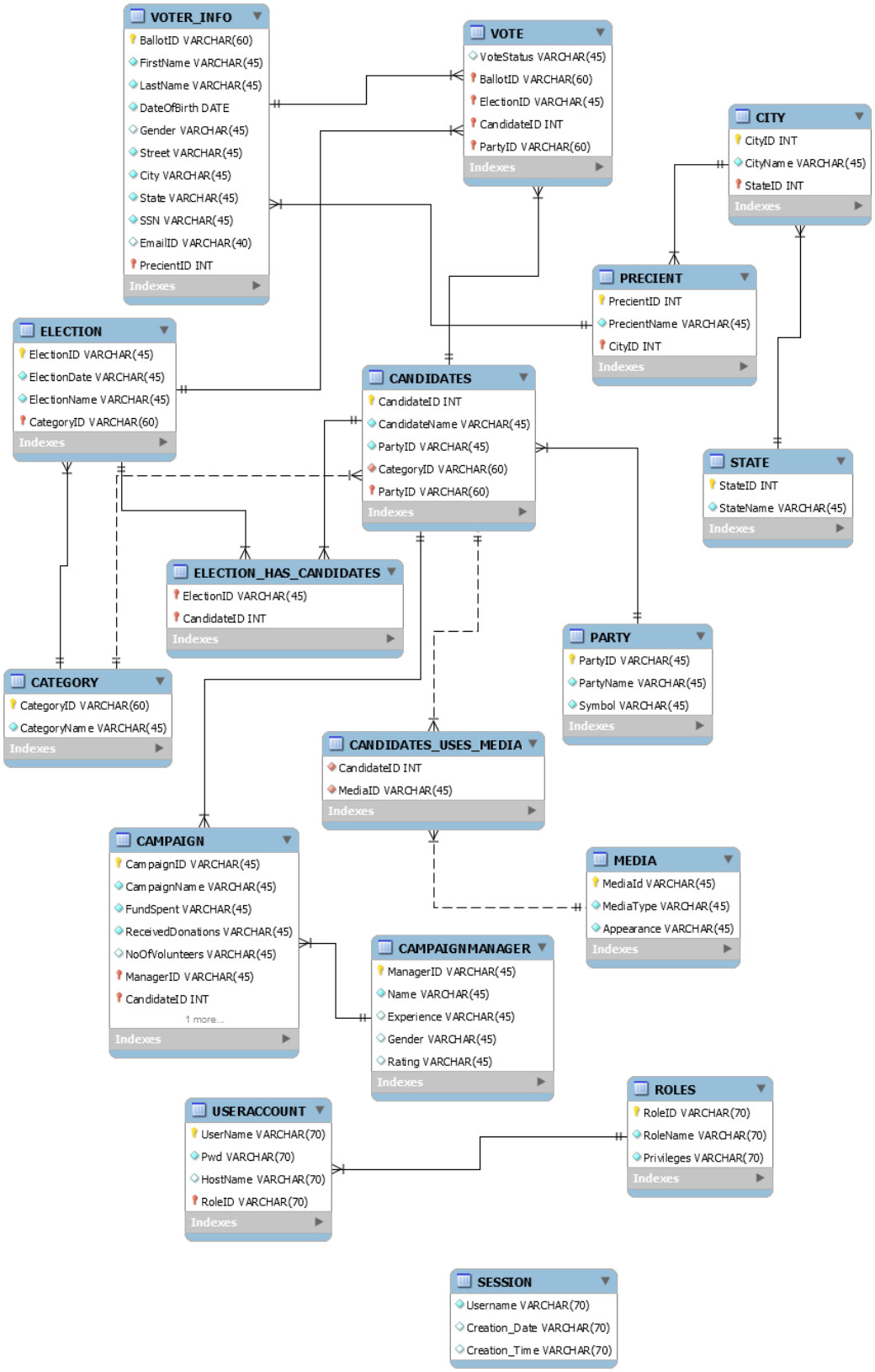
ELECTION DATABASE SYSTEM

Summary:

The main objective of this project is to calculate and retrieve results of one or more elections based on the database created. By adding users to the database and grouping them according to their roles, we provide the basic security for the election database.

Created a database which stores information of multiple elections with voter's information, whom they voted for and where they belong to. In addition to it, the database also consists of candidates (people standing for elections), their campaigns and how much revenue is generated from campaigns. Generated results of an election by displaying the winning candidate's information by using stored procedures, user defined functions, multiple joins, triggers, functions and views.

EER Diagram:


VOTER_INFO
BallotID VARCHAR(60)
FirstName VARCHAR(45)
LastName VARCHAR(45)
DateOfBirth DATE
Gender VARCHAR(45)
Street VARCHAR(45)
City VARCHAR(45)
State VARCHAR(45)
SSN VARCHAR(45)
EmailID VARCHAR(40)
PrecientID INT
Indexes
VOTE
VoteStatus VARCHAR(45)
BallotID VARCHAR(60)
ElectionID VARCHAR(45)
CandidateID INT
PartyID VARCHAR(60)
Indexes
CITY
CityID INT
CityName VARCHAR(45)
StateID INT
Indexes
PRECIENT
PrecientID INT
PrecientName VARCHAR(45)
CityID INT
Indexes
ELECTION
ElectionID VARCHAR(45)
ElectionDate VARCHAR(45)
ElectionName VARCHAR(45)
CategoryID VARCHAR(60)
Indexes
CANDIDATES
CandidateID INT
CandidateName VARCHAR(45)
PartyID VARCHAR(45)
CategoryID VARCHAR(60)
PartyID VARCHAR(60)
Indexes
STATE
StateID INT
StateName VARCHAR(45)
Indexes
ELECTION_HAS_CANDIDATES
ElectionID VARCHAR(45)
CandidateID INT
Indexes
PARTY
PartyID VARCHAR(45)
PartyName VARCHAR(45)
Symbol VARCHAR(45)
Indexes
CATEGORY
CategoryID VARCHAR(60)
CategoryName VARCHAR(45)
Indexes
CANDIDATES_USES_MEDIA
CandidateID INT
MediaID VARCHAR(45)
Indexes
CAMPAIGN
CampaignID VARCHAR(45)
CampaignName VARCHAR(45)
FundSpent VARCHAR(45)
ReceivedDonations VARCHAR(45)
NoOfVolunteers VARCHAR(45)
ManagerID VARCHAR(45)
CandidateID INT
1 more...
Indexes
MEDIA
MediaId VARCHAR(45)
MediaType VARCHAR(45)
Appearance VARCHAR(45)
Indexes
CAMPAIGNMANAGER
ManagerID VARCHAR(45)
Name VARCHAR(45)
Experience VARCHAR(45)
Gender VARCHAR(45)
Rating VARCHAR(45)
Indexes
USERACCOUNT
UserName VARCHAR(70)
Pwd VARCHAR(70)
HostName VARCHAR(70)
RoleID VARCHAR(70)
Indexes
ROLES
RoleID VARCHAR(70)
RoleName VARCHAR(70)
Privileges VARCHAR(70)
Indexes
SESSION
Username VARCHAR(70)
Creation_Date VARCHAR(70)
Creation_Time VARCHAR(70)

QUERIES USED:

Since the data to be retrieved for computation was scattered in multiple tables, usage of joins and subqueries has been the most.

JOINS:

1) **Displaying the ballot numbers (votingid) and election details only of those voters(people) who have voted.**

Query >

```
SELECT V. BALLOTID, V. ELECTIONID, E. ELECTIONNAME, V. CANDIDATEID, C. CANDIDATENAME,
V. PARTYID, P. PARTYNAME
FROM VOTE V
INNER JOIN ELECTION E INNER JOIN CANDIDATES C INNER JOIN PARTY P
ON V. ELECTIONID=E.ELECTIONID AND V. CANDIDATEID=C.CANDIDATEID
AND V. PARTYID=P.PARTYID
WHERE V. VOTESTATUS='YES'
ORDER BY electionid;
```

BALLOTID	ELECTIONID	ELECTIONNAME	CANDIDATEID	CANDIDATENAME	PARTYID	PARTYNAME
331-00	e01	General Presidential 2012	2001	Donald J.Trump	p2	Republican
211-00	e01	General Presidential 2012	1001	Hillary Clinton	p1	Democrats
411-44	e01	General Presidential 2012	1001	Hillary Clinton	p1	Democrats
221-67	e01	General Presidential 2012	2001	Donald J.Trump	p2	Republican
311-24	e01	General Presidential 2012	1001	Hillary Clinton	p1	Democrats
431-41	e01	General Presidential 2012	3001	Ahmed Abdi	p3	Independents
311-36	e02	General US Senator 2012	2002	Mike Pence	p2	Republican
221-68	e02	General US Senator 2012	2002	Mike Pence	p2	Republican
431-45	e02	General US Senator 2012	1002	Brooks Salazar	p1	Democrats
411-54	e02	General US Senator 2012	2002	Mike Pence	p2	Republican
331-04	e02	General US Senator 2012	1002	Brooks Salazar	p1	Democrats
211-04	e02	General US Senator 2012	2002	Mike Pence	p2	Republican
431-52	e03	General Judge 2012	2003	Rex Tillerson	p2	Republican
221-69	e03	General Judge 2012	1003	Ryan Calkins	p1	Democrats
311-43	e03	General Judge 2012	2003	Rex Tillerson	p2	Republican
211-05	e03	General Judge 2012	1003	Ryan Calkins	p1	Democrats
331-05	e03	General Judge 2012	3003	Preeti Shridhra	p3	Independents
411-69	e03	General Judge 2012	2003	Rex Tillerson	p2	Republican
221-79	e04	Primary State Senator 2...	1004	John Creighton	p1	Democrats
211-06	e04	Primary State Senator 2...	1004	John Creighton	p1	Democrats
431-53	e04	Primary State Senator 2...	1004	John Creighton	p1	Democrats
331-08	e04	Primary State Senator 2...	1004	John Creighton	p1	Democrats
311-53	e04	Primary State Senator 2...	1004	John Creighton	p1	Democrats
411-75	e04	Primary State Senator 2...	2004	Steve Mnuchin	p2	Republican
431-54	e05	Primary County Executiv...	3004	Sonny Perdue	p3	Independents
411-77	e05	Primary County Executiv...	3004	Sonny Perdue	p3	Independents
211-09	e05	Primary County Executiv...	3004	Sonny Perdue	p3	Independents
221-81	e05	Primary County Executiv...	2005	Wilbur Ross	p2	Republican

2) Display the total number of votes received by each candidate (person standing in an election) in each election.

Query>

```
select  v.electionid,e.electionname, v.candidateid, c.candidatename, p.partyname, count(v.candidateid) AS Total_Votes
from vote v INNER JOIN Candidates c INNER JOIN Election e INNER JOIN Party p
ON v.ElectionID=e.electionid AND v.CandidateID=c.candidateid AND v.partyid=p.partyid
where v.candidateid is NOT NULL AND v.VoteStatus='yes'
group by v.electionid, v.CandidateID
order by v.electionID asc, total_votes desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

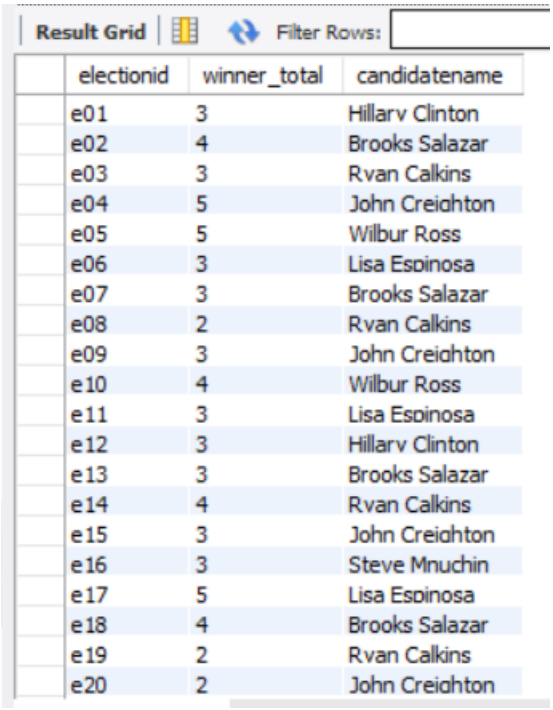
electionid	electionname	candidateid	candidatename	partyname	Total_Votes
e01	General Presidential 2012	1001	Hillary Clinton	Democrats	3
e01	General Presidential 2012	2001	Donald J.Trump	Republican	2
e01	General Presidential 2012	3001	Ahmed Abdi	Independents	1
e02	General US Senator 2012	2002	Mike Pence	Republican	4
e02	General US Senator 2012	1002	Brooks Salazar	Democrats	2
e03	General Judge 2012	2003	Rex Tillerson	Republican	3
e03	General Judge 2012	1003	Ryan Calkins	Democrats	2
e03	General Judge 2012	3003	Preeti Shridhra	Independents	1
e04	Primary State Senator 2013	1004	John Creighton	Democrats	5
e04	Primary State Senator 2013	2004	Steve Mnuchin	Republican	1
e05	Primary County Executive ...	3004	Sonny Perdue	Independents	5
e05	Primary County Executive ...	2005	Wilbur Ross	Republican	1
e06	Primary City Council 2013	1005	Lisa Espinosa	Democrats	3
e06	Primary City Council 2013	3005	Ben Carson	Independents	2
e07	General US Senator 2014	1002	Brooks Salazar	Democrats	3
e07	General US Senator 2014	2002	Mike Pence	Republican	1
e08	General Judge 2014	2003	Rex Tillerson	Republican	2
e08	General Judge 2014	1003	Ryan Calkins	Democrats	1
e08	General Judge 2014	3003	Preeti Shridhra	Independents	1
e09	Primary State Senator 2015	1004	John Creighton	Democrats	3
e10	Primary County Executive ...	2005	Wilbur Ross	Republican	4
e11	Primary City Council 2015	1005	Lisa Espinosa	Democrats	3
e12	General Presidential 2016	1001	Hillary Clinton	Democrats	3

VIEWS:

1) **This view is created to support the next view by retrieving total votes per candidate per election.**

Query>

```
CREATE VIEW voteSummary AS
SELECT v.electionid,v.candidateid, count(v.candidateid) as totalvotes
FROM vote v
WHERE v.votestatus='yes'
GROUP BY v.electionid, v.candidateid;
```



Result Grid | Filter Rows:

electionid	winner_total	candidatename
e01	3	Hillary Clinton
e02	4	Brooks Salazar
e03	3	Ryan Calkins
e04	5	John Creighton
e05	5	Wilbur Ross
e06	3	Lisa Espinosa
e07	3	Brooks Salazar
e08	2	Ryan Calkins
e09	3	John Creighton
e10	4	Wilbur Ross
e11	3	Lisa Espinosa
e12	3	Hillary Clinton
e13	3	Brooks Salazar
e14	4	Ryan Calkins
e15	3	John Creighton
e16	3	Steve Mnuchin
e17	5	Lisa Espinosa
e18	4	Brooks Salazar
e19	2	Ryan Calkins
e20	2	John Creighton

2) **Using the voteSummary view, we create another view to combine the two aggregate functions in order to get the complete election result summary.**

Query>

```
create view completeElectionSummary
AS SELECT  t.electionid, t.electionname, t.total_voters, s.winner_total, is.candidatename
FROM totalvoters t INNER JOIN votesummary s
ON t.electionid = s.electionid;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

electionid	electionname	total_voters	winner_total	candidatename
e01	General Presidential 2012	6	3	Hillary Clinton
e02	General US Senator 2012	6	4	Brooks Salazar
e03	General Judge 2012	6	3	Ryan Calkins
e04	Primary State Senator 2013	6	5	John Creighton
e05	Primary County Executive 2013	6	5	Wilbur Ross
e06	Primary City Council 2013	5	3	Lisa Espinosa
e07	General US Senator 2014	4	3	Brooks Salazar
e08	General Judge 2014	4	2	Ryan Calkins
e09	Primary State Senator 2015	3	3	John Creighton
e10	Primary County Executive 2015	4	4	Wilbur Ross
e11	Primary City Council 2015	3	3	Lisa Espinosa
e12	General Presidential 2016	4	3	Hillary Clinton
e13	General US Senator 2016	5	3	Brooks Salazar
e14	General Judge 2016	5	4	Ryan Calkins
e15	Primary State Senator 2017	5	3	John Creighton
e16	Primary County Executive 2017	5	3	Steve Mnuchin
e17	Primary City Council 2017	5	5	Lisa Espinosa
e18	General US Senator 2018	5	4	Brooks Salazar
e19	General Judge 2018	4	2	Ryan Calkins
e20	Primary State Senator 2019	3	2	John Creighton
e21	Primary County Executive 2019	3	2	Wilbur Ross
e22	Primary City Council 2019	3	3	Lisa Espinosa
e23	General Presidential 2020	3	2	Hillary Clinton
e24	General US Senator 2020	4	2	Brooks Salazar

3) To find total number of votes received by each candidate in each election requires joining and calculation of many tables so we use a view here to simplify the task : Query>

```
CREATE VIEW Winner
AS SELECT v.electionid,e.electionname, v.candidateid, c.candidatename, p.partyname, count(v.candidateid) AS Total_Votes
FROM vote v INNER JOIN Candidates c INNER JOIN Election e INNER JOIN Party p
ON v.ElectionID=e.electionid AND v.CandidateID=c.candidateid AND v.partyid=p.partyid
WHERE v.candidateid is NOT NULL AND v.VoteStatus='yes'
GROUP BY v.electionid, v.CandidateID
ORDER BY v.electionID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

electionid	electionname	candidateid	candidatename	partyname	Total_Votes
e01	General Presidential 2012	1001	Hillary Clinton	Democrats	3
e01	General Presidential 2012	2001	Donald J.Trump	Republican	2
e01	General Presidential 2012	3001	Ahmed Abdi	Independents	1
e02	General US Senator 2012	1002	Brooks Salazar	Democrats	2
e02	General US Senator 2012	2002	Mike Pence	Republican	4
e03	General Judge 2012	1003	Ryan Calkins	Democrats	2
e03	General Judge 2012	2003	Rex Tillerson	Republican	3
e03	General Judge 2012	3003	Preeti Shridhra	Independents	1
e04	Primary State Senator 2013	1004	John Creighton	Democrats	5
e04	Primary State Senator 2013	2004	Steve Mnuchin	Republican	1
e05	Primary County Executive ...	2005	Wilbur Ross	Republican	1
e05	Primary County Executive ...	3004	Sonny Perdue	Independents	5
e06	Primary City Council 2013	1005	Lisa Espinosa	Democrats	3
e06	Primary City Council 2013	3005	Ben Carson	Independents	2
e07	General US Senator 2014	1002	Brooks Salazar	Democrats	3
e07	General US Senator 2014	2002	Mike Pence	Republican	1
e08	General Judge 2014	1003	Ryan Calkins	Democrats	1
e08	General Judge 2014	2003	Rex Tillerson	Republican	2
e08	General Judge 2014	3003	Preeti Shridhra	Independents	1
e09	Primary State Senator 2015	1004	John Creighton	Democrats	3
e10	Primary County Executive ...	2005	Wilbur Ross	Republican	4
e11	Primary City Council 2015	1005	Lisa Espinosa	Democrats	3
e12	General Presidential 2016	1001	Hillary Clinton	Democrats	3
e12	General Presidential 2016	2001	Donald J.Trump	Republican	1

winner 6

STORED PROCEDURES:

In this database, it is shown that candidates can generate funds and influence people by arranging different campaigns.

1) **To calculate revenue generated by each campaign, we use the IN parameter of stored procedures as follows:**

Query>

```
DELIMITER //
CREATE PROCEDURE campaignrevenue(IN b varchar(60))
BEGIN
SELECT c.campaignid, c.campaignname, a.candidatename, p.partyname,
TRUNCATE(Income(c.FundSpent,c.ReceivedDonations), AS INCOME_Million
FROM Campaign c
INNER JOIN candidates a
INNER JOIN Party p
ON c.candidateid=a.candidateid AND p.partyid=a.PartyID
WHERE Campaignid= b
order by income_million desc;
END
//
```

CALL campaignrevenue('e');

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

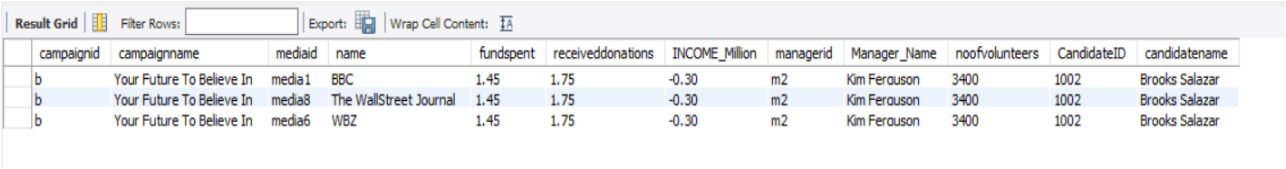
campaignid	campaignname	candidatename	partyname	INCOME_Million
e	People Fighting Back	Ben Carson	Independents	0.01

2) **To get all campaign details of a candidate along with the various media types used for promotion purpose is given by:**

Query>

```
DELIMITER //
CREATE PROCEDURE campaigndetails(IN v varchar(60))
BEGIN
SELECT c.campaignid, c.campaignname, cm.mediaid, mm.name, c.fundspent, c.receiveddonations,
TRUNCATE(Income(c.FundSpent,c.ReceivedDonations),2) AS INCOME_Million, c.managerid,
m.name as Manager_Name, c.noofvolunteers, c.CandidateID, a.candidatename
FROM Campaign c
inner join candidates_uses_media cm
INNER JOIN candidates a
INNER JOIN campaignManager m
INNER JOIN media mm
ON c.candidateid=a.candidateid AND m.managerid=c.managerid AND mm.MediaID=cm.MediaID
AND cm.candidateid=a.CandidateID
where c.candidateid=v order by campaignid;
END
//
```

call campaigndetails(1002);



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

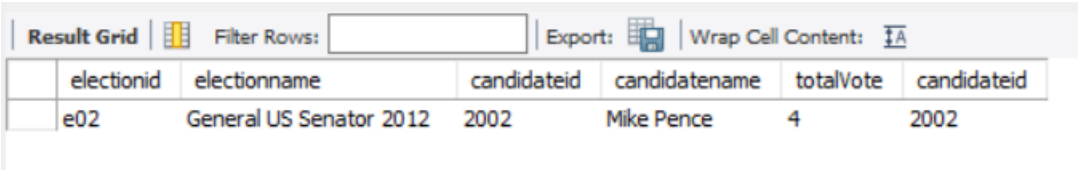
campaignid	campaignname	mediaid	name	fundspent	receiveddonations	INCOME_Million	managerid	Manager_Name	noofvolunteers	CandidateID	candidatename
b	Your Future To Believe In	media1	BBC	1.45	1.75	-0.30	m2	Kim Ferguson	3400	1002	Brooks Salazar
b	Your Future To Believe In	media8	The WallStreet Journal	1.45	1.75	-0.30	m2	Kim Ferguson	3400	1002	Brooks Salazar
b	Your Future To Believe In	media6	WBZ	1.45	1.75	-0.30	m2	Kim Ferguson	3400	1002	Brooks Salazar

3) To find the candidate who won an election based on user input electionid is:

Query>

```
DELIMITER //
CREATE PROCEDURE electionresult(IN a varchar(60))
BEGIN
SELECT t.electionid, t.electionname, t.candidateid,t.candidatename,max(t.total)as totalVote, t.candidateid
FROM (SELECT electionid, electionname, candidateid, candidatename, max(total_votes) AS Total
FROM Winner GROUP BY candidateid, electionid) t
WHERE electionid = a
GROUP BY t.electionid;
END
//
```

CALL electionresult('e02');



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

electionid	electionname	candidateid	candidatename	totalVote	candidateid
e02	General US Senator 2012	2002	Mike Pence	4	2002

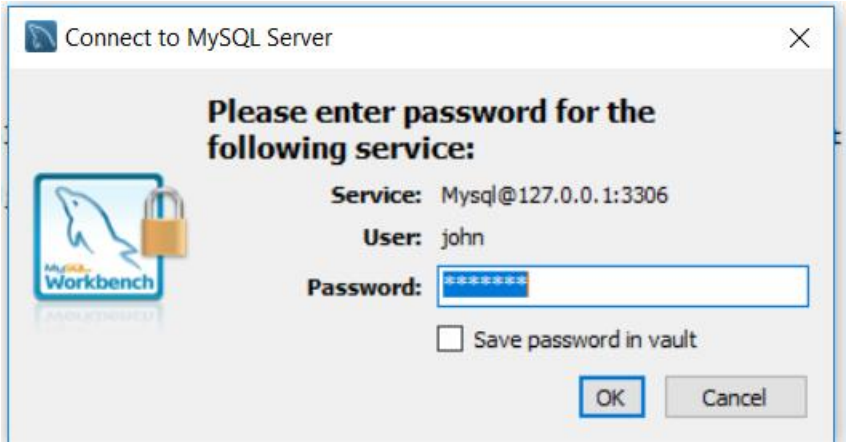
USER PRIVILEGES:

There are two user roles defined in this database. One is the Poll Worker who has limited access to database and other is Election Admin who has all the privileges to make changes to the database.

To Create Poll worker Users:

1)
Query>

```
CREATE USER 'john'@'localhost'
IDENTIFIED BY 'john123';
GRANT SELECT,EXECUTE ON electionsystem.*
TO 'john'@'localhost';
```

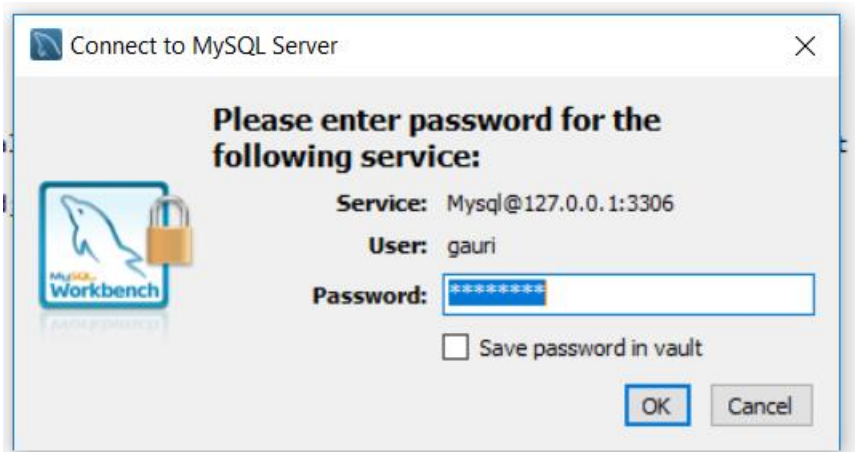


2)

```
CREATE USER 'jack'@'localhost'
IDENTIFIED BY 'jack123';
GRANT SELECT,EXECUTE ON electionsystem.*
TO 'jack'@'localhost';
```

To Create Election Admin User:

1)
CREATE USER 'gauri'@'localhost'
IDENTIFIED BY 'gauri123';
GRANT ALL on electionsystem.*
TO 'gauri'@'localhost';



ADDITIONAL FEATURES

ENCRYPTION:

Encryption of password field in useraccount table so it won't be accessed by anyone.
Query>

```
insert into useraccount
values('u1','John',MD5('john123'),'localhost','r1');

insert into useraccount
values('u2','Jack',MD5('jack123'),'localhost','r1');

insert into useraccount
values('u3','Gauri',Password('gauri123'),'localhost','r2');
```

Result Grid | Filter Rows: | Edit: | Export/Import:

userid	username	pwd	hostname	roleid
u1	John	6e0b7076126a29d5dfcbd54835387b7b	localhost	r1
u2	Jack	1d6c1e168e362bc0092f247399003a88	localhost	r1
u3	Gauri	*384F41980A268B224BE3FCA849F17A99E7A0...	localhost	r2
NULL	NULL	NULL	NULL	NULL

XML:

Getting the result of election summary in XML format for presenting it as a report.

Query>

```
select concat(
'<? xml version="1.0" encoding="UTF-8" standalone="no" ?>',
group_concat('<ElectionID>',electionid,'<ElectionName>',electionname,'<Total_Voters>',total_voters,'<Votes_For_
Winner>',winner_total,'<Winner_Name>',candidatename)
) AS XML
from completeelectionsummary;
```

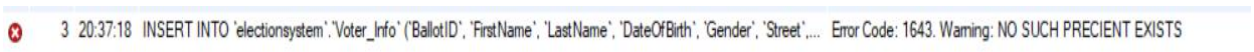
```
'<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"no\" ?>
<Election>
<ElectionID>e01<ElectionName>General Presidential 2012<Total_Voters>6<Votes_For_Winner>3<Winner_Name>Hillary Clinton,
<ElectionID>e02<ElectionName>General US Senator 2012<Total_Voters>6<Votes_For_Winner>4<Winner_Name>Brooks Salazar,
<ElectionID>e03<ElectionName>General Judge 2012<Total_Voters>6<Votes_For_Winner>3<Winner_Name>Ryan Calkins,
<ElectionID>e04<ElectionName>Primary State Senator 2013<Total_Voters>6<Votes_For_Winner>5<Winner_Name>John Creighton,
<ElectionID>e05<ElectionName>Primary County Executive 2013<Total_Voters>6<Votes_For_Winner>5<Winner_Name>Wilbur Ross,
<ElectionID>e06<ElectionName>Primary City Council 2013<Total_Voters>5<Votes_For_Winner>3<Winner_Name>Lisa Espinosa,
<ElectionID>e07<ElectionName>General US Senator 2014<Total_Voters>4<Votes_For_Winner>3<Winner_Name>Brooks Salazar,
<ElectionID>e08<ElectionName>General Judge 2014<Total_Voters>4<Votes_For_Winner>2<Winner_Name>Ryan Calkins,
<ElectionID>e09<ElectionName>Primary State Senator 2015<Total_Voters>3<Votes_For_Winner>3<Winner_Name>John Creighton'
</Election>
```

TRIGGERS:

1) **To restrict the entry of a voter with a precientid that is not present in the given prescient list.**

   Query>

```
DELIMITER $$
CREATE TRIGGER triggervoter
BEFORE
INSERT ON Voter_Info
FOR EACH ROW BEGIN
   IF (NEW.precientid NOT IN (select precientid from precient)) THEN
      SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'Warning: NO SUCH PRECIENT EXISTS';
   END IF;
END$$
DELIMITER;
```

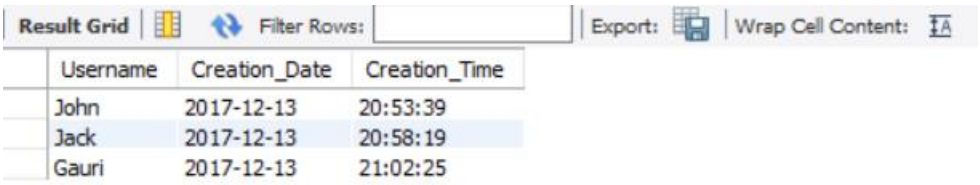


2) **To update the session table with the username, creation date and creating time as per the new users who are been given certain access to this database in useraccount123 table.**

   Query>

```
DELIMITER $$
CREATE TRIGGER timerabc
AFTER
INSERT ON useraccount123
```

```
FOR EACH ROW
BEGIN
   insert into session(username,creation_date,creation_time) values(NEW.username,curdate(),curtime());
END$$
DELIMITER;
```



Username	Creation_Date	Creation_Time
John	2017-12-13	20:53:39
Jack	2017-12-13	20:58:19
Gauri	2017-12-13	21:02:25

FUNCTIONS:

1) **TRUNCATE ();** - The TRUNCATE () function returns a number truncated to a certain number of decimal places.

2) **CURDATE ();** -Gives the current system date.

3) **CURTIME ();** - Gives the current system time.

4) **PASSWORD ();** - Generates a Hashed password from a plaintext.

5) **MD5 ();** - Calculates MD5 checksum for the given field.