

Name: Jinho Nam, Brady Wheelhouse

Part I. Project Definition, Data Dictionary and Entity Relationship Diagram

Abstract:



Paymo is a new money transfer company that needs a database for their operations. The database will need to track clients, senders, receivers, benefits, cards and bank accounts used, and transactions.

Our clients are anyone who wants to send money to another person. They will have the ability to add a card number or bank account to send or receive their money to. Also, clients will have the ability to earn and use benefits earning cashback on their transactions.

Client. Our client data will consist of a unique identifier for each client, as-well-as, their last name, first name, current balance, phone number, address, and email.

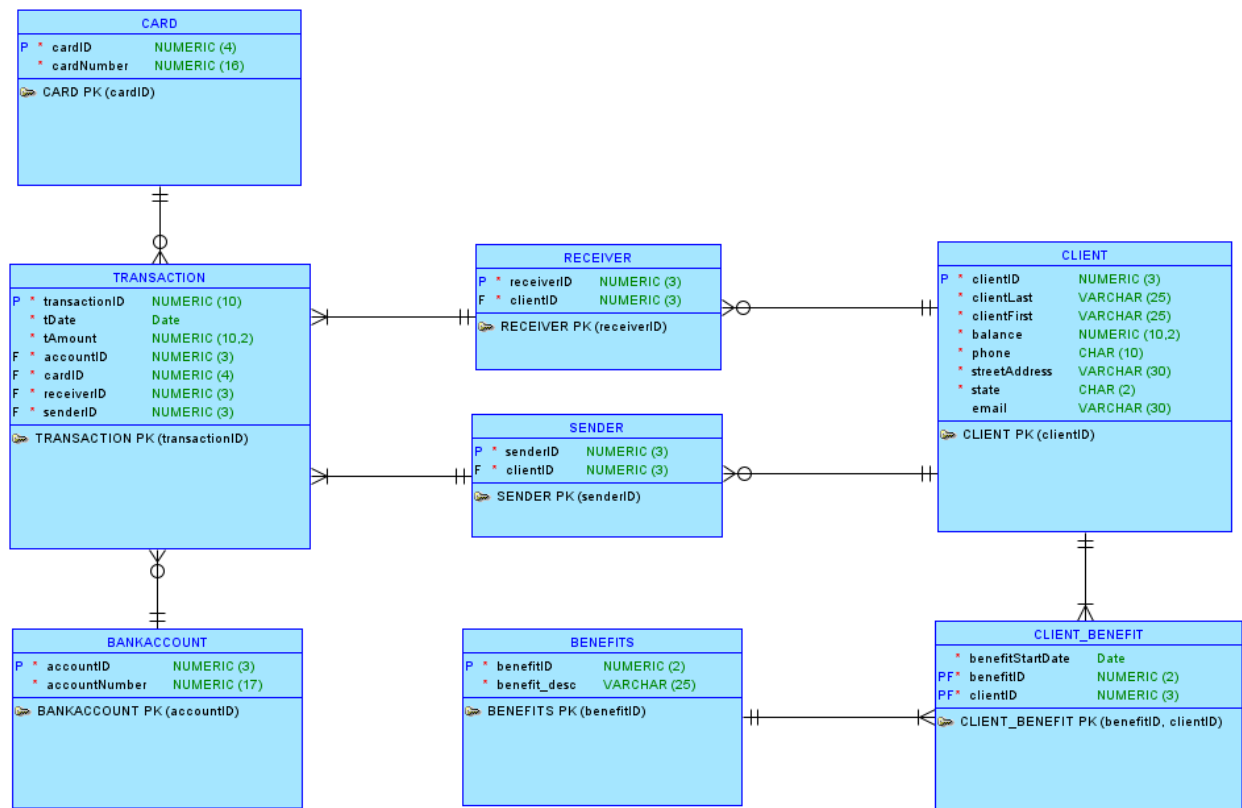
The card and bank account will be very similar. Both will have a unique identifier for the bank account or card and the account number.

Benefits. The benefit data will have a unique identifier for each benefit and a description of each benefit. The benefits will consist of a free instant transfer and cashback amounts from 0.5, 1, 2, and 3 percent cash back.

Sender and Receiver. The sender and receiver tables will link the clients to a transaction allowing the transaction data to contain both a sender and a receiver.

Client Benefit. The client benefit table will link benefits to clients that have the benefits.

Transaction. The transaction table will contain all the data regarding when the transaction took place, who sent and who received the money, and how much money was sent. This table will consist of a unique identifier for each transaction, the transfer date, transaction amount, the sender, the receiver, and the card or account that was used to send the money.



	A	B	C	D	E
1	Payroll management System List of Assumptions				
2	- card & bank: one of payments will be used in transaction				
3					
4	Attribute	Description	Table	Datatype	Constraint
5	cardID	unique ID for card	CARD	NUMERIC(4)	PK
6	cardNumber	actual card number for each card		NUMERIC(16)	NOT NULL
7	accountID	unique ID for bank account	BANKACC	NUMBER(3)	PK
8	accountNumber	actual bank account number for each bank account		NUMBER(17)	NOT NULL
9	benefitID	all benefits (e.g. no additional fee, quick send, ...)	BENEFITS	NUMERIC(2)	PK
10	benefit_desc	all benefits (e.g. no additional fee, quick send, ...)	BENEFITS	VARCHAR(25)	PK
11	benefitID	A benefit that a user owns	CLIENT_BENEFIT	NUMERIC(2)	PK, FK
12	senderID	A userID who receives a benefit		NUMERIC(3)	PK, FK
13	benefitStartDate	Date when user started to receive benefit		Date	NOT NULL
14	senderID	unique ID for sender	SENDER	NUMERIC(3)	PK
15	clientID	unique ID for client		NUMERIC(3)	FK, NOT NULL
16	receiverID	unique ID for receiver	RECEIVER	NUMERIC(3)	PK
17	clientID	unique ID for client		NUMERIC(3)	FK, NOT NULL
18	clientID	unique ID for client	CLIENT	NUMERIC(3)	PK
19	clientLast	last name of client		VARCHAR(25)	NOT NULL
20	clientFirst	first name of client		VARCHAR(25)	NOT NULL
21	balance	amount of money a client has		NUMERIC(10,2)	NOT NULL
22	phone	phone number of client		CHAR(10)	NOT NULL
23	streetAddress	street address of client		VARCHAR(30)	NOT NULL
24	state	state address of client		CHAR(2)	NOT NULL
25	email	email of client		VARCHAR(30)	
26	transactionID	unique ID for transaction	TRANSACTION	NUMERIC(3)	PK
27	tDate	transaction date		DATE	NOT NULL
28	tAmount	amount of money in transaction		NUMERIC(10,2)	NOT NULL
29	cardID	card that will be used in transaction		NUMERIC(4)	FK, NOT NULL
30	accountID	bank account that will be used in transaction		NUMERIC(3)	FK, NOT NULL
31	senderID	sender in transaction		NUMERIC(3)	FK, NOT NULL
32	receiverID	receiver in transaction		NUMERIC(3)	FK, NOT NULL

Part II. Relational Schema

Level 1:

CARD (cardID, cardNumber)

BANKACCOUNT (accountID, accountNumber)

BENEFITS (benefitID, benefit_desc)

CLIENT (clientID, clientLast, clientFirst, balance, phone, streetAddress, state, email)

- ※ Sequence statements can be used to generate primary keys for “cardID” and “accountID”.
- ※ “accountNumber” and “cardNumber” have UNIQUE constraints.

Level 2:

SENDER (senderID, *clientID*)

RECEIVER (receiverID, *clientID*)

CLIENT_BENEFIT (*benefitID*, *clientID*, benefitStartDate)

Level 3:

TRANSACTION (transactionID, tDate, tAmount, *senderID*, *receiverID*, *cardID*, *accountID*)

Part V.

-- 4. Create an index.

-- screenshots attached below

-- The index didn't improve the query response.

The screenshot shows the ATPCloudDB interface with a query in the Worksheet tab. The query is:

```
1 SELECT clientlast, clientfirst, balance
2 FROM client
3 WHERE (balance >= 5000 and balance <= 10000);
4
```

The Explain Plan tab shows the execution plan for this query. The plan includes a PX COORDINATOR, PX SEND, PX BLOCK, and TABLE ACCESS operations. The cost is 3, and the cardinality is 2. The distribution is 2QC (RANDOM).

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	DISTRIBUTION
SELECT STATEMENT				3	2
PX COORDINATOR					
PX SEND	SYS..TQ10000	QC (RANDOM)		3	2QC (RANDOM)
PX BLOCK		ITERATOR		3	2
TABLE ACCESS	CLIENT	STORAGE FULL		3	2

The Other XML section shows the following information:

```
{info}
  info type="derived_cpu_dop"
    2
  info type="derived_io_dop"
    2
  info type="dop_reason" note="y"
    degree limit
  info type="dop" note="y"
    4
```

The screenshot shows the ATPCloudDB interface with a query in the Worksheet tab. The query is:

```
1 SELECT clientlast, clientfirst, balance
2 FROM client
3 WHERE (balance >= 5000 and balance <= 10000);
4
5
6 CREATE INDEX client_bal_idx ON client(balance);
7
8 explain plan for select /*+ INDEX(client_bal_idx) */ clientlast, clientfirst, balance
9 from client where (balance >= 5000 and balance <= 10000);
10
11 select * from table(client(sql_id=>'null', format=>'ALLSTATS LAST'));
12
```

The Explain Plan tab shows the execution plan for this query. The plan includes a CREATE INDEX BUILD, INDEX BUILD, SORT, and TABLE ACCESS operations. The cost is 82, and the cardinality is 2. The distribution is 2.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	DISTRIBUTION
CREATE INDEX STATEMENT					
INDEX BUILD	CLIENT_BAL_IDX	NON UNIQUE		82	3
INDEX BUILD		CREATE INDEX		82	
SORT		STORAGE FULL		82	2
TABLE ACCESS	CLIENT				

The Other XML section shows the following information:

```
{info}
  info type="derived_cpu_dop"
    2
  info type="derived_io_dop"
    2
  info type="dop_reason" note="y"
    degree limit
  info type="dop" note="y"
    4
```