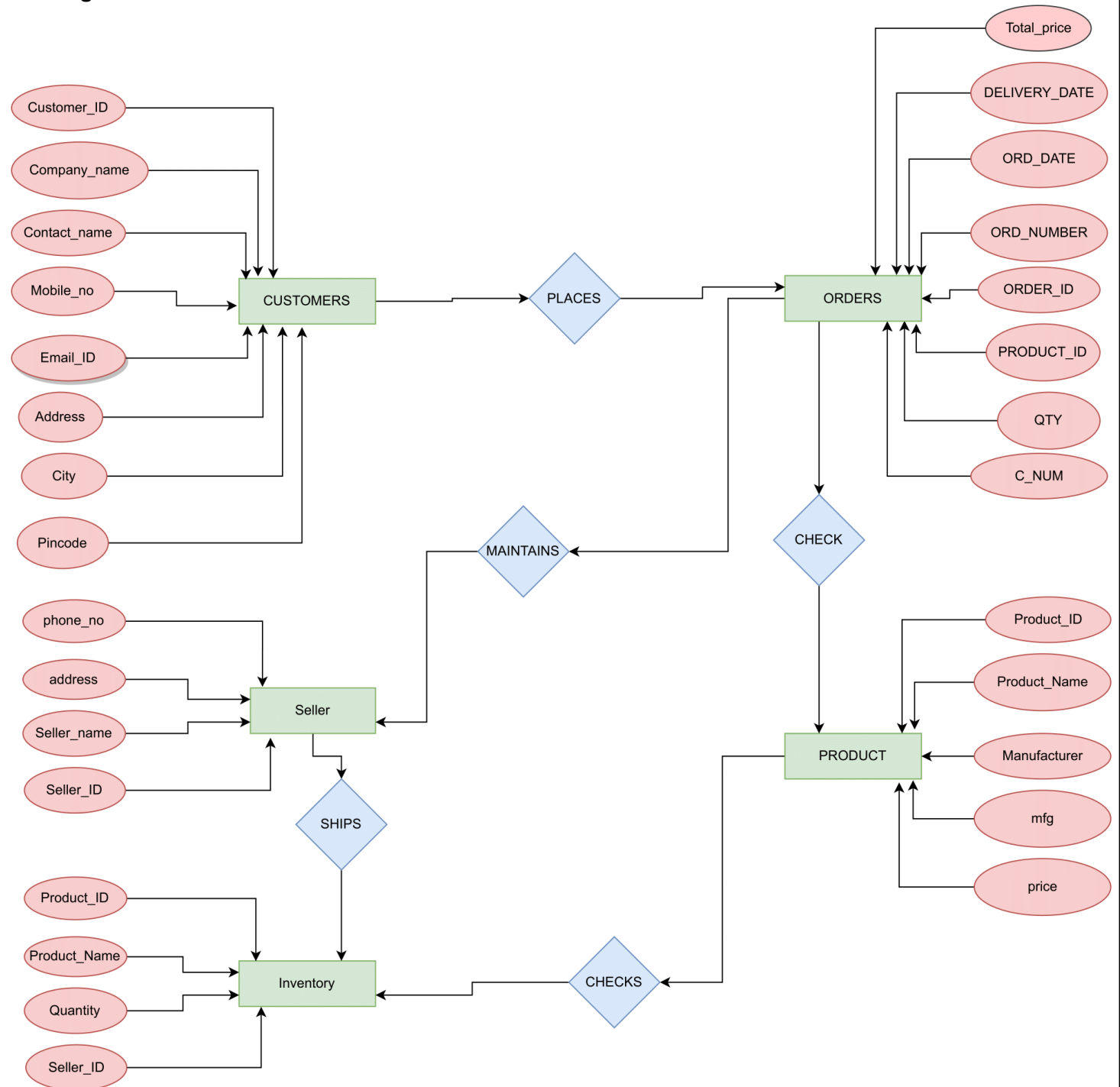


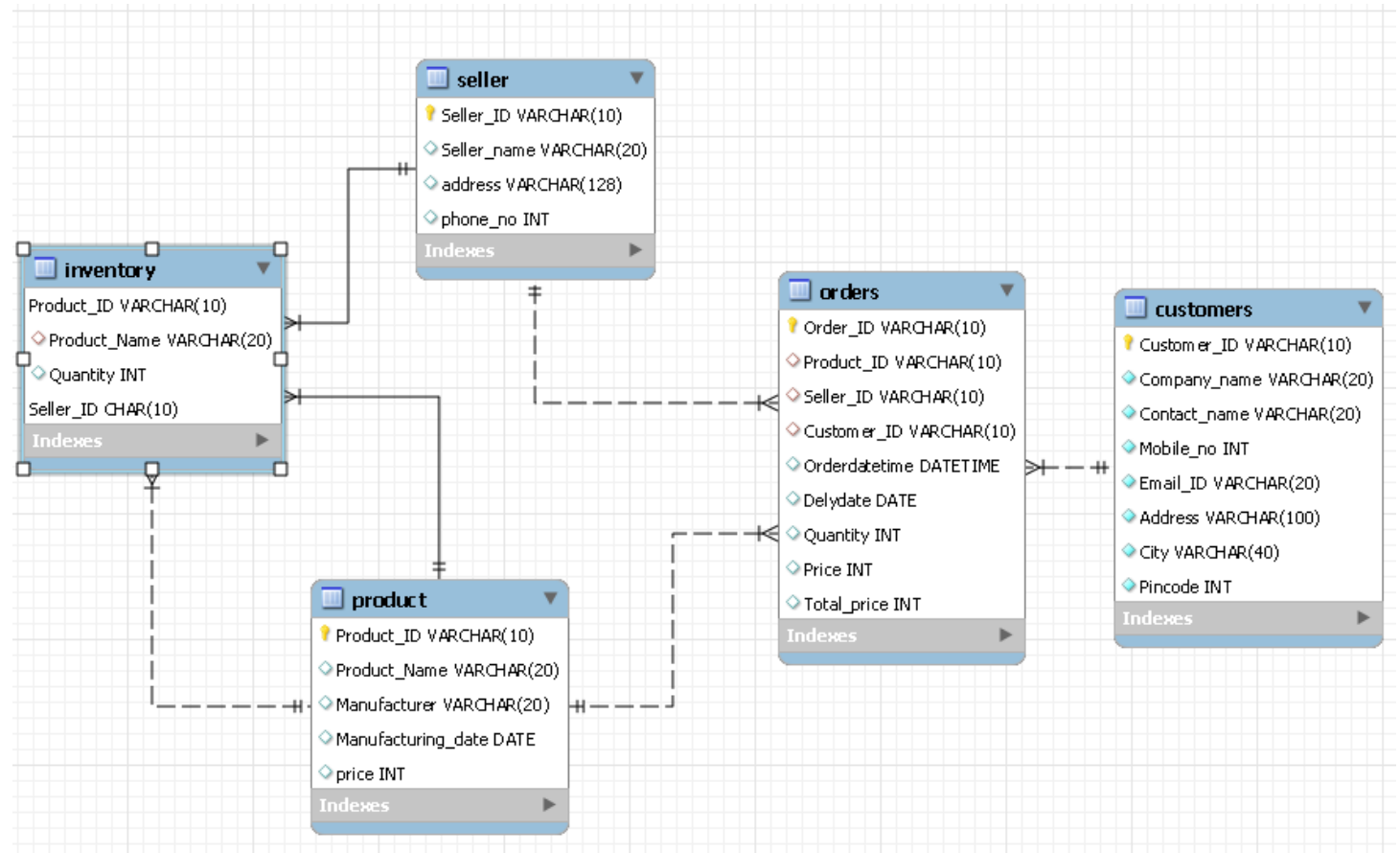
DBT (ER-Assignment)

Team Members (KH)
Shridhar Wakshe(105),
Sakshi Sharma(094),
Sagar Gudekar(092)

ER Diagram



Tables Diagram



MySQL FILE

```

create database project;
use project;
show tables;
  
```

```

create table Customers(
Customer_ID varchar(10) not null,
Company_name varchar(20) not null,
Contact_name varchar(20) not null,
Mobile_no int not null,
Email_ID varchar(20) not null,
Address varchar(100) not null,
City varchar(40) not null,
Pincode int not null,
PRIMARY KEY (Customer_ID)
);
  
```

```

insert into Customers (Customer_ID, Company_name,Contact_name, Mobile_no, Email_ID, Address, City,
Pincode)values
('C_101','AMAZON','JENNY',68128511,'jenny@amazon.com','chandni chawk ,andheri','mumbai',400035),
('C_102','SNAPDEAL','JACKY',58714511,'jacky@shoppy.com','sitara chawk, Bandra','mumbai',400012),
('C_103','FLIPKART','RONNY',54728511,'rommy@flipcart.com','pyaremohan chawk, cst','mumbai',400002),
  
```

```
('C_104','WALMART', 'KENNY',7463574, 'Kenny@walmart.com','LAS VEGAS', 'NEW YORK', 422347 ),
('C_105','APPLE', 'STEVE',4583478, 'STEVE@APPLE.com','DELHI', 'NEW YORK', 456847 ),
('C_106','TATA', 'RATAN',45457378, 'RATAN@TATA.com','KARVENAGAR', 'PUNE', 4564356 ),
('C_107','RELIANCE', 'MUKESH',9537378, 'MUKESH@RELIANCE.com','KOTHRUD', 'PUNE', 4553356 ),
('C_108','TESLA', 'ELON',97357378, 'ELON@TESLA.com','CALIFORNIA', 'USA', 4001356 ),
('C_109','SPACE X', 'ELON',87557378, 'ELON@SPACE X.com','TEXAS', 'USA', 5634356 ),
('C_110','MICROSOFT', 'BILL',86327378, 'BILL@MICROSOFT.com','TIME SQUARE', 'CHINA', 4000356 );
```

```
select * from customers;
DROP TABLE CUSTOMERS;
```

-- Extract all data from the customer table if and only if one or more of the customers in the customer table are located in London

```
SELECT Customer_ID,Contact_name, City -- (sub query)
FROM Customers WHERE EXISTS
(SELECT *FROM Customers WHERE city='CHINA');
```

```
CREATE TABLE Seller (
    Seller_ID varchar(10) NOT NULL,
    Seller_name varchar(20) DEFAULT NULL,
    address varchar(128) DEFAULT NULL,
    phone_no int(10) DEFAULT NULL,
    PRIMARY KEY (Seller_ID)
);
```

insert into seller (Seller_ID, Seller_name, address, phone_no) values

```
('S_001','SAMSUNG', 'Jakkuru Bengaluru',78458745),
('S_002','LG', 'Embassy TechVillage',5879782),
('S_003', 'DELL', 'Okala Industrial Area',7834257),
('S_004', 'HP', 'Panaji',9934257),
('S_005','INTEL', 'Madurai',8879782),
('S_006','AMD', 'Mysore',7779782),
('S_007','NVIDIA', 'Amritsar',6669782),
('S_008','SALSEFORCE', 'Hyderabad',5579782),
('S_009','GOOGLE', 'Udaipur',4479782),
('S_010','HDFC', 'Chennai',3379782);
```

```
select * from seller;
DROP TABLE SELLER;
```

```
create table Product (
    Product_ID varchar(10) NOT NULL,
    Product_Name varchar(20) DEFAULT NULL,
    Manufacturer varchar(20) DEFAULT NULL,
    Manufacturing_date date DEFAULT NULL,
    price int DEFAULT NULL,
    PRIMARY KEY (Product_ID),
    UNIQUE KEY (Product_Name) -- UNIQUE KEY Product_Name(Product_Name)
);
```

insert into product(Product_ID, Product_Name, Manufacturer, Manufacturing_date, price) values

```
('P_001', 'Headphones', 'SAMSUNG', '2021-04-23',1000),
('P_002','CAMERA', 'LG', '2022-03-11',70000),
('P_003', 'MIC', 'DELL', '2021-02-20',40000 ),
('P_004','laptop', 'HP', '2019-06-05',10000),
('P_005','PROCESSOR', 'INTEL', '2020-05-23',5000),
('P_006','GRAPHIC CARD', 'AMD', '2021-09-23',65000),
('P_007','RAM', 'NVIDIA', '2021-04-19',4000),
('P_008','SOFTWARE', 'SALSEFORCE', '2019-05-23',10000),
('P_009','MAPS', 'GOOGLE', '2020-06-25',34000),
```

```
('P_010','ONLINE BANKING', 'HDFC', '2021-03-23',56000);
```

```
select * from product;  
drop table product;
```

```
SELECT * -- (sub query)  
FROM product  
WHERE Product_ID IN (SELECT Product_ID  
FROM product  
WHERE price > 45000) ;
```

```
TRUNCATE PRODUCT;  
select * from product;  
DELETE FROM PRODUCT WHERE PRODUCT_ID = 'P_001';  
UPDATE PRODUCT SET PRICE = 20000 WHERE Product_ID = 'P_002';
```

```
CREATE TABLE Inventory (  
Product_ID varchar(10) NOT NULL,  
Product_Name varchar(20) DEFAULT NULL,  
Quantity int unsigned DEFAULT NULL,  
Seller_ID char(10) NOT NULL,  
PRIMARY KEY (Product_ID,Seller_ID),  
CONSTRAINT fk01 FOREIGN KEY (Product_ID) REFERENCES product (Product_ID) ON DELETE  
CASCADE ,  
CONSTRAINT fk02 FOREIGN KEY (Product_Name) REFERENCES product (Product_Name) ON DELETE  
CASCADE ,  
CONSTRAINT fk03 FOREIGN KEY (Seller_ID) REFERENCES seller (Seller_ID) ON DELETE CASCADE  
);  
insert into Inventory(Product_ID, Product_Name, Quantity, Seller_ID ) values  
( 'P_001', 'Headphones', 100, 'S_001'),  
( 'P_002','CAMERA', 100,'S_002'),  
( 'P_003', 'MIC',200, 'S_003' ),  
( 'P_004','laptop',245, 'S_004' ),  
( 'P_005','PROCESSOR', 98, 'S_005' ),  
( 'P_006','GRAPHIC CARD', 100, 'S_006'),  
( 'P_007','RAM', 150, 'S_007' ),  
( 'P_008','SOFTWARE',135, 'S_008' ),  
( 'P_009','MAPS', 120, 'S_009' ),  
( 'P_010','ONLINE BANKING',10,'S_010' );
```

```
select * from inventory;  
drop table inventory;
```

```
create table Orders(  
Order_ID varchar(10) NOT NULL,  
Product_ID varchar(10) DEFAULT NULL,  
Seller_ID varchar(10) DEFAULT NULL,  
Customer_ID varchar(10) DEFAULT NULL,  
Orderdatetime datetime DEFAULT NULL,  
Delydate date,  
Quantity int DEFAULT NULL,  
Price int unsigned,  
Total_price int,  
PRIMARY KEY (Order_ID),  
CONSTRAINT fk04 FOREIGN KEY (Product_ID) REFERENCES product (Product_ID) ON DELETE  
CASCADE ,  
CONSTRAINT fk05 FOREIGN KEY (Seller_ID) REFERENCES seller (Seller_ID) ON DELETE CASCADE ,  
CONSTRAINT fk06 FOREIGN KEY (Customer_ID) REFERENCES customers (Customer_ID) ON DELETE
```

CASCADE

);

insert into Orders(Order_ID, Product_ID, Seller_ID, Customer_ID, Orderdatetime, Delydate, Quantity, Price)
values

('O_001', 'P_001', 'S_001', 'C_101', sysdate(), DATE_ADD(sysdate(), INTERVAL 3 DAY), 2, 1000);

insert into Orders(Order_ID, Product_ID, Seller_ID, Customer_ID, Orderdatetime, Delydate, Quantity, Price)
values

('O_002', 'P_002', 'S_002', 'C_102', NOW(), DATE_ADD(sysdate(), INTERVAL 3 DAY), 2, 70000);

insert into Orders(Order_ID, Product_ID, Seller_ID, Customer_ID, Orderdatetime, Delydate, Quantity, Price)
values

('O_003', 'P_003', 'S_003', 'C_103', sysdate(), DATE_ADD(sysdate(), INTERVAL 3 DAY), 2, 40000);

insert into Orders(Order_ID, Product_ID, Seller_ID, Customer_ID, Orderdatetime, Delydate, Quantity, Price)
values

('O_004', 'P_004', 'S_004', 'C_104', sysdate(), DATE_ADD(sysdate(), INTERVAL 3 DAY), 2, 10000);

insert into Orders(Order_ID, Product_ID, Seller_ID, Customer_ID, Orderdatetime, Delydate, Quantity, Price)
values

('O_005', 'P_005', 'S_005', 'C_105', sysdate(), DATE_ADD(sysdate(), INTERVAL 3 DAY), 2, 5000);

insert into Orders(Order_ID, Product_ID, Seller_ID, Customer_ID, Orderdatetime, Delydate, Quantity, Price)
values

('O_006', 'P_006', 'S_006', 'C_106', sysdate(), DATE_ADD(sysdate(), INTERVAL 3 DAY), 2, 65000);

insert into Orders(Order_ID, Product_ID, Seller_ID, Customer_ID, Orderdatetime, Delydate, Quantity, Price)
values

('O_007', 'P_007', 'S_007', 'C_107', sysdate(), DATE_ADD(sysdate(), INTERVAL 3 DAY), 2, 4000);

select * from orders;

DROP TABLE ORDERS;

truncate orders;

SELECT * -- (joins)

FROM Customers

NATURAL JOIN Inventory

NATURAL JOIN PRODUCT;

Triggers

DELIMITER //

CREATE TRIGGER updatetime BEFORE INSERT ON orders FOR EACH ROW

BEGIN

SET NEW.orderdatetime = NOW();

END//

DELIMITER ;

DELIMITER //

CREATE TRIGGER inventorytrigger AFTER INSERT ON orders

FOR EACH ROW

begin

DECLARE Q int;

DECLARE P varchar(10);

SELECT Product_ID INTO P FROM orders ORDER BY Order_ID DESC LIMIT 1;

SELECT Quantity INTO Q FROM orders ORDER BY Order_ID DESC LIMIT 1;

```
UPDATE inventory
SET Quantity=Quantity-Q WHERE Product_ID=P;
END//
DELIMITER ;
```

PROCEDURE

```
DELIMITER //
CREATE PROCEDURE getsellerorders(IN TEMP VARCHAR(20))
BEGIN
    SELECT * FROM orders where Seller_ID=TEMP;
END //
DELIMITER ;
```

```
DROP PROCEDURE getsellerorders;
call getsellerorders('S_001');
```

```
DELIMITER //
CREATE PROCEDURE getorders(IN temp2 VARCHAR(20))
BEGIN
    declare x int;
    DECLARE Y INT;
    SELECT Quantity INTO X FROM orders WHERE Customer_ID=TEMP2 ;
    SET Y = TOTAL(x); -- function call to update total price
    UPDATE ORDERS SET total_price = Y WHERE Customer_ID = TEMP2;
    SELECT * FROM orders WHERE Customer_ID=temp2 ;
END //
DELIMITER ;
```

```
DROP procedure getorders;
call getorders('C_101');
```

FUNCTION

```
DELIMITER //
CREATE FUNCTION TOTAL(Q INT)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE PID VARCHAR(10);
    DECLARE P INT;
    DECLARE Z INT;
    SELECT Product_ID INTO PID FROM ORDERS WHERE QUANTITY = Q;
    SELECT PRICE INTO P FROM PRODUCT WHERE Product_ID = PID;
    SET Z= (P*Q);
    INSERT INTO XYZ VALUES (Z);
    RETURN Z;
END; //
DELIMITER ;
```

```
DROP FUNCTION TOTAL;
```

