

### Problem 3

Description: Gold's Gym maintains a database system to manage customer information across their many franchises. for each customer the system stores their name, address, a unique email address, their saved payment information, and the ids of the franchises locations at which they have a membership.

for each franchise the database stores a unique address, contact information consisting of a contact email and phone number, and the name of the franchise owner. Some franchises support the use of fingerprint access to their facility, so the database also stores the fingerprint of customers who choose to have access to this feature.

Each customer has an account that records their balance, payment interval, and optionally the credit card a customer has selected to use for autopay, if they have chosen to do so.

From this description we can derive the following set of functional dependencies:

$$\begin{aligned} &\{email \rightarrow name, address, fingerprint; \\ &\quad fingerprint \rightarrow name, address, email; \\ &franchise\_address, customer\_email \rightarrow payment\_interval, auto\_pay\_card; \\ &franchise\_address \rightarrow contact\_phone, contact\_email, owner\_name; \\ &referred\_email, referrer\_email \rightarrow date\_referred; \\ &\quad card\_number \rightarrow billing\_address; \} \end{aligned}$$

We propose the following schema each of which are in Third Normal Form:

Schema: **Customer(email, name, address, fingerprint)**

From the description we have that email is unique for each customer. Since each customer can have only one fingerprint, and since the database records only one address for each customer this gives us the dependencies:

$$\begin{aligned} &\{email \rightarrow name, address, fingerprint \\ &\quad fingerprint \rightarrow name, address, email\} \end{aligned}$$

Using Armstrong's Axioms we can compute a non-trivial part of the closure of this set of dependencies:

$$\begin{aligned} &\{email \rightarrow name, address, fingerprint; \\ &\quad email \rightarrow name, address; \\ &\quad email \rightarrow name, fingerprint; \\ &\quad email \rightarrow fingerprint, address; \\ &\quad email \rightarrow fingerprint; \\ &\quad email \rightarrow address; \\ &\quad email \rightarrow name; \} \end{aligned}$$

```

fingerprint  $\rightarrow$  name, address, email;
fingerprint  $\rightarrow$  name, address;
fingerprint  $\rightarrow$  email, address;
fingerprint  $\rightarrow$  name, email;
fingerprint  $\rightarrow$  name;
fingerprint  $\rightarrow$  email;
fingerprint  $\rightarrow$  address; }

```

From this set of dependencies we can deduce that both fingerprint and email are minimal superkeys, and thus candidate keys, and thus also they are prime attributes for this schema.

Claim: this schema is in Third Normal Form

*proof:*

All attributes in Customer are atomic, thus it is in First Normal Form

Above are listed (at least) all of the functional dependencies  $X \rightarrow A$  such that  $X$  is a candidate key and  $A$  is an attribute in Customer. Trivially, any superset  $U$  of  $X$  for  $X \in \{email, fingerprint\}$  must be a superkey of Customer, so any rule in  $F^+$ ,  $U \rightarrow A$ , must satisfy the property that  $U$  be a superkey of Customer. All other rules derived from armstrong's axioms will be trivial.

□

Schema: **membership(franchise\_address, customer\_email, auto\_pay\_card, payment\_interval)**

From the description we have that an email is unique for each customer, and each customer can have only one membership at each franchise. Naturally, each franchise can only be at one location and each membership can only have one interval on which it is paid for. From this information we deduce the following set of functional dependencies:

```

{franchise_address, customer_email  $\rightarrow$  payment_interval, auto_pay_card}

```

From this set we can compute all of the rules in  $F^+$ ,  $X \rightarrow Y$  such that  $X$  is minimal using Armstrong's Axioms:

```

{franchise_address, customer_email  $\rightarrow$  payment_interval, auto_pay_card;
franchise_address  $\rightarrow$  franchise_address;
customer_email  $\rightarrow$  customer_email;
payment_interval  $\rightarrow$  payment_interval;
auto_pay_card  $\rightarrow$  auto_pay_card;
franchise_address, customer_email  $\rightarrow$  payment_interval;
franchise_address, customer_email  $\rightarrow$  auto_pay_card;
franchise_address, customer_email  $\rightarrow$  payment_interval, franchise_address;
franchise_address, customer_email  $\rightarrow$  auto_pay_card, franchise_address;
franchise_address, customer_email  $\rightarrow$  payment_interval, customer_email;
franchise_address, customer_email  $\rightarrow$  auto_pay_card, customer_email;
franchise_address, customer_email
 $\rightarrow$  auto_pay_card, customer_email, franchise_address;

```

$$\begin{aligned} & \text{franchise\_address, customer\_email} \\ & \rightarrow \text{payment\_interval, customer\_email, franchise\_address;} \\ & \text{franchise\_address, customer\_email} \\ & \rightarrow \text{auto\_pay\_card, payment\_interval, customer\_email, franchise\_address} \end{aligned}$$

Claim: This schema is in Third Normal Form

*proof*

All of the attributes in the table membership are atomic, thus the table is in First Normal Form

Every rule not included in the set above is of the form  $U \rightarrow Y$  where  $U \supset X$ . Each dependency in the set above is either trivial, or the LHS is a superkey of membership. Since all dependencies not included above must have a LHS that is a superset of these superkeys their LHS must also be a superkey. Thus we have that every dependency  $X \rightarrow A \in F^+$  must either have the property that  $X$  is a superkey of membership, or the dependency is trivial.  $\square$

Schema: **franchise(address, contact\_phone, contact\_email, owner\_name)**

From the description we know that each franchise can exist at only one address. With that information we obtain the functional dependency:

$\{ \text{address} \rightarrow \text{contact\_phone, contact\_email, owner\_name} \}$

From this dependency we can use Armstrong's Axioms to compute the rules  $X \rightarrow Y$  in  $F^+$  such that  $X$  is minimal:

$$\begin{aligned} & \{ \text{address} \rightarrow \text{address;} \\ & \text{contact\_phone} \rightarrow \text{contact\_phone;} \\ & \text{contact\_email} \rightarrow \text{contact\_email;} \\ & \text{owner\_name} \rightarrow \text{owner\_name;} \\ & \text{address} \rightarrow \text{address, contact\_phone, contact\_email, owner\_name;} \\ & \text{address} \rightarrow \text{contact\_phone, contact\_email, owner\_name;} \\ & \text{address} \rightarrow \text{contact\_phone, contact\_email;} \\ & \text{address} \rightarrow \text{contact\_phone, owner\_name;} \\ & \text{address} \rightarrow \text{contact\_email, owner\_name;} \\ & \text{address} \rightarrow \text{contact\_email;} \\ & \text{address} \rightarrow \text{owner\_name;} \\ & \text{address} \rightarrow \text{contact\_phone;} \end{aligned}$$

} Claim: This schema is in Third Normal Form

*proof*

All of the attributes in the franchise schema are atomic, so it is in First Normal Form

The above set contains all dependencies  $X \rightarrow Y$  such that  $X$  is minimal, so by inspection we can see that any other rules in  $F^+$  must either be trivial or

have the property that their LHS is a superset of *address* which we can see is a candidate key, so those rules have the property that their LHS is a superkey of franchise.

□

Schema: **referrals**(*referred\_email*, *referrer\_email*, *date\_referred*)

From the description we can parse that emails are unique for customers of Gold's Gym, and that referrals last for a month. It is not logical to store multiple referrals for the same pair of users. From this information we derive the following set of functional dependencies:

$$\{referred\_email, referrer\_email \rightarrow date\_referred\}$$

Using Armstrong's Axioms we can compute  $F^+$ :

$$\begin{aligned} & \{referred\_email \rightarrow referred\_email; \\ & \quad referrer\_email \rightarrow referrer\_email; \\ & \quad date\_referred \rightarrow date\_referred; \\ & referred\_email, referred\_email \rightarrow referrer\_email, referred\_email; \\ & referred\_email, date\_referred \rightarrow referred\_email, date\_referred; \\ & date\_referred, referrer\_email \rightarrow date\_referred, referrer\_email; \\ & \quad date\_referred, referrer\_email, referred\_email \\ & \rightarrow date\_referred, referrer\_email, referred\_email; \\ & \quad referred\_email, referrer\_email \rightarrow date\_referred; \\ & \quad referred\_email, referrer\_email \rightarrow referrer\_email; \\ & \quad referred\_email, referrer\_email \rightarrow referred\_email; \\ & \quad referred\_email, referrer\_email \rightarrow date\_referred; \\ & referred\_email, referrer\_email, date\_referred \rightarrow date\_referred; \\ & referred\_email, referrer\_email, date\_referred \rightarrow referred\_email; \\ & referred\_email, referrer\_email, date\_referred \rightarrow referrer\_email; \\ & \quad referrer\_email, referred\_email, date\_referred \\ & \rightarrow referrer\_email, referred\_email; \\ & referred\_email, referrer\_email, date\_referred \rightarrow referred\_email, date\_referred; \\ & referred\_email, date\_referred, referrer\_email \rightarrow date\_referred, referrer\_email; \end{aligned}$$

} Claim: This schema is in Third Normal Form

*proof:*

All attributes in the referrals table are atomic, so this schema is in First Normal Form

By inspection we can see that all of the dependencies  $X \rightarrow A \in F^+$  where  $A$  is a single attribute have the property that either they are trivial or  $X \supset \{referred\_email, referrer\_email\}$  which we can see is a candidate key for referrals.

□

Schema: **customer\_card(card\_number, customer\_email)**

From the description we parse that a customer is uniquely identified by their email, but each customer could have multiple credit cards. We cannot derive any functional dependencies from the description, so the  $F^+$  on this schema must only have trivial dependencies, and thus since this schema has only atomic attributes it is trivially in Third Normal Form.

Schema: **card\_billing(billing\_address, card\_number)**

Each credit card can only have one billing address. From this information we can derive the following set of functional dependencies:

$$\{card\_number \rightarrow billing\_address;\}$$

From this set of dependencies we can use Armstrong's Axioms to derive all of the dependencies  $X \rightarrow A$  such that  $A$  is a single attribute of payment\_info in  $F^+$ :

$$\begin{aligned} &\{billing\_address \rightarrow billing\_address; card\_number \rightarrow \\ &\quad card\_number; card\_number \rightarrow \\ &\quad billing\_address; card\_number, billing\_address \rightarrow \\ &\quad card\_number; card\_number, billing\_address \rightarrow billing\_address;\} \end{aligned}$$

Claim: This schema is in Third Normal Form

*proof:*

All attributes in the card\_billing table are atomic, so this schema is in First Normal Form

By inspection we can see that every rule  $X \rightarrow AF^+$  is either trivial, or  $X$  is a superkey of card\_billing.

□