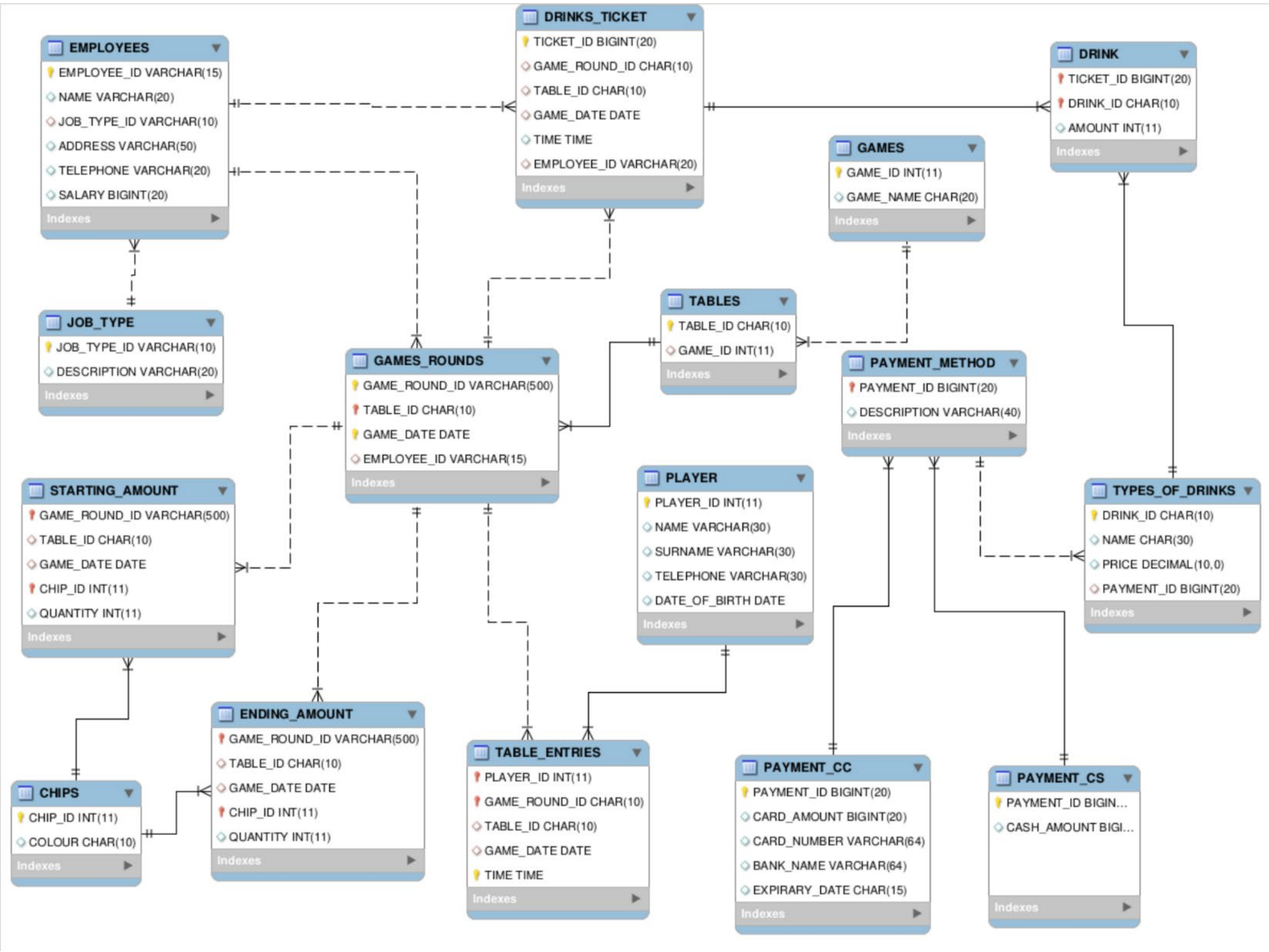
SQL DATAWAREHOUSE ASSIGNMENT

GROUP H – CASINO DATABASE DATA MODEL

Members: Martina Cilia, Youssef Zaroub, Thomas Stenger, Paloma Van Ginderachter, Orlando Montalvo, Quirijn Bolhuis, Mikael Dzhaneryan .

Table of Contents

Entity Relationship Diagram (ERD) of the Casino:



DRINKS_TICKET
TICKET_ID BIGINT(20)
GAME_ROUND_ID CHAR(10)
TABLE_ID CHAR(10)
GAME_DATE DATE
TIME TIME
EMPLOYEE_ID VARCHAR(20)
Indexes

EMPLOYEES
EMPLOYEE_ID VARCHAR(15)
NAME VARCHAR(20)
JOB_TYPE_ID VARCHAR(10)
ADDRESS VARCHAR(50)
TELEPHONE VARCHAR(20)
SALARY BIGINT(20)
Indexes

DRINK
TICKET_ID BIGINT(20)
DRINK_ID CHAR(10)
AMOUNT INT(11)
Indexes

GAMES
GAME_ID INT(11)
GAME_NAME CHAR(20)
Indexes

JOB_TYPE
JOB_TYPE_ID VARCHAR(10)
DESCRIPTION VARCHAR(20)
Indexes

TABLES
TABLE_ID CHAR(10)
GAME_ID INT(11)
Indexes

GAMES_ROUNDS
GAME_ROUND_ID VARCHAR(500)
TABLE_ID CHAR(10)
GAME_DATE DATE
EMPLOYEE_ID VARCHAR(15)
Indexes

PAYMENT_METHOD
PAYMENT_ID BIGINT(20)
DESCRIPTION VARCHAR(40)
Indexes

STARTING_AMOUNT
GAME_ROUND_ID VARCHAR(500)
TABLE_ID CHAR(10)
GAME_DATE DATE
CHIP_ID INT(11)
QUANTITY INT(11)
Indexes

PLAYER
PLAYER_ID INT(11)
NAME VARCHAR(30)
SURNAME VARCHAR(30)
TELEPHONE VARCHAR(30)
DATE_OF_BIRTH DATE
Indexes

TYPES_OF_DRINKS
DRINK_ID CHAR(10)
NAME CHAR(30)
PRICE DECIMAL(10,0)
PAYMENT_ID BIGINT(20)
Indexes

ENDING_AMOUNT
GAME_ROUND_ID VARCHAR(500)
TABLE_ID CHAR(10)
GAME_DATE DATE
CHIP_ID INT(11)
QUANTITY INT(11)
Indexes

TABLE_ENTRIES
PLAYER_ID INT(11)
GAME_ROUND_ID CHAR(10)
TABLE_ID CHAR(10)
GAME_DATE DATE
TIME TIME
Indexes

PAYMENT_CC
PAYMENT_ID BIGINT(20)
CARD_AMOUNT BIGINT(20)
CARD_NUMBER VARCHAR(64)
BANK_NAME VARCHAR(64)
EXPIRARY_DATE CHAR(15)
Indexes

PAYMENT_CS
PAYMENT_ID BIGIN...
CASH_AMOUNT BIGI...
Indexes

CHIPS
CHIP_ID INT(11)
COLOUR CHAR(10)
Indexes

Explanation of the data model:

Our group created the above Casino data model based on an assumption that the casino is not part of a chain of casinos and that it is has no chain of casinos either. This data model is comprised of 16 tables, with the names of the entities and their relevant attributes as show in the above ERD.

The games table is showing the game name with its relevant game identification number. Each game is played on different tables each day. The table identifier is connected to a game played on a particular day supervised by an employee.

Each game round starts off with a fixed amount of chips every day and finishes with another fixed amount of chips each day. The latter and former amounts are updated each day.

The chips making up the amounts are identified by value, with a specific colour.

The employees in the casino work in different job types; either as a dealer in games or a waiter serving the drinks.

Drinks can be bought at a table during a particular game round, and the ticket amount can be either paid by card (payment_cc) or cash (payment_cs). Furthermore, each drink bought in a ticket is identified by a drink id (synonymous to a product id). The information about each type of drink is found in the type of drinks table.

The relations existing between the tables are all *one to many relations*, described hereunder:

- GAMES and TABLES. A single game can be played on multiple tables at one go but a table can only hold one game at any point in time (i.e you cannot have multiple games played on the same table at the same time).
- GAMES_ROUNDS and TABLE_ENTRIES. A game can have multiple table entries but a table entry can only belong to a particular game.
- PLAYER and TABLE_ENTRIES. A player can play more than one game at different tables. However, a game is played by a single player.
- GAMES_ROUNDS and STARTING_AMOUNT. A game round can have multiple starting amounts but a single starting amount can only belong to a game round.
- GAMES_ROUNDS and STARTING_AMOUNT. A game round can have multiple ending amounts but a single ending amount can only belong to a game round.
- CHIPS and STARTING_AMOUNT. A starting amount is comprised of a fixed set of chips but various chips can make up different starting amounts.
- CHIPS and ENDING_AMOUNT. An ending amount is comprised of a fixed set of chips but various chips can make up different ending amounts.
- EMPLOYEES and GAMES_ROUNDS. More than one employee can be at more than one game round but a game round can only have one employee at a point in time.
- JOB_TYPE and EMPLOYEES. An employee has only one job type and a job type has many employees.

- EMPLOYEE and DRINKS_TICKET. An employee can (waiter) can issue multiple drink tickets, but a drink ticket can only be issued by a single employee.
- DRINKS_TICKET and DRINK. A drink ticket can contain many different types of purchased drinks in varying amounts, but a bought drink is identified by one particular drink ticket identification number.
- TYPE_OF_DRINKS and DRINK. A type of drink, identified by a drink identification number, can be related to more than one ticket. However, a drink is only represented by one identifier.
- PAYMENT_CC and PAYMENT_METHOD. More than one payment can be done by a credit card, with each purchase being given a payment identifier. Each payment id is representing a particular payment done by a credit card.
- PAYMENT_CS and PAYMENT_METHOD. A payment done by cash can occur more than once, but each purchase made by cash is identified by a single payment id.

Moreover, two relations link tables, depending on whether the child table can or cannot be identified without the parent table. The need for a parent table signifies an identifying relationship between the two tables in question. In our model the relations are distributed as follows:

Identifying Relations	
Child table	***Parent table***
Tables	Games_Rounds
Table_Entries	Player
Payment_CC	Payment_Method
Payment_CS	Payment_Method
Drinks	Types_of_Drinks
Drinks_Ticket	Drinks
Starting_Amount	Chips
Ending_Amount	Chips
Non-Identifying Relations	
Employees	Drinks_Ticket
Employees	Job_Type
Games_Rounds	Employees
Starting_Amount	Games_Rounds
Ending_Amount	Games_Rounds
Games_Rounds	Table_Entries
Games	Tables
Payment_Method	Types_of_Drinks

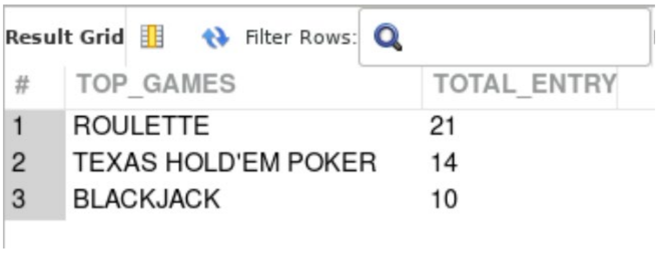
Conclusively, there are *7 datatypes* in this Casino Data Model, represented as show in this table:

Data Type	**Total**
Integer	**10**
Big Integer	**7**
Decimal	**1**
Char	**12**
Varchar	**17**
Date	**4**
Time	**2**

Queries:

1) What are the 3 top demanding casino games?

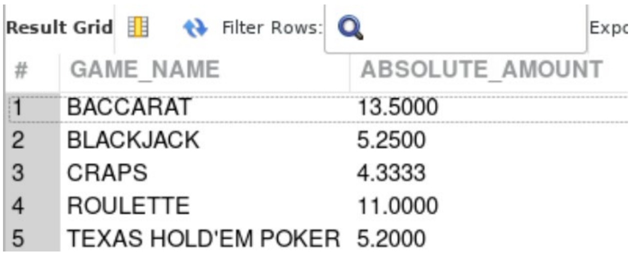
```
SELECT GAME_NAME AS TOP_GAMES, COUNT(*) AS TOTAL_ENTRY
FROM GAMES, TABLES, TABLE_ENTRIES, GAMES_ROUNDS
WHERE GAMES.GAME_ID = TABLES.GAME_ID
AND TABLES.TABLE_ID = GAMES_ROUNDS.TABLE_ID
AND GAMES_ROUNDS.TABLE_ID=TABLE_ENTRIES.TABLE_ID
GROUP BY GAME_NAME
ORDER BY COUNT(*) DESC
LIMIT 3;
```



#	TOP_GAMES	TOTAL_ENTRY
1	ROULETTE	21
2	TEXAS HOLD'EM POKER	14
3	BLACKJACK	10

2) Show the average number of chips per type of game and per day.

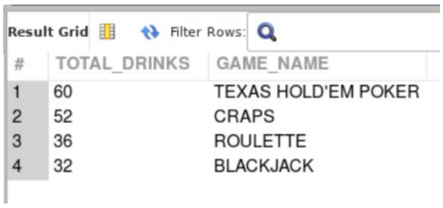
```
SELECT GAME_NAME, ABS(SUM(STARTING_AMOUNT.QUANTITY)-
SUM(ENDING_AMOUNT.QUANTITY))/COUNT(DISTINCT GAMES_ROUNDS.GAME_DATE)
FROM STARTING_AMOUNT, GAMES_ROUNDS, ENDING_AMOUNT, TABLES, GAMES
WHERE STARTING_AMOUNT.GAME_ROUND_ID = GAMES_ROUNDS.GAME_ROUND_ID
AND ENDING_AMOUNT.GAME_ROUND_ID = GAMES_ROUNDS.GAME_ROUND_ID
AND GAMES_ROUNDS.TABLE_ID = TABLES.TABLE_ID
AND TABLES.GAME_ID = GAMES.GAME_ID
GROUP BY GAME_NAME;
```



#	GAME_NAME	ABSOLUTE_AMOUNT
1	BACCARAT	13.5000
2	BLACKJACK	5.2500
3	CRAPS	4.3333
4	ROULETTE	11.0000
5	TEXAS HOLD'EM POKER	5.2000

3) Which games favour purchasing drinks? Assuming that drink payment is only done by cash.

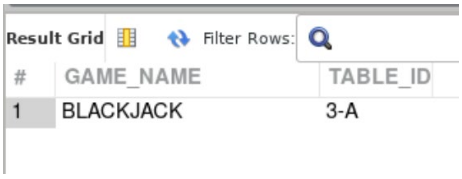
```
SELECT SUM(DRINK.AMOUNT) AS TOTAL_DRINKS,GAME_NAME
FROM GAMES, TABLES, GAMES_ROUNDS, DRINKS_TICKET, DRINK
WHERE GAMES.GAME_ID = TABLES.GAME_ID
AND TABLES.GAME_ID=GAMES_ROUNDS.GAME_ROUND_ID
AND GAMES_ROUNDS.GAME_ROUND_ID = DRINKS_TICKET.GAME_ROUND_ID
AND DRINKS_TICKET.TICKET_ID = DRINK.TICKET_ID
GROUP BY GAME_NAME
ORDER BY SUM(DRINK.AMOUNT) DESC;
```



#	TOTAL_DRINKS	GAME_NAME
1	60	TEXAS HOLD'EM POKER
2	52	CRAPS
3	36	ROULETTE
4	32	BLACKJACK

4) What is the easiest game to win money, and at which table? Supposing that each table begins each day with an amount of chips.

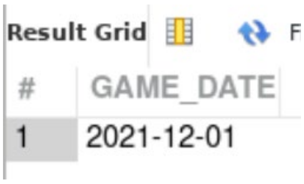
```
SELECT GAME_NAME, TABLES.TABLE_ID
FROM GAMES, TABLES
WHERE GAMES.GAME_ID = TABLES.GAME_ID
AND GAME_NAME = (SELECT GAME_NAME FROM GAMES, TABLES, GAMES_ROUNDS, STARTING_AMOUNT,
ENDING_AMOUNT
WHERE GAMES.GAME_ID = TABLES.GAME_ID
AND TABLES.TABLE_ID = GAMES_ROUNDS.TABLE_ID
AND GAMES_ROUNDS.GAME_ROUND_ID = ENDING_AMOUNT.GAME_ROUND_ID
AND GAMES_ROUNDS.GAME_ROUND_ID = STARTING_AMOUNT.GAME_ROUND_ID
GROUP BY GAME_NAME
HAVING SUM(ENDING_AMOUNT.CHIP_ID*ENDING_AMOUNT.QUANTITY) -
SUM(STARTING_AMOUNT.CHIP_ID*STARTING_AMOUNT.QUANTITY)
>= ALL(SELECT SUM(ENDING_AMOUNT.CHIP_ID*ENDING_AMOUNT.QUANTITY) -
SUM(STARTING_AMOUNT.CHIP_ID*STARTING_AMOUNT.QUANTITY)
FROM STARTING_AMOUNT, ENDING_AMOUNT, GAMES_ROUNDS, TABLES, GAMES
WHERE GAMES_ROUNDS.GAME_ROUND_ID = ENDING_AMOUNT.GAME_ROUND_ID
AND GAMES_ROUNDS.GAME_ROUND_ID = STARTING_AMOUNT.GAME_ROUND_ID
AND GAMES.GAME_ID = TABLES.GAME_ID
AND TABLES.TABLE_ID = GAMES_ROUNDS.TABLE_ID
GROUP BY GAME_NAME))
AND TABLES.TABLE_ID = (SELECT TABLES.TABLE_ID FROM TABLES, GAMES_ROUNDS, STARTING_AMOUNT,
ENDING_AMOUNT
WHERE TABLES.TABLE_ID = GAMES_ROUNDS.TABLE_ID
AND GAMES_ROUNDS.GAME_ROUND_ID = ENDING_AMOUNT.GAME_ROUND_ID
AND GAMES_ROUNDS.GAME_ROUND_ID = STARTING_AMOUNT.GAME_ROUND_ID
AND GAME_NAME = (SELECT GAME_NAME FROM GAMES, TABLES, GAMES_ROUNDS, STARTING_AMOUNT,
ENDING_AMOUNT
WHERE GAMES.GAME_ID = TABLES.GAME_ID
AND TABLES.TABLE_ID = GAMES_ROUNDS.TABLE_ID
AND GAMES_ROUNDS.GAME_ROUND_ID = ENDING_AMOUNT.GAME_ROUND_ID
AND GAMES_ROUNDS.GAME_ROUND_ID = STARTING_AMOUNT.GAME_ROUND_ID
GROUP BY GAME_NAME
HAVING SUM(ENDING_AMOUNT.CHIP_ID*ENDING_AMOUNT.QUANTITY) -
SUM(STARTING_AMOUNT.CHIP_ID*STARTING_AMOUNT.QUANTITY)
>= ALL(SELECT SUM(ENDING_AMOUNT.CHIP_ID*ENDING_AMOUNT.QUANTITY) -
SUM(STARTING_AMOUNT.CHIP_ID*STARTING_AMOUNT.QUANTITY)
FROM STARTING_AMOUNT, ENDING_AMOUNT, GAMES_ROUNDS, TABLES, GAMES
WHERE GAMES_ROUNDS.GAME_ROUND_ID = ENDING_AMOUNT.GAME_ROUND_ID
AND GAMES_ROUNDS.GAME_ROUND_ID = STARTING_AMOUNT.GAME_ROUND_ID
AND GAMES.GAME_ID = TABLES.GAME_ID
AND TABLES.TABLE_ID = GAMES_ROUNDS.TABLE_ID
GROUP BY GAME_NAME))
GROUP BY TABLES.TABLE_ID
HAVING SUM(ENDING_AMOUNT.CHIP_ID*ENDING_AMOUNT.QUANTITY) -
SUM(STARTING_AMOUNT.CHIP_ID*STARTING_AMOUNT.QUANTITY)
>= ALL(SELECT SUM(ENDING_AMOUNT.CHIP_ID*ENDING_AMOUNT.QUANTITY) -
SUM(STARTING_AMOUNT.CHIP_ID*STARTING_AMOUNT.QUANTITY)
FROM STARTING_AMOUNT, ENDING_AMOUNT, GAMES_ROUNDS
WHERE GAMES_ROUNDS.GAME_ROUND_ID = ENDING_AMOUNT.GAME_ROUND_ID
AND GAMES_ROUNDS.GAME_ROUND_ID = STARTING_AMOUNT.GAME_ROUND_ID
GROUP BY GAMES_ROUNDS.TABLE_ID));
```



#	GAME_NAME	TABLE_ID
1	BLACKJACK	3-A

5) Which was the most crowded day at the casino?

```
SELECT GAME_DATE
FROM TABLE_ENTRIES
GROUP BY GAME_DATE
HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM TABLE_ENTRIES
GROUP BY GAME_DATE);
```



#	GAME_DATE
1	2021-12-01

Data Definition Language (DDL) used:

```
CREATE DATABASE CASINO;

USE CASINO;

CREATE TABLE GAMES
(GAME_ID INT PRIMARY KEY NOT NULL,
GAME_NAME CHAR(20));

CREATE TABLE TABLES
(TABLE_ID CHAR (10) PRIMARY KEY NOT NULL,
GAME_ID INT);
ALTER TABLE TABLES
        ADD FOREIGN KEY (GAME_ID)
  REFERENCES GAMES(GAME_ID);

CREATE TABLE JOB_TYPE
(JOB_TYPE_ID VARCHAR(10) PRIMARY KEY NOT NULL,
DESCRIPTION VARCHAR (20));

CREATE TABLE EMPLOYEES
(EMPLOYEE_ID VARCHAR(15) PRIMARY KEY NOT NULL,
NAME VARCHAR(20),
JOB_TYPE_ID VARCHAR(10),
ADDRESS VARCHAR (50),
TELEPHONE VARCHAR(20),
SALARY BIGINT);
ALTER TABLE EMPLOYEES
        ADD FOREIGN KEY (JOB_TYPE_ID)
  REFERENCES JOB_TYPE (JOB_TYPE_ID);
```

```
CREATE TABLE GAMES_ROUNDS
(GAME_ROUND_ID VARCHAR(500) NOT NULL,
TABLE_ID CHAR (10) NOT NULL,
GAME_DATE DATE NOT NULL,
EMPLOYEE_ID VARCHAR(15),
PRIMARY KEY(GAME_ROUND_ID,TABLE_ID,GAME_DATE));
ALTER TABLE GAMES_ROUNDS
        ADD FOREIGN KEY (TABLE_ID)
  REFERENCES TABLES(TABLE_ID);
  ALTER TABLE GAMES_ROUNDS
        ADD FOREIGN KEY (EMPLOYEE_ID)
  REFERENCES EMPLOYEES (EMPLOYEE_ID);

CREATE TABLE CHIPS
(CHIP_ID INT PRIMARY KEY NOT NULL,
COLOUR CHAR (10));

CREATE TABLE STARTING_AMOUNT
(GAME_ROUND_ID VARCHAR(500),
TABLE_ID CHAR (10),
GAME_DATE DATE,
CHIP_ID INT,
QUANTITY INT,
PRIMARY KEY(GAME_ROUND_ID, CHIP_ID));
ALTER TABLE STARTING_AMOUNT
        ADD FOREIGN KEY (GAME_ROUND_ID,TABLE_ID,GAME_DATE)
  REFERENCES GAMES_ROUNDS(GAME_ROUND_ID,TABLE_ID,GAME_DATE);
ALTER TABLE STARTING_AMOUNT
  ADD FOREIGN KEY (CHIP_ID)
  REFERENCES CHIPS (CHIP_ID);
```

```
CREATE TABLE ENDING_AMOUNT
(GAME_ROUND_ID VARCHAR(500),
TABLE_ID CHAR (10),
GAME_DATE DATE,
CHIP_ID INT,
QUANTITY INT,
PRIMARY KEY(GAME_ROUND_ID, CHIP_ID));
ALTER TABLE ENDING_AMOUNT
        ADD FOREIGN KEY (GAME_ROUND_ID,TABLE_ID,GAME_DATE)
         REFERENCES GAMES_ROUNDS(GAME_ROUND_ID,TABLE_ID,GAME_DATE);
ALTER TABLE ENDING_AMOUNT
          ADD FOREIGN KEY (CHIP_ID)
          REFERENCES CHIPS (CHIP_ID);

CREATE TABLE PLAYER
(PLAYER_ID INT PRIMARY KEY NOT NULL,
NAME VARCHAR(30),
SURNAME VARCHAR(30),
TELEPHONE VARCHAR(30),
DATE_OF_BIRTH DATE);

CREATE TABLE TABLE_ENTRIES
(PLAYER_ID INT,
GAME_ROUND_ID CHAR(10),
TABLE_ID CHAR (10),
GAME_DATE DATE,
TIME TIME);
ALTER TABLE TABLE_ENTRIES
        ADD FOREIGN KEY (PLAYER_ID)
         REFERENCES PLAYER(PLAYER_ID);
ALTER TABLE TABLE_ENTRIES
        ADD FOREIGN KEY (GAME_ROUND_ID,TABLE_ID,GAME_DATE)
```

```
            REFERENCES GAMES_ROUNDS(GAME_ROUND_ID,TABLE_ID,GAME_DATE);
ALTER TABLE TABLE_ENTRIES
            ADD PRIMARY KEY(PLAYER_ID,GAME_ROUND_ID,TIME);

CREATE TABLE DRINKS_TICKET
(TICKET_ID BIGINT PRIMARY KEY NOT NULL,
GAME_ROUND_ID CHAR(10),
TABLE_ID CHAR (10),
GAME_DATE DATE,
TIME TIME,
EMPLOYEE_ID VARCHAR(20));
ALTER TABLE DRINKS_TICKET
            ADD FOREIGN KEY (GAME_ROUND_ID,TABLE_ID,GAME_DATE)
             REFERENCES GAMES_ROUNDS(GAME_ROUND_ID,TABLE_ID,GAME_DATE);
   ALTER TABLE DRINKS_TICKET
            ADD FOREIGN KEY (EMPLOYEE_ID)
              REFERENCES EMPLOYEES (EMPLOYEE_ID);

CREATE TABLE PAYMENT_METHOD
(PAYMENT_ID BIGINT(20) PRIMARY KEY NOT NULL,
DESCRIPTION VARCHAR(40));

CREATE TABLE PAYMENT_CC
(PAYMENT_ID BIGINT(20) PRIMARY KEY NOT NULL,
CARD_AMOUNT BIGINT(40),
CARD_NUMBER VARCHAR (64),
BANK_NAME VARCHAR(64),
EXPIRARY_DATE CHAR(15));
```

```
CREATE TABLE PAYMENT_CS
(PAYMENT_ID BIGINT (20) PRIMARY KEY NOT NULL,
CASH_AMOUNT BIGINT(40));

ALTER TABLE PAYMENT_METHOD
            ADD FOREIGN KEY (PAYMENT_ID)
             REFERENCES PAYMENT_CC (PAYMENT_ID),
             ADD FOREIGN KEY (PAYMENT_ID)
             REFERENCES PAYMENT_CS (PAYMENT_ID);

CREATE TABLE TYPES_OF_DRINKS
(DRINK_ID CHAR(10) PRIMARY KEY NOT NULL,
NAME CHAR(30),
PRICE DECIMAL,
PAYMENT_ID BIGINT(20));
ALTER TABLE TYPES_OF_DRINKS
            ADD FOREIGN KEY (PAYMENT_ID)
             REFERENCES PAYMENT_METHOD(PAYMENT_ID);

CREATE TABLE DRINK
(TICKET_ID BIGINT REFERENCES DRINKS_TICKET(GAME_ROUND_ID),
DRINK_ID CHAR(10) REFERENCES TYPES_OF_DRINKS(DRINK_ID),
AMOUNT INT);
ALTER TABLE DRINK
            ADD FOREIGN KEY (TICKET_ID)
             REFERENCES DRINKS_TICKET(TICKET_ID);
   ALTER TABLE DRINK
            ADD FOREIGN KEY (DRINK_ID)
             REFERENCES TYPES_OF_DRINKS(DRINK_ID);
ALTER TABLE DRINK
            ADD PRIMARY KEY(TICKET_ID,DRINK_ID);
```

Data Manipulation Language (DML) used:

```sql
USE CASINO;

INSERT INTO GAMES

VALUES

(1, "TEXAS HOLD'EM POKER"),

(2, "ROULETTE"),

(3, "BLACKJACK"),

(4, "CRAPS"),

(5, "BACCARAT");

INSERT INTO TABLES

VALUES

("1-A", 1),

("1-B", 1),

("1-C", 1),

("1-D", 1),

("2-A", 2),

("2-B", 2),

("2-C", 2),

("2-D", 2),

("3-A", 3),

("3-B", 3),

("3-C", 3),

("3-D", 3),

("4-A", 4),

("4-B", 4),

("4-C", 4),

("4-D", 4),

("5-A", 5),

("5-B", 5),

("5-C", 5),

("5-D", 5);
```

```
INSERT INTO GAMES_ROUNDS (GAME_ROUND_ID, TABLE_ID, GAME_DATE)

VALUES

("1-A-3", "1-A", "2021-06-12"),

("1-A-4", "1-A", "2021-05-06"),

("2-A-2", "2-A", "2021-12-01"),

("2-A-3", "2-A", "2021-05-06"),

("3-A-7", "3-A", "2021-12-01"),

("3-A-1", "3-A", "2021-06-12"),

("3-A-8", "3-A", "2021-05-06"),

("2-B-2", "2-B", "2021-12-01"),

("2-B-3", "2-B", "2021-05-06"),

("4-B-5", "4-B", "2021-12-01"),

("4-C-3", "4-C", "2021-06-11"),

("1-D-4", "1-D", "2021-05-16"),

("2-C-2", "2-C", "2021-12-01"),

("2-C-3", "2-C", "2021-06-26"),

("5-D-2", "5-D", "2021-12-21"),

("5-B-3", "5-B", "2021-06-12"),

("3-B-4", "3-B", "2021-08-06"),

("1-A-7", "1-A", "2021-12-01"),

("2-B-4", "2-B", "2021-05-06"),

("1-B-2", "1-B", "2021-12-01"),

("1-D-5", "1-D", "2021-07-12"),

("4-C-5", "4-C", "2021-07-06"),

("2-D-4", "2-D", "2021-12-01"),

("2-D-3", "2-D", "2021-02-06");
```

```
INSERT INTO CHIPS
VALUES
(1, "WHITE"),
(5, "RED"),
(10, "BLUE"),
(25, "ORANGE"),
(50, "BLACK"),
(100, "PURPLE");

INSERT INTO STARTING_AMOUNT
VALUES
("1-A-3", "1-A", "2021-06-12", 1, 5),
("1-A-4", "1-A", "2021-05-06", 25, 5),
("2-A-2", "2-A", "2021-12-01", 25, 10),
("2-A-3", "2-A", "2021-05-06", 50, 15),
("3-A-7", "3-A", "2021-12-01", 100, 4),
("3-A-7", "3-A", "2021-12-01", 50, 3),
("3-A-1", "3-A", "2021-06-12", 10, 5),
("3-A-1", "3-A", "2021-06-12", 50, 6),
("3-A-8", "3-A", "2021-05-06", 100, 5),
("2-B-2", "2-B", "2021-12-01", 1, 20),
("2-B-3", "2-B", "2021-05-06", 25, 5),
("4-B-5", "4-B", "2021-12-01", 50, 8),
("4-C-3", "4-C", "2021-06-11", 100, 4),
("4-C-3", "4-C", "2021-06-11", 25, 6),
("1-D-4", "1-D", "2021-05-16", 25, 10),
("2-C-2", "2-C", "2021-12-01", 10, 10),
("2-C-3", "2-C", "2021-06-26", 50, 4),
("5-D-2", "5-D", "2021-12-21", 10, 8),
("5-D-2", "5-D", "2021-12-21", 1, 15),
("5-D-2", "5-D", "2021-12-21", 100, 3),
("5-B-3", "5-B", "2021-06-12", 50, 2),
("3-B-4", "3-B", "2021-08-06", 10, 8),
("1-A-7", "1-A", "2021-12-01", 100, 2),
("2-B-4", "2-B", "2021-05-06", 25, 10),
("1-B-2", "1-B", "2021-12-01", 50, 4),
("1-B-2", "1-B", "2021-12-01", 10, 5),
("1-B-2", "1-B", "2021-12-01", 100, 4),
("1-B-2", "1-B", "2021-12-01", 25, 3),
("1-D-5", "1-D", "2021-07-12", 50, 6),
("4-C-5", "4-C", "2021-07-06", 1 ,5),
("2-D-4", "2-D", "2021-12-01", 50, 5),
("2-D-3", "2-D", "2021-02-06", 100, 4);

INSERT INTO ENDING_AMOUNT
VALUES
("1-A-3", "1-A", "2021-06-12", 1, 2),
("1-A-4", "1-A", "2021-05-06", 50, 4),
("2-A-2", "2-A", "2021-12-01", 25, 0),
("2-A-3", "2-A", "2021-05-06", 25, 5),
("3-A-7", "3-A", "2021-12-01", 100, 5),
("3-A-7", "3-A", "2021-12-01", 50, 10),
("3-A-1", "3-A", "2021-06-12", 10, 10),
("3-A-1", "3-A", "2021-06-12", 100, 7),
("3-A-8", "3-A", "2021-05-06", 100, 1),
("2-B-2", "2-B", "2021-12-01", 25, 10),
("2-B-3", "2-B", "2021-05-06", 25, 8),
("4-B-5", "4-B", "2021-12-01", 25, 10),
("4-C-3", "4-C", "2021-06-11", 100, 4),
("4-C-3", "4-C", "2021-06-11", 25, 9),
("1-D-4", "1-D", "2021-05-16", 25, 11),
("2-C-2", "2-C", "2021-12-01", 10, 4),
("2-C-3", "2-C", "2021-06-26", 50, 1),

("5-D-2", "5-D", "2021-12-21", 10, 10),

("5-D-2", "5-D", "2021-12-21", 1, 5),

("5-D-2", "5-D", "2021-12-21", 100, 1),

("5-B-3", "5-B", "2021-06-12", 50, 5),

("3-B-4", "3-B", "2021-08-06", 10, 5),

("1-A-7", "1-A", "2021-12-01", 100, 1),

("2-B-4", "2-B", "2021-05-06", 10, 6),

("1-B-2", "1-B", "2021-12-01", 25, 10),

("1-B-2", "1-B", "2021-12-01", 10, 8),

("1-B-2", "1-B", "2021-12-01", 100, 0),

("1-B-2", "1-B", "2021-12-01", 50, 6),

("1-D-5", "1-D", "2021-07-12", 100, 4),

("4-C-5", "4-C", "2021-07-06", 1 ,10),

("2-D-4", "2-D", "2021-12-01", 25, 3),

("2-D-3", "2-D", "2021-02-06", 100, 2);


INSERT INTO PLAYER (PLAYER_ID, DATE_OF_BIRTH)

VALUES

(1001, "1990-05-06"),

(1002, "1950-04-18"),

(1003, "1985-03-20"),

(1004, "1975-02-24"),

(1005, "1960-11-26"),

(1006, "1961-11-26"),

(1007, "1991-05-06"),

(1008, "1960-04-18"),

(1009, "1985-03-20"),

(1010, "1995-02-24"),

(1011, "1999-11-26"),

(1012, "1980-11-26"),
(1013, "1995-05-06"),
(1014, "1978-04-18"),
(1015, "1985-03-20"),
(1016, "1979-02-24"),
(1017, "1968-11-26"),
(1018, "1960-11-26"),
(1019, "1988-11-26"),
(1020, "1950-11-26");

INSERT INTO TABLE_ENTRIES
VALUES
(1001, "1-A-3", "1-A", "2021-06-12", "12:05:25"),
(1002,"1-A-4", "1-A", "2021-05-06", "17:07:20"),
(1003, "2-A-2", "2-A", "2021-12-01", "20:08:50"),
(1004,"2-A-3", "2-A", "2021-05-06", "22:06:55"),
(1005, "3-A-7", "3-A", "2021-12-01", "23:07:30"),
(1006, "3-A-1", "3-A", "2021-06-12", "03:10:30"),
(1007,"3-A-8", "3-A", "2021-05-06", "02:07:30"),
(1008,"2-B-2", "2-B", "2021-12-01", "21:15:30"),
(1009,"2-B-3", "2-B", "2021-05-06", "19:07:20"),
(1010, "4-B-5", "4-B", "2021-12-01","19:50:23"),
(1011, "4-C-3", "4-C", "2021-06-11", "01:25:20"),
(1012, "1-D-4", "1-D", "2021-05-16", "23:07:20"),
(1013, "2-C-2", "2-C", "2021-12-01", "02:07:20"),
(1014, "2-C-3", "2-C", "2021-06-26","19:27:20" ),
(1015, "5-D-2", "5-D", "2021-12-21", "00:07:20"),
(1016, "5-B-3", "5-B", "2021-06-12", "19:57:20"),
(1017, "3-B-4", "3-B", "2021-08-06", "23:07:20"),
(1018, "1-A-7", "1-A", "2021-12-01", "22:07:20"),
(1019, "2-B-4", "2-B", "2021-05-06", "23:22:21"),
(1020,"1-B-2", "1-B", "2021-12-01", "20:07:20"),
```

```sql
(1020,"1-D-5", "1-D", "2021-07-12", "22:07:20"),
(1018,"4-C-5", "4-C", "2021-07-06", "18:07:20"),
(1005,"2-D-4", "2-D", "2021-12-01", "22:07:20"),
(1019,"2-D-3", "2-D", "2021-02-06", "05:07:20");

INSERT INTO DRINKS_TICKET (TICKET_ID, GAME_ROUND_ID, TABLE_ID, GAME_DATE, TIME)
VALUES
(2201, "1-A-3", "1-A", "2021-06-12", "12:05:25"),
(4502,"1-A-4", "1-A", "2021-05-06", "17:07:20"),
(6603, "2-A-2", "2-A", "2021-12-01", "20:08:50"),
(5504,"2-A-3", "2-A", "2021-05-06", "22:06:55"),
(7705, "3-A-7", "3-A", "2021-12-01", "23:07:30"),
(9006, "3-A-1", "3-A", "2021-06-12", "03:10:30"),
(4507,"3-A-8", "3-A", "2021-05-06", "02:07:30"),
(2308,"2-B-2", "2-B", "2021-12-01", "21:15:30"),
(2209,"2-B-3", "2-B", "2021-05-06", "19:07:20"),
(4410, "4-B-5", "4-B", "2021-12-01","19:50:23"),
(3311, "4-C-3", "4-C", "2021-06-11", "01:25:20"),
(1112, "1-D-4", "1-D", "2021-05-16", "23:07:20"),
(8813, "2-C-2", "2-C", "2021-12-01", "02:07:20"),
(9914, "2-C-3", "2-C", "2021-06-26","19:27:20" ),
(1815, "5-D-2", "5-D", "2021-12-21", "00:07:20"),
(8916, "5-B-3", "5-B", "2021-06-12", "19:57:20"),
(9817, "3-B-4", "3-B", "2021-08-06", "23:07:20"),
(7818, "1-A-7", "1-A", "2021-12-01", "22:07:20"),
(8719, "2-B-4", "2-B", "2021-05-06", "23:22:21"),
(7620,"1-B-2", "1-B", "2021-12-01", "20:07:20"),
(1920,"1-D-5", "1-D", "2021-07-12", "22:07:20"),
(5618,"4-C-5", "4-C", "2021-07-06", "18:07:20"),
(7505,"2-D-4", "2-D", "2021-12-01", "22:07:20"),
(3419,"2-D-3", "2-D", "2021-02-06", "05:07:20");
```

```
INSERT INTO TYPES_OF_DRINKS (DRINK_ID, NAME, PRICE)
VALUES
("B1001", "BEER", 3),
("C1005", "CIDER", 4),
("H1001", "WATER", 2),
("W1001", "WHISKEY", 3),
("I0505", "VODKA" , 6),
("R999", "RUM", 6),
("Y9090", "SPICEDRUM", 5);

INSERT INTO DRINK
VALUES
(9006, "B1001", 3),
(7620, "C1005", 5),
(5504, "C1005", 1),
(1920, "H1001", 3),
(5618, "W1001", 5),
(2201, "R999", 4),
(6603, "Y9090", 3),
(3419, "W1001", 2),
(4507, "W1001", 5),
(3311, "I0505", 8),
(1112, "W1001", 3),
(8813, "C1005", 3);
```