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	Туре	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:-

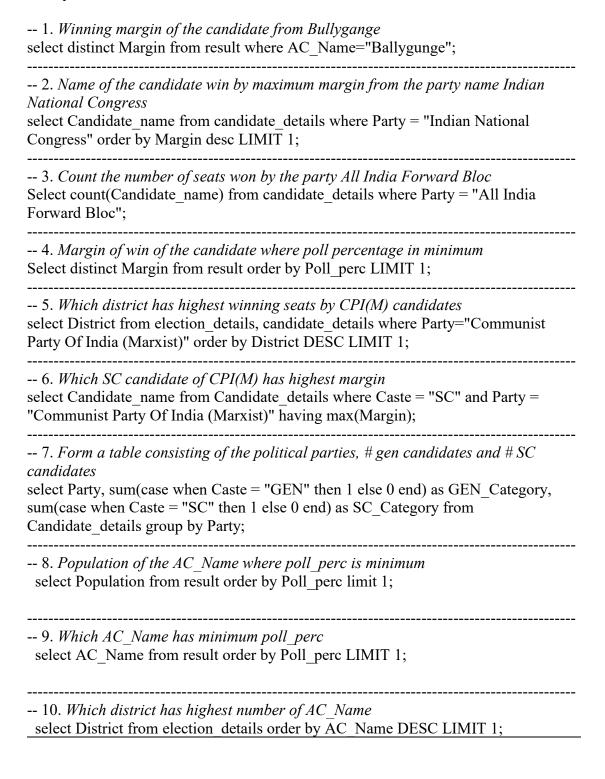
Field	Туре	Null	Key	Default
AC_Number	decimal(6,0)	YES	MUL	NULL
AC_Name	varchar(100)	NO	PRI	NULL
District	varchar(100)	NO		NULL

Result table:-

Field	Туре	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:-



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;