuarioId) into cantidad from pedido where medioPagp=mediopago;

END



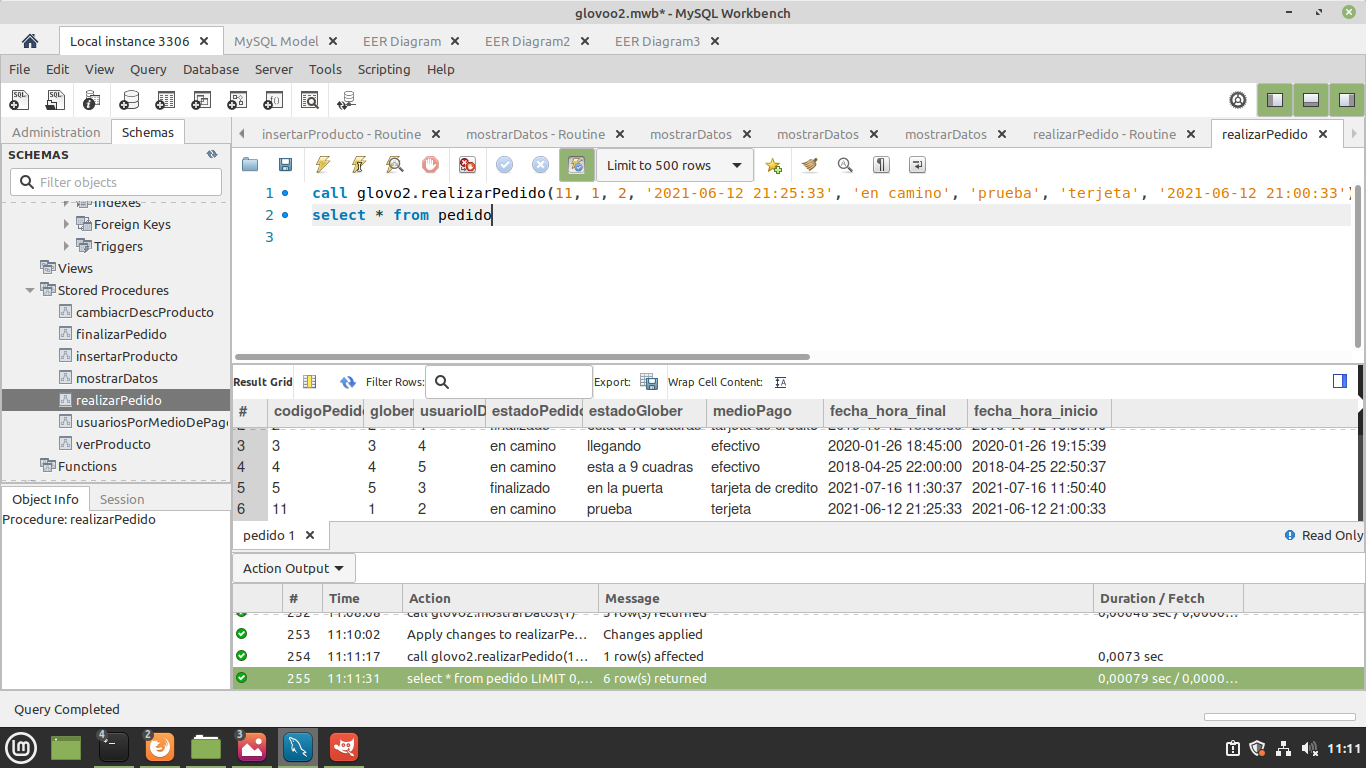
-finalizarPedido: cambia el estado del pedido a finalizado

CREATE DEFINER=`root`@`localhost` PROCEDURE `finalizarPedido`(in codPedido int)

BEGIN

update pedido set estadoPedido="finalizado" where codPedido=codigoPedido;

END



-MostrarDatos: recibe un codigo de usuario y devuelve su nombre,telefono y mail

CREATE DEFINER=`root`@`localhost` PROCEDURE `mostrarDatos`(in idUsuario int, out nombreUsuario varchar(40), out telefonoU int, mailU varchar(30))

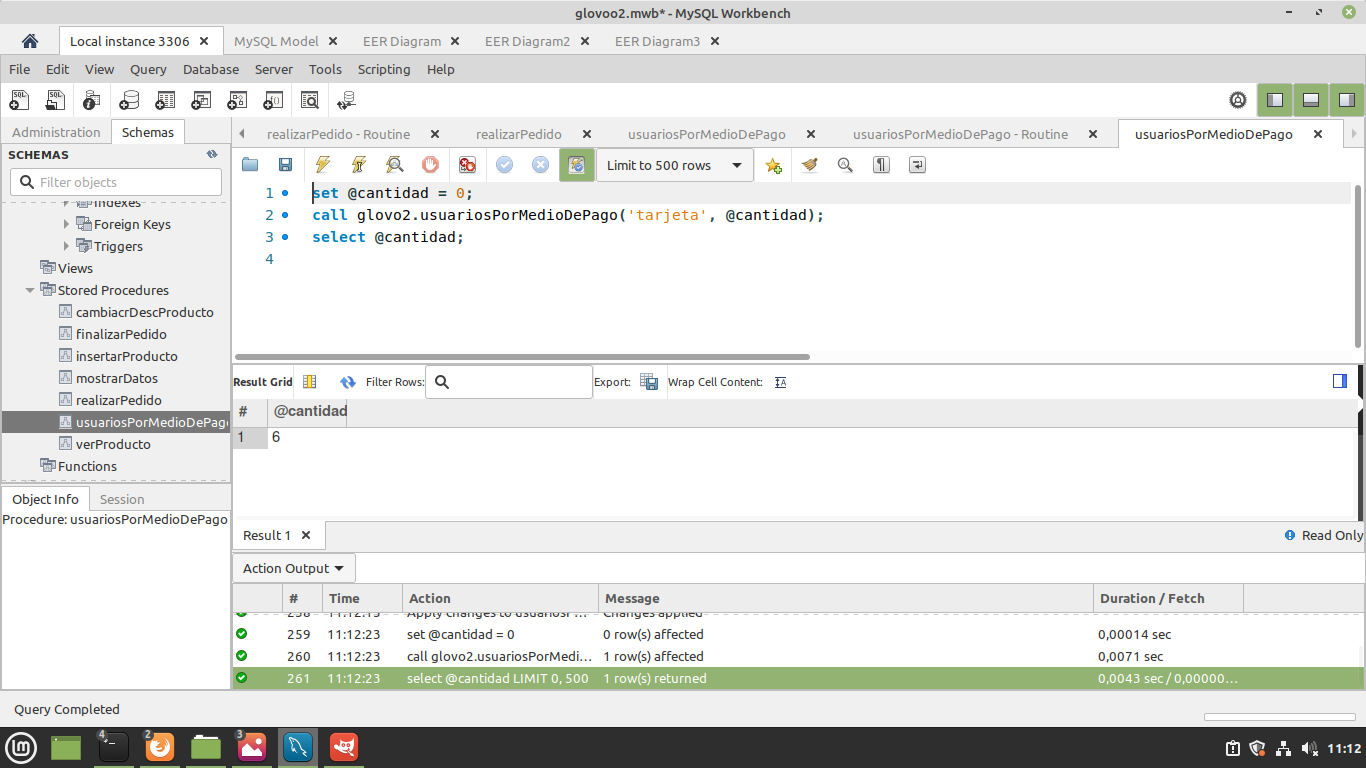
BEGIN

select nombre into nombreUsuario from usuario where idUsuario=idusuario;

select telefono into telefonoU from usuario where idUsuario=idusuario;

select mail into mailU from usuario where idUsuario=idusuario;

END



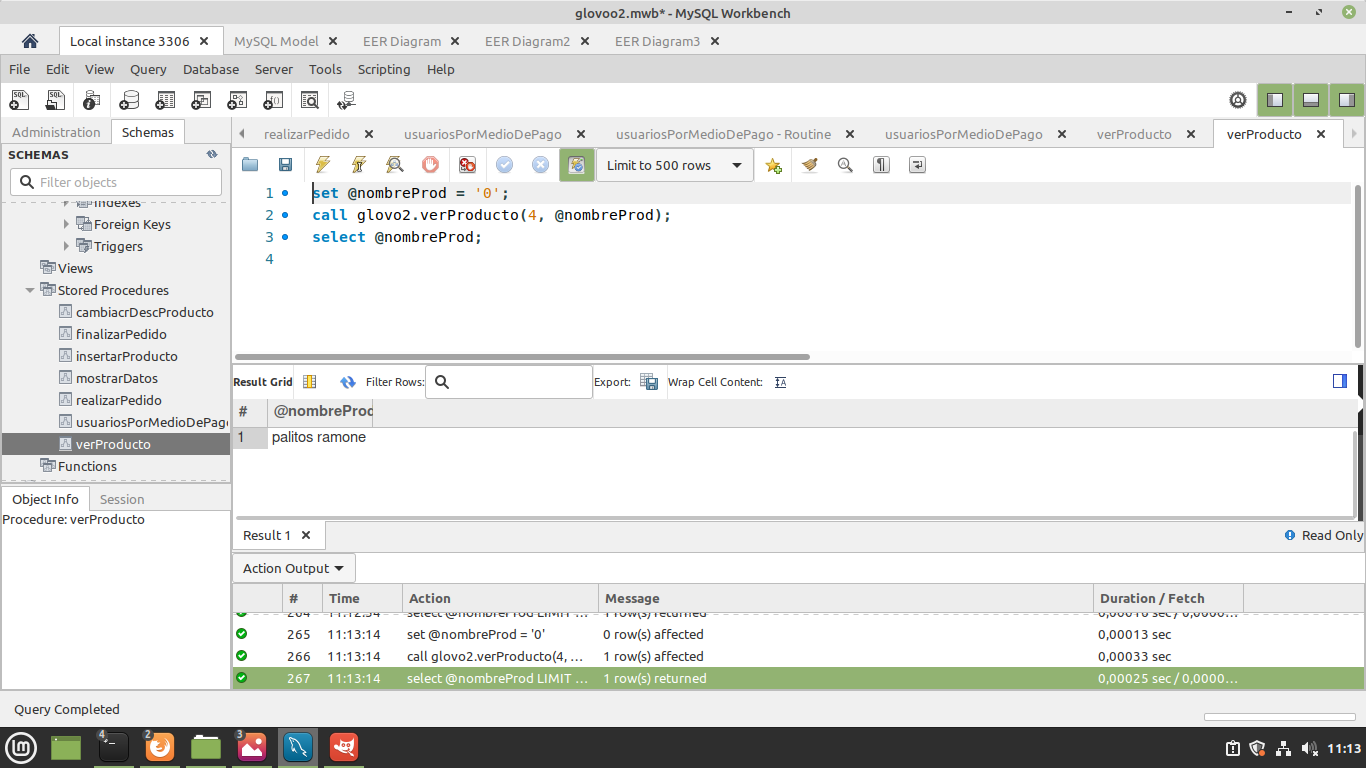
-verProductos: recibe un codigo de pedido y devuelve los nombres de los productos comprados.

CREATE DEFINER=`root`@`localhost` PROCEDURE `verProducto`(in codPedido int, out nombreProd varchar(30))

BEGIN

select nombre into nombreProd from producto join detalle\_pedido on detalle\_pedido.producto=codProducto where codPedido=pedido;

END



|  |
| --- |
|  |
| -- -------------------- TRIGGERS -------------------------------- |

Horasacumuladas: calcula las horas acumuladas

CREATE DEFINER = CURRENT\_USER TRIGGER `glovo2`.`pedido\_AFTER\_UPDATE` AFTER UPDATE ON `pedido` FOR EACH ROW

BEGIN

declare minutos time;

if(new.estadoPedido = "finalizado") then

set minutos = timestampdiff(minute, old.fecha\_hora\_inicio, old.fecha\_hora\_final);

end if;

update usuario set HorasAcumuladas = HorasAcumuladas + minutos;

END

Precio\_pedido: Calcula precio de un pedido

CREATE DEFINER=`root`@`localhost` TRIGGER `detalle\_pedido\_AFTER\_UPDATE` AFTER UPDATE ON `detalle\_pedido` FOR EACH ROW BEGIN

declare precio float default 0;

select precioProducto into precio from producto where new.producto = codProducto;

if(new.cantidad != old.cantidad) then

update detalle\_pedido set precio\_producto = cantidad \* precio;

end if;

END

-- ----------------- STORE FUNCTIONS ----------------------------

cantidadDeVentas: recibe un codigo de empresa y devulve la cantidad total de ventas que realizó

CREATE DEFINER=`root`@`localhost` FUNCTION `cantidadVentas` (`idempresa` INT) RETURNS INT(11) NO SQL

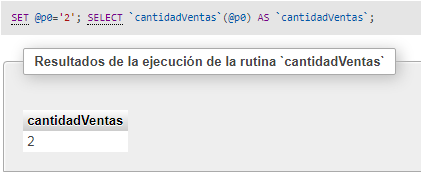
BEGIN

DECLARE cantidad int;

SELECT count(\*) into cantidad from pedido join detalle\_pedido on pedido = codigoPedido join producto on producto = codProducto where empresa = idempresa;

RETURN cantidad;

END



estadoDelGlober: recibe un idUsuario y devuelve 0 si esta entregando un pedido y 1 si está libre

CREATE DEFINER=`root`@`localhost` FUNCTION `estadoGlober` (`usuarioid` INT) RETURNS INT(11) NO SQL

BEGIN

DECLARE res int;

declare cantidad int default 0;

select count(\*) into cantidad from pedido where glober = usuarioid and estadoPedido != "finalizado";

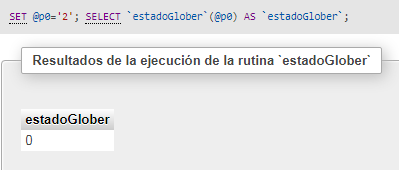
if cantidad = 0 THEN

SET res = 1;

ELSE SET res = 0;

end if;

RETURN res;



ImporteEmpresa: recibe una idEmpresa y devuelve el importe que tiene que pagar del mes actual

CREATE DEFINER=`root`@`localhost` FUNCTION `importeEmpresa` (`idEmpresa` INT) RETURNS INT(11) NO SQL

BEGIN

DECLARE importe int;

DECLARE porcentaje int;

select sum(precio\_producto \* cantidad) into importe

from detalle\_pedido join producto on producto = codProducto

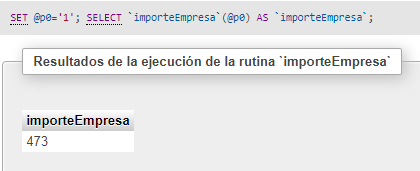
where empresa = idEmpresa;

select comision into porcentaje from empresa where codEmpresa = idEmpresa;

SET importe = importe + (porcentaje \* importe) / 100;

RETURN importe;

END



SueldoGlober: recibe un año y un idusuario y devuelve la cantidad de plata ganada por el glober en ese año.

CREATE DEFINER=`root`@`localhost` FUNCTION `sueldoGlober` (`anio` INT, `usuario` INT) RETURNS INT(11) NO SQL

BEGIN

declare sueldo int;

declare horasTrabajadas int;

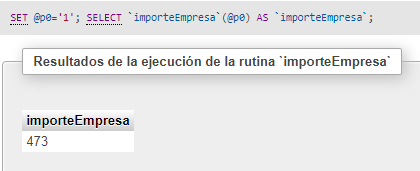
select cast(HorasAcumuladas AS int) into horasTrabajadas from usuario where idusuario = usuario;

SET horasTrabajadas = horasTrabajadas / 10000;

SET sueldo = horasTrabajadas \* 30;

return sueldo;

END



UltimoPedido: recibe un numero de usuario y devuelve la fecha de su último pedido.

CREATE DEFINER=`root`@`localhost` FUNCTION `ultimoPedido` (`usuario` INT) RETURNS DATE NO SQL

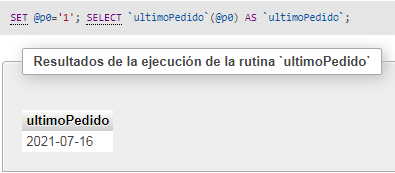
BEGIN

DECLARE ultima\_fecha DATE;

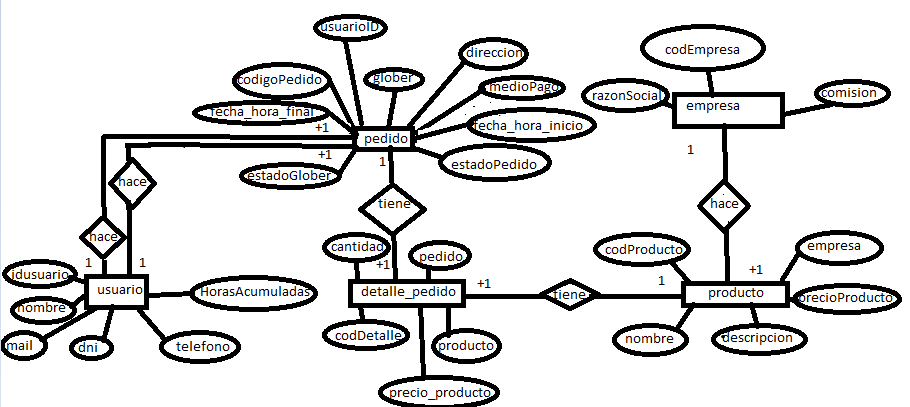
select max(fecha\_hora\_final) into ultima\_fecha from pedido where usuarioId = usuario;

RETURN ultima\_fecha;

END



DER:



Código .sql:

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema glovo2

-- -----------------------------------------------------

DROP SCHEMA IF EXISTS `glovo2` ;

-- -----------------------------------------------------

-- Schema glovo2

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `glovo2` ;

USE `glovo2` ;

-- -----------------------------------------------------

-- Table `glovo2`.`usuario`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `glovo2`.`usuario` ;

CREATE TABLE IF NOT EXISTS `glovo2`.`usuario` (

`idusuario` INT NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(40) NOT NULL,

`dni` INT NOT NULL,

`telefono` VARCHAR(20) NOT NULL,

`mail` VARCHAR(30) NOT NULL,

`HorasAcumuladas` TIME NOT NULL,

PRIMARY KEY (`idusuario`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8

COLLATE = utf8\_spanish\_ci;

-- -----------------------------------------------------

-- Table `glovo2`.`pedido`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `glovo2`.`pedido` ;

CREATE TABLE IF NOT EXISTS `glovo2`.`pedido` (

`codigoPedido` INT NOT NULL AUTO\_INCREMENT,

`glober` INT NOT NULL,

`usuarioID` INT NOT NULL,

`estadoPedido` VARCHAR(40) NOT NULL,

`estadoGlober` VARCHAR(40) NOT NULL,

`medioPago` VARCHAR(45) NOT NULL,

`fecha\_hora\_final` TIMESTAMP NOT NULL,

`fecha\_hora\_inicio` TIMESTAMP NOT NULL,

PRIMARY KEY (`codigoPedido`),

INDEX `glober` (`glober` ASC) VISIBLE,

INDEX `fk\_pedido\_usuario1\_idx` (`usuarioID` ASC),

CONSTRAINT `fk\_pedido\_usuario1`

FOREIGN KEY (`usuarioID`)

REFERENCES `glovo2`.`usuario` (`idusuario`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `pedido\_ibfk\_3`

FOREIGN KEY (`glober`)

REFERENCES `glovo2`.`usuario` (`idusuario`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8

COLLATE = utf8\_spanish\_ci;

-- -----------------------------------------------------

-- Table `glovo2`.`empresa`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `glovo2`.`empresa` ;

CREATE TABLE IF NOT EXISTS `glovo2`.`empresa` (

`codEmpresa` INT NOT NULL,

`razonSocial` VARCHAR(40) NOT NULL,

`comision` INT NOT NULL,

PRIMARY KEY (`codEmpresa`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8

COLLATE = utf8\_spanish\_ci;

-- -----------------------------------------------------

-- Table `glovo2`.`producto`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `glovo2`.`producto` ;

CREATE TABLE IF NOT EXISTS `glovo2`.`producto` (

`codProducto` INT NOT NULL AUTO\_INCREMENT,

`nombre` VARCHAR(30) NOT NULL,

`descripcion` TEXT NOT NULL,

`empresa` INT NOT NULL,

`precioProducto` INT NOT NULL,

PRIMARY KEY (`codProducto`),

INDEX `empresa` (`empresa` ASC) ,

CONSTRAINT `producto\_ibfk\_1`

FOREIGN KEY (`empresa`)

REFERENCES `glovo2`.`empresa` (`codEmpresa`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8

COLLATE = utf8\_spanish\_ci;

-- -----------------------------------------------------

-- Table `glovo2`.`detalle\_pedido`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `glovo2`.`detalle\_pedido` ;

CREATE TABLE IF NOT EXISTS `glovo2`.`detalle\_pedido` (

`codDetalle` INT NOT NULL,

`producto` INT NOT NULL,

`pedido` INT NOT NULL,

`cantidad` INT NOT NULL,

`precio\_producto` FLOAT NOT NULL,

INDEX `producto` (`producto` ASC) ,

INDEX `pedido` (`pedido` ASC) ,

CONSTRAINT `detalle\_pedido\_ibfk\_1`

FOREIGN KEY (`pedido`)

REFERENCES `glovo2`.`pedido` (`codigoPedido`),

CONSTRAINT `detalle\_pedido\_ibfk\_2`

FOREIGN KEY (`producto`)

REFERENCES `glovo2`.`producto` (`codProducto`))

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8

COLLATE = utf8\_spanish\_ci;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

-- -------------------- INSERTS ---------------------------------

insert into empresa values ( 1, "Hermanos Gómez y Ripold restaurante", 5);

insert into empresa values ( 2, "open 24 hrs!", 7);

insert into empresa values ( 3, "farmacia leopolda", 10);

insert into empresa values ( 4, "hamburgueseria los tito", 5);

insert into empresa values ( 5, "pizzas mirandas", 10);

-- ----------------------------------------------------------------------

insert into producto values ( 1, "pizza Fugazzetta", "la increible pizza Fugazzeta", 5, 500);

insert into producto values ( 2, "hamburguesa la hdp", "un gusto traicieonero y un gusto descomunal", 4, 1000);

insert into producto values ( 3, "adermisina", "tenes un dolor muscular? adermisina", 3, 80);

insert into producto values ( 4, "palitos ramone", "quere un palito? quere ramone", 2, 150);

insert into producto values ( 5, "empanadas tucumanas", "las mejores esmpanadas argentina", 1, 700);

-- -----------------------------------------------------------------------

insert into usuario values ( 1, "javierMilei", 12345, 4576-7781, "vivalalibertadcarajo@gmail.com", "8:00:13");

insert into usuario values ( 2, "juancaelcapo", 54321, 4576-7782, "juancarlosmedina@gmail.com", "5:00:45");

insert into usuario values ( 3, "nemielmasbueno", 67890, 4576-7783, "nemielb@gmail.com", "6:00:20");

insert into usuario values ( 4, "brainbulei", 09876, 4576-7784, "elmarginal@gmail.com", "15:00:45");

insert into usuario values ( 5, "marcelogallardo", 10298, 4576-7785, "mgriver@gmail.com", "17:00:67");

-- ------------------------------------------------------------------------

insert into pedido values ( 1, 1, 5, "en camino", "esta a 5 cuadras", "tarjeta de credito", "2018-08-15 21:00:10", "2018-08-15 21:30:00");

insert into pedido values ( 2, 2, 4, "finalizado", "esta a 10 cuadras", "tarjeta de credito", "2019-10-12 15:00:50", "2019-10-12 16:50:40");

insert into pedido values ( 3, 3, 4, "en camino", "llegando", "efectivo", "2020-01-26 18:45:00", "2020-01-26 19:15:39");

insert into pedido values ( 4, 4, 5, "en camino", "esta a 9 cuadras", "efectivo", "2018-04-25 22:00:00", "2018-04-25 22:50:37");

insert into pedido values ( 5, 5, 3, "finalizado", "en la puerta", "tarjeta de credito", "2021-07-16 11:30:37", "2021-07-16 11:50:40");

-- ------------------------------------------------------------------------

insert into detalle\_pedido values ( 1, 5, 5, 2, 1400);

insert into detalle\_pedido values ( 2, 4, 4, 1, 150);

insert into detalle\_pedido values ( 3, 3, 3, 3, 240);

insert into detalle\_pedido values ( 4, 2, 2, 2, 2000);

insert into detalle\_pedido values ( 5, 1, 1, 4, 2000);

-- -------------------- STORE PROCEDURE -------------------------

CREATE DEFINER=`root`@`localhost` PROCEDURE `realizarPedido` (in codPedido int , in codGlober int, codigoUsuario int, in direc varchar(30), in fecha date, in estadopedido varchar(40), estadoglober varchar(40), in mediopago varchar(45), fechainicio varchar(45) )

BEGIN

insert into pedido values (codPedido, codglober,codigoUsuario, direc, fecha, estadopedido, estadoglober, mediopago, fechainicio);

END

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` PROCEDURE `insertarProducto`(in codProducto int , in nombre varchar(30),in descripcion text,in empresa int, in precioProducto int )

BEGIN

insert into produo values (codProducto, nombre, descripcion, empresa, precioProducto);

END

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` PROCEDURE `cambiacrDescProducto`(in codigo INT, in ponerDescripcion TEXT)

BEGIN

update producto set descripcion = ponerDescripcion where codProducto = codigo;

END

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` PROCEDURE `usuariosPorMedioDePago`(in mediopago varchar(15), out cantidad int )

BEGIN

SELECT count(usuarioId) into cantidad from pedido where medioPagp=mediopago;

END

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` PROCEDURE `finalizarPedido`(in codPedido int)

BEGIN

update pedido set estadoPedido="finalizado" where codPedido=codigoPedido;

END

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` PROCEDURE `mostrarDatos`(in idUsuario int, out nombreUsuario varchar(40), out telefonoU int, mailU varchar(30))

BEGIN

select nombre into nombreUsuario from usuario where idUsuario=idusuario;

select telefono into telefonoU from usuario where idUsuario=idusuario;

select mail into mailU from usuario where idUsuario=idusuario;

END

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` PROCEDURE `verProducto`(in codPedido int, out nombreProd varchar(30))

BEGIN

select nombre into nombreProd from producto join detalle\_pedido on detalle\_pedido.producto=codProducto where codPedido=pedido;

END

-- -------------------- TRIGGERS --------------------------------

CREATE DEFINER=`root`@`localhost` TRIGGER `pedido\_AFTER\_UPDATE` AFTER UPDATE ON `pedido` FOR EACH ROW BEGIN

if(new.estadoPedido = "producto obtenido" and new.fecha\_hora\_inicio > "2021-06-23 00:00:00" and new.fecha\_hora\_inicio < "2021-10-10 00:00:00")then

update detalle\_pedido set precio\_producto = 2000.0 where pedido = new.codigoPedido;

end if;

END

------------------------------------------------------------------

CREATE DEFINER = CURRENT\_USER TRIGGER `glovo2`.`pedido\_AFTER\_UPDATE` AFTER UPDATE ON `pedido` FOR EACH ROW

BEGIN

declare minutos time;

if(new.estadoPedido = "finalizado") then

set minutos = timestampdiff(minute, old.fecha\_hora\_inicio, old.fecha\_hora\_final);

end if;

update usuario set HorasAcumuladas = HorasAcumuladas + minutos;

END

---------------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` TRIGGER `detalle\_pedido\_AFTER\_UPDATE` AFTER UPDATE ON `detalle\_pedido` FOR EACH ROW BEGIN

declare precio float default 0;

select precioProducto into precio from producto where new.producto = codProducto;

if(new.cantidad != old.cantidad) then

update detalle\_pedido set precio\_producto = cantidad \* precio;

end if;

END

-----------------------------------------------------------------------

-- ----------------- STORE FUNCTIONS ----------------------------

CREATE DEFINER=`root`@`localhost` FUNCTION `cantidadVentas` (`idempresa` INT) RETURNS INT(11) NO SQL

BEGIN

DECLARE cantidad int;

SELECT count(\*) into cantidad from pedido join detalle\_pedido on pedido = codigoPedido join producto on producto = codProducto where empresa = idempresa;

RETURN cantidad;

END

-- ---------------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` FUNCTION `estadoGlober` (`usuarioid` INT) RETURNS INT(11) NO SQL

BEGIN

DECLARE res int;

declare cantidad int default 0;

select count(\*) into cantidad from pedido where glober = usuarioid and estadoPedido != "finalizado";

if cantidad = 0 THEN

SET res = 1;

ELSE SET res = 0;

end if;

RETURN res;

-- ---------------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` FUNCTION `importeEmpresa` (`idEmpresa` INT) RETURNS INT(11) NO SQL

BEGIN

DECLARE importe int;

DECLARE porcentaje int;

select sum(precio\_producto \* cantidad) into importe

from detalle\_pedido join producto on producto = codProducto

where empresa = idEmpresa;

select comision into porcentaje from empresa where codEmpresa = idEmpresa;

SET importe = importe + (porcentaje \* importe) / 100;

RETURN importe;

END

-- ---------------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` FUNCTION `sueldoGlober` (`anio` INT, `usuario` INT) RETURNS INT(11) NO SQL

BEGIN

declare sueldo int;

declare horasTrabajadas int;

select cast(HorasAcumuladas AS int) into horasTrabajadas from usuario where idusuario = usuario;

SET horasTrabajadas = horasTrabajadas / 10000;

SET sueldo = horasTrabajadas \* 30;

return sueldo;

END

-- ---------------------------------------------------------------------

CREATE DEFINER=`root`@`localhost` FUNCTION `ultimoPedido` (`usuario` INT) RETURNS DATE NO SQL

BEGIN

DECLARE ultima\_fecha DATE;

select max(fecha\_hora\_final) into ultima\_fecha from pedido where usuarioId = usuario;

RETURN ultima\_fecha;

END