URL: http://flip1.engr.oregonstate.edu:7919/

Class: CS340-400

Assignment: Project Step 6 (Portfolio Assignment) (Group 37)

Authors: Mathew Pazaris and Patrick Dougan

1) Summary of feedback and changes made:

- Minor changes to ERD, to correct key names and remove redundancy. We corrected the locBill PK name to "lkKey", and removed locNum from transactions. We added lkKey to transactions.
- Fixed inconsistencies between our outline, ERD, and schema. We have gone through each entity and attribute name and corrected the inconsistencies.
- Logins done through E-mail Addresses instead of customer ID number
- Location page will keep the date filter because locations can be added or removed over time
- All options to add data now say "Insert(admin only)"
- Capitalization and grammar have been normalized between the webpages
- UI field names have been normalized between pages
- Documentation updated to reflect correct ID fields for relational tables
- ERD/Schema resized and formatting fixed
- tranType removed from Transactions table. Duplicated from Price Plans
- Ignoring pbKey as Plan Bill pbKey is it's primary key and Transactions table uses pbKey as a foreign key
- Transactions tranType removed. ERD updated from Transaction.tranNum to transNum
- Made transactions table "nullable". Can have no transactions for a given billing cycle.
- Schema corrected for FK and planID per comment.
- Added queries for transactions to select fields for input, so that invalid input can't be added, also added CustomerID, billNumber, and transactioniD to ensure that the data being added matches what's available for these options. Fixed Transactions table query, and insert for transactions based on this logic
- Removed admin login, as having trouble and was causing bugs, may add back, but not critical to the project.
- Added query to select valid customers for login, rather than text entry.

2) Project Outline

Overview

The Mindboggle corp is a software startup company that uses Artificial Intelligence to detect billing and payment fraud for clients at the point of transaction (POT); that is to say that customers process their transactions through Mindboggle corp software, which uses proprietary databases and AI to link the transactions to other fraudulent transactions, or payees, and identify the risk of the transaction being fraudulent. They have successfully launched

their first product, and now have a small client base of 20 customers, with hopes of expanding to 100 in the next 2 years. To do so, they need a better database to keep track of client billing and payments.

Mindbiggle corp bills customers on different billing cycles,such as weekly, monthly, bi-monthly or annual, and billing can vary by frequency. Bill frequency could change over time, but is fixed for a given bill. Bills are based on the number of transactions processed, with surcharges applied for the type of location and payment type processed for each payment that is validated by Mindboggle software. Clients may have multiple locations. Location types can include websites, malls, or independent stores, and the software requirements, therefore billing, differ for these locations.Customers are also billed surcharges based on 3 transaction types, Check/ACH, credit/debit card, EFT payment, as the requirements for processing these payment types vary.

Using Maria DB, Mindboggle will develop a relational database infrastructure that will be needed to maintain flexible and accurate accounting of clients and transactions and billing. This will allow surcharges for location and transaction type to be seperate, and allow bill amount to be generated by multiplying the transactions, by payment plan surcharge price per transaction, for transaction price for the location, and then sum over surcharge by transType (i.e. Check, EFT, ACT).

3) Database outline

Responsibilities for the web pages: We will collaborate, but Matt will have primary responsibilities for Customers and Billing pages and Patrick for Transactions and PricePlans pages. We will collaborate together for backend development. We will implement both M:M relationships CustBilling:PricePlan and CustBilling: LocationType.

Customers**: Table to record the details of Mindboggle corp customers**

- customerID: int, auto_increment, unique, not NULL, PK
- email: varchar, not NULL
- phone: varchar, not NULL
- address: varchar, not NULL
- relationship: a 1:M relationship between Customers and CustBilling is implemented with customerid as a FK inside of Billing (A customer can have multiple bills)

Customer Billings**: Table to track customer billing**

- billNum: int, auto_increment, unique, not NULL, PK
- customerID: int, not NULL, FK
- billDate: date, not NULL
- relationship: a M:M relationship between CustBilling and PricePlan is implemented through an associative table PlanBill (customer can have a different payment plan for different transTypes, and locations)
- relationship: a M:M relationship between CustBilling and LocationType is implemented through an associative table LocBill (customer can have a different payments for a price plan for different location types)

Transactions**: Table to track transactions per bill**

- transNum: int, auto_increment, unique, not NULL, PK
- pbKey: int, not NULL, FK

- lkKey: int, not NULL, FK
- tranCount, int, not NULL, Number of transactions
- relationship: a 1:1 relationship between Transactions and the associative table LocBill (customer can have a different payment for a price plan for different location types)
- relationship: a 1:1 relationship between Transactions and the associative table PlanBill (customer can have a different payment plan for different transTypes, and locations)

Price Plans: **Table to track customer billing by payment frequency for bill cycle, and transaction type**

- planID, int, auto_increment, unique, not NULL, PK
- billFreq: char, key indicating the billing frequency: w=weekly, m=monthly, b=bi-monthly, y=annual.
- transType, varchar not NULL, values Check/ACH, credit/debit card, EFT
- tranTypedSurcharge, float, not NULL surcharge, if any for using bill method
- relationship: a 1:M relationship between PricePlan and the associative table PlanBill (customer can have a different payment plan for different transTypes, and locations)

Location Types: **Table to track customer billing by location**

- locKey: int, auto_increment, unique, not NULL, PK
- planID: int, not NULL, FK
- locType, varchar, not NULL, Website, Store, Mall
- locationPrice, float, notNULL, charge for transaction at location type
- relationship: a 1:M relationship between Transactions and the associative table LocBill (customer can have a different payment for a price plan for different location types)
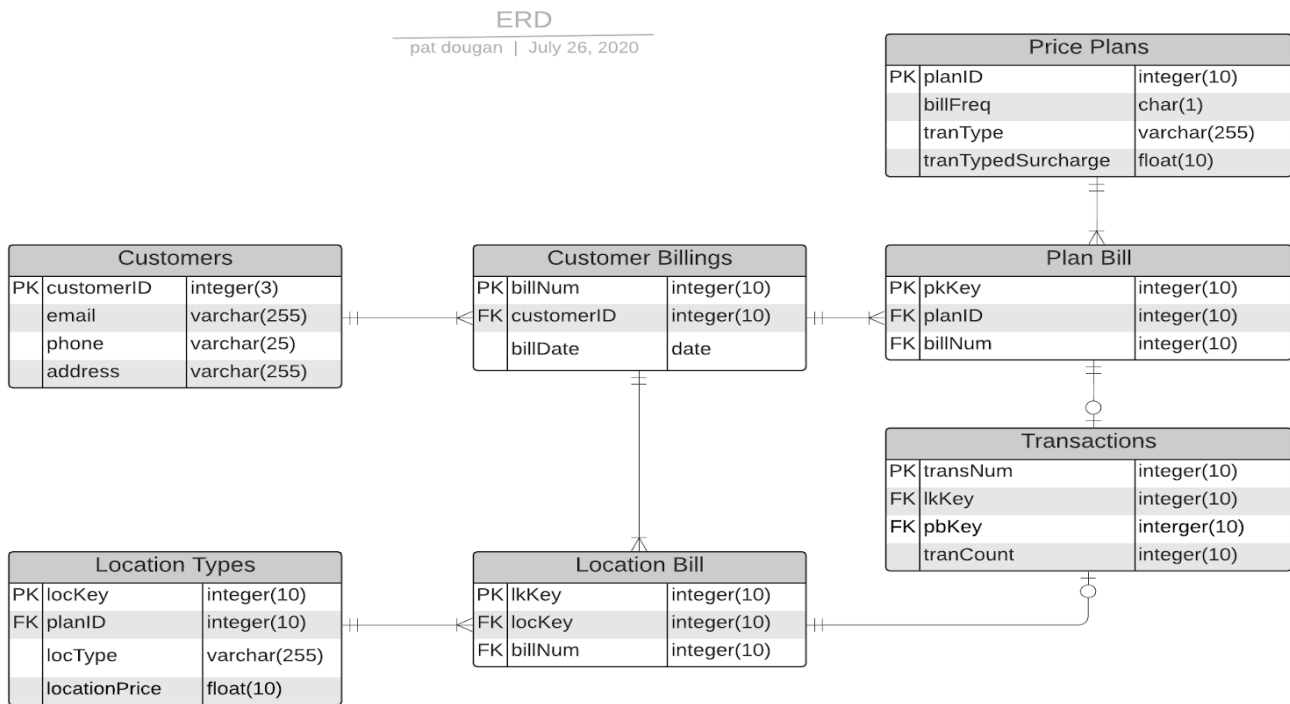
Plan Bill: **Associative table for M:M billing by PricePlan and CustBilling**

- pbKey: int, auto_increment, unique, not NULL, PK
- planID: int, not NULL
- billNum, int, not NULL
- relationship: a 1:M relationship between Transactions (a transaction can have 1 PlanBill and a PlanBill can apply to many transactions)

Location Bill: **Associative table for M:M billing by LocationType and CustBilling**

- lkKey: int, auto_increment, unique, not NULL, PK
- locKey: int, not NULL
- billNum, int, not NULL
- relationship: a 1:M relationship between Transactions (a transaction can have 1 LocBill and a LocBill can apply to many transactions)
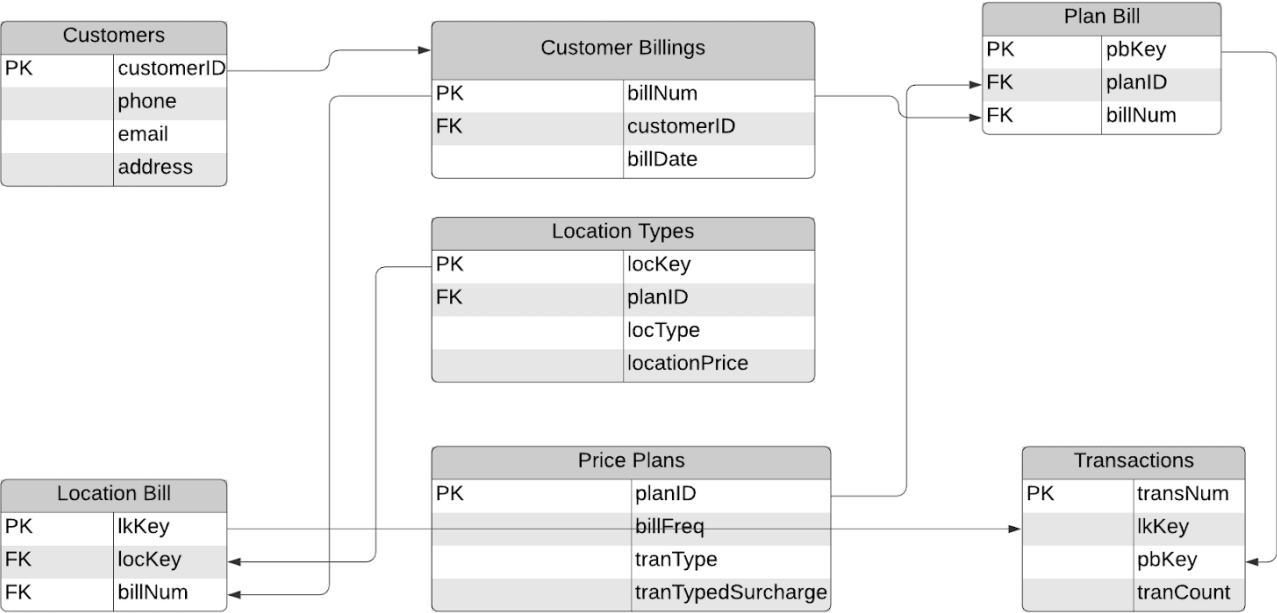
4) Entity-Relationship Diagram

Price Plans

PK	planID	integer(10)
	billFreq	char(1)
	tranType	varchar(255)
	tranTypedSurcharge	float(10)

Customers

PK	customerID	integer(3)
	email	varchar(255)
	phone	varchar(25)
	address	varchar(255)

Customer Billings

PK	billNum	integer(10)
FK	customerID	integer(10)
	billDate	date

Plan Bill

PK	pkKey	integer(10)
FK	planID	integer(10)
FK	billNum	integer(10)

Transactions

PK	transNum	integer(10)
FK	lkKey	integer(10)
FK	pbKey	interger(10)
	tranCount	integer(10)

Location Types

PK	locKey	integer(10)
FK	planID	integer(10)
	locType	varchar(255)
	locationPrice	float(10)

Location Bill

PK	lkKey	integer(10)
FK	locKey	integer(10)
FK	billNum	integer(10)

5) Schema

Customers

PK	customerID
	phone
	email
	address

Customer Billings

PK	billNum
FK	customerID
	billDate

Location Types

PK	locKey
FK	planID
	locType
	locationPrice

Price Plans

PK	planID
	billFreq
	tranType
	tranTypedSurcharge

Plan Bill

PK	pbKey
FK	planID
FK	billNum

Transactions

PK	transNum
	lkKey
	pbKey
	tranCount

Location Bill

PK	lkKey
FK	locKey
FK	billNum

6) Screen captures of each of the UI pages

a) **READ/BROWSE/DISPLAY/INSERT/UPDATE/EDIT** The "ACCOUNT" page interacts with the client table, and allows viewing, and insertion of new accounts, and updates to account information. All other tables filter to a customer's info if a customer logs in. There is no passwords for the demo.

ACCOUNT

note: corresponds to "Customers" table

Home	Account	Billing	Transactions	Price Plan	Location

Note: admin may want to change to differnt users to edit data. So need to know if logged out

Log in

(note: "admin" allows additional manipulation of tables

(note: "admin" not implemented for demo - use client login

(clients should change all info such as inserts as noted)

Please enter your client ID: [1 ▾]
Please Enter Your password (note all blank for class project): [_____]

[Login]

(note: shows and populates on login)

Client ID	Email	Phone#	Address	Edit

Sign up (note: this is an insert function)

Email: [_____]
Phone Number: [_____]
Address: [_____]
[Create new Account]

Congrats! Here is your client information

Note only displays if signing up

Client ID	1

b) READ/BROWSE/DISPLAY/INSERT The "BILLING" page allows selection of Customer Billing to view, and insertion.

BILLING

note:coresponds to "Customer Billings" table

| Home | Account | Billing | Transactions | Price Plan | Location |

Note: blank row and insert button only show for "admin"

Bill Number	Bill Date	Total
1	Wed Apr 01 2020 00:00:00 GMT-0700 (PDT)	188670
2	Fri May 01 2020 00:00:00 GMT-0700 (PDT)	337440
3	Mon Jun 01 2020 00:00:00 GMT-0700 (PDT)	149340
4	Wed Jul 01 2020 00:00:00 GMT-0700 (PDT)	182210
5	Sat Aug 01 2020 00:00:00 GMT-0700 (PDT)	217550
6	Tue Sep 01 2020 00:00:00 GMT-0700 (PDT)	0
7	Thu Oct 01 2020 00:00:00 GMT-0700 (PDT)	0
8	Sun Nov 01 2020 00:00:00 GMT-0700 (PDT)	0
9	Tue Dec 01 2020 00:00:00 GMT-0800 (PST)	0
10	Wed Apr 01 2020 00:00:00 GMT-0700 (PDT)	34580
11	Fri May 01 2020 00:00:00 GMT-0700 (PDT)	43540
12	Mon Jun 01 2020 00:00:00 GMT-0700 (PDT)	44030
13	Wed Jul 01 2020 00:00:00 GMT-0700 (PDT)	19460
14	Sat Aug 01 2020 00:00:00 GMT-0700 (PDT)	35210
15	Tue Sep 01 2020 00:00:00 GMT-0700 (PDT)	0
16	Thu Oct 01 2020 00:00:00 GMT-0700 (PDT)	0
17	Sun Nov 01 2020 00:00:00 GMT-0700 (PDT)	0
18	Tue Dec 01 2020 00:00:00 GMT-0800 (PST)	0
19	Wed Apr 01 2020 00:00:00 GMT-0700 (PDT)	64885
20	Fri May 01 2020 00:00:00 GMT-0700 (PDT)	85880
21	Mon Jun 01 2020 00:00:00 GMT-0700 (PDT)	44650
22	Wed Jul 01 2020 00:00:00 GMT-0700 (PDT)	60800
23	Sat Aug 01 2020 00:00:00 GMT-0700 (PDT)	74955
24	Tue Sep 01 2020 00:00:00 GMT-0700 (PDT)	0
25	Thu Oct 01 2020 00:00:00 GMT-0700 (PDT)	0
26	Sun Nov 01 2020 00:00:00 GMT-0700 (PDT)	0
27	Tue Dec 01 2020 00:00:00 GMT-0800 (PST)	43320
28	Wed Aug 05 2020 00:00:00 GMT-0700 (PDT)	
29	Thu Aug 06 2020 00:00:00 GMT-0700 (PDT)	
30	Wed Aug 05 2020 00:00:00 GMT-0700 (PDT)	

Customer ID: [_____]
Bill Date: [mm/dd/yyyy 📅] [Insert]

c) **SEARCH/FILTER/DELETE/READ/BROWSE/DISPLAY/INSERT** Transactions allows viewing (select) and insertion of new transactions. You can select a date range, to filter to a range and must select a date range and login to insert. The transaction table utilizes two many to many relationships. The login and date selection are necessary for the insert drop down selections, in order to make sure you are selecting valid options for a given customer and ID. There is also a delete function, which deletes a given transaction.

TRANSACTIONS

note:coresponds to "TRANSACTIONS" table and M:M associative tables

Home	Account	Billing	Transactions	Price Plan	Location

2020-04-01 ▼
View bill for selected date

Note: blank row and insert button only show for "admin"

Customer ID	Bill Number	Transaction ID	Billing Frequency	Transaction Type	Location Type	Transaction Count	Total	Delete(Admin only)
1	2	3	w	Check	Mall	959	115080	Delete
1	2	4	w	Check	Website	817	57190	Delete
2	11	12	m	Check	Website	622	43540	Delete
3	20	17	m	Check	Store	904	85880	Delete
	-999 ▼		NA ▼	NA ▼	NA ▼			Insert Row

d) **READ/BROWSE/DISPLAY/INSERT** Price Plans - Allows selection, and insertion of the Price Plans

PRICE PLAN

note:coresponds to "Price Plans" table

Home	Account	Billing	Transactions	Price Plan	Location

Plan ID	Billing Frequency	Transaction Type	Transaction Surcharge	Insert(Admin Only)
1	w	Check	20	
2	m	Check	20	
3	b	Check	20	
4	y	Check	10	
5	w	Credit	30	
6	m	Credit	30	
7	b	Credit	30	
8	y	Credit	20	
9	w	EFT	10	
10	m	EFT	10	
11	b	EFT	10	
12	y	EFT	5	
13	B	Debit	0	
Auto Incremented	Weekly ▼	Check ▼		Insert(Admin Only)

e) READ/BROWSE/DISPLAY/INSERT Locations - Allows selection, and insertion of the locations

LOCATION

note:coresponds to "location Types" table and M:M associative tables

Home	Account	Billing	Transactions	Price Plan	Location

Location Key	Current Plan ID	Location Type	Current Location Price	Insert (Admin Only)
1	1	Mall	100	
2	2	Website	50	
3	3	Store	75	
4	4	Stor	1	
Autoincremented		Website ▾		Insert(Admin Only)