CS275 Project Report – Where's My Bus

By Eric Rouse

1 Outline

The public transit system of Albany, Oregon does not presently have any bus tracking technology. It is certainly a boon to riders and to the transit manager to know, in real time, the position of each bus. Thus a database has been created to track each bus. The information collected is bus position, the current driver and the list of stops. These are then correlated to the pertinent Bus number.

2 Database Outline

2.1 Entities:

- buses – Each bus has a unique identifier.
- positions - Comprised of a unique ID number, GPS latitude, longitude, timestamp, accuracy of measurement and the bus associated with the position.
- stops – list of all the stops, made up of a unique id, the stop name and stop position.
- drivers – First and Last names of each driver a unique id and a user submitted driver rating.

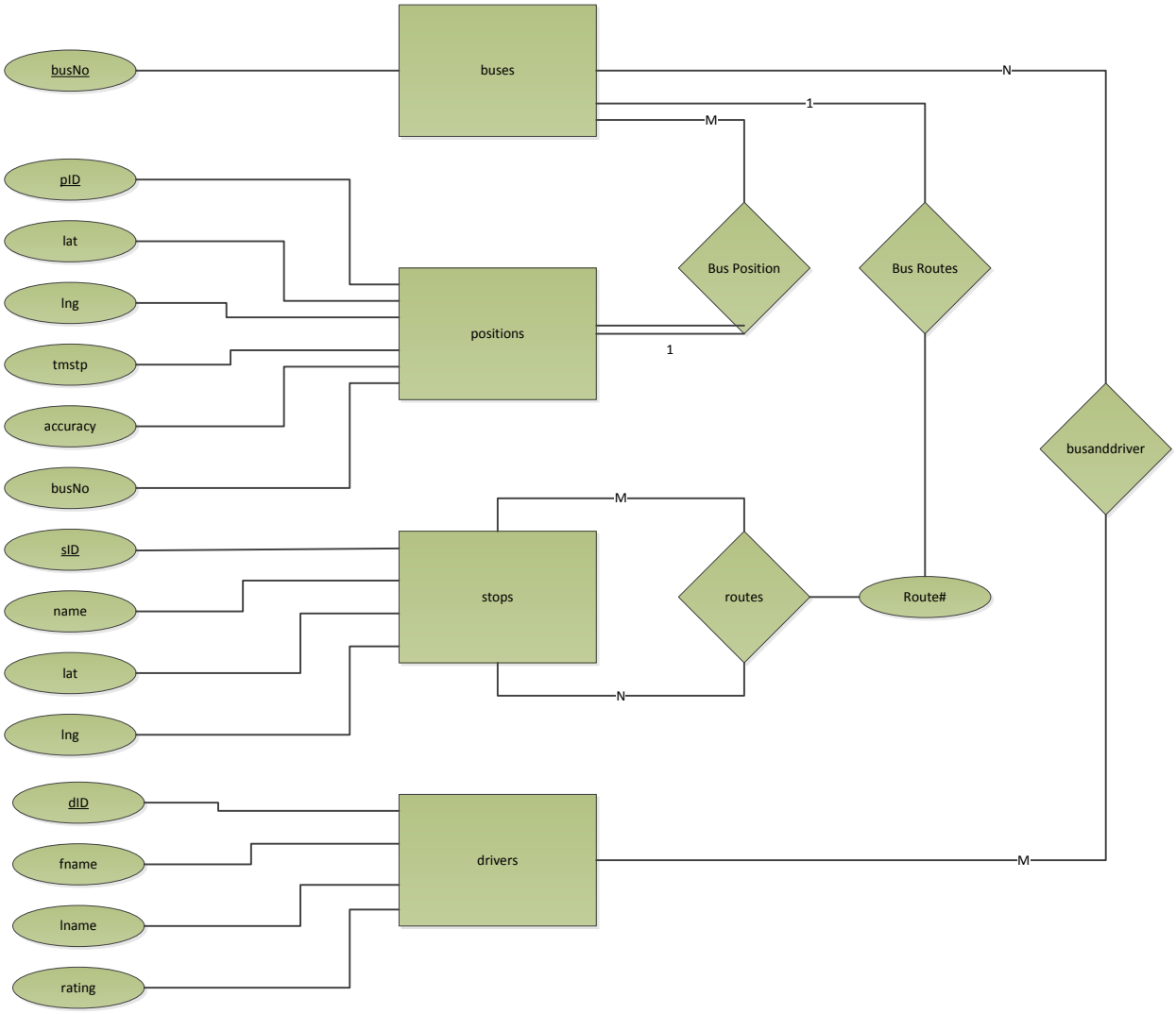
2.2 Relationships:

- Any driver can drive any bus. This is a many-to-many relationship.
- Any bus can be on any route. This is a many-to-many relationship.
- A bus has a position.
- A route is a list of stops.

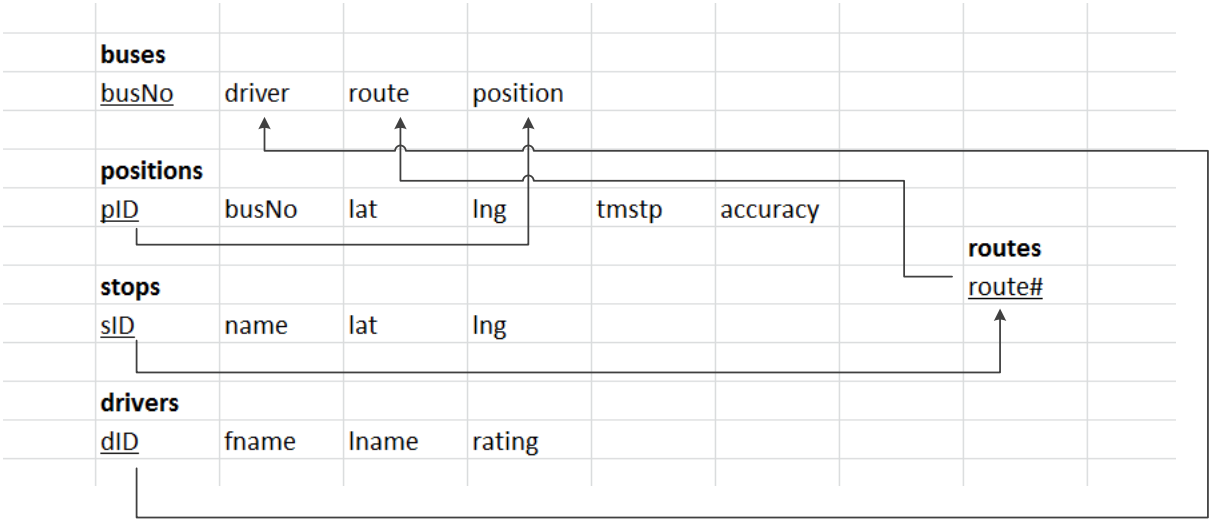
2.3 How Relationships are accomplished.

- Driver/Bus: This is accomplished with a relationship table, busanddrivers.
- Bus/Route: This is accomplished with a relationship table, routes.
- Bus Position: This is accomplished using a foreign key in the position table.
- Route/Stops: There are four routes. Each route has its own table, route1, route2, route3, route4.

3 ER DIAGRAM


busNo
buses
N
1
M
pID
lat
lng
tmstp
accuracy
busNo
positions
Bus Position
Bus Routes
1
busanddriver
M
sID
name
lat
lng
stops
routes
Route#
N
dID
fname
lname
rating
drivers
M

4 SCHEMA



5 TABLE CREATION QUERIES

Also included as “Tables.sql”.

```
#Create tables queries:


#buses table
CREATE TABLE buses(
busNo INT PRIMARY KEY AUTO_INCREMENT
) ENGINE = INNODB;


#positions table
CREATE TABLE positions(
pID INT PRIMARY KEY AUTO_INCREMENT ,
busNo INT NOT NULL ,
lat FLOAT NOT NULL ,
lng FLOAT NOT NULL ,
tmstp TIMESTAMP,
accuracy INT,
FOREIGN KEY ( busNo ) REFERENCES buses (busNo)
```

```
) ENGINE = INNODB;

#stops table
CREATE TABLE stops(
sID INT PRIMARY KEY AUTO_INCREMENT ,
name VARCHAR( 255 ) ,
lat FLOAT NOT NULL ,
lng FLOAT NOT NULL
) ENGINE = INNODB

#drivers table
CREATE TABLE drivers(
dID INT PRIMARY KEY AUTO_INCREMENT ,
fname VARCHAR( 255 ) ,
lname VARCHAR( 255 ) ,
rating INT
) ENGINE = INNODB;

#bus and driver relationship table
CREATE TABLE busanddriver(
bus INT NOT NULL ,
driver INT NOT NULL ,
FOREIGN KEY ( bus ) REFERENCES buses( busNo ) ,
FOREIGN KEY ( driver ) REFERENCES drivers( dID )
) ENGINE = INNODB;

#route tables
CREATE TABLE routes(
route INT PRIMARY KEY ,
bus INT,
FOREIGN KEY ( bus ) REFERENCES buses( busNo )
```

```
) ENGINE = INNODB;
#populate routes table
INSERT INTO routes( )
VALUES ( 1, NULL );
INSERT INTO routes( )
VALUES ( 2, NULL );
INSERT INTO routes( )
VALUES ( 3, NULL );
INSERT INTO routes( )
VALUES ( 4, NULL );
#other route tables
CREATE TABLE route1(
stop INT,
FOREIGN KEY (stop) REFERENCES stops ( sID)
) ENGINE = INNODB;
CREATE TABLE route2(
stop INT,
FOREIGN KEY (stop) REFERENCES stops ( sID)
) ENGINE = INNODB;
CREATE TABLE route3(
stop INT,
FOREIGN KEY (stop) REFERENCES stops ( sID)
) ENGINE = INNODB;
CREATE TABLE route4(
stop INT,
FOREIGN KEY (stop) REFERENCES stops ( sID)
) ENGINE = INNODB;
```

6 GENERAL USE QUERIES

Also included as PHP files.

```
UPDATE busanddriver SET driver = [drverSelect] WHERE bus = [busSelect]
```

```
INSERT INTO positions(pID, busNo, lat, lng, tmstp, accuracy) VALUES
(NULL, [busSelect],[latInput],[lngInput],NULL,[accuracyInput])

DELETE FROM positions WHERE busNo = [busSelect]

UPDATE routes SET bus = NULL WHERE bus = [busSelect}

INSERT INTO drivers(dID, fname, lname, rating) VALUES
(NULL,[fnameInput],[lnameInput],[ratingInput])

INSERT INTO buses() VALUES (NULL)

INSERT INTO busanddriver () VALUES ((SELECT MAX(busNo) FROM buses),
999)

INSERT INTO stops(name, lat, lng) VALUES
([nameEntry],[latInput],[lngInput])

INSERT INTO route1 () VALUES ([stopSelection])

INSERT INTO route2 () VALUES ([stopSelection])

INSERT INTO route3 () VALUES ([stopSelection])

INSERT INTO route4 () VALUES ([stopSelection])

DELETE FROM route1 WHERE stop = [stopSelection]

DELETE FROM route2 WHERE stop = [stopSelection]

DELETE FROM route3 WHERE stop = [stopSelection]

DELETE FROM route4 WHERE stop = [stopSelection]

DELETE FROM busanddriver WHERE bus = [busSelection]

DELETE FROM buses WHERE busNo = [busSelection]

DELETE FROM drivers WHERE dID = [driverSelection]

DELETE FROM stops WHERE sID = [stopSelection]

UPDATE routes SET bus = [busSelection] WHERE route = [routeSelection]

SELECT route FROM routes

SELECT b.busNo, d.fname, d.lname, p.lat, p.lng, p.tmstp, r.route
           FROM buses AS b
           INNER JOIN busanddriver AS bd ON b.busNo = bd.bus
           INNER JOIN drivers AS d ON d.dID = bd.driver
           INNER JOIN positions AS p ON b.busNo = p.busNo
           INNER JOIN routes AS r ON b.busNo = r.bus
           GROUP BY r.route
```

```
SELECT busNo FROM buses

SELECT dID, fname, lname FROM drivers

SELECT busNo FROM buses

SELECT * FROM drivers

SELECT dID, fname, lname FROM drivers

SELECT * FROM positions

SELECT busNo FROM buses

SELECT pID, lat, lng FROM positions

SELECT name, lat, lng FROM stops

SELECT sID, name FROM stops

SELECT stops.name FROM route1 INNER JOIN stops ON sID = stop

SELECT stops.name FROM route2 INNER JOIN stops ON sID = stop

SELECT stops.name FROM route3 INNER JOIN stops ON sID = stop

SELECT stops.name FROM route4 INNER JOIN stops ON sID = stop

SELECT stops.sID, stops.name FROM stops

SELECT stops.sID, stops.name FROM route1 INNER JOIN stops ON sID =
stop

SELECT stops.sID, stops.name FROM route2 INNER JOIN stops ON sID =
stop

SELECT stops.sID, stops.name FROM route3 INNER JOIN stops ON sID =
stop

SELECT stops.sID, stops.name FROM route4 INNER JOIN stops ON sID =
stop
```

7 LINK TO WEBSITE

http://web.engr.oregonstate.edu/~rousee/CS275/