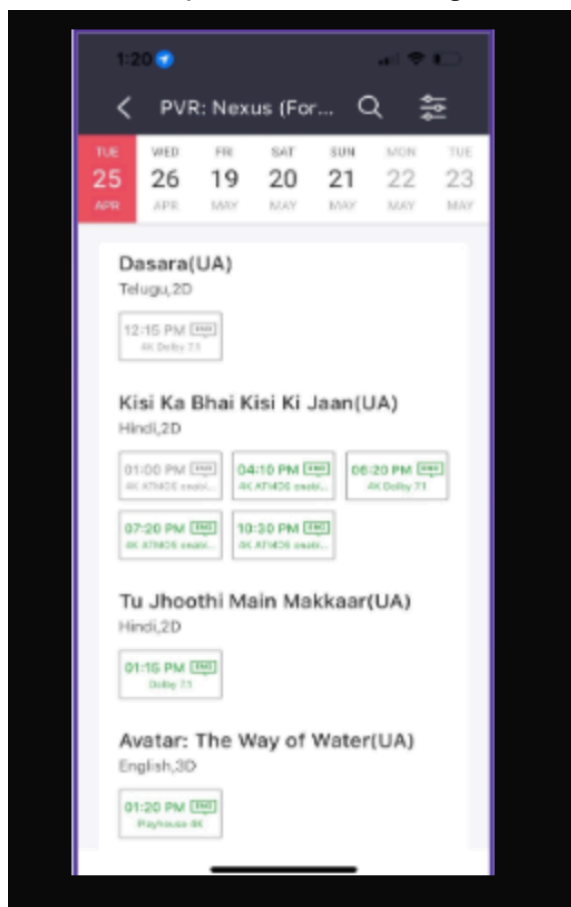


Problem Statement:

Bookmyshow is a ticketing platform where you can book tickets for a movie show. The image attached represents that for a given theatre we can see the next 7 dates. As one chooses the date, we get list of all shows running in that theatre along with the show timings.

P1 - As part of this assignment, we need to list down all the entities, their attributes and the table structures for the scenario mentioned in the previous slide. You also need to write the SQL queries required to create these tables along with few sample entries. Ensure the tables follow 1NF, 2NF, 3NF and BCNF rules.

P2 - Write a query to list down all the shows on a given date at a given theatre along with their respective show timings.



Submission guidelines

Expected output of this assignment is a doc with a list of all tables, their attributes, a few example rows, SQL for P1 and P2 solutions as a PR on Github. Ensure the SQL queries are directly executable on MySQL database.

Solution P1:

P1: As part of this assignment, we need to list down all the entities, their attributes and the table structures for the scenario mentioned in the previous slide. You also need to write the SQL queries required to create these tables along with few sample entries. Ensure the tables follow 1NF, 2NF, 3NF and BCNF rules.

TABLES/ENTITIES

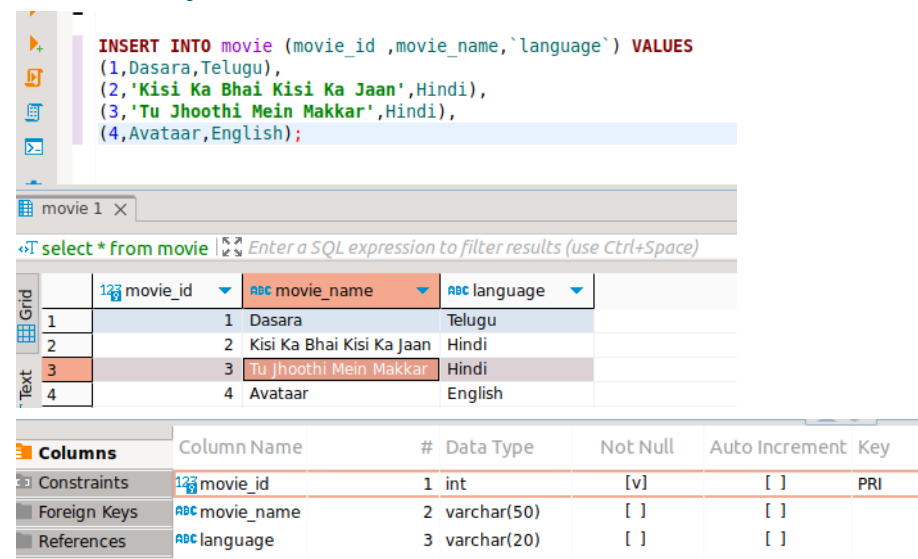
1. movie
2. theatre
3. showtimes

ATTRIBUTES

movie:

- movie_name
- language

```
CREATE TABLE movie (  
  movie_id int NOT NULL,  
  movie_name varchar(50) DEFAULT NULL,  
  language varchar(20) DEFAULT NULL,  
  PRIMARY KEY (movie_id)  
)  
  
INSERT INTO movie (movie_id ,movie_name,`language`) VALUES  
(1,Dasara,Telugu),  
(2,'Kisi Ka Bhai Kisi Ka Jaan',Hindi),  
(3,'Tu Jhoothi Mein Makkar',Hindi),  
(4,Avataar,English);
```



The screenshot shows a database management tool interface. At the top, there's a SQL editor with the following code:

```
INSERT INTO movie (movie_id ,movie_name,`language`) VALUES  
(1,Dasara,Telugu),  
(2,'Kisi Ka Bhai Kisi Ka Jaan',Hindi),  
(3,'Tu Jhoothi Mein Makkar',Hindi),  
(4,Avataar,English);
```

Below the editor, there's a tab labeled 'movie 1'. Underneath it, there's a SQL query: `select * from movie`. Below the query, there's a table with 4 rows and 3 columns: `movie_id`, `movie_name`, and `language`.

	movie_id	movie_name	language
1	1	Dasara	Telugu
2	2	Kisi Ka Bhai Kisi Ka Jaan	Hindi
3	3	Tu Jhoothi Mein Makkar	Hindi
4	4	Avataar	English

At the bottom, there's a 'Columns' table showing the structure of the 'movie' table:

	Column Name	#	Data Type	Not Null	Auto Increment	Key
Constraints	movie_id	1	int	[v]	[]	PRI
Foreign Keys	movie_name	2	varchar(50)	[]	[]	
References	language	3	varchar(20)	[]	[]	

theatre:

- Theatre_id
- Auditorium

```
CREATE TABLE theatre (  
  theatre_id int NOT NULL,
```

```
Auditorium varchar(50),  
PRIMARY KEY (theatre_id)
```

```
)  
INSERT INTO theatre (theatre_id,Auditorium) VALUES (1,Venus), (2,Mars);
```

The screenshot shows a database client interface with tabs for *localhost 2> Movie_table, *book_my_show, *localhost 2> showtimes, and *localhost. The SQL editor contains the following queries:

```
select * from theatre;  
INSERT INTO theatre (theatre_id,Auditorium) VALUES (1,Venus), (2,Mars);
```

The results pane shows the output of the first query:

theatre_id	Auditorium
1	Venus
2	Mars

The screenshot shows the 'Columns' tab of a database client. The table structure is as follows:

Column Name	#	Data Type	Not Null	Auto Increment	Key
theatre_id	1	int	[v]	[]	PRI
Auditorium	2	varchar(50)	[]	[]	

The screenshot shows the 'References' tab of a database client. The foreign key relationship is as follows:

Name	Column	Owner	Ref Table	Type	Ref Object	On Delete	On Update
showtimes_ibfk_2	—	showtimes	theatre	FOREIGN KEY	PRIMARY	Restrict	Restrict

Showtimes:

- id
- movie_id
- theatre_id
- date
- time
- available_seats

Query used:

```
CREATE TABLE showtimes(  
id INT NOT NULL AUTO_INCREMENT,  
movie_id INT NOT NULL,  
theatre_id INT NOT NULL,  
date DATE NOT NULL,  
time TIME NOT NULL,  
available_seats INT NOT NULL,  
PRIMARY KEY (id),  
FOREIGN KEY (movie_id) REFERENCES movie(movie_id),  
FOREIGN KEY (theatre_id) REFERENCES theatre (theatre_id)  
)
```

```
INSERT INTO showtimes (movie_id, theatre_id, `date`, `time`, available_seats)  
VALUES  
(1,1,'2024-01-26', '10:00:00', 112)  
(2,1,'2024-01-27', '10:00:00', 112)  
(3,1,'2024-01-28', '10:00:00', 112),  
(4,1,'2024-01-29', '12:00:00', 115),  
(1,2,'2024-01-30', '14:00:00', 135),  
(2,2,'2024-02-01', '16:00:00', 134),  
(3,2,'2024-02-02', '18:00:00', 100),  
(4,2,'2024-02-03', '20:00:00', 180);
```

```
select * from showtimes;

INSERT INTO showtimes (movie_id, theatre_id, `date`, `time`, available_seats)
VALUES
(1,1,'2024-01-26', '10:00:00', 112),
(2,1,'2024-01-27', '10:00:00', 112),
(3,1,'2024-01-28', '10:00:00', 112),
(4,1,'2024-01-29', '12:00:00', 115),
(1,2,'2024-01-30', '14:00:00', 135),
(2,2,'2024-02-01', '16:00:00', 134),
(3,2,'2024-02-02', '18:00:00', 100),
(4,2,'2024-02-03', '20:00:00', 180);
```

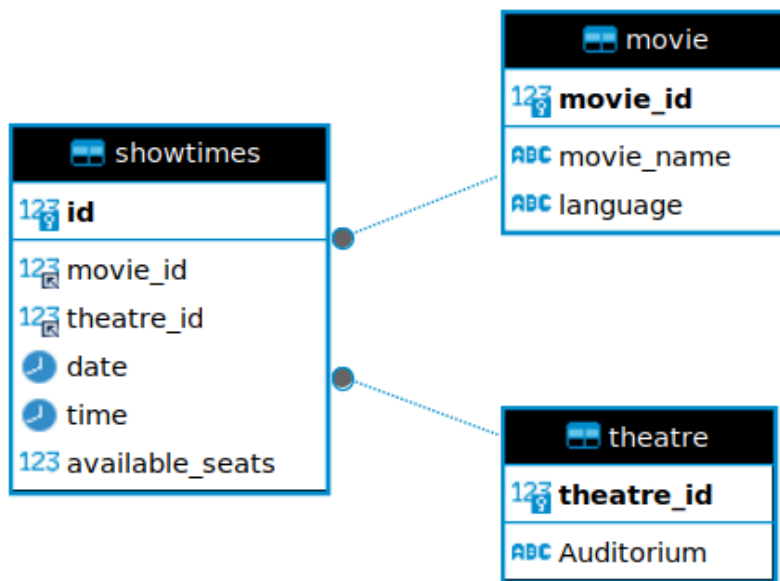
showtimes 1 X

select * from showtimes Enter a SQL expression to filter results (use Ctrl+Space)

id	movie_id	theatre_id	date	time	available_seats
1	1	1	2024-01-26	10:00:00	100
2	2	1	2024-01-27	10:00:00	100
9	3	1	2024-01-28	10:00:00	112
10	4	1	2024-01-29	12:00:00	115
11	1	2	2024-01-30	14:00:00	135
12	2	2	2024-02-01	16:00:00	134
13	3	2	2024-02-02	18:00:00	100
14	4	2	2024-02-03	20:00:00	180

Columns	Column Name	#	Data Type	Not Null	Auto Increment	Key	Default	Extra
Constraints	id	1	int	[v]	[v]	PRI		auto_increment
Foreign Keys	movie_id	2	int	[v]	[]	MUL		
References	theatre_id	3	int	[v]	[]	MUL		
Triggers	date	4	date	[v]	[]			
Indexes	time	5	time	[v]	[]			
Partitions	available_seats	6	int	[v]	[]			

Columns	Name	Column	Owner	Ref Table	Type	Ref Object	On Delete	On Update
Constraints	showtimes_ibfk_1	—	showtimes	movie	FOREIGN KEY	PRIMARY	Restrict	Restrict
Foreign Keys	showtimes_ibfk_2	—	showtimes	theatre	FOREIGN KEY	PRIMARY	Restrict	Restrict



Solution P2:

```

SELECT m.movie_name, s.time, s.available_seats
FROM showtimes s
INNER JOIN movie m ON s.movie_id = m.movie_id
WHERE s.date='2024-01-29' AND s.theatre_id = 1;
  
```

SQL query editor showing the same query as above.

Results window showing the output of the query:

	ABC movie_name	time	123 available_seats
1	Avataar	12:00:00	115