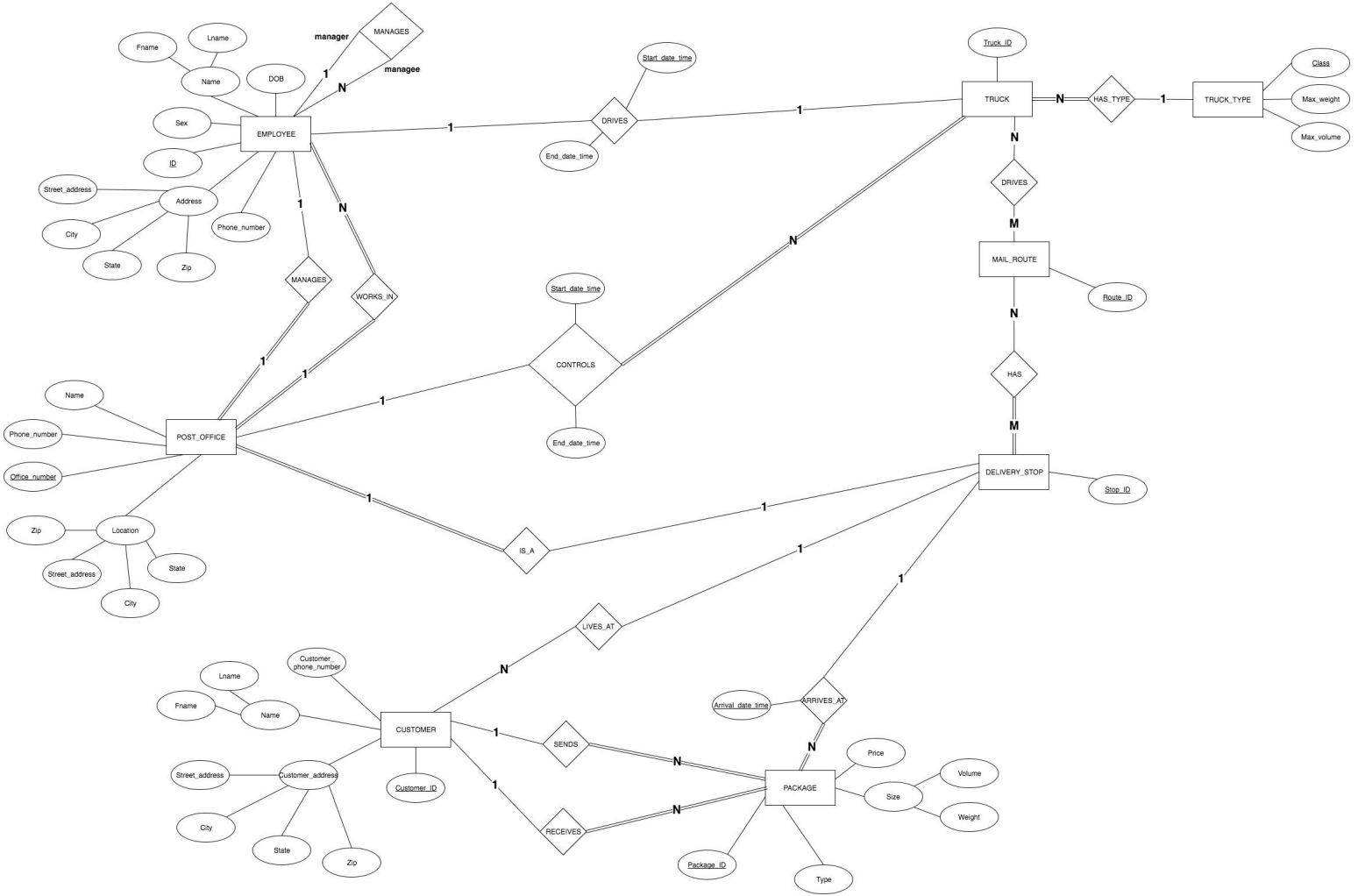
Going Postal

Perry A. , Jasenko C. ,
Nicolas K. , Sawyer K. ,
Elias M.

About Our Database

- Our Database revolves around a Post Office/Delivery Service application.
  - This Database stores information about a nationwide service that keeps track of Post offices, Employees, Customers, Trucks, Packages, Routes, and more.
- Assumptions:
  - Employees who are managers cannot drive trucks.
  - Every route stop on a route is visited when a truck drives the route.
  - Any post office can use a truck, but only one post office can use one truck at a time.
  - All packages are shipped via normal shipping (so no express shipping). Therefore price is a function of type, weight, and volume.
  - Package price is determined by external factors such as shipping speed, in addition to type, weight, and volume.
  - All weights are in pounds and all volumes are in cubic inches.
  - Each zip code only has one post office.
  - Not all CUSTOMERS live at a DELIVERY_STOP, for example if a CUSTOMER moves somewhere where mail is not delivered

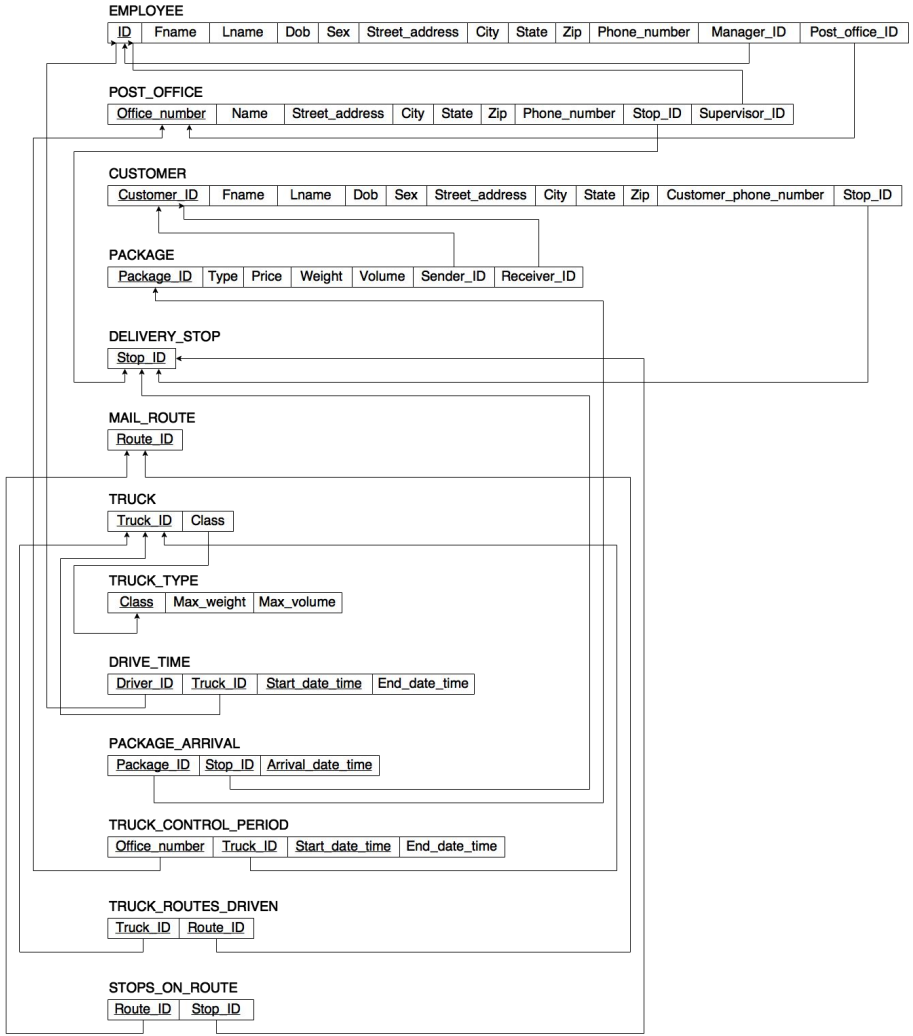
ER Diagram


Fname
Lname
Name
DOB
Sex
ID
EMPLOYEE
manager
MANAGES
manageе
Street_address
City
State
Zip
Address
Phone_number
MANAGES
WORKS_IN
Start_date_time
End_date_time
DRIVES
Truck_ID
TRUCK
HAS_TYPE
TRUCK_TYPE
Class
Max_weight
Max_volume
DRIVES
MAIL_ROUTE
Route_ID
HAS
DELIVERY_STOP
Stop_ID
Start_date_time
End_date_time
CONTROLS
Name
Phone_number
Office_number
POST_OFFICE
Zip
Location
Street_address
City
State
IS_A
LIVES_AT
Customer_phone_number
Lname
Fname
Name
CUSTOMER
Customer_ID
Street_address
Customer_address
City
State
Zip
SENDS
RECEIVES
Arrival_date_time
ARRIVES_AT
PACKAGE
Price
Volume
Size
Weight
Package_ID
Type

About Our ER Diagram

- Recursive relationships:
  - A manager (EMPLOYEE) manages employees (EMPLOYEE)
- Binary Relationships:
  - ROUTE and STOPS_ON_ROUTE
  - EMPLOYEE and TRUCK
  - DELIVERY_STOP and PACKAGE
  - TRUCK and TRUCK_TYPE
- Cardinality Ratios:
  - 1:1 : 1 EMPLOYEE drives 1 TRUCK at a given time
  - 1:N : 1 CUSTOMER can send N PACKAGES
  - N:1 : N PACKAGES arrive at 1 DELIVERY_STOP
  - N:M : N TRUCKS drive M ROUTES
- Total Participation Relationships:
  - EMPLOYEES work in a POST_OFFICE
  - A manager (EMPLOYEE) manages a POST_OFFICE
  - POST_OFFICE is a DELIVERY_STOP
  - A CUSTOMER lives at a DELIVERY_STOP

Relationship Schema



Relationship Schema Notes

EMPLOYEE: Holds information about post office employees, some of which are drivers.
POST_OFFICE: Holds information about a specific post office.
CUSTOMER: Holds information about people who can send or receive mail.
PACKAGE: Holds information about a specific package.
DELIVERY_STOP: Describes a stop a driver can deliver mail to (either a POST_OFFICE or a CUSTOMER's residence).
MAIL_ROUTE: A collection of DELIVERY_STOPS that a driver will visit when driving that route.
TRUCK: Holds information about a specific truck.
TRUCK_TYPE: Holds information about a certain class of truck.
DRIVE_TIME: Holds information about what time an EMPLOYEE was driving a TRUCK.
PACKAGE_ARRIVAL: Holds information about what time a PACKAGE was dropped off at a delivery stop.
TRUCK_CONTROL_PERIOD: Holds information about which POST_OFFICE is using a TRUCK for a certain period of time. Useful for keeping track of long-distance routes, such as Los Angeles to New York.
TRUCK_ROUTES_DRIVEN: Keeps track of which TRUCK has driven a specific ROUTE.
STOPS_ON_ROUTE: Holds information about which DELIVERY_STOPS are on a specific route. DELIVERY_STOPS can be assigned to multiple ROUTES.

Functional Dependencies

EMPLOYEE

Employee_ID	Fname	Lname	Dob	Sex	Street_address	City	State	Zip	Phone_number	Manager_ID	Post_office_ID

FD1

POST_OFFICE

Office_number	Name	Street_address	City	State	Zip	Phone_number	Stop_ID	Supervisor_ID

FD2

CUSTOMER

Customer_ID	Fname	Lname	Dob	Sex	Street_address	City	State	Zip	Customer_phone_number	Stop_ID

FD3

PACKAGE

Package_ID	Type	Price	Weight	Volume	Sender_ID	Receiver_ID

FD4

DELIVERY_STOP

Stop_ID

MAIL_ROUTE

Route_ID

TRUCK

Truck_ID	Class

FD5

TRUCK_TYPE

Class	Max_weight	Max_volume

FD6

DRIVE_TIME

Driver_ID	Truck_ID	Start_date_time	End_date_time

FD7

PACKAGE_ARRIVAL

Package_ID	Stop_ID	Arrival_date_time

TRUCK_CONTROL_PERIOD

Office_number	Truck_ID	Start_date_time	End_date_time

FD8

TRUCK_ROUTES_DRIVEN

Truck_ID	Route_ID

STOPS_ON_ROUTE

Route_ID	Stop_ID

Normalization Form

- All tables present in our project are in BCNF.
- Additional Notes:
  - Because we found rare instances where a zip code could not accurately determine a city and state (e.g., there are zip codes present that cross state lines), we do not have the functional dependency Zip → {City, State} for EMPLOYEE, CUSTOMER, and POST_OFFICE.
  - PACKAGE Price is determined by factors not represented in our table, such as shipping speed. Therefore, the price is determined by a worked at the post office and there is no functional dependency such as {Type, Weight, Volume} -> Price.

Query Examples Pt. 1

 # List employee information for all employees who work in a Post Office located in CA

SELECT Emp_id, Emp_fname, Emp_lname, Emp_sex, Emp_DOB FROM EMPLOYEE, POST_OFFICE WHERE Emp_postofficeID = PO_ID AND PO_state = 'CA';

```
sqlite> SELECT Emp_id, Emp_fname, Emp_lname, Emp_sex, Emp_DOB FROM EMPLOYEE, POST_OFFICE WHERE Emp_postofficeID = PO_ID AND PO_state = 'CA';
Emp_ID      Emp_fname   Emp_lname   Emp_sex     Emp_DOB
----------  ----------  ----------  ----------  ----------
E99612      Chee        Tos         M           1977-04-05
E74015      Jill        Heinz       F           1991-08-12
E21882      Bill        Boggut      M           1979-04-12
E77665      Joann       Mauch       F           1990-03-09
```

 # List customer information for all customers who live on the West Coast (via Zip)

SELECT * FROM CUSTOMER WHERE Cus_ZIP BETWEEN 90028 AND 98036;

```
sqlite> SELECT * FROM CUSTOMER WHERE Cus_ZIP BETWEEN 90028 AND 98036;
Cus_ID      Cus_fname   Cus_lname   Cus_DOB     Cus_sex     Cus_streetaddress  Cus_city    Cus_state   Cus_ZIP     Cus_phonenumber  Cus_stopID
----------  ----------  ----------  ----------  ----------  -----------------  ----------  ----------  ----------  ---------------  ----------
C00000      Buster      Keaton      1992-01-10  M           222 10th St        Bellevue    WA          98006       2064426464       S11424
C03401      Gina        Smith       1990-05-11  F           643 111th ave      Bellevue    WA          98008       2064311464       S25352
C41411      Andy        Garcia      1974-06-07  M           555 190th ave      Bellevue    WA          98008       2064656859       S23457
C13525      Bill        Nye         1944-04-01  M           153 1st st         Bellevue    WA          98006       2069387464       S27252
C63425      Shane       Lol         2000-01-20  M           475 124 st         Bellevue    WA          98008       2069183518       S23252
C63730      Michael     Clark       1970-11-12  M           501 Cherry St      Lynnwood    WA          98036       2069326464       S11525
C52561      Julie       Nonickel    1979-02-14  F           508 Tree ave       Lynnwood    WA          98036       2043343278       S71525
C90656      Conor       Mark        1981-07-15  M           965 M ave          Lynnwood    WA          98036       2067340098       S75524
C87656      Elijah      Edelweiss   1984-03-17  M           675 Mellow st      Lynnwood    WA          98036       2067340539       S75127
C04264      Kim         Night       1987-06-23  F           699 29th st        Lynnwood    WA          98036       2069164053       S45827
C86300      Kelly       Louis       1964-12-10  F           6781 Baker St      Los Angele  CA          90028       2062347664       S22124
C26430      Heather     Johnson     1980-09-25  F           753 Sprint ave     Los Angele  CA          90028       2132337744       S32724
C26311      Jack        Rivers      1981-08-11  M           212 Sprint ave     Los Angele  CA          90028       2137137121       S51249
C24124      Nancy       Esther      1985-10-09  F           899 50th st        Los Angele  CA          90028       2134441651       S57549
C24871      Jim         Drew        1990-06-11  M           149 51th st        Los Angele  CA          90028       2133541511       S90519
sqlite>
```

Query Examples Pt.2

 # List the ID, full name, and full addresses of all customers who have both sent and received a package. Sort by the first name of the customer

SELECT DISTINCT Cus_ID, Cus_fname, Cus_lname, Cus_streetaddress, Cus_city, Cus_state FROM CUSTOMER, PACKAGE WHERE Cus_ID = Pack_senderID AND Cus_ID IN (SELECT Pack_receiverID FROM PACKAGE) ORDER BY CUSTOMER.Cus_fname;

```
sqlite> SELECT DISTINCT Cus_ID, Cus_fname, Cus_lname, Cus_streetaddress, Cus_city, Cus_state FROM CUSTOMER, PACKAGE WHERE Cus_ID = Pack_senderID AND Cus_ID IN
   ...> ( SELECT Pack_receiverID FROM PACKAGE)
   ...> ORDER BY CUSTOMER.Cus_fname;
Cus_ID      Cus_fname   Cus_lname   Cus_streetaddress  Cus_city    Cus_state
----------  ----------  ----------  -----------------  ----------  ----------
C41411      Andy        Garcia      555 190th ave      Bellevue    WA
C87656      Elijah      Edelweiss   675 Mellow st      Lynnwood    WA
C15476      Ghetto      Thechoppa   123 Grand ave      New York    NY
C26430      Heather     Johnson     753 Sprint ave     Los Angele  CA
C26311      Jack        Rivers      212 Sprint ave     Los Angele  CA
C45756      Jason       Stripe      908 Jello st       New York    NY
C86300      Kelly       Louis       6781 Baker St      Los Angele  CA
C63730      Michael     Clark       501 Cherry St      Lynnwood    WA
C24124      Nancy       Esther      899 50th st        Los Angele  CA
C42699      Plain       Jane        527 Burger ave     New York    NY
sqlite>
```

Query Examples Pt.3

 # List all the information for each Truck type: including: the class, max weight, max volume, and number of trucks with that type

SELECT Truck_type_class AS Class, Truck_type_max_weight AS Weight, Truck_type_max_volume AS Volume, COUNT(*) AS Number_Of_Trucks FROM TRUCK, TRUCK_TYPE WHERE TRUCK.Truck_class = TRUCK_TYPE.Truck_type_class GROUP BY TRUCK_TYPE.Truck_type_class ORDER BY TRUCK_TYPE.Truck_type_class;

```
sqlite> SELECT Truck_type_class AS Class, Truck_type_max_weight AS Weight, Truck_type_max_volume AS Volume, COUNT(*) AS Number_Of_Trucks FROM TRUCK, TRUCK_TYPE WHERE TRUCK.Tru
ck_class = TRUCK_TYPE.Truck_type_class GROUP BY TRUCK_TYPE.Truck_type_class ORDER BY TRUCK_TYPE.Truck_type_class;
Class       Weight        Volume      Number_Of_Trucks
----------  ------------  ----------  ----------------
A           900000000000  2.6e+26     2
B           30000         9000000     3
C           10000         3000000     5
D           2000          700000      7
sqlite>
```

 # List all truck information (Truck ID) that belongs to class ‘C’

SELECT * FROM TRUCK WHERE Truck_class = 'C';

```
sqlite> SELECT * FROM TRUCK WHERE Truck_class = 'C';
Truck_ID    Truck_class
----------  -----------
T47683      C
T59237      C
T24386      C
T34687      C
T03648      C
sqlite>
```

Query Examples Pt.4

 # List how many stops a package made long with the pickup date and delivery date

SELECT PA_packageID Package_id, COUNT(*) Num_stops, MIN(PA_arrivaldate_time) Pickup_date, MAX(PA_arrivaldate_time) Delivery_date FROM PACKAGE_ARRIVAL GROUP BY PA_packageID;

Package_id	Num_stops	Pickup_date	Delivery_date
PA0057	2	2016-09-11 18:01	2016-09-16 12:41
PA1234	2	2016-09-10 08:31	2016-09-11 18:15
PA1492	3	2016-09-14 09:31	2016-09-16 17:50
PA2271	3	2016-09-11 18:31	2016-09-14 17:23
PA2387	2	2016-09-12 15:01	2016-09-13 14:14
PA3298	2	2016-09-13 20:41	2016-09-14 12:09
PA4521	3	2016-09-10 17:29	2016-09-16 08:35
PA4993	2	2016-09-13 06:47	2016-09-14 08:55
PA5210	3	2016-09-14 08:47	2016-09-20 15:38
PA5679	2	2016-09-13 06:40	2016-09-15 08:37
PA5918	3	2016-09-14 11:46	2016-09-16 20:30
PA7120	2	2016-09-12 14:11	2016-09-13 11:01
PA7651	2	2016-09-13 18:48	2016-09-17 08:31
PA8301	2	2016-09-13 06:54	2016-09-14 12:16
PA8800	3	2016-09-11 06:12	2016-09-15 07:07
PA9492	2	2016-09-11 17:38	2016-09-16 21:31
PA9834	3	2016-09-12 14:19	2016-09-20 14:21
PA9991	3	2016-09-14 10:31	2016-09-16 13:21

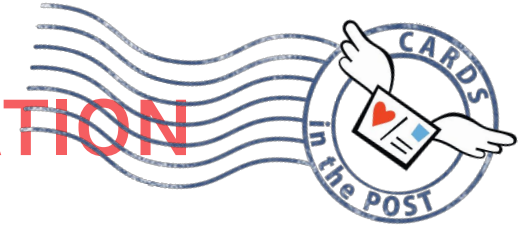
Query Examples Pt.5

 # List information about all trucks which started driving on 9/11

SELECT TCP_officenumber Controlling_PO, DT_truckID Truck_ID, DT_startdate_time Start_time, DT_enddate_time End_time FROM DRIVE_TIME, TRUCK_CONTROL_PERIOD where DT_startdate_time >= '2016-09-11' and DT_startdate_time < '2016-09-12' and DT_truckID=TCP_truckID and TCP_startdate_time <= DT_startdate_time and TCP_enddate_time >= DT_enddate_time;

Controlling_PO	Truck_ID	Start_time	End_time
P06735	T12345	2016-09-11 12:50	2016-09-15 10:19
P25986	T03648	2016-09-11 05:22	2016-09-13 14:53
P25986	T59237	2016-09-11 16:43	2016-09-14 06:50

EVALUATION



- Our Project required more time than we had originally anticipated; however, we started early and were able to complete everything in a timely manner.
- Our initial design was predominantly well thought out and constructed.
    - As a result we faced minimal roadblocks in constructing our Database
    - Our final project structure design required minimal revisions.
    - Our database did not require additional table decomposition during the Normalization phase to meet requirement minimum of 3NF per table.
    - Some queries suggested in our initial Proposal Report we found to not be possible to achieve with our final database application.
- Further aspects of our Database application that entailed more complexity than initially anticipated were:
    - Construction and relationship development with regards to package delivery & arrival times.
    - Imposed constraints to further accommodate data integrity.