Database Normalisation

Design

Voter

PK	id
	name: string
	dateOfBirth: date
	specialNumber: string
	citizenship: string
	email: string
	phoneNumber: string
	password: string
	authenticated: bool
	walletPrivateKey: string
	walletAddress: string
FK	securityQuestion1: string
	securityAnswer1: string
FK	securityQuestion2: string
	securityAnswer2: string
	authToken: string
	resetToken: string
	tokenAuthenticated: string

Vote

PK	id
FK	voterId: int
FK	electionId: int
	encryptedVote: string
	blindedSignature: string

Candidate

PK	id
FK	electionId: int
	name: string
	voice: string
	party: string
	image: file
	dateOfBirth: date
	biography: string
	isWinner: bool
	primeNumber: int

Election

PK	id
FK	adminId: int
FK	results: int
	title: string
	description: string
	startDate: date
	endDate: date
	resultDate: date
	candidateNumber: int
	ageRestriction: int
	authEmail: string
	authCitizenship: string
	isActive: bool
	type: string
	publishKey: string

Admin

PK	id
	email: string
	password: string
	blindPrivateKeyPath: string
	blindPublicKey: string
	paillierPrivateKeyPath: string
	paillierPublicKey: string

Tally

PK	id
FK	electionId: int
	sum: bigint

RankTally

PK	id
FK	electionId: int
	sum: bigint
	rank: int

ScoreTally

PK	id
FK	electionId: int
	sum: bigint
FK	candidateId: int

Result

PK	id
FK	winner: int
	electionId: int

SecurityQuestion

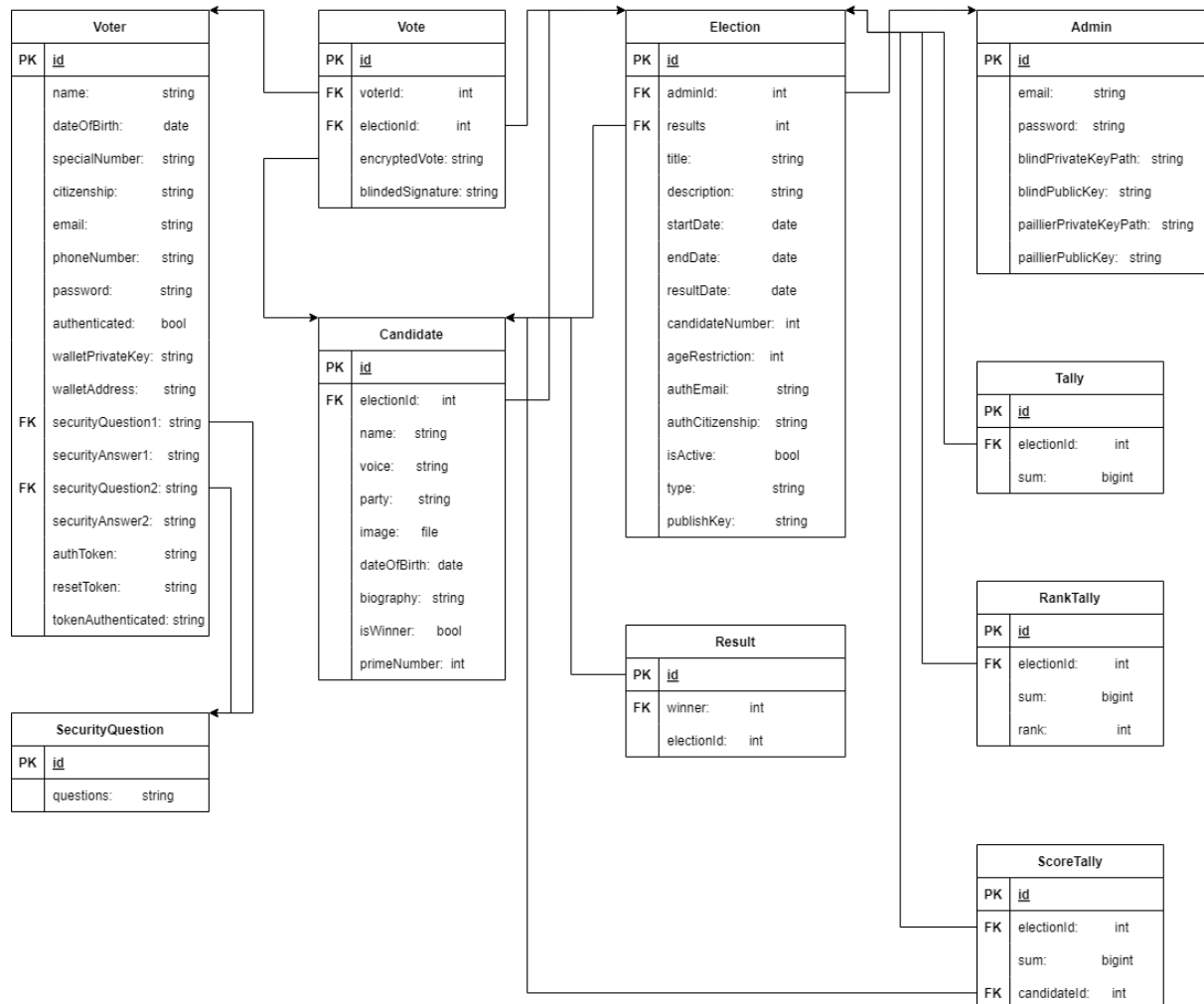|	
PK	id
	questions: string

Figure1

Normalisation First Form
- Each record should have a unique name: passed
- Each table cell should contain a single value: passed

Normalisation Second Form
- Should be in first normal form: passed
- No partially dependant keys: passed

Normalisation Third Form
- Should be in second normal form: passed

- No transitive functional dependencies: passed

Normalisation Boyce-Codd Form
- Should be in third normal form: passed
- No more than one candidate key: passed