SUPERTEAM
Sam Chase, Jamie Corr, Andrew Rose
CSC 366-01

[Github Repo]

Description & Tradeoffs

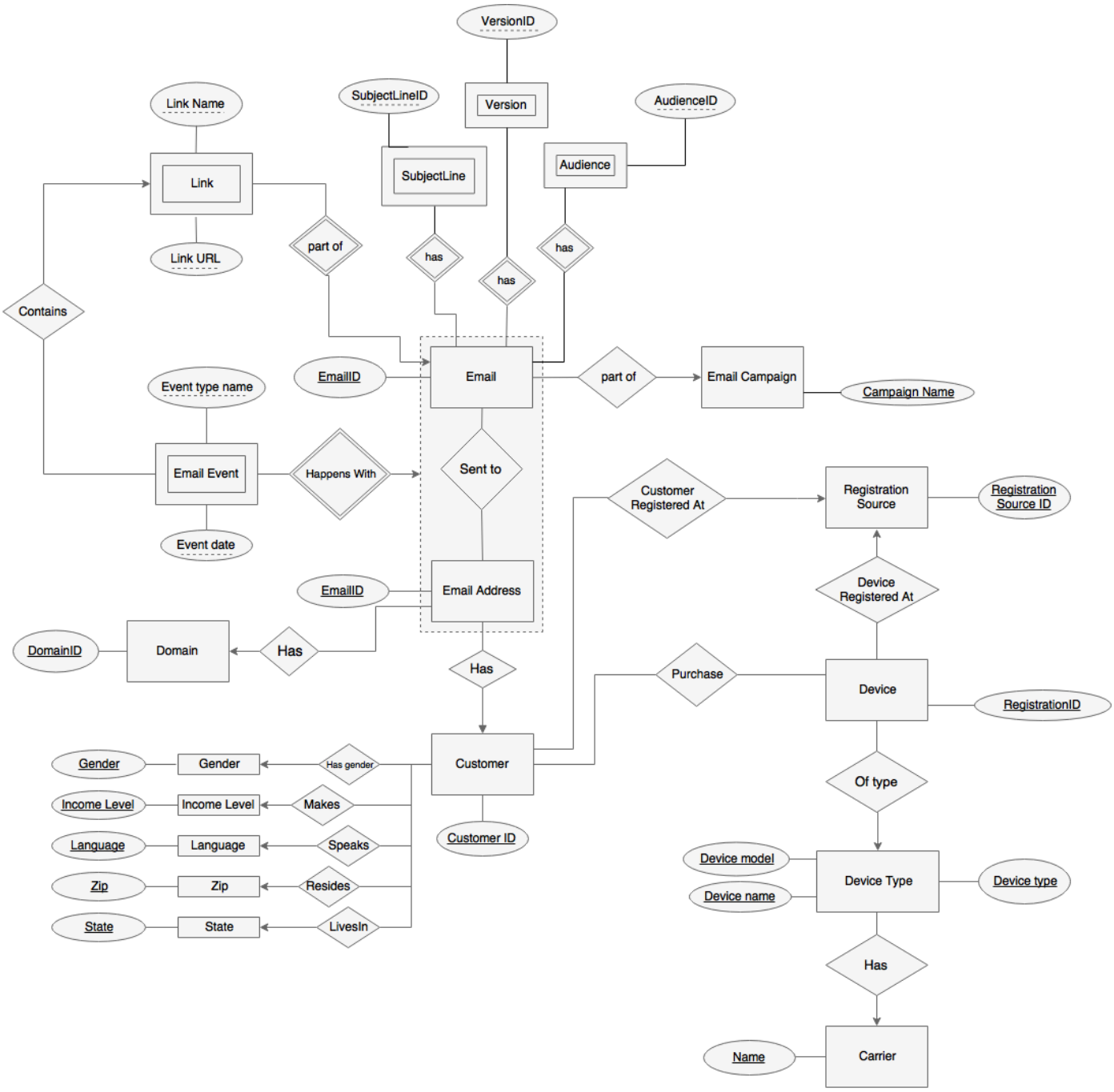
With our customer's priorities in mind, we modeled the data in a way that avoids unnecessary duplication and maximizes the performance of the queries. Our goal was to maintain best practices of warehouse design by utilizing the star schema design principle in multiple areas. Examples of star schemas include the modeling various customer account attributes (i.e State, Gender, Language) as well as email attributes (i.e. Version, Audience, Subject Line). In handling these entity and relationship sets this way, we were able to improve efficiency and clarity by avoiding null or empty values.

During the design phase, it was ultimately decided that we would develop a highly normalized database, which would subsequently also be very space efficient. The reasoning for this was twofold. The first was that the client had indicated preference for a normalized database free from anomalies. This preference made the choice for us clear. Additionally, the benefits of a performance optimized database were negligent. Our data warehouse tasks showed short runtimes, and as such the performance gains of a non normalized database would be minor at best, while the drawbacks would be large.

Size & Runtimes

Database Size:	161MB
Query 1 Runtime	61 sec
Query 2 Runtime	.98 sec
Query 3 Runtime	3.77 sec

E-R Diagram



Relational Model

```
Carrier(ID, CarrierName)
Device_Type(DeviceModel,DeviceName,Devicetype,CarrierID)
RegistrationSource(regSourceID,regSourceName)
Gender(id,Gender)
IncomeLevel(id,IncomeLevel)
```

Language(id,Language)
Zip(id,Zip)
State(id,State)
Customer(CustomerID,Permission, Tier, RegistrationDate, NumRegistrations,
RegisteredAt, GenderID, IncomeLevelID, LanguageID, ZipID, StateID)
Device(id, DeviceModel, SerialNumber, RegistrationDate, RegistrationID)
PurchaseLocation(id, PurchaseStoreName, PurchaseStoreState, PurchaseStoreCity)
Purchase(id, Ecomm, DeviceRegistrationID, CustomerID, PurchaseLocationID, PurchaseDate)
DeviceRegistration(deviceRegistrationID, registeredAt, registrationDate)
EmailCampaign(Id, CampaignName, DeploymentDate)
SubjectLine(id, SubjectLine)
Audience(id, Audience)
Version(id, Version)
Email(id, EmailCampaignID)
EmailVersion(EmailID, VersionID)
EmailSubject(EmailID, SubjectLineID)
EmailAudience(EmailID, AudienceID)
Domain(DomainID, DomainName)
EmailAddress(EmailAddressID, CustomerID, DomainID)
EmailSentTo(emailAddressID, EmailID)
Link(LinkID, LinkName, LinkURL, EmailID)
EmailEvent(eventID,eventType, eventDate, EmailID, emailAddressID, linkID)

Schemas

```
CREATE TABLE IF NOT EXISTS Carrier
(
    ID INT NOT NULL AUTO_INCREMENT,
    CarrierName VARCHAR(64),
    PRIMARY KEY (ID, CarrierName)
);

CREATE TABLE IF NOT EXISTS Device_Type
(
    DeviceModel VARCHAR(255) NOT NULL,
    DeviceName VARCHAR(255),
    Devicetype VARCHAR(32),
    CarrierID INT,
    PRIMARY KEY (DeviceModel, DeviceName, DeviceType, CarrierID),
    FOREIGN KEY (CarrierID) REFERENCES Carrier(ID)
);

CREATE TABLE IF NOT EXISTS RegistrationSource(
    regSourceID INT PRIMARY KEY,
    regSourceName VARCHAR(32)
);

CREATE TABLE IF NOT EXISTS Gender
(   id INT AUTO_INCREMENT,
    Gender CHAR(1) NOT NULL,
```

```
    PRIMARY KEY (Gender),
    UNIQUE(id)
);

CREATE TABLE IF NOT EXISTS IncomeLevel
(   id INT AUTO_INCREMENT,
    IncomeLevel VARCHAR(32) NOT NULL,
    PRIMARY KEY (IncomeLevel),
    UNIQUE(id)
);

CREATE TABLE IF NOT EXISTS Language
(   id INT AUTO_INCREMENT,
    Language CHAR(3) NOT NULL,
    PRIMARY KEY (Language),
    UNIQUE(id)
);

CREATE TABLE IF NOT EXISTS Zip
(   id INT AUTO_INCREMENT,
    Zip INT,
    PRIMARY KEY (Zip),
    UNIQUE(id)
);

CREATE TABLE IF NOT EXISTS State
(   id INT AUTO_INCREMENT,
    State VARCHAR(127) NOT NULL,
    PRIMARY KEY (State),
    UNIQUE (id)
);

CREATE TABLE IF NOT EXISTS Customer
(
    CustomerID VARCHAR(32) NOT NULL,
    Permission CHAR(1),
    Tier VARCHAR(32),
    RegistrationDate DATE,
    NumRegistrations INT,
    RegisteredAt INT,
    GenderID INT,
    IncomeLevelID INT,
    LanguageID INT,
    ZipID INT,
    StateID INT,
    PRIMARY KEY (CustomerID),
    FOREIGN KEY (GenderID) REFERENCES Gender(id),
    FOREIGN KEY (IncomeLevelID) REFERENCES IncomeLevel(id),
    FOREIGN KEY (LanguageID) REFERENCES Language(id),
    FOREIGN KEY (ZipID) REFERENCES Zip(id),
    FOREIGN KEY (StateID) REFERENCES State(id),
    FOREIGN KEY (registeredAt) REFERENCES RegistrationSource(regSourceID)
```

```
);

CREATE TABLE IF NOT EXISTS Device
(
    id INT AUTO_INCREMENT,
    DeviceModel VARCHAR(255),
    SerialNumber VARCHAR(64),
    RegistrationDate DATE,
    RegistrationID VARCHAR(64) NOT NULL,
    CONSTRAINT Device_RegistrationID_pk PRIMARY KEY (RegistrationID),
    CONSTRAINT Device_Device_Type_DeviceModel_fk FOREIGN KEY (DeviceModel) REFERENCES
Device_Type (DeviceModel),
    UNIQUE(ID)
);

CREATE TABLE IF NOT EXISTS PurchaseLocation
(
    id int AUTO_INCREMENT,
    PurchaseStoreName VARCHAR(255),
    PurchaseStoreState CHAR(3),
    PurchaseStoreCity VARCHAR(255),
    PRIMARY KEY (PurchaseStoreName,PurchaseStoreState,PurchaseStoreCity),
    UNIQUE(id)
);

CREATE TABLE IF NOT EXISTS Purchase
(
    id INT AUTO_INCREMENT NOT NULL,
    Ecomm CHAR(1),
    DeviceRegistrationID VARCHAR(64) NOT NULL,
    CustomerID VARCHAR(32) NOT NULL,
    PurchaseLocationID int,
    PurchaseDate DATE,
    FOREIGN KEY (DeviceRegistrationID) REFERENCES Device(RegistrationID),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (PurchaseLocationID) REFERENCES PurchaseLocation(id),
    PRIMARY KEY (CustomerID, DeviceRegistrationID),
    UNIQUE(id)
);

CREATE TABLE IF NOT EXISTS DeviceRegistration(
    deviceRegistrationID VARCHAR(64),
    registeredAt INT,
    registrationDate DATE,
    PRIMARY KEY (deviceRegistrationID),
    FOREIGN KEY (deviceRegistrationID) REFERENCES Device(RegistrationID),
    FOREIGN KEY (registeredAt) REFERENCES RegistrationSource(regSourceID)
);

CREATE TABLE IF NOT EXISTS EmailCampaign
(
    id INT AUTO_INCREMENT,
```

```
    CampaignName VARCHAR(255),
    DeploymentDate DATE,
    CONSTRAINT EmailCampaign_pk PRIMARY KEY (CampaignName, DeploymentDate),
    Unique(id)
);

CREATE TABLE IF NOT EXISTS SubjectLine
(
    id INT AUTO_INCREMENT,
    SubjectLine VARCHAR(255),
    PRIMARY KEY (SubjectLine),
    UNIQUE(id)
);

CREATE TABLE IF NOT EXISTS Audience
(
    id INT AUTO_INCREMENT,
    Audience VARCHAR(255),
    PRIMARY KEY (Audience),
    UNIQUE(id)
);

CREATE TABLE IF NOT EXISTS Version
(
    id INT AUTO_INCREMENT,
    Version VARCHAR(255),
    PRIMARY KEY (Version),
    UNIQUE(id)
);

CREATE TABLE IF NOT EXISTS Email
(
    id INT AUTO_INCREMENT,
    EmailCampaignID INT,
    PRIMARY KEY (id),
    FOREIGN KEY (EmailCampaignID) REFERENCES EmailCampaign (id),
    UNIQUE(id)
);

CREATE TABLE IF NOT EXISTS EmailVersion
(
    EmailID INT,
    VersionID INT,
    PRIMARY KEY (EmailID),
    FOREIGN KEY (EmailID) REFERENCES Email(id),
    FOREIGN KEY (VersionID) REFERENCES Version(id)
);

CREATE TABLE IF NOT EXISTS EmailSubject
(
    EmailID INT,
    SubjectLineID INT,
```

```
    PRIMARY KEY (EmailID),
    FOREIGN KEY (EmailID) REFERENCES Email(id),
    FOREIGN KEY (SubjectLineID) REFERENCES SubjectLine(id)

);

CREATE TABLE IF NOT EXISTS EmailAudience
(
    EmailID INT,
    AudienceID INT,
    PRIMARY KEY (EmailID),
    FOREIGN KEY (EmailID) REFERENCES Email(id),
    FOREIGN KEY (AudienceID) REFERENCES Audience(id)

);

CREATE TABLE IF NOT EXISTS Domain
(
    DomainID INT AUTO_INCREMENT,
    DomainName VARCHAR(64),
    PRIMARY KEY (DomainID)
);

CREATE TABLE IF NOT EXISTS EmailAddress
(
    EmailAddressID INT(32) NOT NULL,
    CustomerID VARCHAR(32),
    DomainID INT,
    PRIMARY KEY (EmailAddressID),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (DomainID) REFERENCES Domain(DomainID)
);

CREATE TABLE IF NOT EXISTS EmailSentTo(
   emailAddressID INT,
   EmailID INT,
   PRIMARY KEY (EmailID),
   FOREIGN KEY (emailAddressID) REFERENCES EmailAddress(emailAddressID),
   FOREIGN KEY (EmailID) REFERENCES Email(id)
);

CREATE TABLE IF NOT EXISTS Link
(
   LinkID INT AUTO_INCREMENT,
   LinkName VARCHAR(255),
   LinkURL VARCHAR(255),
   EmailID INT,
   PRIMARY KEY (LinkID),
   UNIQUE KEY (LinkID, LinkName, LinkURL, EmailID),
   FOREIGN KEY (EmailID) REFERENCES Email(id)
);
```

```
CREATE TABLE IF NOT EXISTS EmailEvent(
   eventID INT AUTO_INCREMENT,
   eventType varchar(32),
   eventDate DATETIME,
   EmailID INT,
   emailAddressID INT,
   linkID INT,
   PRIMARY KEY (eventID),
   UNIQUE KEY (eventType, eventDate, EmailID, emailAddressID, linkID),
   CONSTRAINT d FOREIGN KEY (EmailID) REFERENCES EmailSentTo(EmailID),
   FOREIGN KEY (emailAddressID) REFERENCES EmailSentTo(emailAddressID),
   FOREIGN KEY (linkID) REFERENCES Link(LinkID)
);
```