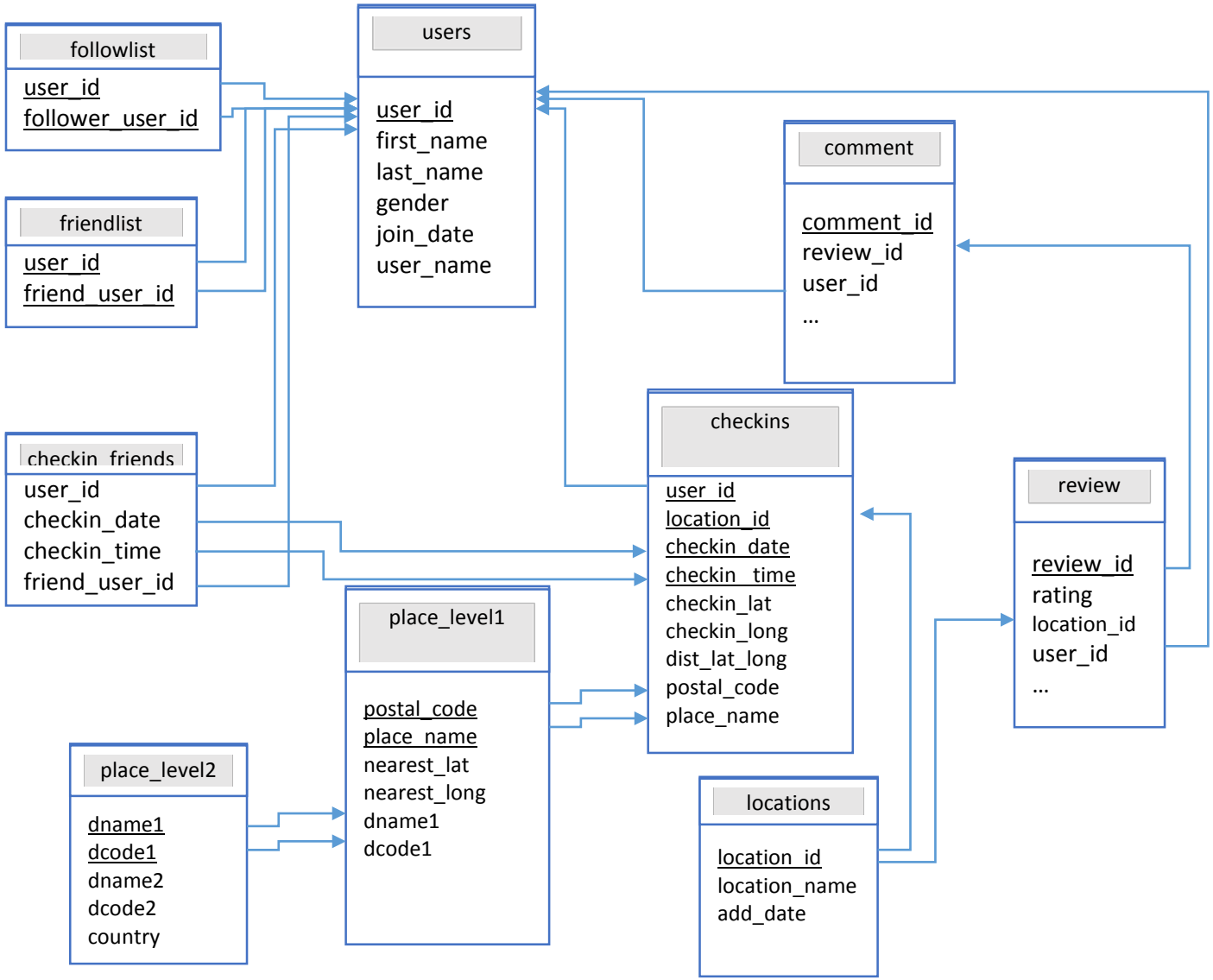
CISC637 Spring 2014, Project check point 1

Parth Patel, Felix Francis

1. **Relational schema diagram:**



2. First normal form

(a) Which of the data table(s) are not in 1NF?

followList.txt and friendList.txt are not in 1NF

(b) Write a program that will read those table(s) and normalize data into 1NF. Import the resulting normalized data into your database.

```
#!/usr/bin/python

import re

import sys

import os

inp_file_name = str(sys.argv[1]) #user defined name for the input file

out1_file_name = str(sys.argv[2]) #user defined name for the first file

out2_file_name = str(sys.argv[3]) #user defined name for the second file

out3_file_name = str(sys.argv[4]) #user defined name for the third file

#out4_file_name = str(sys.argv[5]) #user defined name for the fourth file

#ini_range = int(str(sys.argv[6]))# begin index for splitting the file

#end_range = int(str(sys.argv[7]))# last index for splitting the file

#write files

out1= open(out1_file_name,'w');
```

```
out2= open(out2_file_name,'w');

out3= open(out3_file_name,'w');

#out4= open(out4_file_name,'w');

#read files

input=open(inp_file_name, 'r')

##for line in input: ##All lines in the input file

while True:

        line = input.readline() # read the line

    if not line: break

        line=line.split('\t'); # tokenize line into small fragments of strings and store it in an array.

#       for i in range (0,5): #writing into the first file

         #print (ini_range)

         #print (line[i])

        out1.write('%s \t' % (line[0]))#writing each token in to file 1

        out1.write('%s \t' % (line[5]))

        out1.write('%s \t' % (line[1]))

        out1.write('%s \t' % (line[2]))

        out1.write('%s \t' % (line[3]))
```

```
            out1.write('%s \t' % (line[4]))
            out1.write('%s \t' % (line[8]))
            out1.write('%s \t' % (line[9]))
            out1.write('%s \t' % (line[10]))
            out1.write('\n')
#                   i=i+1
#           out1.write('\n') # newline
#           for j in range (5,10): #writing into the second file
            out2.write('%s \t' % (line[9]))#writing each token in to file 2
            out2.write('%s \t' % (line[10]))
            out2.write('%s \t' % (line[6]))
            out2.write('%s \t' % (line[7]))
            out2.write('%s \t' % (line[11]))
            out2.write('%s \t' % (line[12]))
            out2.write('\n')
#                   j=j+1
#           out2.write('\n')# new line
#           for k in range (5,12): #writing into the second file
```

```
            out3.write('%s \t' % (line[11]))#writing each token in to file 3
            out3.write('%s \t' % (line[12]))
            out3.write('%s \t' % (line[13]))
            out3.write('%s \t' % (line[14]))
            out3.write('%s \t' % (line[15]))
            out3.write('\n')
#                   k=k+1
#           out3.write('\n')# new line
#           for l in range (5,12): #writing into the second file
#           out4.write('%s \t' % (line[5]))#writing each token in to file 4
#           out4.write('%s \t' % (line[13]))
#           out4.write('%s \t' % (line[14]))
#           out4.write('\n')
            #           l=l+1
            #out4.write('\n')# new line


input.close()
out1.close()
```

```
out2.close()

out3.close()

#out4.close()


# end of this script
```

Importing data into MySQL commands:

i. mysqlimport – u ppatel –p 2486 –local ppatel followlist.txt
ii. mysqlimport – u ppatel –p 2486 –local ppatel freindlist.txt

3.

(a) checkins.txt, followList.txt, and friendList.txt

(b)

i. user_id -> first_name, last_name, gender,join_date,user_name
ii. location_id -> location_name, add_date
iii. follow_user_id, user_id -> follow_user_id, user_id
iv. freind_user_id, user_id -> freind_user_id, user_id
v. user_id, location_id, checkin_date, checkin_time -> checkin_lat, checkin_long, dist_lat_long, postal_code, place_name, nearest_lat, nearest_lat, nearest_long, dname1, dcode1, dname2, dcode2, country

(c) 5th functional dependency violates BCNF.

user_id, location_id, checkin_date, checkin_time -> checkin_lat, checkin_long, dist_lat_long, postal_code, place_name, nearest_lat, nearest_lat, nearest_long, dname1, dcode1, dname2, dcode2, country

Solution is to decompose this relation into 3 new relations to have the following functional dependencies:

i. user_id, location_id, checkin_date, checkin_time -> checkin_lat, checkin_long, dist_lat_long, postal_code, place_name

ii. postal_code, place_name -> nearest_lat, nearest_lat, nearest_long, dname1, dcode1

iii. dname1, dcode1 -> dname2, dcode2, country

(d) Code is given in the answer for 2 (b)

Importing data into MySQL commands:

i. mysqlimport – u ppatel –p 2486 –local ppatel checkins.txt
ii. mysqlimport – u ppatel –p 2486 –local ppatel place_level1.txt
iii. mysqlimport – u ppatel –p 2486 –local ppatel place_level2.txt

The original checkins.txt was divided into three files. Here are their respective schema:

1. checkins (user_id, location_id, checkin_date, checkin _time, checkin_lat, checkin_long, dist_lat_long, postal_code, place_name)

2. place_level1 (postal_code, place_name, nearest_lat, nearest_long, dname1, dcode1)

3. place_level2 (dname1, dcode1, dname2, dcode2, country)