***Here I am introducing a little bit of MySQL work***

***Trigger***

***Store procedure***

***View***

***DDL***

***DML***

***CALCULATE FUNCTION***

*###store procedure*

*USE `GET THE HANDS DIRTY `;*

*SELECT \* FROM customers*

*drop procedure if exists uspGetAddress;*

*DELIMITER //*

*CREATE PROCEDURE uspGetAddress()*

*BEGIN*

*SELECT \* from customers;*

*END //*

*Call uspGetAddress()*

*drop procedure if exists GetPostCode;*

*DELIMITER //*

*CREATE PROCEDURE GetPostCode(IN Postcode VARCHAR(50))*

*BEGIN*

*SELECT \**

*FROM customers*

*WHERE addressLine1 = Postcode;*

*END //*

*DELIMITER ;*

*Call GetPostCode('54, rue Royale');*

*DROP PROCEDURE IF EXISTS GetPostCode;*

*###### /\*VIEW\*/*

*create view Myfristview as*

*select \* from employees*

*select \* from Myfristview*

*/\* Trigger \*/*

*create table employee\_audit(*

*employeeNumber int,*

*lastname varchar(255),*

*changedon datetime,*

*action varchar (255)*

*);*

*USE `handdirty`;*

*DELIMITER $$*

*CREATE TRIGGER before\_employee\_update*

*BEFORE UPDATE ON employees*

*FOR EACH ROW BEGIN*

*INSERT INTO employee\_audit*

*SET action = 'update',*

*employeeNumber = OLD.employeeNumber,*

*lastname = OLD.lastname, -- Corrected column name from 'lasname' to 'lastname'*

*changedon = NOW();*

*END$$*

*DELIMITER ;*

*update employees*

*set lastname = 'nichole kidman'*

*where employeeNumber= 1056*

*select \* from employees*

*select \* from employee\_audit*

*/\*calculation\*/*

*SELECT COUNT(\*) FROM payments*

*SELECT(*

*SELECT COUNT(\*)*

*FROM payments*

*) AS tOTAL\_amount,*

*(SELECT COUNT(\*)*

*FROM Customers*

*) AS No\_Of\_Customers*

*FROM dual*

*################################################*

*use dreamhome*

*create temporary table tempstaff as select \* from staff;*

*select \* from tempstaff;*

*create temporary table results(*

*maxsalary int,*

*Noofpositions int*

*);*

*select \* from results;*

*insert into results*

*select max(salary) maxsalary, count(position) as Noofpositions*

*from staff where sex= 'F';*

*drop temporary table staffdata;*

*create temporary table staffdata(*

*Sno varchar(5),*

*salary int,*

*name varchar (25),*

*tel\_no varchar (25),*

*bno varchar(5),*

*street varchar(60)*

*);*

*select \* from staffdata;*

*insert into staffdata*

*select s.sno, s.salary, concat(s.Fname, ' ', s.Lname) as name, b.tel\_no, b.bno, b.street*

*from staff s*

*inner join branch b*

*on s.bno= b.bno*

*###############################################################*

*Create Database DataMart; /\* datamart\*/*

*Use Dreamhome;*

*Create VIEW View1 AS*

*SELECT \* FROM STAFF;*

*Create VIEW View2 AS*

*SELECT \* FROM Branch;*

*Use Dreamhome;*

*Create VIEW View3 AS*

*SELECT \* FROM Owner;*

*######We then connect to our own business database*

*USE DataMart;*

*# Next we create a table within our new Data Mart database of each SQL query we have developed (eg: Query1, 2 AND 3)*

*# To answer Question 1*

*Create Table DataMart.Query1\_Data SELECT \* FROM Dreamhome.view1;*

*# To answer Question 2*

*Create Table DataMart.Query2\_Data SELECT \* FROM Dreamhome.view2;*

*# To answer Question 3*

*Create Table DataMart.Query3\_Data SELECT \* FROM Dreamhome.view3;*

*# Add in here your testing scripts to show me the result of Query 1, 2 AND 3*

*Use Datamart;*

*SELECT \* FROM Query1\_Data;*

*SELECT \* FROM Query2\_Data;*

*SELECT \* FROM Query3\_Data;*

CREATE TABLE `customers` (

`customerNumber` int(11) NOT NULL,

`customerName` varchar(50) NOT NULL,

`contactLastName` varchar(50) NOT NULL,

`contactFirstName` varchar(50) NOT NULL,

`phone` varchar(50) NOT NULL,

`addressLine1` varchar(50) NOT NULL,

`addressLine2` varchar(50) DEFAULT NULL,

`city` varchar(50) NOT NULL,

`state` varchar(50) DEFAULT NULL,

`postalCode` varchar(15) DEFAULT NULL,

`country` varchar(50) NOT NULL,

`salesRepEmployeeNumber` int(11) DEFAULT NULL,

`creditLimit` double DEFAULT NULL,

PRIMARY KEY (`customerNumber`),

KEY `salesRepEmployeeNumber` (`salesRepEmployeeNumber`),

CONSTRAINT `customers\_ibfk\_1` FOREIGN KEY (`salesRepEmployeeNumber`) REFERENCES `employees` (`employeeNumber`)

);

CREATE TABLE `employees` (

`employeeNumber` int(11) NOT NULL,

`lastName` varchar(50) NOT NULL,

`firstName` varchar(50) NOT NULL,

`extension` varchar(10) NOT NULL,

`email` varchar(100) NOT NULL,

`officeCode` varchar(10) NOT NULL,

`reportsTo` int(11) DEFAULT NULL,

`jobTitle` varchar(50) NOT NULL,

PRIMARY KEY (`employeeNumber`),

KEY `reportsTo` (`reportsTo`),

KEY `officeCode` (`officeCode`),

CONSTRAINT `employees\_ibfk\_2` FOREIGN KEY (`officeCode`) REFERENCES `offices` (`officeCode`),

CONSTRAINT `employees\_ibfk\_1` FOREIGN KEY (`reportsTo`) REFERENCES `employees` (`employeeNumber`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `offices` (

`officeCode` varchar(10) NOT NULL,

`city` varchar(50) NOT NULL,

`phone` varchar(50) NOT NULL,

`addressLine1` varchar(50) NOT NULL,

`addressLine2` varchar(50) DEFAULT NULL,

`state` varchar(50) DEFAULT NULL,

`country` varchar(50) NOT NULL,

`postalCode` varchar(15) NOT NULL,

`territory` varchar(10) NOT NULL,

PRIMARY KEY (`officeCode`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `orderdetails` (

`orderNumber` int(11) NOT NULL,

`productCode` varchar(15) NOT NULL,

`quantityOrdered` int(11) NOT NULL,

`priceEach` double NOT NULL,

`orderLineNumber` smallint(6) NOT NULL,

PRIMARY KEY (`orderNumber`, `productCode`),

KEY `productCode` (`productCode`),

CONSTRAINT `orderdetails\_ibfk\_2` FOREIGN KEY (`productCode`) REFERENCES `products` (`productCode`),

CONSTRAINT `orderdetails\_ibfk\_1` FOREIGN KEY (`orderNumber`) REFERENCES `orders` (`orderNumber`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `orders` (

`orderNumber` int(11) NOT NULL,

`orderDate` date NOT NULL,

`requiredDate` date NOT NULL,

`shippedDate` date DEFAULT NULL,

`status` varchar(15) NOT NULL,

`comments` text,

`customerNumber` int(11) NOT NULL,

PRIMARY KEY (`orderNumber`),

KEY `customerNumber` (`customerNumber`),

CONSTRAINT `orders\_ibfk\_1` FOREIGN KEY (`customerNumber`) REFERENCES `customers` (`customerNumber`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

select \* from payments

CREATE TABLE `payments` (

`customerNumber` int(11) NOT NULL,

`checkNumber` varchar(50) NOT NULL,

`paymentDate` date NOT NULL,

`amount` double NOT NULL,

PRIMARY KEY (`customerNumber`,`checkNumber`),

CONSTRAINT `payments\_ibfk\_1` FOREIGN KEY (`customerNumber`) REFERENCES `customers` (`customerNumber`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `productlines` (

`productLine` varchar(50) NOT NULL,

`textDescription` varchar(4000) DEFAULT NULL,

`htmlDescription` mediumtext,

`image` mediumblob,

PRIMARY KEY (`productLine`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `products` (

`productCode` varchar(15) NOT NULL,

`productName` varchar(70) NOT NULL,

`productLine` varchar(50) NOT NULL,

`productScale` varchar(10) NOT NULL,

`productVendor` varchar(50) NOT NULL,

`productDescription` text NOT NULL,

`quantityInStock` smallint(6) NOT NULL,

`buyPrice` double NOT NULL,

`MSRP` double NOT NULL,

PRIMARY KEY (`productCode`),

KEY `productLine` (`productLine`),

CONSTRAINT `products\_ibfk\_1` FOREIGN KEY (`productLine`) REFERENCES `productlines` (`productLine`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

///////////################################################################################################################

SELECT DISTINCT salesRepEmployeeNumber

FROM customers

WHERE salesRepEmployeeNumber IS NOT NULL AND salesRepEmployeeNumber NOT IN (

SELECT employeeNumber

FROM employees

);

#####################################################################################################################################

select \* from payments

insert into `products`(`productCode`,`productName`,`productLine`,`productScale`,`productVendor`,`productDescription`,`quantityInStock`,`buyPrice`,`MSRP`)

values ('S10\_1678','1969 Harley Davidson Ultimate Chopper','Motorcycles','1:10','Min Lin Diecast','This replica features working kickstand, front suspension, gear-shift lever, footbrake lever, drive chain, wheels and steering. All parts are particularly delicate due to their precise scale and require special care and attention.',7933,48.81,95.7),

('S10\_1949','1952 Alpine Renault 1300','Classic Cars','1:10','Classic Metal Creations','Turnable front wheels; steering function; detailed interior; detailed engine; opening hood; opening trunk; opening doors; and detailed chassis.',7305,98.58,214.3),

('S10\_2016','1996 Moto Guzzi 1100i','Motorcycles','1:10','Highway 66 Mini Classics','Official Moto Guzzi logos and insignias, saddle bags located on side of motorcycle, detailed engine, working steering, working suspension, two leather seats, luggage rack, dual exhaust pipes, small saddle bag located on handle bars, two-tone paint with chrome accents, superior die-cast detail , rotating wheels , working kick stand, diecast metal with plastic parts and baked enamel finish.',6625,68.99,118.94),

('S10\_4698','2003 Harley-Davidson Eagle Drag Bike','Motorcycles','1:10','Red Start Diecast','Model features, official Harley Davidson logos and insignias, detachable rear wheelie bar, heavy diecast metal with resin parts, authentic multi-color tampo-printed graphics, separate engine drive belts, free-turning front fork, rotating tires and rear racing slick, certificate of authenticity, detailed engine, display stand\r\n, precision diecast replica, baked enamel finish, 1:10 scale model, removable fender, seat and tank cover piece for displaying the superior detail of the v-twin engine',5582,91.02,193.66),

('S10\_4757','1972 Alfa Romeo GTA','Classic Cars','1:10','Motor City Art Classics','Features include: Turnable front wheels; steering function; detailed interior; detailed engine; opening hood; opening trunk; opening doors; and detailed chassis.',3252,85.68,136),

('S10\_4962','1962 LanciaA Delta 16V','Classic Cars','1:10','Second Gear Diecast','Features include: Turnable front wheels; steering function; detailed interior; detailed engine; opening hood; opening trunk; opening doors; and detailed chassis.',6791,103.42,147.74),

('S12\_1099','1968 Ford Mustang','Classic Cars','1:12','Autoart Studio Design','Hood, doors and trunk all open to reveal highly detailed interior features. Steering wheel actually turns the front wheels. Color dark green.',68,95.34,194.57),

('S12\_1108','2001 Ferrari Enzo','Classic Cars','1:12','Second Gear Diecast','Turnable front wheels; steering function; detailed interior; detailed engine; opening hood; opening trunk; opening doors; and detailed chassis.',3619,95.59,207.8),

('S12\_1666','1958 Setra Bus','Trucks and Buses','1:12','Welly Diecast Productions','Model features 30 windows, skylights & glare resistant glass, working steering system, original logos',1579,77.9,136.67),

('S12\_2823','2002 Suzuki XREO','Motorcycles','1:12','Unimax Art Galleries','Official logos and insignias, saddle bags located on side of motorcycle, detailed engine, working steering, working suspension, two leather seats, luggage rack, dual exhaust pipes, small saddle bag located on handle bars, two-tone paint with chrome accents, superior die-cast detail , rotating wheels , working kick stand, diecast metal with plastic parts and baked enamel finish.',9997,66.27,150.62);

insert into `customers` (`customerNumber`,`customerName`,`contactLastName`,`contactFirstName`,`phone`,`addressLine1`,`addressLine2`,`city`,`state`,`postalCode`,`country`,`salesRepEmployeeNumber`,`creditLimit`)

values (103,'Atelier graphique','Schmitt','Carine ','40.32.2555','54, rue Royale',NULL,'Nantes',NULL,'44000','France',1002,21000),

(112,'Signal Gift Stores','King','Jean','7025551838','8489 Strong St.',NULL,'Las Vegas','NV','83030','USA',1056,71800),

(114,'Australian Collectors, Co.','Ferguson','Peter','03 9520 4555','636 St Kilda Road','Level 3','Melbourne','Victoria','3004','Australia',1088,117300),

(119,'La Rochelle Gifts','Labrune','Janine ','40.67.8555','67, rue des Cinquante Otages',NULL,'Nantes',NULL,'44000','France',1102,118200),

(121,'Baane Mini Imports','Bergulfsen','Jonas ','07-98 9555','Erling Skakkes gate 78',NULL,'Stavern',NULL,'4110','Norway',1143,81700),

(124,'Mini Gifts Distributors Ltd.','Nelson','Susan','4155551450','5677 Strong St.',NULL,'San Rafael','CA','97562','USA',1165,210500),

(125,'Havel & Zbyszek Co','Piestrzeniewicz','Zbyszek ','(26) 642-7555','ul. Filtrowa 68',NULL,'Warszawa',NULL,'01-012','Poland',NULL,0),

(128,'Blauer See Auto, Co.','Keitel','Roland','+49 69 66 90 2555','Lyonerstr. 34',NULL,'Frankfurt',NULL,'60528','Germany',1166,59700),

(129,'Mini Wheels Co.','Murphy','Julie','6505555787','5557 North Pendale Street',NULL,'San Francisco','CA','94217','USA',1188,64600),

(131,'Land of Toys Inc.','Lee','Kwai','2125557818','897 Long Airport Avenue',NULL,'NYC','NY','10022','USA',1216,114900);

insert into `productlines`(`productLine`,`textDescription`,`htmlDescription`,`image`)

values ('Classic Cars','Attention car enthusiasts: Make your wildest car ownership dreams come true. Whether you are looking for classic muscle cars, dream sports cars or movie-inspired miniatures, you will find great choices in this category. These replicas feature superb attention to detail and craftsmanship and offer features such as working steering system, opening forward compartment, opening rear trunk with removable spare wheel, 4-wheel independent spring suspension, and so on. The models range in size from 1:10 to 1:24 scale and include numerous limited edition and several out-of-production vehicles. All models include a certificate of authenticity from their manufacturers and come fully assembled and ready for display in the home or office.',NULL,NULL),

('Motorcycles','Our motorcycles are state of the art replicas of classic as well as contemporary motorcycle legends such as Harley Davidson, Ducati and Vespa. Models contain stunning details such as official logos, rotating wheels, working kickstand, front suspension, gear-shift lever, footbrake lever, and drive chain. Materials used include diecast and plastic. The models range in size from 1:10 to 1:50 scale and include numerous limited edition and several out-of-production vehicles. All models come fully assembled and ready for display in the home or office. Most include a certificate of authenticity.',NULL,NULL),

('Planes','Unique, diecast airplane and helicopter replicas suitable for collections, as well as home, office or classroom decorations. Models contain stunning details such as official logos and insignias, rotating jet engines and propellers, retractable wheels, and so on. Most come fully assembled and with a certificate of authenticity from their manufacturers.',NULL,NULL),

('Ships','The perfect holiday or anniversary gift for executives, clients, friends, and family. These handcrafted model ships are unique, stunning works of art that will be treasured for generations! They come fully assembled and ready for display in the home or office. We guarantee the highest quality, and best value.',NULL,NULL),

('Trains','Model trains are a rewarding hobby for enthusiasts of all ages. Whether you\'re looking for collectible wooden trains, electric streetcars or locomotives, you\'ll find a number of great choices for any budget within this category. The interactive aspect of trains makes toy trains perfect for young children. The wooden train sets are ideal for children under the age of 5.',NULL,NULL),

('Trucks and Buses','The Truck and Bus models are realistic replicas of buses and specialized trucks produced from the early 1920s to present. The models range in size from 1:12 to 1:50 scale and include numerous limited edition and several out-of-production vehicles. Materials used include tin, diecast and plastic. All models include a certificate of authenticity from their manufacturers and are a perfect ornament for the home and office.',NULL,NULL),

('Vintage Cars','Our Vintage Car models realistically portray automobiles produced from the early 1900s through the 1940s. Materials used include Bakelite, diecast, plastic and wood. Most of the replicas are in the 1:18 and 1:24 scale sizes, which provide the optimum in detail and accuracy. Prices range from $30.00 up to $180.00 for some special limited edition replicas. All models include a certificate of authenticity from their manufacturers and come fully assembled and ready for display in the home or office.',NULL,NULL);

insert into `payments`(`customerNumber`,`checkNumber`,`paymentDate`,`amount`)

values (103,'HQ336336','2004-10-19',6066.78),

(112,'JM555205','2003-06-05',14571.44),

(114,'OM314933','2004-12-18',1676.14),

(119,'BO864823','2004-12-17',14191.12),

(121,'HQ55022','2003-06-06',32641.98),

(124,'ND748579','2004-08-20',33347.88),

(125,'GG31455','2003-05-20',45864.03),

(128,'MA765515','2004-12-15',82261.22),

(129,'NP603840','2003-05-31',7565.08),

(131,'NR27552','2004-03-10',44894.74);

insert into `orders`(`orderNumber`, `orderDate`, `requiredDate`, `shippedDate`, `status`, `comments`, `customerNumber`)

values

(10100, '2003-01-06', '2003-01-13', '2003-01-10', 'Shipped', NULL, 103),

(10101, '2003-01-09', '2003-01-18', '2003-01-11', 'Shipped', 'Check on availability.', 112),

(10102, '2003-01-10', '2003-01-18', '2003-01-14', 'Shipped', NULL, 114),

(10103, '2003-01-29', '2003-02-07', '2003-02-02', 'Shipped', NULL, 119),

(10104, '2003-01-31', '2003-02-09', '2003-02-01', 'Shipped', NULL, 121),

(10105, '2003-02-11', '2003-02-21', '2003-02-12', 'Shipped', NULL, 124),

(10106, '2003-02-17', '2003-02-24', '2003-02-21', 'Shipped', NULL, 125),

(10107, '2003-02-24', '2003-03-03', '2003-02-26', 'Shipped', 'Difficult to negotiate with customer. We need more marketing materials', 128),

(10108, '2003-03-03', '2003-03-12', '2003-03-08', 'Shipped', NULL, 129),

(10109, '2003-03-10', '2003-03-19', '2003-03-11', 'Shipped', 'Customer requested that FedEx Ground is used for this shipping', 131);

insert into `orderdetails`(`orderNumber`,`productCode`,`quantityOrdered`,`priceEach`,`orderLineNumber`)

values (10100,'S10\_1678',30,136,3),

(10100,'S10\_1949',50,55.09,2),

(10100,'S10\_2016',22,75.46,4),

(10100,'S10\_4698',49,35.29,1),

(10101,'S10\_4757',25,108.06,4),

(10101,'S10\_4962',26,167.06,1),

(10101,'S12\_1099',45,32.53,3),

(10101,'S12\_1108',46,44.35,2),

(10102,'S12\_1666',39,95.55,2),

(10102,'S12\_2823',41,43.13,1);

insert into `offices`(`officeCode`,`city`,`phone`,`addressLine1`,`addressLine2`,`state`,`country`,`postalCode`,`territory`)

values ('1','San Francisco','+1 650 219 4782','100 Market Street','Suite 300','CA','USA','94080','NA'),

('2','Boston','+1 215 837 0825','1550 Court Place','Suite 102','MA','USA','02107','NA'),

('3','NYC','+1 212 555 3000','523 East 53rd Street','apt. 5A','NY','USA','10022','NA'),

('4','Paris','+33 14 723 4404','43 Rue Jouffroy D\'abbans',NULL,NULL,'France','75017','EMEA'),

('5','Tokyo','+81 33 224 5000','4-1 Kioicho',NULL,'Chiyoda-Ku','Japan','102-8578','Japan'),

('6','Sydney','+61 2 9264 2451','5-11 Wentworth Avenue','Floor #2',NULL,'Australia','NSW 2010','APAC'),

('7','London','+44 20 7877 2041','25 Old Broad Street','Level 7',NULL,'UK','EC2N 1HN','EMEA');

insert into `employees`(`employeeNumber`,`lastName`,`firstName`,`extension`,`email`,`officeCode`,`reportsTo`,`jobTitle`)

values (1002,'Murphy','Diane','x5800','dmurphy@classicmodelcars.com','1',NULL,'President'),

(1056,'Patterson','Mary','x4611','mpatterso@classicmodelcars.com','1',1002,'VP Sales'),

(1076,'Firrelli','Jeff','x9273','jfirrelli@classicmodelcars.com','1',1002,'VP Marketing'),

(1088,'Patterson','William','x4871','wpatterson@classicmodelcars.com','6',1056,'Sales Manager (APAC)'),

(1102,'Bondur','Gerard','x5408','gbondur@classicmodelcars.com','4',1056,'Sale Manager (EMEA)'),

(1143,'Bow','Anthony','x5428','abow@classicmodelcars.com','1',1056,'Sales Manager (NA)'),

(1165,'Jennings','Leslie','x3291','ljennings@classicmodelcars.com','1',1143,'Sales Rep'),

(1166,'Thompson','Leslie','x4065','lthompson@classicmodelcars.com','1',1143,'Sales Rep'),

(1188,'Firrelli','Julie','x2173','jfirrelli@classicmodelcars.com','2',1143,'Sales Rep'),

(1216,'Patterson','Steve','x4334','spatterson@classicmodelcars.com','2',1143,'Sales Rep');

***Trigger***

use gabriele

drop table employees

#######################################################

select \* from employees

create table employees(

employee\_id int primary key,

first\_name varchar (50),

last\_name varchar (50),

hourly\_pay decimal (10,2),

job varchar (50),

hire\_date date,

supervisor\_id int null

);

insert into employees

value (1, "Eugene","Krabs", 25.50, "manager", "2023-01-02", null),

(2, "Suidward","Tentacles", 15.00, "cashier", "2023-01-03", 5),

(3, "Spongebob","Sqaurepants", 12.50, "cook", "2023-01-04", 5),

(4, "Patrick","star", 12.50, "cook", "2023-01-05", 5),

(5, "Sandy","Cheeck", 17.25, "asst. manager", "2023-01-06", 1),

(6, "Shaldon","Pankton", 10.00, "janitor", "2023-01-07", 5);

##########################################################################

alter table employees

add column salary decimal (10,2) after hourly\_pay

update employees

set salary = hourly\_pay \* 2080

###############################################################

create trigger before\_hourly\_pay\_update

before update on employees

for each row

set new.salary = ( new.hourly\_pay \* 2080);

show triggers;

#############################################################################

update employees

set hourly\_pay = 50

where employee\_id = 1;

update employees

set hourly\_pay = hourly\_pay +1;

#######################################################################

delete from employees

where employee\_id =6

DROP TRIGGER IF EXISTS before\_hourly\_pay\_insert;

create Trigger before\_hourly\_pay\_insert

before insert on employees

for each row

set new.salary = ( new.hourly\_pay \* 2080);

##################################################################################

insert into employees

values(6, "Shaldon","Pankton", 10, Null, "janitor", "2023-01-07", 5);

select \* from employees;

########################################################################

select \* from expenses

create table expenses(

expenses\_id int primary key,

expense\_name varchar (50),

expense\_total decimal (10,2)

);

insert into expenses

value (1, "salary", 0),

(2, "supplies", 0),

(3, "taxes", 0);

update expenses

set expense\_total = ( select sum(salary) from employees)

where expense\_name = "salary";

#######################################################################################

create trigger after\_salary\_delete

after delete on employees

for each row

update expenses

set expense\_total = expense\_total - old.salary

where expense\_name = "salary"

delete from employees

where employee\_id = 6;

DROP TRIGGER IF EXISTS after\_salary\_insert;

create trigger after\_salary\_insert

after insert on employees

for each row

update expenses

set expense\_total = expense\_total + new.salary

where expense\_name = "salary"

insert into employees

values(6, "Shaldon","Pankton", 10, Null, "janitor", "2023-01-07", 5);

######################################################################################

create trigger after\_salary\_update

after update on employees

for each row

update expenses

set expense\_total = expense\_total + (new.salary - old.salary)

where expense\_name = "salary";

update employees

set hourly\_pay= 100

where employee\_id = 1;