DDL SQL – creation of the tables

GAMES

Create table games (game_id char(2) not null primary key, game_desc varchar(20) not null)

We will create a table for the different games with the previously shown formula. Each game will be associated with a unique 2 letter id, hence the char(2).

```
db2 => describe table games
```

```
                                Data type                       Column
Column name                     schema    Data type name        Length     Scale Nulls
------------------------------- --------- ------------------- ---------- ----- ------
GAME_ID                         SYSIBM    CHARACTER                    2     0 No
GAME_DESC                       SYSIBM    VARCHAR                     20     0 No

  2 record(s) selected.
```

TABLES

Create table tables (table_id char(5) not null primary key, game_id char(2) not null references games (game_id), seats integer default 0)

At the casino the games might be played on multiple tables, hence we need to create a table with unique 5 alphanumeric values as its id. Each table will also record the seat number, as long as the table has seats, hence nulls can be recorded in this column. game_id will be our foreign key.

```
db2 => describe table tables
```

```
                                Data type                       Column
Column name                     schema    Data type name        Length     Scale Nulls
------------------------------- --------- ------------------- ---------- ----- ------
TABLE_ID                        SYSIBM    CHARACTER                    5     0 No
GAME_ID                         SYSIBM    CHARACTER                    2     0 No
SEATS                           SYSIBM    INTEGER                      4     0 Yes

  3 record(s) selected.
```

DRINKS

Create table drinks (drink_id char(2) not null primary key, drink_desc varchar(20) not null, drink_price integer not null)

We also need a table for the drinks. Here we will associate the drink name to a unique 2 letter id and its price.

```
db2 => describe table drinks
```

```
                                Data type                       Column
Column name                     schema    Data type name        Length     Scale Nulls
------------------------------- --------- ------------------- ---------- ----- ------
DRINK_ID                        SYSIBM    CHARACTER                    2     0 No
DRINK_DESC                      SYSIBM    VARCHAR                     20     0 No
DRINK_PRICE                     SYSIBM    INTEGER                      4     0 No

  3 record(s) selected.
```

TABLE_DAILY_CUSTOMER

Create table table_daily_customer (game_date date not null, table_id char(5) not null references tables (table_id), constraint constraint_name primary key (game_date, table_id), customer_count integer not null default 0, round_count integer not null default 0)

Here we wanted to record the number of customers at each table each day, as well as the number of rounds played. This allows us to build statistics on customer winnings and how busy the tables get in a day.

```
db2 => describe table table_daily_customer
```

Column name	Data type schema	Data type name	Column Length	Scale	Nulls
GAME_DATE	SYSIBM	DATE	4	0	No
TABLE_ID	SYSIBM	CHARACTER	5	0	No
CUSTOMER_COUNT	SYSIBM	INTEGER	4	0	No
ROUND_COUNT	SYSIBM	INTEGER	4	0	No

```
  4 record(s) selected.
```

TABLE_DAILY_CHIPS

Create table table_daily_chips (game_date date not null, table_id char(5) not null references tables (table_id), chips_value integer not null, constraint constraint_name primary key (game_date, table_id, chips_value), chips_start integer not null default 0, amount_end integer not null default 0)

To record daily statistics in the casino we need to create a table where we record chips movement, for all the chips values. Hence, we created a table that records as a primary key a combination of day, table and chips value, then we recorded the quantity at the beginning and at the end of the day, that allowed us to measure the daily movement for each chips value each day.

```
db2 => describe table table_daily_chips
```

Column name	Data type schema	Data type name	Column Length	Scale	Nulls
GAME_DATE	SYSIBM	DATE	4	0	No
TABLE_ID	SYSIBM	CHARACTER	5	0	No
CHIPS_VALUE	SYSIBM	INTEGER	4	0	No
CHIPS_START	SYSIBM	INTEGER	4	0	No
AMOUNT_END	SYSIBM	INTEGER	4	0	No

```
  5 record(s) selected.
```

DRINK_TICKET

Create Table drink_ticket (ticket_no integer not null primary key, table_id char(5) not null references tables (table_id), drink_id char(2) not null references drinks (drink_id), qty integer not null, total_price integer not null, date date not null)

To create a table for the tickets, we need to consider the previously created tables, “drinks” and “tables”, our foreign keys will be drink_id and table_id. Here we will record the quantity ordered for each drink at each table, we will also show the date of each order in order to calculate daily averages. Furthermore, we will also need to have the total price for each ticket in order to understand the revenue.

```
db2 => describe table drink_ticket
```

Column name	Data type schema	Data type name	Column Length	Scale	Nulls
TICKET_NO	SYSIBM	INTEGER	4	0	No
TABLE_ID	SYSIBM	CHARACTER	5	0	No
DRINK_ID	SYSIBM	CHARACTER	2	0	No
QTY	SYSIBM	INTEGER	4	0	No
TOTAL_PRICE	SYSIBM	INTEGER	4	0	No
DATE	SYSIBM	DATE	4	0	No

```
  6 record(s) selected.
```

DML SQL – adding values to the tables

GAMES ('game_id (2)', 'game')

Insert into games values ('pk','poker'),('bj','blackjack'),('rl','roulette'),('cr','craps')
Insert into games values ('bc', 'baccarat'),('ft', 'fan-tan')

DRINKS ('drink_id (2)', 'drink', price)

insert into drinks values ('gt','gin tonic',10),('wt','water',5),('ck','coke',8),('vc','vodka coke',10)
insert into drinks values ('rm', 'rhum',10), ('lm', 'lemonade',8),('wk','whiskey',10),('wc','whiskey cola',10),('vk','vodka',10),('rb','red bull',8),('vr', 'vodka red bull',10)

TABLES (table_id (5)', 'game_id (2)', seats)

Insert into tables values ('tb001', 'pk',6), ('tb002', 'pk',8), ('tb003', 'bj',5), ('tb004', 'bj',5), ('tb005', 'rl',default),('tb006', 'rl',default), ('tb007', 'rl', default), ('tb008', 'cr', default), ('tb009', 'pk',6), ('tb010', 'rl', default)
Insert into tables values ('tb011','bc',4),('tb012', 'ft', default),('tb013','pk',6),('tb014','rl', default),('tb015', 'bj',5)

DRINK_TICKET (ticket_id integer, 'table_id (5)', 'drink', qty, total price, 'date')

Insert into drink_ticket values (1, 'tb002', 'gt', 4, 40,'2019-11-24'),(2, 'tb007', 'vc', 3, 30, '2019-11-24'), (3, 'tb004', 'gt', 5, 50, '2019-11-24'), (4, 'tb009', 'ck', 2, 15,'2019-11-25'), (5, 'tb003', 'gt', 2, 20,'2019-11-25'), (6, 'tb007', 'gt', 1, 10,'2019-11-25'), (7, 'tb010', 'wt', 3, 15,'2019-11-25'), (9, 'tb006', 'vc', 3, 30,'2019-11-25'), (10, 'tb010','gt', 2, 20,'2019-11-25'), (13, 'tb002', 'vc', 4, 40,'2019-11-25'), (15, 'tb003', 'wt', 4, 20,'2019-11-25')
Insert into drink_ticket values (11, 'tb012', 'rb',5,40, '2019-11-26'), (12, 'tb012', 'vr',2,20, '2019-11-26'), (14, 'tb014', 'lm',3,24, '2019-11-25')

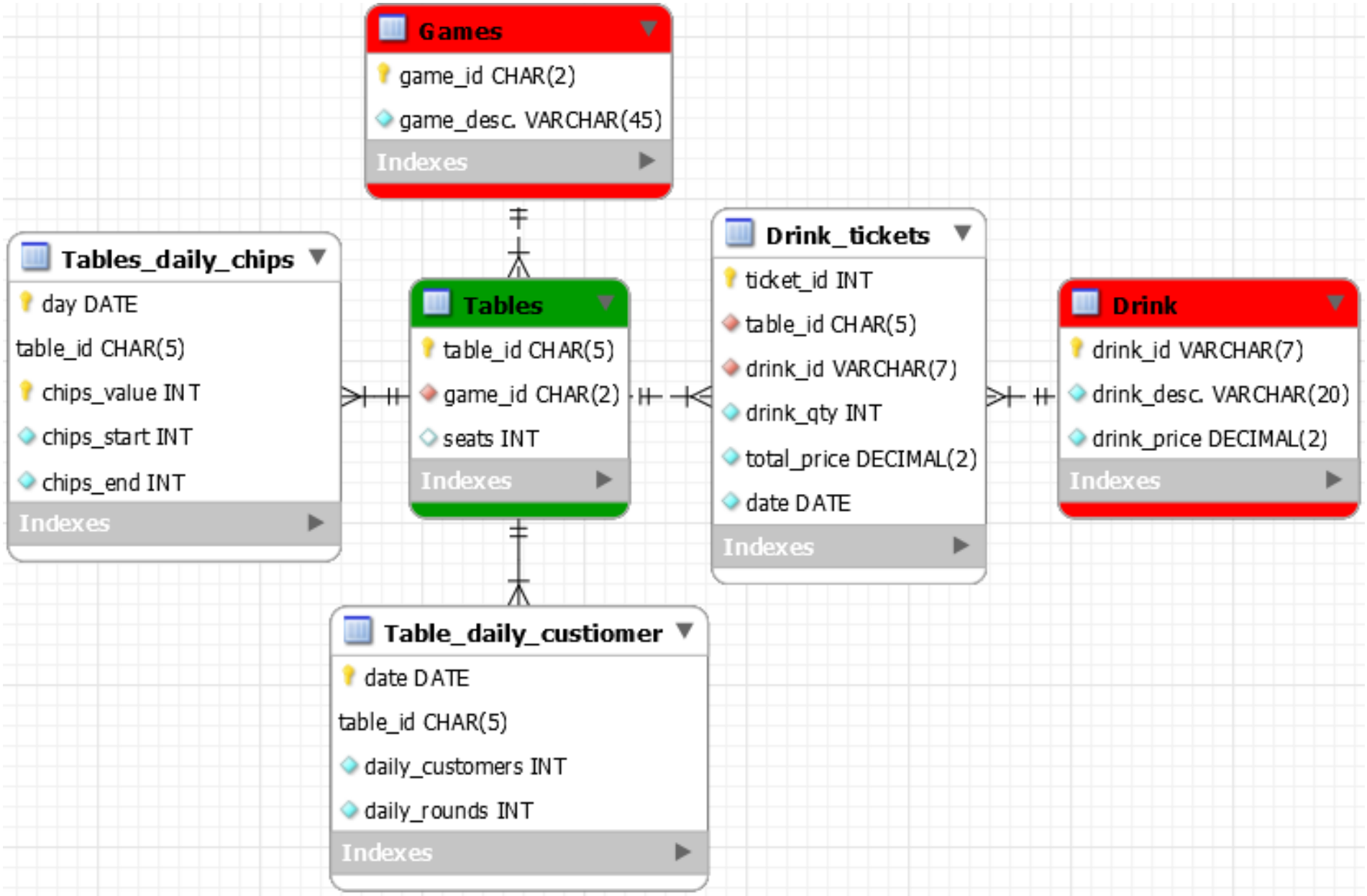
TABLE_DAILY_CUSTOMER ('date', 'table_id (5)',integer customer, integer games)

Insert into table_daily_customer values ('2019-11-25', 'tb001',74,15),('2019-11-25', 'tb002', 190,100), ('2019-11-25','tb003',400,100),('2019-11-25', 'tb004',1400 ,300), ('2019-11-25', 'tb005',485,100), ('2019-11-25', 'tb006', 4500,800), ('2019-11-25', 'tb007', 2000,600), ('2019-11-25', 'tb008', 300,80), ('2019-11-24', 'tb001',90,18),('2019-11-24', 'tb002', 162,82), ('2019-11-24', 'tb003', 467,102), ('2019-11-24', 'tb004',1482,275), ('2019-11-24', 'tb005', 3002,985), ('2019-11-24', 'tb006', 6027, 1200), ('2019-11-24', 'tb007',2086,604), ('2019-11-24', 'tb008',965,120), ('2019-11-24', 'tb009',875,150), ('2019-11-24', 'tb010',10721,1000)
Insert into table_daily_customer values ('2019-11-26', 'tb001',54,10),('2019-11-26', 'tb002', 172,87), ('2019-11-26','tb003',512,101),('2019-11-26', 'tb004',1629 ,312), ('2019-11-26', 'tb005',499,100), ('2019-11-26', 'tb006', 4499,798), ('2019-11-26', 'tb007', 2012,601), ('2019-11-26', 'tb008', 365,84), ('2019-11-23', 'tb001',89,18),('2019-11-23', 'tb002', 114,70), ('2019-11-23', 'tb003', 312,99), ('2019-11-23', 'tb004',1200,222), ('2019-11-23', 'tb005', 2202,895), ('2019-11-23', 'tb006', 3000, 500), ('2019-11-23', 'tb007',2100,605), ('2019-11-23', 'tb008',512,111), ('2019-11-23', 'tb009',800,140), ('2019-11-23', 'tb010',520,100)

TABLE_DAILY_CHIPS ('date', 'table_id (5)',integer value, integer qty start, integer qty end)

Insert into table_daily_chips values ('2019-11-25', 'tb001',20,40,50),('2019-11-25', 'tb001',50,120,101), ('2019-11-25', 'tb001',100,15,20),('2019-11-25', 'tb002',20,45,56),('2019-11-25', 'tb002',50,160,201), ('2019-11-25', 'tb002',100,46,79),('2019-11-25', 'tb003',20,6,42),('2019-11-25', 'tb003',50,57,46), ('2019-11-25', 'tb003',100,11,2),('2019-11-25', 'tb004',20,52,42),('2019-11-25', 'tb004',50,11,54), ('2019-11-25', 'tb004',100,67,105),('2019-11-25', 'tb005',20,120,106),('2019-11-25', 'tb005',50,12,14), ('2019-11-25', 'tb005',100,2,4),('2019-11-24', 'tb001',20,7,30),('2019-11-24', 'tb001',50,10,49), ('2019-11-24', 'tb001',100,30,70), ('2019-11-24', 'tb002',20,5,45), ('2019-11-24', 'tb002',50,4,60), ('2019-11-24', 'tb002',100,30,80), ('2019-11-24', 'tb003',20,10,55), ('2019-11-24', 'tb003',50,15,73), ('2019-11-24', 'tb003',100,10,68), ('2019-11-24', 'tb004',20,15,95), ('2019-11-24', 'tb004',50,28,90), ('2019-11-24', 'tb004',100,7,25), ('2019-11-24', 'tb005',20,10,100), ('2019-11-24', 'tb005',50,10,80), ('2019-11-24', 'tb005',100,10,80)
Insert into table_daily_chips values ('2019-11-25', 'tb006',20,120,150),('2019-11-25', 'tb006',50,50,300), ('2019-11-25', 'tb006',100,14,21),('2019-11-25', 'tb007',20,32,56),('2019-11-25', 'tb007',50,152,198), ('2019-11-25', 'tb007',100,42,65),('2019-11-25', 'tb008',20,57,43),('2019-11-25', 'tb008',50,45,46), ('2019-11-25', 'tb008',100,58,89),('2019-11-25', 'tb009',20,43,78),('2019-11-25', 'tb009',50,4,54), ('2019-11-25', 'tb009',100,65,60),('2019-11-25', 'tb010',20,122,106),('2019-11-25', 'tb010',50,43,43), ('2019-11-25', 'tb010',100,6,47),('2019-11-24', 'tb006',20,14,30),('2019-11-24', 'tb006',50,14,49), ('2019-11-24', 'tb006',100,43,76), ('2019-11-24', 'tb007',20,19,21), ('2019-11-24', 'tb007',50,7,9), ('2019-11-24', 'tb007',100,65,11), ('2019-11-24', 'tb008',20,43,40), ('2019-11-24', 'tb008',50,58,61), ('2019-11-24', 'tb008',100,96,69), ('2019-11-24', 'tb009',20,46,78), ('2019-11-24', 'tb009',50,45,65), ('2019-11-24', 'tb009',100,67,78), ('2019-11-24', 'tb010',20,11,9), ('2019-11-24', 'tb010',50,67,86), ('2019-11-24', 'tb010',100,76,40)

ERD – Entity Relationship Diagram



Answer to the questions – Creation of question 5

Qs 1 - Which are the top 3 demanding casino games?

The answer to Qs1 will be given by the highest number of customers playing a game. To get an answer to the question we had to connect the game_id to the customer_count via the only shared value: table_id. We used the following command to perform this calculation:

```
select b.game_id, sum(a.customer_count) / sum(a.round_count) as turnover_ratio from table_daily_customer a full outer join tables b on a.table_id = b.table_id group by b.game_id order by (sum(a.customer_count) / sum(a.round_count))
```

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

To obtain the average chips movement per game and per day we had to filter the chips movement each day, for each table and then assign it to the correct game. We used the following command in order to do so:

```
select b.game_id, sum(a.amount_end - a.chips_start) / count(*) as average_chips from table_daily_chips a full outer join tables b on a.table_id = b.table_id group by b.game_id
select game_date, sum(amount_end - chips_start) / count(*) as average_chips from table_daily_chips group by game_date
```

Qs 3 - Which games favors purchasing drinks? Let's suppose drink payment is done only by cash.

To understand what games favours the purchase of drinks we had performed the sum of the number of drinks, rather than the revenue. We had to consider again the relationship with the game, rather than the tables, together with the quantities ordered in each ticket. We performed this operation by using the following:

```
select b.game_id, sum(a.qty) as total_drinks from drink_ticket a full outer join tables b on a.table_id = b.table_id group by b.game_id
```

Qs 4 - What is the easiest game to win money and at which table? Let's suppose that each table begins each day with an amount of chips.

The easiest game for the casino to make money, consider revenue, rather than the chips average, as previously calculated. We assumed we only had 3 different valued chips. We needed the different revenue for the games, as wells as tables, so, we needed the chips amount and number played in a day, together with the game. They were, once again merged considering the table_id as the key, using these commands:

```
alter table table_daily_chips add Balance integer

update table_daily_chips set Balance = (amount_end - chips_start) * chips_value

select b.game_id, sum(a.balance) as payout from table_daily_chips a full outer join tables b on a.table_id = b.table_id group by b.game_id order by (sum(a.balance)

select a.table_id, b.game_id, sum(a.balance) as payout from table_daily_chips as a, tables as b where a.table_id = b.table_id group by a.table_id, b.game_id order by (sum(a.balance)
```

Qs 5 - What are the big betting tables.

It is also important for the casinos, not only what are the best winning games, but where the customers bet money, in order to understand how to better decide the chips to assign every day to the tables. To do so we have to merge the 2 different tables together, by date and by table_id in order to then obtain the desired results, by performing the following:

```
select a.game_date, a.table_id, avg(a.balance / b.customer_count) as winnings_per_customer from table_daily_chips as a, table_daily_customer as b where a.table_id = b.table_id and a.game_date = b.game_date group by a.table_id, a.game_date order by (avg(a.balance / b.customer_count))
```