

- فاز سوم -

پیاده سازی طراحی ها

نام گروه: Sakila

اسامی اعضای گروه: پگاه گرایلو

مهتاب جیحانی

نرگس قنبری

حسین میرزاگل

استاد مربوطه: دکتر فاطمه بهاریفرد

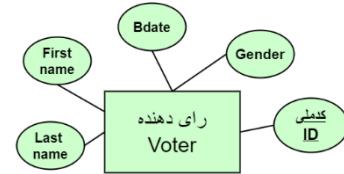
فهرست

3	تعاریف جداول
19	پیاده سازی دید های کاربری
31	پیاده سازی محدودیت های معنایی
43	پیاده سازی index
44	دیتاست ها و چند نمونه از آزمون پیاده سازی ها

تعاریف جدول ها

جدول رای دهنده:

voter				
NationalID	FirsName	LastName	Bdate	Gender



Query Query History

```

1  create table Voter (
2      NationalID char(10), FirstName varchar(10),
3      LastName varchar(10), Bdate date, Gender varchar(6),
4      check (Gender = 'Female' or Gender = 'Male'),
5      primary key(NationalId)
6  );

```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 65 msec.

----- جدول کاندیدها:

NationalID	FirstName	LastName	Bdate	Religen	City	PoliticalOrientation	Gender	EducationDegree	EducationField

Query Query History

```

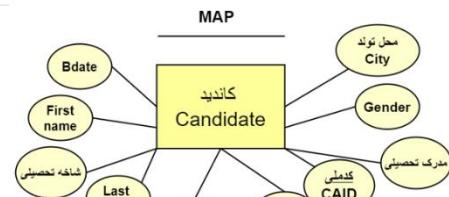
7  /*-----*/
8  create table Candidate (
9      NationalID char(10), FirstName varchar(10),
10     LastName varchar(10), Bdate date,
11     Religen varchar(20), City varchar(10),
12     PoliticalOrientation varchar(20),
13     Gender varchar(6),
14     check (Gender = 'Female' or Gender = 'Male'),
15     EducationDegree varchar(20),
16     EducationField varchar(20),
17     check (Religen = 'Islam' or Religen = 'Christianity'),
18     primary key(NationalId)
19  );

```

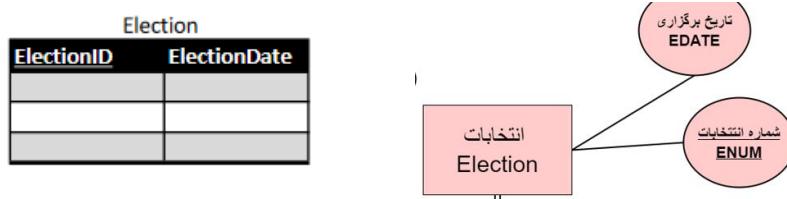
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 63 msec.



جدول انتخابات:



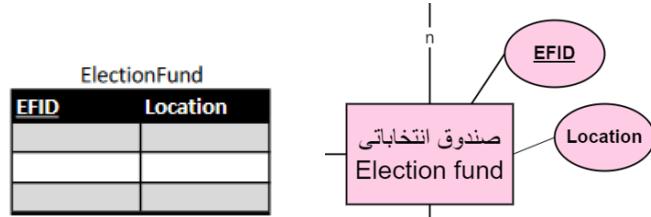
```
20 /*-----*/
21 create table Election(
22     ElectionID smallint, ElectionDate date,
23     primary key (ElectionID)
24 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 60 msec.

جدول صندوق انتخاباتی:



```
25 /*-----*/
26 create table ElectionFund(
27     EFID smallint, Location varchar(20),
28     primary key (EFID)
29 );
```

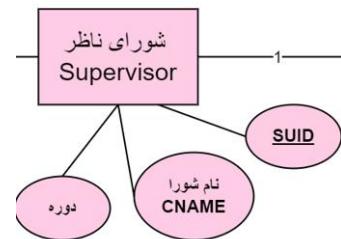
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 50 msec.

جدول شورای ناظر:

Supervisor		
SUID	CouncilName	CouncilPeriod



```

30 /*-----*/
31 create table Supervisor(
32         SUID smallint, CouncilName varchar(20),
33         CouncilPeriod smallint,
34         primary key (SUID)
35 );
  
```

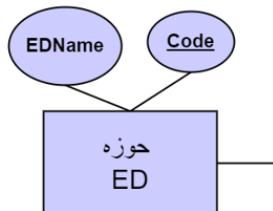
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 51 msec.

جدول حوزه:

ElectoralDistrict	
EDID	EDName



```

36 /*-----*/
37 create table ElectoralDistrict(
38         EDID smallint, EDName varchar(10),
39         primary key (EDID)
40 );
  
```

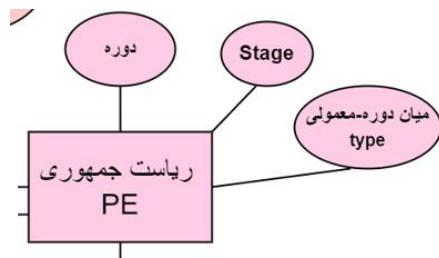
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 45 msec.

جدول انتخابات ریاست جمهوری:

PresidentialElection			
ElectionID	ElectionPeriod	Stage	Type



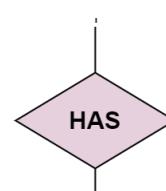
```
41 /*-----*/
42 create table PresidentialElection(
43     ElectionID smallint, ElectionPeriod smallint,
44     Stage smallint, Type varchar(10),
45     check(Type = 'normal' or Type = 'mid term'),
46     primary key (ElectionID),
47     foreign key (ElectionID) references Election(ElectionID)
48     on update cascade on delete cascade
49 );
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 57 msec.

PEEF(HAS)	
ElectionID	EFID



```
50 /*-----*/
51 create table PEEF(
52     ElectionID smallint, EFID smallint,
53     primary key (ElectionID, EFID),
54     foreign key (ElectionID) references Election(ElectionID)
55     on update cascade on delete cascade,
56     foreign key (EFID) references ElectionFund(EFID)
57     on update cascade on delete cascade
58 );
```

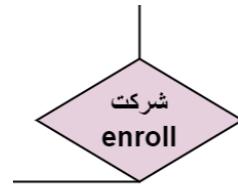
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 58 msec.

----- جدول شرکت در انتخابات ریاست جمهوری: -----

PEEnroll	
ElectionID	CANID



```
59 /*-----*/
60 create table PEEroll(
61     ElectionID smallint, CANID char(10),
62     primary key (ElectionID, CANID),
63     foreign key (ElectionID) references Election(ElectionID)
64     on update cascade on delete cascade,
65     foreign key (CANID) references Candidate(NationalID)
66     on update cascade on delete cascade
67 );
```

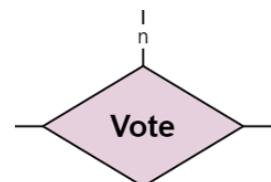
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 51 msec.

----- جدول vote در انتخابات ریاست جمهوری: -----

PEVote		
VOID	CANID	EFID



```
69 create table PEVote(
70     VOID char(10), CANID char(10),
71     EFID smallint,
72     primary key (VOID, CANID, EFID),
73     foreign key (VOID) references Voter(NationalID)
74     on update cascade on delete cascade,
75     foreign key (CANID) references Candidate(NationalID)
76     on update cascade on delete cascade,
77     foreign key (EFID) references ElectionFund(EFID)
78     on update cascade on delete cascade
79 );
```

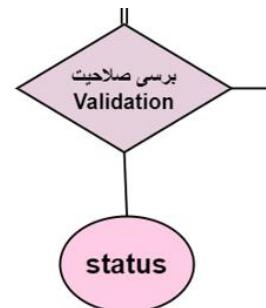
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 47 msec.

----- جدول بررسی صلاحیت انتخابات ریاست جمهوری: -----

PEValidation		
SUID	CANID	status



```
80 /*-----*/
81 create table PEValidation(
82     SUID smallint, CANID char(10),
83     status varchar(20),
84     check( status = 'qualified' or status = 'disqualified'),
85     primary key (SUID, CANID),
86     foreign key (SUID) references Supervisor(SUID)
87     on update cascade on delete cascade,
88     foreign key (CANID) references Candidate(NationalID)
89     on update cascade on delete cascade
90 );
```

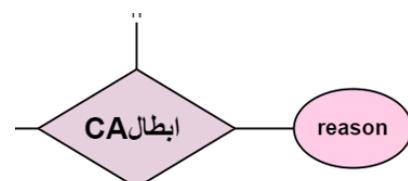
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 55 msec.

----- جدول رابطه ابطال انتخابات ریاست جمهوری: -----

EFPECanelation		
SUID	EFID	reason



```
91 /*-----*/
92 create table EFPECanelation(
93     SUID smallint, EFID smallint,
94     reason varchar(100),
95     primary key (SUID, EFID),
96     foreign key (SUID) references Supervisor(SUID)
97     on update cascade on delete cascade,
98     foreign key (EFID) references ElectionFund(EFID)
99     on update cascade on delete cascade
100 );
```

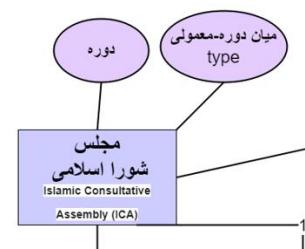
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 50 msec.

جدول انتخابات مجلس شورای اسلامی:

IslamicConsultativeAssemblyElection		
ElectionID	ElectionPeriod	Type



```
101 /*-----*/
102 create table IslamicConsultativeAssemblyElection(
103     ElectionID smallint, ElectionPeriod int,
104     Type varchar(10),
105     check(Type = 'normal' or Type ='mid term'),
106     primary key (ElectionID),
107     foreign key (ElectionID) references Election(ElectionID)
108     on update cascade on delete cascade
109 );
```

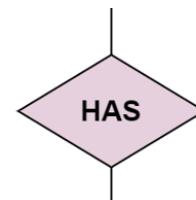
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 65 msec.

جدول has انتخابات مجلس شورای اسلامی:

ICAEF(HAS)	
ElectionID	EFID



```
110 /*-----*/
111 create table ICAEF(
112     ElectionID smallint, EFID smallint,
113     primary key (ElectionID, EFID),
114     foreign key (ElectionID) references Election(ElectionID)
115     on update cascade on delete cascade,
116     foreign key (EFID) references ElectionFund(EFID)
117     on update cascade on delete cascade
118 );
```

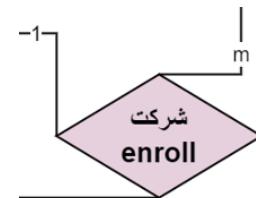
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 49 msec.

----- جدول شرکت در انتخابات مجلس شورای اسلامی: -----

ElectionID	CANID	EDID



```
119 /*-----*/
120 create table ICAEnroll(
121     ElectionID smallint, CANID char(10),
122     EDID smallint,
123     primary key (ElectionID, CANID, EDID),
124     foreign key (ElectionID) references Election(ElectionID)
125     on update cascade on delete cascade,
126     foreign key (CANID) references Candidate(NationalID)
127     on update cascade on delete cascade,
128     foreign key (EDID) references ElectoralDistrict(EDID)
129     on update cascade on delete cascade
130 );
```

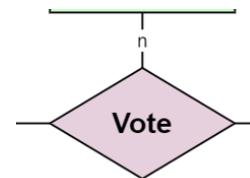
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 51 msec.

----- جدول vote در انتخابات مجلس شورای اسلامی: -----

VOID	CANID	EFID



```
131 /*-----*/
132 create table ICAVote(
133     VOID char(10), CANID char(10), EFID smallint,
134     primary key (VOID, CANID, EFID),
135     foreign key (VOID) references Voter(NationalID)
136     on update cascade on delete cascade,
137     foreign key (CANID) references Candidate(NationalID)
138     on update cascade on delete cascade,
139     foreign key (EFID) references ElectionFund(EFID)
140     on update cascade on delete cascade
141 );
```

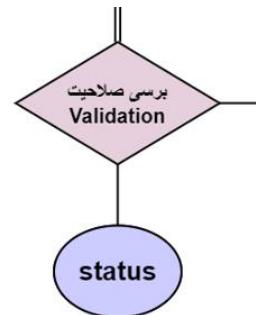
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 48 msec.

جدول بررسی صلاحیت در انتخابات مجلس شورای اسلامی:

ICAValidation		
SUID	CANID	status



```
142 /*-----*/
143 create table ICAValidation(
144     SUID smallint, CANID char(10),
145     status varchar(20),
146     check( status = 'qualified' or status = 'disqualified'),
147     primary key (SUID, CANID),
148     foreign key (SUID) references Supervisor(SUID)
149     on update cascade on delete cascade,
150     foreign key (CANID) references Candidate(NationalID)
151     on update cascade on delete cascade
152 );|
```

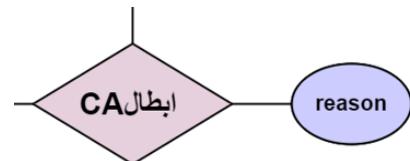
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 50 msec.

جدول رابطه ابطال در انتخابات مجلس شورای اسلامی:

EFICACancelation		
SUID	EFID	reason



```
153 /*-----*/
154 create table EFICACancelation(
155     SUID smallint, EFID smallint,
156     reason varchar(100),
157     primary key (SUID, EFID),
158     foreign key (SUID) references Supervisor(SUID)
159     on update cascade on delete cascade,
160     foreign key (EFID) references ElectionFund(EFID)
161     on update cascade on delete cascade
162 );|
```

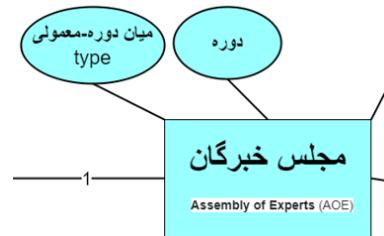
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 45 msec.

جدول رابطه انتخابات مجلس خبرگان:

AssemblyOfExpertsElection		
ElectionID	ElectionPeriod	Type



```
163 /*-----*/
164 create table AssemblyOfExpertsElection(
165     ElectionID smallint, ElectionPeriod int,
166     Type varchar(10),
167     check(Type = 'normal' or Type ='mid term'),
168     primary key (ElectionID),
169     foreign key (ElectionID) references Election(ElectionID)
170     on update cascade on delete cascade
171 );|
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 154 msec.

جدول رابطه has انتخابات مجلس خبرگان:

AOEEF(HAS)	
ElectionID	EFID



```
172 /*-----*/
173 create table AOEEF(
174     ElectionID smallint, EFID smallint,
175     primary key (ElectionID, EFID),
176     foreign key (ElectionID) references Election(ElectionID)
177     on update cascade on delete cascade,
178     foreign key (EFID) references ElectionFund(EFID)
179     on update cascade on delete cascade
180 );|
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 142 msec.

----- جدول رابطه شركت انتخابات مجلس خبرگان:

AOEEnroll		
ElectionID	CANID	EDID



```
181 /*-----*/
182 create table AOEEnroll(
183     ElectionID smallint, CANID char(10),
184     EDID smallint,
185     primary key (ElectionID, CANID, EDID),
186     foreign key (ElectionID) references Election(ElectionID)
187     on update cascade on delete cascade,
188     foreign key (CANID) references Candidate(NationalID)
189     on update cascade on delete cascade,
190     foreign key (EDID) references ElectoralDistrict(EDID)
191     on update cascade on delete cascade
192 );
```

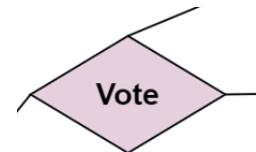
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 47 msec.

----- جدول رابطه vote انتخابات مجلس خبرگان:

AOEVote		
VOID	CANID	EFID



```
193 /*-----*/
194 create table AOEVote(
195     VOID char(10), CANID char(10), EFID smallint,
196     primary key (VOID, CANID, EFID),
197     foreign key (VOID) references Voter(NationalID)
198     on update cascade on delete cascade,
199     foreign key (CANID) references Candidate(NationalID)
200     on update cascade on delete cascade,
201     foreign key (EFID) references ElectionFund(EFID)
202     on update cascade on delete cascade
203 );
```

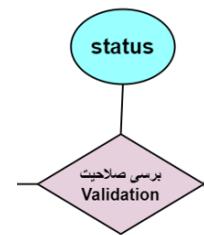
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 49 msec.

جدول رابطه برسي صلاحيت انتخابات مجلس خبرگان:

AOEValidation		
SUID	CANID	status



```

204  /*-----*/
205  create table AOEValidation(
206      SUID smallint, CANID char(10),
207      status varchar(20),
208      check( status = 'qualified' or status = 'disqualified'),
209      primary key (SUID, CANID),
210      foreign key (SUID) references Supervisor(SUID)
211      on update cascade on delete cascade,
212      foreign key (CANID) references Candidate(NationalID)
213      on update cascade on delete cascade
214 );
215  /*-----*/
    
```

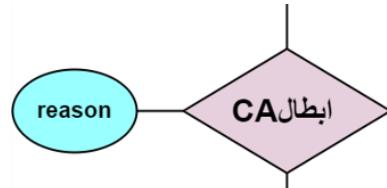
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 49 msec.

جدول رابطه ابطال انتخابات مجلس خبرگان:

EFAOECanelation		
SUID	EFID	reason



```

215  /*-----*/
216  create table EFAOECanelation(
217      SUID smallint, EFID smallint,
218      reason varchar(100),
219      primary key (SUID, EFID),
220      foreign key (SUID) references Supervisor(SUID)
221      on update cascade on delete cascade,
222      foreign key (EFID) references ElectionFund(EFID)
223      on update cascade on delete cascade
224 );
225  /*-----*/
    
```

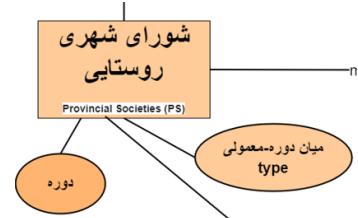
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 50 msec.

جدول انتخابات شهری روستایی:

ProvincialSocietiesElection		
ElectionID	ElectionPeriod	Type



```

225 /*-----*/
226 create table ProvincialSocietiesElection(
227     ElectionID smallint, ElectionPeriod int,
228     Type varchar(10),
229     check(Type = 'normal' or Type = 'mid term'),
230     primary key (ElectionID),
231     foreign key (ElectionID) references Election(ElectionID)
232     on update cascade on delete cascade
233 );
234 /*-----*/
235

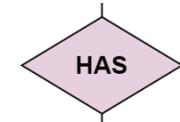
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 48 msec.

PSEF(HAS)	
ElectionID	EFID



```

234 /*-----*/
235 create table PSEF(
236     ElectionID smallint, EFID smallint,
237     primary key (ElectionID, EFID),
238     foreign key (ElectionID) references Election(ElectionID)
239     on update cascade on delete cascade,
240     foreign key (EFID) references ElectionFund(EFID)
241     on update cascade on delete cascade
242 );
243 /*-----*/
244

```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 55 msec.

----- جدول رابطه enroll انتخابات شهری روستایی: -----

PSEnroll	
ElectionID	CANID



```
243 /*-----*/
244 create table PSEnroll(
245     ElectionID smallint, CANID char(10),
246     EDID smallint,
247     primary key (ElectionID, CANID, EDID),
248     foreign key (ElectionID) references Election(ElectionID)
249     on update cascade on delete cascade,
250     foreign key (CANID) references Candidate(NationalID)
251     on update cascade on delete cascade,
252     foreign key (EDID) references ElectoralDistrict(EDID)
253     on update cascade on delete cascade
254 );
255 /*-----*/
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 48 msec.

----- جدول رابطه vote انتخابات شهری روستایی: -----

PSVote		
VOID	CANID	EFID



```
255 /*-----*/
256 create table PSVote(
257     VOID char(10), CANID char(10), EFID smallint,
258     primary key (VOID, CANID, EFID),
259     foreign key (VOID) references Voter(NationalID)
260     on update cascade on delete cascade,
261     foreign key (CANID) references Candidate(NationalID)
262     on update cascade on delete cascade,
263     foreign key (EFID) references ElectionFund(EFID)
264     on update cascade on delete cascade
265 );
```

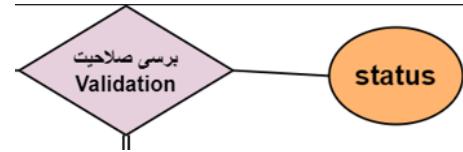
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 54 msec.

----- جدول رابطه برسي صلاحیت انتخابات شهری روستایی: -----

PSValidation		
SUID	CANID	status



```

266 /*-----*/
267 create table PSValidation(
268     SUID smallint, CANID char(10),
269     status varchar(20),
270     check( status = 'qualified' or status = 'disqualified'),
271     primary key (SUID, CANID),
272     foreign key (SUID) references Supervisor(SUID)
273     on update cascade on delete cascade,
274     foreign key (CANID) references Candidate(NationalID)
275     on update cascade on delete cascade
276 );
    
```

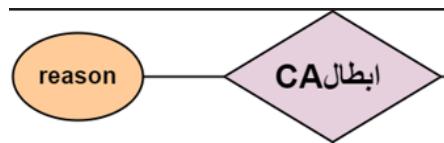
Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 51 msec.

----- جدول رابطه ابطال انتخابات شهری روستایی: -----

EFPSCancelation		
SUID	EFID	reason



```

277 /*-----*/
278 create table EFPSCancelation(
279     SUID smallint, EFID smallint,
280     reason varchar(100),
281     primary key (SUID, EFID),
282     foreign key (SUID) references Supervisor(SUID)
283     on update cascade on delete cascade,
284     foreign key (EFID) references ElectionFund(EFID)
285     on update cascade on delete cascade
286 );
    
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 85 msec.

Browser



Tables (30)

- > aoef
- > aoenroll
- > aoevalidation
- > aoevote
- > assemblyofexpertselection
- > candidate
- > efaoecanelation
- > eficacanelation
- > efpecanelation
- > efpscanelation
- > election
- > electionfund
- > electoraldistrict
- > icaef
- > icaenroll
- > icavaldiation
- > icavote
- > islamicconsultativeassemblyelect
- > peef
- > peenroll
- > pevalidation
- > pvote
- > presidentalelection
- > provincialsocietieselection
- > psef
- > psenroll
- > psvalidation
- > psvote
- > supervisor

voter

خوب جداول به سلامت اجرا شدن 😊

پیاده سازی دید های کاربری

کاربر موردنظر: کاربر رای دهنده (Voter)

- رای دهنده می خواهد لیست نامرد های انتخاباتی که می تواند به آن رای دهد را ببیند. ویژگی هایی که برایش مهم است و می تواند بر تصمیم گیری اش در انتخاب نامزد تاثیر گذار باشد را انتخاب می کنیم.
- ویویی برای معرفی نامزد های انتخاباتی ریاست جمهوری:

```
287 /*-----VIEWS FOR USERS-----*/
288 /*-----REQUIRED VIEWS FOR USERS AS VOTER (PEElection)--*/
289 CREATE VIEW Candidate_Qualified_List AS
290 SELECT Firstname,
291     Lastname,
292     AGE(NOW(), Bdate) as Age,
293     Religen,
294     City,
295     PoliticalOrientation,
296     Gender,
297     EducationDegree,
298     EducationField
299 FROM Candidate INNER JOIN PEValidation
300 ON Candidate.NationalID = PEValidation.CANID
301 WHERE Status = 'qualified';
302 SELECT * FROM Candidate_Qualified_List;
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 52 msec.

```
302 SELECT * FROM Candidate_Qualified_List;
303
Data Output    Messages    Notifications
```

firstname	lastname	age	religen	city	politicalorientation	gender	educationdegree	educationfield
character varying (10)	character varying (10)	interval	character varying (20)	character varying (10)	character varying (20)	character varying (6)	character varying (20)	character varying (20)

برای مشاهده دقیق ویو می توانید با + زوم کنید!
-VIEW1-

▪ ویویی برای نامزد های انتخاباتی مجلس شورای اسلامی:

```
303 /*-----REQUIRED VIEWS FOR USERS AS VOTER (ICAEElection)---*/
304 CREATE VIEW CandidateICA_Qualified_List AS
305 SELECT Firstname,
306     Lastname,
307     AGE(NOW(), Bdate) as Age,
308     Religen,
309     City,
310     PoliticalOrientation,
311     Gender,
312     EducationDegree,
313     EducationField
314 FROM Candidate INNER JOIN ICAValidation
315 ON Candidate.NationalID = ICAValidation.CANID
316 WHERE Status = 'qualified';
317 SELECT * FROM Candidate_Qualified_List;
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 51 msec.

-VIEW2-

جدول آن دقیقاً مثله بقیه انتخابات است صرفاً نوع دیتا های آن متفاوت است.

▪ ویویی برای نامزد های انتخاباتی مجلس خبرگان:

```
317 /*-----REQUIRED VIEWS FOR USERS AS VOTER (AOEElection)---*/
318 CREATE VIEW CandidateAOE_Qualified_List AS
319 SELECT Firstname,
320     Lastname,
321     AGE(NOW(), Bdate) as Age,
322     Religen,
323     City,
324     PoliticalOrientation,
325     Gender,
326     EducationDegree,
327     EducationField
328 FROM Candidate INNER JOIN AOEVValidation
329 ON Candidate.NationalID = AOEVValidation.CANID
330 WHERE Status = 'qualified';
331 SELECT * FROM CandidateAOE_Qualified_List;
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 54 msec.

-VIEW-3

▪ ویویی برای نامزد های انتخاباتی شورای شهری روستایی:

```
332 /*-----REQUIRED VIEWS FOR USERS AS VOTER (AOEElection)---*/
333 CREATE VIEW CandidatePS_Qualified_List AS
334 SELECT Firstname,
335     Lastname,
336     AGE(NOW(), Bdate) as Age,
337     Religen,
338     City,
339     PoliticalOrientation,
340     Gender,
341     EducationDegree,
342     EducationField
343 FROM Candidate INNER JOIN PSValidation
344 ON Candidate.NationalID = PSValidation.CANID
345 WHERE Status = 'qualified';
346 SELECT * FROM CandidatePS_Qualified_List;
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 43 msec.

-VIEW4-

کاربر مورد نظر: کاربر نامزد انتخاباتی

- هر کاندید میخواهد ببیند که در بررسی شورای ناظران، واحد شرایط انتخابات منحصر به خوشون هستند یا نه. به عبارتی دیگر تائید شده اند یا رد یا به طور کل کدام یک از رقیب هایشان تائید و کدام ها رد شدند!

▪ ویویی برای وضعیت کاندیدهای انتخابات ریاست جمهوری:

```
347 /*-----REQUIRED VIEWS FOR USERS AS A CANDIDATE (PEElection)---*/
348 CREATE VIEW PECandidate_qualification AS
349 SELECT NationalID, Firstname, Lastname, Status, ElectionID
350 FROM Candidate INNER JOIN PEValidation ON Candidate.NationalID = PEValidation.CANID
351 INNER JOIN PEEroll ON PEEroll.CANID = Candidate.NationalID;
352 SELECT * FROM PECandidate_qualification;
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 42 msec.

-VIEW5-

```
352 SELECT * FROM PECandidate_qualification;
```

Data Output Messages Notifications

	nationalid character (10)	firstname character varying (10)	lastname character varying (10)	status character varying (20)	electionid smallint	

در این جدول کاندید به راحتی می تواند مشخصات را ببیند که در انتخابات کدام دوره و با چه وضعیتی است.

و یویی برای وضعیت نامزد های انتخاباتی مجلس شورای اسلامی:

```
353 /*-----REQUIRED VIEWS FOR USERS AS A CANDIDATE (ICASelection)--*/  
354 CREATE VIEW ICACandidate_qualification AS  
355 SELECT NationalID, Firstname, Lastname, Status, ElectionID  
356 FROM Candidate INNER JOIN ICAValidation ON Candidate.NationalID = ICAValidation.CANID  
357 INNER JOIN ICACandidate_enroll ON ICACandidate_enroll.CANID = Candidate.NationalID;  
358 SELECT * FROM ICACandidate_qualification;
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 86 msec.

-VIEW6-

شکل جدول یکسان و دیتاها متفاوت است

و یویی برای وضعیت نامزد های انتخاباتی مجلس خبرگان:

```
353 /*-----REQUIRED VIEWS FOR USERS AS A CANDIDATE (AOEElection)--*/  
354 CREATE VIEW AOECandidate_qualification AS  
355 SELECT NationalID, Firstname, Lastname, Status, ElectionID  
356 FROM Candidate INNER JOIN AOEVValidation ON Candidate.NationalID = AOEVValidation.CANID  
357 INNER JOIN AOEECandidate_enroll ON AOEECandidate_enroll.CANID = Candidate.NationalID;  
358 SELECT * FROM AOECandidate_qualification;
```

Data Output Messages Notifications

	nationalid character (10)	firstname character varying (10)	lastname character varying (10)	status character varying (20)	electionid smallint	

-VIEW7-

▪ ویویی برای وضعیت نامزد های انتخاباتی شورای شهری روستایی:

```
359 /*-----REQUIRED VIEWS FOR USERS AS A CANDIDATE (PSElection)---*/  
360 CREATE VIEW PSCandidate_qualification AS  
361 SELECT NationalID, Firstname, Lastname, Status, ElectionID  
362 FROM Candidate INNER JOIN PSValidation ON Candidate.NationalID = PSValidation.CANID  
363 INNER JOIN PSEnroll ON PSEnroll.CANID = Candidate.NationalID;  
364 SELECT * FROM PSCandidate_qualification;
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 147 msec.

-VIEW8-

کاربر موردنظر: شورای ناظر

- شورای ناظر لازم دارد هر زمان باید بتواند علت ابطال صندوقی در دوره های مختلف انتخاباتی ببیند تا شاید بتواند از اتفاق دوباره آن در آینده جلوگیری کند.
- صندوق های انتخاباتی ریاست جمهوری

```
365 /*-----REQUIRED VIEWS FOR USERS AS A SUPPERVISOR (PSElection)---*/  
366 CREATE VIEW Evaluation_of_PEElectionfund AS  
367 SELECT EFPECanelation.EFID, Location, reason, ElectionPeriod  
368 FROM EFPECanelation INNER JOIN ElectionFund ON EFPECanelation.EFID = ElectionFund.EFID  
369 INNER JOIN PEEF ON PEEF.EFID = ElectionFund.EFID  
370 INNER JOIN PresidentialElection ON PresidentialElection.ElectionID = PEEF.ElectionID  
371 SELECT * FROM Evaluation_of_PEElectionfund
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 51 msec.

-VIEW9-

371 **SELECT * FROM Evaluation_of_PEElectionfund**

Data Output Messages Notifications

	efid	location	reason	electionperiod
	smallint	character varying (20)	character varying (100)	smallint

▪ صندوق های انتخاباتی مجلس شورای اسلامی:

```
372 /*-----REQUIRED VIEWS FOR USERS AS A SUPPERVISOR (ICAselection)---*/
373 CREATE VIEW Evaluation_of_ICAselectionfund AS
374 SELECT EFICACanelation.EFID, Location, reason, ElectionPeriod
375 FROM EFICACanelation INNER JOIN ElectionFund ON EFICACanelation.EFID = ElectionFund.EFID
376 INNER JOIN ICAEF ON ICAEF.EFID = ElectionFund.EFID
377 INNER JOIN IslamicConsultativeAssemblyElection ON IslamicConsultativeAssemblyElection.ElectionID = ICAEF.ElectionID
378 SELECT * FROM Evaluation_of_ICAselectionfund
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 105 msec.

-VIEW10-

▪ صندوق های انتخاباتی مجلس خبرگان:

```
379 /*-----REQUIRED VIEWS FOR USERS AS A SUPPERVISOR (AOEselection)---*/
380 CREATE VIEW Evaluation_of_AOEselectionfund AS
381 SELECT EFAOECanelation.EFID, Location, reason, ElectionPeriod
382 FROM EFAOECanelation INNER JOIN ElectionFund ON EFAOECanelation.EFID = ElectionFund.EFID
383 INNER JOIN AOEEF ON AOEEF.EFID = ElectionFund.EFID
384 INNER JOIN AssemblyOfExpertsElection ON AssemblyOfExpertsElection.ElectionID = AOEEF.ElectionID
385 SELECT * FROM Evaluation_of_AOEselectionfund
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 44 msec.

-VIEW11-

▪ صندوق های انتخاباتی شورای شهری روستایی:

```
386 /*-----REQUIRED VIEWS FOR USERS AS A SUPPERVISOR (PSelection)---*/
387 CREATE VIEW Evaluation_of_PSelectionfund AS
388 SELECT EFPSCanelation.EFID, Location, reason, ElectionPeriod
389 FROM EFPSCanelation INNER JOIN ElectionFund ON EFPSCanelation.EFID = ElectionFund.EFID
390 INNER JOIN PSEF ON PSEF.EFID = ElectionFund.EFID
391 INNER JOIN ProvincialSocietiesElection ON ProvincialSocietiesElection.ElectionID = PSEF.ElectionID
392 SELECT * FROM Evaluation_of_PSelectionfund
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 46 msec.

-VIEW12-

- همچنین شورای ناظر باید بتواند ببیند که از تمامی کاندیدا های موجود در هر انتخابات، چه شرایطی دارند تا بررسی صلاحیت رو نظارت کند.
- شرایط نامزدهای انتخاباتی ریاست جمهوری: در ریاست جمهوری یک ویژگی بیشتر نسبت به سایر انتخابات در نظر گرفته شده که متناسبانه برای اعضای ناظر در بررسی شرایط واجدین این صفت مهم است که ببینند و آن هم جنسیت است که خانم نباشد!

```

393 /*-----REQUIRED VIEWS FOR USERS AS A SUPPERVISOR (PElection)---*/
394 CREATE VIEW ALL_PECandidates AS
395 SELECT NationalID,
396     AGE(NOW(), Bdate) as Age,
397     Religen,
398     PoliticalOrientation,
399     Gender,
400     EducationDegree,
401     EducationField
402 FROM Candidate INNER JOIN PEEroll ON Candidate.NationalID = PEEroll.CANID|
403 SELECT * FROM ALL_PECandidates;

```

Data Output Messages Notifications

nationalid [PK] character (10)	age interval	religen character varying (20)	politicalorientation character varying (20)	gender character varying (6)	educationdegree character varying (20)	educationfield character varying (20)

-VIEW13-

از آنجایی که ملاک تنها صفات شرکت کننده ها برای بررسی شرایط است و نه اسم و فامیل و شهرشون تنها ستون های بالا کافی است.

- شرایط نامزدهای انتخاباتی مجلس شورای اسلامی: صفت جنسیت لزوماً لازم نیست دیده شود چون زن و مرد می توانند نماینده باشند.

```

404 /*-----REQUIRED VIEWS FOR USERS AS A SUPPERVISOR (ICAselection)---*/
405 CREATE VIEW ALL_ICACandidates AS
406 SELECT NationalID,
407     AGE(NOW(), Bdate) as Age,
408     Religen,
409     PoliticalOrientation,
410     EducationDegree,
411     EducationField
412 FROM Candidate INNER JOIN ICAEnroll ON Candidate.NationalID = ICAEnroll.CANID;
413 SELECT * FROM ALL_ICACandidates;

```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 140 msec.

-VIEW14-

▪ شرایط نامزدهای انتخاباتی مجلس خبرگان:

```
414 /*-----REQUIRED VIEWS FOR USERS AS A SUPERVISOR (AOEelection)---*/  
415 CREATE VIEW ALL_AOECandidates AS  
416 SELECT NationalID,  
417     AGE(NOW(), Bdate) as Age,  
418     Religen,  
419     PoliticalOrientation,  
420     EducationDegree,  
421     EducationField  
422 FROM Candidate INNER JOIN AOEEnroll ON Candidate.NationalID = AOEEnroll.CANID;  
423 SELECT * FROM ALL_AOECandidates;
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 46 msec.

-VIEW15-

▪ شرایط نامزدهای انتخاباتی شورای شهری روسایی:

```
424 /*-----REQUIRED VIEWS FOR USERS AS A SUPERVISOR (PSelection)---*/  
425 CREATE VIEW ALL_PSCandidates AS  
426 SELECT NationalID,  
427     AGE(NOW(), Bdate) as Age,  
428     Religen,  
429     PoliticalOrientation,  
430     EducationDegree,  
431     EducationField  
432 FROM Candidate INNER JOIN PSEnroll ON Candidate.NationalID = PSEnroll.CANID;  
433 SELECT * FROM ALL_PSCandidates;
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 50 msec.

-VIEW16-

از طرفی چهار نوع کاربر (رأی دهنده، کاندید، ناظر و یا کاربر آزاد) می‌خواهد نتیجه نهایی انتخاباتی هر انتخابات را ببیند.

▪ نتیجه انتخاباتی ریاست جمهوری:

```

427 CREATE VIEW final_peresult AS
428 WITH all_votes AS (
429   SELECT PresidentialElection.ElectionID, PresidentialElection.stage, Candidate.firstname, Candidate.lastname, canid, COUNT(*) AS vote_count
430   FROM PEvote
431   INNER JOIN candidate ON Candidate.NationalID = PEvote.CANID
432   INNER JOIN PEEF ON PEEF.EFID = PEvote.EFID
433   inner join PresidentialElection on PresidentialElection.ElectionID = PEEF.ElectionID
434   GROUP BY PresidentialElection.electionid, PresidentialElection.stage, firstname, lastname, canid
435 ),
436 rejected_votes AS (
437   SELECT PresidentialElection.ElectionID, PresidentialElection.stage, Candidate.firstname, Candidate.lastname, canid, COUNT(*) AS vote_count
438   FROM PEvote
439   INNER JOIN candidate ON Candidate.NationalID = PEvote.CANID
440   INNER JOIN PEEF ON PEEF.EFID = PEvote.EFID
441   inner join PresidentialElection on PresidentialElection.ElectionID = PEEF.ElectionID
442   INNER JOIN EFPECancelation ON EFPECancelation.EFID = PEEF.EFID
443   GROUP BY PresidentialElection.electionid, PresidentialElection.stage, firstname, lastname, canid
444 ),
445 vote_counts AS(
446   SELECT all_votes.ElectionID, all_votes.stage, all_votes.firstname, all_votes.lastname, all_votes.canid, (all_votes.vote_count - COALESCE(rejected_votes.vote_count
447   FROM
448   all_votes LEFT JOIN rejected_votes on (all_votes.ElectionID , all_votes.canid) = (rejected_votes.ElectionID , rejected_votes.canid)
449 ),
450 total_votes AS (
451   SELECT electionid, SUM(vote_count) AS total_count
452   FROM vote_counts
453   GROUP BY electionid
454 ),
455 ranked_candidates AS (
456   SELECT electionid, stage, firstname, lastname, canid,
457   RANK() OVER (PARTITION BY electionid, stage ORDER BY vote_count DESC) AS rank
458   FROM vote_counts
459 )
460 (SELECT vc.electionid, vc.stage, vc.firstname, vc.lastname, vc.canid, vc.vote_count,
461 CASE
462 WHEN vc.vote_count > tv.total_count / 2 THEN 'Winner'
463 WHEN rc.rank = 1 and vc.vote_count < tv.total_count / 2 then 'Runner-up'
464 WHEN rc.rank = 2 THEN
465   CASE
466     WHEN MAX(CASE WHEN rc.rank = 1 THEN vc.vote_count END) OVER (PARTITION BY vc.electionid) > tv.total_count / 2 THEN 'Loser'
467     ELSE 'Runner-up'
468   END
469 ELSE 'Loser'
470 END AS status
471 FROM vote_counts vc
472 JOIN total_votes tv ON vc.electionid = tv.electionid
473 JOIN ranked_candidates rc ON vc.electionid = rc.electionid AND vc.stage = rc.stage AND vc.firstname = rc.firstname AND vc.lastname = rc.lastname AND vc.canid = r
474 ORDER BY vc.electionid, vc.stage, vc.vote_count DESC);
475

```

Data Output Messages Notifications ↗

CREATE VIEW

Query returned successfully in 72 msec.

شما در اینجا وضعیت هر رقابت کنند را در قالب Loser یا winner یا runner-up خواهید دید که رئیس جمهور فرد winner خواهد بود.

نتیجه انتخاباتی مجلس شورای اسلامی:

```
455 /*-----REQUIRED VIEWS FOR ALL USERS (FINAL_ICARESULT)-----*/
456 CREATE VIEW final_ICAResult AS
457 with rejected_vote as (SELECT FirstName, LastName, total_vote, EDName, CANID
458 FROM (
459     SELECT Candidate.FirstName, Candidate.LastName, COUNT(ICAVote.CANID) AS total_vote, ElectoralDistrict.EDName, ICAVote.CANID,
460     ROW_NUMBER() OVER (PARTITION BY ElectoralDistrict.EDName ORDER BY COUNT(ICAVote.CANID) DESC) as row_num
461     FROM Candidate
462     INNER JOIN ICAVote ON Candidate.NationalID = ICAVote.CANID
463     INNER JOIN ICAEnroll ON Candidate.NationalID = ICAEnroll.CANID
464     INNER JOIN ElectoralDistrict ON ICAEnroll.EDID = ElectoralDistrict.EDID
465     INNER JOIN EFICACanelation ON EFICACanelation.EFID = ICAVote.EFID
466     GROUP BY Candidate.NationalID, ElectoralDistrict.EDName, Candidate.FirstName, Candidate.LastName, ICAVote.CANID
467 ) as subquery,
468 allvote as(SELECT ElectionID, FirstName, LastName, total_vote, EDName, CANID
469 FROM (
470     SELECT ICAEnroll.ElectionID, Candidate.FirstName, Candidate.LastName, COUNT(ICAVote.CANID) AS total_vote, ElectoralDistrict.EDName, ICAVote.CANID,
471     ROW_NUMBER() OVER (PARTITION BY ElectoralDistrict.EDName ORDER BY COUNT(ICAVote.CANID) DESC) as row_num
472     FROM Candidate
473     INNER JOIN ICAVote ON Candidate.NationalID = ICAVote.CANID
474     INNER JOIN ICAEnroll ON Candidate.NationalID = ICAEnroll.CANID
475     INNER JOIN ElectoralDistrict ON ICAEnroll.EDID = ElectoralDistrict.EDID
476     inner join IslamicConsultativeAssemblyElection on IslamicConsultativeAssemblyElection.ElectionID = ICAEnroll.ElectionID
477     GROUP BY Candidate.NationalID, ElectoralDistrict.EDName, Candidate.FirstName, Candidate.LastName, ICAVote.CANID, ICAEnroll.ElectionID
478 ) as subquery)
479 ,
480
481 SELECT ElectionPeriod, type, FirstName, LastName, total_vote, EDName, CANID
482 FROM (
483     SELECT IslamicConsultativeAssemblyElection.ElectionPeriod, IslamicConsultativeAssemblyElection.Type , allvote.FirstName, allvote.LastName, (allvote.total_vote
484     ,ROW_NUMBER() OVER (PARTITION BY allvote.EDName ORDER BY (allvote.total_vote - COALESCE(rejected_vote.total_vote, 0)) DESC) as row_num
485     FROM allvote
486     left join rejected_vote ON rejected_vote.CANID = allvote.CANID
487     inner join IslamicConsultativeAssemblyElection on IslamicConsultativeAssemblyElection.ElectionID = allvote.ElectionID
488 ) as subquery
489 WHERE row_num <= 5
490 ORDER BY EDName, row_num;
491
492 select * from final_icareresult ORDER BY EDName, total_vote DESC;
493
--TRIGGER
```

Data Output Messages Notifications

electionperiod	type	firstname	lastname	total_vote	edname	canid
integer	character varying (10)	character varying (10)	character varying (10)	bigint	character varying (10)	character (10)

از آنجا که قطعه کوئری ها کامل در تصویر پوشش داده نمی شود لطفا به فایل سورس کد اصلی مراجعه کنید. در اینجا می توانید لیست نمایندگان منتخب مجلس را مشاهده کنید.

نتیجه انتخاباتی مجلس خبرگان:

```

491 /*-----REQUIRED VIEWS FOR ALL USERS (FINAL_AOERESULT)-----*/
492 CREATE VIEW final_AOEResult AS
493 with rejected_vote as (SELECT FirstName, LastName, total_vote, EDName, CANID
494   FROM (
495     SELECT Candidate.FirstName, Candidate.LastName, COUNT(AOEVote.CANID) AS total_vote, ElectoralDistrict.EDName, AOEVote.CANID,
496     ROW_NUMBER() OVER (PARTITION BY ElectoralDistrict.EDName ORDER BY COUNT(AOEVote.CANID) DESC) as row_num
497     FROM Candidate
498     INNER JOIN AOEVote ON Candidate.NationalID = AOEVote.CANID
499     INNER JOIN AOEEnroll ON Candidate.NationalID = AOEEnroll.CANID
500     INNER JOIN ElectoralDistrict ON AOEEnroll.EDID = ElectoralDistrict.EDID
501     INNER JOIN EFAOECanelation ON EFAOECanelation.EFID = AOEVote.EFID
502     GROUP BY Candidate.NationalID, ElectoralDistrict.EDName, Candidate.FirstName, Candidate.LastName, AOEVote.CANID
503   ) as subquery),
504 allvote as(SELECT ElectionID, FirstName, LastName, total_vote, EDName, CANID
505   FROM (
506     SELECT AOEEnroll.ElectionID, Candidate.FirstName, Candidate.LastName, COUNT(AOEVote.CANID) AS total_vote, ElectoralDistrict.EDName, AOEVote.CANID,
507     ROW_NUMBER() OVER (PARTITION BY ElectoralDistrict.EDName ORDER BY COUNT(AOEVote.CANID) DESC) as row_num
508     FROM Candidate
509     INNER JOIN AOEVote ON Candidate.NationalID = AOEVote.CANID
510     INNER JOIN AOEEnroll ON Candidate.NationalID = AOEEnroll.CANID
511     INNER JOIN ElectoralDistrict ON AOEEnroll.EDID = ElectoralDistrict.EDID
512     inner join AssemblyOfExpertsElection on AssemblyOfExpertsElection.ElectionID = AOEEnroll.ElectionID
513     GROUP BY Candidate.NationalID, ElectoralDistrict.EDName, Candidate.FirstName, Candidate.LastName, AOEVote.CANID,AOEEnroll.ElectionID
514   ) as subquery)
515
516
517 SELECT ElectionPeriod, type, FirstName, LastName, total_vote, EDName, CANID
518   FROM (
519     SELECT AssemblyOfExpertsElection.ElectionPeriod, AssemblyOfExpertsElection.Type , allvote.FirstName, allvote.LastName, (allvote.total_vote - COALESCE(rejected_vote, 0)),
520     ,ROW_NUMBER() OVER (PARTITION BY allvote.EDName ORDER BY (allvote.total_vote - COALESCE(rejected_vote.total_vote, 0)) DESC) as row_num
521   FROM allvote
522   left join rejected_vote ON rejected_vote.CANID = allvote.CANID
523   inner join AssemblyOfExpertsElection on AssemblyOfExpertsElection.ElectionID = allvote.ElectionID
524 ) as subquery
525 WHERE row_num <= 5
526 ORDER BY EDname, row_num;
527
528 select * from final_AOEResult ORDER BY EDName, total_vote DESC;
529 TRIGGER

```

Data Output Messages Notifications

electionperiod	type	firstname	lastname	total_vote	edname	canid
integer	character varying (10)	character varying (10)	character varying (10)	bigint	character varying (10)	character (10)

نتیجه انتخاباتی شورای شهری روستایی:

```

527 /*-----REQUIRED VIEWS FOR ALL USERS (FINAL_PSRESULT)-----*/
528 CREATE VIEW final_PSResult AS
529 with rejected_vote as (SELECT FirstName, LastName, total_vote, EDName, CANID
530 FROM (
531     SELECT Candidate.FirstName, Candidate.LastName, COUNT(PSVote.CANID) AS total_vote, ElectoralDistrict.EDName, PSVote.CANID,
532     ROW_NUMBER() OVER (PARTITION BY ElectoralDistrict.EDName ORDER BY COUNT(PSVote.CANID) DESC) as row_num
533     FROM Candidate
534     INNER JOIN PSVote ON Candidate.NationalID = PSVote.CANID
535     INNER JOIN PSEnroll ON Candidate.NationalID = PSEnroll.CANID
536     INNER JOIN ElectoralDistrict ON PSEnroll.EDID = ElectoralDistrict.EDID
537     INNER JOIN EFPSCancelation ON EFPSCancelation.EFID = PSVote.EFID
538     GROUP BY Candidate.NationalID, ElectoralDistrict.EDName, Candidate.FirstName, Candidate.LastName, PSVote.CANID
539 ) as subquery),
540 allvote as(SELECT ElectionID, FirstName, LastName, total_vote, EDName, CANID
541 FROM (
542     SELECT PSEnroll.ElectionID, Candidate.FirstName, Candidate.LastName, COUNT(PSVote.CANID) AS total_vote, ElectoralDistrict.EDName, PSVote.CANID,
543     ROW_NUMBER() OVER (PARTITION BY ElectoralDistrict.EDName ORDER BY COUNT(PSVote.CANID) DESC) as row_num
544     FROM Candidate
545     INNER JOIN PSVote ON Candidate.NationalID = PSVote.CANID
546     INNER JOIN PSEnroll ON Candidate.NationalID = PSEnroll.CANID
547     INNER JOIN ElectoralDistrict ON PSEnroll.EDID = ElectoralDistrict.EDID
548     inner join ProvincialSocietiesElection on ProvincialSocietiesElection.ElectionID = PSEnroll.ElectionID
549     GROUP BY Candidate.NationalID, ElectoralDistrict.EDName, Candidate.FirstName, Candidate.LastName, PSVote.CANID,PSEnroll.ElectionID
550 ) as subquery)
551 |
552
553 SELECT ElectionPeriod, type, FirstName, LastName, total_vote, EDName, CANID
554 FROM (
555     SELECT ProvincialSocietiesElection.ElectionPeriod, ProvincialSocietiesElection.Type , allvote.FirstName, allvote.LastName, (allvote.total_vote - COALESCE(rejected_vote.total_vote, 0))
556     ,ROW_NUMBER() OVER (PARTITION BY allvote.EDName ORDER BY (allvote.total_vote - COALESCE(rejected_vote.total_vote, 0)) DESC) as row_num
557     FROM allvote
558     left join rejected_vote ON rejected_vote.CANID = allvote.CANID
559     inner join ProvincialSocietiesElection on ProvincialSocietiesElection.ElectionID = allvote.ElectionID
560 ) as subquery
561 WHERE row_num <= 5
562 ORDER BY EDName, row_num;
563
564
565 select * from final_PSResult ORDER BY EDName, total_vote DESC;
566

```

Data Output Messages Notifications

electionperiod	type	firstname	lastname	total_vote	edname	canid
integer	character varying (10)	character varying (10)	character varying (10)	bigint	character varying (10)	character (10)

پیاده سازی محدودیت های معنایی

```
587 --TRIGGER-----
588
589 CREATE FUNCTION public.check_age() RETURNS trigger
590     LANGUAGE plpgsql
591     AS $$
592 BEGIN
593     IF (NEW.bdate > CURRENT_DATE - INTERVAL '18 years') THEN
594         RAISE EXCEPTION 'Must be at least 18 years old.';
595     END IF;
596     RETURN NEW;
597 END;
598 $$;
599 CREATE TRIGGER check_age
600 BEFORE INSERT OR UPDATE ON public.voter
601 FOR EACH ROW EXECUTE FUNCTION public.check_age();
602 -----
```

این سینتکس یک تابع تریگر است که در PostgreSQL تعریف شده است. وظیفه‌ی این تابع تریگر، بررسی سن یک فرد در زمان اضافه کردن یا به روز رسانی اطلاعات در دیتابیس است.

به عنوان مثال، اگر شما یک فیلد سن داشته باشید که مقدار سن را به صورت عددی نمایش میدهد، و میخواهید بررسی کنید که آیا سن وارد شده نسبت به زمان لحظه ثبت نام برای این فرد بالای 18 سال است یا خیر، میتوانید از این تابع تریگر استفاده کنید. با استفاده از این تابع تریگر، میتوانید برای هر اضافه کردن یا به روز رسانی در دیتابیس، بررسی کنید که آیا سن وارد شده برای فرد مربوطه بیشتر از 18 سال است یا خیر، و در صورتی که سن کمتر از 18 سال باشد، عملیات اضافه کردن یا به روز رسانی را به صورت خودکار متوقف کنید.

```
602 -----
603 CREATE FUNCTION public.check_age_candidate() RETURNS trigger
604     LANGUAGE plpgsql
605     AS $$
606 BEGIN
607     IF (NEW.bdate > CURRENT_DATE - INTERVAL '25 years') THEN
608         RAISE EXCEPTION 'Must be at least 25 years old.';
609     END IF;
610     RETURN NEW;
611 END;
612 $$;
613 CREATE TRIGGER check_age_can
614 BEFORE INSERT OR UPDATE ON public.candidate
615 FOR EACH ROW EXECUTE FUNCTION public.check_age_candidate();
616 -----
```

این تابع برای اعتبار سنی سن نامزد های انتخاباتی است. تابع زمانی که یک رکورد جدید به جدول Candidate اضافه می شود وارد عمل میشود. عملکردش به ای صورت است که اگر تاریخ تولد کاندید جدید نسبت به زمان همان لحظه ثبت نام کمتر از 25 سال باشد، یک خطا (exception) رخ می دهد و پایگاه داده اجازه ثبت این نامزد جدید را نمی دهد.

به طور کلی استفاده این تابع به منظور جلوگیری از ثبت نامزدهایی که سنشان کمتر از حد مجاز است و داشتن اطمینان از اینکه همه نامزد ها حداقل 25 سال سن دارند مفید است.

```

613
614 CREATE FUNCTION public.check_agefromelectiondate() RETURNS trigger
615   LANGUAGE plpgsql
616   AS $$
617▼ BEGIN
618▼   IF (NEW.bdate + INTERVAL '18 years') > (SELECT electiondate FROM election ORDER BY Electiondate DESC LIMIT 1) THEN
619     RAISE EXCEPTION 'The person must be older than 18.';
620   END IF;
621   RETURN NEW;
622 END;
623 $$;
624 CREATE TRIGGER check_age_trigger_
625 AFTER INSERT OR UPDATE ON public.voter
626 FOR EACH ROW EXECUTE FUNCTION public.check_agefromelectiondate();
627

```

این محدودیت هم برای سنین رای دهنده است اما با این تفاوت که بررسی می کند سن رای دهنده نسبت به تاریخ برگزاری انتخابات نیز بیشتر از 18 باشد.

```

627
628 CREATE FUNCTION public.check_agefromelectiondatecan() RETURNS trigger
629   LANGUAGE plpgsql
630   AS $$
631▼ BEGIN
632▼   IF (NEW.bdate + INTERVAL '25 years') > (SELECT electiondate FROM election ORDER BY Electiondate DESC LIMIT 1) THEN
633     RAISE EXCEPTION 'The person must be older than 25.';
634   END IF;
635   RETURN NEW;
636 END;
637 $$;
638 CREATE TRIGGER check_age_trigger
639 AFTER INSERT OR UPDATE ON public.candidate
640 FOR EACH ROW EXECUTE FUNCTION public.check_agefromelectiondatecan();
641

```

این محدودیت هم برای سینم کاندیدهای انتخاباتی است اما با این تفاوت که بررسی می کند سن کاندید نسبت به تاریخ برگزاری انتخابات نیز بیشتر از 25 باشد.

```

644 ----
645 CREATE FUNCTION public.check_city() RETURNS trigger
646   LANGUAGE plpgsql
647   AS $$
648 BEGIN
649   allowed_cities TEXT[] := ARRAY['Tehran', 'Mashhad', 'Isfahan', 'Karaj', 'Shiraz', 'Tabriz', 'Qom', 'Ahvaz', 'Kermanshah', 'Urmia', 'Rasht', 'Zahedan', 'Hamadan'];
650 END;
651 IF NEW.city <> ALL (allowed_cities) THEN
652   RAISE EXCEPTION 'City must be one of %', allowed_cities;
653 END IF;
654 RETURN NEW;
655 END;
656 $$;
657 CREATE TRIGGER check_city_trigger
658 BEFORE INSERT OR UPDATE ON public.candidate
659 FOR EACH ROW EXECUTE FUNCTION public.check_city();
660 ----

```

این محدودیت بررسی می کند که در صورتی که شهر وارد شده توسط کاربر، در لیست شهر های معتبر تابع نباشد با خطا رو به رو می شود؛ به این صورت که ابتدا یک آرایه از شهر های معتبر در نظر گرفته شده است و در متغیر `allowed_cities` ذخیره شده است. سپس با استفاده از دستور `if`، شهر وارد شده در مقایسه با تمامی عناصر این آرایه قرار می گیرد و در صورتی مه در لیست معتبر نباشد، یک خطا با استفاده از دستور `raise exception` ایجاد می شود. در نهایت تابع با بازگرداندن `new` اجازه می دهد تا عملیات افزودن یا به روزرسانی در جدول انجام شود.

```

657 ----
658 CREATE FUNCTION public.check_duplicate_vote_PS() RETURNS trigger
659   LANGUAGE plpgsql
660   AS $$
661 BEGIN
662 IF EXISTS (SELECT 1 FROM PSVote WHERE void = NEW void AND canid = NEW.canid AND efid=new.efid) THEN
663   RAISE EXCEPTION 'Duplicate vote not allowed';
664 END IF;
665 RETURN NEW;
666 END;
667 $$;
668 CREATE TRIGGER check_duplicate_vote_trigger_PS
669 BEFORE INSERT ON PSvote
670 FOR EACH ROW EXECUTE PROCEDURE check_duplicate_vote_PS();
671 ----

```

این محدودیت برای جلوگیری از ثبت رای های تکراری است به این صورت که اگر رای دهنده ای مثلاً یک بار به یک نامزد انتخاباتی رای داده باشه و بخواهد دوباره رای بدء خطای مشاهده خواهیم کرد که فرد مجاز به این کار نیست. این محدودیت برای هر 4 انتخابات تعبیه شده و در تصاویر 3 تا مورد پایین می توانید مشاهده کنید:

```

671 -----
672 CREATE FUNCTION public.check_duplicate_vote_ICA() RETURNS trigger
673   LANGUAGE plpgsql
674   AS $$
675 BEGIN
676 IF EXISTS (SELECT 1 FROM ICAVote WHERE void = NEW void AND canid = NEW.canid AND efid=new.efid) THEN
677   RAISE EXCEPTION 'Duplicate vote not allowed';
678 END IF;
679 RETURN NEW;
680 END;
681 $$;
682 CREATE TRIGGER check_duplicate_vote_trigger_ICA
683 BEFORE INSERT ON icavote
684 FOR EACH ROW EXECUTE PROCEDURE check_duplicate_vote_ICA();
685 -----
686 -----
687 -----
688 -----
689 BEGIN
690 IF EXISTS (SELECT 1 FROM AOEVote WHERE void = NEW void AND canid = NEW.canid AND efid=new.efid) THEN
691   RAISE EXCEPTION 'Duplicate vote not allowed';
692 END IF;
693 RETURN NEW;
694 END;
695 $$;
696 CREATE TRIGGER check_duplicate_vote_trigger_AOE
697 BEFORE INSERT ON AOEvote
698 FOR EACH ROW EXECUTE PROCEDURE check_duplicate_vote_AOE();
699 -----
700 -----
701 -----
702 -----
703 BEGIN
704 IF EXISTS (SELECT 1 FROM PEVote WHERE void = NEW void AND canid = NEW.canid AND efid=new.efid) THEN
705   RAISE EXCEPTION 'Duplicate vote not allowed';
706 END IF;
707 RETURN NEW;
708 END;
709 $$;
710 CREATE TRIGGER check_duplicate_vote_trigger_PE
711 BEFORE INSERT ON PEvote
712 FOR EACH ROW EXECUTE PROCEDURE check_duplicate_vote_PE();
713 -----

```

```

713 -----
714 CREATE FUNCTION public.check_name_length() RETURNS trigger
715   LANGUAGE plpgsql
716   AS $$
717 BEGIN
718 IF (char_length(NEW.firstname) < 3 OR char_length(NEW.lastname) < 3) THEN
719   RAISE EXCEPTION 'Name must be longer than 3 characters.';
720 END IF;
721 RETURN NEW;
722 END;
723 $$;
724 CREATE TRIGGER check_name_length
725 BEFORE INSERT OR UPDATE ON public.candidate
726 FOR EACH ROW EXECUTE FUNCTION public.check_name_length();
727
728 CREATE TRIGGER check_name_length_
729 BEFORE INSERT OR UPDATE ON public.voter
730 FOR EACH ROW EXECUTE FUNCTION public.check_name_length();
731 -----

```

در این محدودیتی که بر روی صفت اسم پیاده سازی کردیم، اسم نمی تواند طولی کمتر از 3 کارکتر داشته باشد و در غیر این صورت با خطأ مواجه خواهد شد:

“Name must be longer than 3 characters.”

```

731 -----
732 CREATE FUNCTION public.check_value_PS() RETURNS trigger
733   LANGUAGE plpgsql
734   AS $$
735 DECLARE
736   value_exists BOOLEAN;
737 BEGIN
738   SELECT EXISTS (SELECT 1 FROM PSValidation WHERE SUID = NEW.SUID AND canid=new.canid) INTO value_exists;
739 IF (value_exists) THEN
740   RAISE EXCEPTION 'You have already qualified this candidate.';
741 END IF;
742 RETURN NEW;
743 END;
744 $$;
745 CREATE TRIGGER check_value_PS
746 BEFORE INSERT OR UPDATE ON public.PSValidation
747 FOR EACH ROW EXECUTE FUNCTION public.check_value_PS();
748 -----

```

این محدودیت داره میگه که هر انتخابات تنها می تواند یک شورای ناظر داشته باشد. برای بقیه انتخابات هم در 3 تصویر پایین این محدودیت را اعمال کردیم.

```

748 -----
749 CREATE FUNCTION public.check_value_ICA() RETURNS trigger
750   LANGUAGE plpgsql
751   AS $$
752 DECLARE
753   value_exists BOOLEAN;
754 BEGIN
755   SELECT EXISTS (SELECT 1 FROM ICAValidation WHERE SUID = NEW.SUID AND canid=new.canid) INTO value_exists;
756   IF ( value_exists) THEN
757     RAISE EXCEPTION 'You have already qualified this candidate';
758   END IF;
759   RETURN NEW;
760 END;
761 $$;
762 CREATE TRIGGER check_value_ICA
763 BEFORE INSERT OR UPDATE ON public.ICAValidation
764 FOR EACH ROW EXECUTE FUNCTION public.check_value_ICA();
765 -----
766 -----
767 CREATE FUNCTION public.check_value_PE() RETURNS trigger
768   LANGUAGE plpgsql
769   AS $$
770 DECLARE
771   value_exists BOOLEAN;
772 BEGIN
773   SELECT EXISTS (SELECT 1 FROM PEValidation WHERE SUID = NEW.SUID AND canid=new.canid) INTO value_exists;
774   IF ( value_exists) THEN
775     RAISE EXCEPTION 'You have already qualified this candidate.';
776   END IF;
777   RETURN NEW;
778 END;
779 $$;
780 CREATE TRIGGER check_value_PE
781 BEFORE INSERT OR UPDATE ON public.PEvaluation
782 FOR EACH ROW EXECUTE FUNCTION public.check_value_PE();
783 -----
784 CREATE FUNCTION public.check_value_AOE() RETURNS trigger
785   LANGUAGE plpgsql
786   AS $$
787 DECLARE
788   value_exists BOOLEAN;
789 BEGIN
790   SELECT EXISTS (SELECT 1 FROM AOEValidation WHERE SUID = NEW.SUID AND canid=new.canid) INTO value_exists;
791   IF ( value_exists) THEN
792     RAISE EXCEPTION 'You have already qualified this candidate.';
793   END IF;
794   RETURN NEW;
795 END;
796 $$;
797 CREATE TRIGGER check_value_AOE
798 BEFORE INSERT OR UPDATE ON public.AOEValidation
799 FOR EACH ROW EXECUTE FUNCTION public.check_value_AOE();

```

```

799 -----
800 CREATE FUNCTION public.max_votes_PS() RETURNS trigger
801   LANGUAGE plpgsql
802   AS $$
803 BEGIN
804 IF (SELECT COUNT(*) FROM PSVote WHERE void = NEW.VOID AND efid=new.efid) > 5 THEN
805   RAISE EXCEPTION 'A voter can vote for a maximum of five candidates.';
806 END IF;
807 RETURN NEW;
808 END;
809 $$;
810 CREATE TRIGGER max_votes_trigger_PS
811 BEFORE INSERT ON public.PSVote
812 FOR EACH ROW EXECUTE FUNCTION public.max_votes_PS();
813 -----

```

در این محدودیت اعمال شده هر فرد می تواند در انتخابات مذکور به حداقل 5 نفر رای بدهد البته این محدودیت رای در انتخابات ریاست جمهوری به 1 نفر حق رای کاهش پیدا می کند. سه شکل زیر برای سایر این انتخابات این محدودیت را اعمال می کند.

```

813 -----
814 CREATE FUNCTION public.max_votes_AOE() RETURNS trigger
815   LANGUAGE plpgsql
816   AS $$
817 BEGIN
818 IF (SELECT COUNT(*) FROM AOEVote WHERE void = NEW.VOID AND efid=new.efid) > 5 THEN
819   RAISE EXCEPTION 'A voter can vote for a maximum of five candidates.';
820 END IF;
821 RETURN NEW;
822 END;
823 $$;
824 CREATE TRIGGER max_votes_trigger_AOE
825 BEFORE INSERT ON public.AOEvote
826 FOR EACH ROW EXECUTE FUNCTION public.max_votes_AOE();
827 -----
828 CREATE FUNCTION public.max_votes_ICA() RETURNS trigger
829   LANGUAGE plpgsql
830   AS $$
831 BEGIN
832 IF (SELECT COUNT(*) FROM ICAVote WHERE void = NEW.VOID AND efid=new.efid) > 5 THEN
833   RAISE EXCEPTION 'A voter can vote for a maximum of five candidates.';
834 END IF;
835 RETURN NEW;
836 END;
837 $$;
838 CREATE TRIGGER max_votes_trigger_ICA
839 BEFORE INSERT ON public.ICAvote
840 FOR EACH ROW EXECUTE FUNCTION public.max_votes_ICA();
841 -----

```

```

841 -----
842 CREATE FUNCTION public.max_votespe_PE() RETURNS trigger
843   LANGUAGE plpgsql
844   AS $$
845 BEGIN
846   IF (SELECT COUNT(*) FROM PEVote WHERE void = NEW.VOID AND efid=new.efid) > 1 THEN
847     RAISE EXCEPTION 'A voter can vote for a maximum of one candidate.';
848   END IF;
849   RETURN NEW;
850 END;
851 $$;
852 CREATE TRIGGER max_votes_trigger_PE
853 BEFORE INSERT ON public.PEvote
854 FOR EACH ROW EXECUTE FUNCTION public.max_votespe_PE();

```

حال در ادامه ما برای هر انتخابات یعنی به تعداد 4 temporary tables ایجاد می کنیم که لیست کاندیدا هایی که به آن ها رای داده میشند را با لیست کاندیدا هایی که وضعیت صلاحیتشون با صفت status مشخص شده، JOIN می دهد. در ادامه بهشون برای تعریف TRIGGER نیاز داریم.

```

634 ---TEMPORARY TABLES WHICH ARE NEEDED FOR OUR TRIGGER ---
635 CREATE TEMPORARY TABLE ica_vote_qual AS
636 SELECT *
637 FROM icavote
638 NATURAL JOIN icavalidation;
639 -----
640 CREATE TEMPORARY TABLE aoe_vote_qual AS
641 SELECT *
642 FROM aoevote
643 NATURAL JOIN aoevalidation;
644 -----
645 CREATE TEMPORARY TABLE pe_vote_qual AS
646 SELECT *
647 FROM pevote
648 NATURAL JOIN pevalidation;
649 -----
650 CREATE TEMPORARY TABLE ps_vote_qual AS
651 SELECT *
652 FROM psvote
653 NATURAL JOIN pevalidation;
654 -----
655
Data Output Messages Notifications
SELECT 0
Query returned successfully in 45 msec.

```

حال بجای اینکه یکسری سینتکس رو در تمام TRIGGER چهارتا انتخابات تکرار کنیم یک تابع جدأگونه برای عملکرد مشابه TRIGGER ها ایجاد می کنیم تا از اون استفاده کنیم:

```

654 -----REQUIRED FUNCTION TO USE IN TRIGGERS-----
655 CREATE OR REPLACE FUNCTION checkqualified()
656 RETURNS TRIGGER AS $$ 
657 BEGIN
658     IF new.status='disqualified' THEN
659         RAISE EXCEPTION 'You cannot vote for a disqualified candidate';
660     END IF;
661     RETURN NEW;
662 END;
663 $$ LANGUAGE plpgsql;
664

```

Data Output Messages Notifications

CREATE FUNCTION

Query returned successfully in 107 msec.

در حقیقت این تابع عملیات اصلی محدودیتی ما رو در TRIGGER ای که در شکل زیر مشاهده خواهید کرد پوشش می دهد بطوریه رای دهنده ها تنها می توانند به آن هایی رای بدهند که از جانب سورای ناظر تأیید صلاحیت شدند یا به اصطلاح Status = qualified است. در این تابع اگر مشاهده کند کاندید منتخب برای رای گیری disqualified است، خطایی را مشاهده خواهد کرد که نمی تواند به آن رای دهد.

```

665 -----TRIGGERS FOR VOTING TO QULIFIED CANDIDATES-----
666 -----ICE-----
667 CREATE TRIGGER check_qualified
668 BEFORE INSERT ON ica_vote_qual
669 FOR EACH ROW EXECUTE PROCEDURE checkqualified();
670 -----AOE-----
671 CREATE TRIGGER check_qualified
672 BEFORE INSERT ON aoe_vote_qual
673 FOR EACH ROW EXECUTE PROCEDURE checkqualified();
674 -----PE-----
675 CREATE TRIGGER check_qualified
676 BEFORE INSERT ON pe_vote_qual
677 FOR EACH ROW EXECUTE PROCEDURE checkqualified();
678 -----PS-----
679 CREATE TRIGGER check_qualified
680 BEFORE INSERT ON ps_vote_qual
681 FOR EACH ROW EXECUTE PROCEDURE checkqualified();
682

```

Data Output Messages Notifications

CREATE TRIGGER

Query returned successfully in 75 msec.

پس این محدودیت ها ناظر بر آن است که حتما به کاندید واحد شرایط رای داده شود.

این محدودیت هایی که در پایین برای هر انتخابات تعریف شده برای اعمال این شرط است که یک صندوق انتخاباتی تنها می تواند مختص یک انتخابات باشد. به عنوان مثال صندوق 2631 تنها میتواند رای های انتخاباتی ریاست جمهوری را نگهدارد نه انتخابات مجلس یا ... و رای ها قاطی نمی شود.

- انتخابات ریاست جمهوری:

```
827 -----EACH ELECTION FUND MUST BE FOR JUST ONE ELECTION NOT BOTH OR MORE-----
828 -----PS-----
829 CREATE OR REPLACE FUNCTION check_efid_electionid()
830 RETURNS TRIGGER AS $$ 
831 DECLARE
832     efid_count INTEGER;
833 BEGIN
834     SELECT COUNT(DISTINCT electionid) INTO efid_count
835     FROM peef
836     WHERE efid = NEW.efid;
837
838 IF efid_count > 1 THEN
839     RAISE EXCEPTION 'efid % matches with more than one electionid', NEW.efid;
840 END IF;
841
842 RETURN NEW;
843 END;
844 $$ LANGUAGE plpgsql;
845
846 CREATE TRIGGER check_efid_electionid_trigger
847 BEFORE INSERT OR UPDATE ON peef
848 FOR EACH ROW EXECUTE FUNCTION check_efid_electionid();
849 -----
```

Data Output Messages Notifications

CREATE TRIGGER

Query returned successfully in 51 msec.

- انتخابات مجلس شورای اسلامی:

```
849 -----ICA-----
850 CREATE FUNCTION check_efid_electionidica()
851 RETURNS TRIGGER AS $$ 
852 DECLARE
853     efid_count INTEGER;
854 BEGIN
855     SELECT COUNT(DISTINCT electionid) INTO efid_count
856     FROM icaef
857     WHERE efid = NEW.efid;
858
859 IF efid_count > 1 THEN
860     RAISE EXCEPTION 'efid % matches with more than one electionid', NEW.efid;
861 END IF;
862
863 RETURN NEW;
864 END;
865 $$ LANGUAGE plpgsql;
866
867 CREATE TRIGGER check_efid_electionid_trigger
868 BEFORE INSERT OR UPDATE ON icaef
869 FOR EACH ROW EXECUTE FUNCTION check_efid_electionidica();
870 -----
```

Data Output Messages Notifications

CREATE TRIGGER

Query returned successfully in 57 msec.

انتخابات مجلس خبرگان:

```
870 -----AOE-----
871 CREATE FUNCTION check_efid_electionidae()
872 RETURNS TRIGGER AS $$ 
873 DECLARE
874     efid_count INTEGER;
875 ▼ BEGIN
876     SELECT COUNT(DISTINCT electionid) INTO efid_count
877     FROM aoef
878     WHERE efid = NEW.efid;
879
880 ▼ IF efid_count > 1 THEN
881     RAISE EXCEPTION 'efid % matches with more than one electionid', NEW.efid;
882 END IF;
883
884     RETURN NEW;
885 END;
886 $$ LANGUAGE plpgsql;
887
888 CREATE TRIGGER check_efid_electionid_trigger
889 BEFORE INSERT OR UPDATE ON aoef
890 FOR EACH ROW EXECUTE FUNCTION check_efid_electionidae();
891
```

Data Output Messages Notifications

CREATE TRIGGER

Query returned successfully in 54 msec.

انتخابات شورای شهر و روستا:

```
891 -----PS-----
892 CREATE FUNCTION check_efid_electionidps()
893 RETURNS TRIGGER AS $$ 
894 DECLARE
895     efid_count INTEGER;
896 ▼ BEGIN
897     SELECT COUNT(DISTINCT electionid) INTO efid_count
898     FROM PSEF
899     WHERE efid = NEW.efid;
900
901 ▼ IF efid_count > 1 THEN
902     RAISE EXCEPTION 'efid % matches with more than one electionid', NEW.efid;
903 END IF;
904
905     RETURN NEW;
906 END;
907 $$ LANGUAGE plpgsql;
908
909 CREATE TRIGGER check_efid_electionid_trigger
910 BEFORE INSERT OR UPDATE ON PSEF
911 FOR EACH ROW EXECUTE FUNCTION check_efid_electionidps();
912
```

Data Output Messages Notifications

CREATE TRIGGER

Query returned successfully in 54 msec.

پیاده سازی index ها

وجود ایندکس ها بر روی یکسری صفات سرعت و کارآمدی دیتابیس را در هنگام سرچ و اجرای کوئری ها بالا میبرد و به جای چک کردن تمام میلیون ها رکورد موجود، به دنبال نود ها با مشخصه خواسته شده در کوئری میگیرد و به عنوان مثال دسته افراد به نام "علی" را به سرعت به کمک ساختار B-Tree پیدا میکنه.

اما توجه باید به این موضوع هم کرد که صرف اینکه ایندکس تعریف کردن سرعت را بالا میبره، تعداد زیاد اون ها میتوانه به نوعی دیگه تأثیر منفی را کارآمدی سیستم داشته باشه. مثلا هر بار که شما یک جدول را آپدیت می کنید تمامی ایندکس های مرتب با آن جدول نیز آپدیت باید بشوند که برای سیستم هندل کردن تعداد زیادی ایندکس ناکارآمد خواهد بود. ما سعی کردیم ایندکس هایی را تعریف کنیم که حس می کنیم کاربر بیشتر با آن ها در اجرای کوئری ها یا سرچ کردن ها سرکار دارد.

```
964  create index Voter_Firstname_idx  
965  on Voter (FirstName);  
966  create index Voter_LastName_idx  
967  on Voter (LastName);  
968  create index Voter_Bdate_idx  
969  on Voter (Bdate);  
970  create index Candidate_FirstName_idx  
971  on Candidate (FirstName);  
972  create index Candidate_LastName_idx  
973  on Candidate (LastName);  
974  create index Candidate_Bdate_idx  
975  on Candidate (Bdate);  
976  create index ElectoralDistrict_name_idx  
977  on ElectoralDistrict (EDName);  
978  create index ElectionFund_name_idx  
979  on ElectionFund (Location);  
980  create index PE_Stage_idx  
981  on PresidentialElection (Stage);  
982  create index PE_Type_idx  
983  on PresidentialElection (Type);  
984  create index ICA_Type_idx  
985  on IslamicConsultativeAssemblyElection (Type);  
986  create index PS_Type_idx  
987  on ProvincialSocietiesElection (Type);  
988  create index AOE_Type_idx  
989  on AssemblyOfExpertsElection (Type);|  
990
```

Data Output Messages Notifications

CREATE INDEX

تست دیتابیس

ورود دیتاست جدول رای هنده

Query Query History

```
993 ----DATABASE TESTING-----  
994 ----INSERTING VALUES IN TABLES-----  
995 INSERT INTO voter VALUES  
996 ('0000000001' , 'Ali' , 'Mohammadi' , '1962-02-02' , 'Male'),  
997 ('0000000002' , 'Ali' , 'Davoodi' , '1973-02-05' , 'Male'),  
998 ('0000000003' , 'Hassan' , 'Mohammadi' , '1955-03-06' , 'Male'),  
999 ('0000000004' , 'Hassan' , 'Akbari' , '1966-04-08' , 'Male'),  
1000 ('0000000005' , 'Hossein' , 'Mohammadi' , '1987-05-10' , 'Female'),  
1001 ('0000000006' , 'Zahra' , 'Mohammadi' , '1998-01-21' , 'Female'),  
1002 ('0000000007' , 'Fatemeh' , 'Mohammadi' , '1999-07-22' , 'Female'),  
1003 ('0000000008' , 'Sara' , 'Davoodi' , '1976-08-12' , 'Female'),  
1004 ('0000000009' , 'Zahra' , 'Davoodi' , '1990-08-07' , 'Female'),  
1005 ('0000000010' , 'Zohreh' , 'Vahidi' , '2001-02-09' , 'Female'),  
1006 ('0000000011' , 'Mohammad' , 'Mohammadi' , '2003-12-20' , 'Male'),  
1007 ('0000000012' , 'Javad' , 'Javadi' , '2002-03-27' , 'Male'),  
1008 ('0000000013' , 'Asghar' , 'Davoodi' , '1999-02-21' , 'Male'),  
1009 ('0000000014' , 'Nima' , 'Mohammadi' , '1983-11-21' , 'Male'),  
1010 ('0000000015' , 'Melika' , 'Sadeghi' , '2002-10-06' , 'Female'));  
1011 SELECT * FROM Voter;  
1012
```

Data Output Messages Notifications

	nationalid [PK] character (10)	firstname character varying (10)	lastname character varying (10)	bdate date	gender character varying (6)
1	0000000001	Ali	Mohammadi	1962-02-02	Male
2	0000000002	Ali	Davoodi	1973-02-05	Male
3	0000000003	Hassan	Mohammadi	1955-03-06	Male
4	0000000004	Hassan	Akbari	1966-04-08	Male
5	0000000005	Hossein	Mohammadi	1987-05-10	Female

Total rows: 15 of 15 Query complete 00:00:00.056

ورود دیتاست کاندید:

Query Query History

```
1012 INSERT INTO candidate VALUES
1013 ('0100000001', 'Ali', 'Mohammadi', '1962-02-02', 'Islam', 'Tehran', 'Osol gara', 'Male', 'Bachelor Degree', 'Law'),
1014 ('0100000002', 'Ali', 'Davoodi', '1973-02-05', 'Islam', 'Tehran', 'Osol gara', 'Male', 'Master Degree', 'Law'),
1015 ('0100000003', 'Hassan', 'Mohammadi', '1955-03-06', 'Islam', 'Tehran', 'Osol gara', 'Male', 'Bachelor Degree', 'Law'),
1016 ('0100000004', 'Hassan', 'Akbari', '1966-04-08', 'Islam', 'Tehran', 'Osol gara', 'Male', 'Master Degree', 'Law'),
1017 ('0100000005', 'Hosseini', 'Mohammadi', '1987-05-10', 'Islam', 'Tehran', 'Osol gara', 'Female', 'Master Degree', 'Law'),
1018 ('0100000006', 'Zahra', 'Mohammadi', '1990-01-21', 'Christianity', 'Tehran', 'Aghaliat mazhabi', 'Female', 'Bachelor Degree', 'Law'),
1019 ('0100000007', 'Fatemeh', 'Mohammadi', '1990-01-21', 'Christianity', 'Tehran', 'Aghaliat mazhabi', 'Female', 'Master Degree', 'Law'),
1020 ('0100000008', 'Sara', 'Davoodi', '1976-08-12', 'Islam', 'Mashhad', 'Osol gara', 'Female', 'Doctoral Degree', 'economics'),
1021 ('0100000009', 'Zahra', 'Davoodi', '1990-08-07', 'Islam', 'Mashhad', 'Osol gara', 'Female', 'Master Degree', 'economics'),
1022 ('0100000010', 'Zohreh', 'Vahidi', '1992-02-09', 'Islam', 'Mashhad', 'Osol gara', 'Female', 'Master Degree', 'economics'),
1023 ('0100000011', 'Mohammad', 'Mohammadi', '1990-12-20', 'Islam', 'Mashhad', 'Osol gara', 'Male', 'Bachelor Degree', 'economics'),
1024 ('0100000012', 'Javad', 'Javadi', '1992-03-27', 'Islam', 'Mashhad', 'Osol gara', 'Male', 'Bachelor Degree', 'Political'),
1025 ('0100000013', 'Asghar', 'Davoodi', '1950-02-21', 'Islam', 'Mashhad', 'Eslah Talab', 'Male', 'Master Degree', 'Political'),
1026 ('0100000014', 'Nima', 'Mohammadi', '1983-11-21', 'Islam', 'Isfahan', 'Eslah Talab', 'Male', 'Bachelor Degree', 'Political'),
1027 ('0100000015', 'Melika', 'Sadeghi', '1960-10-06', 'Islam', 'Isfahan', 'Eslah Talab', 'Female', 'Master Degree', 'Political'),
1028 ('0100000016', 'Sorosh', 'Azizi', '1962-02-02', 'Islam', 'Isfahan', 'Eslah Talab', 'Male', 'Doctoral Degree', 'Political'),
1029 ('0100000017', 'Ali', 'Motahatri', '1973-02-05', 'Islam', 'Isfahan', 'Eslah Talab', 'Male', 'Master Degree', 'History'),
1030 ('0100000018', 'Mahmood', 'Ahmadi', '1955-03-06', 'Islam', 'Isfahan', 'Eslah Talab', 'Male', 'Doctoral Degree', 'History'),
1031 ('0100000019', 'Hassan', 'Ghahreman', '1966-04-08', 'Islam', 'Isfahan', 'Eslah Talab', 'Male', 'Master Degree', 'History')

Data Output    Messages    Notifications
```

	nationalid [PK] character (10)	firstname character varying (10)	lastname character varying (10)	bdate date	religen character varying (20)	city character varying (10)	politicalorientation character varying (20)	gender character varying (6)	educationdegree character varying (20)	educationfield character varying
1	0100000001	Ali	Mohammadi	1962-02-02	Islam	Tehran	Osol gara	Male	Bachelor Degree	Law
2	0100000002	Ali	Davoodi	1973-02-05	Islam	Tehran	Osol gara	Male	Master Degree	Law
3	0100000003	Hassan	Mohammadi	1955-03-06	Islam	Tehran	Osol gara	Male	Bachelor Degree	Law
4	0100000004	Hassan	Akbari	1966-04-08	Islam	Tehran	Osol gara	Male	Master Degree	Law

Total rows: 61 of 61 Query complete 00:00:00.070 Ln 1028, Col 27

ورود دیتاست صندوق:

Query Query History

```
1008
1087 INSERT INTO electionfund values
1088 (1102, 'Tehran'),
1089 (1202, 'Tehran'),
1090 (1302, 'Tehran'),
1091 (2102, 'Mashhad'),
1092 (2202, 'Mashhad'),
1093 (2302, 'Mashhad'),
1094 (3102, 'Isfahan'),
1095 (3202, 'Isfahan'),
1096 (1101, 'Tehran'),
1097 (1201, 'Tehran'),
1098 (1301, 'Tehran'),
1099 (2101, 'Mashhad'),
1100 (2201, 'Mashhad'),
1101 (2301, 'Mashhad'),
```

Data Output Messages Notifications

	efid [PK] smallint	location character varying (20)
1	1102	Tehran
2	1202	Tehran
3	1302	Tehran
4	2102	Mashhad
5	2202	Mashhad
6	2302	Mashhad
7	3102	Isfahan
8	3202	Isfahan

Total rows: 30 of 30 Query complete 00:00:00.066

Query Query History

```
1077
1078 INSERT INTO election Values
1079 (1100, '2020-02-02'),
1080 (1200, '2016-02-02'),
1081 (1201, '2016-02-03'),
1082 (2100, '2022-03-10'),
1083 (3100, '2022-01-19'),
1084 (4100, '2021-01-11'),
1085 (4110, '2021-05-11');|
1086 select * from election;
```

Data Output Messages Notifications

	electionid [PK] smallint	electiondate date
1	1100	2020-02-02
2	1200	2016-02-02
3	1201	2016-02-03
4	2100	2022-03-10
5	3100	2022-01-19
6	4100	2021-01-11
7	4110	2021-05-11

ورود دیتاست حوزه:

```
1125 INSERT INTO electoraldistrict values  
1126 (11, 'Tehran'),  
1127 (12, 'Mashhad'),  
1128 (13, 'Isfahan');  
1129 select * from electoraldistrict;  
1130  
1131
```

Data Output Messages Notifications

	edid [PK] smallint	edname character varying (10)
1	11	Tehran
2	12	Mashhad
3	13	Isfahan

ورود دیتاست شورای ناظر:

```
1119 INSERT INTO supervisor values  
1120 (11, 'Guardian council','6'),  
1121 (12, 'Guardian council','7'),  
1122 (21, 'Islamic council', '10'),  
1123 (22, 'Islamic council', '11');  
1124 select * from Supervisor;  
1125  
1126
```

INSERT INTO electoraldistrict values

Data Output Messages Notifications

	suid [PK] smallint	councilname character varying (20)	councilperiod smallint
1	11	Guardian council	6
2	12	Guardian council	7
3	21	Islamic council	10
4	22	Islamic council	11

ورود دیتاست مجلس شورای اسلامی:

```
1129  
1130 INSERT INTO IslamicConsultativeAssemblyElection VALUES  
1131 (2100, 10, 'normal');  
1132 select * from IslamicConsultativeAssemblyElection;  
1133  
1134
```

Data Output Messages Notifications

	electionid [PK] smallint	electionperiod integer	type character varying (10)
1	2100	10	normal

ورود دیتاست رابطه ابطال مجلس شورای اسلامی:

```
1234  
1235 INSERT into EFICACanelation values  
1236 (11,2202,'reason 1');  
1237 select * from EFICACanelation;  
1238  
1239
```

Data Output Messages Notifications

	suid [PK] smallint	efid [PK] smallint	reason character varying (100)
1	11	2202	reason 1

ورود دیتاست شرکت مجلس شورای اسلامی:

```

1143 INSERT INTO ICAEnroll VALUES
1144 (2100,'0100000001',11),
1145 (2100,'0100000002',11),
1146 (2100,'0100000003',11),
1147 (2100,'0100000004',11),
1148 (2100,'0100000005',11),
1149 (2100,'0100000006',11),
1150 (2100,'0100000007',11),
1151 (2100,'0100000008',12),
1152 (2100,'0100000009',12),
1153 (2100,'0100000010',12),
1154 (2100,'0100000011',12),
1155 (2100,'0100000012',12),
1156 (2100,'0100000013',12),
1157 (2100,'0100000014',13),

```

Data Output Messages Notifications

	electionid [PK] smallint	canid [PK] character (10)	edid [PK] smallint
1	2100	0100000001	11
2	2100	0100000002	11
3	2100	0100000003	11
4	2100	0100000004	11
5	2100	0100000005	11
6	2100	0100000006	11
7	2100	0100000007	11
8	2100	0100000008	12
Total	8	8	8

ورود دیتاست بررسی صلاحیت مجلس اسلامی:

```

1213 INSERT into ICAValidation values
1214 (12 , '0100000001' , 'qualified'),
1215 (12 , '0100000002' , 'qualified'),
1216 (12 , '0100000003' , 'qualified'),
1217 (12 , '0100000004' , 'qualified'),
1218 (12 , '0100000005' , 'qualified'),
1219 (12 , '0100000006' , 'qualified'),
1220 (12 , '0100000007' , 'qualified'),
1221 (12 , '0100000008' , 'qualified'),
1222 (12 , '0100000009' , 'qualified'),
1223 (12 , '0100000010' , 'qualified'),
1224 (12 , '0100000011' , 'qualified'),
1225 (12 , '0100000012' , 'qualified'),
1226 (12 , '0100000013' , 'qualified'),
1227 (12 , '0100000014' , 'qualified').

```

Data Output Messages Notifications

	suid [PK] smallint	canid [PK] character (10)	status character varying (20)
13	12	0100000013	qualified
14	12	0100000014	qualified
15	12	0100000015	qualified
16	12	0100000016	qualified
17	12	0100000017	qualified
18	12	0100000018	qualified
19	12	0100000019	disqualified
20	12	0100000020	disqualified

ورود دیتاست رابطه has مجلس شورای اسلامی:

```

1133 INSERT INTO ICAEF VALUES
1134 (2100,1102),
1135 (2100,1202),
1136 (2100,1302),
1137 (2100,2102),
1138 (2100,2202),
1139 (2100,2302),
1140 (2100,3102),
1141 (2100,3202);
1142 select * from ICAEF;
1143

```

Data Output Messages Notifications

	electionid [PK] smallint	efid [PK] smallint
1	2100	1102
2	2100	1202
3	2100	1302
4	2100	2102
5	2100	2202
6	2100	2302
7	2100	3102
8	2100	3202

ورود دیتاست رابطه vote در مجلس شورای اسلامی:

```

1165 INSERT INTO ICAVote VALUES
1166 ('0000000001','0100000001',1102),
1167 ('0000000001','0100000005',1102),
1168 ('0000000001','0100000004',1102),
1169 ('0000000001','0100000003',1102),
1170 ('0000000001','0100000002',1102),
1171 ('0000000002','0100000001',1102),
1172 ('0000000002','0100000005',1102),
1173 ('0000000002','0100000004',1102),
1174 ('0000000002','0100000003',1102),
1175 ('0000000002','0100000002',1102),
1176 ('0000000003','0100000001',1202),
1177 ('0000000003','0100000006',1202),
1178 ('0000000003','0100000004',1202),
1179 ('0000000003','0100000003',1202),

```

Data Output Messages Notifications

	void [PK] character (10)	canid [PK] character (10)	efid [PK] smallint
1	0000000001	0100000001	1102
2	0000000001	0100000005	1102
3	0000000001	0100000004	1102
4	0000000001	0100000003	1102
5	0000000001	0100000002	1102
6	0000000002	0100000001	1102
7	0000000002	0100000005	1102
8	0000000002	0100000004	1102

ورود دیتاست انتخابات ریاست جمهوری:

```

1240 (1100, 10, 1, 'normal'),
1241 (1200, 11, 1, 'normal'),
1242 (1201, 11, 2, 'normal');
1243 select * from PresidentialElection;
1244 -----
1245

```

Data Output Messages Notifications

	electionid [PK] smallint	electionperiod smallint	stage smallint	type character varying (10)
1	1100	10	1	normal
2	1200	11	1	normal
3	1201	11	2	normal

ورود دیتاست رابطه شرکت ریاست جمهوری:

```

1255 INSERT INTO PEEroll VALUES
1256 (1100 , '0200000001'),
1257 (1100 , '0200000002'),
1258 (1100 , '0200000003'),
1259 (1100 , '0200000004'),
1260 (1100 , '0200000005'),
1261 (1200 , '0200000004'),
1262 (1200 , '0200000006'),
1263 (1200 , '0200000007'),
1264 (1200 , '0200000008'),
1265 (1200 , '0200000009'),
1266 (1200 , '0200000010'),
1267 (1201 , '0200000007'),
1268 (1201 , '0200000008')

```

Data Output Messages Notifications

	electionid [PK] smallint	canid [PK] character (10)
1	1100	0200000001
2	1100	0200000002
3	1100	0200000003
4	1100	0200000004
5	1100	0200000005
6	1200	0200000004
7	1200	0200000006
8	1200	0200000007
9	1200	0200000008
10	1200	0200000009

ورود دیتاست رابطه has ریاست جمهوری:

```

1244 INSERT INTO PEEF VALUES
1245 (1100,1101),
1246 (1100,2101),
1247 (1100,3101),
1248 (1200,1201),
1249 (1200,2201),
1250 (1200,3201),
1251 (1201,1301),
1252 (1201,2301),
1253 (1201,3301);
1254 select * from PEEF;
1255 -----

```

Data Output Messages Notifications

	electionid [PK] smallint	efid [PK] smallint
1	1100	1101
2	1100	2101
3	1100	3101
4	1200	1201
5	1200	2201
6	1200	3201
7	1201	1301
8	1201	2301
9	1201	3301

ورود دیتاست رابطه vote ریاست جمهوری:

```

1282 INSERT INTO PEVote VALUES
1283 ('0000000001','0200000001',1101),
1284 ('0000000002','0200000001',1101),
1285 ('0000000003','0200000001',1101),
1286 ('0000000004','0200000001',1101),
1287 ('0000000005','0200000002',2101),
1288 ('0000000006','0200000002',2101),
1289 ('0000000007','0200000002',2101),
1290 ('0000000008','0200000001',2101),
1291 ('0000000009','0200000001',3101),
1292 ('0000000010','0200000003',3101),
1293 ('0000000001','0200000004',1201),
1294

```

Data Output Messages Notifications

	void [PK] character (10)	canid [PK] character (10)	efid [PK] smallint
1	0000000001	0200000001	1101
2	0000000002	0200000001	1101
3	0000000003	0200000001	1101
4	0000000004	0200000001	1101
5	0000000005	0200000002	2101
6	0000000006	0200000002	2101
7	0000000007	0200000002	2101
8	0000000008	0200000001	2101
9	0000000009	0200000001	3101
10	0000000010	0200000003	3101

ورود دیتاست بررسی صلاحیت ریاست جمهوری:

```

1269 INSERT INTO PEValidation VALUES
1270 (12 , '0200000001' , 'qualified'),
1271 (12 , '0200000002' , 'qualified'),
1272 (12 , '0200000003' , 'qualified'),
1273 (12 , '0200000004' , 'qualified'),
1274 (12 , '0200000005' , 'disqualified'),
1275 (11 , '0200000004' , 'qualified'),
1276 (11 , '0200000006' , 'qualified'),
1277 (11 , '0200000007' , 'qualified'),
1278 (11 , '0200000008' , 'qualified'),
1279 (11 , '0200000009' , 'disqualified'),
1280 (11 , '0200000010' , 'disqualified');
1281 select * from PEValidation;
1282

```

Data Output Messages Notifications

	suid [PK] smallint	canid [PK] character (10)	status character varying (20)
1		12	0200000001
2		12	0200000002
3		12	0200000003
4		12	0200000004
5		12	0200000005
6		11	0200000004
7		11	0200000006
8		11	0200000007

ورود دیتاست انتخابات مجلس خبرگان:

```

1321 INSERT INTO AssemblyOfExpertsElection VALUES
1322 (3100, 7, 'normal');
1323 select * from AssemblyOfExpertsElection;
1324
1325 INSERT INTO AOEEF VALUES

```

Data Output Messages Notifications

	electionid [PK] smallint	electionperiod integer	type character varying (10)
1	3100	7	normal

ورود دیتاست رابطه شرکت مجلس خبرگان:

```

1332 INSERT INTO AOEEEnroll VALUES
1333 (3100, '0300000001', 11),
1334 (3100, '0300000002', 11),
1335 (3100, '0300000003', 11),
1336 (3100, '0300000004', 11),
1337 (3100, '0300000005', 11),
1338 (3100, '0300000006', 11),
1339 (3100, '0300000007', 11),
1340 (3100, '0300000008', 12),
1341 (3100, '0300000009', 12),
1342 (3100, '0300000010', 12).

```

Data Output Messages Notifications

	electionid [PK] smallint	canid [PK] character (10)	edid [PK] smallint
1	3100	0300000001	11
2	3100	0300000002	11
3	3100	0300000003	11
4	3100	0300000004	11
5	3100	0300000005	11
6	3100	0300000006	11
7	3100	0300000007	11
8	3100	0300000008	12
9	3100	0300000009	12
10	3100	0300000010	12
11	3100	0300000011	12

Total rows: 13 of 13 Query complete 00:00:00.067

ورود دیتاست رابطه has مجلس خبرگان:

```

1324 INSERT INTO AOEEF VALUES
1325 (3100,1103),
1326 (3100,1203),
1327 (3100,1303),
1328 (3100,2103),
1329 (3100,2203),
1330 (3100,2303);
1331 select * from AOEEF;
1332

```

Data Output Messages Notifications

	electionid [PK] smallint	efid [PK] smallint
1	3100	1103
2	3100	1203
3	3100	1303
4	3100	2103
5	3100	2203
6	3100	2303

ورود دیتاست رابطه vote مجلس خبرگان:

```

1362 INSERT INTO AOEVote VALUES
1363 ('0000000001','0300000001',1103),
1364 ('0000000001','0300000005',1103),
1365 ('0000000001','0300000004',1103),
1366 ('0000000001','0300000003',1103),
1367 ('0000000001','0300000002',1103),
1368 ('0000000002','0300000001',1103),
1369 ('0000000002','0300000005',1103),
1370 ('0000000002','0300000004',1103),
1371 ('0000000002','0300000003',1103),
1372 ('0000000002','0300000002',1103),
1373 ('0000000003','0300000001',1203),
1374 ('0000000003','0300000006',1203),
1375 ('0000000003','0300000004',1203),

```

Data Output Messages Notifications

	void [PK] character (10)	canid [PK] character (10)	efid [PK] smallint
1	0000000001	0300000001	1103
2	0000000001	0300000005	1103
3	0000000001	0300000004	1103
4	0000000001	0300000003	1103
5	0000000001	0300000002	1103
6	0000000002	0300000001	1103
7	0000000002	0300000005	1103
8	0000000002	0300000004	1103
9	0000000002	0300000003	1103

Total rows: 13 of 13 Query complete 00:00:00.064

ورود دیتاست بررسی صلاحیت مجلس خبرگان:

```

1347 INSERT INTO AOEValidation VALUES
1348 (12, '0300000001', 'qualified'),
1349 (12, '0300000002', 'qualified'),
1350 (12, '0300000003', 'qualified'),
1351 (12, '0300000004', 'qualified'),
1352 (12, '0300000005', 'qualified'),
1353 (12, '0300000006', 'qualified'),
1354 (12, '0300000007', 'disqualified'),
1355 (12, '0300000008', 'qualified'),
1356 (12, '0300000009', 'qualified'),
1357 (12, '0300000010', 'qualified'),
1358 (12, '0300000011', 'qualified'),

```

Data Output Messages Notifications

	suid [PK] smallint	canid [PK] character (10)	status character varying (20)
1	12	0300000001	qualified
2	12	0300000002	qualified
3	12	0300000003	qualified
4	12	0300000004	qualified
5	12	0300000005	qualified
6	12	0300000006	qualified
7	12	0300000007	disqualified
8	12	0300000008	qualified
9	12	0300000009	qualified
10	12	0300000010	qualified

Total rows: 13 of 13 Query complete 00:00:00.063

ورود دیتاست رابطه has شورای شهر و روستا:

```

1414 INSERT INTO PSEF VALUES
1415 (4100, 1104),
1416 (4100, 1204),
1417 (4110, 1304),
1418 (4110, 1404),
1419 (4100, 3104),
1420 (4100, 3204);
1421 select * from PSEF;
1422

```

Data Output		
	electionid [PK] smallint	efid [PK] smallint
1	4100	1104
2	4100	1204
3	4110	1304
4	4110	1404
5	4100	3104
6	4100	3204

ورود دیتاست انتخابات شورای شهر و روستا:

```

1409
1410 INSERT INTO ProvincialSocietiesElection VALUES
1411 (4100, 8, 'normal'),
1412 (4110, 8, 'mid term');
1413 select * from ProvincialSocietiesElection;
1414

```

Data Output			
	electionid [PK] smallint	electionperiod integer	type character varying (10)
1	4100	8	normal
2	4110	8	mid term

ورود دیتاست رابطه بررسی صلاحیت شورای شهر و روستا:

```

1444 INSERT INTO PSValidation VALUES
1445 (21,'0400000001', 'qualified'),
1446 (21,'0400000002', 'qualified'),
1447 (21,'0400000003', 'qualified'),
1448 (21,'0400000004', 'qualified'),
1449 (21,'0400000005', 'qualified'),
1450 (21,'0400000006', 'qualified'),
1451 (21,'0400000007', 'qualified'),
1452 (21,'0400000014', 'qualified'),
1453 (21,'0400000015', 'qualified'),
1454 (21,'0400000016', 'qualified'),
1455 (21,'0400000017', 'qualified'),
1456 (21,'0400000018', 'qualified').

```

Data Output			
	suid [PK] smallint	canid [PK] character (10)	status character varying (20)
1	21	0400000001	qualified
2	21	0400000002	qualified
3	21	0400000003	qualified
4	21	0400000004	qualified
5	21	0400000005	qualified
6	21	0400000006	qualified
7	21	0400000007	qualified
8	21	0400000014	qualified
9	21	0400000015	qualified

Total rows: 20 of 20 Query complete 00:00:00.053

ورود دیتاست رابطه شرکت شورای شهر و روستا:

```

1422 INSERT INTO PSEnroll VALUES
1423 (4100,'0400000001', 11),
1424 (4100,'0400000002', 11),
1425 (4100,'0400000003', 11),
1426 (4100,'0400000004', 11),
1427 (4100,'0400000005', 11),
1428 (4100,'0400000006', 11),
1429 (4100,'0400000007', 11),
1430 (4100,'0400000014', 13),
1431 (4100,'0400000015', 13),
1432 (4100,'0400000016', 13),
1433 (4100,'0400000017', 13),
1434 (4100,'0400000018', 13),

```

Data Output			
	electionid [PK] smallint	canid [PK] character (10)	edid [PK] smallint
1	4100	0400000001	11
2	4100	0400000002	11
3	4100	0400000003	11
4	4100	0400000004	11
5	4100	0400000005	11
6	4100	0400000006	11
7	4100	0400000007	11
8	4100	0400000014	13
9	4100	0400000015	13

ورود دیتاست رابطه ابطال شورای شهر و روستا:

```

1554 INSERT INTO EFPECancelation VALUES
1555 (22, 1404, 'reason 5');
1556 select * from EFPECancelation;
1557
1558
Data Output Messages Notifications

```

	suid [PK] smallint	efid [PK] smallint	reason character varying (100)
1	22	1404	reason 5

ورود دیتاست رابطه vote شورای شهر و روستا:

```

1466 INSERT INTO PSVote VALUES
1467 ('0000000001', '0400000001', 1104),
1468 ('0000000001', '0400000005', 1104),
1469 ('0000000001', '0400000004', 1104),
1470 ('0000000001', '0400000003', 1104),
1471 ('0000000001', '0400000002', 1104),
1472 ('0000000002', '0400000001', 1104),
1473 ('0000000002', '0400000005', 1104),
1474 ('0000000002', '0400000004', 1104),
1475 ('0000000002', '0400000003', 1104),
1476 ('0000000002', '0400000002', 1104),
1477 ('0000000003', '0400000001', 1204),
1478 ('0000000003', '0400000005', 1204),
1479 ('0000000003', '0400000004', 1204),
1480 ('0000000003', '0400000003', 1204).
1481
Data Output Messages Notifications

```

	void [PK] character (10)	canid [PK] character (10)	efid [PK] smallint
1	0000000001	0400000001	1104
2	0000000001	0400000005	1104
3	0000000001	0400000004	1104
4	0000000001	0400000003	1104
5	0000000001	0400000002	1104
6	0000000002	0400000001	1104
7	0000000002	0400000005	1104
8	0000000002	0400000004	1104

دیتا تایپ با تیبل های همخوئی داشته و جدول های موجودیت ها به درستی اجرا می شوند.

نمونه ای از بررسی تریگر های تعریف شده:

```

1590 INSERT INTO voter VALUES
1591 ('0000009001' , 'Ali' , 'Mohammadi' , '2009-02-02' , 'Male');
1592
Data Output Messages Notifications

```

ERROR: Must be at least 18 years old.
 CONTEXT: PL/pgSQL function check_age() line 4 at RAISE
 SQL state: P0001

```

1591 INSERT INTO voter VALUES
1592 ('0000008001' , 'Al' , 'Mohammadi' , '1962-02-02' , 'Male');
1593
Data Output Messages Notifications

```

ERROR: Name must be longer than 3 characters.
 CONTEXT: PL/pgSQL function check_name_length() line 4 at RAISE
 SQL state: P0001

```

1591 INSERT INTO PSVote VALUES
1592 ('0000000001','0400000022',1104);
1593

```

چون این کاربر قبلاً به کاندید

Data Output Messages Notifications

ERROR: Duplicate vote not allowed
CONTEXT: PL/pgSQL function check_duplicate_vote_ps() line 4 at RAISE
SQL state: P0001

دیگه ای رای داده بود.

```

1591 INSERT INTO Candidate VALUES
1592 ('0100000001','Ali','Mohammadi','1962-02-02','Islam','Paris','Osol gara','Male','Bachelor Degree','Law');
1593

```

Data Output Messages Notifications

ERROR: City must be one of {Tehran,Mashhad,Isfahan,Karaj,Shiraz,Tabriz,Qom,Ahvaz,Kermanshah,Urmia,Rasht,Zahedan,Hamadan,Kerman,Yazd,Ardabil,"Bandar Abbas",Arap,Eslamshahr,Qazvin,Zanjan,Khorramabad,Sanandaj,Malayer,Borujerd,Abadan,Dezful,Kashan,Sari,Gorgan}
CONTEXT: PL/pgSQL function check_city() line 6 at RAISE
SQL state: P0001

این ها جهت نمونه تست سیستم بودن و تمامی ویو ها و تریگرها تست شده اند. جهت اطمینان میتوانید در سورس کد می توانید امتحان کنید.

نمونه ای از ویو ها که نتیجه انتخابات ریاست جمهوری را به ما نشان می دهد:

```

1587
1588 SELECT * FROM final_peresult;
1589
1590

```

Data Output Messages Notifications

	electionid smallint	stage smallint	firstname character varying (10)	lastname character varying (10)	canid character (10)	vote_count bigint	status text
1	1100	1	Hossein	Molavi	0200000001	6	Winner
2	1100	1	Hossein	Asghari	0200000002	3	Loser
3	1100	1	Ali	Mortazavi	0200000003	1	Loser
4	1200	1	Javad	Davoodi	0200000007	5	Runner-up
5	1200	1	Nima	Davoodi	0200000008	4	Runner-up
6	1200	1	Hassan	Ghalibaf	0200000004	2	Loser
7	1200	1	Asghar	Heidari	0200000006	1	Loser
8	1201	2	Javad	Davoodi	0200000007	11	Winner
9	1201	2	Nima	Davoodi	0200000008	4	Loser

شاید با خودتون بگین حسین که در مرحله اول برنده انتخابات شده پس چرا باز انتخابات رفته برای مرحله دو؟ چون حسین اصلا تو یک انتخابات دیگه با شماره انتخاباتی متفاوت در یک دوره دیگه برنده شده.

شاید با خودتون بگین که نیما وجود هم رای یکسان ماسکیم رو ندارن پس چرا رفتن مرحله دو؟ باید بگم در اصل شرط رفتن دو نفر اول با بیشترین رای به مرحله دو اینه که کسی بالای 50 درصد کل آرا را اخذ نکنه! هیچ کس در انتخابات 1200 پنجاه درصد آرا را اخذ نکرده پس دو نفر با بیشترین رای وارد مرحله دوم می شوند.