Isabel Arvelo
isabel.c.arvelo@vanderbilt.edu
Project 1 Part 2 (Normalization)

There is a number before each query in the FD SQL file. The number in parentheses next to FDs described below refer to the query number to make it easier to track between the two files.

Bad Data/Data Irregularities:
Duplicate Rows: I identified records that were completely identical (including registration date) except for voter registration number (1) and took these to be duplicates in which the same voter was given two different voter registration numbers because the likelihood of two individuals that are identical across all attributes registering at the same exact time is very low and the more likely scenario is that these are duplicates. I kept the record with the maximum registration number (which is associated with most recent registration date) and put the remaining records in the misc_table, a separate miscellaneous table allows for further investigation and potential data cleansing efforts while maintaining the quality of the main dataset.

Missing Precinct Data: I discovered instances where the pct_portion was missing for some voters()2. This attribute determines precinct_desc (14), nc_house_desc(13), and the nc_senate_desc(15) fields. Missing precinct and legislative district information can have several implications:

- It would be unclear which precinct and legislative districts these voters belong to, making it difficult to verify their eligibility to vote in specific races.
- Precincts and legislative districts are crucial for organizing and administering elections.
- Precinct-level data is often used for analyzing and reporting election results.

To handle missing precinct data, I moved the records missing pct_portion to the miscellaneous table to ensure data integrity and maintain referential integrity in the main dataset.

Inconsistent Names: I identified cases where full_name_mail does not determine the first_name, middle_name, last_name, and name_suffix_lbl for each record (3). Upon further investigation, I found that in these 13 cases, the same middle name was in both records but it was either placed in the first name or last name field in one record and the middle name field in the other record. Since a full name should be uniquely defined by its component parts (first name, middle name, last name, and suffix) and only occurred 13 times, I considered this a data irregularity. I found similar inconsistencies in which there were 21 occurrences where multiple records had the same full_name_mail, age, house_num, street_dir, street_name, street_type_cd, street_sufx_cd, zip_code, but different voter_reg_num (2). It is highly unlikely for two people with the same exact full name and age to live in the same household.

In addition, for both cases above, I found that other attributes such as age, party code, ethnic code, and race codes remained consistent between the records. The discrepancies in party code, ethnic code, and race codes were limited to differences between a specific code and "Undesignated" (UNA for ethnic code and U for race). There were no cases where the records showed two different party affiliations, races, or ethnicities. This consistency in other attributes, along with the fact that full_name_mail determines the name components and full_name_mail and residential address information determines voter_reg_num for mostly all of the rows and the

FD makes logical sense, strongly suggests that the inconsistencies were likely data entry errors for the same individual, during the registration process either after they moved or updated their registration for any other reason.

To handle these cases, I kept the record with the most recent registration date (which corresponds to the higher registration number) in the main dataset and moved the other record to a separate miscellaneous table. In two specific cases where the ages did not match between the records, I did not have sufficient information to determine if they belonged to the same person. To be conservative, I moved both records to the miscellaneous table. In total, there are 309 rows in the miscellaneous table.

Decomposition:

1. *First Normal Form (1NF)*

The original meckcountyvoters table was not in 1NF so I began with decomposing the table into one where each column in the table contains atomic values, and there are no repeating groups or arrays. The election attributes e1, e1_date, e1_votingmethod, e1_partycd, e2, e2_date, e2_votingmethod, e2_partycd, e3, e3_date, e3_votingmethod, e3_partycd, e4, e4_date, e4_votingmethod, e4_partycd are all repeated four times so the election-related attributes were moved into a separate table with renamed columns (no new information)

- election_votes (voter_reg_num, election_number, date, voting_method, party_cd)

Although it could be argued that columns such as mail_addr1, mail_city_state_zip, full_name_mail and e1_date could be divided into smaller units, one of the constraints in this project is to not add additional columns so we will assume these values are atomic.

2. *Second Normal Form (2NF)*

For a table to be in 2NF, every non-prime key must be dependent on full of every candidate key. I found that full_name_mail, mail_addr1, registr_dt, age -> voter_reg_num (22). By transitivity, this makes full_name_mail, mail_addr1, registr_dt, age a candidate key. I then looked for partial dependencies in which certain columns depended on subset of this key.
Partial dependencies found:

- full_name_mail -> first_name, middle_name, last_name, name_suffix_lbl (26)
    - This is the result of my data cleaning to uphold this FD.
- full_name_mail, mail_addr1-> mail_addr2, mail_city_state_zip (27)

I created separate tables to eliminate these dependencies: a voter table for voter's personal information, a voter_address table for voter's address information.

- voter (voter_reg_num, full_name_mail, age, registr_dt, party_cd, race_code, ethnic_code, sex_code, precinct_desc, pct_portion, nc_senate_desc, nc_house_desc, house_num,street_dir,street_name,street_type_cd, street_sufx_cd, zip_code, full_name_mail, mail_addr1, e1, e2, e3, e4)
- voter_name (full_name_mail, first_name, middle_name, last_name, name_suffix_lbl)
- voter_mailing_address (full_name_mail, mail_addr1, mail_addr2, mail_city_state_zip)

election_votes is not in 2NF because e1 -> e1_date, e2 -> e2_date, e3 -> e3_date, e4 -> e4_date (8) so election_number -> election_date is a partial dependency in election_votes. I split this into two tables:

- election_votes (voter_reg_num, election_number, voting_method, party_cd)
- election_info (election_number, election_date)

3. *Third Normal Form (3NF)*

In order for a table to in 3NF, it must be in 2NF and the left side of all non-trivial FDs must be super keys or the right side must be a key attribute.
I found the following dependencies between non prime attributes that violate 3NF:

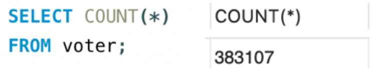
- voter
    - zip_code → res_city_desc, state_cd (20) but res_city_desc, state_cd does not determine zip_code (21)
    - precinct_desc → nc_senate_desc (11)
    - pct_portion → precinct_desc, nc_house_desc (13)
    - house_num, street_dir,street_name,street_type_cd, street_sufx_cd, zip_code -> pct_portion (24)

I created separate tables to eliminate these transitive dependencies: city_state table to store the mapping between zip_code, res_city_desc, and state_cd and table to store the house and senate information. I did several checks to make the decomposed tables are in 3NF to show that there was no dependency between non-prime attributes and it is therefore appropriate to keep all the attributes in each table.

- voter (voter_reg_num, full_name_mail, age, registr_dt, party_cd, pct_portion, race_code, ethnic_code, sex_code, house_num, street_dir, street_name, street_type_cd, street_sufx_cd, zip_code, mail_addr1)
    - When checking if ethnic_code depends on race_code, the query returned 7 rows, indicating that race_code does not uniquely determine ethnic_code (5)
    - full_name_mail does not uniquely determine sex_code (6)
    - race_code, ethnic_code, sex_code do not determine party_cd (7)
    - house_num does not depend on street_name, street_type_cd, street_sufx_cd (18)
    - I found that house_num, street_dir, street_name, street_type_cd, street_sufx_cd -> zip_code (19), however from domain knowledge I know that address details do not determine zip code but it just uncommon for the same exact address to exist in different zip codes. For example, in New Jersey there is a 3 Julia Ct. in the zip codes 08057 and 07005 so I will not enforce this FD in my table
    - There should be no dependency between mailing and residential address attributes because it is possible for one of these to change without affecting the other
- voter_mailing_address (full_name_mail, mail_addr1, mail_addr2, mail_city_state_zip)
    - mail_addr2 does not depend on mail_addr1 (17)
    - mail_city_state_zip does not depend only on mail_addr1(23)
- nc_house_desc (pct_portion, precinct_desc, nc_house_desc)
    - nc_house_desc does not depend on precinct_desc (12)
- election votes
    - When checking if e1_votingmethod depends on e1_date (8). The query returned 1 row, indicating that e1_date does not uniquely determine e1_votingmethod and the same follows for the remaining election attributes.
- city_state, nc_house_desc, election_date are in 2NF and only have one non-prime attribute so these are automatically in 3NF

Final decomposition and Count Screenshots:
I moved 290 records into misc_table, so I am only using 383107 rows from the original data in the decomposed tables.

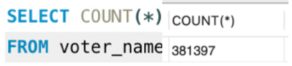
voter (voter_reg_num, full_name_mail, age, registr_dt, party_cd, pct_portion, race_code, ethnic_code, sex_code, house_num, street_dir, street_name, street_type_cd, street_sufx_cd, zip_code, mail_addr1)



```sql
SELECT COUNT(*)
FROM voter;
```

COUNT(*)
383107

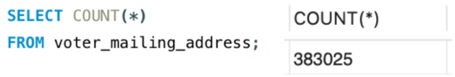
voter_name (full_name_mail, first_name, middle_name, last_name, name_suffix_lbl)



```sql
SELECT COUNT(*)
FROM voter_name
```

COUNT(*)
381397

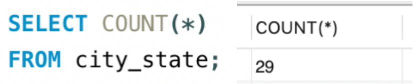
voter_mailing_address (full_name_mail, mail_addr1, mail_addr2, mail_city_state_zip)



```sql
SELECT COUNT(*)
FROM voter_mailing_address;
```

COUNT(*)
383025

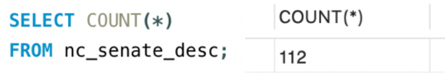
city_state (zip_code, res_city_desc, state_cd)



```sql
SELECT COUNT(*)
FROM city_state;
```

COUNT(*)
29

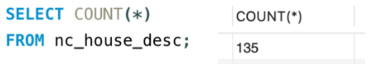
nc_senate_desc (precinct_desc, nc_senate_desc)



```sql
SELECT COUNT(*)
FROM nc_senate_desc;
```

COUNT(*)
112

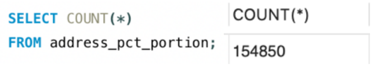
nc_house_desc (pct_portion, precinct_desc, nc_house_desc)



```sql
SELECT COUNT(*)
FROM nc_house_desc;
```

COUNT(*)
135

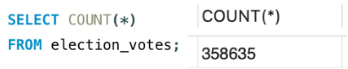
address_pct_portion (street_dir, street_name, street_type_cd, street_sufx_cd, zip_code
,pct_portion)



```sql
SELECT COUNT(*)
FROM address_pct_portion;
```

COUNT(*)
154850

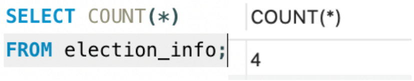
election_votes (voter_reg_num, election_number, voting_method, party_cd)



```sql
SELECT COUNT(*)
FROM election_votes;
```

COUNT(*)
358635

election_info (election_number, election_date)



```sql
SELECT COUNT(*)
FROM election_info;
```

COUNT(*)
4