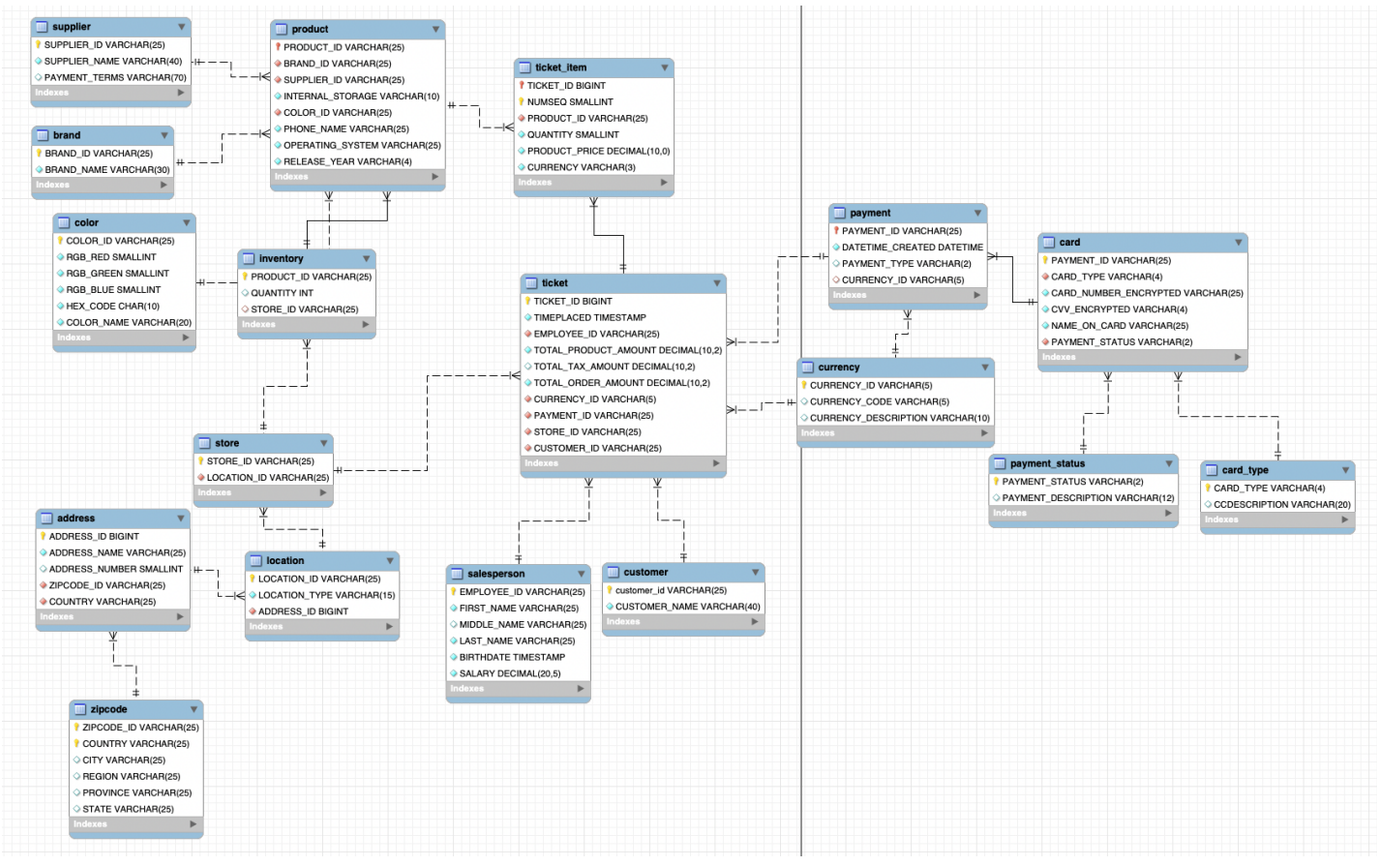
SQL II - GROUP 5 ASSIGNMENT

ADRIAN KU
IGNACIO PIRE RUBIO
IVAN LOPEZ CANSADO
LAKSHMI PRIYANKA
LIA DOLLISON
LUDOVICO GANDOLFI
ROMAN ZOTKIN

TABLE OF CONTENTS

01 - ERD



*find attached .mwb file

02 – DDL & DMLS

CREATE STATEMENTS:

```
CREATE TABLE BRAND(
BRAND_ID VARCHAR(25) NOT NULL,
BRAND_NAME VARCHAR(30) NOT NULL,
PRIMARY KEY (BRAND_ID)
);

CREATE TABLE SUPPLIER(
SUPPLIER_ID VARCHAR(25) NOT NULL,
SUPPLIER_NAME VARCHAR(40) NOT NULL,
PAYMENT_TERMS VARCHAR(70),
PRIMARY KEY (SUPPLIER_ID));

CREATE TABLE COLOR(
COLOR_ID VARCHAR(25) NOT NULL,
RGB_RED SMALLINT NOT NULL,
RGB_GREEN SMALLINT NOT NULL,
RGB_BLUE SMALLINT NOT NULL,
HEX_CODE CHAR(10) NOT NULL,
COLOR_NAME VARCHAR(20) NOT NULL,
PRIMARY KEY (COLOR_ID));

CREATE TABLE ZIPCODE(
ZIPCODE_ID VARCHAR(25) NOT NULL,
COUNTRY VARCHAR(25) NOT NULL,
CITY VARCHAR(25),
REGION VARCHAR(25),
PROVINCE VARCHAR(25),
STATE VARCHAR(25),
PRIMARY KEY(ZIPCODE_ID, COUNTRY));

create table ADDRESS(
ADDRESS_ID BIGINT NOT NULL,
ADDRESS_NAME VARCHAR(25) NOT NULL,
ADDRESS_NUMBER SMALLINT,
ZIPCODE_ID VARCHAR(25) NOT NULL,
COUNTRY VARCHAR(25) NOT NULL,
PRIMARY KEY (ADDRESS_ID),
FOREIGN KEY(ZIPCODE_ID, COUNTRY) REFERENCES ZIPCODE(ZIPCODE_ID, COUNTRY));

CREATE TABLE LOCATION(
LOCATION_ID VARCHAR(25) NOT NULL,
LOCATION_TYPE VARCHAR(15) NOT NULL,
ADDRESS_ID BIGINT NOT NULL,
PRIMARY KEY (LOCATION_ID),
FOREIGN KEY (ADDRESS_ID) REFERENCES ADDRESS(ADDRESS_ID));

CREATE TABLE STORE (
STORE_ID VARCHAR(25) NOT NULL,
LOCATION_ID VARCHAR(25) NOT NULL,
PRIMARY KEY (STORE_ID),
FOREIGN KEY (LOCATION_ID) REFERENCES LOCATION(LOCATION_ID));

create table INVENTORY (
PRODUCT_ID VARCHAR(25) NOT NULL,
QUANTITY INT,
STORE_ID VARCHAR(25),
PRIMARY KEY (PRODUCT_ID),
FOREIGN KEY(STORE_ID) REFERENCES STORE(STORE_ID));

CREATE TABLE PRODUCT(
PRODUCT_ID VARCHAR(25) NOT NULL,
BRAND_ID VARCHAR(25) NOT NULL,
SUPPLIER_ID VARCHAR(25) NOT NULL,
INTERNAL_STORAGE VARCHAR(10) NOT NULL,
COLOR_ID VARCHAR(25) NOT NULL,
PHONE_NAME VARCHAR(25) NOT NULL,
OPERATING_SYSTEM VARCHAR(25) NOT NULL,
RELEASE_YEAR VARCHAR(4) NOT NULL,
PRIMARY KEY (PRODUCT_ID),
FOREIGN KEY (BRAND_ID) REFERENCES BRAND(BRAND_ID),
FOREIGN KEY (SUPPLIER_ID) REFERENCES SUPPLIER(SUPPLIER_ID),
FOREIGN KEY (PRODUCT_ID) REFERENCES INVENTORY(PRODUCT_ID),
FOREIGN KEY (COLOR_ID) REFERENCES COLOR(COLOR_ID));

CREATE TABLE SALESPERSON (
EMPLOYEE_ID VARCHAR(25) NOT NULL,
FIRST_NAME VARCHAR(25) NOT NULL,
MIDDLE_NAME VARCHAR(25),
LAST_NAME VARCHAR(25) NOT NULL,
BIRTHDATE TIMESTAMP NOT NULL,
SALARY DECIMAL(20,5) NOT NULL,
PRIMARY KEY (EMPLOYEE_ID));

create table CUSTOMER( customer_id VARCHAR(25) not null,
CUSTOMER_NAME varchar(40) not null,
PRIMARY KEY(CUSTOMER_ID));

CREATE TABLE CARD_TYPE(
CARD_TYPE VARCHAR(4) NOT NULL,
CCDESCRIPTION VARCHAR(20),
PRIMARY KEY (CARD_TYPE));

CREATE TABLE PAYMENT_STATUS (
PAYMENT_STATUS VARCHAR(2) NOT NULL,
PAYMENT_DESCRIPTION VARCHAR(12),
PRIMARY KEY (PAYMENT_STATUS));

CREATE TABLE CARD (
PAYMENT_ID VARCHAR(25),
CARD_TYPE VARCHAR(4) NOT NULL,
CARD_NUMBER_ENCRYPTED VARCHAR(25) NOT NULL,
CVV_ENCRYPTED VARCHAR(4) NOT NULL,
NAME_ON_CARD VARCHAR(25) NOT NULL,
PAYMENT_STATUS VARCHAR(2) NOT NULL,
PRIMARY KEY (PAYMENT_ID),
FOREIGN KEY (CARD_TYPE) REFERENCES CARD_TYPE(CARD_TYPE),
FOREIGN KEY (PAYMENT_STATUS) REFERENCES PAYMENT_STATUS(PAYMENT_STATUS));

CREATE TABLE CURRENCY(
CURRENCY_ID VARCHAR(5),
CURRENCY_CODE VARCHAR(5),
CURRENCY_DESCRIPTION VARCHAR(10),
PRIMARY KEY(CURRENCY_ID));

CREATE TABLE PAYMENT (
PAYMENT_ID VARCHAR(25) NOT NULL,
DATETIME_CREATED DATETIME NOT NULL,
PAYMENT_TYPE VARCHAR(2),
CURRENCY_ID VARCHAR(5),
PRIMARY KEY (PAYMENT_ID),
FOREIGN KEY(PAYMENT_ID) REFERENCES CARD(PAYMENT_ID),
FOREIGN KEY(CURRENCY_ID) REFERENCES CURRENCY(CURRENCY_ID));

CREATE TABLE TICKET (
TICKET_ID BIGINT NOT NULL,
TIMEPLACED TIMESTAMP NOT NULL,
EMPLOYEE_ID VARCHAR(25) NOT NULL,
TOTAL_PRODUCT_AMOUNT DECIMAL(10,2) NOT NULL,
TOTAL_TAX_AMOUNT DECIMAL(10,2),
TOTAL_ORDER_AMOUNT DECIMAL(10,2) NOT NULL,
CURRENCY_ID VARCHAR(5) NOT NULL,
PAYMENT_ID VARCHAR(25) NOT NULL,
STORE_ID VARCHAR(25) NOT NULL,
CUSTOMER_ID VARCHAR(25) NOT NULL,
PRIMARY KEY (TICKET_ID),
FOREIGN KEY (STORE_ID) REFERENCES STORE(STORE_ID),
FOREIGN KEY (CURRENCY_ID) REFERENCES CURRENCY(CURRENCY_ID),
FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID),
FOREIGN KEY (EMPLOYEE_ID) REFERENCES SALESPERSON(EMPLOYEE_ID),
FOREIGN KEY (PAYMENT_ID) REFERENCES PAYMENT(PAYMENT_ID));

CREATE TABLE TICKET_ITEM(
TICKET_ID BIGINT NOT NULL,
NUMSEQ SMALLINT NOT NULL,
PRODUCT_ID VARCHAR(25) NOT NULL,
QUANTITY SMALLINT NOT NULL,
PRODUCT_PRICE DECIMAL NOT NULL,
CURRENCY VARCHAR(3) NOT NULL,
PRIMARY KEY (TICKET_ID, NUMSEQ),
FOREIGN KEY (PRODUCT_ID) REFERENCES PRODUCT(PRODUCT_ID),
FOREIGN KEY (TICKET_ID) REFERENCES TICKET(TICKET_ID));
```

INSERT STATEMENTS:

```
INSERT INTO SALESPERSON
(EMPLOYEE_ID, FIRST_NAME, MIDDLE_NAME, LAST_NAME, BIRTHDATE, SALARY) VALUES
(781, 'Ivan',NULL, 'Lopez','1998-07-08',75500),
(782,'Nacho',NULL, 'Pire', '1995-06-25',76100),
(783, 'Ludovico', NULL, 'Gandolfi', '1997-09-27',73000),
(784,'Adrian', NULL, 'Ku', '1999-03-20', 65700),
(785, 'Priyanka', NULL, 'Jakka', '1993-05-13', 77000);

INSERT INTO ZIPCODE VALUES
('28006','Spain','Madrid','Principe De Vergara','Madrid','Madrid'),
('28007','Spain','Madrid','Urbita Kalea','Madrid','Madrid'),
('28042','Spain','Madrid','Hispanidad','Spain','Madrid'),
('3007','Spain','barahas','Oscar Esplá','barahas','Madrid'),
('41007','Spain','Alicante','Seville','Alicante','Madrid');

INSERT INTO ADDRESS VALUES
('101','Maria De Molina',31,'28006','Spain'),
('102','Urbita Kalea',25,'28007','Spain'),
('103','Hispanidad',NULL,'28042','Spain'),
('104','Seville',NULL,'41007','Spain'),
('105','Oscar Esplá',49,'3007','Spain');

INSERT INTO LOCATION VALUES
('6010','Mall','101'),
('6011','Street','102'),
('6012','Airport','103'),
('6013','Station','104'),
('6014','Avenue','105');

INSERT INTO STORE VALUES
('581','6010'),
('582','6011'),
('583','6012'),
('584','6013'),
('585','6013'),
('586','6014');

INSERT INTO BRAND VALUES
('SAM1', 'SAMSUNG'),
('APP1', 'APPLE'),
('XIAOMI1', 'XIAOMI');

INSERT INTO SUPPLIER VALUES
('V01279','Vodafone','Payment will be done once the order is completed'),
('O01094','Orange','Payment will be done once the order is completed'),
('M25413','Movie-Star','Payment will be done once the order is completed');

INSERT INTO INVENTORY (PRODUCT_ID,QUANTITY,STORE_ID) VALUES
('A2894',1000, '581'),
('A2895',2469, '585'),
('A2896',5423, '583'),
('A2897',7423, '584'),
('SM-G793',1235, '586'),
('SM-G793N',2541,'582'),
('SUM-G793U',153,'583'),
('XM39028U',2513,'581'),
('XM29028S',1254,'581'),
('XM29028N',2513, '582');

INSERT INTO COLOR VALUES
('E123M',143,43,6,'#FB0000','MIDNIGHT'),
('E173R',136,8,8,'#FF0000','RED'),
('E133P',159,43,104,'#FFC0CB','PINK'),
('E113B',0,255,255,' #0000FF','BLUE'),
('E103G',127,255,212,'#00FF00', 'GREEN'),
('E199O',265,165,0,'#FFA500', 'ORANGE');

INSERT INTO PRODUCT VALUES
('A2894', 'APP1','V01279','128', 'E123M', 'iPhone 14 Pro Max 128GB', 'IOS', '2022'),
('A2895', 'APP1','O01094','256', 'E173R', 'iPhone 14 Pro Max 256GB', 'IOS', '2022'),
('A2896', 'APP1','V01279', '128', 'E133P', 'iPhone 14 Pro 128GB', 'IOS', '2022'),
('A2897', 'APP1','O01094', '256', 'E113B', 'iPhone 14 Pro 256GB', 'IOS', '2022'),
('SM-G793', 'SAM1','V01279','128', 'E103G', 'Samsung Galaxy S22', 'ANDROID', '2021'),
('SM-G793N','SAM1','O01094','256','E199O','Samsung Galaxy Note11', 'ANDROID', '2021'),
('SUM-G793U', 'SAM1','V01279','128','E133P', 'Samsung Galaxy S22U', 'ANDROID', '2021'),
('XM39028U', 'XIAOMI1','M25413','256','E113B', 'Xiaomi 12 Ultra', 'ANDROID','2022'),
('XM29028S', 'XIAOMI1','M25413','128','E103G', 'Xiaomi 12S', 'ANDROID', '2022'),
('XM29028N', 'XIAOMI1','M25413','256', 'E123M', 'Xiaomi 12 Note', 'ANDROID', '2022');

INSERT INTO CURRENCY VALUES
('EUR','E101','Euros');

INSERT INTO CARD_TYPE (CARD_TYPE, CCDESCRIPTION) VALUES ('MC','Master Card'),
('VC','Visa Card'),
('DS','Discover Card'),
('AM','American Express'),
('DC','Diners Club'),
('BK', 'Others Bank Cards');

INSERT INTO PAYMENT_STATUS (PAYMENT_STATUS, PAYMENT_DESCRIPTION) VALUES
('NW','New_Payment'),
('SU','Success'),
('AP','Approving'),
('F','Failed'),
('C','Cancelled'),
('E', 'Expired');

INSERT INTO CARD (PAYMENT_ID, PAYMENT_STATUS, CARD_TYPE,
CARD_NUMBER_ENCRYPTED, CVV_ENCRYPTED, NAME_ON_CARD) VALUES
('1001','SU','MC','543565785423xxxx','XXX','Jessica Alonzo'),
('1002','SU','VC','465789785412xxxx','XXX','Constante Noppers'),
('1003','F', 'DS','956123425417xxxx','XXX','Mikael Matsyuma'),
('1004','SU','DS','912356765419xxxx','XXX','Quy Kantola'),
('1005','SU','DS','923467825475xxxx','XXX','Norman Lutters'),
('1006','SU','MC','548126500674xxxx','XXX','Bao Aliyev'),
('1007','SU','MC','548126746911xxxx','XXX','Leokadia Yli-Hannuksela'),
('1008','SU','MC','548126109505xxxx','XXX','Valrie Kuester');

INSERT INTO PAYMENT (PAYMENT_ID, DATETIME_CREATED,
PAYMENT_TYPE,CURRENCY_ID) VALUES
('1001', '2022-06-05 13:28:00','CC','EUR'),
('1002', '2022-08-05 14:28:10','CC','EUR'),
('1003', '2022-12-09 10:45:00','CC','EUR'),
('1004', '2022-12-20 20:28:20','CC','EUR'),
('1005', '2022-12-11 21:28:50','CC','EUR'),
('1006', '2022-12-22 20:28:00','CC','EUR'),
('1007', '2022-12-23 19:16:00','CC','EUR'),
('1008', '2022-12-24 18:28:50','CC','EUR');

INSERT INTO CUSTOMER VALUES
('C1425867','Kenny Cisneros'),
('C254769','Stuart Nash'),
('C25478662','Constance Barnes'),
('C54758632','Phillip Fisher'),
('C87654422','Warren Mayer'),
('C1452876','Yasmine Cross'),
('C5423687','Keelan Reid'),
('C2475963','Trinity Burnett');

INSERT INTO TICKET
(TICKET_ID, TIMEPLACED, EMPLOYEE_ID, TOTAL_PRODUCT_AMOUNT,
TOTAL_TAX_AMOUNT, TOTAL_ORDER_AMOUNT, CURRENCY_ID, PAYMENT_ID,
STORE_ID,CUSTOMER_ID) VALUES
(981, '2022-05-06 13:28:00', '781', 2298.00, 482.58, 2780.58, 'EUR', '1001',
'581','C1425867'),
(982, '2022-05-08 14:28:00', '782', 999.00, 209.79, 1208.79, 'EUR', '1002',
'581','C254769'),
(983, '2022-09-12 10:45:00', '783', 2179.00, 457.59, 2636.59, 'EUR', '1003',
'582','C25478662'),
(984, '2022-12-20 20:28:00', '784', 1830.00, 384.30, 2214.30, 'EUR', '1004',
'582','C54758632'),
(985, '2022-12-11 21:28:00', '785', 1000.00, 210.00, 1210.00, 'EUR', '1005',
'583','C87654422'),
(986, '2022-11-22 09:58:00', '782', 1360.00, 285.60, 1645.60, 'EUR', '1006',
'586','C1452876'),
(987, '2022-08-15 10:32:00', '781', 2398.00, 503.58, 2901.58, 'EUR', '1007',
'585','C5423687'),
(988, '2022-06-03 13:22:00', '785', 2430.00, 510.30, 2940.30, 'EUR', '1008',
'584','C2475963');

INSERT INTO TICKET_ITEM (Ticket_ID, NUMSEQ, Product_ID, Quantity, Product_price,
Currency) VALUES
( 981, 1, 'A2894', 1, 1099, 'EUR'),
( 981, 2, 'A2895', 1, 1199, 'EUR'),
( 982, 3, 'A2896', 1, 999, 'EUR'),
( 983, 4, 'A2897', 1, 1099, 'EUR'),
( 983, 5, 'SM-G793', 1, 680, 'EUR'),
( 983, 6, 'SM-G793N', 1, 400, 'EUR'),
( 984, 7, 'SUM-G793U', 1, 1020, 'EUR'),
( 984, 8, 'XM39028U', 1, 810, 'EUR'),
( 985, 9, 'XM29028S', 1, 505, 'EUR'),
( 985, 10, 'XM29028N', 1, 495, 'EUR'),
( 986, 11, 'SM-G793', 2, 680, 'EUR'),
( 987, 12, 'A2895', 2, 1199, 'EUR'),
( 988, 13, 'XM39028U', 3, 810, 'EUR');
```

03 - MODEL EXPLANATION

THE MOBILE PHONE STORE MODEL

Our model is designed to include all mobile phone shops in Spain. This model helps shops all over Spain to manage purchases of different types of phones (Apple, Samsung, and Xiaomi) in any shop. The ticket is the main entity of this model. The details of the shop, such as its location and the address where the customer buys the product, are linked to the ticket details. The payment details are also included in our database in encrypted format in accordance with government-issued standards. The vendor who sold the specific product to the customer is also linked to the ticket entity. As this model is designed for mobile shops in Spain, it only handles EURO currency. This model has information about all the products that are available in the mobile shops in Spain and the supplier of the product to the mobile shop. The product has information about the color and its year of release along with its internal storage details which are mandatory for a customer to select purchase options. (The product details are also linked to the Ticket entity).

This model has the TICKET as the main entity and the seller, the shop, the product, the brand, and the payment details are linked to the ticket entity.

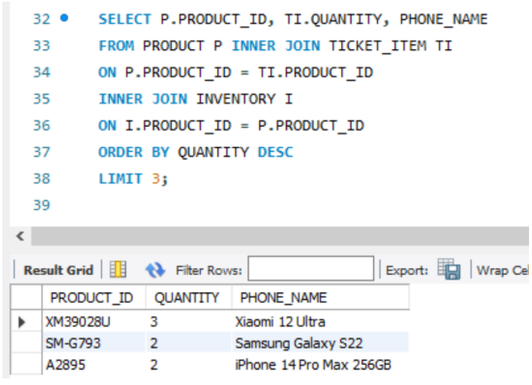
DISCLAIMER:
This model stores the customer and the customer's payment information on the terms agreed upon by the customer during the purchase of the products.

WHAT DOES THE MODEL ADD TO THE COMPANY:
This model helps shops throughout Spain to manage the customer's product purchase information. They have this information stored in their operational database, which helps shops analyze which product sells best in which shop, which payment method is used most for payments, and other data that management can use for data analysis to improve services accordingly.

04 – QUESTIONS & QUERIES

1) Which are the top 3 phone sales?

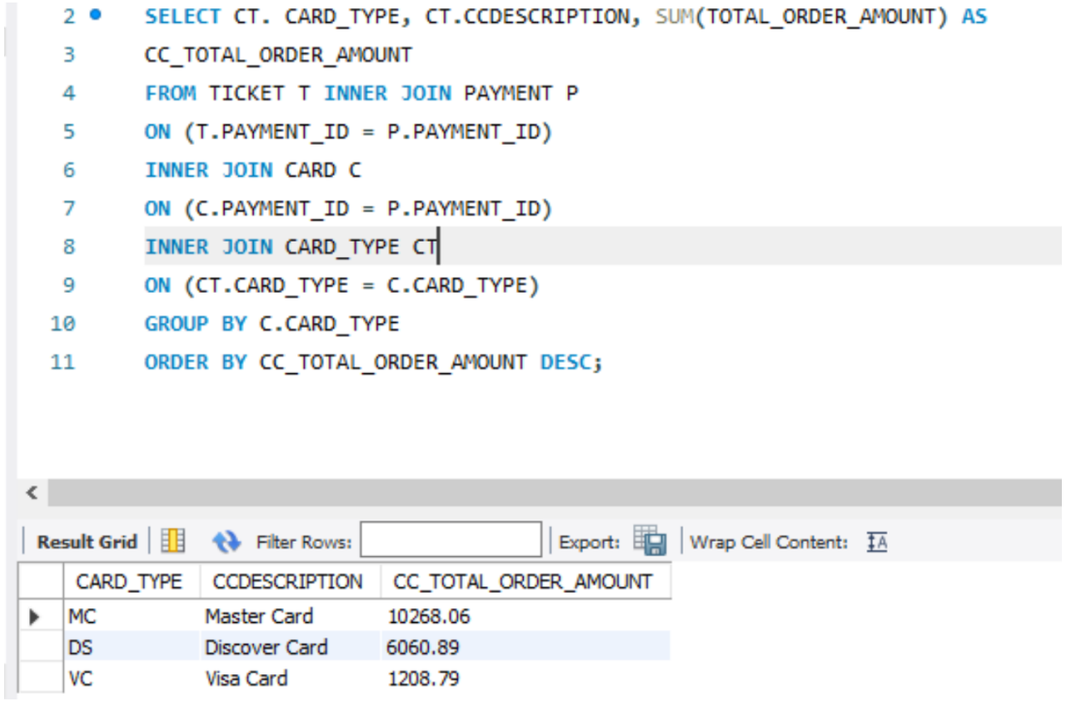
```
SELECT P.PRODUCT_ID, TI.QUANTITY, PHONE_NAME
FROM PRODUCT P INNER JOIN TICKET_ITEM TI
ON P.PRODUCT_ID = TI.PRODUCT_ID
INNER JOIN INVENTORY I
ON I.PRODUCT_ID = P.PRODUCT_ID
ORDER BY QUANTITY DESC
LIMIT 3;
```



PRODUCT_ID	QUANTITY	PHONE_NAME
XM39028U	3	Xiaomi 12 Ultra
SM-G793	2	Samsung Galaxy S22
A2895	2	iPhone 14 Pro Max 256GB

2) Show the revenue per type of credit card.

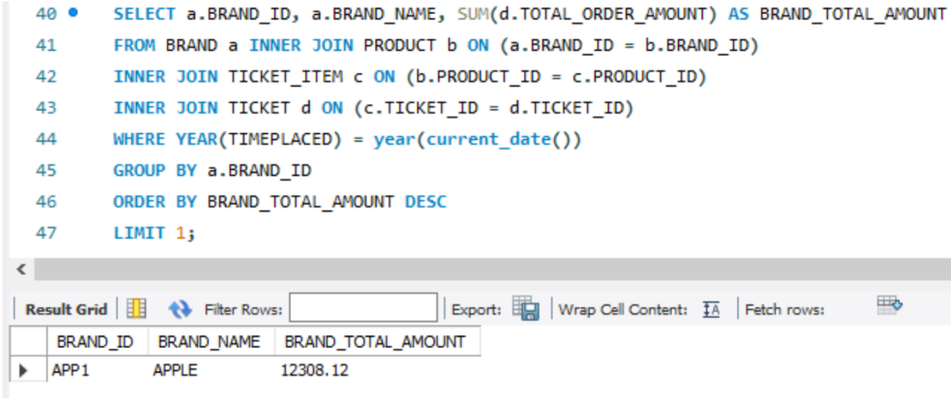
```
SELECT CT. CARD_TYPE, CT.CCDESCRIPTION, SUM(TOTAL_ORDER_AMOUNT) AS
CC_TOTAL_ORDER_AMOUNT
FROM TICKET T INNER JOIN PAYMENT P
ON (T.PAYMENT_ID = P.PAYMENT_ID)
INNER JOIN CARD C
ON (C.PAYMENT_ID = P.PAYMENT_ID)
INNER JOIN CARD_TYPE CT
ON (CT.CARD_TYPE = C.CARD_TYPE)
GROUP BY C.CARD_TYPE
ORDER BY CC_TOTAL_ORDER_AMOUNT DESC;
```



CARD_TYPE	CCDESCRIPTION	CC_TOTAL_ORDER_AMOUNT
MC	Master Card	10268.06
DS	Discover Card	6060.89
VC	Visa Card	1208.79

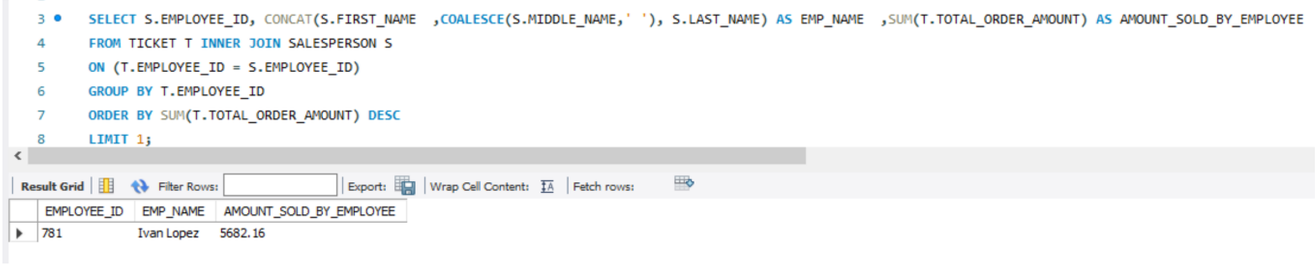
3) Which brand are we selling the most this year?

```
SELECT a.BRAND_ID, a.BRAND_NAME, SUM(d.TOTAL_ORDER_AMOUNT) AS BRAND_TOTAL_AMOUNT
FROM BRAND a INNER JOIN PRODUCT b ON (a.BRAND_ID = b.BRAND_ID)
INNER JOIN TICKET_ITEM c ON (b.PRODUCT_ID = c.PRODUCT_ID)
INNER JOIN TICKET d ON (c.TICKET_ID = d.TICKET_ID)
WHERE YEAR(TIMEPLACED) = year(current_date())
GROUP BY a.BRAND_ID
ORDER BY BRAND_TOTAL_AMOUNT DESC
LIMIT 1;
```



4) Who (salesman) makes us earn more money?

```
SELECT S.EMPLOYEE_ID, CONCAT(S.FIRST_NAME  ,COALESCE(S.MIDDLE_NAME,' '), S.LAST_NAME) AS EMP_NAME,
SUM(T.TOTAL_ORDER_AMOUNT) AS AMOUNT_SOLD_BY_EMPLOYEE
FROM TICKET T INNER JOIN SALESPERSON S
ON (T.EMPLOYEE_ID = S.EMPLOYEE_ID)
GROUP BY T.EMPLOYEE_ID
ORDER BY SUM(T.TOTAL_ORDER_AMOUNT) DESC
LIMIT 1;
```



05 – OPEN QUESTION

5) Which is the best selling store based on revenue?

```
SELECT S.STORE_ID,SUM(T.TOTAL_ORDER_AMOUNT) AS STORE_TOTAL_AMOUNT,
A.ADDRESS_NAME, A.ADDRESS_NUMBER, A.ZIPCODE_ID, A.COUNTRY
FROM STORE S INNER JOIN LOCATION L
ON S.LOCATION_ID = L.LOCATION_ID
INNER JOIN ADDRESS A
ON L.ADDRESS_ID = A.ADDRESS_ID
INNER JOIN TICKET T
ON S.STORE_ID = T.STORE_ID
GROUP BY T.STORE_ID
ORDER BY SUM(T.TOTAL_ORDER_AMOUNT) DESC
LIMIT 1;
```

```
49 •  SELECT S.STORE_ID,SUM(T.TOTAL_ORDER_AMOUNT) AS STORE_TOTAL_AMOUNT,
50    A.ADDRESS_NAME, A.ADDRESS_NUMBER, A.ZIPCODE_ID, A.COUNTRY
51    FROM STORE S INNER JOIN LOCATION L
52    ON S.LOCATION_ID = L.LOCATION_ID
53    INNER JOIN ADDRESS A
54    ON L.ADDRESS_ID = A.ADDRESS_ID
55     INNER JOIN TICKET T
56     ON S.STORE_ID = T.STORE_ID
57    GROUP BY T.STORE_ID
58    ORDER BY SUM(T.TOTAL_ORDER_AMOUNT) DESC
59    LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

STORE_ID	STORE_TOTAL_AMOUNT	ADDRESS_NAME	ADDRESS_NUMBER	ZIPCODE_ID	COUNTRY
582	4850.89	Urbita Kalea	25	28007	Spain