**SQL Based Data Architectures.** SettingUp a Data Model for a Fictional Casino

1. Introduction

This report summarises our work on the “Casino” data model. In the following, we first provide a fundamental description of the data model across the four main sections it consists of. We then proceed to answer the four questions that came with the “Casino” case, plus an additional question we formulated ourselves, by showing the relevant SQL statements as well as the output the model yields for our fictional data. Finally, the annex will contain the DDL and DML.

1. Data Model Description

Our data model comprises a total of 25 entities. In principle, the model can be structured in four sections (underscored) containing the following entities:

* Casino & Location:

CASINO, LOCATION, LOCATION\_TYPE, ADDRESS, ADDRESS\_TYPE, ZIPCODE, EMPLOYEE, ROLES

* Game & Table:

CASINO\_TABLE, GAME, GAMETYPE

* Drinks:

DRINKS, TICKET\_LINE\_DRINK, TICKET\_DRINK, DRINK\_TYPES, BRANDS, UNIT\_MEASURES

* Player & Bet:

PLAYER, CASINO\_TRANSACTION, TRANSACTION\_TYPE, PAYMENT, PAYMENT\_TYPE, CURRENCY\_TYPE, STACK, BET

Next, we will explain the entities and relationships within each section and among them. To facilitate the visual understanding of the description, we will maintain the conventions of displaying all entities/tables in uppercase, e.g., CASINO, and all attributes italics, e.g. *casino\_id*.

* 1. Casino & Location

This section gives details about the casino, its employees and the locations. In terms of the EMPLOYEE table, which contains multiple descriptive attributes of each employee (*first\_name*, *last\_name*, *date\_hired*, etc.), has three relationships: (1) To the ROLES , (2) to CASINO and (3) to CASINO\_TABLE. The first is a one-to-many relationship with the ROLES table. ROLES is a lookup table containing all the different roles (ex: dealer, waitress) inside the casino and therefore one role can apply to many employees but each employee can have only one single role. The CASINO-EMPLOYEE relationship is a one-to-many relationship. One casino can have multiple employees working in it, but an employee can only work for a single casino at a given time. The EMPLOYEE table is later joined with CASINO\_TABLE which allows the user to identify which employees (dealers) are being assigned to each game table (one-to-many relationship).

On the other hand, the CASINO table is connected to the LOCATION in a one-to-many relationship because one location might have several casinos if the company were to build or buy a new casino at the same location. After that, the LOCATION is normalised by separating redundant values. We have an ADDRESS table (one-to-many with LOCATION), an ADDRESS TYPE with a one-to-many with ADDRESS (one address can only have one address type, but one address type can apply to many addresses) containing the values for whether the address is a street, avenue, etc. Also, we created a ZIPCODE table which contains a composite primary key made of the ZIPCODE and the COUNTRY which links to the ADDRESS table in order to comply with normalisation.

* 1. Game & Table

The game & table section acts as our main hub, as this is where all the casino games are played. Many of our tables join to CASINO\_TABLE; players place bets here, buy drinks and are served by employees of the casino. We are able to determine what game is played at what table and helps us differentiate between different casino activities in different games.

CASINO has a one-to-many relationship with CASINO\_TABLE. CASINO\_TABLE contains the primary key *table\_id* as well as the two foreign keys *casino\_id* and *game\_id*, respectively relating it to CASINO or GAME. Since certain games like poker are played at many tables while every table can host only one game type, GAME has a one-to-many relationship to CASINO\_TABLE. GAME further has the three descriptive attributes *game\_name*, *max\_players*, and *min\_buyin*. Lastly, it possesses the foreign key *game\_type*, establishing a one-to-many relationship to GAME\_TYPE (lookup table), as each game has exactly one game type but there are many game types in GAME.

* 1. Drink

Besides the different types of games, a casino can also profit from selling drinks. Therefore, we need to add a drinks sub-model to our casino data model. Every time a player orders a drink or a set of drinks, a ticket is generated.

The ticket generated from an order (*ticket\_id*) is a surrogate key and is stored on our first table TICKET\_DRINK as the primary key. Each occurrence in this table is described by several attributes such as the time placed, the identification of the player assigned to a ticket, table, employee, and price amounts (total price, total tax, and sum). Since a player can order more than one drink and to assure a normalized data model, we created a TICKET\_LINE\_DRINK table that specifies each ticket’s drink with a *numseq ­*– primary key composed by the *ticket\_id* (also foreign key that relates to the TICKET\_DRINK table) *and the numseq.* To describe each occurrence in this table, we use several attributes such as the drink itself, description, quantity, and price amounts. Since a *ticket\_id* at the TICKET\_DRINK table can belong to several occurrences in the TICKET\_LINE\_DRINKS due to the *numseq*, but a ticket\_id in this table can only belong to a ticket in the TICKET\_DRINK table, therefore we can conclude that these tables all have a one-to-many relationship.

To describe each drink, we created the DRINKS table that also has a one-to-many relationship with the TICKET\_LINE\_DRINK table since each a drink can belong to many tickets but must only contain one drink. This entity describes a drink by several attributes such as type, the size code, the size amount of the drink, the name of the drink, and the brand. The primary key of this table is the id of the drink and has the type of drink, code of the size, and brand as foreign keys that relate to the other 3 look-up tables that we need to create to ensure a normalized data model.

The DRINK\_TYPES table has all types of drinks such as beer, wine, vodkas and many other, and has *drink\_type* as primary key. This table has a one-to-many relationship with the DRINKS table since a drink can only have one type but a type can belong to many drinks. The BRANDS table contains all brands of beverages such as Barcardi, Mahou and many others. The primary key of this table is *brand\_id* and has a one-to-many relationship with DRINKS table since a drink can only belong to one brand but a brand can belong to many drinks. The UNIT\_MEASURES table has all the different size codes and descriptions such as “ml”, “milliliters”. This table has *size\_code* as primary key and it has also a one-to-many relationship with the DRINKS table since a drink can only have one unit measure but a unit measure can be in many drinks.

* 1. Player & Bet

The player and bet section plays an important role in understanding player activity and betting across all games. This helps the casino understand the amount of chips won and lost during games, what tables are most active and so on.

The table PLAYER contains *player\_id* as the primary key as well as descriptive attributes concerning the players’ first and last name, email, phone number, and passport number. PLAYER is directly related to three tables: CASINO\_TRANSACTION, BET and STACK. Since one player can cash in/out various times and can place multiple bets, PLAYER has a one-to-many relationship to the tables CASINO\_TRANSACTION and BET. Since a player’s stack of chips will vary in size across a casino visit but will keep the same ID to match it to a player, PLAYER has a one-to-one with STACK.

Besides *stack\_id* as its primary key, STACK contains *stack\_amount* as the monetary amount and *player\_id* as a foreign key. BET contains *bet\_id* as its primary key, *chips\_in* and *chips\_out* as the monetary amount put into or pulled out of a game, *start\_time* and *end\_time* as timestamps for each bet, and *player\_id* and *table\_id* as foreign keys linking the table to PLAYER and CASINO\_TABLE. BET relates one-to-many to CASINO\_TABLE since multiple bets can be placed at a time at one table.

CASINO\_TRANSACTION with *transaction\_id* as the primary key further contains *total\_amount* for the monetary amount a player converts into chips (or vice versa) at the cashier as well as the two foreign keys *player\_id* and *transaction\_type*, relating to the tables PLAYER and TRANSACTION\_TYPE. CASINO\_TRANSACTION links one-to-many to TRANSACTION\_TYPE (lookup table), since all transactions will each be executed by exactly one type while each type will be attributable to many transactions.

We created PAYMENT to maintain data about a player’s payment methods and their cash in/out activity. PAYMENT contains *payment\_id* as the primary key and *transaction\_id*, *payment\_type*, and *currency\_type* as its foreign keys. The latter relate PAYMENT to the lookup tables PAYMENT\_TYPE and CURRENCY\_TYPE. Therefore, PAYMENT comprises all possible combinations of payment methods and currencies a player can use to cash in/out. It stands in a one-to-one relationship with CASINO\_TRANSACTION, since each transaction is executed through exactly one of the mentioned combinations.

Overall, the data model enables the user to understand inner workings of players regarding their initial cash in at the casino, the drinks they ordered, the games they played, which tables they played at, the employees they interacted with, the amount they bet, the amounts they won/lost and finally their cash out. This seamless relationship between our 4 main model section also allows us to tackle the below questions among many more.

1. Case-Specific Questions
   1. Which Are the Top 3 Demanding Casino Games?

We count the number of bets per game in order to find the most demanding casino games

Select c.game\_name,

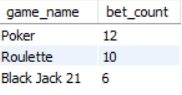
count(a.bet\_id) as bet\_count

from bet a

inner join casino\_table b

on a.table\_id = b.table\_id

inner join game c

on b.game\_id = c.game\_id

group by 1

order by 2 DESC

LIMIT 3;

Hence, the three most actively bet on, i.e. demanding, games are poker, roulette and black jack 21.

* 1. What Is the Average Number of Chips Per Game Type and Day?

We first needed to sum the total chips involved at a table and game per day in a subselect. We then calculate the mean of the result by game and day.

select x.game\_name,

x.day\_num,

x.month\_num,

avg(x.total\_chips\_table) as avg\_chips\_table

from

(select distinct b.game\_name,

a.table\_id,

day(start\_time) as day\_num,

month(start\_time) as month\_num,

sum(c.chips\_in) as total\_chips\_table

from casino\_table a

inner join game b

on a.game\_id =b.game\_id

inner join bet c

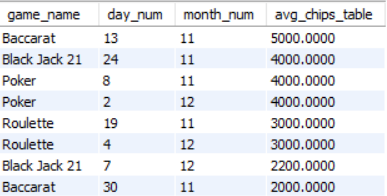
on a.table\_id = c.table\_id

group by 1,2,3,4) x

group by 1,2,3

order by 4 desc;

Here is a snapshot of some of the results:



* 1. Which Games Favour Purchasing Drinks?

We join together all of the drink tables and count at the most granular *drink\_id* level, grouped by *game\_name*:

select distinct b.game\_name,

count(e.drink\_id) as drink\_cnt

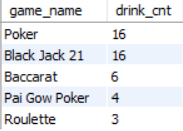
from casino\_table a

inner join game b

on a.game\_id =b.game\_id

inner join ticket\_drink c

on a.table\_id = c.table\_id

inner join ticket\_line\_drink d

on c.ticket\_id = d.ticket\_id

inner join drinks e

on d.drink\_id = e.drink\_id

group by 1

order by 2 DESC;

Poker and Black Jack 21 are the games that most favour purchasing drinks with 16 drinks at each.

* 1. What Is the Easiest Game and Table to Win Money?

We calculate the average *chips\_out* (game winnings) - average *chips\_in* (cost of playing), in order to determine which game and table are on average the easiest to win money at:

select distinct b.game\_name,

a.table\_id,

avg(c.chips\_out)-avg(c.chips\_in) as avg\_chips\_won

Graphical user interface, table

Description automatically generated with medium confidencefrom casino\_table a

inner join game b

on a.game\_id =b.game\_id

inner join bet c

on a.table\_id = c.table\_id

group by 1,2

order by 3 DESC;

The easiest game and table to win money is at Roulette at table 99

* 1. Our Own Question: Who Ordered the Same Drink Type the Most?

We have to utilize a series of subselects and subselect within a having clause in order to make this work.

We need to start with a subselect that counts number of drinks per player and drink type.

We then need to find the maximum number of drinks per player and all the drink types.

We use a having clause to isolate only the drink and player combination that had the highest number drinks.

Within the having clause we have another subselect that selects only the maximum number of drinks from our previous select of count of drinks per player & drink type.

We go through all these steps instead of using a LIMIT 1 to avoid the trap of missing out on scenarios where there might be more than a single player who ordered a particular drink the most. This was not the case here, as only John Lennon had a single drink type 3 times – beer.

select x.first\_name,

x.last\_name,

x.drink\_type,

max(x.drink\_cnt) as max\_drink\_cnt

from

(select distinct f.first\_name,

f.last\_name,

g.description as drink\_type,

count(e.drink\_id) as drink\_cnt

from casino\_table a

inner join game b

on a.game\_id =b.game\_id

inner join ticket\_drink c

on a.table\_id = c.table\_id

inner join ticket\_line\_drink d

on c.ticket\_id = d.ticket\_id

inner join drinks e

on d.drink\_id = e.drink\_id

inner join player f

on c.player\_id = f.player\_id

inner join drink\_types g

on e.drink\_type=g.drink\_type

group by 1,2,3

order by 2 DESC) x

group by 1,2,3

having max(x.drink\_cnt) = (select max(z.drink\_cnt) from

(select distinct f.first\_name,

f.last\_name,

g.description as drink\_type,

count(e.drink\_id) as drink\_cnt

from casino\_table a

inner join game b

on a.game\_id =b.game\_id

inner join ticket\_drink c

on a.table\_id = c.table\_id

inner join ticket\_line\_drink d

on c.ticket\_id = d.ticket\_id

inner join drinks e

on d.drink\_id = e.drink\_id

inner join player f

on c.player\_id = f.player\_id

inner join drink\_types g

on e.drink\_type=g.drink\_type

Graphical user interface, application

Description automatically generatedgroup by 1,2,3) z);

1. Annex
   1. Data Model DDL & DML

**DDL**

CREATE TABLE EMPLOYEE (

EMPLOYEE\_ID CHAR(9) NOT NULL,

FIRST\_NAME VARCHAR(50) NOT NULL,

LAST\_NAME VARCHAR(50) NOT NULL,

ROLE\_ID CHAR(3) NOT NULL,

DATE\_HIRED DATE NOT NULL,

DATE\_OF\_BIRTH DATE,

CASINO\_ID INT NOT NULL,

TABLE\_ID VARCHAR(3),

PRIMARY KEY (EMPLOYEE\_ID)

);

CREATE TABLE ROLES (

ROLE\_ID CHAR(3) NOT NULL,

ROLE\_NAME VARCHAR(50) NOT NULL,

PRIMARY KEY (ROLE\_ID)

);

CREATE TABLE CASINO (

CASINO\_ID INT NOT NULL,

CASINO\_NAME VARCHAR(50) NOT NULL,

LOCATION\_ID BIGINT NOT NULL,

PRIMARY KEY (CASINO\_ID)

);

CREATE TABLE LOCATION (

LOCATION\_ID BIGINT NOT NULL,

ADDRESS\_ID BIGINT NOT NULL,

LOCATION\_TYPE CHAR(2) NOT NULL,

PRIMARY KEY (LOCATION\_ID)

);

CREATE TABLE LOCATION\_TYPE (

LOCATION\_TYPE CHAR(2) NOT NULL,

DESCRIPTION VARCHAR(120) NOT NULL,

PRIMARY KEY(LOCATION\_TYPE)

);

CREATE TABLE ADDRESS (

ADDRESS\_ID BIGINT NOT NULL,

ADDRESS\_TYPE CHAR(2) NOT NULL,

ADDRESS\_NAME VARCHAR(50) NOT NULL,

ADDRESS\_NUM VARCHAR(10) NOT NULL,

ZIPCODE VARCHAR(20) NOT NULL,

COUNTRY VARCHAR (120) NOT NULL,

PRIMARY KEY (ADDRESS\_ID)

);

CREATE TABLE ADDRESS\_TYPE(

ADDRESS\_TYPE CHAR(2) NOT NULL,

DESCRIPTION VARCHAR(120) NOT NULL,

PRIMARY KEY(ADDRESS\_TYPE)

);

CREATE TABLE ZIPCODE (

ZIPCODE VARCHAR(20),

COUNTRY VARCHAR (120),

CITY VARCHAR(120),

STATE VARCHAR(120),

PRIMARY KEY (ZIPCODE, COUNTRY)

);

CREATE TABLE GAMETYPE

(GAME\_TYPE CHAR(1) NOT NULL,

DESCRIPTION VARCHAR(120) NOT NULL,

PRIMARY KEY (GAME\_TYPE));

CREATE TABLE GAME

(GAME\_ID CHAR(3) NOT NULL,

GAME\_TYPE CHAR(1) NOT NULL,

GAME\_NAME VARCHAR(100) NOT NULL,

MAX\_PLAYERS SMALLINT NOT NULL,

MIN\_BUYIN BIGINT,

PRIMARY KEY (GAME\_ID),

FOREIGN KEY (GAME\_TYPE) REFERENCES GAMETYPE(GAME\_TYPE));

CREATE TABLE CASINO\_TABLE

(TABLE\_ID VARCHAR(3) NOT NULL,

GAME\_ID CHAR(3) NOT NULL,

CASINO\_ID INT NOT NULL,

PRIMARY KEY (TABLE\_ID),

FOREIGN KEY (GAME\_ID) REFERENCES GAME(GAME\_ID),

FOREIGN KEY (CASINO\_ID) REFERENCES CASINO(CASINO\_ID));

CREATE TABLE TICKET\_DRINK (

TICKET\_ID BIGINT NOT NULL PRIMARY KEY,

TIMEPLACED TIMESTAMP NOT NULL,

PLAYER\_ID CHAR(9) NOT NULL,

TABLE\_ID VARCHAR(3) NOT NULL,

EMPLOYEE\_ID CHAR(9) NOT NULL,

TOTAL\_DRINK DECIMAL(20,2) NOT NULL,

TOTAL\_TAX DECIMAL(20,2) NOT NULL,

TOTAL\_ORDER DECIMAL (20,5) NOT NULL);

CREATE TABLE TICKET\_LINE\_DRINK (

/\*ASK PROF IF WE ONLY NEED $ AMOUNTS IN THIS TABLE OR IN BOTH\*/

TICKET\_ID BIGINT NOT NULL,

NUMSEQ SMALLINT NOT NULL,

DRINK\_ID SMALLINT NOT NULL,

DESCRIPTION VARCHAR(128),

QUANTITY SMALLINT(6) NOT NULL,

PRICE DECIMAL(20,2) NOT NULL,

TAX\_AMOUNT DECIMAL (20,5) NOT NULL,

DRINK\_AMOUNT DECIMAL (20,5) NOT NULL,

PRIMARY KEY (TICKET\_ID, NUMSEQ));

CREATE TABLE DRINKS (

DRINK\_ID SMALLINT NOT NULL PRIMARY KEY,

DRINK\_TYPE CHAR(2) NOT NULL,

SIZE\_CODE CHAR(2) NOT NULL,

SIZE DECIMAL (10,2) NOT NULL,

DRINK\_NAME VARCHAR(40) NOT NULL,

BRAND\_ID CHAR(3) NOT NULL);

CREATE TABLE UNIT\_MEASURES (

SIZE\_CODE CHAR(2) NOT NULL PRIMARY KEY,

DESCRIPTION VARCHAR(128) NOT NULL);

CREATE TABLE DRINK\_TYPES (

DRINK\_TYPE CHAR(2) NOT NULL PRIMARY KEY,

DESCRIPTION VARCHAR(128) NOT NULL);

CREATE TABLE BRANDS (

BRAND\_ID CHAR(3) NOT NULL PRIMARY KEY,

DESCRIPTION VARCHAR(128) NOT NULL);

CREATE TABLE PLAYER (

PLAYER\_ID CHAR(9) NOT NULL PRIMARY KEY,

FIRST\_NAME VARCHAR (120) NOT NULL,

LAST\_NAME VARCHAR (120) NOT NULL,

PLAYER\_EMAIL VARCHAR (120) NOT NULL,

PLAYER\_PHONE VARCHAR (20) NOT NULL,

PLAYER\_PASSPORT VARCHAR (9) NOT NULL

);

CREATE TABLE CASINO\_TRANSACTION (

TRANSACTION\_ID BIGINT NOT NULL PRIMARY KEY,

TRANSACTION\_TYPE CHAR(2) NOT NULL,

TOTAL\_AMOUNT BIGINT NOT NULL,

PLAYER\_ID CHAR(9) NOT NULL,

TRANSACTION\_TIME TIMESTAMP NOT NULL);

CREATE TABLE TRANSACTION\_TYPE (

TRANSACTION\_TYPE CHAR(2) NOT NULL PRIMARY KEY,

TRANSACTION\_DESCRIPTION VARCHAR(120) NOT NULL);

CREATE TABLE PAYMENT (

PAYMENT\_ID BIGINT NOT NULL PRIMARY KEY,

PAYMENT\_TYPE CHAR(2) NOT NULL,

CURRENCY\_TYPE CHAR(3) NOT NULL,

TRANSACTION\_ID BIGINT NOT NULL);

CREATE TABLE PAYMENT\_TYPE (

PAYMENT\_TYPE CHAR(2) NOT NULL PRIMARY KEY,

PAYMENT\_DESCRIPTION VARCHAR(120) NOT NULL);

CREATE TABLE CURRENCY\_TYPE (

CURRENCY\_TYPE CHAR(3) NOT NULL PRIMARY KEY,

CURRENCY\_DESCRIPTION VARCHAR(120) NOT NULL);

CREATE TABLE STACK (

STACK\_ID BIGINT NOT NULL PRIMARY KEY,

STACK\_AMOUNT BIGINT NOT NULL,

PLAYER\_ID CHAR(9) NOT NULL);

CREATE TABLE BET (

BET\_ID BIGINT NOT NULL PRIMARY KEY,

CHIPS\_IN SMALLINT NOT NULL,

CHIPS\_OUT SMALLINT NOT NULL,

START\_TIME TIMESTAMP NOT NULL,

END\_TIME TIMESTAMP NOT NULL,

TABLE\_ID VARCHAR(3) NOT NULL,

PLAYER\_ID CHAR(9) NOT NULL);

ALTER TABLE EMPLOYEE

ADD FOREIGN KEY (ROLE\_ID)

REFERENCES ROLES (ROLE\_ID);

ALTER TABLE EMPLOYEE

ADD FOREIGN KEY (CASINO\_ID)

REFERENCES CASINO (CASINO\_ID);

ALTER TABLE EMPLOYEE

ADD FOREIGN KEY (TABLE\_ID)

REFERENCES CASINO\_TABLE (TABLE\_ID);

ALTER TABLE CASINO

ADD FOREIGN KEY (LOCATION\_ID)

REFERENCES LOCATION (LOCATION\_ID);

ALTER TABLE LOCATION

ADD FOREIGN KEY (ADDRESS\_ID)

REFERENCES ADDRESS (ADDRESS\_ID);

ALTER TABLE LOCATION

ADD FOREIGN KEY (LOCATION\_TYPE)

REFERENCES LOCATION\_TYPE (LOCATION\_TYPE);

ALTER TABLE ADDRESS

ADD FOREIGN KEY (ADDRESS\_TYPE)

REFERENCES ADDRESS\_TYPE (ADDRESS\_TYPE);

ALTER TABLE ADDRESS

ADD FOREIGN KEY (ZIPCODE, COUNTRY)

REFERENCES ZIPCODE (ZIPCODE, COUNTRY);

ALTER TABLE TICKET\_DRINK

ADD FOREIGN KEY (TABLE\_ID)

REFERENCES CASINO\_TABLE(TABLE\_ID);

ALTER TABLE TICKET\_LINE\_DRINK

ADD FOREIGN KEY (TICKET\_ID)

REFERENCES TICKET\_DRINK(TICKET\_ID);

ALTER TABLE TICKET\_LINE\_DRINK

ADD FOREIGN KEY(DRINK\_ID)

REFERENCES DRINKS (DRINK\_ID);

ALTER TABLE DRINKS

ADD FOREIGN KEY (SIZE\_CODE)

REFERENCES UNIT\_MEASURES (SIZE\_CODE);

ALTER TABLE DRINKS

ADD FOREIGN KEY (DRINK\_TYPE)

REFERENCES DRINK\_TYPES (DRINK\_TYPE);

ALTER TABLE DRINKS

ADD FOREIGN KEY (BRAND\_ID)

REFERENCES BRANDS (BRAND\_ID);

ALTER TABLE CASINO\_TRANSACTION

ADD FOREIGN KEY (PLAYER\_ID)

REFERENCES PLAYER(PLAYER\_ID);

ALTER TABLE CASINO\_TRANSACTION

ADD FOREIGN KEY (TRANSACTION\_TYPE)

REFERENCES TRANSACTION\_TYPE(TRANSACTION\_TYPE);

ALTER TABLE PAYMENT

ADD FOREIGN KEY (TRANSACTION\_ID)

REFERENCES CASINO\_TRANSACTION(TRANSACTION\_ID);

ALTER TABLE PAYMENT

ADD FOREIGN KEY (PAYMENT\_TYPE)

REFERENCES PAYMENT\_TYPE(PAYMENT\_TYPE);

ALTER TABLE PAYMENT

ADD FOREIGN KEY (CURRENCY\_TYPE)

REFERENCES CURRENCY\_TYPE(CURRENCY\_TYPE);

ALTER TABLE STACK

ADD FOREIGN KEY (PLAYER\_ID)

REFERENCES PLAYER(PLAYER\_ID);

ALTER TABLE BET

ADD FOREIGN KEY (TABLE\_ID)

REFERENCES CASINO\_TABLE(TABLE\_ID);

ALTER TABLE BET

ADD FOREIGN KEY (PLAYER\_ID)

REFERENCES PLAYER(PLAYER\_ID);

**DML**

INSERT INTO ZIPCODE VALUES

(1134784587, 'SPAIN', 'MADRID', 'COMUNIDAD DE MADRID') ;

INSERT INTO ADDRESS\_TYPE VALUES

('ST', 'STREET'),

('AV', 'AVENUE'),

('BL', 'BOULEVARD'),

('HW', 'HIGHWAY');

INSERT INTO ADDRESS VALUES

(4556, 'ST', 'Calle Carlos Lang Best Teacher Ever', 117, 1134784587, 'SPAIN');

INSERT INTO LOCATION\_TYPE VALUES

('CA', 'Casino'),

('OF', 'Offices'),

('PK', 'Parking Lot'),

('WH', 'Warehouse'),

('HO', 'Hotel');

INSERT INTO LOCATION VALUES

(4589863565785, 4556,'CA'),

(4545363555555, 4556,'PK');

INSERT INTO CASINO VALUES

(1552, 'MARE TURTUS CASINO & BAR', 4589863565785) ;

INSERT INTO ROLES VALUES

('A01', 'DEALERS & GAME MANAGEMENT'),

('B01', 'DRINKS SERVICE'),

('C01', 'SECURITY'),

('D01', 'MANAGEMENT'),

('E01', 'CLEANING');

INSERT INTO GAMETYPE

VALUES

("C","Card Game"),

("T","Table Game")

;

INSERT INTO GAME (GAME\_ID, GAME\_TYPE,GAME\_NAME,MAX\_PLAYERS,MIN\_BUYIN)

VALUES

("PKR","C","Poker",8,50),

("BJK","C","Black Jack 21",7,25),

("RLT","T","Roulette",5,25),

("BCC","C","Baccarat",7,25),

("PGP","C","Pai Gow Poker",6,25)

;

INSERT INTO CASINO\_TABLE

VALUES

("11","PKR",1552),

("22","BJK",1552),

("33","RLT",1552),

("44","BCC",1552),

("55","PGP",1552),

("66","PKR",1552),

("77","BJK",1552),

("88","PKR",1552),

("99","RLT",1552)

;

INSERT INTO EMPLOYEE VALUES

(79403168, 'Shirley', 'Hogan', 'E01','2018-01-01','1990-01-02', 1552,NULL ),

(73403111, 'Ramon', 'Sikunt', 'A01','2018-05-11','1996-06-10', 1552,22 ),

(89405169, 'Yara', 'Lamar', 'A01','2022-04-01','2000-01-02', 1552,11 ),

(69106661, 'Raul', 'Bond', 'C01','2021-02-01','1994-04-01', 1552,NULL ),

(81400145, 'Jorge', 'Jones', 'B01','2022-06-10','1998-03-11', 1552,NULL ),

(83201132, 'Philipp', 'Doolittle', 'A01','2020-09-01','1999-11-08', 1552,33 ),

(74413332, 'Dilhan', 'Jefe', 'C01','2020-02-01','1993-02-27', 1552,NULL ),

(54036817, 'Francesco', 'Nino', 'B01','2018-01-01','1989-06-22', 1552,NULL),

(87132045, 'Alana', 'Hunt', 'B01','2019-01-01','1991-05-25', 1552,NULL),

(17348056, 'Hayden', 'Park', 'A01','2022-04-07','1980-01-02', 1552,44),

(80412536, 'Lucie', 'Lopez', 'B01','2021-08-04','1990-08-28', 1552,NULL),

(57089243, 'Trystan', 'Rhodes', 'C01','2018-01-01','1989-11-29', 1552,NULL),

(86524907, 'Grayson', 'Hampton', 'C01','2020-05-20','1995-12-02', 1552,NULL),

(53042197, 'Dillon', 'Day', 'C01','2020-07-14','1987-08-14', 1552,NULL),

(40598721, 'Vanessa', 'Shannon', 'A01','2020-08-11','1990-06-18', 1552,55),

(45627830, 'Stevie', 'Martinez', 'C01','2021-02-09','1988-06-02', 1552,NULL),

(57938601, 'Juanita', 'Mcdowell', 'D01','2018-01-01','1990-01-02', 1552,NULL),

(87430659, 'Mariah', 'Stevenson', 'E01','2019-01-01', '1990-01-02', 1552,NULL)

;

INSERT INTO PLAYER VALUES

('123456789','Adam', 'Smith','adam.smith@web.com','+44573920673923','G571HD375'),

('133456789','John', 'Lennon','john.lennon@web.com','+44309473857125','J372DN679'),

('134456789','Michael', 'Ballack','michael.ballack@web.com','+49793057183495','S579FA195'),

('134556789','Gerhard', 'Schroeder','gerhard.schroeder@web.com','+49193405800348','A062OD184'),

('134566789','Morgan', 'Freeman','morgan.freeman@web.com','+1049387624582','U295IE269'),

('134567789','Al', 'Pacino','al.pacino@web.com','+119458956894','K379DA729'),

('134567889','Roger', 'Federer','roger.federer@web.com','+417772904058123','X682UN172'),

('134667899','Frida', 'Kahlo','frida.kahlo@web.com','+52275891904759','P582JM143'),

('144567899','Neymar', 'Dos Santos','neyney@web.com','+522855919047144','L212NP756'),

('134555199','Lionel', 'Messi','thegoat@web.com','+12264211411111','N345MM999'),

('134567899','Daniel', 'Negreanu','pokergod@web.com','+51244411204555','M182BB149');

INSERT INTO TRANSACTION\_TYPE VALUES

('CI','Cash in at cashier'),

('CO','Cash out at cashier');

INSERT INTO CASINO\_TRANSACTION VALUES

(112393058295708,"CI",500,123456789,"2022-11-16 13:18:23"),

(112393058375930,"CI",100,133456789,"2022-11-12 19:47:16"),

(112393058358301,"CI",250,134456789,"2022-11-30 00:00:19"),

(112393058103985,"CI",600,134556789,"2022-11-25 00:10:03"),

(112393058194584,"CI",50,134566789,"2022-12-08 04:35:49"),

(112393058018359,"CI",650,134567789,"2022-12-01 18:52:54"),

(112393058248629,"CI",800,134567889,"2022-11-23 02:53:18"),

(112393058219543,"CO",1000,134567899,"2022-11-07 23:18:24"),

(112393079219666,"CI",5500,144567899,"2022-11-10 13:29:44"),

(113393189219616,"CI",6000,134555199,"2022-12-07 18:34:27"),

(102323079219636,"CI",7000,134567899,"2022-12-06 15:42:50"),

(112393058295728,"CI",1200,123456789,"2022-11-21 18:30:25"),

(112393058295729,"CI",150,133456789,"2022-12-06 13:36:02"),

(112393058295730,"CI",2500,134456789,"2022-11-17 08:09:30"),

(112393058295731,"CI",2800,134556789,"2022-11-08 04:20:43"),

(112393058295732,"CI",4800,134566789,"2022-12-08 04:37:42"),

(112393058295733,"CI",4700,134567789,"2022-11-12 06:38:36"),

(112393058295734,"CO",1800,134567889,"2022-11-25 02:50:08"),

(112393058295735,"CI",4550,134567899,"2022-11-18 04:53:46"),

(112393058295736,"CI",550,144567899,"2022-11-18 11:40:16"),

(112393058295737,"CI",950,134555199,"2022-11-11 09:30:18"),

(112393058295738,"CI",2000,134567899,"2022-12-04 19:56:38"),

(112393058295739,"CI",1650,123456789,"2022-11-20 21:35:06"),

(112393058295740,"CI",4050,133456789,"2022-11-25 08:57:25"),

(112393058295741,"CO",3500,134456789,"2022-11-25 21:12:45"),

(112393058295742,"CI",1150,134556789,"2022-11-24 18:02:55"),

(112393058295743,"CI",4450,134566789,"2022-12-06 06:39:48"),

(112393058295744,"CI",2300,134567789,"2022-11-29 04:16:13"),

(112393058295745,"CI",1350,134567889,"2022-12-02 01:22:11"),

(112393058295746,"CI",4750,134567899,"2022-12-06 14:40:50"),

(112393058295747,"CO",3500,144567899,"2022-12-03 16:32:50"),

(112393058295748,"CI",4300,134555199,"2022-12-01 21:54:02"),

(112393058295749,"CI",350,134567899,"2022-11-26 00:03:38"),

(112393058295750,"CI",3700,123456789,"2022-11-07 12:56:06"),

(112393058295751,"CI",3300,133456789,"2022-11-08 10:58:37"),

(112393058295752,"CO",3650,134456789,"2022-11-11 10:44:37"),

(112393058295753,"CI",1500,134556789,"2022-11-26 05:29:22"),

(112393058295754,"CI",550,134566789,"2022-11-17 10:07:18"),

(112393058295755,"CI",4750,134567789,"2022-11-29 09:56:51"),

(112393058295756,"CI",250,134567889,"2022-12-02 15:54:02")

;

INSERT INTO PAYMENT\_TYPE VALUES

('CS','Cash'),

('CC','Credit card'),

('CQ','Cheque');

INSERT INTO CURRENCY\_TYPE VALUES

('AUD','Australian Dollar'),

('BRL','Real'),

('BGN','Lev'),

('CAD','Canadian Dollar'),

('CNY','Yan Renmibi'),

('HRK','Kuna'),

('CYP','Cyprian Pound'),

('CZK','Czech Kroner'),

('DKK','Danish Kroner'),

('EEK','Kroon'),

('EUR','Euro'),

('HKD','Hong Kong Dollar'),

('HUF','Forint'),

('ISK','Krónal'),

('IDR','Rupiah'),

('JPY','Yen'),

('KRW','Won'),

('LVL','Lats'),

('LTL','Litas'),

('MYR','Ringgit'),

('MTL','Lire'),

('NZD','New Zealand Dollar'),

('NOK','Kroner'),

('PHP','Peso'),

('PLN','Zloty'),

('RON','Leu'),

('RUB','Rubles'),

('SGD','Singapore Dollar'),

('SKK','Slovakian Kroner'),

('SIT','Tolar'),

('ZAR','Rand'),

('SEK','Swedish Kroner'),

('CHF','Franc'),

('THB','Baht'),

('TRY','Lire'),

('GBP','British Pound'),

('USD','US Dollar');

INSERT INTO PAYMENT VALUES

(957281028519437,'CC','EUR',112393058295728),

(759301035834910,'CC','EUR',112393058375930),

(583010439230458,'CC','EUR',112393058358301),

(639852010391349,'CC','EUR',112393058103985),

(239852010358204,'CC','EUR',112393058194584),

(339852847110492,'CC','EUR',112393058018359),

(183591235430123,'CC','EUR',112393058248629),

(486293027820249,'CC','EUR',112393058248629);

INSERT INTO STACK VALUES

(718205833929,50000,'123456789'),

(518265132534,180000,'133456789'),

(618267438229,75000,'134456789'),

(118347432853,76000,'134556789'),

(710267612895,54700,'134566789'),

(629574593759,65000,'134567789'),

(249602876729,80000,'134567889'),

(872057682951,120500,'134567899'),

(778347432853,53000,'144567899'),

(114347432823,84000,'134555199'),

(314347432813,57000,'134567899');

INSERT INTO BET VALUES

(239305829572810285,200,0,'2022-12-07 23:15:29','2022-12-07 23:15:43','11','123456789'),

(271301829572010287,500,0,'2022-12-07 23:14:19','2022-12-07 23:14:55','11','144567899'),

(208305429532810286,1000,3000,'2022-12-07 23:15:29','2022-12-07 23:15:43','99','134555199'),

(221304129522810285,2000,0,'2022-12-07 23:15:29','2022-12-07 23:15:43','77','134567899'),

(239305837593010358,20,400,'2022-12-07 23:16:01','2022-12-07 23:16:15','66','133456789'),

(239305835830104392,200,0,'2022-12-07 23:23:53','2022-12-07 23:23:59','33','134456789'),

(239305810398520103,50,200,'2022-12-07 23:23:53','2022-12-07 23:23:59','44','134556789'),

(239305810398528471,50,0,'2022-12-07 23:33:23','2022-12-07 23:33:36','55','134556789'),

(239305819458430348,200,0,'2022-12-07 23:40:36','2022-12-07 23:40:48','66','134566789'),

(239305801835912354,200,0,'2022-12-07 23:41:12','2022-12-07 23:41:27','77','134567789'),

(239305824862930278,300,400,'2022-12-07 23:48:58','2022-12-07 23:49:13','99','134567889'),

(299305829572810280,2000,0,'2022-11-07 23:49:17','2022-11-07 23:49:25','11','123456789'),

(281301829572010286,4000,0,'2022-11-08 03:36:43','2022-11-08 03:36:47','11','144567899'),

(288305429532810280,1000,3000,'2022-11-09 01:39:43','2022-11-09 01:39:48','99','134555199'),

(291304129522810286,1000,0,'2022-11-09 21:26:15','2022-11-09 21:26:22','77','134567899'),

(289305837593010350,200,4000,'2022-11-09 23:00:23','2022-11-09 23:00:29','66','133456789'),

(299305835830104396,600,800,'2022-11-10 21:22:18','2022-11-10 21:22:22','33','134456789'),

(289305810398520100,5000,6000,'2022-11-13 00:31:25','2022-11-13 00:31:31','44','134556789'),

(299305810398528476,700,3000,'2022-11-13 03:05:09','2022-11-13 03:05:14','55','134556789'),

(299305819458430340,600,700,'2022-11-13 18:19:37','2022-11-13 18:19:41','66','134566789'),

(289305801835912356,2000,0,'2022-11-17 20:30:15','2022-11-17 20:30:20','77','134567789'),

(289305824862930270,3000,4000,'2022-11-19 02:10:36','2022-11-19 02:10:41','99','134567889'),

(299305823572810280,300,1000,'2022-11-19 02:10:36','2022-11-19 02:10:41','11','123456789'),

(281301873572010286,400,600,'2022-11-21 18:02:31','2022-11-21 18:02:39','11','144567899'),

(288305443532810280,2000,3000,'2022-11-22 01:55:01','2022-11-22 01:55:09','99','134555199'),

(291304229522810286,4000,0,'2022-11-24 01:09:29','2022-11-24 01:09:33','77','134567899'),

(289308937593010350,2000,4000,'2022-11-28 18:32:51','2022-11-28 18:32:57','66','133456789'),

(299305809830104396,900,1000,'2022-11-29 03:51:36','2022-11-29 03:51:43','33','134456789'),

(289305879398520100,2000,0,'2022-11-30 18:12:46','2022-11-30 18:12:51','44','134556789'),

(299305810898528476,300,500,'2022-12-01 21:10:36','2022-12-01 21:10:45','55','134556789'),

(299305819128430340,4000,0,'2022-12-02 18:54:25','2022-12-02 18:54:31','66','134566789'),

(289305801675912356,200,2000,'2022-12-03 21:21:51','2022-12-03 21:21:58','77','134567789'),

(289305824868833470,3000,4000,'2022-12-04 03:38:11','2022-12-04 03:38:16','99','134567889'),

(289305824868838470,1000,4000,'2022-12-05 22:13:51','2022-12-05 22:13:57','99','134567889');

INSERT INTO BRANDS VALUES

('BAC', 'Bacardi Carta Blanca'),

('GGO', 'Grey Goose'),

('SSB', 'SuperBock'),

('KEO', 'Kettle One'),

('CCO', 'Coca Cola'),

('ICT', 'Iced Tea'),

('SOC', 'Solan de Cabras'),

('TON', 'Schweppes'),

('PTC', 'Porto Cruz Porto Wine'),

('CTS', 'Cutty Sark'),

('QTZ', 'Quinta de Ventozelo Dry Gin'),

('VAC', 'Vino Blanco Domaine Vacheron'),

('RUT', 'Rutini'),

('PLC', 'Pago de Los Capellanes'),

('DAL', 'Dalva Tinto Wine 2019'),

('BEL', 'Belvedere'),

('ABS', 'Absolut'),

('TDJ', 'Don Julio'),

('GIB', 'Gibson'),

('HEN', 'Hendricks'),

('BBS', 'Bombay Sapphire'),

('AQU', 'Aquafina'),

('VOS', 'VOSS'),

('DOM', 'Dom Perignon Vintage 2012'),

('MUR', 'Murganheira Sparkling Wine 2018'),

('ORA', 'Vitamin Orange Juice'),

('MAH', 'Cerveza MAHOU'),

('EST', 'Estrella Damm'),

('PEL', 'Pellegrino'),

('MCC', 'Macallan Double 12 Years Old'),

('SOM', 'Sommersby'),

('TER', 'Terra Nostra');

INSERT INTO DRINK\_TYPES VALUES

('RU', 'Rhum'),

('VO', 'Vodka'),

('SO', 'Soda'),

('WA', 'Water'),

('PW', 'Porto Wine'),

('SW', 'Sparkling Wine'),

('CH', 'Champagne'),

('WH', 'Whiskey'),

('CI', 'Cider'),

('GI', 'Gin'),

('WI', 'Wine'),

('JU', 'Juice'),

('BE', 'Beer'),

('TE', 'Tequilla');

INSERT INTO UNIT\_MEASURES VALUES

('ML', 'Milliliters'),

('CL', 'Centiliters'),

('DL', 'Deciliters'),

('LI', 'Liters');

INSERT INTO DRINKS VALUES

(10098 , 'BE', 'LI', 0.5, 'Beer Super Bock ', 'SSB'),

(6669, 'BE', 'LI',0.5, 'Beer Mahou', 'MAH'),

(19028, 'BE', 'LI', 0.5, 'Beer Estrella Damm','EST'),

(4093, 'VO', 'DL', 2 , 'Vodka Grey Goose', 'GGO'),

(29851, 'CH' , 'ML', 175, 'Champagne Dom Perignon', 'DOM'),

(18929, 'TE', 'ML', 120, 'Tequilla Don Julio', 'TDJ'),

(9565, 'SW' , 'ML', 250, 'Sparkling Wine Murganheira', 'MUR'),

(3376, 'RU', 'DL', 2, 'Rhum Bacardi', 'BAC'),

(12675, 'VO', 'DL', 2, 'Vodka Kettle One', 'KEO'),

(10294, 'VO', 'DL', 2, 'Vodka Belvedere', 'BEL'),

(28352, 'VO', 'DL', 2, 'Vodka Absolut', 'ABS'),

(24853, 'SO', 'ML', 330, 'Coca-Cola', 'CCO'),

(12005, 'SO', 'ML', 330, 'Iced-Tea', 'ICT'),

(24757, 'WA', 'LI', 0.5, 'Water Solan de Cabras', 'SOC'),

(1212, 'WA', 'LI', 0.5, 'Tonic Water', 'TON'),

(4184, 'WA', 'LI', 0.5, 'Pelligrino Sparkling Water', 'PEL'),

(3111, 'WA', 'LI', 0.5, 'Water Aquafina', 'AQU'),

(5421, 'WA', 'LI', 0.5, 'Voss Water', 'VOS'),

(8629, 'PW', 'ML', 300, 'Porto Cruz Porto Wine', 'PTC'),

(17365, 'WH', 'ML', 300, 'Cutty Sark Whisky', 'CTS'),

(16071, 'WH', 'ML', 300, 'Macallan Double 12 Years Old', 'MCC'),

(19000, 'GI', 'DL', 2, 'Quinta de Ventozelo Dry Gin', 'QTZ'),

(1235, 'GI', 'DL', 2, 'Gibson Dry Gin', 'GIB'),

(4332, 'GI', 'DL',2, 'Hendricks Gin', 'HEN'),

(9643, 'GI', 'DL', 2, 'Bombay Sapphire Gin', 'BBS'),

(9122, 'WI', 'ML', 300, 'Common Vino Blanco', 'VAC'),

(10968, 'WI', 'ML', 2, 'Rutini White Wine', 'RUT'),

(4787, 'WI', 'ML', 2, 'Pago de los Capellanes White Wine', 'PLC'),

(9291, 'WI', 'ML', 2, 'Dalva White Wine', 'DAL'),

(96, 'CI', 'ML', 330, 'Sommersby Cider', 'SOM'),

(1259, 'JU', 'ML', '300', 'Orange Juice', 'ORA');

INSERT INTO TICKET\_DRINK VALUES

(4013323,'2022-11-07 23:23:28',134567899,22,87132045,2.5,0.58,3.08),

(7596545,'2022-11-09 00:29:21',133456789,44,80412536,21.2,4.88,26.08),

(2281185,'2022-11-10 00:03:01',134456789,44,54036817,8.5,1.96,10.46),

(6701110,'2022-11-10 23:28:55',123456789,88,54036817,8.5,1.96,10.46),

(455208,'2022-11-11 19:21:46',134566789,66,87132045,24.5,5.66,30.16),

(4699484,'2022-11-13 01:13:00',134556789,66,54036817,14,3.22,17.22),

(4103221,'2022-11-14 00:12:12',134567789,77,80412536,17,3.92,20.92),

(2038373,'2022-11-15 03:23:18',134567889,11,54036817,12,2.76,14.76),

(3780942,'2022-11-15 23:03:51',134566789,22,87132045,9.5,2.21,11.71),

(3204281,'2022-11-17 02:19:17',123456789,88,87132045,17,3.92,20.92),

(4107124,'2022-11-17 19:15:18',134556789,44,54036817,10,2.31,12.31),

(9646814,'2022-11-19 01:15:31',134567889,55,80412536,8.5,1.96,10.46),

(7880356,'2022-11-19 02:30:09',133456789,77,87132045,15,2.09,17.09),

(8078146,'2022-11-20 01:10:03',134456789,55,80412536,17,3.92,20.92),

(5415307,'2022-11-21 01:18:53',134567789,99,87132045,42,9.67,51.67),

(7134296,'2022-11-22 01:50:17',134567899,88,80412536,8.5,1.96,10.46),

(4288080,'2022-11-23 01:04:03',133456789,77,87132045,9,2.08,11.08),

(7617106,'2022-11-24 19:40:18',134567899,44,54036817,17,3.92,20.92),

(3019076,'2022-11-25 18:12:41',134456789,66,54036817,16,3.7,19.7),

(1451271,'2022-11-26 20:15:16',134567789,77,54036817,8,1.84,9.84),

(1496463,'2022-11-28 23:42:44',123456789,55,87132045,17,3.92,20.92),

(1014805,'2022-11-29 18:08:18',134566789,11,80412536,25.5,5.88,31.38),

(742513,'2022-11-30 23:58:26',134556789,66,54036817,5,1.16,6.16),

(6564324,'2022-12-02 00:39:17',134567889,77,54036817,25.5,5.88,31.38),

(6054417,'2022-12-03 01:35:16',134567889,55,87132045,2,0.46,2.46),

(159671,'2022-12-03 23:06:08',134566789,66,87132045,17,3.92,20.92),

(554261,'2022-12-05 02:04:48',134567899,22,87132045,20,4.61,24.61),

(192779,'2022-12-05 19:06:48',134556789,66,80412536,8.5,1.96,10.46),

(25225,'2022-12-06 23:39:12',123456789,88,80412536,6,1.38,7.38),

(437760,'2022-12-07 20:15:32',134567789,22,87132045,19.5,4.5,24);

INSERT INTO TICKET\_LINE\_DRINK

VALUES

(4013323,1,6669,"Beer Mahou",1,2.5,0.58,3.08),

(7596545,1,9291,"Dalva White Wine",2,6,1.38,14.76),

(7596545,2,16071,"McCallan Double",1,9.2,2.12,11.32),

(2281185,1,4093,"Vodka Grey Goose",1,8.5,1.96,10.46),

(6701110,1,4093,"Vodka Grey Goose",1,8.5,1.96,10.46),

(455208,1,24853,"Coca Cola",2,1.5,0.35,3.7),

(455208,2,3376,"Rhum Bacardi",2,8.5,1.96,20.92),

(455208,3,4787,"Pago de Los Capellanes White Wine",1,4.5,1.04,5.54),

(4699484,1,4332,"Hendricks Dry Gin",1,8,1.84,9.84),

(4699484,2,10098,"Beer Super Bock ",2,3,0.69,7.38),

(4103221,1,4093,"Vodka Grey Goose",2,8.5,1.96,20.92),

(2038373,1,4184,"Pellegrino Sparkling Water",3,4,0.92,14.76),

(3780942,1,24853,"Coca Cola",1,1.5,0.35,1.85),

(3780942,2,24757,"Water Solan de Cabras",2,1.5,0.35,3.7),

(3780942,3,19028,"Beer Estrella Damm",2,2.5,0.58,6.16),

(3204281,1,17365,"Cutty Sark",2,8.5,1.96,20.92),

(4107124,1,17365,"Cutty Sark",1,8.5,1.96,10.46),

(4107124,2,24853,"Coca Cola",1,1.5,0.35,1.85),

(9646814,1,4093,"Vodka Grey Goose",1,8.5,1.96,10.46),

(7880356,1,24757,"Water Solan de Cabras",1,1.5,0.35,1.85),

(7880356,2,6669,"Beer Mahou",2,2.5,0.58,6.16),

(7880356,3,19028,"Beer Estrella Damm",1,2.5,0.58,3.08),

(8078146,1,17365,"Cutty Sark Whisky",2,8.5,1.96,20.92),

(5415307,1,5421,"Voss Water",1,5,1.15,6.15),

(5415307,2,17365,"Cutty Sark Whisky",2,8.5,1.96,20.92),

(5415307,3,29851,"Champagne Dom Perignon ",1,20,4.6,24.6),

(7134296,1,10294,"Belvedere Vodka",1,8.5,1.96,10.46),

(4288080,1,24757,"Water Solan de Cabras",2,1.5,0.35,3.7),

(4288080,2,10098,"SuperBock Beer",2,3,0.69,7.38),

(7617106,1,17365,"Cutty Sark Whisky",2,8.5,1.96,20.92),

(3019076,1,9122,"Common Vino Blanco",3,2.5,0.58,9.24),

(3019076,2,12675,"Vodka Kettle One ",1,8.5,1.96,10.46),

(1451271,1,4184,"Pellegrino Sparkling Water",2,4,0.92,9.84),

(1496463,1,17365,"Cutty Sark Whisky",2,8.5,1.96,20.92),

(1014805,1,17365,"Cutty Sark Whisky",3,8.5,1.96,31.38),

(742513,1,6669,"Beer Mahou",2,2.5,0.58,6.16),

(6564324,1,17365,"Cutty Sark Whisky",3,8.5,1.96,31.38),

(6054417,1,96,"Orange Juice",1,2,0.46,2.46),

(159671,1,3376,"Rhum Bacardi",2,8.5,1.96,20.92),

(554261,1,17365,"Cutty Sark Whisky",2,8.5,1.96,20.92),

(554261,2,10098,"Beer Super Bock ",1,3,0.69,3.69),

(192779,1,10294,"Belvedere Vodka",1,8.5,1.96,10.46),

(25225,1,9291,"Dalva White Wine",1,6,1.38,7.38),

(437760,1,17365,"Cutty Sark Whisky",2,8.5,1.96,20.92),

(437760,2,9122,"Common Vino Blanco",1,2.5,0.58,3.08);