

Database Design: Movie Theatre

*A guide for a database design for a movie theatre &
movie booking website*

Ben Brumm

www.databasesstar.com

Database Design: Movie Theatre

This guide is a companion to my YouTube video on designing a movie theatre & movie booking website. The video used the website hoyts.com.au as an example, but the design should work for other movie theatre websites.

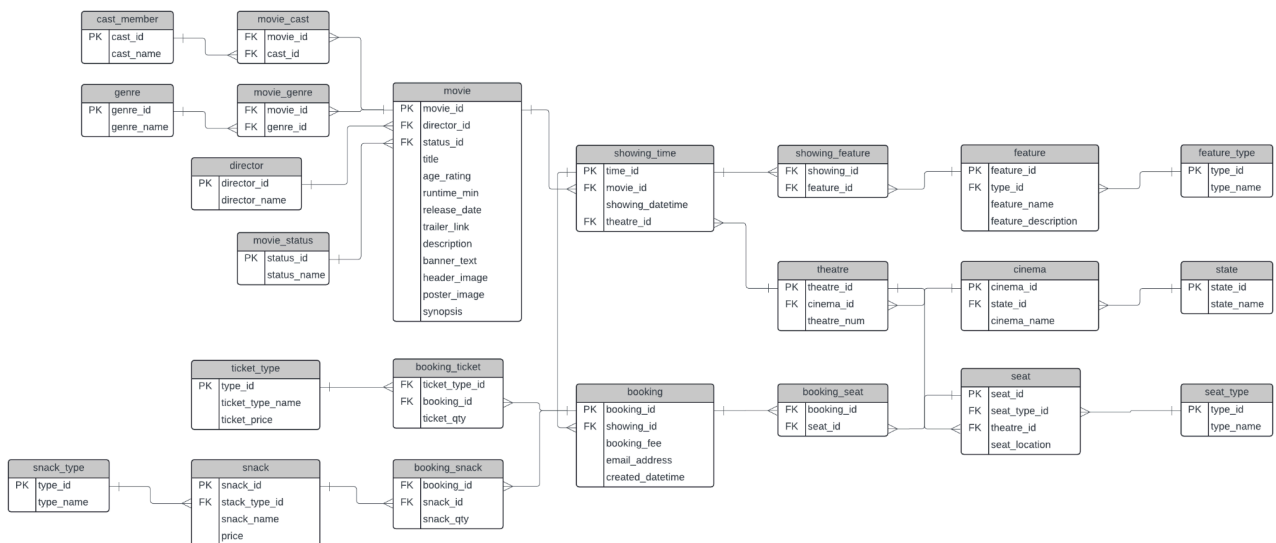
In this guide, you'll see:

- An Entity Relationship Diagram for the movie booking website, from my YouTube video.
- An explanation of the purpose of each table and field, with sample data.
- SQL scripts to create each of these tables with some sample data.

Let's get into it.

Entity Relationship Diagram

Here's the ERD for this database:



A PNG file of this ERD is available here:

https://dbshostedfiles.s3.us-west-2.amazonaws.com/dbs/erd_movie_booking.png

SQL Scripts

The SQL Script to create this database, and populate some sample data, is available on GitHub here:

https://github.com/bbrumm/databasestar/tree/main/videos/139_movie

The script has been written for MySQL, but with some minor tweaks (such as data types), it should work for all other databases.

Database Definition

This section explains each of these tables and fields.

movie

A record that represents a movie that is being shown at a cinema.

Column	Description
movie_id	The primary key for the row.
director_id	A foreign key for the director row, which is the director of this movie.
status_id	A foreign key to the movie_status row, to indicate if a movie is now showing or coming soon.
title	The title of the movie, such as "Inside Out 2"
age_rating	The age rating of the movie, such as PG or M.
runtime_min	The number of minutes that the movie runs for.
release_date	The date the movie was released to cinemas.
trailer_link	The URL to the trailer video.
description	The description of the movie or what it is about.
banner_text	A paragraph that encourages people to see the movie.
header_image	A URL for the image to show for the movie's header or cover.
poster_image	A URL for the image for the movie's poster
synopsis	A longer text description of the movie.

cast_member

A person who has acted in a movie.

Column	Description
cast_id	The primary key for the row.
cast_name	The name of the actor.

movie_cast

A joining table to record the cast members in movies.

Column	Description
movie_id	A foreign key to the movie table.
cast_id	A foreign key to the cast_member table.

genre

A list of the genres for movies (the category or theme of the movie).

Column	Description
genre_id	The primary key for the row.
genre_name	The name of the genre, such as "action" or "comedy".

movie_genre

A joining table between movies and genres.

Column	Description
movie_id	The foreign key for the movie.
genre_id	The foreign key for the genre.

director

The person that can direct a movie.

Column	Description
director_id	The primary key for the row.
director_name	The name of the director.

movie_status

A list of possible statuses for the movie, so we can differentiate between movies that are coming soon, now showing, or have stopped showing.

Column	Description
status_id	The primary key for the row.
status_name	The name of the status, such as "Coming Soon" or "Now Showing".

showing_time

A list of dates and times that a movie is being shown in a cinema.

Column	Description
time_id	The primary key for the row.
movie_id	A foreign key to the movie.
theatre_id	A foreign key to the theatre.
showing_datetime	The date and time that the movie is being shown in a theatre.

showing_feature

A joining table for the showing_time and feature tables.

Column	Description
showing_id	The foreign key to the showing_time table.
feature_id	The foreign key to the feature table.

feature

A list of the features that can be applied to a movie showing.

Column	Description
feature_id	The primary key for the row.
type_id	The foreign key to the feature type.
feature_name	The name of the feature, such as "Dolby Sound" or "Daybeds".
feature_description	A description of the feature to provide the user with more information.

feature_type

A list of types for features.

Column	Description
type_id	The primary key for the row.
type_name	The name of this feature type, such as "Experience" or "Accessibility".

theatre

A record that represents a physical room within a cinema complex that has a number of seats and a screen for showing a movie.

Column	Description
theatre_id	The primary key for the row.
cinema_id	A foreign key to the cinema.
theatre_num	An identifier for the theatre, which is shown on the outside of the room for customers to identify it.

cinema

A complex that contains one or more theatre rooms.

Column	Description
cinema_id	The primary key for the row.
state_id	A foreign key to the state table, for the state that this cinema is in.
cinema_name	The name of the cinema, such as "Chadstone" or "Balmoral"

state

The state within the country that a cinema can belong to.

Column	Description
state_id	The primary key for the row.
state_name	The name of the state, such as "VIC" or "NSW" (in Australia).

booking

A record for the customer's booking of one or more tickets for a movie showing.

Column	Description
booking_id	The primary key for the row.
showing_id	A foreign key to the showing_time.
booking_fee	The fee charged by the company to the customer for the booking.
email_address	The email address for the customer.
created_datetime	The date and time the booking was created.

booking_seat

A joining table between booking and seat, to record the seats for the booking.

Column	Description
booking_id	A foreign key to the booking table,
seat_id	A foreign key to the seat table.

seat

A record for the individual seat within a theatre.

Column	Description
seat_id	The primary key for the row.
seat_type_id	A foreign key for the seat_type.
theatre_id	A foreign key for the theatre.
seat_location	The location of the seat in the cinema, such as "J2" or "B15".

seat_type

A list of the types of seats.

Column	Description
type_id	The primary key for the row.
type_name	The name of the seat type, such as "Daybed" or "Standard".

ticket_type

The type of ticket that can be offered.

Column	Description
type_id	The primary key for the row.
ticket_type_name	The name of the type of ticket, such as "Adult" or "Student".
ticket_price	The price of this type of ticket.

booking_ticket

A joining table between booking and ticket_type.

Column	Description
ticket_type_id	A foreign key for the ticket type.
booking_id	A foreign key for the booking.
ticket_qty	The number of tickets of this type for the booking.

snack_type

A list of the types of snacks.

Column	Description
type_id	The primary key for the row.
type_name	The name of the snack type, such as "Drinks" or "Hot Food".

snack

A snack that can be purchased by the customer when booking.

Column	Description
snack_id	The primary key for the row.
snack_type_id	A foreign key for the snack type.
snack_name	The name of the snack, such as "Large Chips" or "Coffee".
price	The price of the snack.

booking_snack

A joining table for the snack and booking.

Column	Description
booking_id	A foreign key for the booking.
snack_id	A foreign key for the snack.
snack_qty	The number of snacks ordered for this booking.

Conclusion

Hopefully you found this guide, diagram, and script useful.

If you have any questions or suggestions, let me know at ben@databasestar.com.

Thanks!

Ben Brumm

www.DatabaseStar.com