

Q1. Write a query to display all users stored in the system.

```
8 • select * from users;
```

user_id	name	email	password
1	HARRY	HRY@GCOM	1234
2	ROMA	ROMA@GCOM	1234
3	JORGE	JOR@GCOM	1234
4	SYNO	SYNO@GCOM	1234
5	CRISTOPHER	CRIS@GCOM	1234
6	ALLEN	ALN@GCOM	1234
7	ROUNE	RON@GCOM	1234
8	CLORE	CLORE@GCOM	1234
9	CRIMSON	CRIM@GCOM	1234
10	POTTER	POTR@GCOM	1234
NULL	NULL	NULL	NULL

Q2. Write a query to list all movies in the movie table.

```
8 • select * from movies;
```

movie_id	title	genre	duration
1	The Silent River	Drama	125
2	Galactic Wars	Sci-Fi	140
3	Love in Paris	Romance	115
4	Mystery Mansion	Thriller	130
5	Laugh Out Loud	Comedy	100
6	The Last Kingdom	Historical	150
7	Fast Wheels	Action	110
8	Ocean Secrets	Adventure	135
9	Haunted Nights	Horror	95
10	Future Dreams	Fantasy	145
NULL	NULL	NULL	NULL

Q3. Write a query to display all available shows.

```
8 • select * from shows;
```

show_id	movie_id	show_date	show_time	price
1	1	2026-02-16	18:30:00	250.00
2	2	2026-02-16	21:00:00	300.00
3	3	2026-02-17	15:00:00	200.00
4	4	2026-02-17	19:00:00	350.00
5	5	2026-02-18	20:30:00	280.00
6	6	2026-02-18	17:00:00	220.00
7	7	2026-02-19	14:00:00	180.00
8	8	2026-02-19	19:30:00	320.00
9	9	2026-02-20	16:00:00	240.00
10	10	2026-02-20	21:15:00	400.00
NULL	NULL	NULL	NULL	NULL

Q4. Write a query to fetch details of a movie using its movie_id.

```
8 • select * from movies where movie_id in (3);
```

movie_id	title	genre	duration
3	Love in Paris	Romance	115
NULL	NULL	NULL	NULL

Q5. Write a query to list all shows of a particular movie.

```
9 • select s.show_time,m.movie_id,m.title from movies m
10 inner join shows s on m.movie_id=s.movie_id where m.movie_id=1;
```

show_time	movie_id	title
18:30:00	1	The Silent River
19:30:00	1	The Silent River
21:00:00	2	Galactic Wars
15:00:00	3	Love in Paris
19:00:00	4	Mystery Mansion
20:30:00	5	Laugh Out Loud
17:00:00	6	The Last Kingdom
14:00:00	7	Fast Wheels
19:30:00	8	Ocean Secrets
16:00:00	9	Haunted Nights
21:15:00	10	Future Dreams

Q6. Write a query to list all seat numbers from the seats table.

```
12 • select seat_number from seats;
```

seat_number
A1
A2
A3
A4
A5
B1
B2
B3
B4
B5

Q7. Write a query to find available seats for a specific show.

```
16 • select s.seat_number, s.seat_id from seats s where s.seat_id
17 not in (select seat_id from bookings where show_id =7);
```

seat_number	seat_id
A1	1
A2	2
A3	3
A4	4
A5	5
B1	6
B3	8
B4	9
B5	10
NULL	NULL

Q8. Write a query to display all booked seats for a specific show.

```
19 • select b.show_id,b.seat_id,s.seat_number from bookings b
20 inner join seats s on b.seat_id=s.seat_id where b.show_id=5;
```

Result Grid

show_id	seat_id	seat_number
5	5	A5
5	4	A4

Q9. Write a query to insert a new user into the users table.

```
22 • Insert into users value(12, 'KEREN', 'KRN@GCOM', '1234');
```

Result Grid

user_id	name	email	password
1	HARRY	HRY@GCOM	1234
2	ROMA	ROMA@GCOM	1234
3	JORGE	JOR@GCOM	1234
4	SYNO	SYNO@GCOM	1234
5	CRISTOPHER	CRIS@GCOM	1234
6	ALLEN	ALN@GCOM	1234
7	ROUNE	RON@GCOM	1234
8	CLORE	CLORE@GCOM	1234
9	CRIMSON	CRIM@GCOM	1234
10	POTTER	POTR@GCOM	1234
12	KEREN	KRN@GCOM	1234
*	NULL	NULL	NULL

Q10. Write a query to insert a new movie into the movies table.

```
23 • INSERT INTO MOVIES VALUE(11,'Harry Potter', 'Fantasy',140);
```

Result Grid

movie_id	title	genre	duration
1	The Silent River	Drama	125
2	Galactic Wars	Sci-Fi	140
3	Love in Paris	Romance	115
4	Mystery Mansion	Thriller	130
5	Laugh Out Loud	Comedy	100
6	The Last Kingdom	Historical	150
7	Fast Wheels	Action	110
8	Ocean Secrets	Adventure	135
9	Haunted Nights	Horror	95
10	Future Dreams	Fantasy	145
11	Harry Potter	Fantasy	140
*	NULL	NULL	NULL

Q11. Write a query to insert a new show into the shows table.

```
24 • INSERT INTO SHOWS VALUE (12, 7, '2026-02-16', '14:00:00',450.00 );
```

Result Grid

show_id	movie_id	show_date	show_time	price
1	1	2026-02-16	18:30:00	250.00
2	2	2026-02-16	21:00:00	300.00
3	3	2026-02-17	15:00:00	200.00
4	4	2026-02-17	19:00:00	350.00
5	5	2026-02-18	20:30:00	280.00
6	6	2026-02-18	17:00:00	220.00
7	7	2026-02-19	14:00:00	180.00
8	8	2026-02-19	19:30:00	320.00
9	9	2026-02-20	16:00:00	240.00
10	10	2026-02-20	21:15:00	400.00
11	1	2026-02-16	19:30:00	300.00
12	7	2026-02-16	14:00:00	450.00
*	NULL	NULL	NULL	NULL

Q12. Write a query to add a new seat to the seats table.

25 • `INSERT INTO SEATS VALUE(11, 'C1');`

Result Grid | Filter Rows: | Edit:

seat_id	seat_number
1	A1
2	A2
3	A3
4	A4
5	A5
6	B1
7	B2
8	B3
9	B4
10	B5
11	C1
HALL	HALL

Q13. Write a query to book a ticket for a user.

`INSERT INTO BOOKINGS VALUE (12,3,8,2,'2026-02-17 14:20:00');`

Q14. Write a query to display all bookings

```
SELECT b.booking_id,  
       b.show_id,  
       b.seat_id,  
       m.movie_id,  
       m.title,  
       s.seat_number,  
       sh.price,  
       sh.show_time  
FROM bookings b  
JOIN shows sh  
  ON sh.show_id = b.show_id  
JOIN movies m  
  ON sh.movie_id = m.movie_id  
JOIN seats s  
  ON b.seat_id = s.seat_id;
```

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

booking_id	show_id	seat_id	movie_id	price	show_time	seat_number	title
1	1	1	1	250.00	18:30:00	A1	The Silent River
2	2	2	2	300.00	21:00:00	A2	Galactic Wars
3	3	3	3	200.00	15:00:00	A3	Love in Paris
4	4	4	4	350.00	19:00:00	A4	Mystery Mansion
5	5	5	5	280.00	20:30:00	A5	Laugh Out Loud
11	5	4	5	280.00	20:30:00	A4	Laugh Out Loud
6	6	6	6	220.00	17:00:00	B1	The Last Kingdom
7	7	7	7	180.00	14:00:00	B2	Fast Wheels
8	8	8	8	320.00	19:30:00	B3	Ocean Secrets
12	8	2	8	320.00	19:30:00	A2	Ocean Secrets
9	9	9	9	240.00	16:00:00	B4	Haunted Nights
10	10	10	10	400.00	21:15:00	B5	Future Dreams

Q15. Write a query to display booking history for a specific user.

SELECT

b.user_id,b.booking_id,b.show_id,b.seat_id,m.movie_id,sh.price,sh.show_time,s.seat_number,m.title
from bookings b

join shows sh on sh.show_id=b.show_id join movies m on sh.movie_id=m.movie_id join seats s on
b.seat_id = s.seat_id where b.user_id=3;

```
27 • SELECT b.user_id,b.booking_id,b.show_id,b.seat_id,m.movie_id,sh.price,sh.show_time,s.seat_number,m.title
28      join shows sh on sh.show_id=b.show_id join movies m on sh.movie_id=m.movie_id
```

	user_id	booking_id	show_id	seat_id	movie_id	price	show_time	seat_number	title
▶	3	3	3	3	3	200.00	15:00:00	A3	Love in Paris
	3	12	8	2	8	320.00	19:30:00	A2	Ocean Secrets

Q16. Write a query to count total bookings for a specific show.

```
30 • select count(booking_id),show_id from bookings
31      group by show_id having show_id in (5,8);
```

	count(booking_id)	show_id
▶	2	5
	2	8

Q17. Write a query to count total bookings for a specific movie.

```
35 select m.title,m.movie_id, count(b.booking_id) from bookings b join
36 shows sh on sh.show_id=b.show_id join movies m on m.movie_id=sh.movie_id
37 where m.movie_id=3 group by m.title,m.movie_id;
```

	title	movie_id	count(b.booking_id)
▶	Love in Paris	3	1

Q18. Write a query to find all movies of a specific genre.

select genre,movie_id,title from movies where genre='fantasy';

Q19. Write a query to update the duration of a movie.

update movies set duration=120 where movie_id=5;

Q20. Write a query to update the seat number for a given seat_id.

update seats set seat_number='B3' where seat_id=4;

Q21. Write a query to delete a specific booking record.

delete from bookings where booking_id=3;

Q22. Write a query to delete a movie from the movies table.

Q23. Write a query to list movies longer than 2 hours.

select title, duration from movies where duration>=120;

Q24. Write a query to find shows scheduled for today's date.

*select * from shows where show_date=current_date();*

Q25. Write a query to list upcoming shows sorted by date and time.

*select * from shows where show_id>=current_date() order by show_date asc , show_time asc;*