IE School of Human Sciences & Technology

Master's in Big Data & Business Analytics

Course Name:

SQL BASED DATA ARCHITECTURES

Supervised by:

Prof. CARLOS ALEJANDRO ANG

Submitted by:

MANUEL MARINA HERRERA

TABLE OF CONTENTS

Business Description

We are *la mezquita dorada*, recipient of Spain's award as the best kebabist company for 4 consecutive years and ranked 3rd as best fast food company in Spain behind TGB and Tierra Burrito Bar. We entered the fast food industry with a visionary approach to redefine the kebab landscape. Kebabs are seen as good cheap fast food made on the spot in front of the customer, yet sometimes people don't trust the owners or the quality of the food. The first differentiating factor with traditional kebabists is our usage of new technologies to optimize our operations, monitor the quality of our ingredients, anticipate and forecast demand, plan future purchases as well as where to open our new kebab places to maximize visibility and business opportunities. Secondly, we possess a strong brand image and very good reputation among the fast food industry because we make delicious and healthy kebabs. We make customers feel safe and unique with our nicely furnished restaurants, our strict respect of European phytosanitary norms and because we try to create a strong connection with our customers in order to solve their problems. The secret of our close bond with our customer base stems from our energetic and genuine presence on various social media platforms and our very easy to use online ordering system.

Data Definition Language (DDL) SQL

Table Creation + Primary keys

Ticket Data Model

Ticket Table

```
CREATE TABLE TICKET_TABLE (
    TICKET_ID BIGINT NOT NULL AUTO_INCREMENT,
    TIMEPLACE TIMESTAMP,
    STORE_ID SMALLINT,
    EMPLOYEE_ID SMALLINT,
    ORDER_NUMBER SMALLINT,
    TOTAL_PRODUCT DECIMAL,
    TAX DECIMAL,
    TOTAL_ORDER DECIMAL,
    CURRENCY VARCHAR(3),
    CC_PAYMENT_ID BIGINT(20),
    PRIMARY KEY (TICKET_ID)
);
```

Ticket Item Table

```
CREATE TABLE TICKET_ITEM_TABLE (
    TICKET_ID BIGINT,
    NUMSEQ SMALLINT(10) NOT NULL AUTO_INCREMENT,
    PRODUCT_ID BIGINT,
    QUANTITY SMALLINT,
    CURRENCY VARCHAR(10),
    PRICE DECIMAL,
    TAX_AMOUNT DECIMAL,
    PRIMARY KEY (NUMSEQ)
);
```

Order Data Model

Order Number Table

```
CREATE TABLE ORDER_NUMBER_TABLE (
    ORDER_NUMBER SMALLINT NOT NULL AUTO_INCREMENT,
    ORDER_TYPE SMALLINT,
    DESCRIP VARCHAR(15),
    PLATFORM_ID SMALLINT,
    PRIMARY KEY (ORDER_NUMBER)
);
```

Platform Table

```
CREATE TABLE PLATFORM_TABLE (
    PLATFORM_ID SMALLINT NOT NULL AUTO_INCREMENT,
    PLATFORM_TYPE SMALLINT,
    DESCRIP VARCHAR(20),
    COMPANY_ID SMALLINT,
    PRIMARY KEY (PLATFORM_ID)
);
```

Delivery Company Table

```
CREATE TABLE DELIVERY_COMPANY_TABLE (
    COMPANY_ID SMALLINT NOT NULL AUTO_INCREMENT,
    DELIVERY_TYPE_ID SMALLINT,
    COMPANY_NAME CHAR(20),
    PRIMARY KEY (COMPANY_ID)
);
```

Payment Data Model

Currency Table

```
CREATE TABLE CURRENCY_TABLE (
    CURRENCY_ID VARCHAR(40) NOT NULL,
    CURRENCY_CODE SMALLINT,
    DESCRIPT VARCHAR(40),
    PRIMARY KEY (CURRENCY_ID)
);
```

CC Payment Table

```
CREATE TABLE CC_PAYMENT_TABLE (
    CC_PAYMENT_ID BIGINT NOT NULL,
    CURRENCY_ID CHAR(3),
    EXPECTED_AMOUNT DECIMAL,
    APPROVING_AMOUNT DECIMAL,
    APPROVED_AMOUNT DECIMAL,
    CC_PAYMENT_STATE CHAR,
    TIME_CREATED DATE,
    TIME_UPDATED DATE,
    TIME_EXPIRED DATE NULL,
    PRIMARY KEY (CC_PAYMENT_ID)
);
```

CC Payment State Table

```
CREATE TABLE CC_PAYMENT_STATE_TABLE (
    CC_STATE CHAR NOT NULL,
    DESCRIPT VARCHAR(40),
    PRIMARY KEY (CC_STATE)
);
```

CC Payment Card Table

```
CREATE TABLE CC_PAYMENT_CARD_TABLE (
    CC_PAYMENT_SEQUENCE BIGINT NOT NULL AUTO_INCREMENT,
    CC_PAYMENT_ID BIGINT ,
    PAYMENT_TYPE VARCHAR(30),
    CARD_NUMBER VARCHAR(64),
    BANK_NAME VARCHAR(64) NOT NULL,
    CC_EXP_DATE TIMESTAMP,
    CC_ENTRY_METHOD VARCHAR(20),
    PRIMARY KEY (CC_PAYMENT_SEQUENCE)
);
```

CC Payment Type Table	CC Entry Method Table
`CREATE TABLE CC_PAYMENT_TYPE_TABLE( CC_TYPE CHAR(2) NOT NULL, DESCRIPT VARCHAR(40), PRIMARY KEY (CC_TYPE) );`	`CREATE TABLE CC_ENTRY_METHOD_TABLE ( CC_METHOD CHAR(1) NOT NULL, DESCRIPT VARCHAR(40), PRIMARY KEY (CC_METHOD) );`

Customer Data Model

Customer Table	Customer Loyalty Table
`CREATE TABLE CUSTOMER_TABLE ( CUSTOMER_ID BIGINT NOT NULL AUTO_INCREMENT, TICKET_ID BIGINT, CUSTOMER_LOCATION_ID BIGINT, CUSTOMER_LOYALTY_PROGRAM_ID BIGINT, PRIMARY KEY (CUSTOMER_ID) );`	`CREATE TABLE CUSTOMER_LOYALTY_TABLE ( CUSTOMER_LOYALTY_PROGRAM_ID BIGINT NOT NULL AUTO_INCREMENT, CUSTOMER_NAME CHAR(20), LAST_NAME CHAR(20), BIRTHDATE DATE, PHONE_NUMBER INT, NATIONAL_ID INT, LOY_PAYMENT_DETAILS_ID SMALLINT, PRIMARY KEY (CUSTOMER_LOYALTY_PROGRAM_ID) );`
Customer Loyalty Paying Details	**Customer Location Table**
`CREATE TABLE CUSTOMER_LOYALTY_PAYING_DETAILS ( LOY_PAYMENT_DETAILS_ID SMALLINT NOT NULL AUTO_INCREMENT, LOY_CARD_NUMBER BIGINT, LOY_BANK_NAME VARCHAR(30), LOY_CCEXPDATE TIMESTAMP, PRIMARY KEY (LOY_PAYMENT_DETAILS_ID) );`	`CREATE TABLE CUSTOMER_LOCATION_TABLE ( CUSTOMER_LOCATION_ID BIGINT NOT NULL, CITY_CODE_ID BIGINT, ZIP_ID BIGINT, PRIMARY KEY (CUSTOMER_LOCATION_ID) );`

City Table	Zip Table
`CREATE TABLE CITY_TABLE ( CITY_CODE_ID BIGINT NOT NULL, CITY_NAME CHAR(40), PRIMARY KEY (CITY_CODE_ID) );`	`CREATE TABLE ZIP_TABLE ( ZIP_ID BIGINT NOT NULL, CITY_ID BIGINT NOT NULL, ZIP_CODE SMALLINT, DESCRIPT CHAR(40), PRIMARY KEY (ZIP_ID, CITY_ID) );`

Store Data Model

Store Table

```
CREATE TABLE STORE_TABLE (
    STORE_ID SMALLINT NOT NULL AUTO_INCREMENT,
    LOCATION_ID BIGINT,
    PRIMARY KEY (STORE_ID)
);
```

Location Table

```
CREATE TABLE LOCATION_TABLE (
    LOCATION_ID BIGINT NOT NULL AUTO_INCREMENT,
    TYPE_ID CHAR(1),
    ADDRESS_ID INT,
    PRIMARY KEY (LOCATION_ID)
);
```

Location Type Table

```
CREATE TABLE LOCATION_TYPE_TABLE (
    TYPE_ID CHAR(1) NOT NULL,
    DESCRIP CHAR(80),
    PRIMARY KEY (TYPE_ID)
);
```

Employee Table

```
CREATE TABLE EMPLOYEE_TABLE (
    EMPLOYEE_ID SMALLINT NOT NULL AUTO_INCREMENT,
    EMPLOYEE_NAME CHAR(25),
    EMPLOYEE_SALARY INT,
    YEARS_OF_EXPERIENCE SMALLINT,
    AGE SMALLINT,
    POSITION_ID SMALLINT,
    STORE_ID SMALLINT,
    PRIMARY KEY (EMPLOYEE_ID)
);
```

Job Title Table

```
CREATE TABLE JOB_TITLE_TABLE (
    POSITION_ID SMALLINT NOT NULL AUTO_INCREMENT,
    DESCRIP CHAR(40),
    PRIMARY KEY (POSITION_ID)
);
```

Phone Table

```
CREATE TABLE PHONE_TABLE (
    LOCATION_ID BIGINT NOT NULL,
    NUMSEQ SMALLINT(10),
    DESCRIP BIGINT
);
```

Address Table

```
CREATE TABLE ADDRESS_TABLE (
    ADDRESS_ID INT NOT NULL AUTO_INCREMENT,
    TYPE_ID CHAR(1),
    ADDRESS_NAME VARCHAR(40),
    ADDRESS_NUMBER SMALLINT,
    ZIPCODE INT,
    CITY VARCHAR(20),
    PRIMARY KEY (ADDRESS_ID)
);
```

Address Type Table

```
CREATE TABLE ADDRESS_TYPE_TABLE (
    TYPE_ID CHAR(1) NOT NULL,
    DESCRIP VARCHAR(40),
    PRIMARY KEY (TYPE_ID)
);
```

Zip Code Table

```
CREATE TABLE ZIPCODE_TABLE (
    ZIPCODE INT NOT NULL,
    CITY VARCHAR(20),
    STATE VARCHAR(20),
    COUNTRY VARCHAR(20),
    PRIMARY KEY (ZIPCODE, CITY)
);
```

Operations Data Model

Menu Table

```
CREATE TABLE MENU_TABLE (
    BELONGING_ID BIGINT NOT NULL,
    PRODUCT_ID BIGINT,
    CATEGORY_ID BIGINT,
    QUANTITY SMALLINT,
    MENU_NAME VARCHAR(40),
    PRIMARY KEY (BELONGING_ID)
);
```

Product Table

```
CREATE TABLE PRODUCT_TABLE (
    PRODUCT_ID BIGINT NOT NULL,
    DESCRIPT VARCHAR(40),
    PRICE INT,
    PRIMARY KEY (PRODUCT_ID)
);
```

Category Table

```
CREATE TABLE CATEGORY_TABLE (
    CATEGORY_ID BIGINT NOT NULL,
    CATEGORY_NAME VARCHAR(20),
    DESCRIPT VARCHAR(128),
    PRIMARY KEY (CATEGORY_ID)
);
```

Foreign Keys

Ticket Data Model

```
ALTER TABLE TICKET_TABLE
    ADD FOREIGN KEY (STORE_ID)
    REFERENCES STORE_TABLE (STORE_ID);

ALTER TABLE TICKET_TABLE
    ADD FOREIGN KEY (EMPLOYEE_ID)
    REFERENCES EMPLOYEE_TABLE (EMPLOYEE_ID);

ALTER TABLE TICKET_TABLE
    ADD FOREIGN KEY (ORDER_NUMBER)
    REFERENCES ORDER_NUMBER_TABLE (ORDER_NUMBER);

ALTER TABLE TICKET_TABLE
    ADD FOREIGN KEY (CURRENCY)
    REFERENCES CURRENCY_TABLE (CURRENCY_ID);

ALTER TABLE TICKET_TABLE
    ADD FOREIGN KEY (CC_PAYMENT_ID)
    REFERENCES CC_PAYMENT_TABLE (CC_PAYMENT_ID);
```

```
ALTER TABLE TICKET_ITEM_TABLE
    ADD FOREIGN KEY (TICKET_ID)
    REFERENCES TICKET_TABLE (TICKET_ID);

ALTER TABLE TICKET_ITEM_TABLE
    ADD FOREIGN KEY (PRODUCT_ID)
    REFERENCES PRODUCT_TABLE (PRODUCT_ID);
```

Order Data Model

```
ALTER TABLE PLATFORM_TABLE
    ADD FOREIGN KEY (COMPANY_ID)
    REFERENCES DELIVERY_COMPANY_TABLE (COMPANY_ID);

ALTER TABLE ORDER_NUMBER_TABLE
    ADD FOREIGN KEY (PLATFORM_ID)
    REFERENCES PLATFORM_TABLE (PLATFORM_ID);
```

Payment Data Model

```
ALTER TABLE CC_PAYMENT_CARD_TABLE
    ADD FOREIGN KEY (PAYMENT_TYPE)
    REFERENCES CC_PAYMENT_TYPE_TABLE (CC_TYPE);

ALTER TABLE CC_PAYMENT_CARD_TABLE
    ADD FOREIGN KEY (CC_ENTRY_METHOD)
    REFERENCES CC_ENTRY_METHOD_TABLE (CC_METHOD);

ALTER TABLE CC_PAYMENT_TABLE
    ADD FOREIGN KEY (CURRENCY_ID)
    REFERENCES CURRENCY_TABLE (CURRENCY_ID);

ALTER TABLE CC_PAYMENT_TABLE
    ADD FOREIGN KEY (CC_PAYMENT_STATE)
    REFERENCES CC_PAYMENT_STATE_TABLE (CC_STATE);
```

```
ALTER TABLE CC_PAYMENT_CARD_TABLE
    ADD FOREIGN KEY (CC_PAYMENT_ID)
    REFERENCES CC_PAYMENT_TABLE (CC_PAYMENT_ID);
```

Customer Data Model

```
 ALTER TABLE CUSTOMER_LOCATION_TABLE
    ADD FOREIGN KEY (CITY_CODE_ID)
    REFERENCES CITY_TABLE (CITY_CODE_ID);

ALTER TABLE CUSTOMER_LOCATION_TABLE
    ADD FOREIGN KEY (ZIP_ID)
    REFERENCES ZIP_TABLE (ZIP_ID);

ALTER TABLE ZIP_TABLE
    ADD FOREIGN KEY (CITY_ID)
    REFERENCES CITY_TABLE (CITY_CODE_ID);

 ALTER TABLE CUSTOMER_LOYALTY_TABLE
    ADD FOREIGN KEY (LOY_PAYMENT_DETAILS_ID)
    REFERENCES CUSTOMER_LOYALTY_PAYING_DETAILS (LOY_PAYMENT_DETAILS_ID);

ALTER TABLE CUSTOMER_TABLE
    ADD FOREIGN KEY (TICKET_ID)
    REFERENCES TICKET_TABLE (TICKET_ID);

ALTER TABLE CUSTOMER_TABLE
    ADD FOREIGN KEY (CUSTOMER_LOCATION_ID)
    REFERENCES CUSTOMER_LOCATION_TABLE (CUSTOMER_LOCATION_ID);

ALTER TABLE CUSTOMER_TABLE
    ADD FOREIGN KEY (CUSTOMER_LOYALTY_PROGRAM_ID)
    REFERENCES CUSTOMER_LOYALTY_TABLE (CUSTOMER_LOYALTY_PROGRAM_ID);

ALTER TABLE CUSTOMER_TABLE
    ADD FOREIGN KEY (TICKET_ID)
    REFERENCES TICKET_TABLE (TICKET_ID);
```

Store Data Model

```
ALTER TABLE ADDRESS_TABLE
    ADD FOREIGN KEY (TYPE_ID)
    REFERENCES ADDRESS_TYPE_TABLE (TYPE_ID);

ALTER TABLE ADDRESS_TABLE
    ADD FOREIGN KEY (ZIPCODE)
    REFERENCES ZIPCODE_TABLE (ZIPCODE);

ALTER TABLE EMPLOYEE_TABLE
    ADD FOREIGN KEY (POSITION_ID)
    REFERENCES JOB_TITLE_TABLE (POSITION_ID);

ALTER TABLE EMPLOYEE_TABLE
    ADD FOREIGN KEY (STORE_ID)
    REFERENCES STORE_TABLE (STORE_ID);

ALTER TABLE LOCATION_TABLE
    ADD FOREIGN KEY (TYPE_ID)
    REFERENCES LOCATION_TYPE_TABLE (TYPE_ID);

ALTER TABLE LOCATION_TABLE
    ADD FOREIGN KEY (ADDRESS_ID)
    REFERENCES ADDRESS_TABLE (ADDRESS_ID);

ALTER TABLE STORE_TABLE
    ADD FOREIGN KEY (LOCATION_ID)
    REFERENCES LOCATION_TABLE (LOCATION_ID);
```

```
ALTER TABLE PHONE_TABLE
    ADD FOREIGN KEY (LOCATION_ID)
    REFERENCES LOCATION_TABLE (LOCATION_ID);
```

Operations Data Model

```
ALTER TABLE MENU_TABLE
    ADD FOREIGN KEY (PRODUCT_ID)
    REFERENCES PRODUCT_TABLE (PRODUCT_ID);

ALTER TABLE MENU_TABLE
    ADD FOREIGN KEY (CATEGORY_ID)
    REFERENCES CATEGORY_TABLE (CATEGORY_ID);

ALTER TABLE MENU_TABLE
    ADD FOREIGN KEY (PRODUCT_ID)
    REFERENCES PRODUCT_TABLE (PRODUCT_ID);
```

Data Manipulation Language (DML) SQL

Data Insert

Ticket Data Model

Ticket Table

```
INSERT INTO TICKET_TABLE
(TICKET_ID, TIMEPLACE, STORE_ID, EMPLOYEE_ID, ORDER_NUMBER, TOTAL_PRODUCT, TAX , TOTAL_ORDER, CURRENCY , CC_PAYMENT_ID)
VALUES
(80000, ('2021-01-12 19:10:05'), 01, 1002, 2000, 4.5, 0.945, 5.445, 'EUR', 000220),
(80001, ('2021-01-12 19:22:05'), 01, 1002, 2001, 5, 1.05, 6.05, 'EUR', 000221),
(80002, ('2021-03-12 12:10:05'), 02, 2001, 2002, 10, 2.1, 12.1, 'EUR', 000222),
(80003, ('2021-04-12 20:10:05'), 01, 1002, 2003, 5.5, 1.155, 6.655, 'EUR', 000223),
(80004, ('2021-06-12 14:36:05'), 01, 1002, 2004, 2.5, 0.525, 3.025, 'EUR', 000224),
(80005, ('2021-05-12 18:10:05'), 01, 1002, 2005, 5.5, 1.155, 6.655, 'EUR', 000225),
(80006, ('2021-06-12 19:05:05'), 01, 1002, 2006, 6, 1.26, 7.26, 'EUR', 000226),
(80007, ('2021-03-12 17:14:05'), 02, 2001, 2007, 11, 2.31, 13.31, 'EUR', 000227),
(80008, ('2021-11-12 22:01:05'), 02, 2002, 2008, 6.5, 1.365, 7.865, 'EUR', 000228),
(80009, ('2021-11-12 15:17:05'), 02, 2002, 2009, 6.5, 1.365, 7.865, 'EUR', 000229),
(80010, ('2021-11-12 18:16:05'), 02, 2002, 2010, 6.5, 1.365, 7.865, 'EUR', 000230),
(80011, ('2021-11-12 13:50:05'), 02, 2002, 2011, 6.5, 1.365, 7.865, 'EUR', 000231),
(80012, ('2021-11-12 12:46:05'), 02, 2002, 2012, 6.5, 1.365, 7.865, 'EUR', 000232),
(80013, ('2021-11-12 20:10:05'), 02, 2002, 2013, 6.5, 1.365, 7.865, 'EUR', 000233),
(80014, ('2021-11-12 18:23:05'), 02, 2002, 2014, 6.5, 1.365, 7.865, 'EUR', 000234),
(80015, ('2021-11-12 23:56:05'), 02, 2001, 2014, 6.5, 1.365, 7.865, 'EUR', 000235);
```

Ticket Item Table

```
INSERT INTO TICKET_ITEM_TABLE
(TICKET_ID, NUMSEQ, PRODUCT_ID, QUANTITY, CURRENCY, PRICE, TAX_AMOUNT)
VALUES
(80000, 1, 5, 1, 'EUR', 4.5, 0.945),
(80001, 2, 2, 1, 'EUR', 5, 1.05),
(80002, 3, 2, 2,  'EUR', 10, 2.1),
(80003, 4, 1, 3, 'EUR', 5.5, 1.155),
(80004, 5, 9, 1, 'EUR', 1, 0.21),
(80004, 6, 11, 1, 'EUR', 1.5, 0.315),
(80005, 7, 1, 1, 'EUR', 5.5, 1.155),
(80006, 8, 8, 1, 'EUR', 6, 1.26),
(80007, 9, 9, 2, 'EUR', 2, 0.42),
(80007, 10, 5, 2, 'EUR', 9, 1.89),
(80008, 11, 3, 1, 'EUR', 6.5, 1.365),
(80009, 12, 3, 1, 'EUR', 6.5, 1.365),
(80010, 13, 3, 1, 'EUR', 6.5, 1.365),
(80011, 14, 3, 1, 'EUR', 6.5, 1.365),
(80012, 15, 3, 1, 'EUR', 6.5, 1.365),
(80013, 16, 3, 1, 'EUR', 6.5, 1.365),
(80014, 17, 3, 1, 'EUR', 6.5, 1.365),
(80015, 18, 3, 1, 'EUR', 6.5, 1.365);
```

Order Data Model

Order Number Table

```
INSERT INTO ORDER_NUMBER_TABLE
(ORDER_NUMBER , ORDER_TYPE , DESCRIP , PLATFORM_ID )
VALUES
(2000, 0, "RESTAURANT", 3888),
(2001, 1, "TAKEAWAY", 3889),
(2002, 0, "RESTAURANT", 3890),
(2003, 0, "RESTAURANT", 3891),
(2004, 0, "RESTAURANT", 3892),
(2005, 1, "TAKEAWAY", 3893),
(2006, 0, "RESTAURANT", 3894),
(2007, 0, "RESTAURANT", 3895),
(2008, 1, "RESTAURANT", 3896),
(2009, 1, "RESTAURANT", 3897),
(2010, 1, "TAKEAWAY", 3898),
(2011, 1, "TAKEAWAY", 3899),
(2012, 1, "TAKEAWAY", 3900),
(2013, 1, "RESTAURANT", 3901),
(2014, 1, "TAKEAWAY", 3902),
(2015, 1, "TAKEAWAY", 3903);
```

Platform Table

```
INSERT INTO PLATFORM_TABLE
(PLATFORM_ID , PLATFORM_TYPE , DESCRIP , COMPANY_ID )
VALUES
(3888, 0, "localconsumption", 4444),
(3889, 2, "app", 4445),
(3890, 0, "localconsumption", 4446),
(3891,  0, "localconsumption", 4447),
( 3892, 1, "webapp", 4448),
(3893, 3, "phonecall", 4449),
(3894, 0, "localconsumption", 4450),
(3895, 1, "webapp", 4451),
(3896, 4, "externalapp", 4452),
(3897, 4, "externalapp", 4453),
(3898, 4, "externalapp", 4454),
(3899, 4, "externalapp", 4455),
(3900, 4, "externalapp", 4456),
(3901, 4, "externalapp", 4457),
(3902,4, "externalapp", 4458),
(3903, 4, "externalapp", 4459);
```

Delivery Company Table

```
INSERT INTO DELIVERY_COMPANY_TABLE
(COMPANY_ID , DELIVERY_TYPE_ID, COMPANY_NAME)
VALUES
(4444, 0, " "),
(4445, 1, "UBER"),
(4446, 0, " "),
(4447, 0, " "),
(4448, 1, "UBER"),
(4449, 0, " "),
(4450, 0, " "),
(4451, 1, "UBER"),
(4452, 2, "GLOVO"),
(4453, 2, "GLOVO"),
(4454, 2, "GLOVO"),
(4455, 2, "GLOVO"),
(4456, 2, "GLOVO"),
(4457, 2, "GLOVO"),
(4458, 2, "GLOVO"),
(4459, 2, "GLOVO");
```

Payment Data Model

Currency Table

```
INSERT INTO CURRENCY_TABLE
(CURRENCY_ID, CURRENCY_CODE , DESCRIPT)
VALUES
("EUR", 978, "EUROS"),
("USD", 840, "UNITED STATES DOLLAR");
```

CC Payment Table

```
INSERT INTO CC_PAYMENT_TABLE
(CC_PAYMENT_ID, CURRENCY_ID, EXPECTED_AMOUNT, APPROVING_AMOUNT, APPROVED_AMOUNT, CC_PAYMENT_STATE, TIME_CREATED, TIME_UPDATED, TIME_EXPIRED)
VALUES
(000220, "EUR", 5.445, 5.445, 5.445, 2, ("2021-12-02 18:56:34"), ("2021-12-02 18:56:44"), null),
(000221, "EUR", 6.05, 6.05, 6.05, 2, ("2021-12-01 20:23:06"), ("2021-12-01 20:23:16"), null),
(000222, "EUR", 12.1, 12.1, 12.1, 2, ("2021-12-03 22:33:32"), ("2021-12-03 22:33:42"), null),
(000223, "EUR", 6.655, 6.655, 6.55, 2, ("2021-12-04 12:45:55"), ("2021-12-04 12:46:05"), null),
(000224, "EUR", 3.025, 3.025, 3.025, 2, ("2021-12-06 13:10:31"), ("2021-12-06 13:10:41"), null),
(000225, "EUR",  6.655, 6.655, 6.655, 2, ("2021-12-05 14:11:01"), ("2021-12-05 14:11:11"), null),
(000226, "EUR", 7.26, 7.26, 7.26, 2, ("2021-12-06 12:19:31"), ("2021-12-06 12:19:41"), null),
(000227, "EUR", 12.31, 13.31, 13.31, 2, ("2021-12-03 20:56:22"), ("2021-12-03 20:56:32"), null),
(000228, "EUR", 7.865, 7.865, 7.865, 2, ("2021-12-11 22:33:41"), ("2021-12-11 22:33:51"), null),
(000229, "EUR", 7.865, 7.865, 7.865, 2, ("2021-12-11 22:34:28"), ("2021-12-11 22:34:38"), null),
(000230, "EUR", 7.865, 7.865, 7.865, 2, ("2021-12-11 22:35:17"), ("2021-12-11 22:35:27"), null),
(000231, "EUR", 7.865, 7.865, 7.865, 2, ("2021-12-11 22:36:20"), ("2021-12-11 22:36:30"), null),
(000232, "EUR", 7.865, 7.865, 7.865, 2, ("2021-12-11 22:37:50"), ("2021-12-11 22:38:00"), null),
(000233, "EUR", 7.865, 7.865, 7.865, 2, ("2021-12-11 22:38:48"), ("2021-12-11 22:38:58"), null),
(000234, "EUR", 7.865, 7.865, 7.865, 2, ("2021-12-11 22:40:00"), ("2021-12-11 22:40:10"), null),
(000235, "EUR", 7.865, 7.865, 7.865, 2, ("2021-12-11 22:49:06"), ("2021-12-11 22:49:16"), null);
```

CC Payment State Table

```
INSERT INTO CC_PAYMENT_STATE_TABLE
(CC_STATE, DESCRIPT)
VALUES
(0, "NEW"),
(1, "APPROVING"),
(2, "APPROVED"),
(3, "FAILED"),
(4, "CANCELLED"),
(5, "EXPIRED");
```

CC Payment Card Table

```
INSERT INTO CC_PAYMENT_CARD_TABLE
(CC_PAYMENT_SEQUENCE, CC_PAYMENT_ID, PAYMENT_TYPE, CARD_NUMBER, BANK_NAME, CC_EXP_DATE, CC_ENTRY_METHOD)
VALUES
(01, 000220, 'MC', 8838883877261121, 'BBVA', ('2028-08-08 00:00:00'), '0'),
(02, 000221, 'VS', 4603321042869504, 'BBVA', ('2023-03-01 00:00:00'), '0'),
(03, 000222, 'AE', "4603321042869504', 'BBVA', ('2023-08-01 00:00:00'), '0'),
(04, 000223, 'MC', '8837998901928837', 'BANKIA', ('2029-09-09 00:00:002'), '1'),
(05, 000224, 'MC', '8738928374661122", 'REVOLUT', ("2022-06-03 00:00:00"), '0'),
(06, 000225, 'VS', "0019827673619829", 'BBVA', ('2022-01-01 00:00:00'), '2'),
(07, 000226, "VS", "6132948009778767", "BBVA", ("2023-09-30 00:00:00"), "1"),
(08, 000227, "VS", "5637463519182938", "SANTANDER", ("2025-08-16 00:00:00"), "2"),
(09, 000228, "VS", "5637463519182938", "SANTANDER", ("2025-08-16 00:00:00"), "2"),
(10,000229, "VS", "5637463519182938", "SANTANDER", ("2025-08-16 00:00:00"), "2"),
(11, 000230, "VS", "5637463519182938", "SANTANDER", ("2025-08-16 00:00:00"), "2"),
(12, 000231, "AE", "5637463519182938", "SANTANDER", ("2025-08-16 00:00:00"), "2"),
(13, 000232, "VS", "5637463519182938", "SANTANDER", ("2025-08-16 00:00:00"), "2"),
(14, 000233, "VS", "5637463519182938", "SANTANDER", ("2025-08-16 00:00:00"), "2"),
(15, 000234, "VS", "5637463519182938", "SANTANDER", ("2025-08-16 00:00:00"), "2"),
(16, 000235, "AE", "8837172676561121", "BANKIA", ("2024-04-04 00:00:00"), "1");
```

CC Payment Type Table

```
INSERT INTO CC_PAYMENT_TYPE_TABLE
(CC_TYPE, DESCRIPT)
VALUES
("VS", "VISA"),
("MC", "MASTERCARD"),
("AE", "AMERICAN_EXPRESS");
```

CC Entry Method

```
INSERT INTO CC_ENTRY_METHOD_TABLE
(CC_METHOD, DESCRIPT)
VALUES
(0, "CONTRACT"),
(1, "INSERT"),
(2, "ONLINE");
```

Customer Data Model

Customer Table

```
INSERT INTO CUSTOMER_TABLE
(CUSTOMER_ID, TICKET_ID, CUSTOMER_LOYALTY_PROGRAM_ID)
VALUES
(6001, 80000, NULL),
(6002, 80001, 9999),
(6003, 80002, 9999),
(6004, 80003, NULL),
(6005, 80004, NULL),
(6006, 80005, NULL),
(6007, 80006, 9996),
(6008, 80007, 9998),
(6009, 80008, 9998),
(6010, 80009, 9998),
(6011, 80010, 9998),
(6012, 80011, 9998),
(6013, 80012, 9998),
(6014, 80013, 9998),
(6015, 80014, NULL);
```

Customer Location Table

```
INSERT INTO CUSTOMER_LOCATION_TABLE
(CUSTOMER_LOCATION_ID, CITY_CODE_ID, ZIP_ID)
VALUES
(000000001, 00000001, 00000001),
(000000002, 00000001, 00000002),
(000000003, 00000001, 00000003),
(000000004, 00000001, 00000004),
(000000005, 00000001, 00000005),
(000000006, 00000001, 00000006),
(000000007, 00000001, 00000007),
(000000008, 00000001, 00000008),
(000000009, 00000002, 00000001),
(000000010, 00000002, 00000002),
(000000011, 00000002, 00000003),
(000000012, 00000002, 00000004),
(000000013, 00000002, 00000005),
(000000014, 00000002, 00000006),
(000000015, 00000002, 00000007);
```

City Table

```
INSERT INTO CITY_TABLE
(CITY_CODE_ID, CITY_NAME)
VALUES
(00000001, "MADRID"),
(00000002, "MOSTOLES"),
(00000003, "ALCORCON");
```

Zip Table

```
INSERT INTO ZIP_TABLE
(CITY_ID, ZIP_ID, ZIP_CODE, DESCRIPT)
VALUES
(00000001, 00000001, 28001, "CENTRO"),
(00000001, 00000002, 28002, "CHAMBERI"),
(00000001, 00000003, 28003, "SALAMANCA"),
(00000001, 00000004, 28004, "MONCLOA"),
(00000001, 00000005, 28005, "EMBAJADORES"),
(00000001, 00000006, 28006, "LAVAPIES"),
(00000001, 00000007, 28007, "PUERTA DE TOLEDO"),
(00000001, 00000008, 28008, "TETUAN"),
(00000002, 00000001, 12001, "MOSTOLES-EL SOTO"),
(00000002, 00000002, 12002, "MOSTOLES CENTRO"),
(00000002, 00000003, 12003, "LAS RETAMAS"),
(00000002, 00000004, 12004, "BOSQUEJO"),
(00000002, 00000005, 12005, "EL MIRADOR"),
(00000002, 00000006, 12006, "TRES AGUAS"),
(00000002, 00000007, 12006, "ORELLANA NORTE");
```

Customer Loyalty Table

```
INSERT INTO CUSTOMER_LOYALTY_TABLE
(CUSTOMER_LOYALTY_PROGRAM_ID , CUSTOMER_NAME , BIRTHDATE, PHONE_NUMBER , NATIONAL_ID , LOY_PAYMENT_DETAILS_ID)
VALUES
(9999, "THOMAS", 04/08/1998, 676763545, 976329837, 7771),
(9998, "KIRIKOU", 03/09/1987, 687611172, 289389183, 7772),
(9997, "RAQUEL", 26/09/1995, 630729768, 183918375, 7773),
(9996, "AITOR MENTAH", 01/01/2000, 931830281, 493849321, 7774);
```

Customer Loyalty Paying Details

```
INSERT INTO CUSTOMER_LOYALTY_PAYING_DETAILS
(LOY_PAYMENT_DETAILS_ID, LOY_CARD_NUMBER , LOY_BANK_NAME, LOY_CCEXPDATE)
VALUES
(7771, 4603321042869504, "BBVA", "2023-08-01 00:00:00"),
(7772, 5637463519182938, "SANTANDER", "2025-08-16 00:00:00"),
(7773, 9830343312939485, "BANKIA", "2022-08-31 00:00:00"),
(7774, 6132948009778767, "BBVA", "2023-09-30 00:00:00");
```

Store Data Model

Store Table

```
INSERT INTO STORE_TABLE
(STORE_ID, LOCATION_ID)
VALUES
(01, 10),
(02, 20);
```

Location Table

```
INSERT INTO LOCATION_TABLE
(LOCATION_ID, TYPE_ID , ADDRESS_ID)
VALUES
(10, "K", 1111),
(20, "K", 1112);
```

Location Type Table

```
INSERT INTO LOCATION_TYPE_TABLE
(TYPE_ID, DESCRIP )
VALUES
("K", "KEBAB_RESTAURANT"),
("H", "HEADQUARTERS");
```

Employee Table

```
INSERT INTO EMPLOYEE_TABLE
(EMPLOYEE_ID, EMPLOYEE_NAME , EMPLOYEE_SALARY, YEARS_OF_EXPERIENCE, AGE, POSITION_ID, STORE_ID )
VALUES
(1001, "Karim", 10000, 7, 28, 01, 01),
(1002, "Boniface", 1500, 1, 24, 02, 01),
(2002, "Shanon", 1500, 2, 23, 02, 02),
(2001, "Gema", 10000, 5, 23, 01, 02);
```

Job Title Table

```
INSERT INTO JOB_TITLE_TABLE
(POSITION_ID, DESCRIP)
VALUES
(01, "MANAGER"),
(02, "KEBABIST"),
(03, "INTERN");
```

Phone Table

```
INSERT INTO PHONE_TABLE
(LOCATION_ID, NUMSEQ, DESCRIP)
VALUES
(10, 1, "630729729"),
(20, 2,"654725288");
```

Address Table

```
INSERT INTO ADDRESS_TABLE
(ADDRESS_ID, TYPE_ID , ADDRESS_NAME , ADDRESS_NUMBER, ZIPCODE, CITY)
VALUES
(1111, "K", "CALLE_RAIMUNDO_FERNANDEZ", 9, 28003, "MADRID"),
(1112, "K", "CALLE_MARIA_MOLINA", 32, 28006, "MADRID");
```

Address Type Table

```
INSERT INTO ADDRESS_TYPE_TABLE
(TYPE_ID, DESCRIP)
VALUES
("K", "CALLE"),
("A", "AVENIDA");
```

Zip Code

```
INSERT INTO ZIPCODE_TABLE
(ZIPCODE, CITY , STATE, COUNTRY)
VALUES
(28003, "MADRID", "MADRID", "SPAIN"),
(28006, "MADRID", "MADRID", "SPAIN");
```

Operations Data Model

Menu Table

```
INSERT INTO MENU_TABLE
(BELONGING_ID, PRODUCT_ID, CATEGORY_ID , MENU_NAME , QUANTITY)
VALUES
(000001, 1, 1, "DURUM_KEBAB", 1),
(000002, 1, 2, "REFRESCO", 1),
(000003, 1, 3, "PATATAS", 1),
(000004, 2, 1, "DONNER_KEBAB", 1),
(000005, 2, 2, "REFRESCO", 1),
(000006, 2, 3, "PATATAS", 1),
(000007, 3, 1, "DURUM_KEBAB_XXL", 1),
(000008, 3, 2, "REFRESCO", 1),
(000009, 3, 3, "PATATAS_XXL", 1),
(000010, 4, 1, "DONNER_KEBAB_XXL", 1),
(000011, 4, 2, "REFRESCO", 1),
(000012, 4, 3, "PATATAS_XXL", 1),
(000013, 5, 1, "DURUM_KEBAB", 1),
(000014, 6, 1, "DURUM_KEBAB_XXL", 1),
(000015, 7, 1, "DONNER_KEBAB_XXL", 1),
(000016, 8, 1, "DONNER_KEBAB_XXL", 1);
```

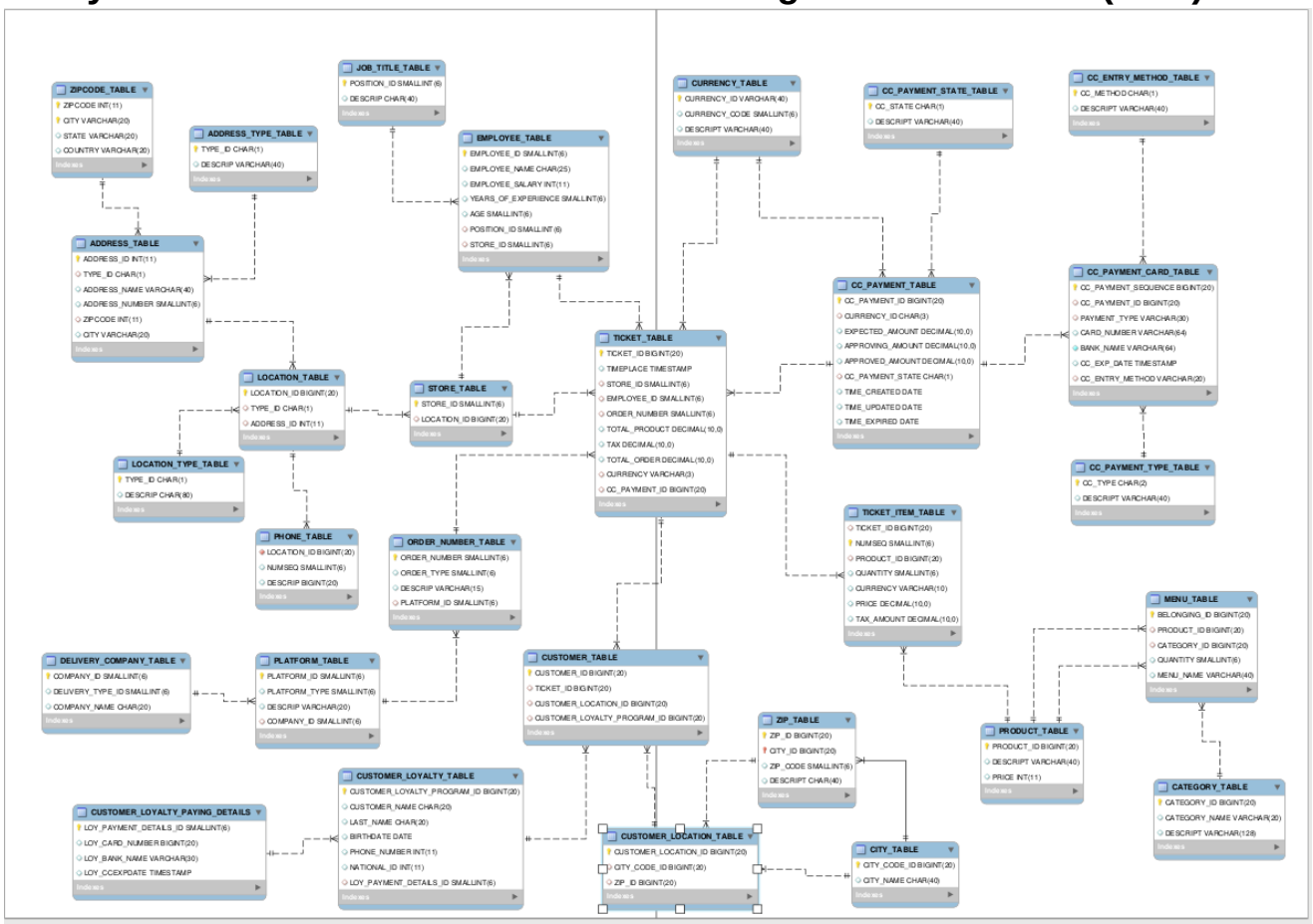
Product Table

```
INSERT INTO PRODUCT_TABLE
(PRODUCT_ID, DESCRIPT, PRICE)
VALUES
(1, "MENU_DURUM", 5.5),
(2, "MENU_DONNER", 5),
(3, "MENU_DURUM_XXL", 6.5),
(4, "MENU DONNER XXL", 6),
(5, "DURUM_KEBAB", 4.5),
(6, "DONNER_KEBAB", 4),
(7,  "DURUM_KEBAB_XXL", 5.5),
(8, "DONNER_KEBAB_XXL", 5),
(9, "REFRESCO", 1),
(10, "PATATAS", 1),
(11, "PATATAS_XXL", 1.5);
```

Category Table

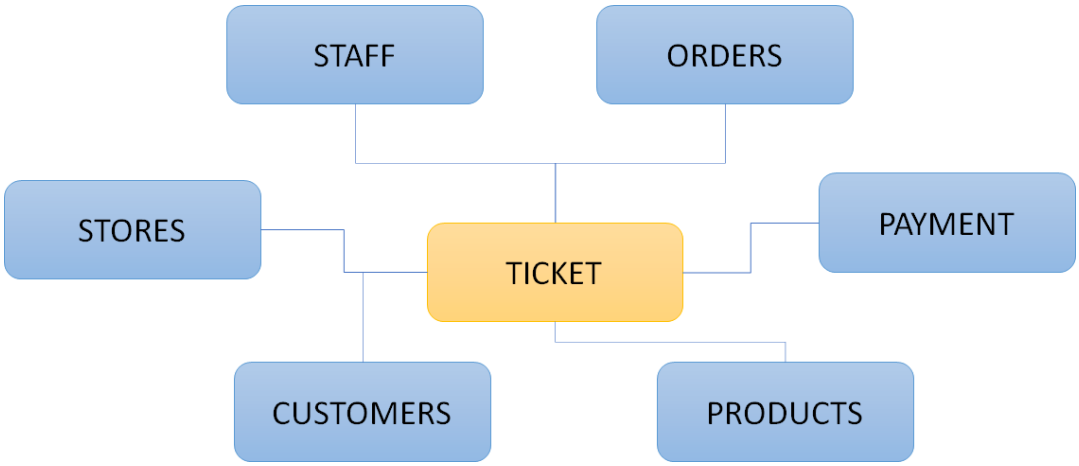
```
INSERT INTO CATEGORY_TABLE
(CATEGORY_ID, CATEGORY_NAME , DESCRIPT)
VALUES
(1, "KEBAB", "AMAZING_SANDWICHS"),
(2, "REFRESCO", "IMPORTANT_TO_REHYDRATE_THE_BODY"),
(3, "PATATAS", "DELICIOUS_POTATOES");
```

Entity Relation Diagram (ERD)



Explanation Of Model

The central entity of our model is the ticket, as our Store Model, Customer Model, Payments Model, Orders Model and Items/Menu model are connected to it.



The ticket provides information about the store in which the order was made and the employee that was attending the customer. The employee table linked to the ticket provides us with information about the employee, such as his store, age, experience and another table specifies the POSITION_ID of our worker. In the store table, we obtain information about the LOCATION_ID out of which we obtain data about the type of location and in another group of tables, the address, number, address type, city and zip code of the store location.

The ticket naturally provides information about what the customer ordered. This can be seen through the TICKET_ITEM table which is connected thanks to the TICKET_ID and NUMSEQ. We specify the PRODUCT_ID, quantity and prices. We created a product table to define the price and name of each PRODUCT_ID, out of which we created a MENU TABLE. This table contains all the items we sell and the different elements composing each PRODUCT_ID as some PRODUCT_IDs correspond to a menu with several items. Finally, we have a CATEGORY TABLE that categorizes the type of item we sell, whether it is a sandwich, drink or fries.

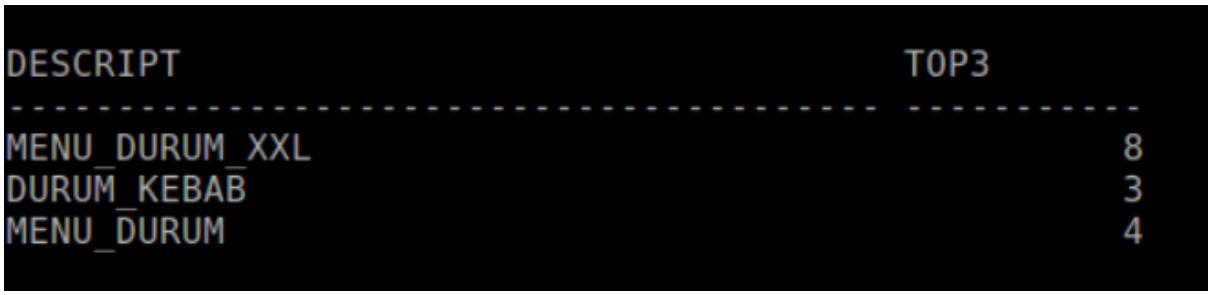
Additionally, we convey information about the order number and type of order, whether it is to takeaway or not. Another table gathers the means of ordering, whether it was face to face, with an app or a phone call for example. Lastly, our system also records in another table which delivery company was in charge of the order which is useful in order to identify who to contact if there was a problem (DELIVERY COMPANY TABLE).

The ticket also provides information about prices, the PAYMENT_ID and the currency of the payment which will be very useful to design our payment model. We develop a payment table that provides general information about the state of the payment, which is defined in the CC_PAYMENT_STATE TABLE, the times of the transactions, the amounts and currency in addition to the PAYMENT_ID. We connect a CREDIT_CARD TABLE to the PAYMENT TABLE that contains the payment details such as credit card number, payment type ID, expiration date and entry method. We later have an entry method and payment table method describing the values from the payment card table, kind of a look up table (ENTRY METHOD TABLE).

Finally, we connected a CUSTOMER_ID to the TICKET_ID. Since we are a business close to our customers, we also possess a customer loyalty program therefore, a customer LOYALTY_ID may eventually be connected to the CUSTOMER_ID. Logically, we have a customer LOYALTY TABLE where we store personal information of our customers in the loyalty program such as their names, family names, birth dates, phone numbers, national ID and an ID for their payment details. This ID is used to create a new table where we gather the financial information of our most loyal customers. It can be used to identify how frequently they trust us to provide them with the best kebab magic in Madrid.

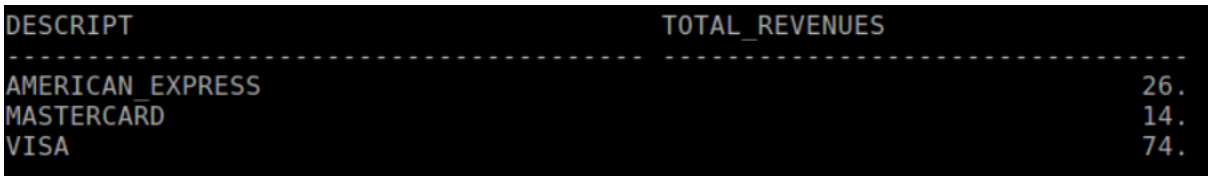
Professor's Questions

- Which are the customers top 3 favorite dishes?
    - SELECT DESCRIPT, COUNT(QUANTITY) AS TOP3
      FROM TICKET_TABLE A, TICKET_ITEM_TABLE B, PRODUCT_TABLE C
      WHERE A.TICKET_ID = B.TICKET_ID
      AND B.PRODUCT_ID = C.PRODUCT_ID
      GROUP BY DESCRIPT
      ORDER BY COUNT(QUANTITY)
      DESC LIMIT 3



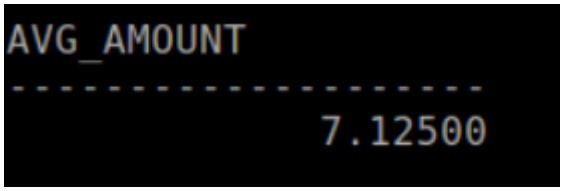
DESCRIPT	TOP3
MENU_DURUM_XXL	8
DURUM_KEBAB	3
MENU_DURUM	4

- Show the revenue per type of credit card?
    - SELECT DESCRIPT, SUM(TOTAL_ORDER) AS TOTAL_REVENUES
      FROM TICKET_TABLE A, CC_PAYMENT_TABLE B,
      CC_PAYMENT_CARD_TABLE C, CC_PAYMENT_TYPE_TABLE D
      WHERE A.CC_PAYMENT_ID = B.CC_PAYMENT_ID
      AND C.CC_PAYMENT_ID = B.CC_PAYMENT_ID
      AND PAYMENT_TYPE = D.CC_TYPE
      GROUP BY D.DESCRIPT



DESCRIPT	TOTAL_REVENUES
AMERICAN_EXPRESS	26.
MASTERCARD	14.
VISA	74.

- What is the average amount spent on each purchase?
    - SELECT CAST(AVG(TOTAL_ORDER) AS DECIMAL(18,3)) AS AVG_AMOUNT FROM TICKET_TABLE



```
AVG_AMOUNT
----------
   7.12500
```

- Which district/area does the customers usually come from?
    - SELECT E.CITY_NAME, D.ZIP_CODE D.DESCRIPT, COUNT(*) AS CUSTOMERS_PER_DISTRICT
      FROM TICKET_TABLE A, CUSTOMER_TABLE B, CUSTOMER_LOCATION_TABLE C, ZIP_TABLE D, CITY_TABLE E
      WHERE A.TICKET_ID = B.TICKET_ID
      AND B.CUSTOMER_LOCATION_ID = C.CUSTOMER_LOCATION_ID
      AND (C.CITY_CODE_ID = D.CITY_ID AND C.ZIP_ID = D.ZIP_ID)
      AND E.CITY_CODE_ID = D.CITY_ID
      GROUP BY D.DESCRIPT, E.CITY_NAME, D.ZIP_CODE

```
CITY_NAME                           ZIP_CODE DESCRIPT                           CUSTOMERS_PER_DISTRICT
----------------------------------- -------- ------------------------------------ ----------------------
MOSTOLES                               12002 MOSTOLES CENTRO                                       1
MADRID                                 28001 CENTRO                                                8
MADRID                                 28002 CHAMBERI                                              2
MADRID                                 28003 SALAMANCA                                             1
MADRID                                 28004 MONCLOA                                               2
MADRID                                 28005 EMBAJADORES                                           1
MADRID                                 28008 TETUAN                                                1
```

Group Question

- Who is the company's best customer, when born and how much money did he/she spend in our company?
    - SELECT CUSTOMER_NAME, LAST_NAME, BIRTHDATE, COUNT(*) AS TOTAL_ORDERS, SUM(TOTAL_ORDER) AS AMOUNT_SPENT
      FROM TICKET_TABLE A, CUSTOMER_TABLE B, CUSTOMER_LOYALTY_TABLE C
      WHERE A.TICKET_ID = B.TICKET_ID
      AND B.CUSTOMER_LOYALTY_PROGRAM_ID = C.CUSTOMER_LOYALTY_PROGRAM_ID
      GROUP BY CUSTOMER_NAME,LAST_NAME,BIRTHDATE
      ORDER BY COUNT(*) DESC
      LIMIT 1

```
CUSTOMER_NAME       LAST_NAME           BIRTHDATE  TOTAL_ORDERS AMOUNT_SPENT
------------------- ------------------- ---------- ------------ --------------------------------
KIRIKOU             SIANG               03/09/1987            8                            56.
```