1. show the schema of the tables

```
4   -- 1 show the schema of the tables
5   desc scores;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
id	int	YES		NULL	
first_name	text	YES		NULL	
last_name	text	YES		NULL	
email	text	YES		NULL	
gender	text	YES		NULL	
part_time_job	text	YES		NULL	

Result 3

2. add primary key in table scores at id column

```
6   -- 2 add primary key in table scores at id column
7   alter table scores add primary key(Id);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
first_name	text	YES		NULL	
last_name	text	YES		NULL	
email	text	YES		NULL	
gender	text	YES		NULL	
part_time_job	text	YES		NULL	

Result 4

Output

3. create second table named as perform with foreign key

```
39  create table student_ids(student_id int,reading_score int,
40  writing_score int,placement_score int,club_join_date varchar(10),
41  foreign key (student_id) references marks(s_id));
42  desc student_ids;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
student_id	int	YES	MUL	NULL	
reading_score	int	YES		NULL	
writing_score	int	YES		NULL	
placement_score	int	YES		NULL	
club_join_date	varchar(10)	YES		NULL	

4.show whole record of table scores

```
11    -- 4 show whole record of table scores
12 •  select * from scores;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

id	first_name	last_name	email	gender	part_time_job	absence_days	extracurricular_activities	wee
1	Paul	Casey	paul.casey.1@gslingacademy.com	male	False	3	False	27
2	Danielle	Sandoval	danielle.sandoval.2@gslingacademy.com	female	False	2	False	47
3	Tina	Andrews	tina.andrews.3@gslingacademy.com	female	False	9	True	13
4	Tara	Clark	tara.clark.4@gslingacademy.com	female	False	5	False	3
5	Anthony	Campos	anthony.campos.5@gslingacademy.com	male	False	5	False	10
6	Kelly	Wade	kelly.wade.6@gslingacademy.com	female	False	2	False	26
7	Anthony	Smith	anthony.smith.7@gslingacademy.com	male	False	3	True	23
8	George	Short	george.short.8@gslingacademy.com	male	True	2	True	34

5. write query to change table name scores to marks

```
13    -- 5 rename table scores to marks
14 •   alter table scores rename marks;
15 •   show tables;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Tables_in_studenttt
marks
perform
student-scores

6. Modify the column name id to s_id

```
17 •  alter table marks rename column id to s_id;
18 •  desc marks;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
s_id	int	NO	PRI	NULL	
first_name	text	YES		NULL	
last_name	text	YES		NULL	
email	text	YES		NULL	
gender	text	YES		NULL	
part_time_job	text	YES		NULL	
absence_days	int	YES		NULL	
extracurricular_activities	text	YES		NULL	

Result 11

7. Write a query to fetch gender using distinct.

```
19 •  select distinct gender from marks;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

gender
male
female

8. Find students whose last name starts with 'S'

```
21 •  Select * from marks where last_name LIKE 'S%';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

s_id	first_name	last_name	email	gender	part_time_job	absence_days	extracurricular_ac
2	Danielle	Sandoval	danielle.sandoval.2@gslingacademy.com	female	False	2	False
7	Anthony	Smith	anthony.smith.7@gslingacademy.com	male	False	3	True
8	George	Short	george.short.8@gslingacademy.com	male	True	2	True
10	Audrey	Simpson	audrey.simpson.10@gslingacademy.com	female	False	3	True
23	Shannon	Simpson	shannon.simpson.23@gslingacademy.com	female	False	9	False
48	Billy	Soto	billy.soto.48@gslingacademy.com	male	False	0	True
51	Christopher	Stewart	christopher.stewart.51@gslingacademy.com	male	False	6	True

9. Display students sorted by their math_score in descending order

```
21 • Select * from marks order by math_score desc;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

eekly_self_study_hours	career_aspiration	math_score	history_score	physics_score	chemistry_score	biology_score	english_score	geogra
	Lawyer	100	82	85	68	97	81	100
	Unknown	100	62	84	66	88	90	71
	Doctor	100	74	80	84	85	82	74
	Doctor	100	97	94	87	89	91	100
	Software Engineer	99	96	97	73	88	76	64
	Software Engineer	99	65	98	75	66	72	100
	Banker	99	84	84	78	78	77	68

marks 15 ×

10. Fetch the first_name, last_name, and history score of student greater than 80

```
22 • select first_name,last_name,history_score From marks where history_score>80;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

first_name	last_name	history_score
Paul	Casey	81
Danielle	Sandoval	86
Tina	Andrews	97
Kelly	Wade	100
Anthony	Smith	96
George	Short	95
Pamela	Jackson	94
Laura	Jackson	90

marks 16 ×

11. Calculate the total number of students

```
23 • Select count(*) AS total_student from marks;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_student
2000

12. Find the maximum minmun score in English

```
24 •  Select max(english_score) AS student_perform From marks;
25
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_perform
99

13. Find the maximum minmun score in English

```
25 •  Select min(english_score) AS student_perform From marks;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_perform
50

14. Write query to fetch data where physics score is between 70 to 80

```
26 •  select s_id,first_name,last_name,physics_score from marks where physics_score between 70 and 80;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

s_id	first_name	last_name	physics_score
16	Roger	Wiley	78
21	Tim	Nichols	72
22	Kyle	Willis	78
27	Jason	Williams	80
36	Kelly	Farley	78
45	David	Gillespie	79
48	Billy	Soto	74
52	Sandra	Thornton	73

marks 21 ×

15. Write query to fetch data where physics score is not between 70 to 80

```
26 • select s_id,first_name,last_name,physics_score from marks where physics_score not between 70 and 80;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows:

s_id	first_name	last_name	physics_score
1	Paul	Casey	93
2	Danielle	Sandoval	96
3	Tina	Andrews	95
4	Tara	Clark	88
5	Anthony	Campos	65
6	Kelly	Wade	67
7	Anthony	Smith	97
8	George	Short	82

marks 22

16. Write query to fetch data where chemistry score is 70, 80 or 90.

```
26 • select s_id,first_name,last_name,chemistry_score from marks where chemistry_score IN(70,80,90);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

s_id	first_name	last_name	chemistry_score
4	Tara	Clark	80
35	Lisa	Burns	80
41	Emily	Holloway	90
48	Billy	Soto	70
49	Kimberly	Kelly	80
52	Sandra	Thornton	70
65	Eric	Reyes	70
77	Jennifer	Garcia	90

marks 23

17. Write Query to fetch data where student name starts from c.

```
27 • select s_id,first_name,last_name from marks where first_name Like "c%";
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

s_id	first_name	last_name
12	Clinton	Randolph
25	Cassandra	West
30	Carol	Hill
32	Cynthia	Knapp
40	Christopher	Taylor
51	Christopher	Stewart
60	Casey	May
79	Chelsea	Craig

18. Write Query to gives first_name in upper case

```
28 •  select upper(first_name) as uppercase,first_name from marks limit 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

uppercase	first_name
PAUL	Paul
DANIELLE	Danielle
TINA	Tina
TARA	Tara
ANTHONY	Anthony

19. Using date function

Now()

```
29 •  select now() as cur_date;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

cur_date
2024-06-17 23:22:08

Curdate()

```
30 •  select curdate();
```

Result Grid | Filter Rows:

curdate()
2024-06-17

20. Write query to fetch number of students in table

```
31 •  select count(s_id) as number_of_students from marks;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

number_of_students
2000

21. Write Query to fetch students first name whose number of math score are more than 2.

```
32 •  select first_name,count(math_score) from marks where math_score<80 group by first_name having count(*)>2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	count(math_score)
Kyle	3
Sean	3
Jeffrey	3
Cynthia	5
Timothy	4
David	9
Elizabeth	4
Christopher	13

Result 32

22. Write Query to find highest geography_score of students using subquery

```
33 •  select s_id,first_name,geography_score from marks where geography_score=(select max(geography_score) from marks);
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

s_id	first_name	geography_score
20	Angela	100
49	Kimberly	100
89	Robert	100
115	Robert	100
142	Marc	100
143	Cathy	100
202	Jordan	100
256	Andrea	100

marks 34 | Apply

23. Using Concat string function for students full name

```
34 •  select concat(first_name,last_name) as fullname from marks;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

fullname
PaulCasey
DanielleSandoval
TinaAndrews
TaraClark
AnthonyCampos
KellyWade

24. Joins

Inner Join

```
49 •  select * from student_ids
50    inner join marks on student_ids.student_id = marks.s_id
51
52
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	reading_score	writing_score	placement_score	club_join_date	s_id	first_name	last_name	email
1	86	67	78	2021	1	Paul	Casey	paul.casey.1@gslingacademy.co
2	85	71	80	2019	2	Danielle	Sandoval	danielle.sandoval.2@gslingacade
3	77	77	84	2021	3	Tina	Andrews	tina.andrews.3@gslingacademy.
4	76	75	75	2021	4	Tara	Clark	tara.clark.4@gslingacademy.com
5	91	62	90	2019	5	Anthony	Campos	anthony.campos.5@gslingacader

Result 10 ×

Left Outer Join

```
52    select * from student_ids
53    left outer join marks on student_ids.student_id = marks.s_id;
54
55
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	reading_score	writing_score	placement_score	club_join_date	s_id	first_name	last_name	email
1	86	67	78	2021	1	Paul	Casey	paul.casey.1@gslingacademy.co
2	85	71	80	2019	2	Danielle	Sandoval	danielle.sandoval.2@gslingacade
3	77	77	84	2021	3	Tina	Andrews	tina.andrews.3@gslingacademy.
4	76	75	75	2021	4	Tara	Clark	tara.clark.4@gslingacademy.com
5	91	62	90	2019	5	Anthony	Campos	anthony.campos.5@gslingacader

Right Outer Join

```
55 •  select * from student_ids
56    right outer join marks on student_ids.student_id = marks.s_id;
57
58
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

student_id	reading_score	writing_score	placement_score	club_join_date	s_id	first_name	last_name	email
1	86	67	78	2021	1	Paul	Casey	paul.casey.1@gslingacademy.co
1	86	67	78	2021	1	Paul	Casey	paul.casey.1@gslingacademy.co
2	85	71	80	2019	2	Danielle	Sandoval	danielle.sandoval.2@gslingacade
2	85	71	80	2019	2	Danielle	Sandoval	danielle.sandoval.2@gslingacade
3	77	77	84	2021	3	Tina	Andrews	tina.andrews.3@gslingacademy.

Result 12 ×

Cross Join

```
55 •  select * from student_ids
56    Cross join marks on student_ids.student_id = marks.s_id;
57
58
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

student_id	reading_score	writing_score	placement_score	club_join_date	s_id	first_name	last_name	email
1	86	67	78	2021	1	Paul	Casey	paul.casey.1@gslingacademy.co
2	85	71	80	2019	2	Danielle	Sandoval	danielle.sandoval.2@gslingacade
3	77	77	84	2021	3	Tina	Andrews	tina.andrews.3@gslingacademy.
4	76	75	75	2021	4	Tara	Clark	tara.clark.4@gslingacademy.com
5	91	62	90	2019	5	Anthony	Campos	anthony.campos.5@gslingacader

Result 13 ×

Output

25. Create View

```
58 •  create view detail_view
59    as select * from marks where math_score>90;
60 •  select * from detail_view;
61
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

s_id	first_name	last_name	email	gender	part_time_job	absence_days	extracurricular_activities
1	Paul	Casey	paul.casey.1@gslingacademy.com	male	False	3	False
2	Danielle	Sandoval	danielle.sandoval.2@gslingacademy.com	female	False	2	False
3	Tina	Andrews	tina.andrews.3@gslingacademy.com	female	False	9	True
4	Tara	Clark	tara.clark.4@gslingacademy.com	female	False	5	False
5	Anthony	Campos	anthony.campos.5@gslingacademy.com	male	False	5	False

detail_view 3 ×