**BIM - 312 DATABASE MANAGEMENT SYSTEMS**

Phase 2

group no: 17

Project no: 7: Online votıng system

Semİh kayan - 34987871568 Furkan allİş - 18920065832

**REQUIREMENT ANALYSIS**

**What is the purpose of the database? What should it do?**

* The point of the database is to provide a secure, easy to use and fast environment to voters and candidates to vote and to be elected.
* The database stores admins’, voters’, candidates’, elections’ information, and the relations between them.
* These relations are admins create elections, candidates attend to them, voters vote those who candidates attended the election according to their background.

**Who are the users and what are their information needs?**

* The user types are admin, voter, candidate.
* All users need id, nationality id, name, password, age, address, contact.
* Candidates also need education info and an introduction that introduce them.

**What are the problems that the system should solve?**

* Complexity of managing election data and keep information second by second.
* Candidates should be ordered by vote count.

**What input data is available to the database?**

* Users’, voters’, candidates’, admins’, elections’, election candidates’ information is available to the database.

**What kind of information should be stored in the database?**

* Voters’ personal data.
* Admins' personal data.
* Candidates’ personal data and educational info.
* Elections’ start – end dates, creation date, who is created it, who are currently managing it and attended candidates to it.
* Votes’ vote date.

**Cardinality Explanation:**

* One admin can create many elections, one election can be created by one admin.
* Many admins can manage one election, one admin can manage more than one election.
* Many candidates can attend one election at the same time, one candidate can attend many elections at the same time.
* One voter can vote one candidate in an election, a candidate can be voted by many voter

**Participation Explanation:**

* An election should be created by an admin, but an admin does not have to create an election, but still exists.
* An election should be managed by at least one admin, but an admin does not have to manage an election, but still exists.
* A candidate does not have to attend an election, an election can exist without any candidate attending.

**Functional/Non-functional Requirements:**

Functional:

* Create a new user.
* Get user info.
* Add a candidate to an election.

Non-functional:

* Separate last name information from full name.
* Get vote count per candidate by counting the tuples by using election candidate id in the “votes” relationship.

**QUERIES**

* What is the most voted candidate in an election?
* Is the user an admin?
* When did voter vote the candidate?
* When was an election created, and by whom?
* Which admins are managing which election?
* Which candidates attended which election?
* What is the education info of a candidate?
* What is information of a user?
* When will an election start and end?

**Available Procedures:**

call get\_election\_votes\_by\_eid (945);

call get\_most\_voted\_candidate\_in\_the\_election (945);

call check\_if\_user\_an\_admin (800);

call get\_vote\_date (212, 461);

call when\_was\_created\_the\_election (945);

call get\_election\_managers (945);

call get\_election\_attenders (945);

call get\_schools\_by\_candidate (500);

call get\_certifications\_by\_candidate (500);

call get\_user\_by\_uid (500);

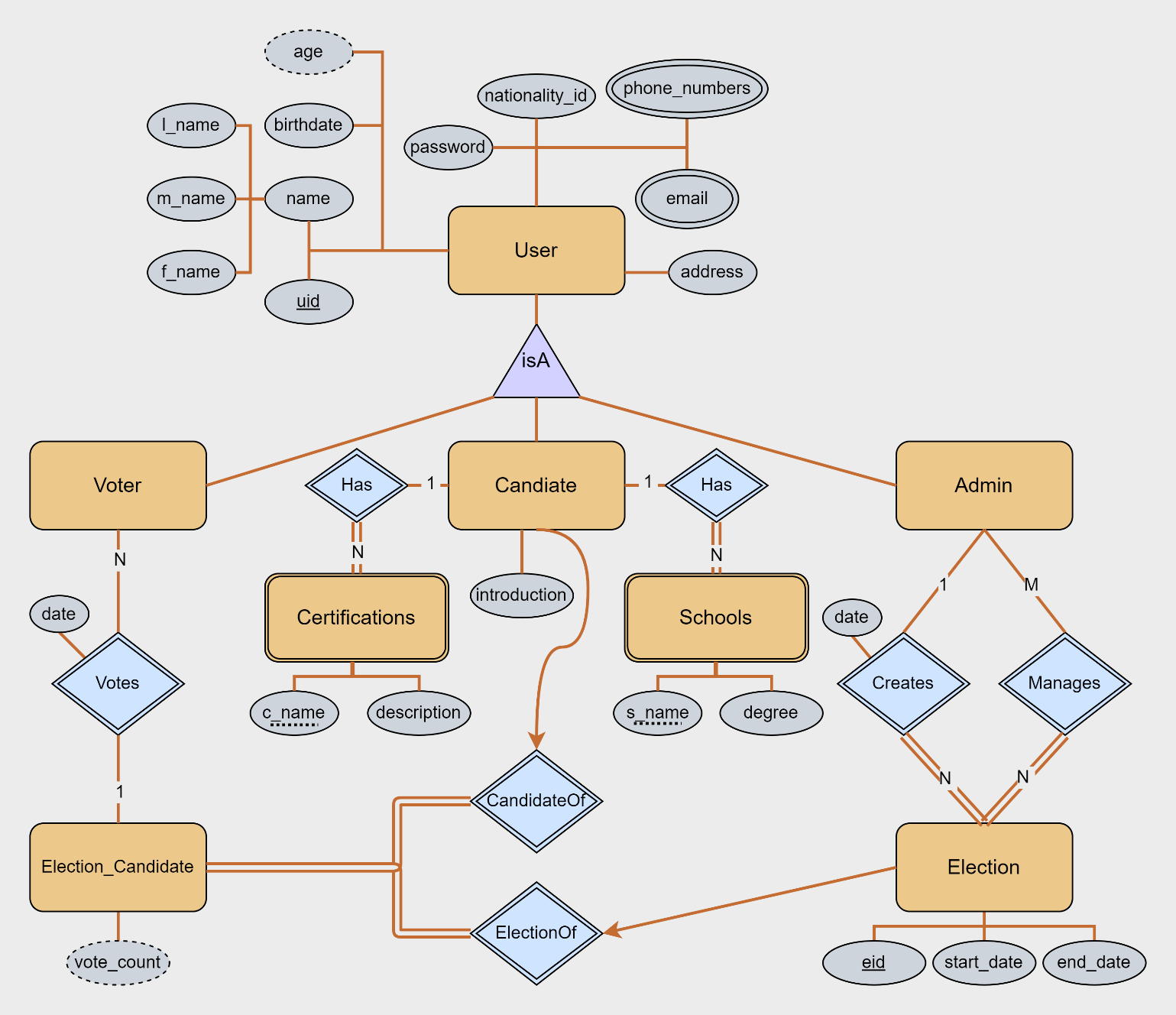
call get\_user\_emails (424);

call get\_user\_phone\_numbers (424);

call when\_will\_start\_and\_end\_the\_election (945);

**Note**: Procedure queries can be viewed in all-in-one (lines between 143 and 241) and addition > separately > procedures.sql files.

**E/R DIAGRAM**

****

**E/R DIAGRAM UPDATE**

We have realized that the previous version of E/R had **conflicts**.

**Conflict-1**: “Education” Weak Entity had a name partial key, and it was redundant. So, we have deleted it. Finally, we have decided to separate out “certifications” and “schools” multi-valued attributes into two entities instead.

**Conflict-2**: The relationship “attends” between “candidate” and “election” was equivalent to “election candidate” entity. So, we have decided to keep “election candidate” entity and delete “attends” relationship. Instead of “attends” relationship, we relate “candidate” with “election candidate” entity. It would be more accurate because “election candidate” entity has relations and could be more relations and attributes as application grows.

**Lastly**, we considered that users can have multiple phone numbers and emails. So, it was better and more flexible if they were multi-valued.

**NORMALIZATION**

1st Normal Form: All attributes should be atomic (not multi-valued or composite). We have two multi-valued attributes “emails” and “phone numbers” in the “user” entity. If we have a user that has multiple email or phone number, there would be redundancy and it violates 1NF. So, we separated these attributes into tables named “user\_emails” and “user\_phone\_numbers”. Also, we have one composite attribute “name” in “user” entity. It should be separated as f\_name, m\_name and l\_name attributes.

2nd Normal Form: “A relation schema R is in second normal form (2NF) if it is in 1NF and if every non-prime attribute A in R is fully functionally dependent on the primary key.” We have no partial dependency. So, there is nothing to fix here.

3rd Normal Form: “A relation schema R is in third normal form (3NF) if it is in 2NF, and no non-prime attribute of R is transitively dependent on the primary key.” We have no transitive dependency. So, there is nothing to fix here too.

Boyce-Codd Normal Form: There should be no “bad” FDs. We have no functional dependency that creates anomaly. So, there is nothing to fix here too.

**RELATIONAL SCHEMA**

**Diagram

Description automatically generated**

**Note**: SQL create table commands that include integrity constraints can be viewed in all-in-one (lines between 4 and 141) and addition > separately > schema.sql files.

**DATABASE SCHEMA**

Graphical user interface, diagram

Description automatically generated with medium confidence

**Note**: All the SQL codes are in the all-in-one.sql file and it can be copy-pasted directly. Also, all codes are in addition folder separately because it is hard to find anything in the all-in-one.sql file (13267 lines of code).