Lab 3 - Database Design

E-R Model Redesign Document

I. Entity Sets:

CustomerEmail Entity Set:

Name	Type	Description
email_id	Int	Unique Identifier of the customer email address
email_domain	String	Domain of customer email
audience	String	Customer segment

CustomerAccount Entity Set:

Name	Type	Description
customer_id	int	Unique customer Identifier
permission	boolean	Optin/optout indicator for marketing communications
language	string	Language of customer
customer_tier	string	Tier customer has been assigned to by BrandX
gender	char	Gender of customer
zip	int	Zip code of customer
state	string	State of customer
income_level	string	Customer Income
num_registrations	int	Number of registrations customer has

RegistrationLocation Entity Set:

Name	Type	Description
registration_source_id	int	Identifier of the account

		registration source
source_name	int	Name of the registration source

DeviceRegistration Entity Set:

Name	Type	Description
registration_id	int	Unique registration
registration_date	Date	The date of registration

PurchaseInformation Entity Set:

Name	Type	Description
purchase_store_name	String	Purchase store name
purchase_store_state	String	Purchase store state
purchase_store_city	String	Purchase store city
ecomm_flag	boolean	ecommFlag

DeviceSerial Entity Set:

Name	Type	Description
serial_number	String	Serial number

DevicePurchaseDate Entity Set:

Name	Type	Description
purchase_date	Date	Purchase date

Event Entity Set:

Name	Type	Description
event_date	Date	When event happened

EventTypes Entity Set:

Name	Type	Description
event_type_id	int	ID for email event types
event_type_name	String	Email event type

EmailMessage Entity Set:

Name	Type	Description
campaign_name	String	Name of email campaign
version	String	Creative version
subject_line	String	Email subject line
audience	String	Customer segment

Deployment Entity Set:

Name	Type	Description
deployment_id	int	Batch ID
deployment_date	Date	When email has been sent

Link Entity Set:

Name	Type	Description
link_name	String	Link Alias, e.g. Hero Module or Contact Customer Service
url	String	Link URL

DeviceModel Entity Set:

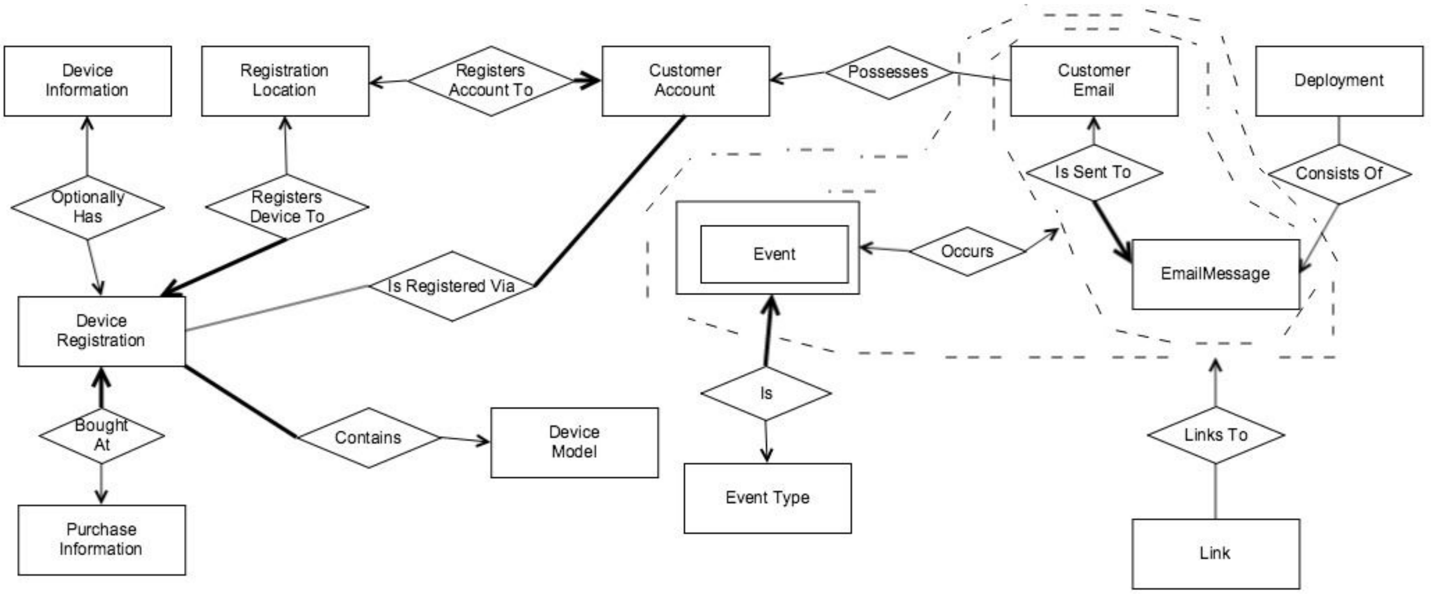
Name	Type	Description
device_name	String	Name of device, such as ‘Constellation 5’

carrier	String	AT&T, Version, etc.
device_model	String	Model of Device
device_type	String	Phone, Tablet, etc.

II. Relationship Sets:

- Contains(DeviceRegistration, DeviceModel)
    - many-to-one
    - DeviceRegistration participates in DeviceModel
- BoughtAt(DeviceRegistration, PurchaseInfo)
    - one-to-one
    - DeviceRegistration participates in PurchaseInfo
    - PurchaseInfo is weak
- OptionallyHas(DeviceRegistration, DeviceInfo)
    - one-to-one
- RegistersDeviceTo(DeviceRegistration, RegistrationLocation)
    - one-to-one
    - DeviceRegistration participates in RegistrationLocation
- IsRegisteredVia(CustomerAccount, DeviceRegistration)
    - many-to-many
    - DeviceRegistration participates in CustomerAccount
- RegistersAccountTo(CustomerAccount, RegistrationLocation)
    - one-to-one
    - CustomerAccount participates in RegistrationLocation
- Possesses(CustomerAccount, CustomerEmail)
    - one-to-many
- IsSentTo(EmailSent, CustomerEmail)
    - many-to-many
    - EmailSentparticipates in CustomerEmail
- Happens(Event, IsSentTo(CustomerEmail, EmailSent))
    - many-to-many
- Link(Links, Happens(Event, IsSentTo(CustomerEmail, EmailSent)))
    - many-to-many
- Is(Event, EventType)
    - one-to-one
    - Event participates in EventType
- ConsistsOf(EmailSent, Deployments)
    - one-to-many
    - Deployments participates in EmailSent
- LinksTo(EmailSent, Link)
    - one-to-many

III. E-R Diagram:



Change Log

Version 1.1

- Created entity sets based on initial interpretation of the data schema
- Determined relationship sets and their participation and multiplicity constraints

Version 1.2

- Split larger entities into more, smaller entities
- Updated entity sets to reflect more complex relationships (initial diagram was too basic)

Version 2.0

- Determined Functional Dependencies for each dataset
- Revised Entity Sets to be based on Functional Dependencies
- Updated E-R Diagram to include aggregation for when an event occurs in relation to emails sent out for a campaign
- Made Events a weak entity set because it determines EventType and the aggregation relationship between Campaign and CustomerEmail

Version 3.0

- Added DB-setup.sql
- Split up SerialNumber and PurchaseDate
- DeploymentId and DeploymentDate are primary keys rather than just DeploymentId

List of Functional Dependencies

Account Registration Dataset:

- CustomerEmail (<u>EmailID</u>, CustomerID, Email Domain)

- CustomerAccount(CustomerID, Permission, Language, Customer Tier, Gender, Zip, State, Income Level)
- Registration(Registration Source ID, Registration Source Name)

Device Registration Dataset:

- DeviceRegistration(RegistrationID, RegistrationDate)
- Registration(RegistrationSourceID, RegistrationSourceName)
- PurchaseInformation(PurchaseStoreName, PurchaseStoreState, PurchaseStoreCity, EcommFlag)
- CustomerAccount(CustomerID, Number of Registrations)
- DeviceInformation(SerialNumber, DeviceModel, PurchaseDate)

Email Event Dataset:

- CustomerEmail (EmailID, Audience)
- Event(EmailID, EventTypeID, EventDate)
- EventTypes(EventTypeID, EventTypeName)
- Campaign(CampaignName, Version, SubjectLine, Audience)
- Deployment(DeploymentID, DeploymentDate)
- Link(LinkName, URL)

Device Dataset:

- DeviceData(DeviceName, Carrier, DeviceModel, DeviceType)

Logical Database Design Document

DB-setup.sql

```
CREATE TABLE RegistrationLocation(
  registration_source_id INT,
  source_name VARCHAR(100),
  PRIMARY KEY(registration_source_id)
);

CREATE TABLE Link(
  link_name VARCHAR(30),
  url VARCHAR(50),
  PRIMARY KEY(link_name)
);

CREATE TABLE DeviceModel(
  device_name VARCHAR(100),
  carrier VARCHAR(100),
  device_model VARCHAR(100),
  device_type VARCHAR(100),
  PRIMARY KEY(device_name, carrier)
);

CREATE TABLE EventType(
  event_type_id INT,
  event_type_name VARCHAR(20),
  PRIMARY KEY(event_type_id, event_type_name)
);

CREATE TABLE DeviceInformation (
  serial_number VARCHAR(50),
  device_model VARCHAR(100),
  purchase_date DATE,
  PRIMARY KEY(serial_number)
);

CREATE TABLE PurchaseInformation(
  purchase_store_id INT,
  purchase_store_name VARCHAR(50),
  purchase_store_state CHAR(2),
  purchase_store_city VARCHAR(50),
  ecomm_flag BOOLEAN,
  PRIMARY KEY(purchase_store_id),
  UNIQUE(purchase_store_name, purchase_store_state, purchase_store_city)
);

CREATE TABLE Campaign(
  campaign_name VARCHAR(100),
  version VARCHAR(50),
  subject_line VARCHAR(100),
  audience VARCHAR(100),
  fk_link_name VARCHAR(30),
  PRIMARY KEY(campaign_name),
  FOREIGN KEY(fk_link_name) REFERENCES Link(link_name)
);

CREATE TABLE CustomerAccount(
  customer_id INT,
  permission BOOL,
  customer_tier VARCHAR(20),
  gender CHAR(1),
  zip INT,
  state CHAR(2),
  income_level VARCHAR(20),
  num_registrations INT,
  fk_registration_source_id INT,
  PRIMARY KEY(customer_id),
  FOREIGN KEY(fk_registration_source_id) REFERENCES
RegistrationLocation(registration_source_id)
);

CREATE TABLE CustomerEmail(
  email_id INT,
  customer_id INT,
  email_domain VARCHAR(50),
  audience VARCHAR(100),
  PRIMARY KEY(email_id),
  FOREIGN KEY (customer_id) REFERENCES CustomerAccount(customer_id)
);

CREATE TABLE IsSentTo(
  fk_campaign_name VARCHAR(100),
  fk_email_id INT,
  FOREIGN KEY(fk_campaign_name) REFERENCES Campaign(campaign_name),
  FOREIGN KEY(fk_email_id) REFERENCES CustomerEmail(email_id)
);

CREATE TABLE DeviceRegistration(
  registration_id INT,
  registration_date DATE,
  fk_device_name VARCHAR(100),
  fk_carrier VARCHAR(100),
  fk_purchase_store_id INT,
  fk_serial_number VARCHAR(50),
  fk_registration_source_id INT,
  PRIMARY KEY(registration_id),
  FOREIGN KEY(fk_device_name, fk_carrier) REFERENCES
DeviceModel(device_name, carrier),
  FOREIGN KEY(fk_purchase_store_id) REFERENCES
PurchaseInformation(purchase_store_id),
  FOREIGN KEY(fk_serial_number) REFERENCES DeviceInformation(serial_number),
  FOREIGN KEY(fk_registration_source_id) REFERENCES
RegistrationLocation(registration_source_id)
);

CREATE TABLE IsRegisteredVia(
  fk_registration_id INT,
  fk_customer_id INT,
  FOREIGN KEY(fk_registration_id) REFERENCES DeviceRegistration(registration_id),
  FOREIGN KEY(fk_customer_id) REFERENCES CustomerAccount(customer_id)
);
```

```
CREATE TABLE Possesses(
  fk_email_id INT,
  fk_customer_id INT,
  FOREIGN KEY(fk_email_id) REFERENCES CustomerEmail(email_id),
  FOREIGN KEY(fk_customer_id) REFERENCES CustomerAccount(customer_id)
);

CREATE TABLE Event(
  email_id INT,
  event_type INT,
  event_date DATE,
  fk_event_type_id INT,
  fk_campaign_name VARCHAR(100),
  PRIMARY KEY(email_id, fk_event_type_id),
  FOREIGN KEY(fk_event_type_id) REFERENCES EventType(event_type_id),
  FOREIGN KEY(fk_campaign_name) REFERENCES Campaign(campaign_name),
  FOREIGN KEY(email_id) REFERENCES CustomerEmail(email_id)
);

CREATE TABLE Deployment(
  deployment_id INT,
  deployment_date DATE,
  fk_campaign_name VARCHAR(100),
  PRIMARY KEY(deployment_id),
  FOREIGN KEY(fk_campaign_name) REFERENCES Campaign(campaign_name)
);
```

DB-cleanup.sql

```
DROP TABLE Deployment;
DROP TABLE IsSentTo;
DROP TABLE Event;
DROP TABLE Possesses;
DROP TABLE IsRegisteredVia;
DROP TABLE DeviceRegistration;
DROP TABLE Campaign;
DROP TABLE PurchaseInformation;
DROP TABLE DeviceInformation;
DROP TABLE EventType;
DROP TABLE DeviceModel;
DROP TABLE Link;
DROP TABLE CustomerEmail;
DROP TABLE CustomerAccount;
DROP TABLE RegistrationLocation;
```

FOR GITHUB:

```
##CustomerEmail Entity Set:

| Name        | Type          | Description  |
| ------------- |:-------------:| -----:|
| **email_id** | int | Unique Identifier of the customer email address |
| customer_id | int | Unique customer Identifier |
| email_domain | String | Domain of customer email |
| audience | String | Customer segment



## CustomerAccount Entity Set:

| Name        | Type          | Description  |
| ------------- |:-------------:| -----:|
| **customer_id** | int | Unique customer Identifier |
| permission | boolean | Optin/optout indicator for marketing communications |
| language | String | Language of customer |
| customer_tier | String | Tier customer has been assigned to by BrandX |
| gender | char | Gender of customer|
| zip | int | Zip code of customer |
| state | String| State of customer |
| income_level | String | Customer Income |
| num_registrations | int | Number of registrations customer has

##RegistrationLocation Entity Set:
| Name        | Type          | Description  |
| ------------- |:-------------:| -----:|
| **registration_source_id** | int | Identifier of the account registration source |
| source_name | int | Name of the registration source

##DeviceRegistration Entity Set
| Name        | Type          | Description  |
| ------------- |:-------------:| -----:|
| **registration_id** | int | Unique registration |
| registration_date | Date | The date of registration|

##PurchaseInformation Entity Sets
| Name        | Type          | Description  |
| ------------- |:-------------:| -----:|
| **purchase_store_name** | String | Purchase store name |
| **purchase_store_state** | String | Purchase store state |
| **purchase_store_city** | String | Purchase store city |
| ecomm_flag | boolean | ecommFlag

##DeviceInformation Entity Set:
| Name        | Type          | Description  |
| ------------- |:-------------:| -----:|
| **serial_number** | String | Serial number |
| device_model | String | Device model |
| purchase_date | Date | Purchase date

##Event Entity Set:
| Name        | Type          | Description  |
| ------------- |:-------------:| -----:|
| **email_id** | int | Unique identifier of the email ID |
| **event_type_id** | int | ID for email event types |
| event_date | Date | When event happened

##EventTypes Entity Set:
| Name        | Type          | Description  |
| ------------- |:-------------:| -----:|
| **event_type_id** | int | ID for email event types |
| **event_type_name** | String | Email event type

##Campaign Entity Set:
| Name        | Type          | Description  |
| ------------- |:-------------:| -----:|
| **campaign_name** | String | Name of email campaign |
| **version** | String | Creative version |
| subject_line | String | Email subject line |
| audience | String | Customer segment

##Deployment Entity Set:
| Name        | Type          | Description  |
| ------------- |:-------------:| -----:|
| **deployment_id** | int | Batch ID |
| deployment_date | Date | When email has been sent

##Link Entity Set:
| Name        | Type          | Description  |
| ------------- |:-------------:| -----:|
| **link_name** | String | Link Alias, e.g. Hero Module or Contact Customer Service |
| url | String | Link URL

##DeviceModel Entity Set:
| Name        | Type          | Description  |
| ------------- |:-------------:| -----:|
| **device_name** | String | Name of device, such as ‘Constellation 5’ |
| **carrier** | String | AT&T, Version, etc. |
```

device_model	String	Model of Device
device_type	String	Phone, Tablet, etc.

Notes:
Id’s are actually strings
Use another table to do CustomerKey Int to CustomerId String
(This improves reading speed)