October 13, 2020
IS620
Group 3

# IS 620 - Advanced Database Projects

*Contact Tracing System*

## Specification of Features

**Feature 1:** Add a person to the database.

Input: Name of person, home address and zip code, phone#, and status to null.

Output: Print a message in different cases

1. Person already exists
2. Newly assigned person ID with existing house
3. New assigned person ID and newly generated house ID

Example of calling the procedure: exec add_person('Jeff','123 Main Street','21044','1-443-987-1029');

**Feature 2:** List positive people's name and phone number.

Input: people whose current status is positive.

Output: List of the people names, phone numbers, and status living in the same house.

Example of calling the procedure: exec risky_people(person_ID, house_ID);

## Feature 3:

Input: Current_Status, Person_ID, Test_date, Test_result

Output: Print message if condition meets.

1. Check if person exist if not print "Error: Person does not exist"
2. If Person_ID and Test_Date match a Test "Update to new test_results" if not "Insert the person with Person_ID, Test_Date, and Test_Result in Person_Test table"
3. If latest test result "Update Current_Status to new result"

Example: exec Person_Status_Check (Person_ID, Test_date, Test_result, Current_Status)
Prints: "Error: Person does not exist" if no match found
Update: "Test_result" and "Current_Status" if "Person_ID" and "Test_Date" match
Update: "Current_Status" with new results

October 13, 2020
IS620
Group 3

## Feature 4:

Input: Person_name, Phone_number, Current_Status, Flight_ID, Flight_Date

Output: Prints person names and phone numbers that were on same flight with Person with Current_Status as Positive within last X days

Example: exec Infected_Person (Person_ID, Flight_ID, Current_Status, Phone_Number)
It will Print: Person Name and Phone Number of the Person with Current Status “Positive” and all other People on Flight ID with that Infected Person.

## Feature 5:

1) input: "Enter the event name, event date and event address"

(event_name, event_date, event_address) from event table

2) function: check if inputs exist in event_table, by matching the input of event name, event date and event address.

3) output:

a) if match: print "event already exists"

b) else: exec add_event ('2020-3-24', '199 Stone cross Rd') into event table

exec give_event;

“please enter the name of the event, date and address:”

('Lunch', '2020-3-24', '199 Stone cross Rd')

“This event does not exist, it will be added to the event list”

## Feature 6:

1) output: "these are the names and phone numbers of people who currently have a positive test"

(pname, phone) from person

2) input: function: within that date (loop) each person with positive test and match event id they

attended with other attendants who attended the same event.

3) output: print the name and phone numbers of each person until the loop ends

exec attend_event;

“This following is a list of people who were positive and attended an event”

('nancy', '1234567890')

"enter number of days since that person has attended that event"

“4”

“The following is a list of people who attended the same event as people who were infected and attended the same event”

**Feature 7:** Enter list of people attending an event.

Input: Event ID and list of person ids.

Output:

1. Print a message “event does not exist” if an event with input event id does not exist.
2. Print a message “ person does not exist” if input person id from the list is not present.
3. Print a message “No need to insert” if an input person id and input event id combination are already present in preson_event table.
4. Insert row into person_event table for event id and person id if the input person id and event id combination does not exist in Person_event table.

Example of calling the procedure: exec insert_person_event(1, personListType(1,2,3)); -- passing varray for personid

**Feature 8:** Given the name and phone number of a person, print out test dates and test results of this person.

Input: Person name and phone number

Output: The feature prints the following.

1. Print a message “No such person” if input person id and input phone number does not exist.
2. Print the results sorted by date in descending order if the person is present in the person table.

Example of calling the procedure: exec get_person_tests('Ann','8765567890');

**Feature 9:** Given a person ID and a test date that person was tested positive, print out the suspected chain of transmission.

Input: Person ID and test date including people in the same flight and event.

Output: The feature prints the following.

1. Print a message "in valid input" if there exists no person with the input person ID and tested positive on the input date D
2. Find close contacts of the person including three cases.
3. Print out names of people who may have got infected by the input person.
4. For each person identified in step 3, we repeat step 2 and 3.

Example: exec print_suspected_people(1, date '2020-10-14');

**Feature 10:** Case Statistics.

Input: Zip code and number of days.

Output: The feature prints the following.

1. Total number of people with positive status in each zip code.
2. Total number of distinctive people who tested positive in the last X days where X is the input.
3. Accumulated number of people tested positive in each zip code.

Example: exec case_stat('21043', 5);

**Feature 11:** Daily statistics.

Input: start date and end date.

Output: Print out daily statistics in the period as follows.

1. Print out daily accumulated positive cases during the period.
2. Print out daily new cases.
3. Print out daily current cases.
4. Print out daily accumulated recovered cases.

Example: exec daily_stat(date '2020-10-10', date '2020-10-14');

October 13, 2020
IS620
Group 3

**Feature 12:** Find zip codes as hot spots.

Input: Date, Number of days, and threshold.

Output: The feature prints the following.

1. Compute number of new cases for each zip code between date D-X+1 day and D.
2. Compute the number of new cases in the same zip code in the date range.
3. Print out this zip code as hot spot if if the count in step 1 is greater or equal to the count in step 2.

Example: exec print_hotspot(date ‘2020-10-14’, 5, 10);