

DBMS MINI PROJECT

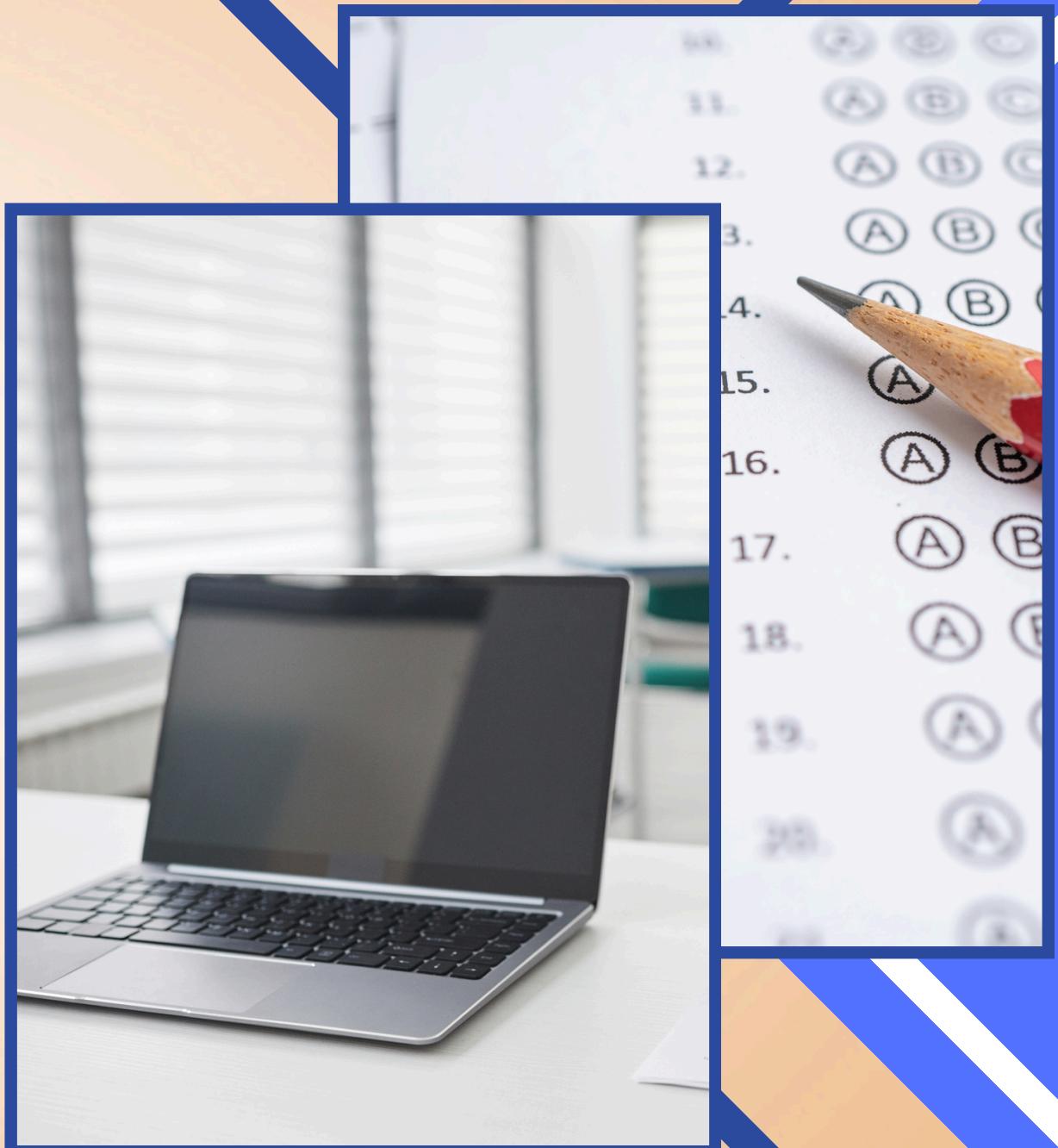


ONLINE EXAMINATION MANAGEMENT SYSTEM

AN ATTEMPT AT DATABASE MANAGEMENT

OUR INSPIRATION

WITH THE RECENT CHANGES IN EDUCATION SCHEMES AND NATIONWIDE TREND OF ENTRANCE EXAMINTAIONS BEING HELD FOR GETTING JOBS AND COLLEGES, THE NEED FOR A SIMPLE YET QUINTESSENTIAL WAY OF ORGANISING EXAMINATIONS WAS APPARENT, ESPECIALLY WITH THE HUGE AND EVERGROWING NUMBER OF CANDIDATES, AND ALONG CAME THE NEED OF ORGANISING THE RELATED DATA.



GROUP MEMBERS

PAJJURU RAGHAVENDRA REDDY

2021IMT-071

GAURAV SINGH

2021IMT-037

DIVJOT SINGH PRUTHI

2021IMT-031

PRITAM YADAV

2021IMT-077

SHIVAM SINGH

2021IMT-091

HARSH DEV

2021IMT-040

Under the guidance of Dr. Debanjan Sadhya

CONTENT

- 1 Significance of our project
- 2 What's new with our project?
- 3 ER Diagram
- 4 ER to Schema
- 5 Functional Dependencies
- 6 Normalization
- 7 Our Database
- 8 Relational Algebra
- 9 SQL Queries

WHY DO WE NEED A ONLINE EXAMINATION DATABASE SYSTEM?



In order to organize the massive amounts of data that can often become erratic and incoherent or not well managed, we need a database management system that has attributes and tables dedicated for all the nitty-gritties as explained further in our project.

Thus our project aims at providing a flexible and robust environment for storing and retrieving the data related to an online examination.

WHAT'S NEW?

1

ACCOMODATION OF STUDENTS IN EXAMINATION CENTRES

A network of centers and their details to be provided to students for reaching the centers to give the examinations

2

A LOGIN SYSTEM FOR STUDENT TO ACCESS THE EXAMINATON AND RESULT

A login format is provided for student to access both the paper as well as the result from the same login, and the student would be able to access only his/her own result

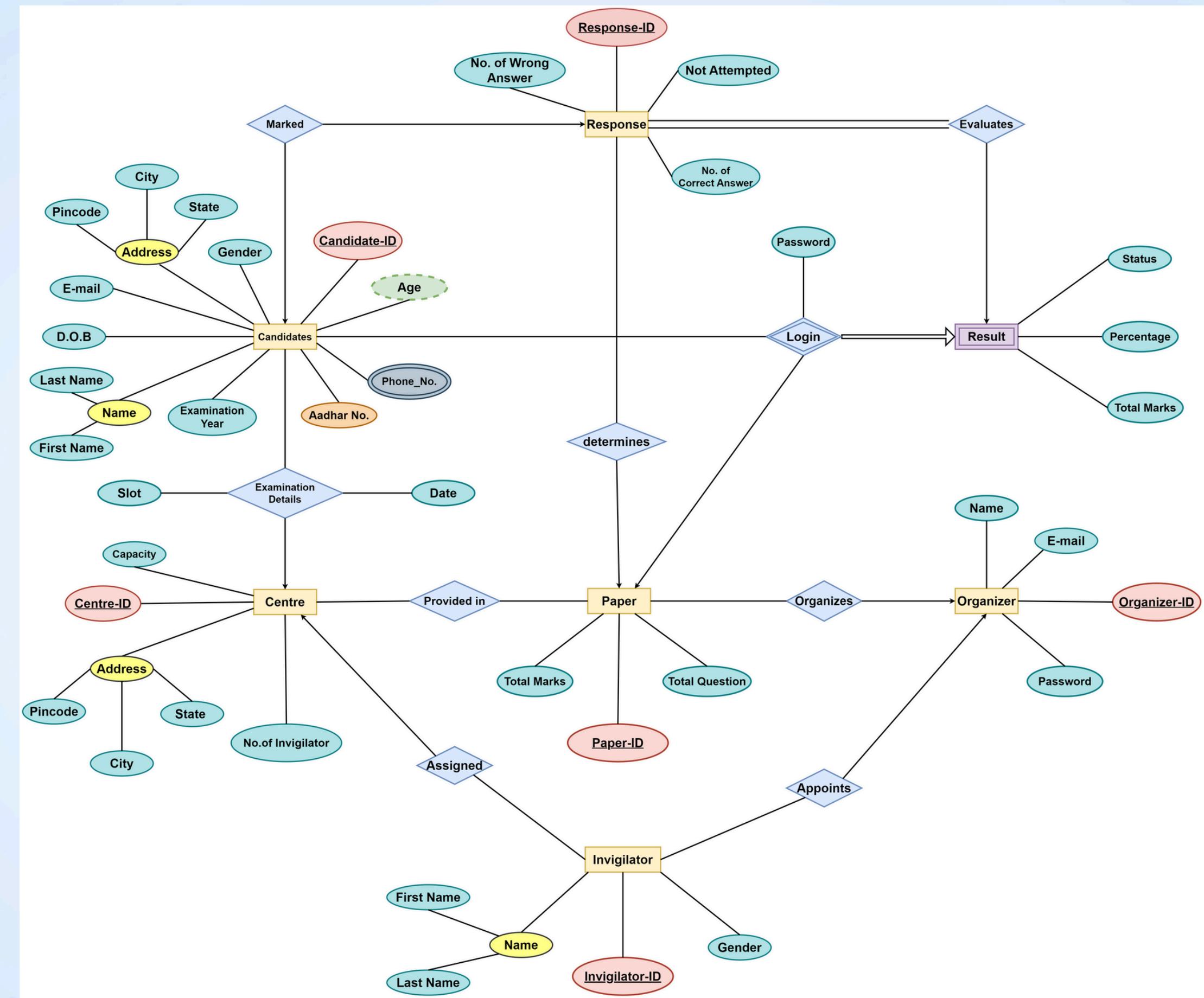
3

INVIGILATORS APPOINTED BY THE ORGANIZER AND ASSIGNED TO CENTERS

Details about invigilators are provided, who are appointed by the organizers and assigned to different centers



ER DIAGRAM



ER TO SCHEMA

The following are the relational schemas for the above ER diagram-

- 1) CANDIDATE(CANDIDATE_ID, PHONE NO. , CITY,STATE,PINCODE,
PASSWORD,First_NAME,LAST_NAME,AGE, AADHAR NO., D.O.B, EMAIL,CENTRE_ID)
- 2) CANDIDATE_PHONENO(CANDIDATE_ID,PHONE_NO)
- 3) CENTRE(CENTRE_ID,NO. OF INVIGILATORS, ADDRESS,CAPACITY)
- 4) LOGIN(CANDIDATE_ID,PASSWORD)
- 5) PAPER(PAPER_ID,TOTAL MARKS,TOTAL QUESTIONS, ORGANIZER_ID)
- 6) ORGANIZER(ORGANIZER_ID,NAME,PASSWORD,EMAIL)

ER TO SCHEMA

The following are the relational schemas for the above ER diagram-

7) RESULT(CANDIDATE_ID,MARKS,PERCENTAGE,STATUS)

8) EXAMINATION DETAILS(CANDIDATE_ID,SLOT,DATE)

9) INVIGILATOR(INVIGILATOR_ID,NAME,GENDER,CENTRE_ID,ORGANIZATION_ID)

10) RESPONSE(RESPONSE_ID,NO_of_WRONG_ANSWERS,NOT_ATTEMPTED,
NO_OF_CORRECT_ANSWERS,CANDIDATE_ID)

11) PROVIDED_IN(CENTRE_ID,PAPER_ID)

FUNCTIONAL DEPENDENCIES

Candidate

- 1) candidate_ID → name, gender, e-mail, D.O.B., age, city, state, pincode, aadhar no., examination_year
- 2) Aadhar no. → First_name, Last_name, Gender, e-mail, D.O.B., city, state, pincode, candidate_ID
- 3) Pincode → city, state
- 4) city → state
- 5) age → D.O.B.

Login

- 1) candidate_ID → password

FUNCTIONAL DEPENDENCIES

Centre

- 1) Centre_ID —→ Capacity, No. of invigilator, city, state, pincode
- 2) Pincode → City, State
- 3) city → state

INVIGILATOR

- 1) Invigilator_ID —→ Name, Gender

FUNCTIONAL DEPENDENCIES

Organizer

1) Organizer_ID → Name,E-Mail,Password

Response

1) Response_ID → No. of Wrong_Answers,Not_Attempted,
No. of_Correct Answer

FUNCTIONAL DEPENDENCIES

Result

- 1) Candidate ID → Status, Percentage, Total Marks
- 2) Total Marks → Percentage, Status
- 3) Percentage → Status

Paper

- 1) Paper_ID → Total-Marks, Total Questions

Examination Details

- 1) Candidate_ID → Slot, Date

NORMALIZATION

- A relational schema is considered to be in 1NF if and only if it contains all atomic values. Therefore, multi-valued or composite attributes, if present, must be broken down into single-valued attributes
- A relational schema is considered to be in 2NF if it is in 1NF and all the non-prime attributes are fully functionally dependent on the primary key.
- A relational schema is considered to be in 3NF if it is in 2NF and does not contain any transitive dependency for non prime attributes.
- A relational schema is considered to be in BCNF if it is in 3NF and for every functional dependency $X \rightarrow Y$, X is a super key.

NORMALIZATION

1) candidate:

Name is not atomic, so it is not 1NF

$\text{candidate_id} \rightarrow \text{first_name, last_name, gender, e-mail, D.O.B., aadhar_no, examination-year, city, pin_code, state}$

$\{\text{city} \rightarrow \text{state} ; \text{pin_code} \rightarrow \text{city, state}\} \rightarrow$ Transitive dependency

$\{\text{candidate_id} \rightarrow \text{city} ; \text{city} \rightarrow \text{state}\} \rightarrow$ Transitive dependency

1NF: Since all the attributes are atomic and all have unique names, so it is 1NF

2NF: Since there is no partial dependency exist among the functional dependency in the relation, so it is in 2NF

3NF: Due to presence of transitive dependency among the attributes in functional dependency, it is not in 3NF.

NORMALIZATION

candidate:

candidate

candidate_address

candidate

(candidate_id, first_name, last_name, gender, e-mail, D.O.B., aadhar_no, examination-year, pin_code)

3NF: No Transitive Dependencies so it is in 3NF

BCNF: Both aadhar_no and candidate_id are superkeys so it is also in BCNF

NORMALIZATION

candidate_address

(city, state, pin_code)

{city → state ; pin_code → city, state} → Transitive dependency

3NF: Not in 3NF due to presence of transitive dependency

candidate_address { C1 (pin_code, city) *[It is in 3NF]*
 C2 (city, state) *[It is in 3NF]*

BCNF: Both are superkeys so it is also in BCNF

NORMALIZATION

2) candidate_age

(D.O.B., age)

1NF : Since all the attributes are atomic and all have unique names, so it is 1NF

2NF : Since there is no partial dependency exist among the functional dependency in the relation, so it is in 2NF

3NF : Due to no presence of transitive dependency among the attributes in functional dependency, it is in 3NF.

BCNF : Since no non-prime attribute defines a prime attribute in the new schemas generated above, so it is in BCNF.

NORMALIZATION

3) candidate_phone

(candidate_id, phone_no)

1NF: Since all the attributes are atomic and all have unique names, so it is 1NF

2NF: Since there is no partial dependency exist among the functional dependency in the relation, so it is in 2NF

3NF: Due to no presence of transitive dependency among the attributes in functional dependency, it is in 3NF.

BCNF: Since no non-prime attribute defines a prime attribute in the new schemas generated above, so it is in BCNF.

NORMALIZATION

4) centre

(centre_id, no.of_invigilator, city, state, pin_code, capacity):

1NF: All are in atomic state

2NF: There is no partial dependency

$\text{centre_id} \rightarrow \text{capacity, no.of_invigilator, city, state, pin_code}$

Transitive dependencies:

{ $\text{city} \rightarrow \text{state}$; $\text{pin_code} \rightarrow \text{city, state}$ }

{ $\text{centre_id} \rightarrow \text{city}$; $\text{city} \rightarrow \text{state}$ }

NORMALIZATION

centre:

centre

centre_address

centre

(centre_id, capacity, no.of_invigilator, pin_code)

3NF: No Transitive Dependencies so it is in 3NF

BCNF: It is in BCNF as $\alpha \rightarrow \beta$ and α is superkey

NORMALIZATION

centre_address

(city, state, pin_code)

{city → state ; pin_code → city, state} → Transitive dependency

3NF: Not in 3NF due to presence of transitive dependency

centre_address {
 C_add1 (pin_code, city) *[It is in 3NF]*
 C_add2 (city, state) *[It is in 3NF]*

BCNF: Both are superkeys so it is also in BCNF

NORMALIZATION

5) invigilator:

#invigilator_id → name, gender

name is not in atomic state so it is not in 1NF

#invigilator_id → first_name, last_name, gender

1NF: Since all the attributes are atomic and all have unique names, so it is 1NF

2NF: Since there is no partial dependency exist among the functional dependency in the relation, so it is in 2NF

3NF: Due to no presence of transitive dependency among the attributes in functional dependency, it is in 3NF.

BCNF: Since no non-prime attribute defines a prime attribute in the new schemas generated above, so it is in BCNF.

NORMALIZATION

6) organizer:

$\#organizer_id \rightarrow name, gender$

name is not in atomic state so it is not in 1NF

$\#organizer_id \rightarrow first_name, last_name, gender$

1NF: Since all the attributes are atomic and all have unique names, so it is 1NF

2NF: Since there is no partial dependency exist among the functional dependency in the relation, so it is in 2NF

3NF: Due to no presence of transitive dependency among the attributes in functional dependency, it is in 3NF.

BCNF: Since no non-prime attribute defines a prime attribute in the new schemas generated above, so it is in BCNF.

NORMALIZATION

7) result:

1NF : Since all the attributes are atomic and all have unique names, so it is 1NF

2NF : Since there is no partial dependency exist among the functional dependency in the relation, so it is in 2NF

#candidate_id → total_marks, status, percentage

#total_marks → status, percentage

#percentage → status

} Transitive dependency

3NF : Due to presence of transitive dependency among the attributes in functional dependency, it is not in 3NF.

NORMALIZATION

result

result

result2

result

(candidate_id,total_marks)

3NF: No Transitive Dependencies so it is in 3NF

BCNF: It is in BCNF as $\alpha \rightarrow \beta$ and α is superkey

NORMALIZATION

result2

(total_marks, percentage, status)

{percentage → status ; total_marks → percentage,status} → Transitive dependency

3NF: Not in 3NF due to presence of transitive dependency

result2 { result21 (total_marks, percentage) *[It is in 3NF]*
 result22 (percentage, status) *[It is in 3NF]*

BCNF: Both are superkeys so it is also in BCNF

NORMALIZATION

8) response

(response_id, no.of wrong_answers, no.of correct_answers, candidate_id)

response_id → not_attempted, no.of correct_answers, no.of wrong_answers

1NF: Since all the attributes are atomic and all have unique names, so it is 1NF

2NF: Since there is no partial dependency exist among the functional dependency in the relation, so it is in 2NF

3NF: Due to no presence of transitive dependency among the attributes in functional dependency, it is in 3NF.

BCNF: Since no non-prime attribute defines a prime attribute in the new schemas generated above, so it is in BCNF.

NORMALIZATION

9) paper

(paper_id, total_marks, total_questions, organizer_id)

paper_id → total_marks, total_questions

1NF: Since all the attributes are atomic and all have unique names, so it is 1NF

2NF: Since there is no partial dependency exist among the functional dependency in the relation, so it is in 2NF

3NF: Due to no presence of transitive dependency among the attributes in functional dependency, it is in 3NF.

BCNF: Since no non-prime attribute defines a prime attribute in the new schemas generated above, so it is in BCNF.

NORMALIZATION

10) login

(candidate_id, password)

candidate_id → password

1NF : Since all the attributes are atomic and all have unique names, so it is 1NF

2NF : Since there is no partial dependency exist among the functional dependency in the relation, so it is in 2NF

3NF : Due to no presence of transitive dependency among the attributes in functional dependency, it is in 3NF.

BCNF : Since no non-prime attribute defines a prime attribute in the new schemas generated above, so it is in BCNF.

NORMALIZATION

11) examination_details

(candidate_id, slot, date)

$\text{candidate_id} \rightarrow \text{slot, date}$

1NF : Since all the attributes are atomic and all have unique names, so it is 1NF

2NF : Since there is no partial dependency exist among the functional dependency in the relation, so it is in 2NF

3NF : Due to no presence of transitive dependency among the attributes in functional dependency, it is in 3NF.

BCNF : Since no non-prime attribute defines a prime attribute in the new schemas generated above, so it is in BCNF.

NORMALIZATION

12) provided_in (centre_id, paper_id)

1NF : Since all the attributes are atomic as well as all have unique names, so it is 1NF

2NF : Since there is no partial dependency exist among the functional dependency in the relation, so it is in 2NF

3NF : Due to no presence of transitive dependency among the attributes in functional dependency, it is in 3NF.

BCNF : Since no non-prime attribute defines a prime attribute in the new schemas generated above, so it is in BCNF.

OUR DATABASE

Made in MySQL Shell

```
desc candidate;
```

Field	Type	Null	Key	Default	Extra
candidate_ID	int	NO	PRI	NULL	
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	NO		NULL	
gender	varchar(6)	NO		NULL	
E_email	varchar(40)	NO		NULL	
Date_of_birth	date	NO		NULL	
pincode	int	NO		NULL	
Examination_year	int	NO		NULL	
centre_ID	int	NO	MUL	NULL	
Aadhar_no	bigint	YES		NULL	

```
select * from candidate;
```

```
desc candidate_address_1;
```

	Field	Type	Null	Key	Default	Extra
▶	pincode	int	NO	PRI	HULL	
	city	varchar(20)	NO		HULL	

```
select * from candidate_address_1;
```

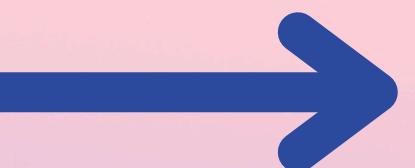


	pincode	city
▶	284012	Indore
	354060	Lucknow
	379200	Mumbai
	384200	Jasmedhpur
	396060	Mumbai
	412615	Chandigarh
	414833	Pune
	419706	Srinagar
	426822	Ahmedabad
	484060	Kanpur
	486000	Gwalior
	486081	Chennai
	486522	Bhopal
	496250	Gwalior
	498605	Warangal
	510102	Jaipur
	510678	Raipur
	514173	Bangalore
	518290	Shillong
	521068	Delhi
	543861	Hyderabad
	548206	Patna
	567820	Ara
*	HULL	HULL

```
desc candidate_address_2;
```

	Field	Type	Null	Key	Default	Extra
▶	city	varchar(20)	NO	PRI	NULL	
	state	varchar(20)	NO		NULL	

```
select * from candidate_address_2;
```

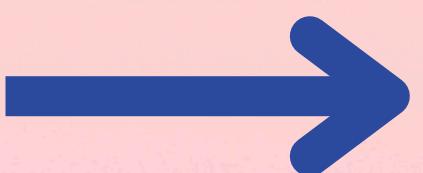


	city	state
▶	Ahmedabad	Gujarat
	Ara	Bihar
	Bangalore	Karnataka
	Bhopal	Madhya Pradesh
	Chandigarh	Punjab
	Chennai	Tamil Nadu
	Delhi	Delhi
	Gwalior	Madhya Pradesh
	Hyderabad	Telangana
	Indore	Madhya Pradesh
	Jaipur	Rajasthan
	Jamshedhpur	Jharkhand
	Kanpur	Uttar Pradesh
	Lucknow	Uttar Pradesh
	Mumbai	Maharashtra
	Patna	Bihar
	Pune	Maharashtra
	Raipur	Telangana
	Shillong	Meghalaya
	Sringar	Jammu and Kas...
	Warangal	Telangana
*	NULL	NULL

```
desc candidate_age;
```

Field	Type	Null	Key	Default	Extra
date_of_birth	date	YES		NULL	
Age	int	YES		NULL	

```
select * from candidate_age;
```

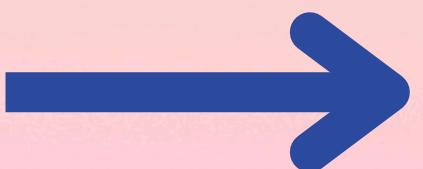


	date_of_birth	Age
▶	2003-04-16	17
	2003-06-12	17
	2002-08-04	18
	2003-06-05	17
	2003-07-21	17
	2003-04-20	17
	2003-09-25	17
	2002-03-12	18
	2003-05-11	17
	2003-08-04	17
	2003-11-14	17
	2002-08-06	18
	2003-07-04	17
	2003-02-16	17
	2003-07-27	17
	2003-09-21	17
	2003-12-16	17
	2003-02-03	17
	2002-03-07	18
	2003-04-13	17
	2003-05-17	17
	2003-07-16	17
	2003-07-19	17
	2003-09-12	17
	2002-05-14	18
	2003-05-04	17
	2003-07-09	17
	2003-05-17	17

```
desc candidate_phoneno;
```

Field	Type	Null	Key	Default	Extra
phone_no	bigint	NO	PRI	NULL	
candidate_ID	int	NO		NULL	

```
select * from candidate_phoneno;
```



	phone_no	candidate_ID
▶	5927895420	20000
	6259797250	30001
	6354294320	10002
	6373334290	10006
	6374599650	10003
	6543904808	40000
	6597654100	10007
	6678794320	3 10007
	6867296860	20002
	6945083290	40000
	7864829801	40001
	7868394884	40004
	7908106006	20006
	8109604860	20003
	8267652090	10001
	8268624520	40005
	8297604290	30000
	8374389439	10000
	8376506501	20001
	8399377009	40003
	8762506438	30002
	8766205493	20005
	8897943276	10005
	8905400680	30003
	8909806333	30005
	9276354290	10001
	9296599290	20002
	9296876540	20004
	9309309300	10008
	9394806430	30003
	9424685439	10004
	9656426106	40002
	9912093609	10005
	NULL	NULL

```
desc centrej
```

Field	Type	Null	Key	Default	Extra
centre_ID	int	NO	PRI	NULL	
capacity	int	NO		NULL	
No_of_invigilator	int	NO		NULL	
pincode	int	NO		NULL	

```
select * from centrej
```



	centre_ID	capacity	No_of_invigilator	pincode
▶	16542	740	24	321612
	18516	830	27	407600
	19864	640	20	386770
	20888	800	26	418224
	22400	850	28	397840
	24652	500	18	501511
	27650	920	31	501511
	28666	880	30	401265
	29716	550	18	523164
	31320	800	25	464822
	32966	720	22	496420
	35071	900	30	410880
	38200	700	21	410880
	41420	750	24	540600
	49606	820	26	474206
	NONE	NONE	NONE	NONE

```
desc centre_address_1;
```

	Field	Type	Null	Key	Default	Extra
▶	pincode	int	NO	PRI	NULL	
	city	varchar(20)	NO		NULL	

```
select * from centre_address_1;
```



	pincode	city
▶	321612	Ahmedabad
	386770	Raipur
	397840	Chandigarh
	401265	Faridabad
	407600	Secunderabad
	410880	Indore
	418224	Gwalior
	464822	Patna
	474206	Gwalior
	496420	Bangalore
	501511	Hyderabad
	523164	Mumbai
	540600	Lucknow
	NULL	NULL

```
desc centre_address_2;
```

Field	Type	Null	Key	Default	Extra
city	varchar(20)	NO	PRI	NULL	
state	varchar(20)	NO		NULL	

```
select * from centre_address_2;
```



city	state
Ahmedabad	Gujarat
Bangalore	Karnataka
Chandigarh	Punjab
Faridabad	Haryana
Gwalior	Madhya Pradesh
Hyderabad	Telangana
Indore	Madhya Pradesh
Lucknow	Uttar Pradesh
Mumbai	Maharashtra
Patna	Bihar
Raipur	Chhattisgarh
Secundera...	Telangana

```
desc examination_details;
```

Field	Type	Null	Key	Default	Extra
slot	varchar(30)	NO		HULL	
date	date	NO		HULL	
candidate_ID	int	NO	PRI	HULL	

```
select * from examination_details;
```



slot	date	candidate_ID
Morning(9-12)	2020-01-21	10000
Evening(3-6)	2020-01-21	10001
Evening(3-6)	2020-01-22	10002
Evening(3-6)	2020-01-22	10003
Morning(9-12)	2020-01-21	10004
Morning(9-12)	2020-01-22	10005
Evening(3-6)	2020-01-21	10006
Morning(9-12)	2019-01-21	10007
Morning(9-12)	2020-01-22	10008
Morning(9-12)	2020-01-24	20000
Evening(3-6)	2020-01-24	20001
Morning(9-12)	2019-01-25	20002
Evening(3-6)	2020-01-24	20003
Morning(9-12)	2020-01-24	20004
Evening(3-6)	2020-01-24	20005
Morning(9-12)	2020-01-25	20006
Morning(9-12)	2020-01-27	30000
Evening(3-6)	2020-01-27	30001
Morning(9-12)	2019-01-28	30002
Morning(9-12)	2020-01-27	30003
Evening(3-6)	2020-01-27	30004
Morning(9-12)	2020-01-29	30005
Morning(9-12)	2020-01-29	40000
Evening(3-6)	2020-01-29	40001
Morning(9-12)	2019-01-30	40002
Evening(3-6)	2020-01-29	40003
Morning(9-12)	2020-01-29	40004
Morning(9-12)	2020-01-30	40005
	NULL	NULL

```
desc invigilator;
```

Field	Type	Null	Key	Default	Extra
invigilator_ID	int	NO	PRI	NULL	
first_name	varchar(40)	NO		NULL	
last_name	varchar(50)	NO		NULL	
gender	varchar(6)	NO		NULL	
centre_ID	int	NO	MUL	NULL	
organizer_id	int	NO	MUL	NULL	

```
select * from invigilator;
```



invigilator_ID	first_name	last_name	gender	centre_ID	organizer_id
101	Shivam	Singh	Male	24652	1100
102	Harsh	Dev	male	31320	1100
103	Karuna	Singhal	Female	32966	1101
104	Raghavendra	Reddy	male	29716	1101
105	Deepak	Kriplani	male	31320	1100
106	Dhruv	Sanghvi	male	41420	1100
107	Rakesh	Bodapatla	male	49606	1100
108	Virat	Kohli	male	19864	1103
109	Aman	Tiwari	male	22400	1103
110	Harsh	Vardhan	male	28666	1101
111	Aditya	Garg	male	35071	1102
112	Vinay	Kumar	male	16542	1102
113	Anushka	Sharma	Female	18516	1102
114	Rupali	Verma	Female	20888	1100
115	Keerti	Sharma	Female	27650	1101

```
desc login;
```

Field	Type	Null	Key	Default	Extra
candidate_ID	int	NO	PRI	NULL	
password	varchar(30)	NO		NULL	

```
select * from login;
```



candidate_ID	password
10000	AAro092
10001	Abhi123
10002	Abhi234
10003	kart345
10004	Divy456
10005	Gaga567
10006	Hima678
10007	Musk780
10008	sake800
20000	Abhi012
20001	Akshi123
20002	Aman234
20003	Deep345
20004	Gaur456
20005	Mith560
20006	Riya600
30000	Abhi012
30001	Aksh123
30002	Aman234
30003	Gaya345
30004	chin450
30005	Hars500
40000	Divi012
40001	Aksh123
40002	Ipsa234
40003	Veda345
40004	Sake450
40005	Kris500
NULL	NULL

```
desc organizer;
```

Field	Type	Null	Key	Default	Extra
name	varchar(30)	NO		NULL	
E_email	varchar(40)	NO		NULL	
password	varchar(30)	NO		NULL	
organizer_ID	int	NO	PRI	NULL	

```
select * from organizer;
```

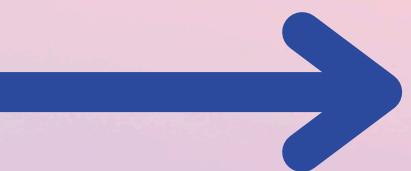


name	E_email	password	organizer_ID
NTA	nta@gmail.com	nta@365	1100
ICAR	icar@gmail.com	icar@1980	1101
UGC	ugc@gmail.com	ugc@963	1102
MLT	mlt@gmail.com	mlt@248	1103
	HULL	HULL	HULL

```
desc paper;
```

Field	Type	Null	Key	Default	Extra
paper_ID	int	NO	PRI	NULL	
total_marks	int	NO		NULL	
total_question	int	NO		NULL	
organizer_ID	int	NO	MUL	NULL	

```
select * from paper;
```

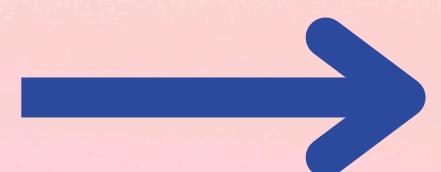


paper_ID	total_marks	total_question	organizer_ID
601	300	75	1103
602	300	75	1103
603	300	75	1103
701	360	75	1102
702	300	75	1102
703	300	75	1102
801	300	75	1101
802	300	75	1101
803	300	75	1101
901	300	75	1100
902	300	75	1100
903	300	75	1100

desc provided_in;

Field	Type	Null	Key	Default	Extra
centre_ID	int	NO		NULL	
paper_ID	int	NO		NULL	

select * from provided_in;

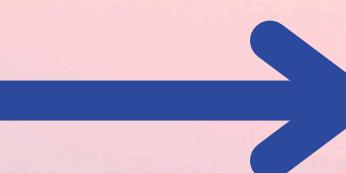


centre_ID	paper_ID
24652	901
31320	902
29716	801
32966	802
41420	903
49606	903
38200	601
19864	602
22400	603
28666	803
35071	701
16542	702
18516	703
20888	901
27650	802

desc response;

Field	Type	Null	Key	Default	Extra
response_ID	int	NO	PRI	HULL	
Not_attempted	int	NO		HULL	
No_of_correct_answers	int	NO		HULL	
no_of_wrong_answers	int	NO		HULL	
candidate_ID	int	NO	MUL	HULL	

```
select * from response;
```



response_ID	Not_attempted	No_of_correct_answers	no_of_wrong_answers	candidate_ID
440	20	45	10	10000
441	18	42	15	10001
442	20	42	13	10002
443	20	50	5	10003
444	15	55	5	10004
445	25	38	12	10005
446	34	26	15	10006
447	40	20	15	10007
448	36	14	25	10008
540	25	39	11	20000
541	31	28	16	20001
542	35	20	20	20002
543	26	24	25	20003
544	24	17	34	20004
545	26	28	21	20005
546	20	40	15	20006
640	13	51	11	30000
641	13	55	7	30001
642	8	60	7	30002
643	20	38	17	30003
644	25	28	22	30004
645	25	30	20	30005
740	23	37	15	40000
741	9	60	6	40001
742	15	42	18	40002
743	17	36	22	40003
744	20	37	18	40004
745	21	32	22	40005
	NUL	NUL	NUL	NUL

```
desc result_1;
```

Field	Type	Null	Key	Default	Extra
candidate_ID	int	NO	PRI	NULL	
total_marks	int	NO		NULL	

```
select * from result_1;
```



candidate_ID	total_marks
10000	180
10001	165
10002	165
10003	200
10004	220
10005	150
10006	100
10007	180
10008	50
20000	150
20001	110
20002	75
20003	90
20004	66
20005	108
20006	159
30000	199
30001	225
30002	240
30003	150
30004	112
30005	198
40000	148
40001	235
40002	166
40003	142
40004	148
40005	126
	NUL

```
desc result_2_1;
```

Field	Type	Null	Key	Default	Extra
total_marks	int	NO	PRI	NULL	
percentage	float	NO		NULL	

```
select * from result_2_1;
```



total_marks	percentage
50	16.67
66	22
75	25
90	30
100	33.33
108	36
110	36.67
112	37.33
126	42
142	47.33
148	49.33
150	50
165	55
166	55.33
180	60
198	66
199	66.33
200	66.67
220	73.33
225	75
235	78.33
240	80

Field	Type	Null	Key	Default	Extra	percentage	status
percentage	float	NO	PRI	NULL		16.67	NQ
status	varchar(9)	NO		NULL		22	NQ
						25	NQ
						30	NQ
						33.33	Q
						35	Q
						36.67	Q
						37.33	Q
						42	Q
						47.33	Q
						49.33	Q
						50	Q
						52	Q
						53	Q
						55	Q
						55.33	Q
						60	Q
						66	Q
						66.33	Q
						66.67	Q
						73.33	Q
						75	Q
						78.33	Q
						80	Q

```
select * from result_2_2;
```



RELATIONAL ALGEBRA

Done in RelaX

RELATIONAL ALGEBRA

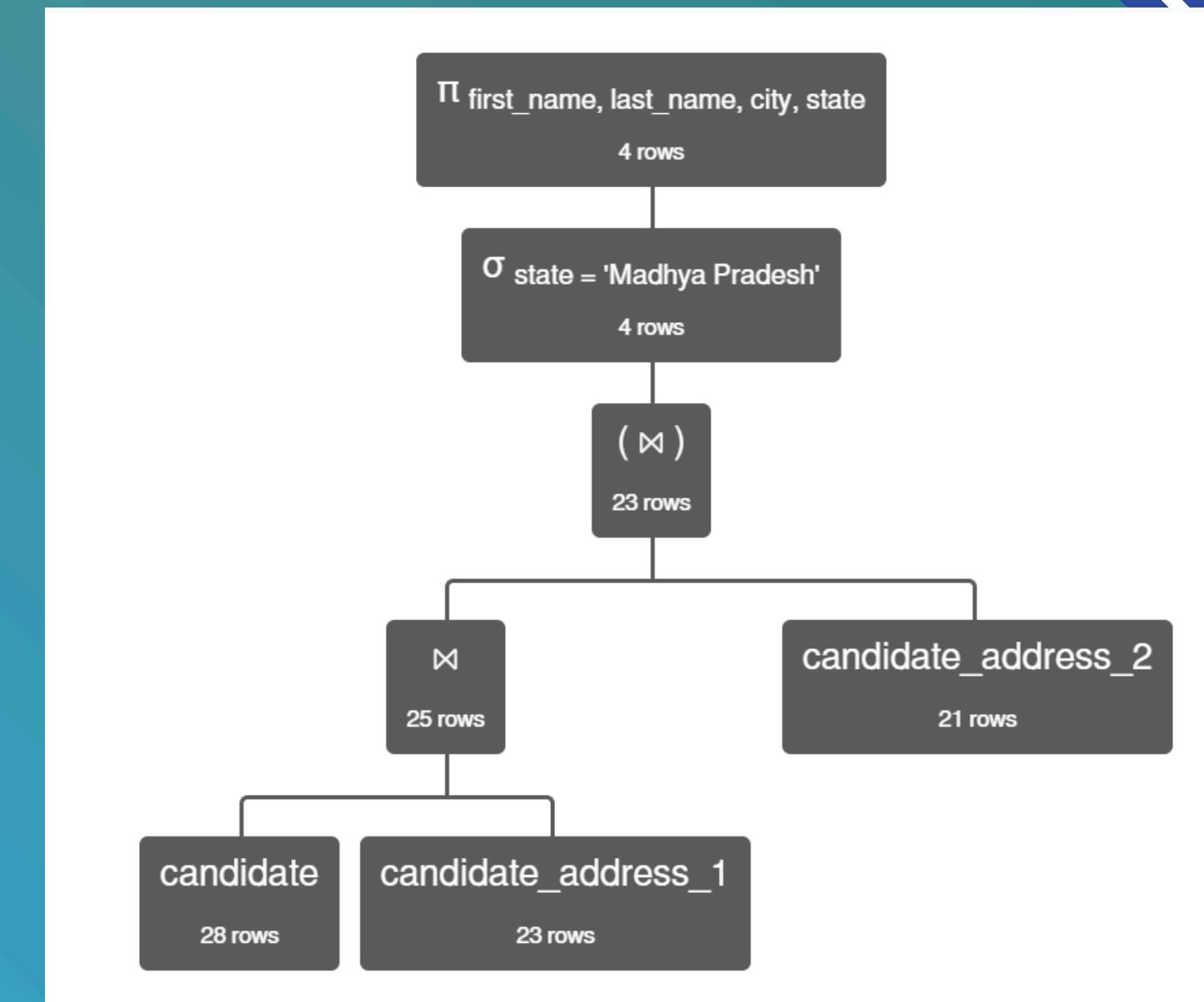
1) Find the Candidates name, state and city
whose state is Madhya pradesh.

```

$$\Pi \text{first\_name, last\_name, city, state} (\sigma \text{state} = \text{'Madhya Pradesh'} ((\text{candidate} \bowtie \text{candidate\_address\_1}) \bowtie \text{candidate\_address\_2}))$$

```

Execution time: 2 ms



candidate.first_name	candidate.last_name	candidate_address_1.city	candidate_address_2.state
'Aaroh'	'Srivastava'	'Gwalior'	'Madhya Pradesh'
'Abhinav'	'Gupta'	'Indore'	'Madhya Pradesh'
'Abhijeet'	'Singh'	'Gwalior'	'Madhya Pradesh'
'Chinmay'	'Gupta'	'Gwalior'	'Madhya Pradesh'

RELATIONAL ALGEBRA

Q2) Find candidate Id who gave more than (or equal to) 30 correct answers and less than (or equal to) 17 wrong answers

```
Π candidate_ID ( σ No_of_correct_answers ≥ 30 and no_of_wrong_answers ≤  
17 ( response ) )
```

Execution time: 4 ms

```
Π candidate_ID  
14 rows
```

```
σ No_of_correct_answers ≥ 30 and no_of_wrong_answers ≤ 17  
14 rows
```

```
response  
28 rows
```

response.candidate_ID

10000

10001

10002

10003

10004

10005

20000

20006

30000

30001

response.candidate_ID

30002

30003

40000

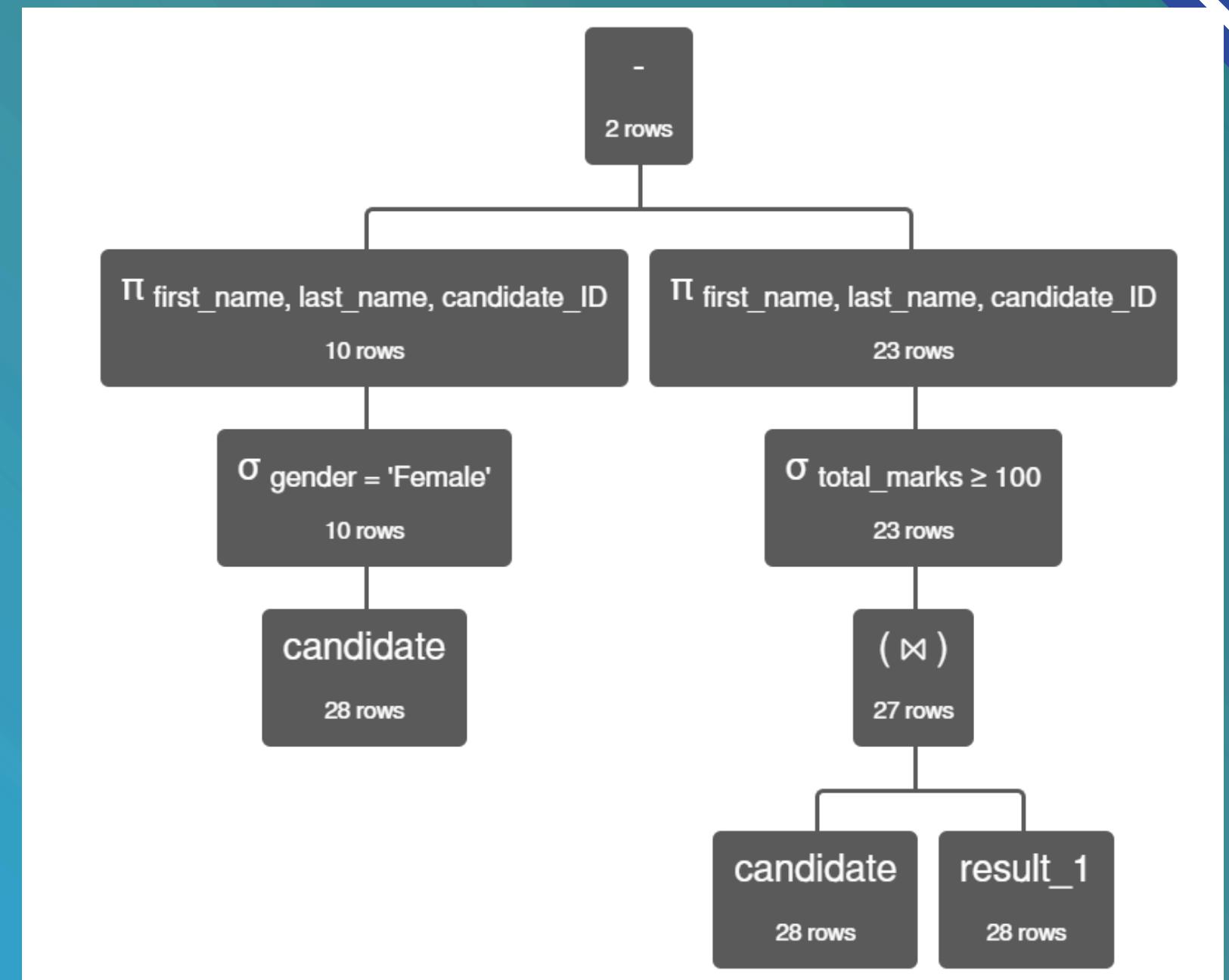
40001

RELATIONAL ALGEBRA

Q3) Find the name of female candidates and Candidate -ID who got less than 100 marks in the exam

```
 $\Pi_{first\_name, last\_name, candidate\_ID} ( \sigma_{gender = 'Female'} ( candidate ) )$ 
-  $\Pi_{first\_name, last\_name, candidate\_ID} ( \sigma_{total\_marks \geq 100} ( candidate \bowtie result\_1 ) )$ 
```

Execution time: 4 ms



candidate.first_name	candidate.last_name	candidate.candidate_ID
-----------------------------	----------------------------	-------------------------------

'Saketh'	'Pothula'	10008
----------	-----------	-------

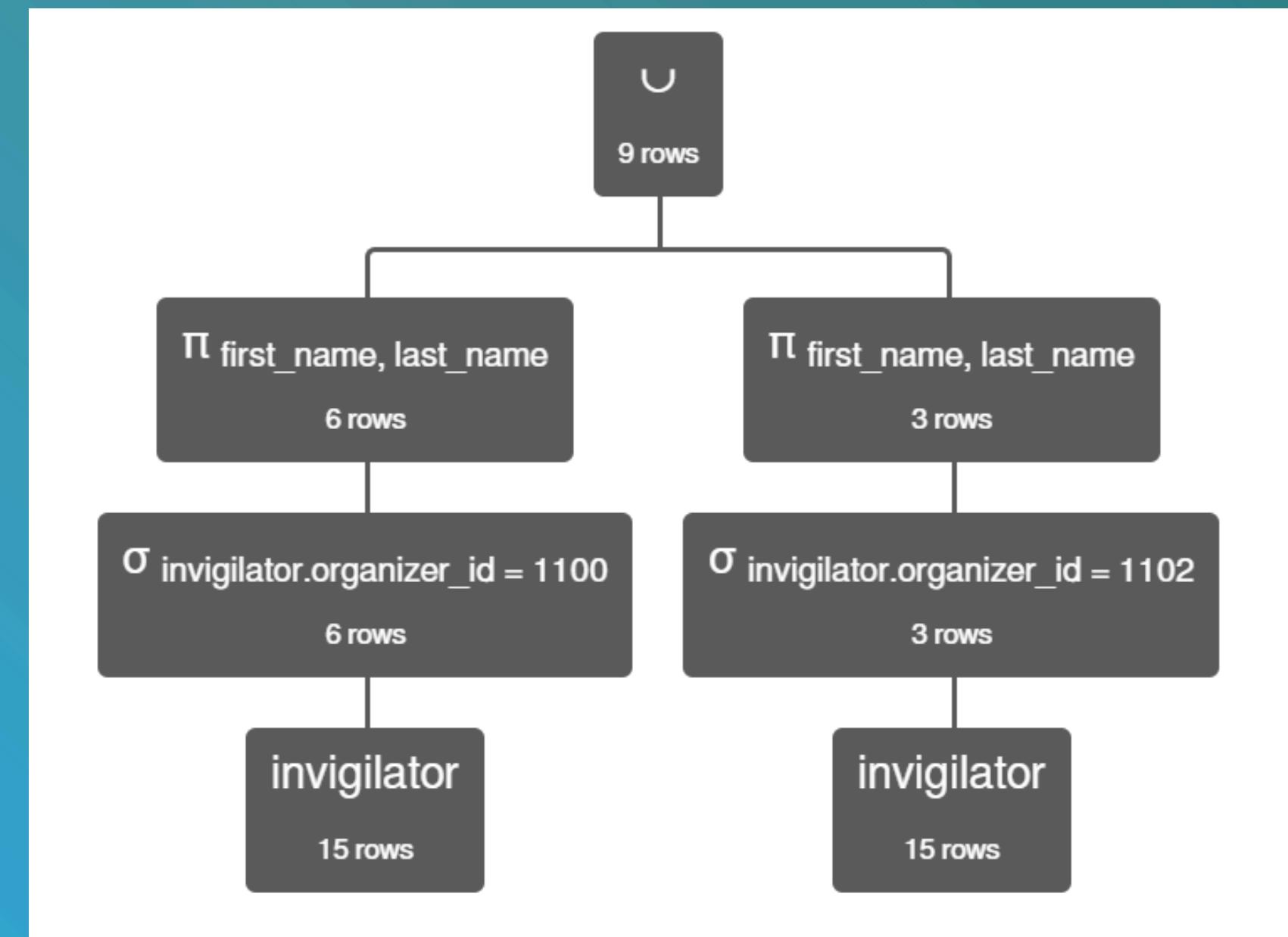
'Deepti'	'Arya'	20003
----------	--------	-------

RELATIONAL ALGEBRA

Q4) Find the name of invigilators who have been appointed by NTA and UGC.

```
 $\Pi_{\text{first\_name}, \text{last\_name}} (\sigma_{\text{invigilator.organizer\_id} = 1100} (\text{invigilator})) \cup$ 
 $\Pi_{\text{first\_name}, \text{last\_name}} (\sigma_{\text{invigilator.organizer\_id} = 1102} (\text{invigilator}))$ 
```

Execution time: 4 ms



invigilator.first_name	invigilator.last_name
'Shivam'	'Singh'
'Harsh'	'Dev'
'Deepak'	'Kriplani'
'Dhruv'	'Sanghvi'
'Rakesh'	'Bodapatla'
'Rupali'	'Verma'
'Aditya'	'Garg'
'Vinay'	'Kumar'
'Anushka'	'Sharma'

RELATIONAL ALGEBRA

Q5) Find the name of candidates with total marks and percentage who have been qualified

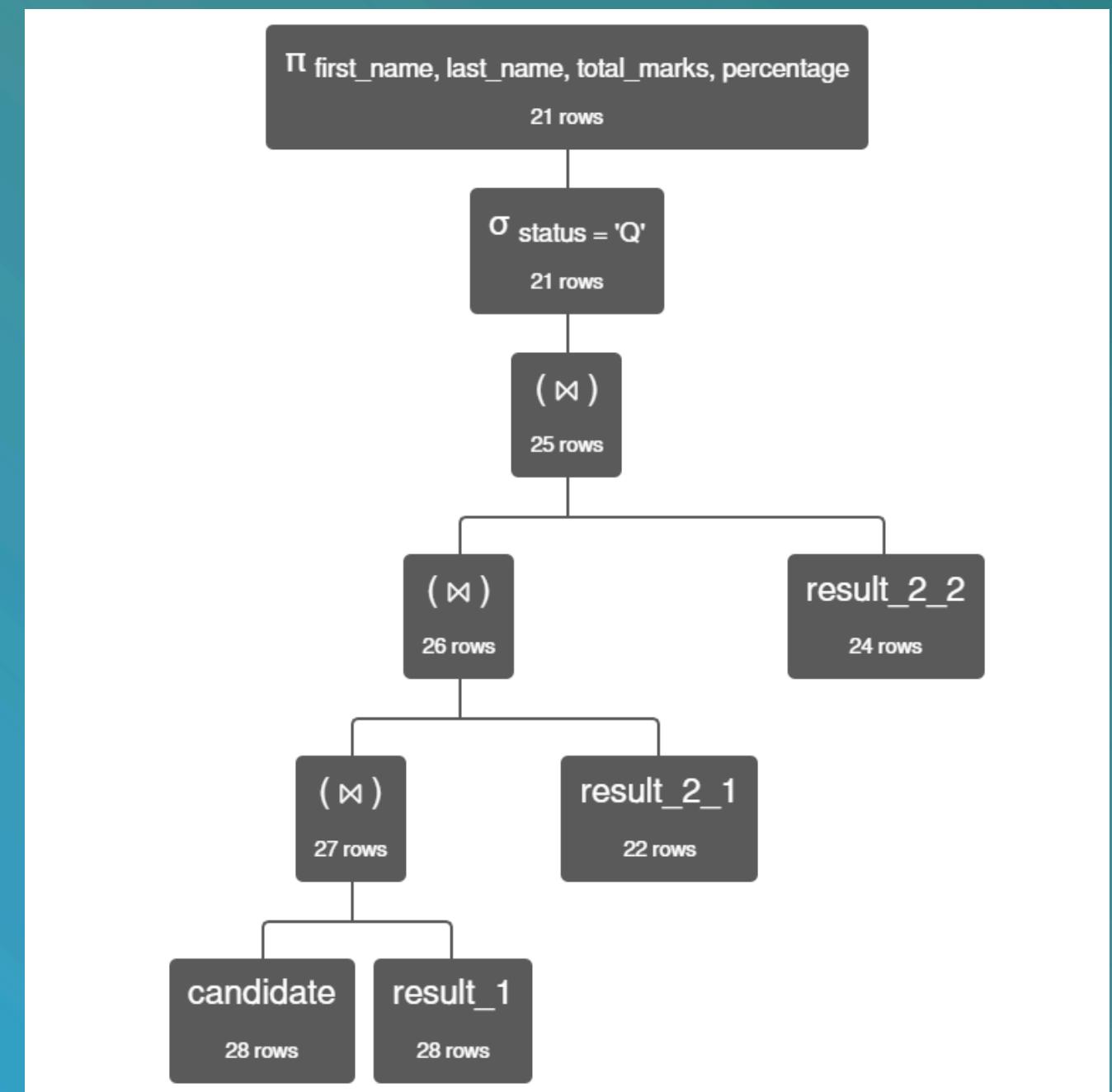
```

$$\Pi_{\text{first\_name, last\_name, total\_marks, percentage}} (\sigma_{\text{status} = 'Q'} (( ( ($$
  

$$\text{candidate} \bowtie \text{result\_1} ) \bowtie \text{result\_2\_1} ) \bowtie \text{result\_2\_2} ) )$$

```

Execution time: 4 ms



candidate.first_name	candidate.last_name	result_1.total_marks	result_2_1.percentage
'Aaroh'	'Srivastava'	180	60
'Abhinav'	'Gupta'	165	55
'Abhishek'	'Singh'	165	55
'karthik'	'Bangu'	200	66.67
'Divyansh'	'Pathak'	220	73.33
'Gagandeep'	'Singh'	150	50
'Himanshi'	'Singh'	100	33.33
'Muskan'	'Debnath'	180	60
'Akshitha'	'Mittapally'	110	36.67
'Abhijeet'	'Singh'	199	66.33

candidate.first_name	candidate.last_name	result_1.total_marks	result_2_1.percentage
'Akshat'	'Jain'	225	75
'Aman'	'Kumar'	240	80
'Gayathri'	'Bhukya'	150	50
'Chinmay'	'Gupta'	112	37.33
'Harsh'	'Dalwadi'	198	66
'Divjot'	'Singh'	148	49.33
'Akshaya'	'Gulla'	235	78.33
'Ipsa'	'Badoniya'	166	55.33
'Vedant'	'Kadam'	142	47.33
'Saketh'	'Satya'	148	49.33

candidate.first_name	candidate.last_name	result_1.total_marks	result_2_1.percentage
'Krish'	'Gujarathi'	126	42

SQL QUERIES

Done in MySQL Workbench

SQL QUERIES

1) Find the total number of candidates who have been qualified in year 2020.

- ```
select count(candidate.candidate_ID) as Total_Qualified
from candidate,result_1,result_2_1,result_2_2
where candidate.candidate_ID=result_1.candidate_ID
and result_1.total_marks=result_2_1.total_marks
and result_2_1.percentage=result_2_2.percentage
and status='Q' and Examination_year=2020;
```

|   | Total_Qualified |
|---|-----------------|
| ▶ | 19              |

# SQL QUERIES

2) Find centre details with their ID's,capacity,no. of invigilators and location of centre having No\_of\_Invigilator >20

```
select centre.centre_ID,capacity,No_of_Invigilator,
centre.pincode,centre_address_1.city,state from centre
natural join centre_address_1 natural join centre_address_2
where No_of_invigilator>20;
```

| centre_ID | capacity | No_of_Invigilator | pincode | city         | state          |
|-----------|----------|-------------------|---------|--------------|----------------|
| 16542     | 740      | 24                | 321612  | Ahmedabad    | Gujarat        |
| 18516     | 830      | 27                | 407600  | Secunderabad | Telangana      |
| 20888     | 800      | 26                | 418224  | Gwalior      | Madhya Pradesh |
| 22400     | 850      | 28                | 397840  | Chandigarh   | Punjab         |
| 27650     | 920      | 31                | 501511  | Hyderabad    | Telangana      |
| 28666     | 880      | 30                | 401265  | Faridabad    | Haryana        |
| 31320     | 800      | 25                | 464822  | Patna        | Bihar          |
| 32966     | 720      | 22                | 496420  | Bangalore    | Karnataka      |
| 35071     | 900      | 30                | 410880  | Indore       | Madhya Pradesh |
| 38200     | 700      | 21                | 410880  | Indore       | Madhya Pradesh |
| 41420     | 750      | 24                | 540600  | Lucknow      | Uttar Pradesh  |
| 49606     | 820      | 26                | 474206  | Gwalior      | Madhya Pradesh |

# SQL QUERIES

3) Find first name, last name, gender,e-mail,examination-year and aadhar no. of candidate who have attended exam on 2020/1/21 morning shift (9-12)

```
select first_name, last_name, gender, E_email, Examination_year, Aadhar_no
from candidate where candidate_ID in
(select candidate_ID from examination_details
where slot='Morning(9-12)' and date="2020/1/21");
```

| first_name | last_name  | gender | E_email            | Examination_year | Aadhar_no    |
|------------|------------|--------|--------------------|------------------|--------------|
| Aaroh      | Srivastava | Male   | aaroh@gmail.com    | 2020             | 165240602196 |
| Divyansh   | Pathak     | Male   | Divyansh@gmail.com | 2020             | 176025604289 |

# SQL QUERIES

4) Find names of city and state where city contains 'pur' string in candidate's address

```
select * from candidate_address_2
where city like '%pur%';
```

| city        | state         |
|-------------|---------------|
| Jaipur      | Rajasthan     |
| Jamshedhpur | Jharkhand     |
| Kanpur      | Uttar Pradesh |
| Raipur      | Telangana     |
| HULL        | NULL          |

# SQL QUERIES

5) Find the candidate's name with their No. of correct answers who have attempted more than 30 and less than 45 correct answers in the exam.

```
select first_name, last_name, No_of_correct_answers
from candidate, response where
candidate.candidate_ID=response.candidate_ID
and No_of_correct_answers between 30 and 45
```

| first_name | last_name  | No_of_correct_answers |
|------------|------------|-----------------------|
| Aaroh      | Srivastava | 45                    |
| Abhinav    | Gupta      | 42                    |
| Abhishek   | Singh      | 42                    |
| Gagandeep  | Singh      | 38                    |
| Abhishek   | Sinha      | 39                    |
| Riya       | Jain       | 40                    |
| Gayathri   | Bhukya     | 38                    |
| Harsh      | Dalwadi    | 30                    |
| Divjot     | Singh      | 37                    |
| Ipsa       | Badoniya   | 42                    |
| Vedant     | Kadam      | 36                    |
| Saketh     | Satya      | 37                    |
| Krish      | Gujarathi  | 32                    |



A photograph of a person's hand holding a white ceramic mug filled with dark coffee. The hand is positioned on the left side of the frame. In the background, a silver laptop is open, showing its keyboard and trackpad. The laptop is resting on a light-colored wooden surface. A large, semi-transparent blue diamond shape is overlaid on the center of the image. Inside this blue shape, the words "THANK YOU" are written in a bold, white, sans-serif font.

THANK  
YOU