PROJECT - NOAH CHOE

PART 1

QUESTION 1

- PRECINCT -> GEO, STATE, LOCALITY
- GEO -> PRECINCT, STATE, LOCALITY
- PRECINCT, TIMESTAMP -> GEO, STATE, LOCALITY, TOTALVOTES, BIDEN, TRUMP, FILESTAMP, ID
- GEO, TIMESTAMP -> PRECINCT, STATE, LOCALITY, TOTALVOTES, BIDEN, TRUMP, FILESTAMP, ID

QUESTION 2

Penna is not in BCNF.

PRECINCT -> GEO, LOCALITY, STATE

PRECINCT, TIMESTAMP -> TOTALVOTES, BIDEN, TRUMP, FILESTAMP, ID

PART 2

QUESTION 1 - The Precinct

a) winner(precinct)

```
DELIMITER //

CREATE PROCEDURE winner(IN input_precinct varchar(255))
BEGIN
	DECLARE bidenvalue, trumpvalue, totalvotesvalue INT DEFAULT 0;

	SELECT biden, trump, totalvotes
	INTO bidenvalue, trumpvalue, totalvotesvalue
	FROM votes x, (SELECT max(timestamp) as maxTime FROM votes WHERE precinct =
input_precinct) y
	WHERE timestamp = maxTime and precinct = input_precinct;

	IF bidenvalue > trumpvalue THEN
		SELECT 'Biden Won' as Winner, ROUND(bidenvalue/totalvotesvalue *100, 2) as
percentage, totalvotesvalue;
	END IF;

	IF trumpvalue > bidenvalue THEN
		SELECT 'Trump Won' as Winner, ROUND(trumpvalue/totalvotesvalue *100, 2) as
percentage, totalvotesvalue;
	END IF;

END //

DELIMITER ;
CALL winner('005 ATGLEN');
```

@winner	@percentage	@totalvotes
Trump	52.02	694

b) rankall(precinct)

```
DELIMITER //
CREATE PROCEDURE rankall(IN input_precinct varchar(255))
BEGIN

   CREATE TABLE ranking (
         precinct varchar(255),
         ranks INT
   );

   INSERT INTO ranking
   SELECT precinct, RANK() over (ORDER BY totalvotes DESC)ranks
   FROM votes x, (SELECT max(timestamp) as maxTime FROM votes) y
   WHERE timestamp = maxTime
   ORDER BY totalvotes DESC;

   SELECT ranks
   FROM ranking
   WHERE precinct = input_precinct;

   DROP TABLE ranking;

END //

DELIMITER ;
CALL rankall('005 ATGLEN');
```

ranks
1056

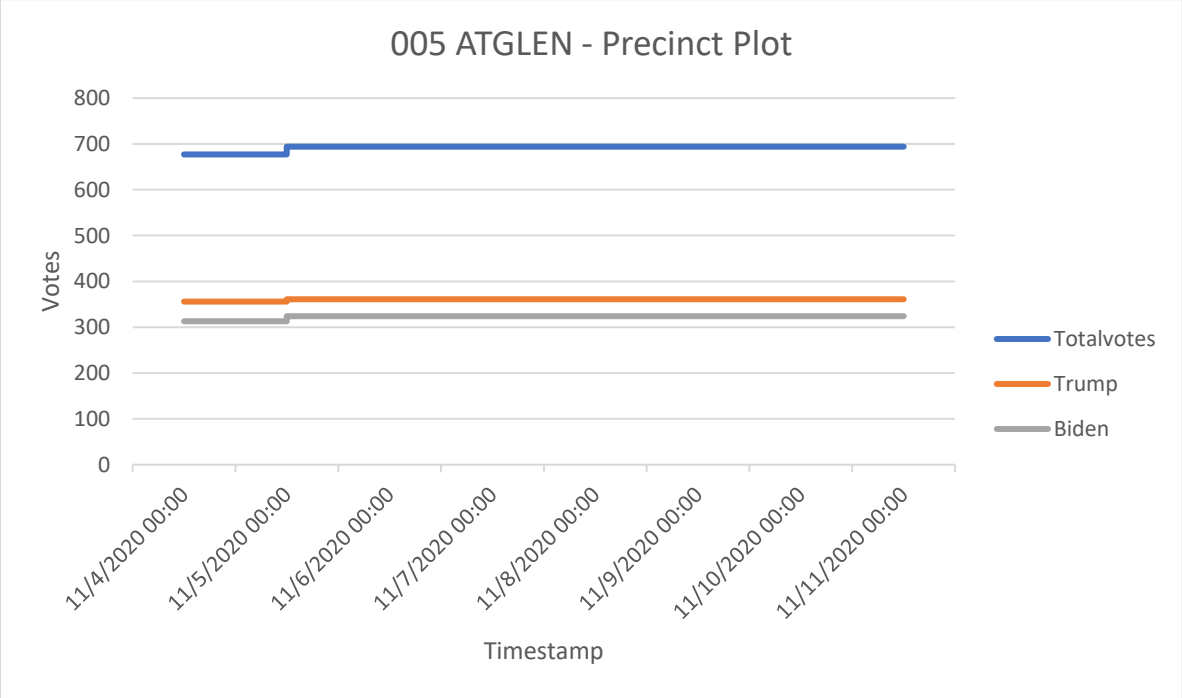
c) rankcounty(precinct)

```
DELIMITER //

CREATE PROCEDURE rankcounty(IN input_precinct varchar(255))
BEGIN

	CREATE TABLE ranking (
		precinct varchar(255),
		ranks INT
	);

	INSERT INTO ranking
	SELECT x.precinct, RANK() over (ORDER BY totalvotes DESC)ranks
		FROM votes x, place y, (SELECT max(timestamp) as maxTime FROM votes) z, (SELECT locality
FROM place WHERE input_precinct = precinct)w
	WHERE timestamp = maxTime AND y.locality = w.locality AND y.precinct = x.precinct
	ORDER BY totalvotes DESC;

	SELECT ranks
	FROM ranking
	WHERE precinct = input_precinct;

	DROP TABLE ranking;

END //

DELIMITER ;
CALL rankcounty('005 ATGLEN');
```

@ranking
194

d) plotprecinct(precinct)

```
DELIMITER //

CREATE PROCEDURE plotprecinct(IN input_precinct varchar(255))
BEGIN

    SELECT timestamp, totalvotes, trump, biden
    FROM votes x
    WHERE precinct = input_precinct
    ORDER BY timestamp ASC;

END //

DELIMITER ;
CALL plotprecinct('005 ATGLEN');
```



e) earliestprecinct(vote_count)

```
DELIMITER //

CREATE PROCEDURE earliestprecinct(IN input_votecount INT)
BEGIN

	CREATE TABLE votecount(
		precinct varchar(255),
		totalvotes INT,
		timestamp DATETIME
	);

	INSERT INTO votecount
	SELECT precinct, totalvotes, timestamp
	FROM votes
	WHERE totalvotes >= input_votecount
	ORDER BY totalvotes, timestamp ASC;

	SELECT precinct
	FROM votecount x, (SELECT min(timestamp) as mintime FROM votecount)y
	WHERE timestamp = mintime
	ORDER BY totalvotes DESC
	LIMIT 1;

	DROP TABLE votecount;

END //

DELIMITER ;
CALL earliestprecinct(350);
```

Precinct
HARRIS EAST

QUESTION 2 – The Candidates

a) precinctswon(candidate)

```
DELIMITER //

CREATE PROCEDURE precinctswon(IN input_candidate VARCHAR(255))
BEGIN

	DECLARE candidate VARCHAR(255);

	SET candidate = input_candidate;

	IF candidate = 'biden' THEN
		SELECT precinct, biden-trump AS votedifference, biden
		FROM votes x, (SELECT max(timestamp) as maxtime FROM votes) y
		WHERE biden > trump AND timestamp = maxtime
		ORDER BY biden-trump;
	END IF;

	IF candidate = 'trump' THEN
		SELECT precinct, trump-biden AS votedifference, trump
		FROM votes x, (SELECT max(timestamp) as maxtime FROM votes) y
		WHERE trump > biden AND timestamp = maxtime
		ORDER BY trump-biden;
	END IF;

END //

DELIMITER ;
CALL precinctswon('Trump');
```

Results:

https://rutgers.box.com/s/bhiof7q9raxymdy1hoichw9mo0xadv3f

b) precinctswoncount(candidate)

```
DELIMITER //

CREATE PROCEDURE precinctswoncount(IN input_candidate VARCHAR(255))
BEGIN

	DECLARE candidate VARCHAR(255);

	SET candidate = input_candidate;

	IF candidate = 'biden' THEN
		SELECT count(biden)
		FROM votes x, (SELECT max(timestamp) as maxtime FROM votes) y
		WHERE biden > trump AND timestamp = maxtime;
	END IF;

	IF candidate = 'trump' THEN
		SELECT count(trump)
		FROM votes x, (SELECT max(timestamp) as maxtime FROM votes) y
		WHERE trump > biden AND timestamp = maxtime;
	END IF;

END //

DELIMITER ;
CALL precinctswoncount('Trump');
```

Count(trump)
691

c) precinctsfulllead(candidate)

```
DELIMITER //

CREATE PROCEDURE precinctsfulllead(IN input_candidate VARCHAR(255))
BEGIN

    DECLARE candidate VARCHAR(255);

    SET candidate = input_candidate;

     IF candidate = 'biden' THEN
        SELECT DISTINCT x.precinct
        FROM (SELECT precinct FROM votes WHERE biden > trump)x LEFT JOIN (SELECT
        DISTINCT precinct FROM votes WHERE trump >= biden)y ON x.precinct = y.precinct;
    END IF;

    IF candidate = 'trump' THEN
        SELECT DISTINCT x.precinct
        FROM (SELECT precinct FROM votes WHERE trump > biden)x LEFT JOIN (SELECT
        DISTINCT precinct FROM votes WHERE biden >= trump)y ON x.precinct = y.precinct;
    END IF;

END //

DELIMITER ;
CALL precinctsfulllead('trump');
```

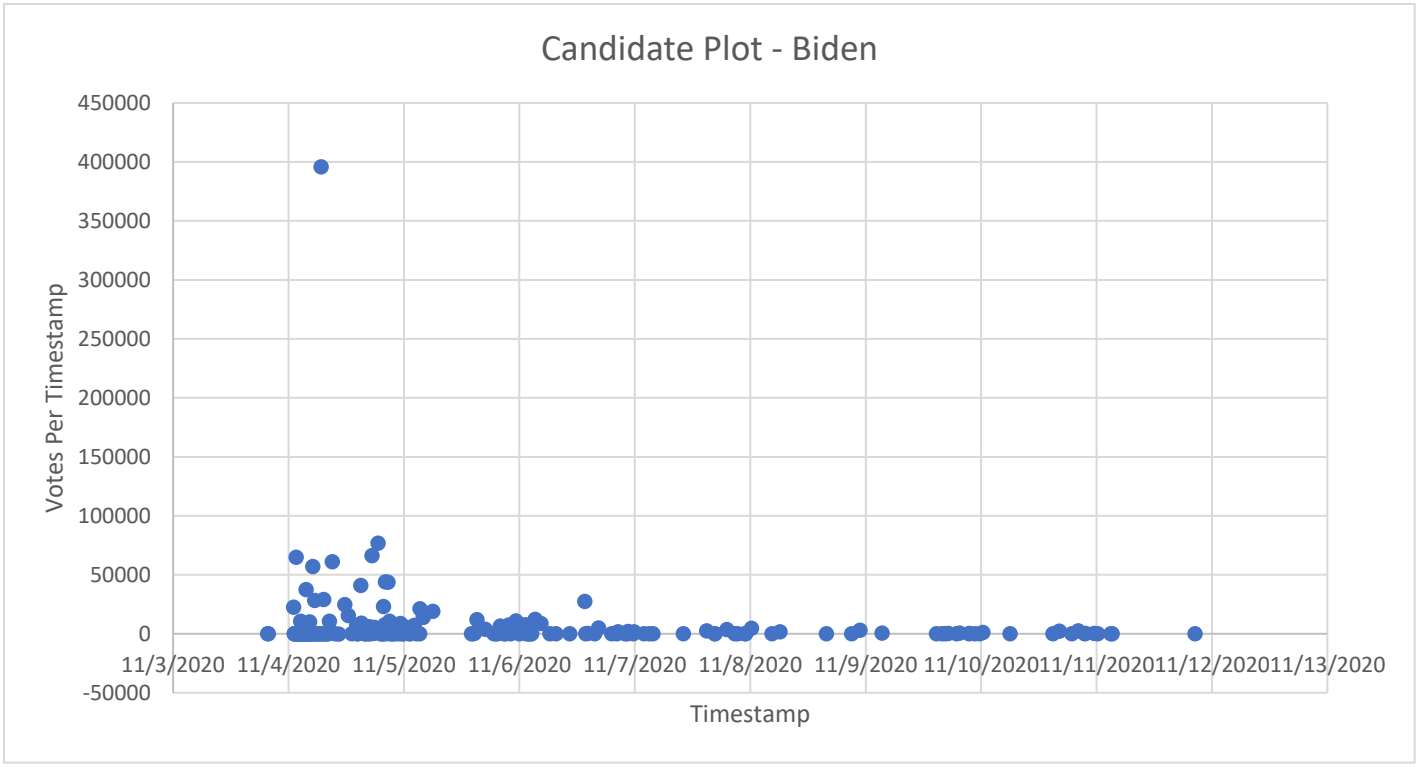
Results:

https://rutgers.box.com/s/f0rkn2bobk3hfhvegcomxyr2fdrnwxwj

d) plotcandidate(candidate)

```
DELIMITER //

CREATE PROCEDURE plotcandidate(IN input_candidate VARCHAR(255))
BEGIN

        DECLARE candidate VARCHAR(255);

        SET candidate = input_candidate;

        IF candidate = 'biden' THEN
                CREATE TABLE temp(
                        id INT NOT NULL AUTO_INCREMENT,
                        timestamp DATETIME,
                        votes INT,
                        PRIMARY KEY (id)
                );

                INSERT INTO temp (timestamp, votes)
                SELECT timestamp, sum(biden) as votes
                FROM votes
                GROUP BY timestamp
                ORDER BY timestamp ASC;

                SELECT x.timestamp, IFNULL(y.votes-x.votes, 0) as votes
                FROM temp x INNER JOIN temp y ON y.id = x.id + 1;

                DROP TABLE temp;
```

```
    END IF;

    IF candidate = 'trump' THEN
        CREATE TABLE temp(
            id INT NOT NULL AUTO_INCREMENT,
            timestamp DATETIME,
            votes INT,
            PRIMARY KEY (id)
        );

        INSERT INTO temp (timestamp, votes)
        SELECT timestamp, sum(biden) as votes
        FROM votes
        GROUP BY timestamp
        ORDER BY timestamp ASC;

        SELECT x.timestamp, IFNULL(y.votes-x.votes, 0) as votes
        FROM temp x INNER JOIN temp y ON y.id = x.id + 1;

        DROP TABLE temp;
    END IF;

END //

DELIMITER ;
CALL plotcandidate('biden');
```


Candidate Plot - Biden
Votes Per Timestamp
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
-50000
11/3/2020
11/4/2020
11/5/2020
11/6/2020
11/7/2020
11/8/2020
11/9/2020
11/10/202011/11/202011/12/202011/13/2020
Timestamp

e) precinctswoncategory()

precinctswontownships()

```
DELIMITER //

CREATE PROCEDURE precinctswontownships()
BEGIN

    SELECT IF(bidenvotes > trumpvotes, "Biden won", "Trump won") as 'Won Township Precincts',
    IF(bidenvotes > trumpvotes, bidenvotes-trumpvotes, trumpvotes-bidenvotes) as 'Vote
    Difference', bidenvotes as 'Votes for Biden', trumpvotes as 'Votes for Trump'
    FROM

        (Select sum(Biden) as bidenvotes
        From votes
        Where precinct like '%Township%'
        Group by timestamp
        Order by timestamp desc
        limit 1)x,

        (Select sum(Trump) as trumpvotes
        From votes
        Where precinct like '%Township%'
        Group by timestamp
        Order by timestamp desc
        limit 1)y;

END //

DELIMITER ;
CALL precinctswontownships();
```

Won Township Precincts	Vote Difference	Votes for Biden	Votes for Trump
Trump Won	18821	28655	47476

precinctswonboroughs()

```
DELIMITER //

CREATE PROCEDURE precinctswonboroughs()
BEGIN

    SELECT IF(bidenvotes > trumpvotes, "Biden won", "Trump won") as 'Won Borough Precincts',
    IF(bidenvotes > trumpvotes, bidenvotes-trumpvotes, trumpvotes-bidenvotes) as 'Vote
    Difference', bidenvotes as 'Votes for Biden', trumpvotes as 'Votes for Trump'
    FROM

        (Select sum(Biden) as bidenvotes
        From votes
        Where precinct like '%Borough%'
        Group by timestamp
        Order by timestamp desc
        limit 1)x,

        (Select sum(Trump) as trumpvotes
        From votes
        Where precinct like '%Borough%'
        Group by timestamp
        Order by timestamp desc
        limit 1)y;

END //

DELIMITER ;
CALL precinctswonboroughs();
```

Won Borough Precincts	Vote Difference	Votes for Biden	Votes for Trump
Trump Won	7597	15977	23574

precinctswonwards()

```
DELIMITER //

CREATE PROCEDURE precinctswonwards()
BEGIN

    SELECT IF(bidenvotes > trumpvotes, "Biden won", "Trump won") as 'Won Ward Precincts',
    IF(bidenvotes > trumpvotes, bidenvotes-trumpvotes, trumpvotes-bidenvotes) as 'Vote
    Difference', bidenvotes as 'Votes for Biden', trumpvotes as 'Votes for Trump'
    FROM

        (Select sum(Biden) as bidenvotes
        From votes
        Where precinct like '%Ward%'
        Group by timestamp
        Order by timestamp desc
        limit 1)x,

        (Select sum(Trump) as trumpvotes
        From votes
        Where precinct like '%Ward%'
        Group by timestamp
        Order by timestamp desc
        limit 1)y;

END //

DELIMITER ;
CALL precinctswonwards();
```

Won Ward Precincts	Vote Difference	Votes for Biden	Votes for Trump
Biden won	33211	73348	40137

QUESTION 3 – The Timestamp

a) totalvotes(timestamp, category)

```
DELIMITER //
CREATE PROCEDURE totalvotes(IN input_timestamp DATETIME, IN input_category varchar(255))
BEGIN
	DECLARE category varchar(255);
	SET category = input_category;
	IF category = 'all' THEN
		SELECT precinct, totalvotes
		FROM votes
		WHERE timestamp = input_timestamp
		ORDER BY totalvotes DESC;
	END IF;

	IF category = 'biden' THEN
		SELECT precinct, biden
		FROM votes
		WHERE timestamp = input_timestamp
		ORDER BY biden DESC;
	END IF;

	IF category = 'trump' THEN
		SELECT precinct, trump
		FROM votes
		WHERE timestamp = input_timestamp
		ORDER BY trump DESC;
	END IF;
END //

DELIMITER ;
CALL totalvotes('2020-11-04 03:58:36', 'biden');
```

Results:

https://rutgers.box.com/s/drc99lr62ckoyhu3djsgwyxuee5bvftf

b) gaindelta(timestamp)

```
DELIMITER //

CREATE PROCEDURE gaindelta(IN input_timestamp DATETIME)

BEGIN

    SELECT DELTA, GAIN, GAIN/DELTA

    FROM (SELECT TIMEDIFF(input_timestamp, y.timestamp) as DELTA, (x.sumvotes - y.sumvotes)
    as GAIN

    FROM (SELECT sum(totalvotes) as sumvotes FROM votes WHERE timestamp =
    input_timestamp)x, (SELECT timestamp, sum(totalvotes) as sumvotes FROM votes WHERE
    input_timestamp > timestamp GROUP BY timestamp ORDER BY timestamp DESC LIMIT 1)y)a;


END //

DELIMITER ;

CALL gaindelta('2020-11-04 03:58:36');
```

DELTA	GAIN	GAIN/DELTA
00:06:41	8927	13.9267

c) ranktimestamp()

```
DELIMITER //

CREATE PROCEDURE ranktimestamp()
BEGIN
        CREATE TABLE ranking (
                id INT NOT NULL AUTO_INCREMENT,
                timestamp DATETIME,
                totalvotes INT,
                PRIMARY KEY (id)
        );

        INSERT INTO ranking(timestamp, totalvotes)
        SELECT timestamp, sum(totalvotes)
        FROM votes
        GROUP BY timestamp
        ORDER BY timestamp ASC;

        SELECT timestamp, RANK() over (ORDER BY GAIN/DELTA DESC)ranks
        FROM (SELECT y.timestamp, TIMEDIFF(y.timestamp, x.timestamp) as DELTA,
        IFNULL(sum(y.totalvotes) - sum(x.totalvotes), 0) as GAIN
        FROM ranking x INNER JOIN ranking y ON y.id = x.id + 1
        GROUP BY timestamp)a;

   DROP TABLE ranking;

END //

DELIMITER ;
CALL ranktimestamp();
```

Results: https://rutgers.box.com/s/f79qx8ed0std3tcg34s6j8j94pvno4md

d) votesperday(day)

```
DELIMITER //

CREATE PROCEDURE votesperday(IN input_day INT)

BEGIN

    SELECT x.bidenvotes - y.bidenvotes as bidenvotes, x.trumpvotes-y.trumpvotes as trumpvotes,
    x.allvotes - y.allvotes as allvotes

    FROM(SELECT sum(biden) as bidenvotes, sum(trump) as trumpvotes, sum(totalvotes) as allvotes

    FROM votes a, (SELECT max(timestamp) as maxtime FROM votes WHERE day(timestamp) =
    input_day)b

    WHERE a.timestamp = b.maxtime)x,

    (SELECT sum(biden) as bidenvotes, sum(trump) as trumpvotes, sum(totalvotes) as allvotes

    FROM votes a, (SELECT max(timestamp) as maxtime FROM votes WHERE day(timestamp) =
    input_day-1)b

    WHERE a.timestamp = b.maxtime)y;

END //

DELIMITER ;

CALL votesperday(5);
```

Bidenvotes	Trumpvotes	Allvotes
111237	26239	139328

QUESTION 4 – Suspicious or Interesting Data

```
SELECT precinct

FROM (SELECT precinct, totalvotes
FROM votes x, (SELECT max(timestamp) as maxtime FROM votes)x
WHERE timestamp = maxtime)b

WHERE b.precinct not in (SELECT precinct FROM votes x, (SELECT min(timestamp)
as mintime FROM votes)x WHERE timestamp = mintime)
```

Results:
https://rutgers.box.com/s/b613b0o27yxjath1rm4oyjyqim39vghd

What I found was that many precincts did not have data at the minimum timestamp. This is suspicious because the data is missing while other precincts have values at the minimum timestamp even if the value is zero.

PART 3

QUESTION A

Select true

From votes

WHERE NOT EXISTS (SELECT precinct FROM votes WHERE (trump + biden) > totalvotes)

Union

Select false

From votes

WHERE EXISTS (SELECT precinct FROM votes WHERE (trump + biden) > totalvotes)

QUESTION B

Select true

From votes

WHERE NOT EXISTS (SELECT timestamp FROM votes WHERE DAY(timestamp) > 11 or DAY(timestamp) < 3)

Union

Select false

From votes

WHERE EXISTS (SELECT timestamp FROM votes WHERE DAY(timestamp) > 11 or DAY(timestamp) < 3)

QUESTION C

WHEN timestamp is less than 2020-11-05 00:00:00

Select true

From votes

WHERE NOT EXISTS (SELECT timestamp FROM (SELECT biden, trump, totalvotes, timestamp FROM votes WHERE timestamp = '2020-11-05 00:00:00' AND precinct = input_precinct)y

WHERE y.biden > input_biden OR y.trump > input_trump OR y.totalvotes > input_totalvotes)

Union

Select false

From votes

WHERE EXISTS (SELECT timestamp FROM (SELECT biden, trump, totalvotes, timestamp FROM votes WHERE timestamp = '2020-11-05 00:00:00' AND precinct = input_precinct)y

WHERE y.biden > input_biden OR y.trump > input_trump OR y.totalvotes > input_totalvotes)

WHEN timestamp is 2020-11-05 00:00:00

Select true

From votes

WHERE NOT EXISTS (SELECT timestamp FROM (SELECT biden, trump, totalvotes, timestamp FROM votes WHERE timestamp > '2020-11-05 00:00:00' AND precinct = input_precinct GROUP BY timestamp)y

WHERE y.biden < input_biden OR y.trump < input_trump OR y.totalvotes < input_totalvotes)

Union

Select false as result

From votes

WHERE EXISTS (SELECT timestamp FROM (SELECT biden, trump, totalvotes, timestamp FROM votes WHERE timestamp > '2020-11-05 00:00:00' AND precinct = input_precinct GROUP BY timestamp)y

WHERE y.biden < input_biden OR y.trump < input_trump OR y.totalvotes < input_totalvotes)

PART 4

ALTER TABLE place ADD PRIMARY KEY (precinct)

ALTER TABLE votes ADD PRIMARY KEY (precinct, timestamp)

ALTER TABLE votes

ADD FOREIGN KEY (precinct) REFERENCES place(precinct)

CONSTRAINTS FROM PART 3 AS PROCEDURES

A

```
DELIMITER //


CREATE PROCEDURE votesconstraint(IN input_biden INT, IN input_trump INT, IN input_totalvotes INT,
OUT output_boolean BOOLEAN)

BEGIN

    DECLARE output BOOLEAN;


    SELECT result

    INTO output

    FROM

    (Select true as result

            From votes

            WHERE NOT EXISTS (SELECT precinct FROM votes WHERE (input_trump + input_biden) >
            input_totalvotes)


            Union


            Select false as result

            From votes

            WHERE EXISTS (SELECT precinct FROM votes WHERE (input_trump + input_biden) >
            input_totalvotes))x;


    SET output_boolean = output;


END//

DELIMITER ;
```

B

```
DELIMITER //

CREATE PROCEDURE dayconstraint(IN input_timestamp DATETIME, OUT output_boolean BOOLEAN)
BEGIN
    DECLARE output BOOLEAN;

    SELECT result
    INTO output
    FROM(
    Select true as result
            From votes
            WHERE NOT EXISTS (SELECT timestamp FROM votes WHERE DAY(input_timestamp) > 11
            or DAY(input_timestamp) < 3)

            Union

            Select false as result
            From votes
            WHERE EXISTS (SELECT timestamp FROM votes WHERE DAY(input_timestamp) > 11 or
            DAY(input_timestamp) < 3))x;

    SET output_boolean = output;

END//
DELIMITER ;
```

C

```
DELIMITER //


CREATE PROCEDURE timestampconstraint(IN input_timestamp DATETIME, IN input_biden INT, IN
input_trump INT, IN input_totalvotes INT, IN input_precinct VARCHAR(255), OUT output_boolean
BOOLEAN)

BEGIN

    DECLARE output BOOLEAN;


    IF input_timestamp > '2020-11-05 00:00:00' THEN

                    SELECT result

                    INTO output

                    FROM(

                    Select true as result

                    From votes

                    WHERE NOT EXISTS (SELECT timestamp FROM (SELECT biden, trump, totalvotes,
                    timestamp FROM votes WHERE timestamp = '2020-11-05 00:00:00' AND
                    precinct = input_precinct)y

                    WHERE y.biden > input_biden OR y.trump > input_trump OR y.totalvotes >
                    input_totalvotes)

                    Union


                    Select false as result

                    From votes

                    WHERE EXISTS (SELECT timestamp FROM (SELECT biden, trump, totalvotes,
                    timestamp FROM votes WHERE timestamp = '2020-11-05 00:00:00' AND
                    precinct = input_precinct)y

                    WHERE y.biden > input_biden OR y.trump > input_trump OR y.totalvotes >
                    input_totalvotes))x;


    ELSEIF input_timestamp = '2020-11-05 00:00:00' THEN

                    SELECT result

                    INTO output
```

```
                    FROM(
                    Select true as result
                    From votes
                    WHERE NOT EXISTS (SELECT timestamp FROM (SELECT biden, trump, totalvotes,
                    timestamp FROM votes WHERE timestamp > '2020-11-05 00:00:00' AND
                    precinct = input_precinct GROUP BY timestamp)y
                    WHERE y.biden < input_biden OR y.trump < input_trump OR y.totalvotes <
                    input_totalvotes)
                    Union

                    Select false as result
                    From votes
                    WHERE EXISTS (SELECT timestamp FROM (SELECT biden, trump, totalvotes,
                    timestamp FROM votes WHERE timestamp > '2020-11-05 00:00:00' AND
                    precinct = input_precinct GROUP BY timestamp)y
                    WHERE y.biden < input_biden OR y.trump < input_trump OR y.totalvotes <
                    input_totalvotes))x;

ELSE
                    SELECT true
                    INTO output;
END IF;

    SET output_boolean = output;

END//
DELIMITER ;
```

INSERT PROCEDURE

```
DELIMITER //

CREATE PROCEDURE insertproc(IN input_precinct VARCHAR(255), IN input_geo
VARCHAR(255), IN input_locality VARCHAR(255), IN input_state VARCHAR(255), IN
input_timestamp DATETIME,
IN input_totalvotes INT, IN input_biden INT, IN input_trump INT, IN input_filestamp INT,
IN input_id INT, IN input_table VARCHAR(255))
insertcommands:BEGIN

    DECLARE tablename VARCHAR(255);

    DECLARE EXIT HANDLER FOR 1452
    BEGIN
        SELECT CONCAT('Insertion rejected due to foreign key violation') as
        message;
    END;

    DECLARE EXIT HANDLER FOR 1062
    BEGIN
        SELECT CONCAT('Insertion rejected due to primary key violation') as
        message;
    END;

    SET tablename = input_table;

    IF tablename = 'place' THEN
        INSERT INTO place(precinct, geo, locality, state)
        VALUES(input_precinct, input_geo, input_locality, input_state);
    END IF;
```

```
IF tablename = 'votes' THEN
                CALL votesconstraint(input_biden, input_trump, input_totalvotes, @output1);
                IF @output1 = false THEN
                        SELECT CONCAT('Insertion rejected due to value violation') as
                        message;
                        LEAVE insertcommands;
                END IF;

                CALL dayconstraint(input_timestamp, @output2);
                        IF @output2 = false THEN
                        SELECT CONCAT('Insertion rejected due to day violation') as message;
                        LEAVE insertcommands;
                END IF;

                CALL timestampconstraint(input_timestamp, input_biden, input_trump,
                input_totalvotes, input_precinct, @output3);
                IF @output3 = false THEN
                        SELECT CONCAT('Insertion rejected due to timestamp violation') as
                        message;
                        LEAVE insertcommands;
                END IF;

                INSERT INTO votes(precinct, id, timestamp, totalvotes, biden, trump, filestamp)
                VALUES(input_precinct, input_id, input_timestamp, input_totalvotes,
                input_biden, input_trump, input_filestamp);

        END IF;

        SELECT 'insertion sucess' as 'modification';
END //
DELIMITER ;
```

UPDATE PROCEDURE

```
DELIMITER //

CREATE PROCEDURE updateproc(IN input_precinct VARCHAR(255), IN input_geo VARCHAR(255), IN input_locality VARCHAR(255), IN input_state VARCHAR(255), IN input_timestamp DATETIME,

IN input_totalvotes INT, IN input_biden INT, IN input_trump INT, IN input_filestamp INT, IN input_id INT, IN input_table VARCHAR(255), IN input_column VARCHAR(255))

updatecommands:BEGIN

    DECLARE tablename, columnname VARCHAR(255);

    DECLARE EXIT HANDLER FOR 1451
    BEGIN
        SELECT CONCAT('Update rejected due to foreign key violation') as message;
    END;

    SET tablename = input_table;
    SET columnname = input_column;

    IF tablename = 'place' THEN
        IF columnname = 'precinct' THEN
            UPDATE place
            SET precinct = input_precinct
            WHERE precinct = ‘005 ATGLEN’;
        END IF;
        IF columnname = 'geo' THEN
            UPDATE place
            SET geo = input_geo
            WHERE geo = '42029-ATGLEN';
        END IF;
        IF columnname = 'locality' THEN
            UPDATE place
            SET locality = input_locality
```

```
WHERE locality = 'Chester';
                    END IF;
                IF columnname = 'state' THEN
                    UPDATE place
                    SET state = input_state
                    WHERE state = 'PA';
                END IF;
            END IF;

            IF tablename = 'votes' THEN
                CALL votesconstraint(input_biden, input_trump, input_totalvotes, @output1);
                IF @output1 = false THEN
                    SELECT CONCAT('Update rejected due to value violation') as message;
                    LEAVE updatecommands;
                END IF;

                CALL dayconstraint(input_timestamp, @output2);
                IF @output2 = false THEN
                    SELECT CONCAT('Update rejected due to day violation') as message;
                    LEAVE updatecommands;
                END IF;

                CALL timestampconstraint(input_timestamp, input_biden, input_trump, input_totalvotes,
                input_precinct, @output3);
                IF @output3 = false THEN
                    SELECT CONCAT('Update rejected due to timestamp violation') as message;
                    LEAVE updatecommands;
                END IF;

                 IF columnname = 'precinct' THEN
                    UPDATE votes
```

```
SET precinct = input_precinct
        WHERE precinct = '005 ATGLEN';
END IF;
IF columnname = 'id' THEN
        UPDATE votes
        SET id = input_id
        WHERE id = '212';
END IF;
IF columnname = 'timestamp' THEN
        UPDATE votes
        SET timestamp = input_timestamp
        WHERE timestamp = '2020-11-05 00:00:00';
END IF;
IF columnname = 'totalvotes' THEN
        UPDATE votes
        SET totalvotes = input_totalvotes
        WHERE precinct = input_precinct AND timestamp = input_timestamp;
END IF;
IF columnname = 'biden' THEN
        UPDATE votes
        SET biden = input_biden
        WHERE precinct = input_precinct AND timestamp = input_timestamp;
END IF;
IF columnname = 'trump' THEN
        UPDATE votes
        SET trump = input_trump
        WHERE precinct = input_precinct AND timestamp = input_timestamp;
END IF;
IF columnname = 'filestamp' THEN
        UPDATE votes
        SET filestamp = input_filestamp
```

```
                    WHERE precinct = input_precinct AND timestamp = input_timestamp;
                END IF;

        END IF;

        SELECT 'update sucess' as 'modification';

END //
DELIMITER ;
```

DELETE PROCEDURE

```
DELIMITER //


CREATE PROCEDURE deleteproc(IN input_precinct VARCHAR(255), IN input_geo VARCHAR(255), IN
input_locality VARCHAR(255), IN input_state VARCHAR(255), IN input_timestamp DATETIME,

IN input_totalvotes INT, IN input_biden INT, IN input_trump INT, IN input_filestamp INT, IN input_id INT,
IN input_table VARCHAR(255), IN input_column VARCHAR(255))

updatecommands:BEGIN


  DECLARE tablename, columnname VARCHAR(255);


  DECLARE EXIT HANDLER FOR 1451

  BEGIN

      SELECT CONCAT('Delete rejected due to foreign key violation') as message;

  END;


  SET tablename = input_table;

  SET columnname = input_column;


  IF tablename = 'place' THEN

            IF columnname = 'precinct' THEN

                  DELETE FROM place

                  WHERE precinct = input_precinct;

            END IF;

            IF columnname = 'geo' THEN

                  DELETE FROM geo

                  WHERE geo = input_geo;

            END IF;

            IF columnname = 'locality' THEN
```

```
                DELETE FROM place
                WHERE locality = input_locality;
        END IF;
        IF columnname = 'state' THEN
                DELETE FROM place
                WHERE state = input_state;
        END IF;
    END IF;

IF tablename = 'votes' THEN
        IF columnname = 'precinct' THEN
                DELETE FROM votes
                WHERE precinct = input_precinct;
        END IF;
        IF columnname = 'id' THEN
                DELETE FROM votes
                WHERE id = input_id;
        END IF;
        IF columnname = 'timestamp' THEN
                DELETE FROM votes
                WHERE timestamp = input_timestamp;
        END IF;
        IF columnname = 'totalvotes' THEN
                DELETE FROM votes
                WHERE totalvotes = input_totalvotes;
        END IF;
        IF columnname = 'biden' THEN
                DELETE FROM votes
                WHERE biden = input_biden;
```

```
            END IF;
            IF columnname = 'trump' THEN
                DELETE FROM votes
                WHERE trump = input_trump;
            END IF;
            IF columnname = 'filestamp' THEN
                DELETE FROM votes
                WHERE filestamp = input_filestamp;
            END IF;

        END IF;

        SELECT 'delete sucess' as 'modification';

END //
DELIMITER ;
```

4.1 – Triggers and Update driven Stored Procedures

TABLE: place

```
CREATE TABLE updatedtuplesplace(
        state VARCHAR(45) NULL,
        locality VARCHAR(45) NULL,
        precinct VARCHAR(45) NULL,
        geo VARCHAR(45) NULL
);

CREATE TABLE insertedtuplesplace(
        state VARCHAR(45) NULL,
        locality VARCHAR(45) NULL,
        precinct VARCHAR(45) NULL,
        geo VARCHAR(45) NULL
);

CREATE TABLE deletedtuplesplace(
        state VARCHAR(45) NULL,
        locality VARCHAR(45) NULL,
        precinct VARCHAR(45) NULL,
        geo VARCHAR(45) NULL
);
```

```
DELIMITER //

CREATE TRIGGER updatetrigplace
BEFORE UPDATE ON place FOR EACH ROW
BEGIN
        INSERT INTO updatedtuplesplace (precinct, geo, locality, state)
        VALUES (OLD.precinct, OLD.geo, OLD.locality, OLD.state);

END //
DELIMITER ;
```

```
DELIMITER //

CREATE TRIGGER inserttrigplace
BEFORE INSERT ON place FOR EACH ROW
BEGIN
        INSERT INTO insertedtuplesplace (precinct, geo, locality, state)
        VALUES (NEW.precinct, NEW.geo, NEW.locality, NEW.state);

END //
DELIMITER ;
```

```
DELIMITER //

CREATE TRIGGER deletetrigplace
BEFORE DELETE ON place FOR EACH ROW
BEGIN
        INSERT INTO deletedtuplesplace (precinct, geo, locality, state)
        VALUES (OLD.precinct, OLD.geo, OLD.locality, OLD.state);
END //
DELIMITER ;
```

TABLE: votes

```
CREATE TABLE updatedtuplesvotes(
        precinct VARCHAR(45) NULL,
        id INT NULL,
        timestamp DATETIME NULL,
        totalvotes INT NULL,
        biden INT NULL,
        trump INT NULL,
        filestamp VARCHAR(45) NULL
);

CREATE TABLE insertedtuplesvotes(
        precinct VARCHAR(45) NULL,
        id INT NULL,
        timestamp DATETIME NULL,
        totalvotes INT NULL,
        biden INT NULL,
        trump INT NULL,
        filestamp VARCHAR(45) NULL
);

CREATE TABLE deletedtuplesvotes(
        precinct VARCHAR(45) NULL,
        id INT NULL,
        timestamp DATETIME NULL,
        totalvotes INT NULL,
        biden INT NULL,
        trump INT NULL,
        filestamp VARCHAR(45) NULL
);
```

```
DELIMITER //

CREATE TRIGGER updatetrigvotes

BEFORE UPDATE ON votes FOR EACH ROW

BEGIN

    INSERT INTO updatedtuplesvotes (precinct, id, timestamp, totalvotes, biden, trump, filestamp)

    VALUES (OLD.precinct, OLD.id, OLD.timestamp, OLD.totalvotes, OLD.biden, OLD.trump,
    OLD.filestamp);


END //

DELIMITER ;
```

```
DELIMITER //

CREATE TRIGGER inserttrigvotes

BEFORE INSERT ON votes FOR EACH ROW

BEGIN

    INSERT INTO insertedtuplesvotes (precinct, id, timestamp, totalvotes, biden, trump, filestamp)

    VALUES (NEW.precinct, NEW.id, NEW.timestamp, NEW.totalvotes, NEW.biden, NEW.trump,
    NEW.filestamp);


END //

DELIMITER ;
```

```
DELIMITER //

CREATE TRIGGER deletetrigvotes

BEFORE DELETE ON votes FOR EACH ROW

BEGIN

    INSERT INTO deletedtuplesvotes (precinct, id, timestamp, totalvotes, biden, trump, filestamp)

    VALUES (OLD.precinct, OLD.id, OLD.timestamp, OLD.totalvotes, OLD.biden, OLD.trump,
    OLD.filestamp);


END //

DELIMITER ;
```

4.2 Stored Procedures

```
DECLARE precinctvar, candidate VARCHAR(255);
    DECLARE timestampvar DATETIME;
    DECLARE boolvar1, boolvar2 BOOLEAN;
    DECLARE temp INT;

    SET precinctvar = input_precinct;
    SET candidate = input_candidate;
    SET timestampvar = input_timestamp;

    SELECT result
    INTO boolvar1
    FROM(
    SELECT true as result
    FROM votes
    WHERE input_precinct IN (SELECT precinct FROM votes)

    UNION

    SELECT false as result
    FROM votes
    WHERE input_precinct NOT IN (SELECT precinct FROM votes))x;

    IF boolvar1 = false THEN
        SELECT CONCAT('Unknown Precinct') as message;
        LEAVE movevotescommand;
    END iF;

    SELECT result
    INTO boolvar2
```

```
FROM(
SELECT true as result
FROM votes
WHERE input_timestamp IN (SELECT timestamp FROM votes)

UNION

SELECT false as result
FROM votes
WHERE input_timestamp NOT IN (SELECT timestamp FROM votes))x;

IF boolvar2 = false THEN
	SELECT CONCAT('Unknown Timestamp') as message;
	LEAVE movevotescommand;
END iF;

IF NOT input_candidate = 'biden' AND NOT input_candidate = 'trump' THEN
	SELECT CONCAT('Wrong Candidate') as message;
	LEAVE movevotescommand;
END iF;

IF input_candidate = 'biden' THEN
	SELECT biden
	INTO temp
	FROM votes
	WHERE precinct = input_precinct AND timestamp = input_timestamp;
ELSE
	SELECT trump
	INTO temp
```

```
        FROM votes
        WHERE precinct = input_precinct AND timestamp = input_timestamp;
END IF;

IF number_of_moved_votes > temp THEN
        SELECT CONCAT('Not enough votes') as message;
        LEAVE movevotescommand;
END iF;




IF candidate = 'biden' THEN
        UPDATE votes
        SET trump = trump + number_of_moved_votes, biden = biden - number_of_moved_votes
        WHERE precinct = input_precinct AND timestamp >= input_timestamp;
END IF;

IF candidate = 'trump' THEN
        UPDATE votes
        SET trump = trump - number_of_moved_votes, biden = biden + number_of_moved_votes
        WHERE precinct = input_precinct AND timestamp >= input_timestamp;
END IF;
```

VIDEO RECORDING

https://rutgers.box.com/s/ns5e55xc1ik2yxqxrsr41dcykxkr7wz2