Data Management for Data Science
Group Members:
Trina Sahoo (1901254)
Debodeep Banerjee(1901253)

Relational Schema:-

create table candidate_details (AC_Number numeric(10) primary key, Candidate_name varchar(100) not null, Party varchar(100) not null, Caste char check (Caste = 'GEN' OR Caste = 'SC' OR Caste = 'ST') , Margin numeric(10) not null);

create table election_details (AC_Number numeric(10), AC_Name varchar(100) primary key, District varchar(100) not null, foreign key (AC_Number) references candidate_details (AC_Number));

create table result (AC_Name varchar(100), Population numeric(20), Votes numeric(20) not null, Margin numeric(10) not null, Poll_perc numeric(10) not null, foreign key (AC_Name) reference election_details(AC_Name));

Description of the schema:-

Candidate_details table:-

Field	Type	Null	Key	Default
AC_Number	decimal(6,0)	NO	PRI	NULL
Candidate_name	varchar(100)	NO		NULL
Party	varchar(100)	NO		NULL
Caste	varchar(100)	NO		NULL
Margin	decimal(6,0)	NO		NULL

Election_details table:-

Field	Type	Null	Key	Default
AC_Number	decimal(6,0)	YES	MUL	NULL
AC_Name	varchar(100)	NO	PRI	NULL
District	varchar(100)	NO		NULL

Result table:-

Field	Type	Null	Key	Default
AC_Name	varchar(100)	YES	MUL	NULL
Population	decimal(20,0)	NO		NULL
Votes	decimal(20,0)	NO		NULL
Margin	decimal(20,0)	NO		NULL
Poll_perc	decimal(20,0)	NO		NULL

Homework 1

Query:-

```
-- 1. Winning margin of the candidate from Bullygange
select distinct Margin from result where AC_Name="Ballygunge";
-----------------------------------------------------------------------------------------------------
-- 2. Name of the candidate win by maximum margin from the party name Indian
National Congress
select Candidate_name from candidate_details where Party = "Indian National
Congress" order by Margin desc LIMIT 1;
-----------------------------------------------------------------------------------------------------
-- 3. Count the number of seats won by the party All India Forward Bloc
Select count(Candidate_name) from candidate_details where Party = "All India
Forward Bloc";
-----------------------------------------------------------------------------------------------------
-- 4. Margin of win of the candidate where poll percentage in minimum
Select distinct Margin from result order by Poll_perc LIMIT 1;
-----------------------------------------------------------------------------------------------------
-- 5. Which district has highest winning seats by CPI(M) candidates
select District from election_details, candidate_details where Party="Communist
Party Of India (Marxist)" order by District DESC LIMIT 1;
-----------------------------------------------------------------------------------------------------
-- 6. Which SC candidate of CPI(M) has highest margin
select Candidate_name from Candidate_details where Caste = "SC" and Party =
"Communist Party Of India (Marxist)" having max(Margin);
-----------------------------------------------------------------------------------------------------
-- 7. Form a table consisting of the political parties, # gen candidates and # SC
candidates
select Party, sum(case when Caste = "GEN" then 1 else 0 end) as GEN_Category,
sum(case when Caste = "SC" then 1 else 0 end) as SC_Category from
Candidate_details group by Party;
-----------------------------------------------------------------------------------------------------
-- 8. Population of the AC_Name where poll_perc is minimum
  select Population from result order by Poll_perc limit 1;

-----------------------------------------------------------------------------------------------------
-- 9. Which AC_Name has minimum poll_perc
  select AC_Name from result order by Poll_perc LIMIT 1;

-----------------------------------------------------------------------------------------------------
-- 10. Which district has highest number of AC_Name
  select District from election_details order by AC_Name DESC LIMIT 1;
```

Homework 2

-- 1. *Name of the candidate win from Kolkata*
Select Candidate_name from Candidate_details C INNER JOIN election_details E ON C.AC_Number=E.AC_Number where District = "Kolkata";

-- 2. *Districts having highest "Non-GEN" category candidates*
select District from election_details E inner join candidate_details C on E.AC_Number=C.AC_Number where Caste not in (select District from election_details, candidate_details where Caste = "GEN") order by District DESC LIMIT 1;

Optimized way by creating index

-- creating index on caste for table candidate_details;
create index index_2 on candidate_details(Caste);
select District from election_details E inner join candidate_details C on E.AC_Number=C.AC_Number where Caste not in (select District from election_details, candidate_details where Caste = "GEN") order by District DESC LIMIT 1;

-- Creating Index on Party of table candidate_details;
create index index_1 on candidate_details(Party);
show index from candidate_details;

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality
candidate_details	0	PRIMARY	1	AC_Number	A	294
candidate_details	0	AC_Number	1	AC_Number	A	294
candidate_details	1	index_1	1	Party	A	9
candidate_details	1	index_2	1	Caste	A	3

-- 3. *Population of the AC_Name where All India Trinamool Congress won by lowest Margin*
select Population from result R inner join candidate_details C on R.Margin=C.Margin where Party = "All India Trinamool Congress" having min(R.Margin);

-- 4. *Number of total votes where the Communist Party of India (Marxist) won by the highest margin*
select Votes from result R inner join candidate_details C on R.Margin=C.Margin where Party= "Communist Party of India (Marxist)" order by C.Margin DESC LIMIT 1;

-- 5. *Total number of electors wherever the Indian Nation Congress has won*
Select sum(Population) as Total_Electors from result R inner join candidate_details C on R.Margin = C.Margin where Party= "Indian National Congress";

Optimized way by creating View

create view Total_Electors as select Population from result R inner join candidate_details C on R.Margin = C.Margin where Party = "Indian National Congress";
select sum(Population) from Total_Electors;