

# Election Project

## PART 1

### API1

```
DELIMITER $$
drop PROCEDURE IF EXISTS API1;
CREATE PROCEDURE API1(IN C VARCHAR(50), IN T datetime, IN P VARCHAR(50), OUT votes INT)
BEGIN
    Declare min_time datetime;
    select min(Timestamp) into min_time from Penna;

    IF not exists (select * from Penna where precinct=P) THEN
        select "wrong precinct name" ;
    ELSEIF not (C="Biden" or C="Trump") THEN
        select "wrong candidate";
    ELSEIF T< min_time THEN
        select 0 into votes;
    ELSE
        IF C="Biden" THEN select max(Biden) INTO votes from Penna where precinct=P and timestamp<=T;
        ELSEIF C="Trump" THEN select max(Trump) INTO votes from Penna where precinct=P and timestamp<=T;
        END IF;
    END IF;

END;

$$
DELIMITER ;
```

test:

```
#precint no exists
call API1("Biden", "2020-11-02 03:58:36","N", @n);
select @n;

#candidate no exists
call API1("B", "2020-11-02 03:58:36","Barr Township Voting Precinct", @n);
select @n;

#input time<min timestamp
call API1("Biden", "2020-11-02 03:58:36","Barr Township Voting Precinct", @n);
select @n;

#input time>max timestamp
call API1("Biden", "2020-11-13 02:40:36","Barr Township Voting Precinct", @n);
select @n;

#no exact timestamp=input time
call API1("Biden", "2020-11-04 02:40:36","Barr Township Voting Precinct", @n);
select @n;
```

### API2

```
DELIMITER $$
drop PROCEDURE IF EXISTS API2;
CREATE PROCEDURE API2(IN date varchar(50), OUT candidate VARCHAR(50), OUT votes int)
BEGIN
    IF not exists (select * from Penna where locate(date, Timestamp)>0 )THEN
        select "invalid date";
    ELSE
        select if(sum(Biden)>sum(Trump), "Biden", "Trump"), if(sum(Biden)>sum(Trump), sum(Biden), sum(Trump)) INTO candidate, votes
        from Penna
        where Timestamp =
            (select max(Timestamp)
            from Penna
            where locate(date, Timestamp)>0);
    END IF;

END;

$$
DELIMITER ;
```

test:

```
call API2("2020-11-06", @a, @b);
select @a, @b;

#date not in Penna
call API2("2020-10-06", @a, @b);
select @a, @b;

call API2('2020-11-10', @a, @b);
select @a, @b;
```

## API3

```
DELIMITER $$
drop PROCEDURE IF EXISTS API3;
CREATE PROCEDURE API3(IN candidate VARCHAR(50))
BEGIN

    IF not (candidate="Biden" or candidate="Trump") THEN
        select "wrong candidate";
    ELSE
        select precinct, totalvotes
        from Penna
        where timestamp =
            (select max(timestamp)
            from Penna)
        and if("Biden"=candidate, Biden>Trump, Trump>Biden)
        order by totalvotes desc
        limit 10;

    END IF;

END;

$$
DELIMITER ;
```

test:

```
call API3("Biden");

#candidate not exists
call API3("B");

call API3('Trump');
```

## API4

```
DELIMITER $$
drop PROCEDURE IF EXISTS API4;
CREATE PROCEDURE API4(IN P VARCHAR(50), OUT Candidate VARCHAR(50), OUT percent Float)
BEGIN

    IF not exists (select * from Penna where precinct=P) THEN
        select "wrong precinct name" ;
    ELSE
        select if(Biden>Trump, "Biden", "Trump"),
        if(Biden>Trump, Biden/totalvotes, Trump/totalvotes) into Candidate, percent
        from Penna
        where precinct=P
        and Timestamp =
            (select max(Timestamp) from Penna);
    END IF;

END;

$$
DELIMITER ;
```

test:

```
#precinct not in Penna
call API4("B", @n, @b);

call API4("Barr Township Voting Precinct", @n, @b);

select @n, @b;
```

## API5

```
DELIMITER $$
drop PROCEDURE IF EXISTS API5;
CREATE PROCEDURE API5(IN s VARCHAR(50), OUT C VARCHAR(50), OUT votes int)
BEGIN
  IF not exists (select * from Penna where locate(s, precinct)>0) THEN
    select "wrong precinct keyword";
  ELSE
    select if(sum(Biden)>sum(Trump),"Biden", "Trump"),
    if(sum(Biden)>sum(Trump),sum(Biden), sum(Trump)) INTO C, votes
    from Penna
    where Timestamp=(select max(Timestamp) from Penna)
    and precinct in (
    select distinct precinct
    from Penna
    where locate(s, precinct)>0);

  END IF;

END;

$$
DELIMITER ;
```

test:

```
call API5("SOUTH", @a, @b);

call API5('CALN', @a, @b);

select @a, @b;

call API5('sql', @a, @b);
```

## PART2

### 2.1

```
DELIMITER $$
drop PROCEDURE IF EXISTS newPenna;
CREATE PROCEDURE newPenna()
BEGIN

  Declare var_count INT;
  Declare var_end_count int;
  Declare P varchar(50);
  Declare cur Cursor For (select distinct precinct from Penna);

  drop table if exists newPenna;
  create table newPenna(
  precinct varchar(50),
  Timestamp datetime,
  newvotes int,
  new_Biden int,
  new_Trump int
  );

  Set var_count = 0;
  Select count(distinct precinct) into var_end_count From Penna;
```

```

Open cur;
While var_count<var_end_count DO
  fetch cur into P;
  Set @t_votes =0;
  Set @b=0;
  Set @t=0;
  insert into newPenna (precinct, Timestamp, newvotes, new_Biden, new_Trump)
  select precinct, Timestamp, 0-@t_votes+(@t_votes:= totalvotes), 0-@b+(@b:=Biden), 0-@t+(@t:=Trump)
  from Penna
  where precinct=P
  order by Timestamp;

  Set var_count=var_count+1;
END WHILE;
CLOSE cur;

END;

$$
DELIMITER ;

```

test:

```

Call newPenna();
Select * from newPenna where newvotes<0;

```

## 2.2

```

DELIMITER $$
drop PROCEDURE IF EXISTS Switch;
CREATE PROCEDURE Switch()
BEGIN

  Declare final_C varchar(50);
  Declare before_C varchar(50);
  Declare MT datetime;

  Declare var_count int;
  Declare var_end_count int;
  Declare P varchar(50);
  Declare cur Cursor For select distinct precinct from Penna;

  drop table if exists switch_list;
  create table switch_list(
    precinct varchar(50),
    Timestamp datetime,
    before_winner varchar(50),
    after_winner varchar(50)
  );

  select count(distinct precinct) into var_end_count from Penna;
  Set var_count=0;

  Open cur;
  While var_count<var_end_count DO
    fetch cur into P;

    select max(Timestamp) into MT
    from Penna where precinct=P;

    select if(max(Biden)>max(Trump),"Biden", "Trump" ) into final_C
    from Penna where precinct=P;

    select if((Biden)>(Trump),"Biden", "Trump" ) into before_C
    from Penna
    where precinct=P and Timestamp=
    (select max(Timestamp) from Penna where Timestamp<=date_sub(MT, interval 8 day));

    IF strcmp(final_C, before_C)=-1 THEN

    insert into switch_list

```

```

select P, Timestamp, before_C, final_C
from Penna
where if( strcmp(before_C, "Biden")=0, Biden>Trump, Trump>Biden ) and precinct=P
order by Timestamp desc
limit 1
;
END IF;

Set var_count=var_count+1;
End while;
Close cur;

END;

$$
DELIMITER ;

```

test:

```

Call Switch();
select * from switch_list;

```

## PART 3

### 3.1

```

DELIMITER $$
drop procedure if exists P3_a;
Create procedure P3_a(OUT result Boolean)
begin

    Declare num int;
    select count(*) into num
    from Penna
    where (Biden+Trump)>totalvotes;

    IF num>0 THEN select False into result ;
    ELSE select True into result;
    END IF;

end; $$
DELIMITER ;

call P3_a(@a);
select @a;

```

### 3.2

```

DELIMITER $$
drop procedure if exists P3_b;
Create procedure P3_b(OUT result Boolean)
begin

    Declare num int;
    select count(*) into num
    from penna
    where Timestamp<'2020-11-03 00:00:00' or Timestamp>='2020-11-12 00:00:00';

    IF num>0 THEN select False into result ;
    ELSE select True into result;
    END IF;

end; $$
DELIMITER ;

call P3_b(@n);
select @n;

```

## 3.3

```
DELIMITER $$
drop procedure if exists P3_c;
Create procedure P3_c(OUT result Boolean)
begin
    Declare num int;
    Declare var_count INT;
    Declare var_end_count int;
    Declare P varchar(50);
    Declare cur Cursor for (select distinct precinct from penna);

    drop table if exists P3_c;
    CREATE TABLE P3_c LIKE penna;
    ALTER TABLE P3_c ADD COLUMN diff INT;

    Set var_count = 0;
    Select count(distinct precinct) into var_end_count From Penna;

    Open cur;
    While var_count<var_end_count DO
        fetch cur into P;
        Set @t_votes =0;

        insert into P3_c
        select *, 0-@t_votes+(@t_votes:= totalvotes)
        from Penna
        where precinct=P
        order by Timestamp;

        Set var_count=var_count+1;
    END WHILE;
    CLOSE cur;

    select count(*) into num
    from P3_c
    where Timestamp>'2020-11-05 00:00:00' and diff<0;

    IF num>0 THEN select False into result;
    ELSE select True into result;
    END IF;

end; $$
DELIMITER ;

call P3_c(@c);
select @c;

select * from P3_c where diff<0 limit 20;
```

## PART 4

### 4.1

```
DELIMITER $$
drop procedure if exists P4_1;
Create procedure P4_1()
begin
    drop table if exists Penna_copy;
    CREATE TABLE Penna_copy LIKE penna;
    insert into Penna_copy select * from penna;

    drop table if exists Updated_Tuples;
    CREATE TABLE Updated_Tuples LIKE penna;

    drop table if exists Inserted_Tuples;
    CREATE TABLE Inserted_Tuples LIKE penna;

    drop table if exists Deleted_Tuples;
    CREATE TABLE Deleted_Tuples LIKE penna;

end; $$
DELIMITER ;
```

```

DELIMITER $$
drop trigger if exists Deletes;
Create trigger Deletes
after Delete on Penna_copy
for each row
begin
insert into Deleted_Tuples
Values(OLD.ID, OLD.Timestamp, OLD.state, OLD.locality, OLD.precinct, OLD.geo, OLD.totalvotes, OLD.Biden, OLD.Trump, OLD.filestamp);
END $$
DELIMITER ;

DELIMITER $$
drop trigger if exists Updates;
Create trigger Updates
after Update on Penna_copy
for each row
begin
insert into Updated_Tuples (ID, Timestamp, state, locality, precinct, geo, totalvotes, Biden, Trump, filestamp)
Values(OLD.ID, OLD.Timestamp, OLD.state, OLD.locality, OLD.precinct, OLD.geo, OLD.totalvotes, OLD.Biden, OLD.Trump, OLD.filestamp);

END $$
DELIMITER ;

DELIMITER $$
drop trigger if exists Inserts;
Create trigger Inserts
after Insert on Penna_copy
for each row
begin
insert into Inserted_Tuples
Values (New.ID, New.Timestamp, New.state, New.locality, New.precinct, New.geo, New.totalvotes, New.Biden, New.Trump, New.filestamp);
END $$
DELIMITER ;

```

test:

```

call P4_1();

select * from Deleted_Tuples;
delete from Penna_copy where Timestamp<"2020-11-05 00:00:00";
select * from Deleted_Tuples;

select * from Inserted_Tuples;
insert into Penna_copy select * from Penna limit 2;
select * from Inserted_Tuples;

select * from Updated_Tuples;
Update Penna_copy SET Biden=999, Trump=111 where Timestamp<"2020-11-05 03:58:36";
select * from Updated_Tuples;

```

## 4.2

```

DELIMITER $$
drop procedure if exists MoveVotes;
Create procedure MoveVotes(IN P varchar(50), IN T datetime, IN C varchar(50), IN Number_of_Moved_Votes varchar(50))
begin

declare current_vote int;

drop table if exists Penna_copy;
CREATE TABLE Penna_copy LIKE penna;
insert into Penna_copy select * from penna;

CASE
WHEN not exists (select * from Penna where precinct=P) THEN select "wrong precinct name" ;
WHEN not exists (select * from Penna where Timestamp=T) THEN select "unknown timestamp";
WHEN not (C="Biden" or C="Trump") THEN select "wrong candidate";
ELSE select if(C="Biden", Biden, Trump) into current_vote from Penna where precinct=P and Timestamp=T;
IF current_vote < Number_of_Moved_Votes THEN select "not enough votes";
ELSE
IF C="Biden" THEN Update Penna_copy SET Biden=Biden-Number_of_Moved_Votes, Trump=Trump+Number_of_Moved_Votes Where pre
ELSE Update Penna_copy SET Biden=Biden+Number_of_Moved_Votes, Trump=Trump-Number_of_Moved_Votes Where precinct=P and T
END IF;
END IF;

END CASE;

```

```
end; $$  
DELIMITER ;
```

test:

```
call MoveVotes("Red Hill", "2020-11-06 15:38:36", "Trump", 100);  
select * from Penna_copy where precinct="Red Hill" and Timestamp>="2020-11-06 15:38:36";  
select * from Penna where precinct="Red Hill" and Timestamp>="2020-11-06 15:38:36";
```