



TEAM NUMBER 67

Phase [3]

Team Member 1 (dxiao36)
Team Member 2 (hyin62)
Team Member 3 (xgu64)
Team Member 4 (she89)

```
-- Create tables
CREATE TABLE User(
  Username      varchar(50),
  Password      int NOT NULL, -- <=====(Can be INT, CHAR, VARCHAR, or BLOB)
  IsAdmin       boolean NOT NULL,
  PRIMARY KEY (Username)
) ENGINE=InnoDB;
```

```
CREATE TABLE Passenger(
  Username      varchar(50),
  Email         varchar(50) NOT NULL,
  PRIMARY KEY (Username), -- <=====(Can be Email also, username is better
  though)
  UNIQUE (Email),
  FOREIGN KEY (Username) REFERENCES User(Username)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;
```

```
CREATE TABLE Breezecard(
  BreezecardNum char(16),
  Value         decimal(6,2) NOT NULL,
  BelongsTo     varchar(50),
  PRIMARY KEY (BreezecardNum),
  FOREIGN KEY (BelongsTo) REFERENCES Passenger(Username) -- <=====(Can
  also reference Email)
  ON DELETE SET NULL ON UPDATE CASCADE, -- <== Must be SET NULL
  CHECK (Value >= 0.00 AND Value <= 1000.00)
) ENGINE=InnoDB;
```

```
CREATE TABLE Conflict(
  Username      varchar(50), -- <=====(Can also be Email, although username is
  better)
  BreezecardNum char(16),
  DateTime      timestamp NOT NULL,
  CONSTRAINT Pk_Conflict PRIMARY KEY (Username, BreezecardNum),
  FOREIGN KEY (Username) REFERENCES Passenger(Username) -- <=====(Can
  also reference Email)
  ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (BreezecardNum) REFERENCES Breezecard(BreezecardNum)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;
```

```
CREATE TABLE Station(
  StopID       varchar(50),
  Name         varchar(50) NOT NULL,
```

```

    EnterFare    decimal(4,2) NOT NULL,
    ClosedStatus boolean NOT NULL,
    IsTrain      boolean NOT NULL,
    PRIMARY KEY (StopID),
    UNIQUE (Name, IsTrain),
    CHECK (EnterFare >= 0.00 AND EnterFare <= 50.00)
) ENGINE=InnoDB;

```

```

CREATE TABLE Trip(
    Tripfare    decimal(4,2) NOT NULL,
    StartTime   timestamp,
    BreezecardNum char(16),
    StartsAt    varchar(50) NOT NULL,
    EndsAt      varchar(50),
    CONSTRAINT Pk_Trip PRIMARY KEY (StartTime, BreezecardNum),
    FOREIGN KEY (BreezecardNum) REFERENCES Breezecard(BreezecardNum)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (StartsAt) REFERENCES Station(StopID)
        ON DELETE RESTRICT ON UPDATE CASCADE, -- <===(ON DELETE SET
NULL ok too)
    FOREIGN KEY (EndsAt) REFERENCES Station(StopID)
        ON DELETE RESTRICT ON UPDATE CASCADE -- <===(ON DELETE SET
NULL ok too)
) ENGINE=InnoDB;

```

```

CREATE TABLE BusStationIntersection(
    StopID      varchar(50),
    Intersection varchar(255), -- <====(OK to be NOT NULL)
    PRIMARY KEY (StopID),
    FOREIGN KEY (StopID) REFERENCES Station(StopID)
        ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;

```

```

-- Our tasks
--

```

```

-- Sign in

```

```

-- For users to sign in if the username and password match an existed administrator
account
SELECT * FROM User WHERE Username='$username' AND Password='$password'
And IsAdmin='1';

```

-- For users to sign in if the username and password match an existed passenger account
SELECT * FROM User WHERE Username='\$username' AND Password='\$password'
And IsAdmin='0';

-- Register

-- Get existed passenger accounts' information to check if username and email address are both unique
SELECT * FROM Passenger;

-- If passenger selects to use existing Breezecard for registration, check if the user's input breezecard number exists in the database
SELECT BreezecardNum FROM Breezecard WHERE
BreezecardNum='\$BreezecardNum';

-- If all inputs satisfy the requirements insert passenger information into User table in database, default passenger
INSERT INTO User (Username, Password, IsAdmin) VALUES('\$Username',
'\$Password', '0');

-- If all inputs satisfy the requirements insert passenger information into Passenger table in database
INSERT INTO Passenger (Username, Email) VALUES('\$Username', '\$Email');

-- Check if passenger's input Breezecard number already belongs to another passenger
SELECT BreezecardNum, BelongsTo FROM Breezecard WHERE BelongsTo
!='\$BelongsTo' AND BreezecardNum ='\$BreezecardNum';

-- If passenger's input Breezecard belongs to another passenger, generate a Breezecard randomly to this passenger and insert card info into Breezecard table
INSERT INTO Breezecard (BreezecardNum, Value, BelongsTo)
VALUES('\$BreezecardNum', '0', '\$BelongsTo');

-- If passenger's input Breezecard number already belongs to another passenger, suspend this card and put information into Conflict table
INSERT INTO Conflict (Username, BreezecardNum, DateTime)VALUES('\$Username',
'\$BreezecardNum', NOW());

-- If passenger's input Breezecard number does not belong to another passenger, update Breezecard table and allocate this card to this passenger
UPDATE Breezecard SET BelongsTo='\$Username' WHERE
BreezecardNum='\$BreezecardNum';

-- Station Management

-- For an administrator, view all the stations

```
SELECT Name, StopID, EnterFare, ClosedStatus FROM Station;
```

-- Create New Station

-- Obtain information from database to check whether the input data has confliction with database data by same name and type or by same StopID

```
SELECT Name, StopID, IsTrain FROM Station;
```

-- After checking data confliction requirements, insert the user input station info into database station table

```
INSERT INTO Station (StopID, Name, EnterFare, ClosedStatus, IsTrain)VALUES('$StopId', '$Name', '$EnterFare', '$ClosedStatus', '$IsTrain');
```

-- After checking data confliction requirements, if the newly created station is a bus station, insert the user input station info into database BusStationIntersection table

```
INSERT INTO BusStationIntersection (StopID, Intersection)VALUES('$StopID', '$Intersection');
```

-- View Station

-- Update a particular station's fare by matching its stopID

```
UPDATE Station SET EnterFare='Fare' WHERE StopID='StopID';
```

-- View a bus station's intersection by matching its stopID

```
SELECT Intersection FROM BusStationIntersection WHERE StopID = 'StopID';
```

-- View the open/close status of a station by matching its stopID

```
SELECT ClosedStatus FROM Station WHERE StopID='StopID';
```

-- Update the open/close status of a station by matching its stopID

```
UPDATE Station SET ClosedStatus='$ClosedStatus' WHERE StopID='StopID';
```

-- Suspended Cards

-- Display all the suspended cards in the order of breezecard number, new owner, date suspended and previous owner

```
SELECT T.BreezecardNum, T.BelongsTo, T.DateTime, T.Username FROM(SELECT * FROM Conflict LEFT JOIN (SELECT BreezecardNum as s, BelongsTo FROM Breezecard) B ON Conflict.BreezecardNum = B.s)T;
```

```

-- Update this suspended card to either new or old owner
UPDATE Breezecard SET BelongsTo='Usermae' WHERE
BreezecardNum='BreezecardNum';

-- Checking if new user would hold a breezecard after the conflict has been resolved
SELECT T.Username as Username FROM (SELECT Username FROM Conflict
WHERE BreezecardNum='$BreezecardNum'AND Username != '$Username')T WHERE
Username NOT IN (SELECT BelongsTo FROM Breezecard);

-- Insert a breezecard into breeze
INSERT INTO Breezecard (BreezecardNum, Value, BelongsTo)
VALUES('BreezecardNum', '0', 'Username');

-- Checking if old user would hold a breezecard after the conflict has been resolved
SELECT DISTINCT BelongsTo FROM Breezecard WHERE BelongsTo='Username';

-- Delete a breezecard and all its relevant information from Conflict table after conflict
resolution
DELETE FROM Conflict WHERE BreezecardNum='BreezecardNum';

-- Breezecard Management

-- Display all the Breezecard information that satisfy the administrator's inputs
requirements
SELECT BreezecardNum, Value, BelongsTo FROM Breezecard WHERE
BreezecardNum NOT IN (SELECT BreezecardNum FROM Conflict) AND
BelongsTo='$owner' AND BreezecardNum='$cardnum' AND Value >='$minValue' AND
Value <='$maxValue';

-- Display 'suspended' for all suspended cards
SELECT count(1) from Conflict WHERE BreezecardNum = '$BreezecardNum';

-- Update Breezecard information after an administrator changes a Breezecard value
UPDATE Breezecard SET Value='$Value'WHERE BreezecardNum='$BreezecardNum';

-- Before transferring selected card to a user, first check if this user exist in the database
SELECT Username FROM Passenger WHERE Username='$Username';

-- Check if this selected card is suspended
SELECT DISTINCT BreezecardNum FROM Conflict WHERE
BreezecardNum='$BreezecardNum';

-- Assign this Breezecard to the selected user
UPDATE Breezecard SET BelongsTo='BelongsTo' WHERE
BreezecardNum='$BreezecardNum';

```

-- Check if the new and previous owner associated with this card still have a Breezecard after transferring

```
SELECT T.Username FROM (SELECT Username, BreezecardNum FROM Conflict
WHERE BreezecardNum = '$BreezecardNum' AND Username NOT IN (SELECT
BelongsTo FROM Breezecard WHERE BelongsTo IN (SELECT Username FROM
Conflict WHERE BreezecardNum = '$BreezecardNum'))))T;
SELECT BreezecardNum FROM Breezecard WHERE BelongsTo='$BelongsTo';
```

-- If any of the previous or new owners have no Breezecard, assign a new card, insert information into Breezecard table

```
INSERT INTO Breezecard (BreezecardNum, Value, BelongsTo)
VALUES('$BreezecardNum', '0', '$BelongsTo');
```

-- Resolve conflicts, delete the records from Conflict table

```
DELETE FROM Conflict WHERE BreezecardNum='$BreezecardNum';
```

-- Passenger Flow Report

-- Create a view with all trips that satisfy the limited time range set by an administrator

```
CREATE OR REPLACE VIEW InitialTrip AS SELECT Tripfare, StartTime,
BreezecardNum, StartsAt, EndsAt From Trip WHERE StartTime >='2016-01-01
12:59:59' AND StartTime <='2018-01-01 03:00:02' AND 1;
```

-- Create a view with all stations with the number of passengers entering the station, the sum of fares for this station in this time period

```
CREATE OR REPLACE VIEW passengerIn AS SELECT StartsAt AS station,
SUM(Tripfare) AS sumfare, COUNT(BreezecardNum) AS p_in, StartTime From
InitialTrip Group by station;
```

-- Create a view with all stations and the number of passengers exiting the station in this time period

```
CREATE OR REPLACE VIEW passengerOut AS SELECT EndsAt AS station,
COUNT(BreezecardNum) AS p_out, StartTime FROM InitialTrip GROUP BY station;
```

-- Create a view with all stations with passengers entering and if there are passengers exiting the station combine into the same tuple

```
CREATE OR REPLACE VIEW passengerInJoinOut AS SELECT station, p_in, p_out,
sumfare, StartTime FROM passengerIn NATURAL LEFT JOIN passengerOut;
```

-- Create a view with all stations with passengers exiting and if there are passengers entering the station combine into the same tuple

```
CREATE OR REPLACE VIEW passengerOutJoinIn AS SELECT station, p_in, p_out,
sumfare, StartTime FROM passengerIn NATURAL RIGHT JOIN passengerOut;
```

```
-- Create a view with all stations with passengers entering and exiting
CREATE OR REPLACE VIEW passengerInAndOut AS SELECT * FROM
passengerInJoinOut UNION SELECT * FROM passengerOutJoinIn;
```

```
-- Create passenger flow report for all stations
CREATE OR REPLACE VIEW passenger_flow_report AS SELECT Station.Name AS
Name, IFNULL(sumfare, 0) AS revenue, IFNULL(p_in, 0) AS passenger_in,
IFNULL(p_out, 0) AS passenger_out, IFNULL(IFNULL(p_in, 0)-IFNULL(p_out, 0), 0) AS
flow, station FROM passengerInAndOut INNER JOIN Station on
passengerInAndOut.station = Station.stopID;
```

```
-- Display the flow report with station name, the number of passengers entering and
exiting, the flow and revenue for all distinct stations
SELECT Name, SUM(passenger_in) AS passenger_in, SUM(passenger_out) AS
passenger_out, SUM(flow) AS flow, SUM(revenue) AS revenue FROM
passenger_flow_report GROUP BY station;
```

-- Passenger Functionality

```
-- View all non-conflicted card numbers belongs to current user
SELECT T.BreezecardNum FROM (SELECT * FROM Breezecard WHERE BelongsTo
= '$BelongsTo' AND BreezecardNum NOT IN (SELECT BreezecardNum FROM
Conflict))T;
```

```
-- View a card's value of choice
SELECT Value FROM Breezecard WHERE BreezecardNum='$BreezecardNum';
```

```
-- View all the opened stations and listing their station name, entering fare and stopID
SELECT Name, EnterFare, StopID FROM Station WHERE ClosedStatus='0';
```

```
-- View current statin's entering fare and stopID
SELECT EnterFare, StopID FROM Station WHERE StopID='$StopID';
```

```
-- To determine if the current user is in trip right now
SELECT StartsAt, Tripfare, BreezecardNum, EndsAt, StartTime FROM Trip WHERE
BreezecardNum IN (SELECT T.BreezecardNum FROM (SELECT * FROM Breezecard
WHERE BelongsTo = '$BelongsTo')T) AND EndsAt IS NULL;
```

```
-- View current user's starting station in this trip
SELECT Name, StopID FROM Station WHERE StopID='StopID';
```

```
-- Check current user's trip end station type has to be consistent with starting station's
type and this end station is open
```



```
SELECT Name, StopID, IsTrain FROM Station WHERE IsTrain IN (SELECT IsTrain
FROM (SELECT StopID, IsTrain FROM Station WHERE StopID ='$StopID')T) AND
ClosedStatus=0;
```

```
-- Update the value of a card after the trip
UPDATE Breezecard SET Value='$Value' WHERE
BreezecardNum='$BreezecardNum';
```

```
-- Record trip after the user starts their trip
INSERT INTO Trip (Tripfare, StartTime, BreezecardNum, StartsAt,
EndsAt)VALUES('$Tripfare', NOW(), '$BreezecardNum', '$StartsAt', NULL);
```

```
-- Update trip's end station after the trip completed
UPDATE Trip SET EndsAt='$EndsAt' WHERE BreezecardNum='$BreezecardNum'
AND StartTime='$StartTime';
```

```
-- Manage Breeze Cards
```

```
-- View all the breezecards with their values that belongs to a particular user
SELECT T.BreezecardNum , T.Value FROM (SELECT * FROM Breezecard WHERE
BelongsTo = 'Username' AND BreezecardNum NOT IN (SELECT BreezecardNum
FROM Conflict))T;
```

```
-- Check if the entered potential new breezecard exists in database already
SELECT BreezecardNum FROM Breezecard WHERE BreezecardNum
='$BreezecardNum';
```

```
-- Check if this entered card belongs to null(no one)
SELECT BreezecardNum FROM Breezecard WHERE BreezecardNum
='$BreezecardNum' AND (BelongsTo IS NULL OR BelongsTo='Username');
```

```
-- When this entered new card belongs to null, update the user of this card to current
user
UPDATE Breezecard SET BelongsTo='Username' WHERE
BreezecardNum='$BreezecardNum';
```

```
-- When this entered new card belongs to someone else already, insert this breezecard
to conflict
INSERT INTO Conflict (Username, BreezecardNum, DateTime)VALUES('Username',
'BreezecardNum', NOW());
```

```
-- 1. When this entered card does not exists in database, add this card into database to
this current user
```

```
-- 2. When a user does not hold any card after deletion of a card, add a new card to this
user
```

```
INSERT INTO Breezecard (BreezecardNum, Value, BelongsTo)
VALUES('BreezecardNum', '0', 'Username');
```

```
-- Update a breezecard's value
```

```
UPDATE Breezecard SET Value="WHERE BreezecardNum='BreezecardNum';
```

```
-- Dissociate a breezecard from current user
```

```
UPDATE Breezecard SET BelongsTo = NULL WHERE
BreezecardNum='BreezecardNum';
```

```
-- View all the breezecard belongs to a user
```

```
SELECT BreezecardNum FROM Breezecard WHERE BelongsTo='BreezecardNum';
```

```
-- View Trip History
```

```
-- Select certain trip histpry according to the given start time and end time
constraints(optional),
```

```
-- and does not show the cards which are suspended,
```

```
-- and shows the histories which belongs to the previous owner.
```

```
SELECT Distinct Trip.StartTime,A.Name AS Name1,B.Name AS
Name2,TripFare,Trip.BreezecardNum FROM Trip, Breezecard, Station AS A, Station
AS B WHERE Trip.BreezecardNum IN (SELECT BreezecardNum FROM Breezecard
WHERE BelongsTo = '$BelongsTo') AND Trip.StartTime >= '2016-01-01 12:59:59' AND
Trip.StartTime <= '2018-01-01 03:00:02' AND A.StopID = Trip.StartsAt AND B.StopID =
Trip.EndsAt AND Trip.BreezecardNum NOT IN (SELECT BreezecardNum FROM
Conflict) UNION SELECT Trip.StartTime, A.Name AS Name1,
EndsAt,TripFare,Trip.BreezecardNum FROM Trip, Breezecard, Station AS A, Station
AS B WHERE Trip.BreezecardNum IN (SELECT BreezecardNum FROM Breezecard
WHERE BelongsTo = '$BelongsTo') AND Trip.StartTime >= '2016-01-01 12:59:59' AND
Trip.StartTime <= '2018-01-01 03:00:02' AND Trip.EndsAt is NULL AND A.StopID =
Trip.StartsAt AND Trip.BreezecardNum NOT IN (SELECT BreezecardNum FROM
Conflict);
```