

Carson Wardell Project 3

```
CREATE TABLE `Director` (  
  `directorId` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,  
  `name` TEXT NOT NULL  
)  
)  
CREATE TABLE `Movie` (  
  `movieId` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,  
  `title` TEXT NOT NULL,  
  `releaseDate` TEXT NOT NULL,  
  `boxOffice` INTEGER NOT NULL,  
  `directorId` INTEGER NOT NULL,  
  FOREIGN KEY(`directorId`) REFERENCES `Director`(`directorId`)  
  ON UPDATE NO ACTION ON DELETE CASCADE  
)  
CREATE TABLE `FunFact` (  
  `id` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,  
  `movieId` INTEGER NOT NULL,  
  `description` TEXT NOT NULL,  
  FOREIGN KEY(`movieId`) REFERENCES `Movie`(`movieId`)  
  ON UPDATE NO ACTION ON DELETE CASCADE  
)  
  
CREATE TABLE room_master_table (  
  id INTEGER PRIMARY KEY,  
  identity_hash TEXT  
)
```

Explanation:

The director table has director names

The movie table has title, release date, boxoffice and director id pointing to director in other table.

Fun fact is a table with fun facts and a movie id pointing to a movie.

I only use movie and director in my ui

Insert statements

```
INSERT INTO director (directorId, name) VALUES  
(1, 'John Lasseter'),  
(2, 'Roger Allers & Rob Minkoff'),  
(3, 'Ridley Scott'),  
(4, 'James Cameron'),  
(5, 'Peter Jackson'),  
(6, 'Christopher Nolan'),
```

(7, 'M. Night Shyamalan'),
(8, 'Gore Verbinski'),
(9, 'Lana Wachowski & Lilly Wachowski'),
(10, 'Robert Zemeckis'),
(11, 'Frank Darabont'),
(12, 'Steven Spielberg'),
(13, 'Quentin Tarantino'),
(14, 'David Fincher'),
(15, 'Francis Ford Coppola'),
(16, 'Irvin Kershner'),
(17, 'Joss Whedon'),
(18, 'George Lucas');

INSERT INTO movie (movieId, title, releaseDate, boxOffice, directorId) VALUES

(1, 'Toy Story', '1995-11-22', 373554033, 1),
(2, 'The Lion King', '1994-06-15', 968511805, 2),
(3, 'Gladiator', '2000-05-05', 460000000, 3),
(4, 'Titanic', '1997-12-19', 2187463944, 4),
(5, 'The Lord of the Rings: The Return of the King', '2003-12-17', 1111920000, 5),
(6, 'Inception', '2010-07-16', 829895144, 6),
(7, 'The Sixth Sense', '1999-08-06', 672806292, 7),
(8, 'Pirates of the Caribbean: The Curse of the Black Pearl', '2003-07-09', 654264015, 8),
(9, 'The Matrix', '1999-03-31', 466284000, 9),
(10, 'Forrest Gump', '1994-07-06', 678113774, 10),
(11, 'The Shawshank Redemption', '1994-09-23', 28341469, 11),
(12, 'Jurassic Park', '1993-06-11', 1070000000, 12),
(13, 'The Dark Knight', '2008-07-18', 1004558444, 6),
(14, 'Pulp Fiction', '1994-10-14', 214179088, 13),
(15, 'Fight Club', '1999-10-15', 100853753, 14),
(16, 'The Godfather', '1972-03-24', 246120974, 15),
(17, 'The Godfather Part II', '1974-12-20', 57300000, 15),
(18, 'The Empire Strikes Back', '1980-05-21', 538375067, 16),
(19, 'Back to the Future', '1985-07-03', 381109762, 10),
(20, 'The Avengers', '2012-05-04', 1518812988, 17),
(21, 'Indiana Jones and the Last Crusade', '1989-05-24', 474171806, 12),
(22, 'Star Wars', '1977-05-25', 775398007, 18),
(23, 'The Lord of the Rings: The Fellowship of the Ring', '2001-12-19', 871530324, 5),
(24, 'The Lord of the Rings: The Two Towers', '2002-12-18', 926287400, 5),
(25, 'A Bug's Life', '1998-11-25', 363258926, 1),
(26, 'Finding Nemo', '2003-05-30', 940335536, 1),
(27, 'The Lion King 2: Simba's Pride', '1998-10-27', 176000000, 2),
(28, 'The Road to El Dorado', '2000-03-31', 76000000, 2),
(29, 'Prometheus', '2012-06-08', 403370000, 3),
(30, 'Alien: Covenant', '2017-05-19', 240000000, 3),

Carson Wardell Project 3

(31, 'Avatar', '2009-12-18', 2787965087, 4),
(32, 'Aliens', '1986-07-18', 131060000, 4),
(33, 'The Hobbit: An Unexpected Journey', '2012-12-14', 1050000000, 5),
(34, 'King Kong', '2005-12-14', 550000000, 5),
(35, 'Dunkirk', '2017-07-21', 526900000, 6),
(36, 'The Prestige', '2006-10-20', 109676000, 6),
(37, 'Signs', '2002-08-02', 408247000, 7),
(38, 'Unbreakable', '2000-11-21', 248118000, 7),
(39, 'The Lone Ranger', '2013-07-03', 260500000, 8),
(40, 'Rango', '2011-03-04', 245000000, 8),
(41, 'Cloud Atlas', '2012-10-26', 130500000, 9),
(42, 'Jupiter Ascending', '2015-02-06', 136000000, 9),
(43, 'Back to the Future Part II', '1989-11-22', 332000000, 10),
(44, 'Cast Away', '2000-12-22', 429600000, 10),
(45, 'The Green Mile', '1999-12-10', 136000000, 11),
(46, 'The Mist', '2007-11-21', 57300000, 11),
(47, 'Indiana Jones and the Kingdom of the Crystal Skull', '2008-05-22', 786600000, 12),
(48, 'The Lost World: Jurassic Park', '1997-05-23', 618000000, 12),
(49, 'Inglourious Basterds', '2009-08-21', 321500000, 13),
(50, 'Django Unchained', '2012-12-25', 425400000, 13),
(51, 'Se7en', '1995-09-22', 327311000, 14),
(52, 'Gone Girl', '2014-10-03', 369000000, 14),
(53, 'The Godfather Part III', '1990-12-25', 136000000, 15),
(54, 'Apocalypse Now', '1979-08-15', 150000000, 15),
(55, 'Return of the Jedi', '1983-05-25', 475000000, 16),
(56, 'Serenity', '2005-09-30', 39000000, 17),
(57, 'Much Ado About Nothing', '2012-06-07', 46000000, 17),
(58, 'The Phantom Menace', '1999-05-19', 1025000000, 18),
(59, 'The Revenge of the Sith', '2005-05-19', 868000000, 18);

INSERT INTO movie_fact (factId, movieId, factText) VALUES

(1, 1, 'The first feature-length film made entirely with CGI.'),
(2, 2, 'The highest-grossing traditionally animated film of all time.'),
(3, 3, 'Won 5 Academy Awards, including Best Picture.'),
(4, 4, 'The first film to reach the billion-dollar mark.'),
(5, 5, 'Won all 11 Academy Awards it was nominated for.'),
(6, 6, 'The rotating hallway fight scene was done without CGI.'),
(7, 7, 'Famous for its twist ending.'),
(8, 8, 'Johnny Depp based his character on Keith Richards.'),
(9, 9, 'Popularized the "bullet time" visual effect.'),
(10, 10, 'The bench scenes were filmed in Savannah, Georgia.'),
(11, 11, 'Based on a novella by Stephen King.'),
(12, 12, 'Used groundbreaking CGI for the dinosaurs.'),
(13, 13, 'Heath Ledger won a posthumous Oscar for his role as the Joker.'),

(14, 13, 'The scene when joker messes with a detonator switch was improv, as there was an actual delay before the pyrotechnics went off.'),
(15, 14, 'Won the Palme dOr at the Cannes Film Festival.'),
(16, 15, 'The first rule of Fight Club is: You do not talk about Fight Club.'),
(17, 16, 'Marlon Brando used cue cards for his lines.'),
(18, 17, 'The first sequel to win the Best Picture Oscar.'),
(19, 18, 'Considered the best Star Wars film by many fans.'),
(20, 19, 'Michael J. Fox filmed the movie while still starring in a TV show.'),
(21, 20, 'The first Marvel film to gross over \$1 billion.'),
(22, 21, 'Sean Connery played Indiana Jones father, despite being only 12 years older than Harrison Ford.'),
(23, 22, 'The original title was "The Star Wars."'),
(24, 22, 'In the original theatrical cut there is a stormtrooper who hits his head.'),
(25, 23, 'Filmed simultaneously with the other two movies in the trilogy.'),
(26, 24, 'Features the epic Battle of Helms Deep.');

So i actually created the file with db browser but then prepopulated my database with it.

Query One

In my Movie dao i used this query for my search functionality

```
@Query("SELECT * FROM movie WHERE title LIKE :query")  
suspend fun searchMovies(query: String): List<Movie> // Now returns List<Movie>
```

In the app you just type in your search and the recycler view will update with a subset of options some neat code involved with this one.

Query Two

I used a neat little room feature that maps relationships for you, i made a custom class

```
data class DirectorMovie (  
    @Embedded val director: Director,  
    @Relation(  
        parentColumn = "directorId",  
        entityColumn = "directorId"  
    )  
    val movies: List<Movie>  
)
```

And then in my dao i used this query that will get a list of directors with a list of their movies

```
@Dao  
interface DirectorMovieDao {  
    @Transaction  
    @Query("select * from director")  
    suspend fun getDirectorMovies(): List<DirectorMovie>  
}
```

And then i logged it in the main activity. Pretty cool.

Discloser It took me a long time to set all the code up and i had a lot of help from chat gpt. But

Carson Wardell Project 3

don't get me wrong it was a lot of work still to make sure the data objects, dao, repository and model view was working and to use it in my activities.

But if you use the app inspector to look at the database and run the app and do all the functions you will see that the database is functional.