Hemin Patel

Professor Imieliński

Principles of Information and Data Management

11 November 2021

1. Decompose Penna to BCNF scheme (2 tables). Use Create table statement to define and populate these tables from Penna. No need to submit csv files this time. Just create commands.

```
-- Location Table
Create table testDB.Location
As Select distinct p.precinct, p.locality, p.geo, p.state
From testDB.penna p

Delete from testDB.Location where geo is null; #removing null geo values

-- Votes Table
Create table testDB.Votes
As Select distinct p.precinct, p.totalvotes, p.Biden, p.Trump, p.Timestamp, p.filestamp
From testDB.penna p
```

2. Define foreign key constraint on your new tables in mySQL (using proper commands)

```
# Making precinct the primary key in Location Table
delete from Location where precinct = "Morris Voting Precinct"
        #encountered more rows when I reuploaded penna table, had to del 2 rows to make pk

ALTER TABLE testDB.Location
ADD PRIMARY KEY (precinct (50));
```

```
# Making precinct the foreign key in Votes Table
Alter table testDB.Votes
Add constraint FK_LocationVotes
Foreign key (precinct)
References location(precinct);
```

3. Write a query in MySQL veryfing that your foreign key constraint is satisfied by the instances of tables in your decomposition (for the current data which you have uploaded).

```
use testDB
Select Votes.precinct
From Votes
Right join Location ON Votes.precinct = Location.precinct
where Location.precinct is null;
```

4. Implement cascade option for your foreign key constraint using a TRIGGER.
When you delete a precinct from referenced table, all tuples with this precinct should be deleted from the referencing table.

```
use testDB
delimiter $$
Create Trigger del_Precinct
   After Delete on Votes
   For each row
   Begin
         Delete from Location
         Where Location.precinct = old.precinct;
   End $$
DELIMITER ;
```

5. Implement a trigger with two functions:

a) Triggered by updates of Trump or Biden votes in any tuple, a new tuple is added into separate table called VoteChangeLog. VotechangeLog will store precinct name, timestamp, old vote, new vote and candidate name for any update which changes votes for either of the two candidates.

```
Create table testDB.VotechangeLog(
        precinct varchar(50),
        Timestamp varchar(225),
        oldVote int,
        newVote int,
        candidate text)

delimiter $$
create trigger update_VotechangeLog
        Before update on Votes
  For each row
  Begin
          if(NEW.Trump > OLD.Trump + 1) then
          set @gg = NEW.Trump + OLD.Trump;
          insert into  VotechangeLog
          values(NEW.precinct, NEW.Timestamp, OLD.Trump, @gg, 'Trump');
        elseif (NEW.Biden > OLD.Biden + 1) THEN
          set @gg = NEW.Biden + OLD.Biden;
          insert into  VotechangeLog
          values(NEW.precinct, NEW.Timestamp, OLD.Biden, @gg, 'Biden');
         end if;
        End$$
delimiter ;
```