**IMDb Database Final Project**

Aiden Cary, Bradley Turnage, Moulc Doumbia

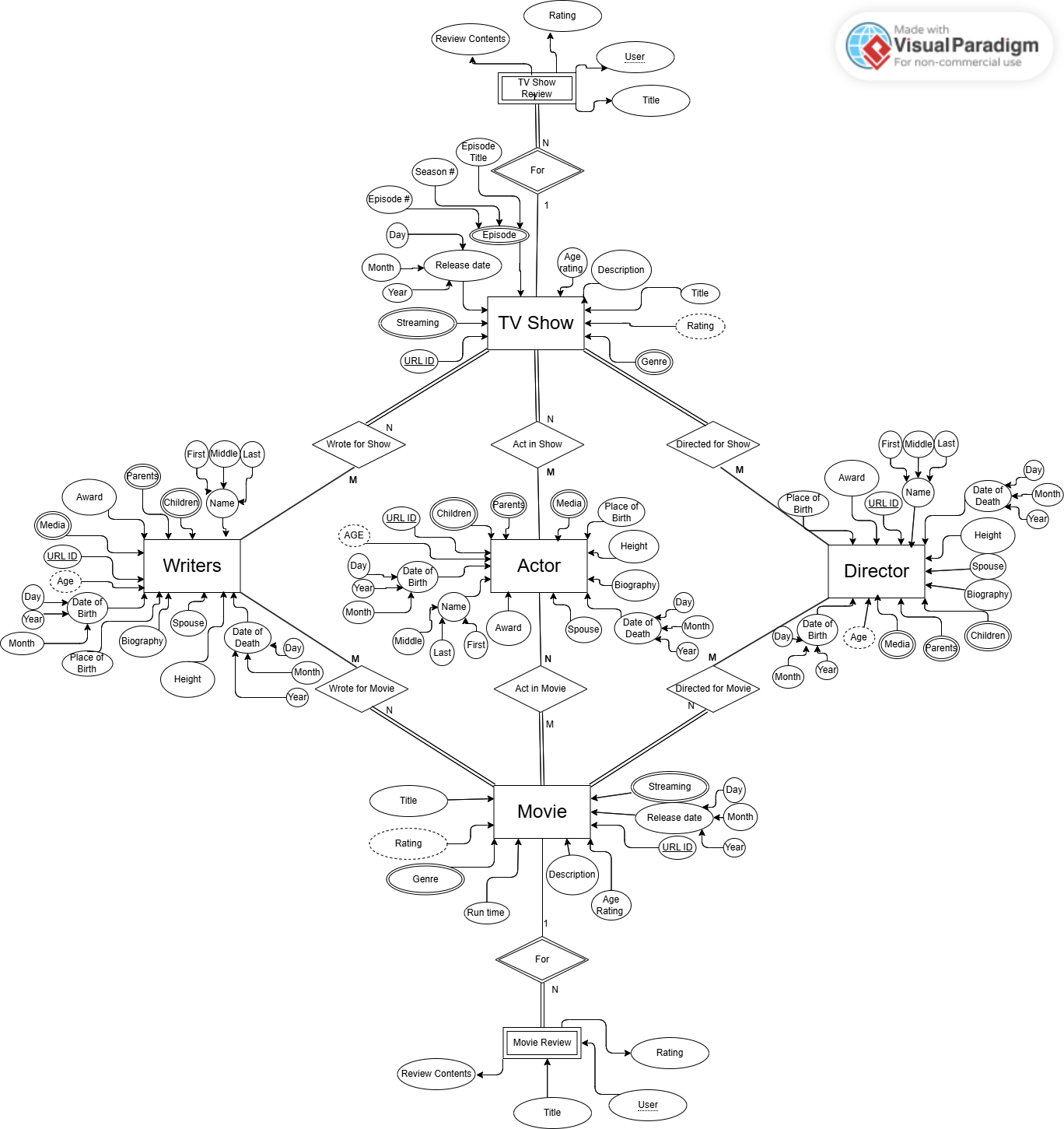
Department of Computer Science and Engineering, University of Central Arkansas

CSCI 3360: Database Systems

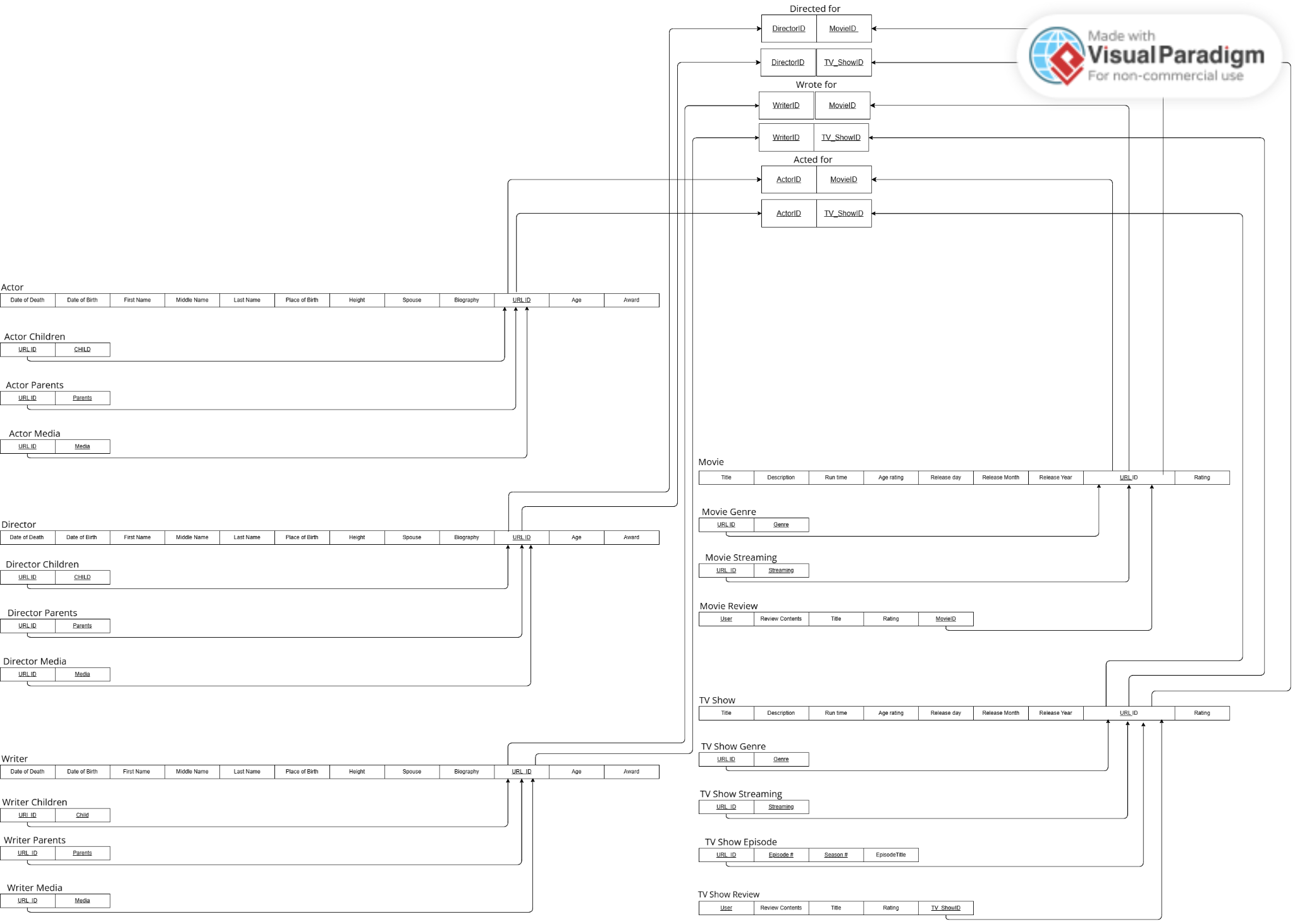
Dr. Bernard Chen

April 24, 2025

**IMDb Database Final Project**

**ER Diagram:**

**Relational Schema Diagram:**



**Oracle DDL Code:**

**-- IMDb Database**

**-- M:N relationships between Movie, TV Show <---> Actor, Writer, Director**

**-- 1:N relationships between Movie <---> Movie\_Genre, Movie\_Streaming, Movie\_Review & TV\_Show <---> TV\_Show\_Episode, TV\_Show\_Genre, TV\_Show\_Streaming, TV\_Show\_Review**

**-- Multivalue tables between Actor, Writer, Director <---> Actor\_Children, Actor\_Parents, Actor\_Media, Writer\_Children, Writer\_Parents, Writer\_Media, Director\_Children, Director\_Parents, Director\_Media**

**-- Derived attributes for Actor, Writer, Director (Age), Movie, TV Show (Average Rating)**

**-- Movie Drop Tables**

**DROP TABLE Movie CASCADE CONSTRAINTS;**

**DROP TABLE Movie\_Genre CASCADE CONSTRAINTS;**

**DROP TABLE Movie\_Streaming CASCADE CONSTRAINTS;**

**DROP TABLE Movie\_Review CASCADE CONSTRAINTS;**

**-- TV Show Drop Tables**

**DROP TABLE TV\_Show CASCADE CONSTRAINTS;**

**DROP TABLE TV\_Show\_Episode CASCADE CONSTRAINTS;**

**DROP TABLE TV\_Show\_Genre CASCADE CONSTRAINTS;**

**DROP TABLE TV\_Show\_Streaming CASCADE CONSTRAINTS;**

**DROP TABLE TV\_Show\_Review CASCADE CONSTRAINTS;**

**-- Actor Drop Tables**

**DROP TABLE Actor CASCADE CONSTRAINTS;**

**DROP TABLE Actor\_Children CASCADE CONSTRAINTS;**

**DROP TABLE Actor\_Parents CASCADE CONSTRAINTS;**

**DROP TABLE Actor\_Media CASCADE CONSTRAINTS;**

**DROP TABLE Actor\_Movie CASCADE CONSTRAINTS;**

**DROP TABLE Actor\_TV\_Show CASCADE CONSTRAINTS;**

**-- Writer Drop Tables**

**DROP TABLE Writer CASCADE CONSTRAINTS;**

**DROP TABLE Writer\_Children CASCADE CONSTRAINTS;**

**DROP TABLE Writer\_Parents CASCADE CONSTRAINTS;**

**DROP TABLE Writer\_Media CASCADE CONSTRAINTS;**

**DROP TABLE Writer\_Movie CASCADE CONSTRAINTS;**

**DROP TABLE Writer\_TV\_Show CASCADE CONSTRAINTS;**

**-- Director Drop Tables**

**DROP TABLE Director CASCADE CONSTRAINTS;**

**DROP TABLE Director\_Children CASCADE CONSTRAINTS;**

**DROP TABLE Director\_Parents CASCADE CONSTRAINTS;**

**DROP TABLE Director\_Media CASCADE CONSTRAINTS;**

**DROP TABLE Director\_Movie CASCADE CONSTRAINTS;**

**DROP TABLE Director\_TV\_Show CASCADE CONSTRAINTS;**

**-- Drop views for derived attributes**

**DROP VIEW Director\_With\_Age;**

**DROP VIEW Actor\_With\_Age;**

**DROP VIEW Writer\_With\_Age;**

**DROP VIEW Movie\_Average\_Rating;**

**DROP VIEW TV\_Show\_Average\_Rating;**

**------------------------------------------------------------**

**-- Movie table**

**CREATE TABLE Movie(**

**Title VARCHAR2(100),**

**ReleaseDate DATE,**

**RunTime NUMBER(5, 1),**

**Description VARCHAR2(2000),**

**AgeRating VARCHAR2(5),**

**URLID CHAR(9) CONSTRAINT Movie\_MovieID\_PK PRIMARY KEY**

**);**

**-- Movie\_Genre table**

**CREATE TABLE Movie\_Genre(**

**MovieID CHAR(9),**

**Genre VARCHAR2(100),**

**CONSTRAINT Movie\_Genre\_PK PRIMARY KEY (MovieID, Genre)**

**);**

**-- Movie\_Streaming table**

**CREATE TABLE Movie\_Streaming(**

**MovieID CHAR(9),**

**Streaming VARCHAR2(100),**

**CONSTRAINT Movie\_Streaming\_PK PRIMARY KEY (MovieID, Streaming)**

**);**

**-- Movie\_Review table**

**CREATE TABLE Movie\_Review (**

**MovieID CHAR(9),**

**Title VARCHAR2(100),**

**Reviewer VARCHAR2(100),**

**ReviewContents VARCHAR2(1000),**

**Rating NUMBER(2) CHECK (Rating BETWEEN 0 AND 10),**

**CONSTRAINT Movie\_Review\_PK PRIMARY KEY (MovieID, Reviewer)**

**);**

**-- TV\_Show Table**

**CREATE TABLE TV\_Show(**

**Title VARCHAR2(100),**

**ReleaseDate DATE,**

**Description VARCHAR2(2000),**

**AgeRating VARCHAR2(5),**

**URLID CHAR(9) CONSTRAINT TV\_Show\_TV\_ShowID\_PK PRIMARY KEY**

**);**

**-- TV\_Show\_Episode table**

**CREATE TABLE TV\_Show\_Episode(**

**TV\_ShowID CHAR(9),**

**EpisodeTitle VARCHAR2(100),**

**EpisodeNumber NUMBER(5),**

**SeasonNumber NUMBER(2),**

**CONSTRAINT TV\_Show\_Episode\_PK PRIMARY KEY (TV\_ShowID, EpisodeNumber, SeasonNumber)**

**);**

**-- Genre table for TV\_Show**

**CREATE TABLE TV\_Show\_Genre(**

**TV\_ShowID CHAR(9),**

**Genre VARCHAR2(100),**

**CONSTRAINT TV\_Show\_Genre\_PK PRIMARY KEY (TV\_ShowID, Genre)**

**);**

**-- Streaming table for TV\_Show**

**CREATE TABLE TV\_Show\_Streaming(**

**TV\_ShowID CHAR(9),**

**Streaming VARCHAR2(100),**

**CONSTRAINT TV\_Show\_Streaming\_PK PRIMARY KEY (TV\_ShowID, Streaming)**

**);**

**CREATE TABLE TV\_Show\_Review (**

**TV\_ShowID CHAR(9),**

**Title VARCHAR2(100),**

**Reviewer VARCHAR2(100),**

**ReviewContents VARCHAR2(1000),**

**Rating NUMBER(2) CHECK (Rating BETWEEN 0 AND 10),**

**CONSTRAINT TV\_Show\_Review\_PK PRIMARY KEY (TV\_ShowID, Reviewer)**

**);**

**-- Director table**

**CREATE TABLE Director(**

**URLID CHAR(9) CONSTRAINT Director\_DirectorID\_PK PRIMARY KEY,**

**FirstName VARCHAR2(100),**

**MiddleName VARCHAR2(100) DEFAULT NULL,**

**LastName VARCHAR2(100) DEFAULT NULL,**

**PlaceOfBirth VARCHAR2(40),**

**Height NUMBER(2),**

**Spouse VARCHAR2(100),**

**Award VARCHAR2(100),**

**Bdate DATE,**

**Biography VARCHAR2(2000),**

**DateOfBirth DATE,**

**DateOfDeath DATE DEFAULT NULL**

**);**

**-- Director\_Children table**

**CREATE TABLE Director\_Children(**

**DirectorID CHAR(9),**

**Child VARCHAR2(100),**

**CONSTRAINT Director\_Children\_PK PRIMARY KEY (DirectorID, Child)**

**);**

**-- Director\_Parents table**

**CREATE TABLE Director\_Parents(**

**DirectorID CHAR(9),**

**Parent VARCHAR2(100),**

**CONSTRAINT Director\_Parents\_PK PRIMARY KEY (DirectorID, Parent)**

**);**

**-- Director\_Media table**

**CREATE TABLE Director\_Media(**

**DirectorID CHAR(9),**

**Media VARCHAR2(100),**

**CONSTRAINT Director\_Media\_PK PRIMARY KEY (DirectorID, Media)**

**);**

**-- Director\_Movie junction table**

**CREATE TABLE Director\_Movie(**

**DirectorID CHAR(9),**

**MovieID CHAR(9),**

**CONSTRAINT Director\_Movie\_PK PRIMARY KEY (DirectorID, MovieID)**

**);**

**-- Director\_TV\_Show junction table**

**CREATE TABLE Director\_TV\_Show(**

**DirectorID CHAR(9),**

**TV\_ShowID CHAR(9),**

**CONSTRAINT Director\_TV\_Show\_PK PRIMARY KEY (DirectorID, TV\_ShowID)**

**);**

**-- Actor table**

**CREATE TABLE Actor(**

**URLID CHAR(9) CONSTRAINT Actor\_ActorID\_PK PRIMARY KEY,**

**FirstName VARCHAR2(100),**

**MiddleName VARCHAR2(100) DEFAULT NULL,**

**LastName VARCHAR2(100) DEFAULT NULL,**

**PlaceOfBirth VARCHAR2(40),**

**Height NUMBER(2),**

**Spouse VARCHAR2(100),**

**Award VARCHAR2(100),**

**Bdate DATE,**

**Biography VARCHAR2(2000),**

**DateOfBirth DATE,**

**DateOfDeath DATE DEFAULT NULL**

**);**

**-- Actor\_Children table**

**CREATE TABLE Actor\_Children(**

**ActorID CHAR(9),**

**Child VARCHAR2(100),**

**CONSTRAINT Actor\_Children\_PK PRIMARY KEY (ActorID, Child)**

**);**

**-- Actor\_Parents table**

**CREATE TABLE Actor\_Parents(**

**ActorID CHAR(9),**

**Parent VARCHAR2(100),**

**CONSTRAINT Actor\_Parents\_PK PRIMARY KEY (ActorID, Parent)**

**);**

**-- Actor\_Media table**

**CREATE TABLE Actor\_Media(**

**ActorID CHAR(9),**

**Media VARCHAR2(100),**

**CONSTRAINT Actor\_Media\_PK PRIMARY KEY (ActorID, Media)**

**);**

**-- Actor\_Movie junction table**

**CREATE TABLE Actor\_Movie(**

**ActorID CHAR(9),**

**MovieID CHAR(9),**

**CONSTRAINT Actor\_Movie\_PK PRIMARY KEY (ActorID, MovieID)**

**);**

**-- Actor\_TV\_Show junction table**

**CREATE TABLE Actor\_TV\_Show(**

**ActorID CHAR(9),**

**TV\_ShowID CHAR(9),**

**CONSTRAINT Actor\_TV\_Show\_PK PRIMARY KEY (ActorID, TV\_ShowID)**

**);**

**-- Writer table**

**CREATE TABLE Writer (**

**URLID CHAR(9) CONSTRAINT Writer\_WriterID\_PK PRIMARY KEY,**

**FirstName VARCHAR2(100),**

**MiddleName VARCHAR2(100) DEFAULT NULL,**

**LastName VARCHAR2(100) DEFAULT NULL,**

**PlaceOfBirth VARCHAR2(40),**

**Height NUMBER(2),**

**Spouse VARCHAR2(100),**

**Award VARCHAR2(100),**

**Bdate DATE,**

**Biography VARCHAR2(2000),**

**DateOfBirth DATE,**

**DateOfDeath DATE DEFAULT NULL**

**);**

**-- Writer\_Children table**

**CREATE TABLE Writer\_Children(**

**WriterID CHAR(9),**

**Child VARCHAR2(100),**

**CONSTRAINT Writer\_Children\_PK PRIMARY KEY (WriterID, Child)**

**);**

**-- Writer\_Parents table**

**CREATE TABLE Writer\_Parents(**

**WriterID CHAR(9),**

**Parent VARCHAR2(100),**

**CONSTRAINT Writer\_Parents\_PK PRIMARY KEY (WriterID, Parent)**

**);**

**-- Writer\_Media table**

**CREATE TABLE Writer\_Media(**

**WriterID CHAR(9),**

**Media VARCHAR2(100),**

**CONSTRAINT Writer\_Media\_PK PRIMARY KEY (WriterID, Media)**

**);**

**-- Writer\_Movie junction table**

**CREATE TABLE Writer\_Movie(**

**WriterID CHAR(9),**

**MovieID CHAR(9),**

**CONSTRAINT Writer\_Movie\_PK PRIMARY KEY (WriterID, MovieID)**

**);**

**-- Writer\_TV\_Show junction table**

**CREATE TABLE Writer\_TV\_Show(**

**WriterID CHAR(9),**

**TV\_ShowID CHAR(9),**

**CONSTRAINT Writer\_TV\_Show\_PK PRIMARY KEY (WriterID, TV\_ShowID)**

**);**

**----------------------------------------------------------------------**

**-- Director Foreign Keys**

**-- Foreign Key Director\_Children DirectorID that points to the URLID of the Director table**

**ALTER TABLE Director\_Children**

**ADD CONSTRAINT director\_children\_fk FOREIGN KEY(DirectorID)**

**REFERENCES Director(URLID);**

**-- Foreign Key Director\_Parents DirectorID that points to the URLID of the Director table**

**ALTER TABLE Director\_Parents**

**ADD CONSTRAINT director\_parents\_fk FOREIGN KEY(DirectorID)**

**REFERENCES Director(URLID);**

**-- Foreign Key Director\_Media DirectorID that points to the URLID of the Director table**

**ALTER TABLE Director\_Media**

**ADD CONSTRAINT director\_media\_fk FOREIGN KEY(DirectorID)**

**REFERENCES Director(URLID);**

**-- Foreign Key Director\_Movie DirectorID that points to the URLID of the Director table**

**ALTER TABLE Director\_Movie**

**ADD CONSTRAINT director\_movie\_director\_fk FOREIGN KEY(DirectorID)**

**REFERENCES Director(URLID);**

**-- Foreign Key Director\_Movie MovieID that points to the URLID of the Movie table**

**ALTER TABLE Director\_Movie**

**ADD CONSTRAINT director\_movie\_movie\_fk FOREIGN KEY(MovieID)**

**REFERENCES Movie(URLID);**

**-- Foreign Key for Director\_TV\_Show: DirectorID references Director(URLID)**

**ALTER TABLE Director\_TV\_Show**

**ADD CONSTRAINT director\_tv\_show\_director\_fk FOREIGN KEY (DirectorID)**

**REFERENCES Director(URLID);**

**-- Foreign Key for Director\_TV\_Show: TV\_ShowID references TV\_Show(URLID)**

**ALTER TABLE Director\_TV\_Show**

**ADD CONSTRAINT director\_tv\_show\_tv\_show\_fk FOREIGN KEY (TV\_ShowID)**

**REFERENCES TV\_Show(URLID);**

**-- Actor Foreign Keys**

**-- Foreign Key Actor\_Children ActorID that points to the URLID of the Actor table**

**ALTER TABLE Actor\_Children**

**ADD CONSTRAINT actor\_children\_fk FOREIGN KEY (ActorID)**

**REFERENCES Actor(URLID);**

**-- Foreign Key Actor\_Parents ActorID that points to the URLID of the Actor table**

**ALTER TABLE Actor\_Parents**

**ADD CONSTRAINT actor\_parents\_fk FOREIGN KEY (ActorID)**

**REFERENCES Actor(URLID);**

**-- Foreign Key Actor\_Media ActorID that points to the URLID of the Actor table**

**ALTER TABLE Actor\_Media**

**ADD CONSTRAINT actor\_media\_fk FOREIGN KEY (ActorID)**

**REFERENCES Actor(URLID);**

**-- Foreign Key Actor\_Movie ActorID that points to the URLID of the Actor table**

**ALTER TABLE Actor\_Movie**

**ADD CONSTRAINT actor\_movie\_actor\_fk FOREIGN KEY (ActorID)**

**REFERENCES Actor(URLID);**

**-- Foreign Key Actor\_Movie MovieID that points to the URLID of the Movie table**

**ALTER TABLE Actor\_Movie**

**ADD CONSTRAINT actor\_movie\_movie\_fk FOREIGN KEY (MovieID)**

**REFERENCES Movie(URLID);**

**-- Foreign Key Actor\_TV\_Show ActorID that points to the URLID of the Actor table**

**ALTER TABLE Actor\_TV\_Show**

**ADD CONSTRAINT actor\_tv\_show\_actor\_fk FOREIGN KEY (ActorID)**

**REFERENCES Actor(URLID);**

**-- Foreign Key Actor\_TV\_Show TV\_ShowID that points to the URLID of the TV\_Show table**

**ALTER TABLE Actor\_TV\_Show**

**ADD CONSTRAINT actor\_tv\_show\_tv\_show\_fk FOREIGN KEY (TV\_ShowID)**

**REFERENCES TV\_Show(URLID);**

**-- Writer Foreign Keys**

**-- Foreign Key Writer\_Children WriterID that points to the URLID of the Writer table**

**ALTER TABLE Writer\_Children**

**ADD CONSTRAINT writer\_children\_fk FOREIGN KEY (WriterID)**

**REFERENCES Writer(URLID);**

**-- Foreign Key Writer\_Parents WriterID that points to the URLID of the Writer table**

**ALTER TABLE Writer\_Parents**

**ADD CONSTRAINT writer\_parents\_fk FOREIGN KEY (WriterID)**

**REFERENCES Writer(URLID);**

**-- Foreign Key Writer\_Media WriterID that points to the URLID of the Writer table**

**ALTER TABLE Writer\_Media**

**ADD CONSTRAINT writer\_media\_fk FOREIGN KEY (WriterID)**

**REFERENCES Writer(URLID);**

**-- Foreign Key Writer\_Movie WriterID that points to the URLID of the Writer table**

**ALTER TABLE Writer\_Movie**

**ADD CONSTRAINT writer\_movie\_writer\_fk FOREIGN KEY (WriterID)**

**REFERENCES Writer(URLID);**

**-- Foreign Key Writer\_Movie MovieID that points to the URLID of the Movie table**

**ALTER TABLE Writer\_Movie**

**ADD CONSTRAINT writer\_movie\_movie\_fk FOREIGN KEY (MovieID)**

**REFERENCES Movie(URLID);**

**-- Foreign Key Writer\_TV\_Show WriterID that points to the URLID of the Writer table**

**ALTER TABLE Writer\_TV\_Show**

**ADD CONSTRAINT writer\_tv\_show\_writer\_fk FOREIGN KEY (WriterID)**

**REFERENCES Writer(URLID);**

**-- Foreign Key Writer\_TV\_Show TV\_ShowID that points to the URLID of the TV\_Show table**

**ALTER TABLE Writer\_TV\_Show**

**ADD CONSTRAINT writer\_tv\_show\_tv\_show\_fk FOREIGN KEY (TV\_ShowID)**

**REFERENCES TV\_Show(URLID);**

**-- Movie Foreign Keys**

**-- Foreign Key Movie GenreID that points to the GenreID of the Movie\_Genre table**

**ALTER TABLE Movie\_Genre**

**ADD CONSTRAINT movie\_genre\_fk FOREIGN KEY(MovieID)**

**REFERENCES Movie(URLID);**

**-- Foreign Key Movie StreamingID that points to the StreamingID of the Movie\_Streaming table**

**ALTER TABLE Movie\_Streaming**

**ADD CONSTRAINT movie\_streaming\_fk FOREIGN KEY(MovieID)**

**REFERENCES Movie(URLID);**

**-- Foreign Key Movie\_Review MovieID that points to the URLID of the Movie table**

**ALTER TABLE Movie\_Review**

**ADD CONSTRAINT movie\_review\_fk FOREIGN KEY (MovieID)**

**REFERENCES Movie(URLID) ON DELETE CASCADE;**

**-- TV Show Foreign Keys**

**-- Foreign Key TV\_Show\_Genre TV\_ShowID that points to the URLID of the TV\_Show table**

**ALTER TABLE TV\_Show\_Genre**

**ADD CONSTRAINT tv\_show\_genre\_tv\_show\_fk FOREIGN KEY (TV\_ShowID)**

**REFERENCES TV\_Show(URLID);**

**-- Foreign Key TV\_Show\_Streaming TV\_ShowID that points to the URLID of the TV\_Show table**

**ALTER TABLE TV\_Show\_Streaming**

**ADD CONSTRAINT tv\_show\_streaming\_tv\_show\_fk FOREIGN KEY (TV\_ShowID)**

**REFERENCES TV\_Show(URLID);**

**-- Foreign Key TV\_Show\_Review TV\_ShowID that points to the URLID of the TV\_Show table**

**ALTER TABLE TV\_Show\_Review**

**ADD CONSTRAINT tv\_show\_review\_fk FOREIGN KEY (TV\_ShowID)**

**REFERENCES TV\_Show(URLID) ON DELETE CASCADE;**

**-- Derived Attributes/Functions**

**-- Example query to calculate the age of a director**

**-- SELECT URLID, Age FROM Director\_With\_Age;**

**-- View to calculate the Age derived attribute for each director**

**CREATE VIEW Director\_With\_Age AS**

**SELECT**

**D.\*,**

**FLOOR(MONTHS\_BETWEEN(SYSDATE, D.DateOfBirth) / 12) AS Age**

**FROM**

**Director D;**

**-- Example query to calculate the age of an actor**

**-- SELECT URLID, Age FROM Actor\_With\_Age;**

**-- View to calculate the Age derived attribute for each actor**

**CREATE VIEW Actor\_With\_Age AS**

**SELECT**

**A.\*,**

**FLOOR(MONTHS\_BETWEEN(SYSDATE, A.DateOfBirth) / 12) AS Age**

**FROM**

**Actor A;**

**-- Example query to calculate the age of a writer**

**-- SELECT URLID, Age FROM Writer\_With\_Age;**

**-- View to calculate the Age derived attribute for each writer**

**CREATE VIEW Writer\_With\_Age AS**

**SELECT**

**W.\*,**

**FLOOR(MONTHS\_BETWEEN(SYSDATE, W.DateOfBirth) / 12) AS Age**

**FROM**

**Writer W;**

**-- View to calculate the average review rating for Movies**

**CREATE VIEW Movie\_Average\_Rating AS**

**SELECT**

**M.URLID AS MovieID,**

**M.Title,**

**NVL(AVG(R.Rating), 0) AS AverageRating**

**FROM**

**Movie M**

**LEFT JOIN**

**Movie\_Review R ON M.URLID = R.MovieID**

**GROUP BY**

**M.URLID, M.Title;**

**-- Example query to calculate the average review rating for Movies**

**-- SELECT MovieID, AverageRating FROM Movie\_Average\_Rating;**

**-- View to calculate the average review rating for TV shows**

**CREATE VIEW TV\_Show\_Average\_Rating AS**

**SELECT**

**T.URLID AS TV\_ShowID,**

**T.Title,**

**NVL(AVG(R.Rating), 0) AS AverageRating**

**FROM**

**TV\_Show T**

**LEFT JOIN**

**TV\_Show\_Review R ON T.URLID = R.TV\_ShowID**

**GROUP BY**

**T.URLID, T.Title;**

**-- Example query to calculate the average review rating for TV Shows**

**-- SELECT TV\_ShowID, AverageRating FROM TV\_Show\_Average\_Rating;**

**Oracle Inserts and Results:**

--------------------------------------------------------------------------

-- Inserts for Actor

INSERT INTO Actor (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('A000001', 'Zendaya', 'Coleman', 'Oakland, CA', TO\_DATE('1996-09-01', 'YYYY-MM-DD'), 70, 'Not Married', 'Primetime Emmy Award', 'Zendaya (which means "to give thanks" in the language of Shona) is an American actress and singer born in Oakland, California. She began her career appearing as a child model working for Macys, Mervyns and Old Navy. She was a backup dancer before gaining prominence for her role as Rocky Blue on the Disney Channel sitcom Shake It Up (2010) which also includes Bella Thorne, Kenton Duty and Roshon Fegan. Zendaya was a contestant on the sixteenth season of the competition series Dancing with the Stars. She went on to produce and star as K.C. Cooper in the Disney Channel sitcom K.C. Undercover (2015) She made her film breakthrough in 2017, starring as Michelle "MJ" Jones in the Marvel Cinematic Universe superhero film Spider-Man: Homecoming (2017) and as Anne Wheeler in the musical drama film The Greatest Showman (2017) alongside actors such as Tom Holland, Hugh Jackman and Zac Efron. Besides acting, singing and dancing she is an ambassador for Convoy of Hope. She has written a book, launched her own clothing line (Daya by Zendaya) and proved herself to be a great role model for young girls all around the world.');

INSERT INTO Actor (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('A000002', 'Timothee', 'Chalamet', 'New York, NY', TO\_DATE('1995-12-27', 'YYYY-MM-DD'), 71, 'Not Married', 'Academy Award Nominee', 'Timothée Hal Chalamet was born in Manhattan, to Nicole Flender, a real estate broker and dancer, and Marc Chalamet, a UNICEF editor. His mother, who is from New York, is Jewish, and of Russian Jewish and Polish Jewish descent. His father, who is from Nîmes, France, is of French and English ancestry. He is the brother of actress Pauline Chalamet, a nephew of director Rodman Flender, and a grandson of screenwriter Harold Flender. He grew up in an artistic family, appearing in commercials and the New York theatre scene, and attending the LaGuardia High School of Music, Art and Performing Arts, where his classmate and friend was actor Ansel Elgort (the two later received their first Golden Globe nominations in the same year, 2017). For a time, Timothée also attended Columbia University. He made his film debut in 2014, as a high school student in Jason Reitmans Men, Women & Children (2014) and Matthew McConaugheys characters teenage son in Interstellar (2014). He subsequently had sizable roles in several indie films, playing the younger version of writer Stephen Elliott in The Adderall Diaries (2015), the male lead, Zac, in the drama One and Two (2015), and Billy in the road trip drama Miss Stevens (2016). On stage, he has appeared in the plays The Talls, by Anna Kerrigan, and John Patrick Shanleys autobiographical Prodigal Son, while on television, he has had a minor role in the film Loving Leah (2009), a big part in Law & Order (1990), and meatier roles on the shows Royal Pains (2009) and Homeland (2011), among other work. ');

INSERT INTO Actor (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('A000003', 'Matt', 'Damon', 'Cambridge, MA', TO\_DATE('1970-10-08', 'YYYY-MM-DD'), 70, 'Luciana Barroso', 'Academy Award for Best Screenplay', 'Matthew Paige Damon was born on October 8, 1970, in Boston, Massachusetts, to Kent Damon, a stockbroker, realtor and tax preparer, and Nancy Carlsson-Paige, an early childhood education professor at Lesley University. Matt has an older brother, Kyle, a sculptor. His father was of English and Scottish descent, and his mother is of Finnish and Swedish ancestry. The family lived in Newton until his parents divorced in 1973, when Damon and his brother moved with his mother to Cambridge. He grew up in a stable community, and was raised near actor Ben Affleck. Damon attended Cambridge Rindge and Latin School and he performed in a number of theater productions during his time there. He attended Harvard University as an English major. While in Harvard, he kept on skipping classes to pursue acting projects, which included the TNT original film, Rising Son (1990), and prep-school drama, School Ties (1992). It was until his film, Geronimo: An American Legend (1993), was expected to be a big success that he decided to drop out of university completely. Arriving in Hollywood, Matt managed to get his first break with a part in the romantic comedy, Mystic Pizza (1988). However, the film did not do too well and his film career failed to take off. Not letting failure discourage him from acting, he went for another audition, and managed to get a starring role in School Ties (1992). Up next for Matt was a role as a soldier who had problems with drug-addiction in the movie, Courage Under Fire (1996). ');

INSERT INTO Actor (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('A000004', 'Cillian', 'Murphy', 'Douglas, Ireland', TO\_DATE('1976-05-25', 'YYYY-MM-DD'), 71, 'Yvonne McGuinness', 'Irish Film and Television Award', 'Striking Irish actor Cillian Murphy was born in Douglas, Co Cork, the oldest child of Brendan Murphy, who works for the Irish Department of Education, and a mother who is a teacher of French. He has three younger siblings. Murphy was educated at Presentation Brothers College, Cork. He went on to study law at University College Cork, but dropped out after about a year. During this time, Murphy also pursued an interest in music, playing guitar in various bands. Upon leaving University, Murphy joined the Corcadorca Theater Company in Cork, and played the lead role in "Disco Pigs", amongst other plays. Various film roles followed, including a film adaptation of Disco Pigs (2001). However, his big film break came when he was cast in Danny Boyles 28 Days Later (2002), which became a surprise international hit. This performance earned him nominations for Best Newcomer at the Empire Awards and Breakthrough Male Performance at the MTV Movie Awards. Murphy went on to supporting roles in high-profile films such as Cold Mountain (2003) and Girl with a Pearl Earring (2003), and then was cast in two villain roles: Dr. Jonathan Crane, aka The Scarecrow, in Batman Begins (2005) and Jackson Rippner in Red Eye (2005). Although slight in nature for a villain, Murphys piercing blue eyes helped to create creepy performances and critics began to take notice. Manhola Dargis of the New York Times cited Murphy as a "picture-perfect villain", while David Denby of The New Yorker noted he was both "seductive" and "sinister". ');

INSERT INTO Actor (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('A000005', 'Tom', 'Holland', 'Kingston upon Thames, England', TO\_DATE('1996-06-01', 'YYYY-MM-DD'), 68, 'Not Married', 'BAFTA Rising Star Award', 'Thomas Stanley Holland was born in Kingston-upon-Thames, Surrey, to Nicola Elizabeth (Frost), a photographer, and Dominic Holland (Dominic Anthony Holland), who is a comedian and author. His paternal grandparents were from the Isle of Man and Ireland, respectively. He lives with his parents and three younger brothers - Paddy and twins Sam and Harry. Tom attended Donhead Prep School. Then, after a successful eleven plus exam, he became a pupil at Wimbledon College. Having successfully completed his GCSEs, in September 2012 Tom started a two-year course in the BRIT School for Performing Arts & Technology notable for its numerous famous alumni.Holland began dancing at a hip hop class at Nifty Feet Dance School in Wimbledon, London. His potential was spotted by choreographer Lynne Page (who was an Associate to Peter Darling, choreographer of Billy Elliot and Billy Elliot the Musical) when he performed with his dance school as part of the Richmond Dance Festival 2006. After eight auditions and subsequent two years of training, on 28 June 2008 Tom made his West End debut in Billy Elliot the Musical as Michael, Billys best friend. He gave his first performance in the title role of Billy on 8 September 2008 getting rave reviews praising his versatile acting and dancing skills.In September 2008 Tom (together with co-star Tanner Pflueger) appeared on the news programme on channel FIVE and gave his first TV interview. ');

----------------------------------------------------------

-- Inserts for Movie

INSERT INTO Movie (Title, ReleaseDate, RunTime, Description, AgeRating, URLID)

VALUES ('Dune: Part One', TO\_DATE('2021-10-22', 'YYYY-MM-DD'), 155.0, 'Paul Atreides arrives on Arrakis after his father accepts the stewardship of the dangerous planet. However, chaos ensues after a betrayal as forces clash to control melange, a precious resource.', 'PG-13', 'M000001');

INSERT INTO Movie (Title, ReleaseDate, RunTime, Description, AgeRating, URLID)

VALUES ('Dune: Part Two', TO\_DATE('2023-11-03', 'YYYY-MM-DD'), 165.0, 'Paul Atreides unites with the Fremen while on a warpath of revenge against the conspirators who destroyed his family. Facing a choice between the love of his life and the fate of the universe, he endeavors to prevent a terrible future.', 'PG-13', 'M000002');

INSERT INTO Movie (Title, ReleaseDate, RunTime, Description, AgeRating, URLID)

VALUES ('Challengers', TO\_DATE('2024-08-11', 'YYYY-MM-DD'), 120.0, 'Tashi, a former tennis prodigy turned coach, transformed her husband into a champion. But to overcome a recent losing streak and redeem himself, hell need to face off against his former best friend and Tashis ex-boyfriend.', 'R', 'M000003');

INSERT INTO Movie (Title, ReleaseDate, RunTime, Description, AgeRating, URLID)

VALUES ('Interstellar', TO\_DATE('2014-11-07', 'YYYY-MM-DD'), 169.0, 'When Earth becomes uninhabitable in the future, a farmer and ex-NASA pilot, Joseph Cooper, is tasked to pilot a spacecraft, along with a team of researchers, to find a new planet for humans.', 'PG-13', 'M000004');

INSERT INTO Movie (Title, ReleaseDate, RunTime, Description, AgeRating, URLID)

VALUES ('Oppenheimer', TO\_DATE('2023-07-21', 'YYYY-MM-DD'), 180.0, 'A dramatization of the life story of J. Robert Oppenheimer, the physicist who had a large hand in the development of the atomic bombs that brought an end to World War II.', 'R', 'M000005');

INSERT INTO Movie (Title, ReleaseDate, RunTime, Description, AgeRating, URLID)

VALUES ('Saving Private Ryan', TO\_DATE('1998-07-24', 'YYYY-MM-DD'), 169.0, 'Following the Normandy Landings, a group of U.S. soldiers go behind enemy lines to retrieve a paratrooper whose brothers have been killed in action.', 'R', 'M000006');

INSERT INTO Movie (Title, ReleaseDate, RunTime, Description, AgeRating, URLID)

VALUES ('Dunkirk', TO\_DATE('2017-07-21', 'YYYY-MM-DD'), 106.0, 'Allied soldiers from Belgium, the British Commonwealth and Empire, and France are surrounded by the German Army and evacuated during a fierce battle in World War II.', 'PG-13', 'M000007');

INSERT INTO Movie (Title, ReleaseDate, RunTime, Description, AgeRating, URLID)

VALUES ('In the Heart of the Sea', TO\_DATE('2015-12-11', 'YYYY-MM-DD'), 121.0, 'A recounting of a New England whaling ships sinking by a giant whale in 1820, an experience that later inspired the great novel Moby-Dick.', 'PG-13', 'M000008');

INSERT INTO Movie (Title, ReleaseDate, RunTime, Description, AgeRating, URLID)

VALUES ('Spider-Man: Homecoming', TO\_DATE('2017-07-07', 'YYYY-MM-DD'), 133.0, 'Peter Parker tries to stop Adrian The Vulture Toomes from selling weapons made with advanced Chitauri technology while trying to balance his life as an ordinary high school student.', 'PG-13', 'M000009');

INSERT INTO Movie (Title, ReleaseDate, RunTime, Description, AgeRating, URLID)

VALUES ('Captain America: Civil War', TO\_DATE('2016-05-06', 'YYYY-MM-DD'), 147.0, 'Political involvement in the Avengers affairs causes a rift between Captain America and Iron Man.', 'PG-13', 'M000010');

INSERT INTO Movie (Title, ReleaseDate, RunTime, Description, AgeRating, URLID)

VALUES ('Uncharted', TO\_DATE('2022-02-18', 'YYYY-MM-DD'), 116.0, 'Street-smart Nathan Drake is recruited by seasoned treasure hunter Victor "Sully" Sullivan to recover a fortune amassed by Ferdinand Magellan, and lost 500 years ago by the House of Moncada.', 'PG-13', 'M000011');

------------------------------------------------------------

-- Inserts for TV Show

INSERT INTO TV\_Show (Title, ReleaseDate, Description, AgeRating, URLID)

VALUES ('Euphoria', TO\_DATE('2019-06-16', 'YYYY-MM-DD'), 'Rue Bennett is a drug addict fresh from rehab, whos struggling to stay clean and make sense of her future. Her life changes drastically when she meets Jules Vaughn, a trans girl who recently moved to town following her parents divorce.', 'TV-MA', 'T000001');

INSERT INTO TV\_Show (Title, ReleaseDate, Description, AgeRating, URLID)

VALUES ('Shake It Up', TO\_DATE('2010-11-07', 'YYYY-MM-DD'), 'Two Chicago teens attempt to realize their dream of becoming professional dancers by landing spots on a popular local show.', 'TV-G', 'T000002');

INSERT INTO TV\_Show (Title, ReleaseDate, Description, AgeRating, URLID)

VALUES ('KC Undercover', TO\_DATE('2015-01-18', 'YYYY-MM-DD'), 'An outspoken and confident technology wiz and skilled black belt discovers that both her parents are spies and expect her to follow in their footsteps.', 'TV-Y7', 'T000003');

INSERT INTO TV\_Show (Title, ReleaseDate, Description, AgeRating, URLID)

VALUES ('Intergalactic', TO\_DATE('2021-04-30', 'YYYY-MM-DD'), 'It follows a crew of fierce female convicts who break free and go on the run.', 'TV-MA', 'T000004');

INSERT INTO TV\_Show (Title, ReleaseDate, Description, AgeRating, URLID)

VALUES ('A Man Named Scott', TO\_DATE('2021-11-05', 'YYYY-MM-DD'), 'A day in the life of a man behind an heroic mask.', 'TV-14', 'T000005');

INSERT INTO TV\_Show (Title, ReleaseDate, Description, AgeRating, URLID)

VALUES ('Will & Grace', TO\_DATE('1998-09-21', 'YYYY-MM-DD'), 'Gay lawyer Will and straight interior designer Grace share a New York City apartment. Their best friends are gleeful and proud gay Jack and charismatic, filthy-rich, amoral socialite Karen.', 'TV-14', 'T000006');

INSERT INTO TV\_Show (Title, ReleaseDate, Description, AgeRating, URLID)

VALUES ('Project Green Light', TO\_DATE('2001-12-02', 'YYYY-MM-DD'), 'The second season of Matt Damon and Ben Afflecks movie making reality series goes a different route when its two professionals realizing their vision on screen instead of just one: writer and director.', 'TV-PG', 'T000007');

INSERT INTO TV\_Show (Title, ReleaseDate, Description, AgeRating, URLID)

VALUES ('Years of Living Dangerously', TO\_DATE('2014-04-13', 'YYYY-MM-DD'), 'The Eames House and Studio is explored via a series of slides.', 'TV-PG', 'T000008');

INSERT INTO TV\_Show (Title, ReleaseDate, Description, AgeRating, URLID)

VALUES ('Peaky Blinders', TO\_DATE('2013-09-12', 'YYYY-MM-DD'), 'A gangster family epic set in 1900s England, centering on a gang who sew razor blades in the peaks of their caps, and their fierce boss Tommy Shelby.', 'TV-MA', 'T000009');

INSERT INTO TV\_Show (Title, ReleaseDate, Description, AgeRating, URLID)

VALUES ('A World of Calm', TO\_DATE('2020-10-01', 'YYYY-MM-DD'), 'An immersive visual journey into another world.', 'TV-PG', 'T000010');

INSERT INTO TV\_Show (Title, ReleaseDate, Description, AgeRating, URLID)

VALUES ('Atlantic: The Wildest Ocean on Earth', TO\_DATE('2015-07-30', 'YYYY-MM-DD'), 'In this landmark series, the BBC Natural History Unit explores a vast ocean that stretches nearly 10,000 miles, from Arctic to Antarctic and from sun-drenched tropical reefs to crushing abyssal depths.', 'TV-PG', 'T000011');

------------------------------------------------------------

-- Inserts for Actor\_Movie relationships

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000001', 'M000001');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000001', 'M000002');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000001', 'M000003');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000001', 'M000009');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000002', 'M000001');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000002', 'M000002');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000002', 'M000004');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000003', 'M000004');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000003', 'M000005');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000003', 'M000006');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000004', 'M000005');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000004', 'M000007');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000004', 'M000008');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000005', 'M000008');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000005', 'M000009');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000005', 'M000010');

INSERT INTO Actor\_Movie (ActorID, MovieID)

VALUES ('A000005', 'M000011');

------------------------------------------------------------

-- Inserts for Actor\_TV\_Show relationships

INSERT INTO Actor\_TV\_Show (ActorID, TV\_ShowID)

VALUES ('A000001', 'T000001');

INSERT INTO Actor\_TV\_Show (ActorID, TV\_ShowID)

VALUES ('A000001', 'T000002');

INSERT INTO Actor\_TV\_Show (ActorID, TV\_ShowID)

VALUES ('A000001', 'T000003');

INSERT INTO Actor\_TV\_Show (ActorID, TV\_ShowID)

VALUES ('A000002', 'T000004');

INSERT INTO Actor\_TV\_Show (ActorID, TV\_ShowID)

VALUES ('A000002', 'T000005');

INSERT INTO Actor\_TV\_Show (ActorID, TV\_ShowID)

VALUES ('A000003', 'T000006');

INSERT INTO Actor\_TV\_Show (ActorID, TV\_ShowID)

VALUES ('A000003', 'T000007');

INSERT INTO Actor\_TV\_Show (ActorID, TV\_ShowID)

VALUES ('A000003', 'T000008');

INSERT INTO Actor\_TV\_Show (ActorID, TV\_ShowID)

VALUES ('A000004', 'T000009');

INSERT INTO Actor\_TV\_Show (ActorID, TV\_ShowID)

VALUES ('A000004', 'T000010');

INSERT INTO Actor\_TV\_Show (ActorID, TV\_ShowID)

VALUES ('A000004', 'T000011');

------------------------------------------------------------

-- Inserts for Actor Parents

-- Zendaya

INSERT INTO Actor\_Parents (ActorID, Parent)

VALUES ('A000001', 'Claire Stoermer');

INSERT INTO Actor\_Parents (ActorID, Parent)

VALUES ('A000001', 'Kazembe Ajamu Coleman');

-- Timothée Chalamet

INSERT INTO Actor\_Parents (ActorID, Parent)

VALUES ('A000002', 'Nicole Flender');

INSERT INTO Actor\_Parents (ActorID, Parent)

VALUES ('A000002', 'Marc Chalamet');

-- Matt Damon

INSERT INTO Actor\_Parents (ActorID, Parent)

VALUES ('A000003', 'Kent Damon');

INSERT INTO Actor\_Parents (ActorID, Parent)

VALUES ('A000003', 'Nancy Carlsson-Paige');

-- Cillian Murphy

INSERT INTO Actor\_Parents (ActorID, Parent)

VALUES ('A000004', 'Brendan Murphy');

INSERT INTO Actor\_Parents (ActorID, Parent)

VALUES ('A000004', 'Jane Murphy');

-- Tom Holland

INSERT INTO Actor\_Parents (ActorID, Parent)

VALUES ('A000005', 'Dominic Holland');

INSERT INTO Actor\_Parents (ActorID, Parent)

VALUES ('A000005', 'Nicola Frost');

------------------------------------------------------------

-- Inserts for Actor Media

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000001', 'Zendaya Interview on Dune');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000001', 'Zendaya Feature on Euphoria');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000001', 'Zendaya Documentary');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000002', 'Timothée Chalamet Interview on Dune');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000002', 'Timothée Chalamet Feature on Interstellar');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000002', 'Timothée Chalamet Documentary');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000003', 'Matt Damon Interview on Interstellar');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000003', 'Matt Damon Feature on Oppenheimer');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000003', 'Matt Damon Documentary');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000004', 'Cillian Murphy Interview on Oppenheimer');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000004', 'Cillian Murphy Feature on Peaky Blinders');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000004', 'Cillian Murphy Documentary');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000005', 'Tom Holland Interview on Spider-Man: Homecoming');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000005', 'Tom Holland Feature on Uncharted');

INSERT INTO Actor\_Media (ActorID, Media)

VALUES ('A000005', 'Tom Holland Documentary');

------------------------------------------------------------

-- Inserts for Actor Children

INSERT INTO Actor\_Children (ActorID, Child)

VALUES ('A000001', 'No Information Available');

INSERT INTO Actor\_Children (ActorID, Child)

VALUES ('A000002', 'No Information Available');

INSERT INTO Actor\_Children (ActorID, Child)

VALUES ('A000003', 'Isabella Damon');

INSERT INTO Actor\_Children (ActorID, Child)

VALUES ('A000003', 'Gia Zavala Damon');

INSERT INTO Actor\_Children (ActorID, Child)

VALUES ('A000003', 'Stella Damon');

INSERT INTO Actor\_Children (ActorID, Child)

VALUES ('A000004', 'Malachy Murphy');

INSERT INTO Actor\_Children (ActorID, Child)

VALUES ('A000004', 'Aran Murphy');

INSERT INTO Actor\_Children (ActorID, Child)

VALUES ('A000005', 'No Information Available');

------------------------------------------------------------

-- Inserts for Writer

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000001', 'Frank', 'Herbert', 'Tacoma, WA', TO\_DATE('1920-10-08', 'YYYY-MM-DD'), 70, 'Beverly Ann Stuart', 'Hugo Award', 'Frank Herbert was born on October 8, 1920 in Tacoma, Washington, USA. He was a writer, known for Dune: Part One (2021), Dune (1984) and Dune: Part Two (2024). He was married to Theresa Shackleford, Beverley Ann Stuart and Flora Parkinson. He died on February 11, 1986 in Madison, Wisconsin, USA.');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000002', 'Sam', 'Levinson', 'United States', TO\_DATE('1985-01-08', 'YYYY-MM-DD'), 72, 'Ashley Lent Levinson', 'Primetime Emmy Award', 'Sam Levinson was born on January 8, 1985 in the USA. He is a writer and producer, known for Euphoria (2019), The Idol (2023) and Assassination Nation (2018).');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000003', 'Jonathan', 'Nolan', 'London, England', TO\_DATE('1976-06-06', 'YYYY-MM-DD'), 71, 'Lisa Joy', 'Academy Award Nominee', 'Attended Loyola Academy in Wilmette, Illinois, graduating in 1994. Graduated from Georgetown University in Washington, D.C. in 1999. Majored in English. Shortly after graduating from Georgetown University, Jonathan Nolan served as a production assistant on Memento (2000). Wrote the short story, "Memento Mori", on which the film Memento (2000) is based. Was co-writer, with his brother Christopher Nolan, on The Prestige (2006), The Dark Knight (2008) and The Dark Knight Rises (2012). Also co-wrote the Terminator Salvation (2009).');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000004', 'George', 'Kay', 'London, England', TO\_DATE('1975-03-15', 'YYYY-MM-DD'), 69, 'Not Married', 'BAFTA Award', 'George Kay is known for Lupin (2021), Criminal: UK (2019) and Stag (2016).');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000005', 'Robert', 'Alexander', 'Los Angeles, CA', TO\_DATE('1980-08-12', 'YYYY-MM-DD'), 70, 'Not Married', 'Peabody Award', 'Robert passed away 11.04.2018. Alexander Robert Graham 05.01.1942 - 11.04.2018 Actor. Late of Glebe. Son of Bob and Jean Alexander (both deceased), brother of Ann, uncle of Sarah and Matt. Adored partner (for 54 years) of Barry Blight; esteemed actor; accomplished musician; kind mentor to emerging young actors; cherished colleague; intelligent, generous, witty and convivial host to a multitude of fortunate friends. Robert passed peacefully away at the Sacred Heart Hospice on Wednesday evening after a lengthy illness.');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000006', 'Christopher', 'Nolan', 'London, England', TO\_DATE('1970-07-30', 'YYYY-MM-DD'), 71, 'Emma Thomas', 'Academy Award Nominee', 'Best known for his cerebral, often nonlinear, storytelling, acclaimed Academy Award winner writer/director/producer Sir Christopher Nolan CBE was born in London, England. Over the course of more than 25 years of filmmaking, Nolan has gone from low-budget independent films to working on some of the biggest blockbusters ever made and became one of the most celebrated filmmakers of modern cinema. At 7 years old, Nolan began making short films with his fathers Super-8 camera. While studying English Literature at University College London, he shot 16-millimeter films at U.C.L.s film society, where he learned the guerrilla techniques he would later use to make his first feature, Following (1998), on a budget of around $6,000. The noir thriller was recognized at a number of international film festivals prior to its theatrical release and gained Nolan enough credibility that he was able to gather substantial financing for his next film. Nolans second film was Memento (2000), which he directed from his own screenplay based on a short story by his brother Jonathan Nolan. Starring Guy Pearce, the film brought Nolan numerous honors, including Academy Award and Golden Globe Award nominations for Best Original Screenplay. Nolan went on to direct the critically acclaimed psychological thriller, Insomnia (2002), starring Al Pacino, Robin Williams and Hilary Swank. ');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000007', 'Robert', 'Rodat', 'Keene, NH', TO\_DATE('1953-05-15', 'YYYY-MM-DD'), 70, 'Not Married', 'Academy Award Nominee', 'Robert Rodat was born in 1953 in New Hampshire, USA. He is a writer and producer, known for Saving Private Ryan (1998), Falling Skies (2011) and Fly Away Home (1996). He is married to Mollie Miller. They have three children.');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000008', 'Max', 'Mutchnick', 'Chicago, IL', TO\_DATE('1965-11-11', 'YYYY-MM-DD'), 69, 'Not Married', 'Primetime Emmy Award', 'Max Mutchnick is Creator/Executive Producer of "Will & Grace", "Good Morning, Miami" and Executive Producer, "The Stones". For his work on "Will & Grace," he has been honored with an Emmy Award for Outstanding Comedy Series; a Peoples Choice Award (Favorite New Comedy); four Golden Globe nominations (Best Comedy Series); four GLAAD Media Awards (Outstanding TV Comedy Series); a Founders Award from the Viewers for Quality Television, and the National Award for Excellence from the Human Rights Campaign. Creative partners Mutchnick and David Kohan began their careers writing together for "The Dennis Miller Show" and from there ventured forward into comedy series. Producers Linda Bloodworth and Harry Thomason gave the future "Will & Grace" Emmy Award winners their first break into the world of sitcoms when they hired them to work on "Hearts Afire" and "Evening Shade." Their other television credits include "Good Advice" with Shelley Long and the acclaimed "Dream On." In addition to "Will & Grace", they also created and served as executive producers on the comedy series "Boston Common" and "Good Morning, Miami." Currently, David and Max serve as Executive Producers on the new CBS comedy, "The Stones." ');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000009', 'Alex', 'Orlovsky', 'New York, NY', TO\_DATE('1978-03-22', 'YYYY-MM-DD'), 70, 'Not Married', 'Peabody Award', 'Writer of Project Green Light.');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000010', 'Joel', 'Bach', 'Los Angeles, CA', TO\_DATE('1975-06-15', 'YYYY-MM-DD'), 71, 'Not Married', 'Peabody Award', 'Joel Bach is known for Years of Living Dangerously (2014), 60 Minutes (1968) and X Marks the Spot (2020).');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000011', 'Charles', 'Leavitt', 'Los Angeles, CA', TO\_DATE('1963-04-10', 'YYYY-MM-DD'), 70, 'Not Married', 'Academy Award Nominee', 'Charles Leavitt was born in Pittsburgh, Pennsylvania, USA. He is known for Blood Diamond (2006), Warcraft (2016) and Those Who Wish Me Dead (2021).');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000012', 'Steven', 'Knight', 'Birmingham, England', TO\_DATE('1959-08-05', 'YYYY-MM-DD'), 71, 'Not Married', 'BAFTA Award', 'Steven Knight is a British screenwriter and film director. He is best known for screenplays he wrote for the films Dirty Pretty Things (2002) and Eastern Promises (2007), and also directed as well as written the film Locke (2013). Knight is also one of three creators of Who Wants to Be a Millionaire?, a game show that has been remade and aired in around 160 countries worldwide, and has written for BBCs Commercial Breakdown, The Detectives , Peaky Blinders and Taboo. Others films based on screenplays that Knight has written include The Hundred-Foot Journey and Pawn Sacrifice both in 2014.');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000013', 'Jane', 'Root', 'London, England', TO\_DATE('1960-02-12', 'YYYY-MM-DD'), 69, 'Not Married', 'Peabody Award', 'Jane Root is a leading creative executive with a reputation for innovation, determination and delivery. Her work was celebrated by Women in Film & TV when she was awarded the prestigious 2017 Business Award acknowledging her outstanding achievements in the industry both in the UK and abroad. In 2008, Jane created Nutopia, a TV production company based in London and Washington committed to thinking big and focused on ambitious factual series of all types, as well as factually-inspired drama. Nutopia is dedicated to creating ground-breaking, specialist factual content on a global scale. Best known for pioneering a new genre of television - the megadoc - by combining epic cinematography and action-driven drama with high-end documentary and A-list talent. Recent credits include highly praised series The World According to Jeff Goldblum (Disney+), award-winning earth science series One Strange Rock (Nat Geo), critically acclaimed drama doc The Last Czars (Netflix), and Grierson-nominated feature doc The Hunt for Jihadi John (HBO/Channel 4). The company continues to development shows for the biggest broadcasters in the world, alongside some of the highest profile partners in the industry. Jane was President of Discovery Channel US until 2007 where she re-positioned the channel giving it a fresh voice and leading to record ratings. During her time at Discovery, Jane commissioned international hits including Man vs Wild (Born Survivor), Deadliest Catch and Dirty Jobs, as well as turning Planet Earth into a worldwide phenomenon. Before her US adventure, Jane was Controller of BBC2 for 5 years.');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000014', 'Huw', 'Cordey', 'Cardiff, Wales', TO\_DATE('1970-03-15', 'YYYY-MM-DD'), 70, 'Not Married', 'Peabody Award', 'Huw Cordey is known for Dancing with the Birds (2019), Our Planet (2019) and Secret Lives of Orangutans (2024).');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000015', 'Jonathan', 'Goldstein', 'New York, NY', TO\_DATE('1969-09-02', 'YYYY-MM-DD'), 71, 'Not Married', 'Academy Award Nominee', 'Jonathan Goldstein was born on September 2, 1968 in New York City, New York, USA. He is a director and writer, known for Spider-Man: Homecoming (2017), Horrible Bosses (2011) and Game Night (2018). He has been married to Adena Halpern since August 26, 2007. They have one child.');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000016', 'Christopher', 'Markus', 'Buffalo, NY', TO\_DATE('1970-01-02', 'YYYY-MM-DD'), 70, 'Not Married', 'Academy Award Nominee', 'Christopher Markus was born on October 16, 1969 in Buffalo, New York, USA. He is a writer and producer, known for Avengers: Endgame (2019), Avengers: Infinity War (2018) and Captain America: The First Avenger (2011). He has been married to Claire Saunders since October 22, 2011.');

INSERT INTO Writer (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('W000018', 'Rafe', 'Judkins', 'Salt Lake City, UT', TO\_DATE('1983-01-08', 'YYYY-MM-DD'), 72, 'Not Married', 'Peabody Award', 'Rafe Judkins was born on January 8, 1983 in Salt Lake City, Utah, USA. He is a producer and writer, known for The Wheel of Time (2021), Uncharted (2022) and Agents of S.H.I.E.L.D. (2013).');

------------------------------------------------------------

-- Inserts for Writer Parents

-- Frank Herbert

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000001', 'Frank Patrick Herbert');

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000001', 'Eileen Herbert');

-- Sam Levinson

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000002', 'Barry Levinson');

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000002', 'Diana Rhodes');

-- Jonathan Nolan

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000003', 'Brendan James Nolan');

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000003', 'Christina Nolan');

-- George Kay

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000004', 'No Information Available');

-- Robert Alexander

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000005', 'No Information Available');

-- Christopher Nolan

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000006', 'Brendan James Nolan');

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000006', 'Christina Nolan');

-- Robert Rodat

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000007', 'No Information Available');

-- Max Mutchnick

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000008', 'No Information Available');

-- Alex Orlovsky

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000009', 'No Information Available');

-- Joel Bach

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000010', 'No Information Available');

-- Charles Leavitt

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000011', 'No Information Available');

-- Steven Knight

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000012', 'No Information Available');

-- Jane Root

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000013', 'No Information Available');

-- Huw Cordey

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000014', 'No Information Available');

-- Jonathan Goldstein

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000015', 'No Information Available');

-- Christopher Markus

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000016', 'No Information Available');

-- Rafe Judkins

INSERT INTO Writer\_Parents (WriterID, Parent)

VALUES ('W000018', 'No Information Available');

------------------------------------------------------------

-- Inserts for Writer Media

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000001', 'Frank Herbert Biography');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000001', 'Dune Documentary Featuring Frank Herbert');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000002', 'Sam Levinson Interview on Euphoria');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000003', 'Jonathan Nolan Interview on Interstellar');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000004', 'George Kay Feature on Intergalactic');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000005', 'Robert Alexander Documentary on A Man Named Scott');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000006', 'Christopher Nolan Interview on Oppenheimer');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000007', 'Robert Rodat Feature on Saving Private Ryan');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000008', 'Max Mutchnick Interview on Will & Grace');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000009', 'Alex Orlovsky Feature on Project Green Light');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000010', 'Joel Bach Documentary on Years of Living Dangerously');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000011', 'Charles Leavitt Feature on Heart of the Sea');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000012', 'Steven Knight Documentary on Peaky Blinders');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000013', 'Jane Root Feature on A World of Calm');

INSERT INTO Writer\_Media (WriterID, Media)

VALUES ('W000014', 'Huw Cordey Documentary on Atlantic: The Wildest Ocean on Earth');

------------------------------------------------------------

-- Inserts for Writer Children

-- Frank Herbert

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000001', 'Brian Herbert');

-- Sam Levinson

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000002', 'No Information Available');

-- Jonathan Nolan

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000003', 'No Information Available');

-- George Kay

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000004', 'No Information Available');

-- Robert Alexander

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000005', 'No Information Available');

-- Christopher Nolan

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000006', 'Rory Nolan');

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000006', 'Magnus Nolan');

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000006', 'Oliver Nolan');

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000006', 'Flora Nolan');

-- Robert Rodat

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000007', 'No Information Available');

-- Max Mutchnick

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000008', 'No Information Available');

-- Alex Orlovsky

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000009', 'No Information Available');

-- Joel Bach

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000010', 'No Information Available');

-- Charles Leavitt

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000011', 'No Information Available');

-- Steven Knight

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000012', 'No Information Available');

-- Jane Root

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000013', 'No Information Available');

-- Huw Cordey

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000014', 'No Information Available');

-- Jonathan Goldstein

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000015', 'No Information Available');

-- Christopher Markus

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000016', 'No Information Available');

-- Rafe Judkins

INSERT INTO Writer\_Children (WriterID, Child)

VALUES ('W000018', 'No Information Available');

-------------------------------------------------------------

-- Writer\_Movie relationships

INSERT INTO Writer\_Movie (WriterID, MovieID)

VALUES ('W000001', 'M000001');

INSERT INTO Writer\_Movie (WriterID, MovieID)

VALUES ('W000001', 'M000002');

INSERT INTO Writer\_Movie (WriterID, MovieID)

VALUES ('W000003', 'M000004');

INSERT INTO Writer\_Movie (WriterID, MovieID)

VALUES ('W000006', 'M000004');

INSERT INTO Writer\_Movie (WriterID, MovieID)

VALUES ('W000007', 'M000006');

INSERT INTO Writer\_Movie (WriterID, MovieID)

VALUES ('W000015', 'M000009');

INSERT INTO Writer\_Movie (WriterID, MovieID)

VALUES ('W000016', 'M000010');

INSERT INTO Writer\_Movie (WriterID, MovieID)

VALUES ('W000018', 'M000011');

-------------------------------------------------------------------

-- Writer\_TV\_Show relationships

INSERT INTO Writer\_TV\_Show (WriterID, TV\_ShowID)

VALUES ('W000002', 'T000001');

INSERT INTO Writer\_TV\_Show (WriterID, TV\_ShowID)

VALUES ('W000004', 'T000004');

INSERT INTO Writer\_TV\_Show (WriterID, TV\_ShowID)

VALUES ('W000008', 'T000006');

INSERT INTO Writer\_TV\_Show (WriterID, TV\_ShowID)

VALUES ('W000009', 'T000007');

INSERT INTO Writer\_TV\_Show (WriterID, TV\_ShowID)

VALUES ('W000010', 'T000008');

INSERT INTO Writer\_TV\_Show (WriterID, TV\_ShowID)

VALUES ('W000012', 'T000009');

INSERT INTO Writer\_TV\_Show (WriterID, TV\_ShowID)

VALUES ('W000013', 'T000010');

INSERT INTO Writer\_TV\_Show (WriterID, TV\_ShowID)

VALUES ('W000014', 'T000011');

------------------------------------------------------------

-- Inserts for Director

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000001', 'Denis', 'Villeneuve', 'Gentilly, Quebec, Canada', TO\_DATE('1967-10-03', 'YYYY-MM-DD'), 72, 'Tanya Lapointe', 'Academy Award Nominee', 'Denis Villeneuve is a French-Canadian film director and writer. He was born in 1967, in Trois-Rivières, Québec, Canada. He started his career as a filmmaker at the National Film Board of Canada. He is best known for his feature films Arrival (2016), Sicario (2015), Prisoners (2013), Enemy (2013), and Incendies (2010). He is married to Tanya Lapointe.');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000002', 'Sam', 'Levinson', 'United States', TO\_DATE('1985-01-08', 'YYYY-MM-DD'), 72, 'Ashley Lent Levinson', 'Primetime Emmy Award', 'Sam Levinson was born on January 8, 1985 in the USA. He is a writer and producer, known for Euphoria (2019), The Idol (2023) and Assassination Nation (2018).');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000003', 'Christopher', 'Nolan', 'London, England', TO\_DATE('1970-07-30', 'YYYY-MM-DD'), 71, 'Emma Thomas', 'Academy Award Nominee', 'Best known for his cerebral, often nonlinear, storytelling, acclaimed Academy Award winner writer/director/producer Sir Christopher Nolan CBE was born in London, England. Over the course of more than 25 years of filmmaking, Nolan has gone from low-budget independent films to working on some of the biggest blockbusters ever made and became one of the most celebrated filmmakers of modern cinema. At 7 years old, Nolan began making short films with his fathers Super-8 camera. While studying English Literature at University College London, he shot 16-millimeter films at U.C.L.s film society, where he learned the guerrilla techniques he would later use to make his first feature, Following (1998), on a budget of around $6,000. The noir thriller was recognized at a number of international film festivals prior to its theatrical release and gained Nolan enough credibility that he was able to gather substantial financing for his next film. ');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000004', 'Kieron', 'Hawkes', 'London, England', TO\_DATE('1980-05-20', 'YYYY-MM-DD'), 70, 'Not Married', 'BAFTA Award', 'Kieron Hawkes is known for Red Eye (2024), Fortitude (2015) and The Burning Girls (2023).');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000005', 'Robert', 'Alexander', 'Los Angeles, CA', TO\_DATE('1980-08-12', 'YYYY-MM-DD'), 70, 'Not Married', 'Peabody Award', 'Robert passed away 11.04.2018. Alexander Robert Graham 05.01.1942 - 11.04.2018 Actor. Late of Glebe. Son of Bob and Jean Alexander (both deceased), brother of Ann, uncle of Sarah and Matt. Adored partner (for 54 years) of Barry Blight; esteemed actor; accomplished musician; kind mentor to emerging young actors; cherished colleague; intelligent, generous, witty and convivial host to a multitude of fortunate friends.Robert passed peacefully away at the Sacred Heart Hospice on Wednesday evening after a lengthy illness.');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000006', 'Steven', 'Spielberg', 'Cincinnati, OH', TO\_DATE('1946-12-18', 'YYYY-MM-DD'), 71, 'Kate Capshaw', 'Academy Award Winner', 'One of the most influential personalities in the history of cinema, Steven Spielberg is Hollywoods best known director and one of the wealthiest filmmakers in the world. He has an extraordinary number of commercially successful and critically acclaimed credits to his name, either as a director, producer or writer since launching the summer blockbuster with Jaws (1975), and he has done more to define popular film-making since the mid-1970s than anyone else. Steven Allan Spielberg was born in 1946 in Cincinnati, Ohio, to Leah Frances (Posner), a concert pianist and restaurateur, and Arnold Spielberg, an electrical engineer who worked in computer development. His parents were both born to Russian Jewish immigrant families. Steven spent his younger years in Haddon Township, New Jersey, Phoenix, Arizona, and later Saratoga, California. He went to California State University Long Beach, but dropped out to pursue his entertainment career. Among his early directing efforts were Battle Squad (1961), which combined World War II footage with footage of an airplane on the ground that he makes you believe is moving. ');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000007', 'James', 'Burrows', 'Los Angeles, CA', TO\_DATE('1940-12-30', 'YYYY-MM-DD'), 70, 'Debbie Burrows', 'Primetime Emmy Award', 'James Burrows was born on December 30, 1940 in Los Angeles, California, USA. He is a director and producer, known for Cheers (1982), Will & Grace (1998) and Taxi (1978). He has been married to Debbie Easton since 1997. He was previously married to Linda Solomon.');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000008', 'Ben', 'Affleck', 'Berkeley, CA', TO\_DATE('1972-08-15', 'YYYY-MM-DD'), 72, 'Jennifer Lopez', 'Academy Award Winner', 'Benjamin Géza "Ben" Affleck-Boldt was born on August 15, 1972 in Berkeley, California and raised in Cambridge, Massachusetts, to mother Chris Anne (Boldt), a school teacher, and father Timothy Byers "Tim" Affleck, a social worker. Ben has a younger brother, actor Casey Affleck, who was born in 1975. He is of mostly English, Irish, German, and Scottish ancestry. His middle name, Géza, is after a Hungarian family friend who was a Holocaust survivor. Affleck wanted to be an actor ever since he could remember, and his first acting experience was for a Burger King commercial, when he was on the PBS mini-series, The Voyage of the Mimi (1984). It was also at that age when Ben met his lifelong friend and fellow actor, Matt Damon. They played little league together and took drama classes together. Bens teen years consisted of mainly TV movies and small television appearances including Hands of a Stranger (1987) and The Second Voyage of the Mimi (1988). He made his big introduction into feature films in 1993 when he was cast in Dazed and Confused (1993). After that, he did mostly independent films like Kevin Smiths Mallrats (1995) and Chasing Amy (1997) which were great for Bens career, receiving renowned appreciation for his works at the Sundance film festival.');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000009', 'Joel', 'Bach', 'Los Angeles, CA', TO\_DATE('1975-06-15', 'YYYY-MM-DD'), 71, 'Not Married', 'Peabody Award', 'Joel Bach is known for Years of Living Dangerously (2014), 60 Minutes (1968) and X Marks the Spot (2020).');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000011', 'Ron', 'Howard', 'Duncan, OK', TO\_DATE('1954-03-01', 'YYYY-MM-DD'), 70, 'Cheryl Howard', 'Academy Award Winner', 'Academy Award-winning filmmaker Ron Howard is one of this generations most popular directors. From the critically acclaimed dramas A Beautiful Mind (2001) and Apollo 13 (1995) to the hit comedies Parenthood (1989) and Splash (1984), he has created some of Hollywoods most memorable films. Howard made his directorial debut in 1978 with the comedy Grand Theft Auto (1977). He began his career in film as an actor. He first appeared in The Journey (1959) and The Music Man (1962), then as Opie on the long-running television series The Andy Griffith Show (1960). Howard later starred in the popular series Happy Days (1974) and drew favorable reviews for his performances in American Graffiti (1973) and The Shootist (1976). Howard and long-time producing partner Brian Grazer first collaborated on the hit comedies "Night Shift" and "Splash." The pair co-founded Imagine Entertainment in 1986 to create independently produced feature films. Howards portfolio includes some of the most popular films of the past 20 years. In 1991, Howard created the acclaimed drama "Backdraft", starring Robert De Niro, Kurt Russell and William Baldwin. He followed it with the historical epic Far and Away (1992), starring Tom Cruise and Nicole Kidman. Howard directed Mel Gibson, Rene Russo, Gary Sinise and Delroy Lindo in the 1996 suspense thriller Ransom (1996). ');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000012', 'Otto', 'Bathurst', 'London, England', TO\_DATE('1971-06-18', 'YYYY-MM-DD'), 71, 'Not Married', 'BAFTA Award', 'Otto Bathurst was born in June 1971 in Hammersmith, London, England, UK. He is a director and producer, known for Robin Hood (2018), Peaky Blinders (2013) and Black Mirror (2011).');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000013', 'Jane', 'Root', 'London, England', TO\_DATE('1960-02-12', 'YYYY-MM-DD'), 69, 'Not Married', 'Peabody Award', 'Jane Root is a leading creative executive with a reputation for innovation, determination and delivery. Her work was celebrated by Women in Film & TV when she was awarded the prestigious 2017 Business Award acknowledging her outstanding achievements in the industry both in the UK and abroad. In 2008, Jane created Nutopia, a TV production company based in London and Washington committed to thinking big and focused on ambitious factual series of all types, as well as factually-inspired drama. Nutopia is dedicated to creating ground-breaking, specialist factual content on a global scale. Best known for pioneering a new genre of television - the megadoc - by combining epic cinematography and action-driven drama with high-end documentary and A-list talent. Recent credits include highly praised series The World According to Jeff Goldblum (Disney+), award-winning earth science series One Strange Rock (Nat Geo), critically acclaimed drama doc The Last Czars (Netflix), and Grierson-nominated feature doc The Hunt for Jihadi John (HBO/Channel 4). The company continues to development shows for the biggest broadcasters in the world, alongside some of the highest profile partners in the industry. Jane was President of Discovery Channel US until 2007 where she re-positioned the channel giving it a fresh voice and leading to record ratings. During her time at Discovery, Jane commissioned international hits including Man vs Wild (Born Survivor), Deadliest Catch and Dirty Jobs, as well as turning Planet Earth into a worldwide phenomenon. Before her US adventure, Jane was Controller of BBC2 for 5 years.');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000014', 'Huw', 'Cordey', 'Cardiff, Wales', TO\_DATE('1970-03-15', 'YYYY-MM-DD'), 70, 'Not Married', 'Peabody Award', 'Huw Cordey is known for Dancing with the Birds (2019), Our Planet (2019) and Secret Lives of Orangutans (2024).');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000015', 'Jon', 'Watts', 'Fountain, CO', TO\_DATE('1981-06-28', 'YYYY-MM-DD'), 71, 'Not Married', 'Academy Award Nominee', 'Jon Watts is an American filmmaker and screenwriter. He directed Cop Car and Clown before he was picked by Marvel and Sony to direct Spider-Man: Homecoming starring Tom Holland and Zendaya. Its success resulted in two sequels, Far from Home in 2019 and No Way Home in 2021. He was also picked by Marvel to direct a Fantastic Four reboot film following the failure of Josh Tranks Fant4stic, but dropped the directing role in April 2022.');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000016', 'Anthony', 'Russo', 'Cleveland, OH', TO\_DATE('1970-02-03', 'YYYY-MM-DD'), 70, 'Not Married', 'Academy Award Nominee', 'Anthony J. Russo is an American filmmaker and producer who works alongside his brother Joseph Russo. They have directed You, Me and Dupree, Cherry and the Marvel films Captain America: The Winter Soldier, Captain America: Civil War, Avengers: Infinity War and Avengers: Endgame. Endgame is one of the highest grossing films of all time.');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000017', 'Joe', 'Russo', 'Cleveland, OH', TO\_DATE('1971-07-18', 'YYYY-MM-DD'), 70, 'Not Married', 'Academy Award Nominee', 'Joseph Vincent Russo is an American filmmaker and producer who works alongside his brother Anthony Russo. They have directed You, Me and Dupree, Cherry and the Marvel films Captain America: The Winter Soldier, Captain America: Civil War, Avengers: Infinity War and Avengers: Endgame. Endgame is one of the highest grossing films of all time.');

INSERT INTO Director (URLID, FirstName, LastName, PlaceOfBirth, DateOfBirth, Height, Spouse, Award, Biography)

VALUES ('D000018', 'Ruben', 'Fleischer', 'Washington, D.C.', TO\_DATE('1974-10-31', 'YYYY-MM-DD'), 72, 'Not Married', 'Peabody Award', 'Ruben Fleischer was born on October 31, 1974 in Washington, District of Columbia, USA. He is a producer and director, known for Zombieland (2009), Gangster Squad (2013) and Venom (2018). He has been married to Holly Shakoor Fleischer since November 10, 2012.');

------------------------------------------------------------

-- Inserts for Director Parents

-- Denis Villeneuve

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000001', 'No Information Available');

-- Sam Levinson

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000002', 'Barry Levinson');

-- Christopher Nolan

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000003', 'Brendan James Nolan');

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000003', 'Christina Nolan');

-- Kieron Hawkes

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000004', 'No Information Available');

-- Robert Alexander

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000005', 'No Information Available');

-- Steven Spielberg

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000006', 'Arnold Spielberg');

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000006', 'Leah Adler');

-- James Burrows

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000007', 'Abe Burrows');

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000007', 'Ruth Levinson');

-- Ben Affleck

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000008', 'Timothy Byers Affleck');

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000008', 'Christine Anne Boldt');

-- Joel Bach

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000009', 'No Information Available');

-- Ron Howard

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000011', 'Rance Howard');

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000011', 'Jean Speegle Howard');

-- Otto Bathurst

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000012', 'No Information Available');

-- Jane Root

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000013', 'No Information Available');

-- Huw Cordey

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000014', 'No Information Available');

-- Jon Watts

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000015', 'No Information Available');

-- Anthony Russo

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000016', 'No Information Available');

-- Joe Russo

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000017', 'No Information Available');

-- Ruben Fleischer

INSERT INTO Director\_Parents (DirectorID, Parent)

VALUES ('D000018', 'No Information Available');

------------------------------------------------------------

-- Updates for Director Media

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000001', 'Denis Villeneuve Interview on Dune');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000001', 'Denis Villeneuve Feature on Dune Part 2');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000002', 'Sam Levinson Interview on Euphoria');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000003', 'Christopher Nolan Interview on Interstellar');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000003', 'Christopher Nolan Interview on Oppenheimer');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000004', 'Kieron Hawkes Feature on Intergalactic');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000005', 'Robert Alexander Documentary on A Man Named Scott');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000006', 'Steven Spielberg Feature on Saving Private Ryan');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000007', 'James Burrows Documentary on Will & Grace');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000008', 'Ben Affleck Feature on Project Green Light');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000009', 'Joel Bach Documentary on Years of Living Dangerously');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000011', 'Ron Howard Feature on Heart of the Sea');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000012', 'Otto Bathurst Documentary on Peaky Blinders');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000013', 'Jane Root Feature on A World of Calm');

INSERT INTO Director\_Media (DirectorID, Media)

VALUES ('D000014', 'Huw Cordey Documentary on Atlantic: The Wildest Ocean on Earth');

------------------------------------------------------------

-- Inserts for Director Children

-- Denis Villeneuve

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000001', 'No Information Available');

-- Sam Levinson

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000002', 'No Information Available');

-- Christopher Nolan

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000003', 'Rory Nolan');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000003', 'Magnus Nolan');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000003', 'Oliver Nolan');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000003', 'Flora Nolan');

-- Kieron Hawkes

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000004', 'No Information Available');

-- Robert Alexander

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000005', 'No Information Available');

-- Steven Spielberg

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000006', 'Max Spielberg');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000006', 'Sasha Spielberg');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000006', 'Theo Spielberg');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000006', 'Mikaela Spielberg');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000006', 'Destry Spielberg');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000006', 'Sawyer Spielberg');

-- James Burrows

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000007', 'No Information Available');

-- Ben Affleck

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000008', 'Violet Affleck');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000008', 'Seraphina Affleck');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000008', 'Samuel Affleck');

-- Joel Bach

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000009', 'No Information Available');

-- Ron Howard

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000011', 'Bryce Dallas Howard');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000011', 'Paige Howard');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000011', 'Jocelyn Howard');

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000011', 'Reed Howard');

-- Otto Bathurst

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000012', 'No Information Available');

-- Jane Root

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000013', 'No Information Available');

-- Huw Cordey

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000014', 'No Information Available');

-- Jon Watts

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000015', 'No Information Available');

-- Anthony Russo

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000016', 'Julian Russo');

-- Joe Russo

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000017', 'Sophia Russo');

-- Ruben Fleischer

INSERT INTO Director\_Children (DirectorID, Child)

VALUES ('D000018', 'No Information Available');

------------------------------------------------------------

-- Updates for Director\_Movie relationships

INSERT INTO Director\_Movie (DirectorID, MovieID)

VALUES ('D000001', 'M000001');

INSERT INTO Director\_Movie (DirectorID, MovieID)

VALUES ('D000001', 'M000002');

INSERT INTO Director\_Movie (DirectorID, MovieID)

VALUES ('D000003', 'M000004');

INSERT INTO Director\_Movie (DirectorID, MovieID)

VALUES ('D000003', 'M000005');

INSERT INTO Director\_Movie (DirectorID, MovieID)

VALUES ('D000006', 'M000006');

INSERT INTO Director\_Movie (DirectorID, MovieID)

VALUES ('D000003', 'M000007');

INSERT INTO Director\_Movie (DirectorID, MovieID)

VALUES ('D000011', 'M000008');

INSERT INTO Director\_Movie (DirectorID, MovieID)

VALUES ('D000015', 'M000009');

INSERT INTO Director\_Movie (DirectorID, MovieID)

VALUES ('D000016', 'M000010');

INSERT INTO Director\_Movie (DirectorID, MovieID)

VALUES ('D000017', 'M000010');

INSERT INTO Director\_Movie (DirectorID, MovieID)

VALUES ('D000018', 'M000011');

------------------------------------------------------------

-- Director\_TV\_Show relationships (Block 1)

INSERT INTO Director\_TV\_Show (DirectorID, TV\_ShowID)

VALUES ('D000002', 'T000001');

INSERT INTO Director\_TV\_Show (DirectorID, TV\_ShowID)

VALUES ('D000004', 'T000004');

INSERT INTO Director\_TV\_Show (DirectorID, TV\_ShowID)

VALUES ('D000005', 'T000005');

INSERT INTO Director\_TV\_Show (DirectorID, TV\_ShowID)

VALUES ('D000007', 'T000006');

INSERT INTO Director\_TV\_Show (DirectorID, TV\_ShowID)

VALUES ('D000008', 'T000007');

INSERT INTO Director\_TV\_Show (DirectorID, TV\_ShowID)

VALUES ('D000009', 'T000008');

INSERT INTO Director\_TV\_Show (DirectorID, TV\_ShowID)

VALUES ('D000012', 'T000009');

INSERT INTO Director\_TV\_Show (DirectorID, TV\_ShowID)

VALUES ('D000013', 'T000010');

INSERT INTO Director\_TV\_Show (DirectorID, TV\_ShowID)

VALUES ('D000014', 'T000011');

------------------------------------------------------------

-- Inserts for Movie Genre

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000001', 'Sci-Fi');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000001', 'Adventure');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000002', 'Sci-Fi');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000002', 'Adventure');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000003', 'Drama');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000004', 'Sci-Fi');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000004', 'Drama');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000005', 'Biography');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000005', 'Drama');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000006', 'War');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000006', 'Drama');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000009', 'Action');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000009', 'Adventure');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000010', 'Action');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000010', 'Superhero');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000011', 'Adventure');

INSERT INTO Movie\_Genre (MovieID, Genre)

VALUES ('M000011', 'Action');

------------------------------------------------------------

-- Inserts for Movie Streaming

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000001', 'HBO Max');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000001', 'Amazon Prime');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000002', 'HBO Max');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000002', 'Netflix');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000003', 'Amazon Prime');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000003', 'Hulu');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000004', 'Netflix');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000004', 'Disney+');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000005', 'Peacock');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000005', 'Apple TV+');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000006', 'Paramount+');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000006', 'HBO Max');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000009', 'Disney+');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000009', 'Amazon Prime');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000010', 'Disney+');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000010', 'HBO Max');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000011', 'Netflix');

INSERT INTO Movie\_Streaming (MovieID, Streaming)

VALUES ('M000011', 'Amazon Prime');

------------------------------------------------------------

-- Inserts for Movie Review

-- Dune

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000001', 'Visually Stunning', 'DesertWanderer15799', 'Dune is a breathtaking visual experience with a compelling story.', 9);

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000001', 'Epic Sci-Fi', 'SpiceLover154', 'A must-watch for sci-fi fans. Denis Villeneuve delivers a masterpiece.', 10);

-- Dune: Part 2

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000002', 'A Grand Sequel', 'ArrakisFan652', 'Dune: Part 2 is a thrilling continuation with stunning visuals.', 8);

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000002', 'Amazing World-Building', 'SandwormRider135', 'The world-building in this sequel is unmatched. A true epic.', 9);

-- Challengers

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000003', 'Unique and Emotional', 'TennisAce12', 'Challengers is a heartfelt drama with a unique perspective.', 7);

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000003', 'Great Performances', 'LoveGameFan468', 'The performances in this movie are top-notch. A great watch.', 8);

-- Interstellar

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000004', 'Mind-Blowing', 'CosmicTraveler123', 'Interstellar is a cinematic masterpiece that will leave you in awe.', 10);

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000004', 'Emotional and Thought-Provoking', 'TimeWarpFan231457', 'A deeply emotional and thought-provoking journey through space.', 9);

-- Oppenheimer

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000005', 'Powerful and Intense', 'AtomicWatcher2721', 'Oppenheimer is a gripping portrayal of history and science.', 9);

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000005', 'A Masterpiece', 'ManhattanProjectFan1268', 'Christopher Nolan delivers a masterpiece with Oppenheimer.', 10);

-- Saving Private Ryan

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000006', 'Heart-Wrenching', 'WW2HistoryBuff3517', 'A powerful and emotional depiction of the sacrifices of war.', 10);

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000006', 'A War Classic', 'SoldierFan231', 'Saving Private Ryan is a timeless war movie that everyone should watch.', 9);

-- Spider-Man: Homecoming

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000009', 'Fun and Entertaining', 'WebHead872', 'Spider-Man: Homecoming is a fresh and fun take on the superhero.', 9);

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000009', 'Great Action', 'FriendlyNeighborhoodFan213', 'The action and humor in this movie are perfectly balanced.', 8);

-- Captain America: Civil War

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000010', 'Epic Showdown', 'ShieldBearer547', 'Civil War is a thrilling and emotional clash between heroes.', 8);

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000010', 'A Marvel Classic', 'TeamCapFan235', 'A must-watch for Marvel fans. The story and action are amazing.', 9);

-- Uncharted

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000011', 'Exciting Adventure', 'TreasureHunter312', 'Uncharted is a fun and action-packed treasure-hunting movie.', 7);

INSERT INTO Movie\_Review (MovieID, Title, Reviewer, ReviewContents, Rating)

VALUES ('M000011', 'Great Adaptation', 'ExplorerFan767', 'A solid adaptation of the video game series. Very enjoyable.', 8);

------------------------------------------------------------

-- Inserts for TV Show Genre

INSERT INTO TV\_Show\_Genre (TV\_ShowID, Genre)

VALUES ('T000001', 'Drama');

INSERT INTO TV\_Show\_Genre (TV\_ShowID, Genre)

VALUES ('T000001', 'Teen');

INSERT INTO TV\_Show\_Genre (TV\_ShowID, Genre)

VALUES ('T000002', 'Comedy');

INSERT INTO TV\_Show\_Genre (TV\_ShowID, Genre)

VALUES ('T000002', 'Family');

INSERT INTO TV\_Show\_Genre (TV\_ShowID, Genre)

VALUES ('T000003', 'Comedy');

INSERT INTO TV\_Show\_Genre (TV\_ShowID, Genre)

VALUES ('T000003', 'Action');

INSERT INTO TV\_Show\_Genre (TV\_ShowID, Genre)

VALUES ('T000004', 'Sci-Fi');

INSERT INTO TV\_Show\_Genre (TV\_ShowID, Genre)

VALUES ('T000004', 'Adventure');

INSERT INTO TV\_Show\_Genre (TV\_ShowID, Genre)

VALUES ('T000005', 'Documentary');

INSERT INTO TV\_Show\_Genre (TV\_ShowID, Genre)

VALUES ('T000006', 'Comedy');

INSERT INTO TV\_Show\_Genre (TV\_ShowID, Genre)

VALUES ('T000007', 'Reality');

INSERT INTO TV\_Show\_Genre (TV\_ShowID, Genre)

VALUES ('T000008', 'Documentary');

------------------------------------------------------------

-- Inserts for TV Show Streaming

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000001', 'HBO Max');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000001', 'Amazon Prime');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000002', 'Disney+');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000002', 'Hulu');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000003', 'Disney+');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000003', 'Netflix');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000004', 'Sky');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000004', 'HBO Max');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000005', 'Amazon Prime');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000005', 'Apple TV+');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000006', 'Peacock');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000006', 'Hulu');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000007', 'HBO Max');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000007', 'Netflix');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000008', 'National Geographic');

INSERT INTO TV\_Show\_Streaming (TV\_ShowID, Streaming)

VALUES ('T000008', 'Disney+');

------------------------------------------------------------

-- Inserts for TV Show Review

-- Euphoria

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000001', 'Raw and Real', 'TeenLifeWatcher', 'Euphoria is a bold and unflinching look at teenage struggles.', 6);

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000001', 'Incredible Acting', 'DramaQueen', 'The acting in Euphoria is phenomenal. A must-watch.', 10);

-- Shake It Up

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000002', 'Fun for All Ages', 'DanceLover', 'Shake It Up is a charming and energetic show for the whole family.', 7);

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000002', 'Great Chemistry', 'BestiesFan', 'The chemistry between the leads makes this show a joy to watch.', 9);

-- KC Undercover

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000003', 'Action and Comedy', 'SpyFanatic', 'KC Undercover is a delightful mix of humor and spy action.', 7);

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000003', 'Family Fun', 'UndercoverWatcher', 'A fun and exciting show for the whole family to enjoy.', 6);

-- Intergalactic

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000004', 'Visually Stunning', 'SpaceExplorer', 'Intergalactic is a visually stunning and action-packed series.', 7);

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000004', 'Great Sci-Fi', 'GalacticFan', 'A thrilling space adventure with a strong cast and story.', 9);

-- A Man Named Scott

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000005', 'Inspiring and Creative', 'MusicLover', 'A Man Named Scott is a fascinating exploration of artistry.', 6);

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000005', 'A Unique Perspective', 'DocuFan', 'A unique and inspiring documentary about creativity.', 8);

-- Will & Grace

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000006', 'Hilarious and Heartwarming', 'SitcomLover', 'Will & Grace is a timeless sitcom that never gets old.', 10);

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000006', 'A Classic', 'ComedyFan', 'A hilarious and heartwarming show with great characters.', 7);

-- Project Green Light

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000007', 'Inspiring for Filmmakers', 'FilmFanatic', 'A great behind-the-scenes look at the filmmaking process.', 8);

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000007', 'Unique and Educational', 'IndieFilmLover', 'Project Green Light is a unique and educational reality series.', 7);

-- Years of Living Dangerously

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000008', 'Eye-Opening', 'ClimateWatcher', 'A powerful documentary series about climate change.', 9);

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000008', 'Informative and Engaging', 'EcoFanatic', 'A must-watch for anyone interested in the environment.', 8);

-- Peaky Blinders

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000009', 'Masterpiece', 'HistoricalFan', 'Peaky Blinders is a gripping and stylish historical drama.', 10);

INSERT INTO TV\_Show\_Review (TV\_ShowID, Title, Reviewer, ReviewContents, Rating)

VALUES ('T000009', 'Brilliant Acting', 'DramaLover', 'The acting and storytelling in Peaky Blinders are phenomenal.', 9);

------------------------------------------------------------

-- Inserts for TV Show Episode

-- Euphoria Episodes

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000001', 'Pilot', 1, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000001', 'Stuntin'' Like My Daddy', 2, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000001', 'Made You Look', 3, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000001', 'Shook Ones Pt. II', 4, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000001', '03 Bonnie and Clyde', 5, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000001', 'The Next Episode', 6, 1);

-- Shake It Up Episodes

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000002', 'Start It Up', 1, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000002', 'Meatball It Up', 2, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000002', 'Give It Up', 3, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000002', 'Add It Up', 4, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000002', 'Kick It Up', 5, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000002', 'Party It Up', 6, 1);

-- KC Undercover Episodes

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000003', 'Pilot', 1, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000003', 'My Sister from Another Mother... Board', 2, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000003', 'Give Me a K! Give Me a C!', 3, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000003', 'Off the Grid', 4, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000003', 'Photo Bombed', 5, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000003', 'How KC Got Her Swag Back', 6, 1);

-- Intergalactic Episodes

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000004', 'It Begins', 1, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000004', 'Prison Break', 2, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000004', 'The Chase', 3, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000004', 'The Truth', 4, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000004', 'Betrayal', 5, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000004', 'Final Stand', 6, 1);

-- A Man Named Scott Episodes

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000005', 'A Man Named Scott', 1, 1);

-- Will & Grace Episodes

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000006', 'Pilot', 1, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000006', 'A New Lease on Life', 2, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000006', 'Head Case', 3, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000006', 'Between a Rock and Harlin''s Place', 4, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000006', 'Boo! Humbug', 5, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000006', 'William, Tell', 6, 1);

-- Project Greenlight Episodes

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000007', 'The Pitch', 1, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000007', 'The Decision', 2, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000007', 'Pre-Production', 3, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000007', 'The Shoot', 4, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000007', 'Post-Production', 5, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000007', 'The Premiere', 6, 1);

-- Years of Living Dangerously Episodes

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000008', 'The Surge', 1, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000008', 'End of the Woods', 2, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000008', 'The Collapse', 3, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000008', 'The Gathering Storm', 4, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000008', 'The Race Against Time', 5, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000008', 'The Reckoning', 6, 1);

-- World of Calm Episodes

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000010', 'The Stillness Within', 1, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000010', 'The Beauty of Nature', 2, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000010', 'The Flow of Water', 3, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000010', 'The Calm of the Forest', 4, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000010', 'The Serenity of the Sky', 5, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000010', 'The Rhythm of Life', 6, 1);

-- Atlantic Episodes

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000011', 'The Wild Atlantic', 1, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000011', 'The Deep Abyss', 2, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000011', 'The Tropical Reefs', 3, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

VALUES ('T000011', 'The Frozen North', 4, 1);

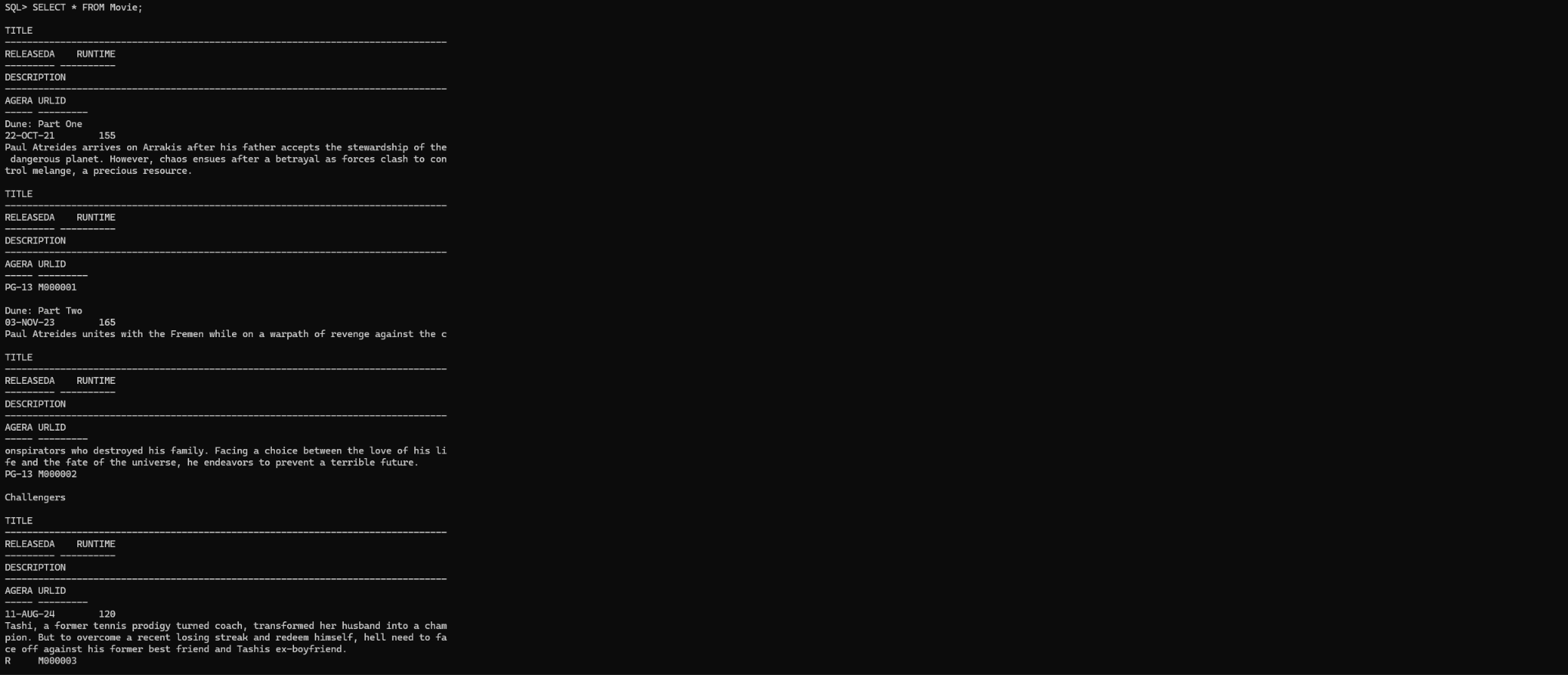
INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeNumber, SeasonNumber)

VALUES ('T000011', 'The Migratory Pathways', 5, 1);

INSERT INTO TV\_Show\_Episode (TV\_ShowID, EpisodeTitle, EpisodeNumber, SeasonNumber)

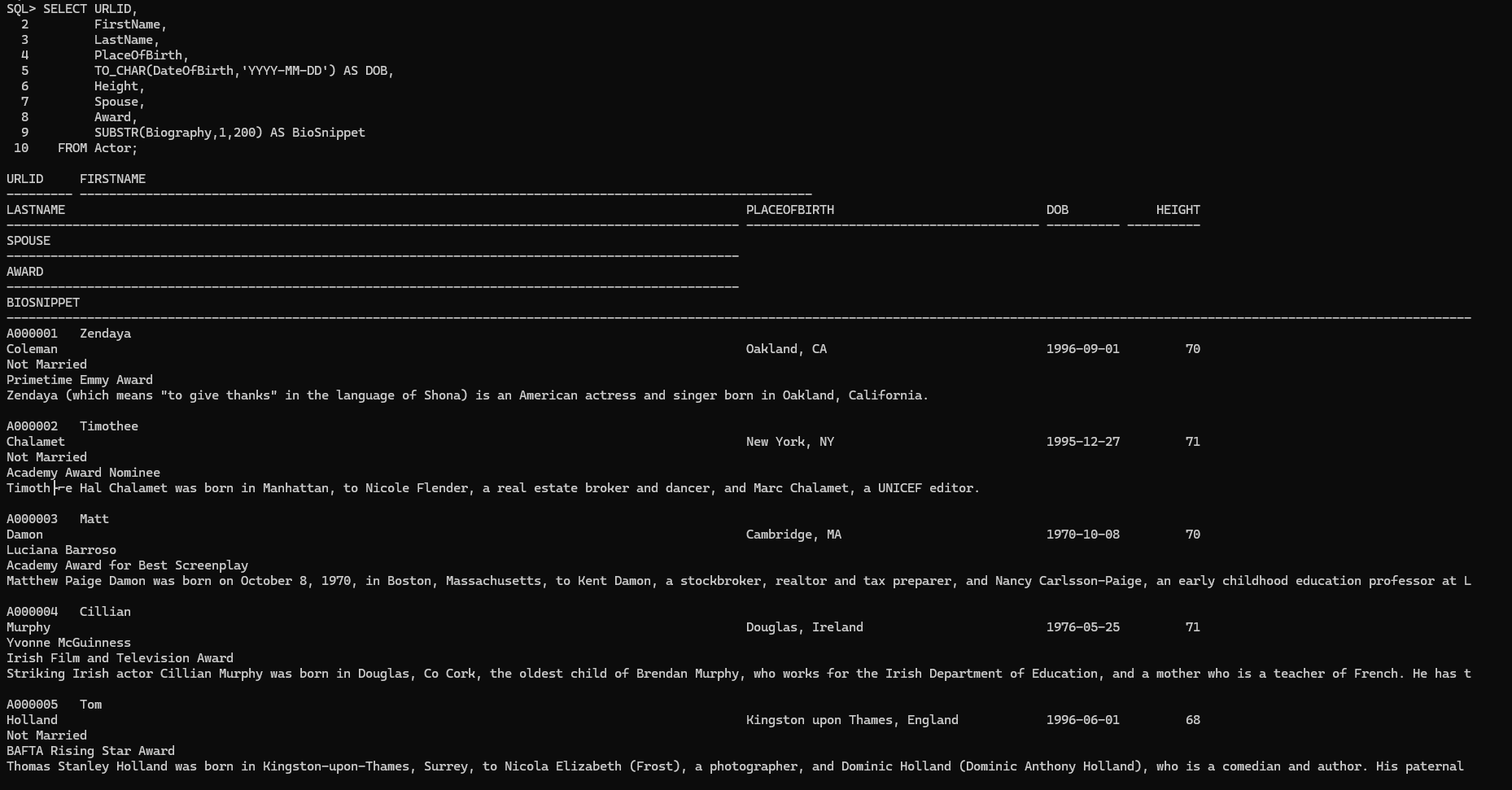
VALUES ('T000011', 'The Changing Ocean', 6, 1);

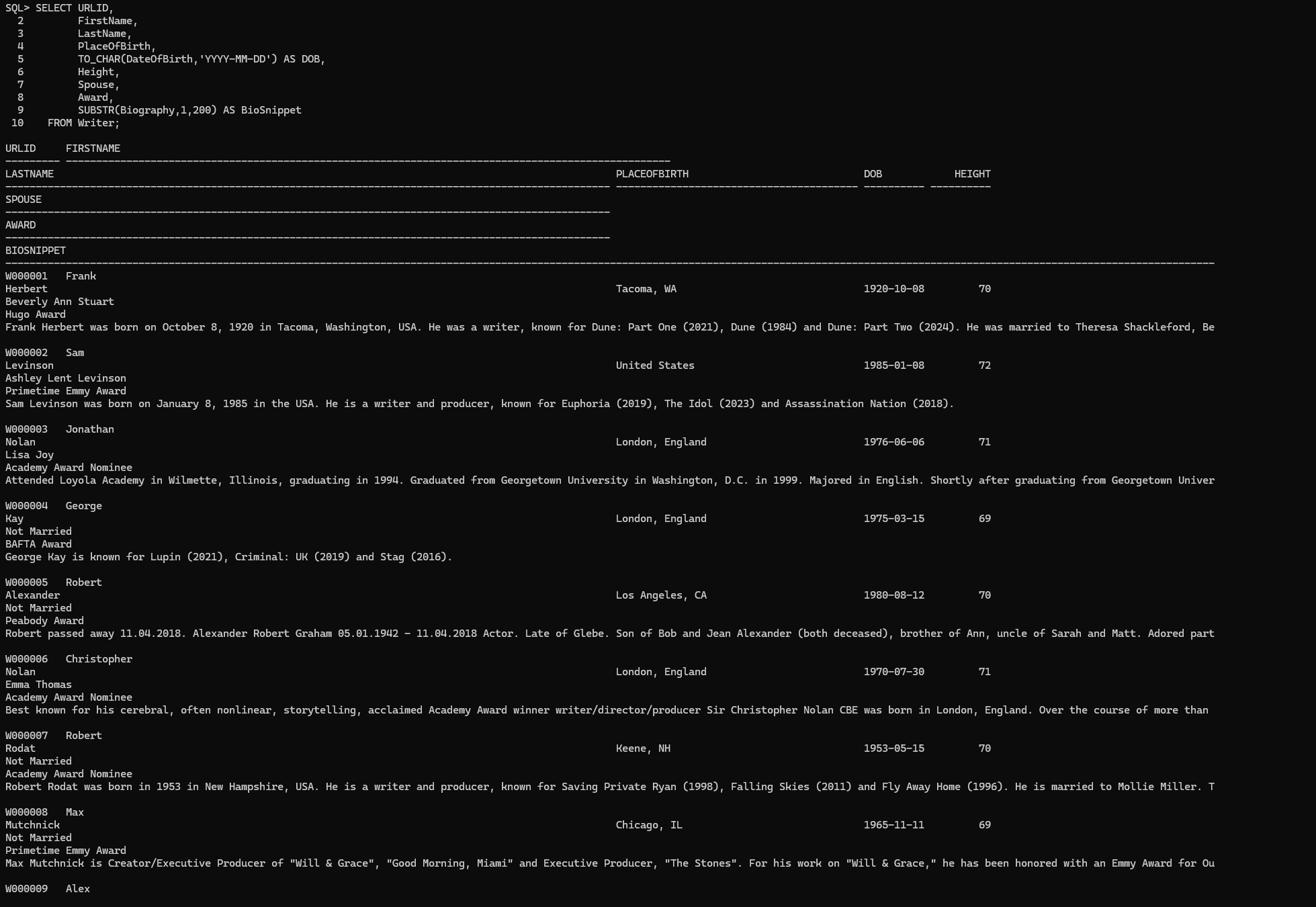
**SELECT \* FROM MOVIE**

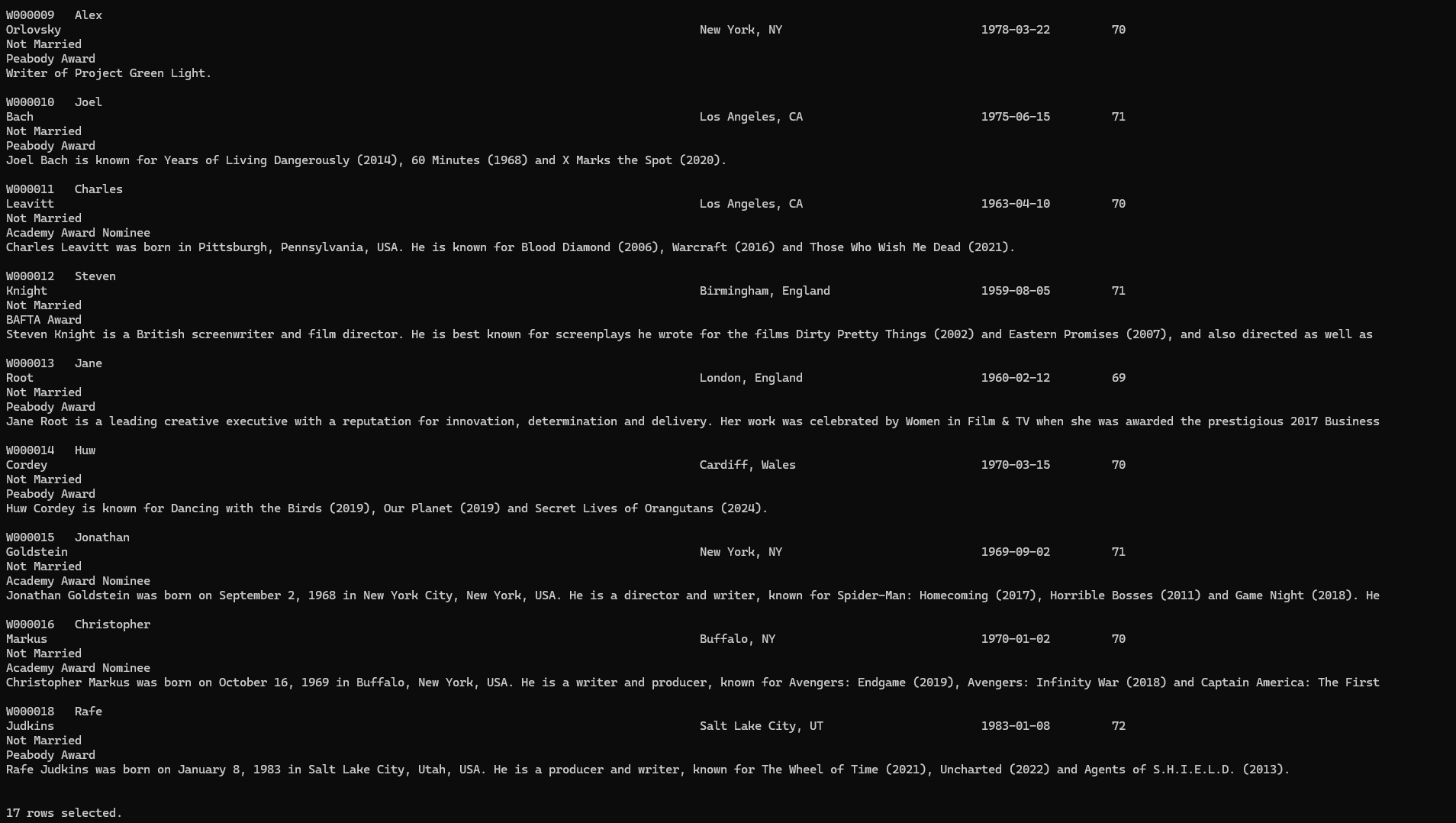
****

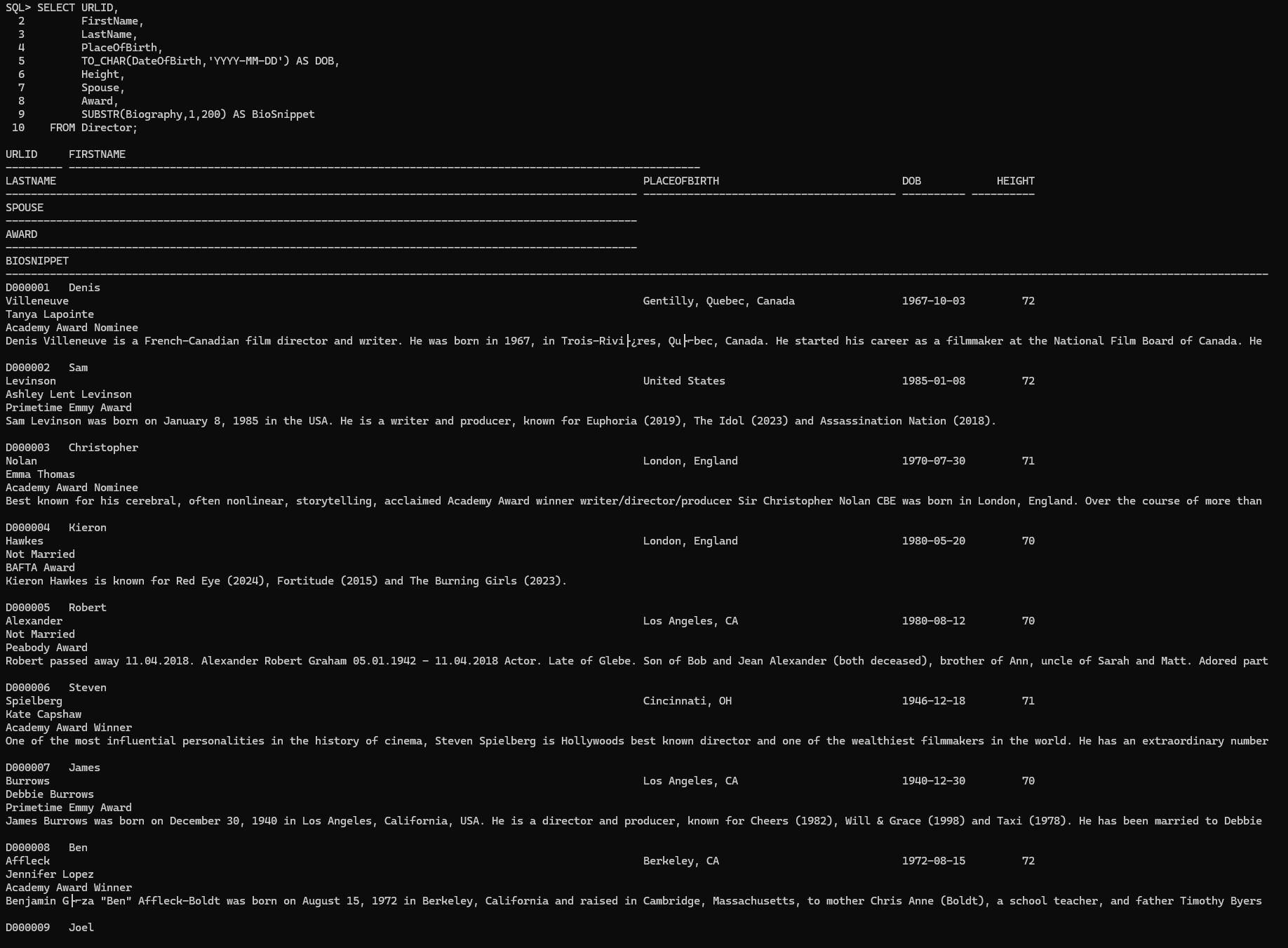
****

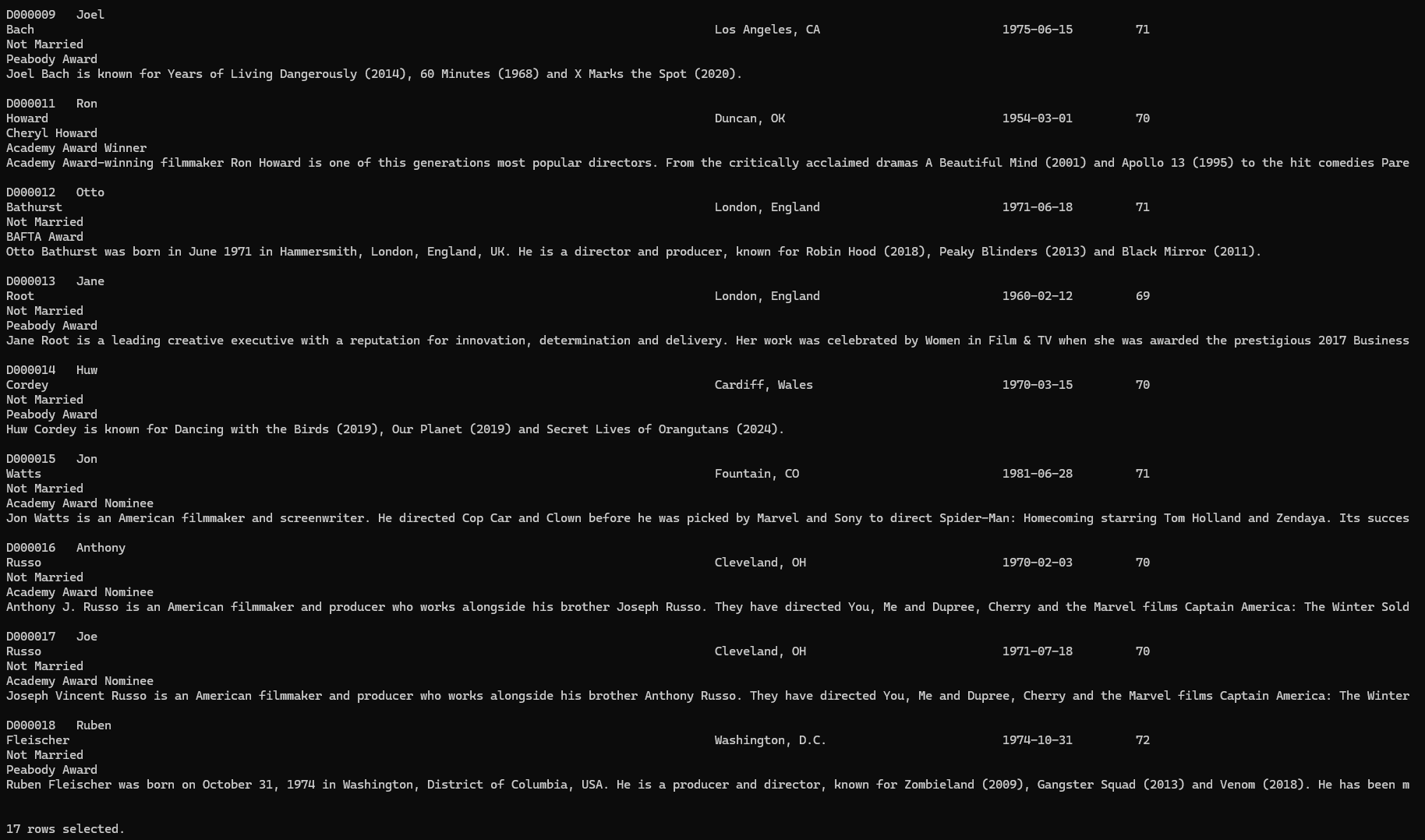
****

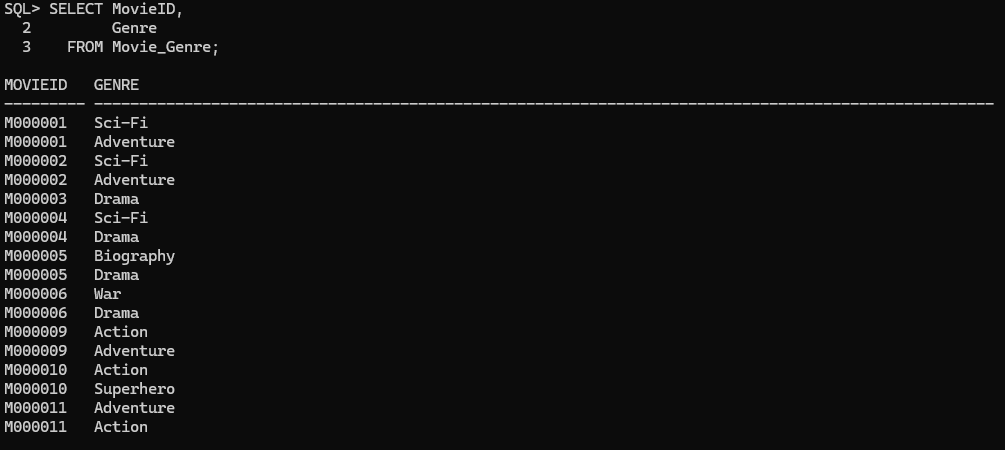
**SELECT \* FROM ACTOR:**

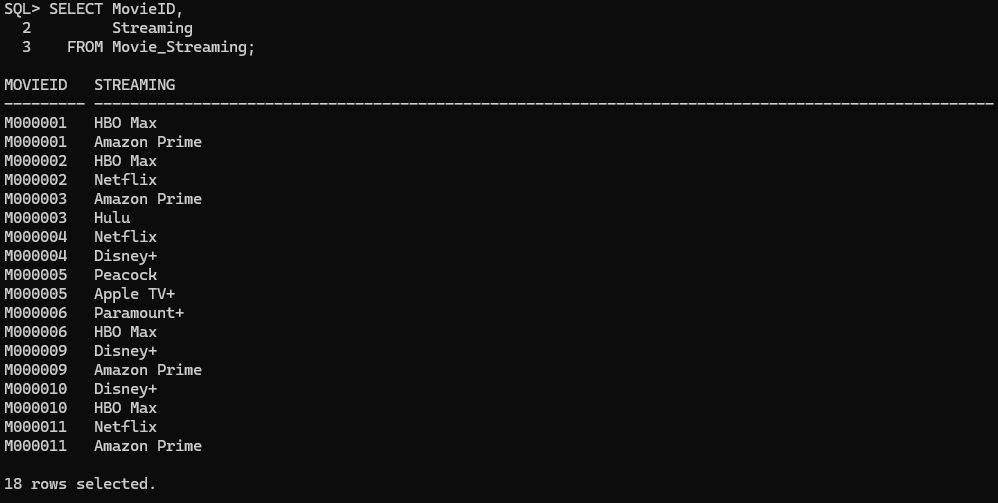
**SELECT \* FROM WRITER:**

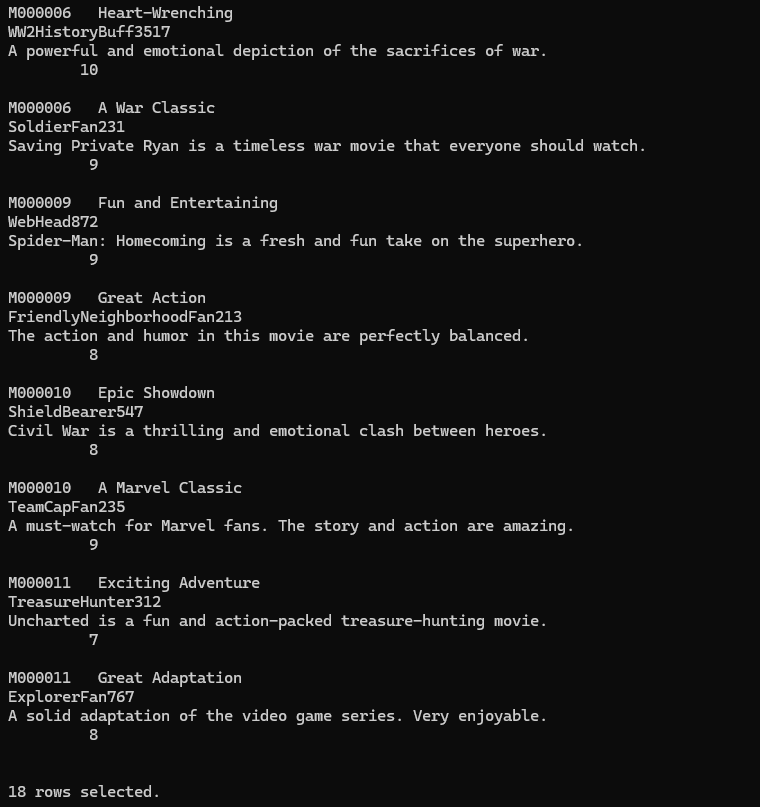
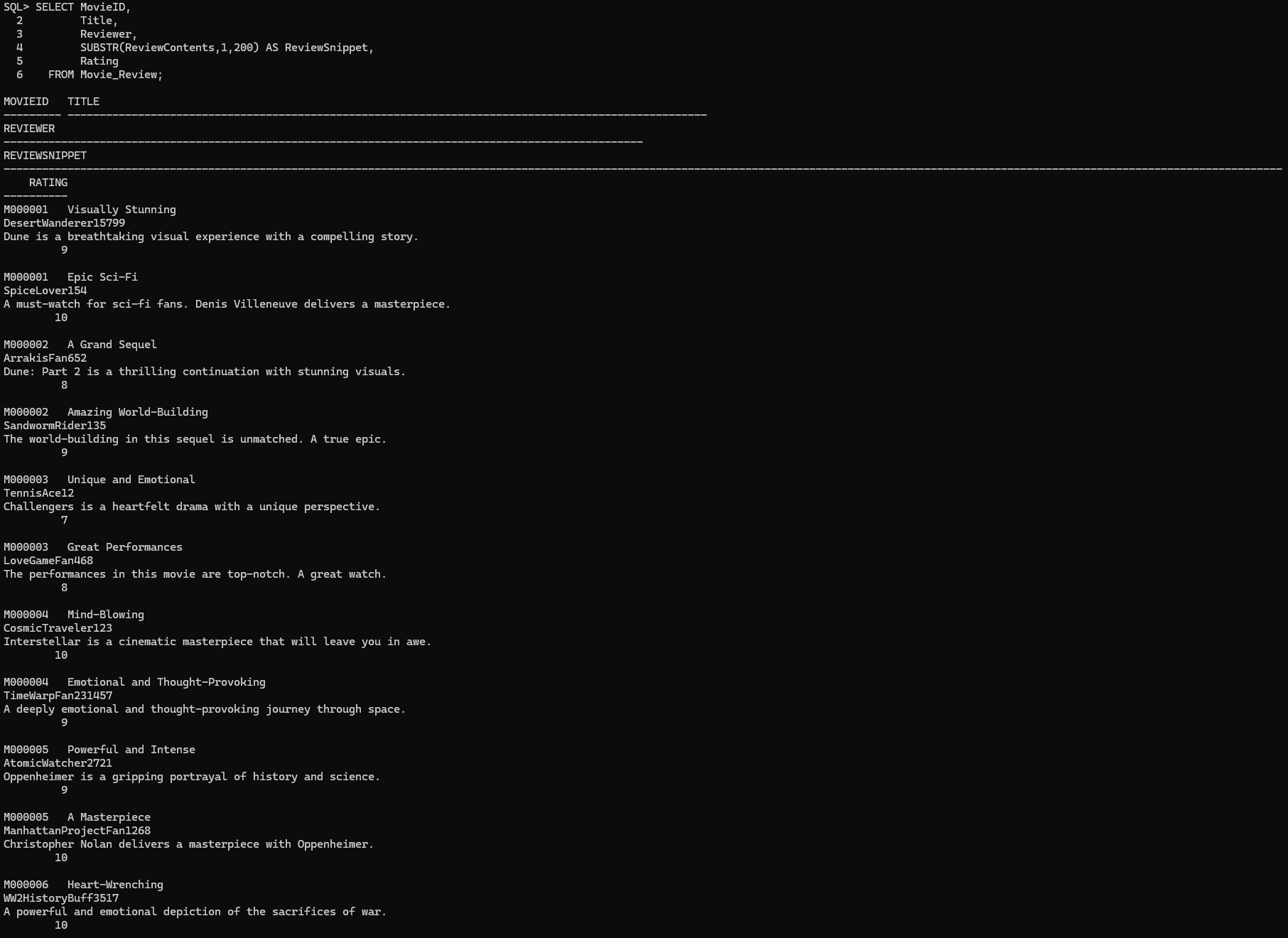
****

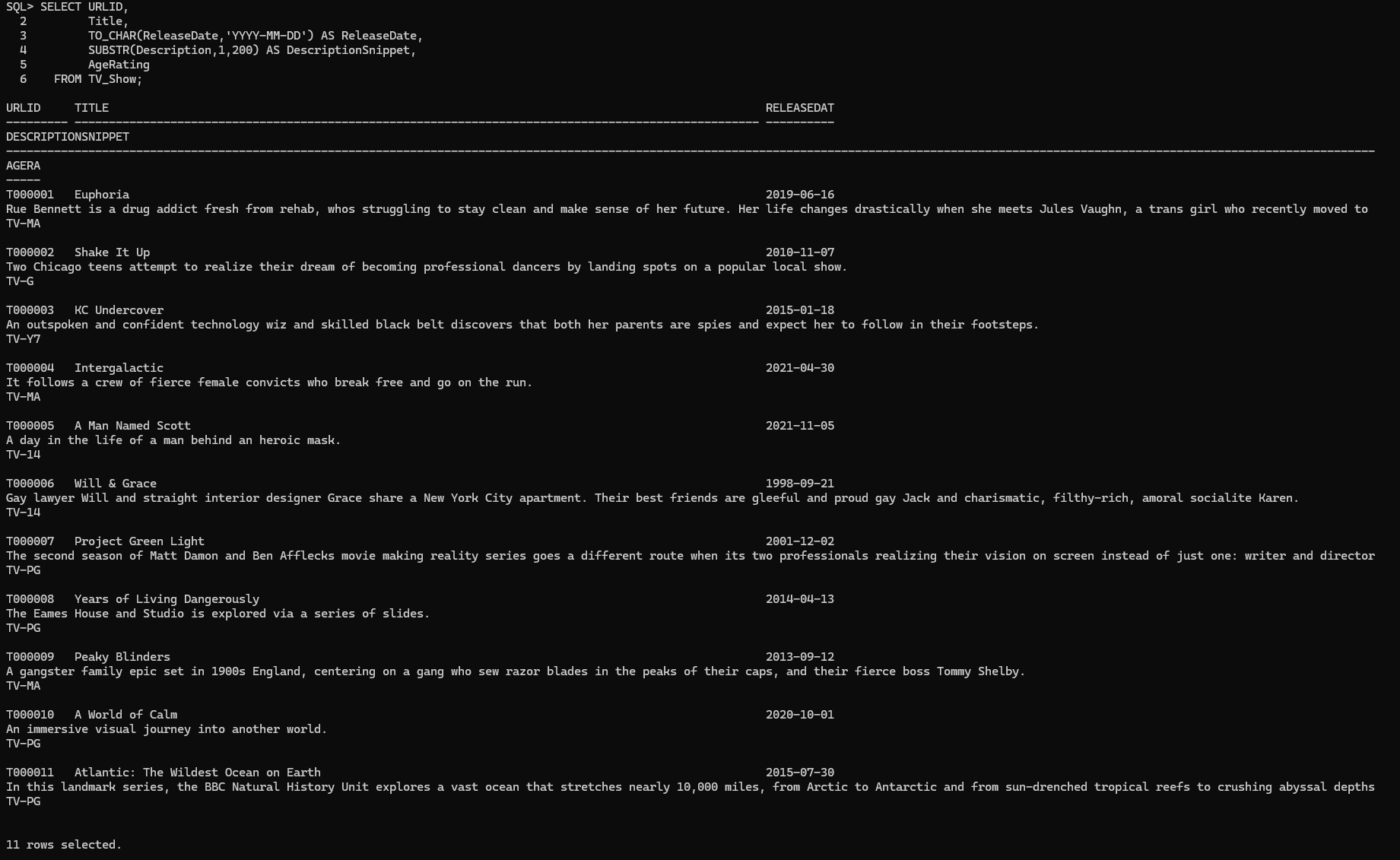
**SELECT \* FROM DIRECTOR:**

****

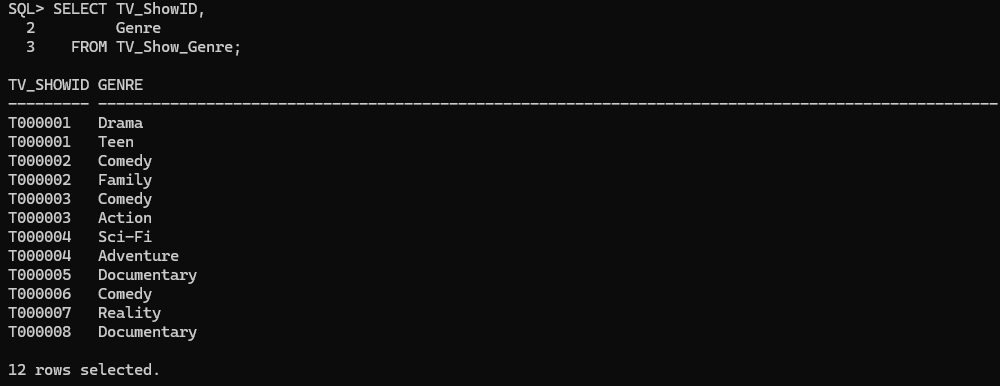
**SELECT \* FROM MOVIE\_GENRE:**

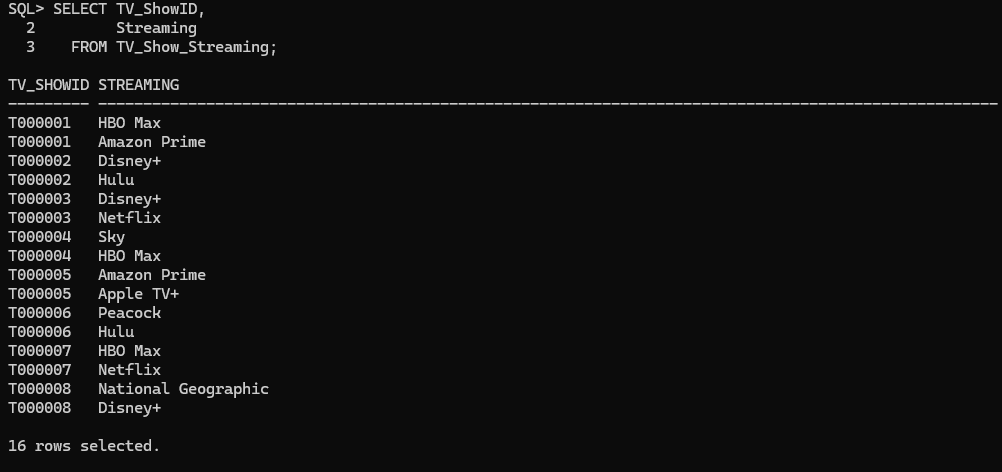
**SELECT \* FROM MOVIE\_STREAMING:**

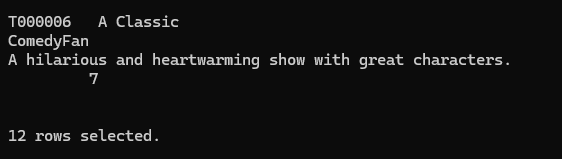
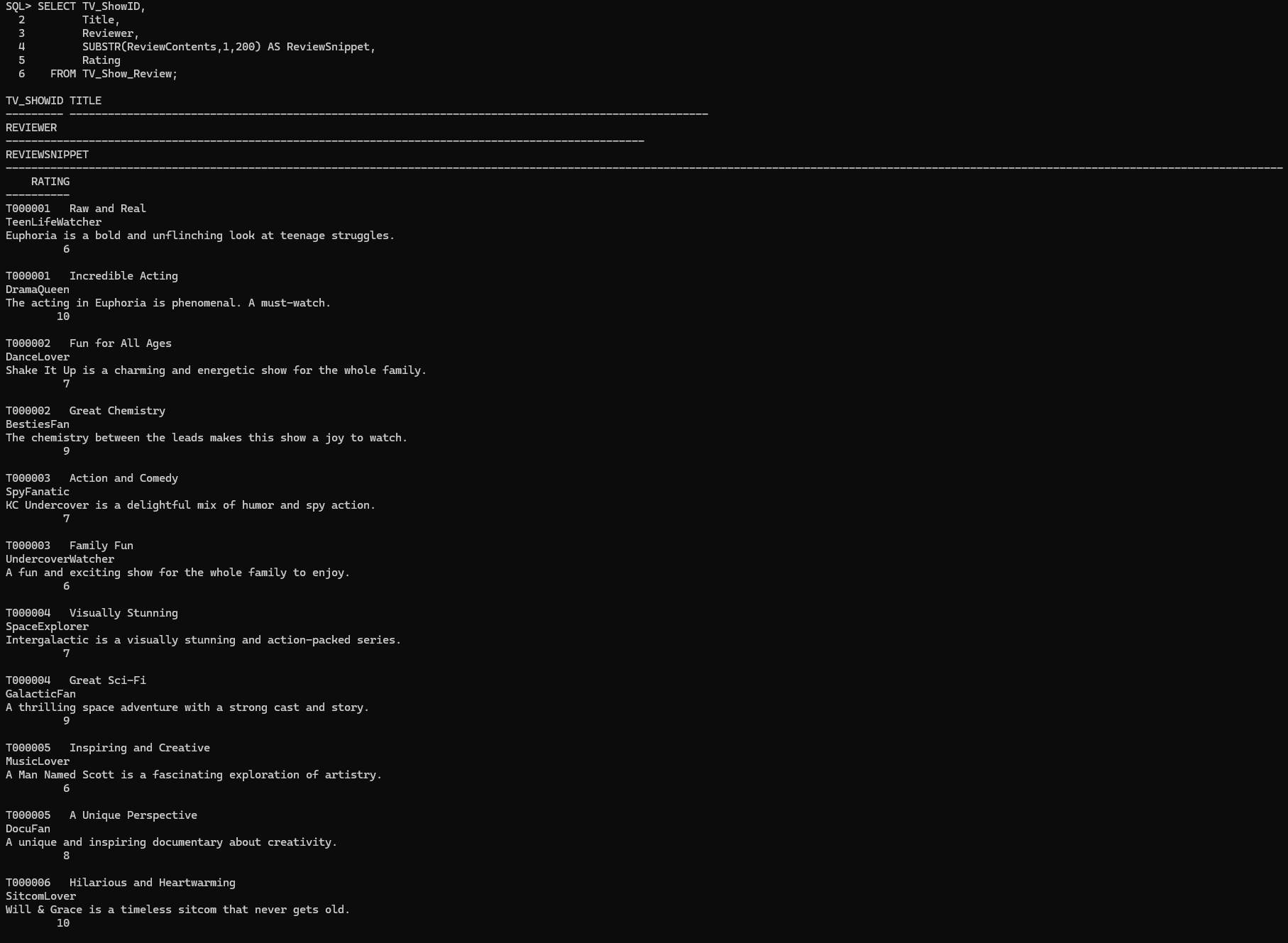
**SELECT \* FROM MOVIE\_REVIEW:**

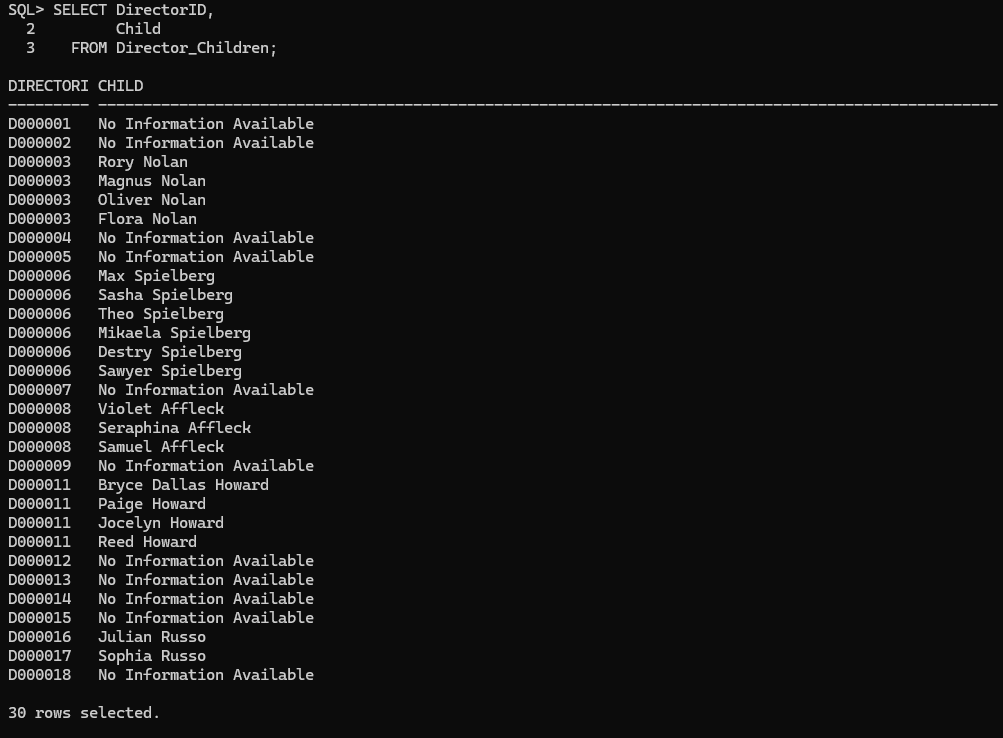
**SELECT \* FROM TV\_SHOW:**

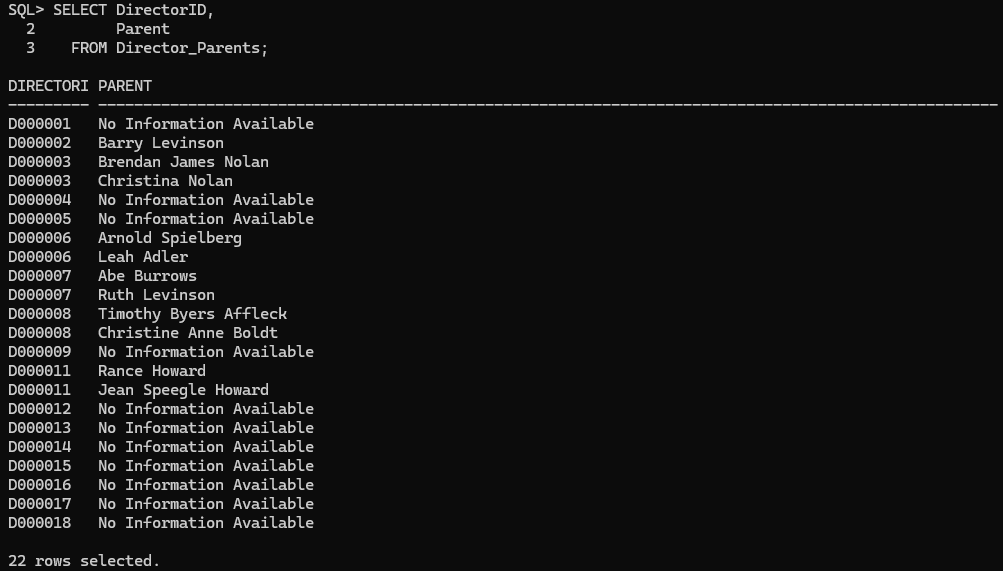
**SELECT \* FROM TV\_SHOW\_EPISODE:**

**SELECT \* FROM TV\_SHOW\_GENRE:**

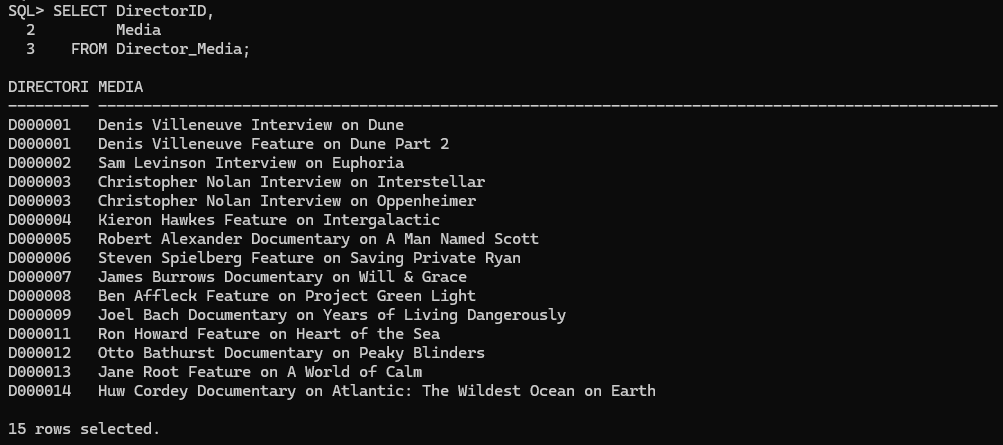
**SELECT \* FROM TV\_SHOW\_STREAMING:**

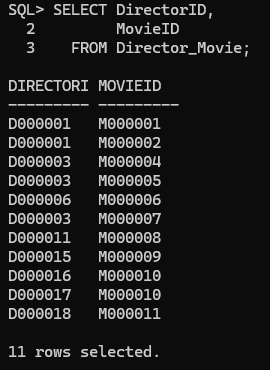
**SELECT \* FROM TV\_SHOW\_REVIEW:**

**SELECT \* FROM DIRECTOR\_CHILDREN:**

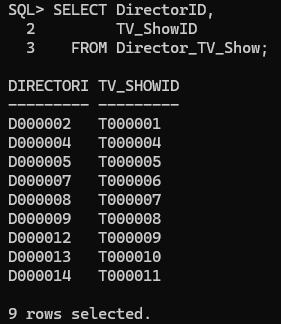
**SELECT \* FROM DIRECTOR\_PARENTS:**

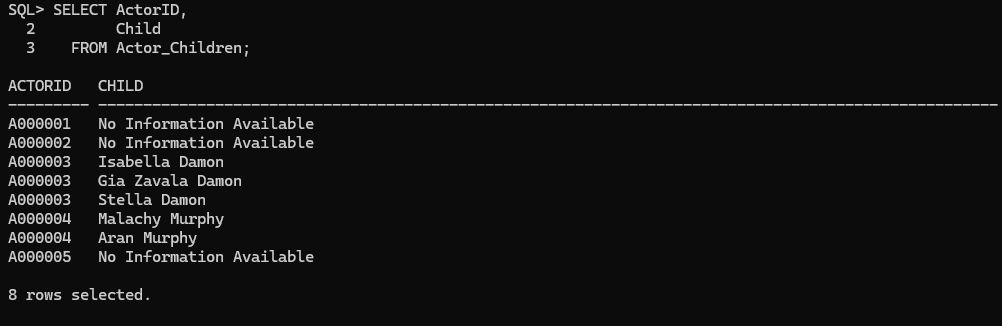
**SELECT \* FROM DIRECTOR\_MEDIA:**

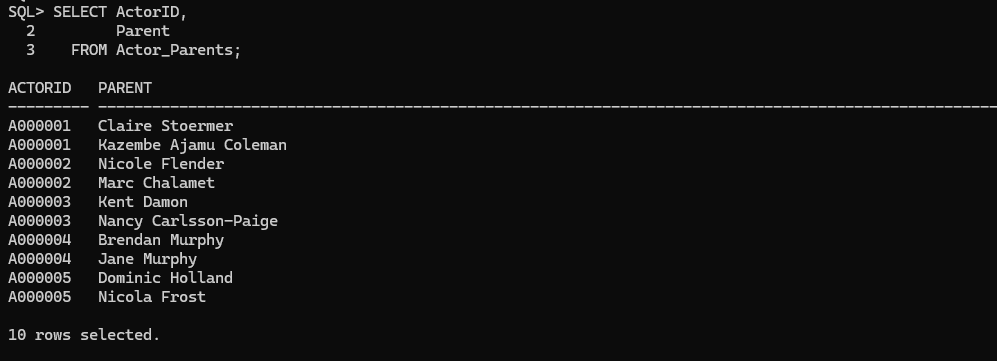
****

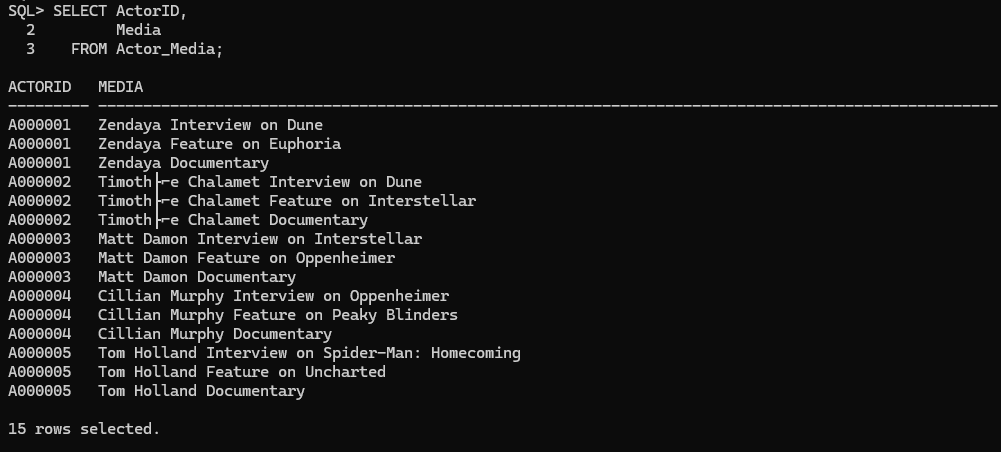
**SELECT \* FROM DIRECTOR\_MOVIE:**

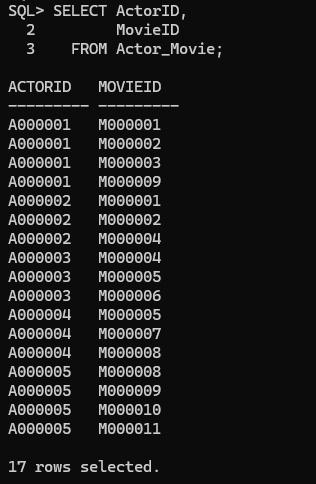
**SELECT \* FROM DIRECTOR\_TV\_SHOW:**

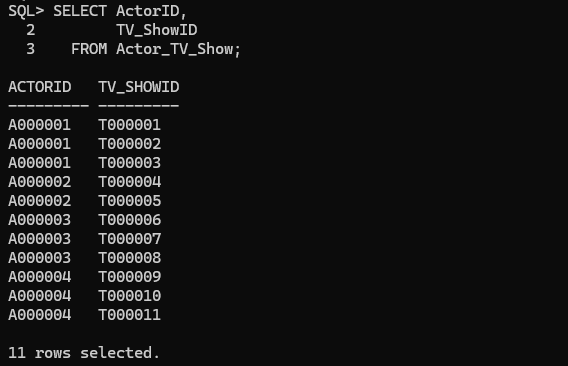
****

**SELECT \* FROM ACTOR\_CHILDREN:**

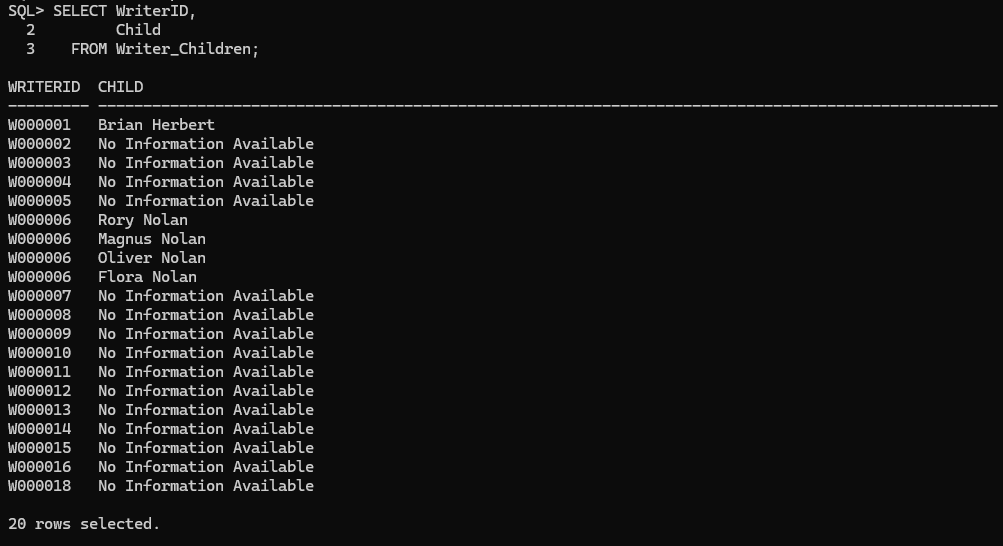
**SELECT \* FROM ACTOR\_PARENTS:**

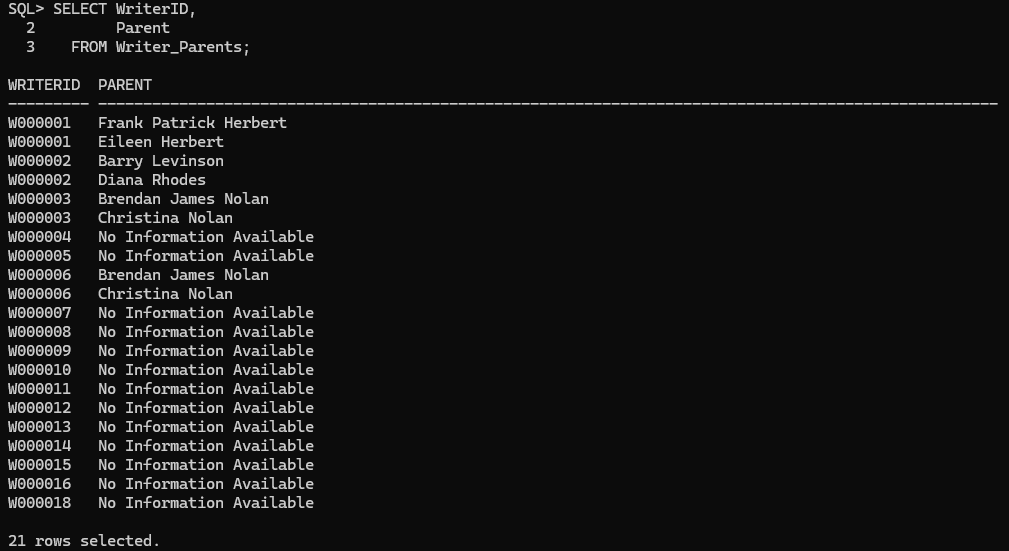
**SELECT \* FROM ACTOR\_MEDIA:**

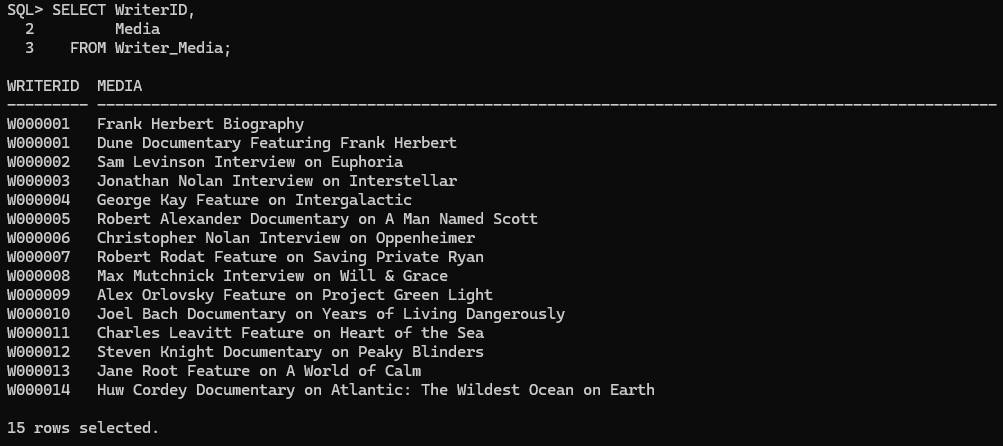
**SELECT \* FROM ACTOR\_MOVIE:**

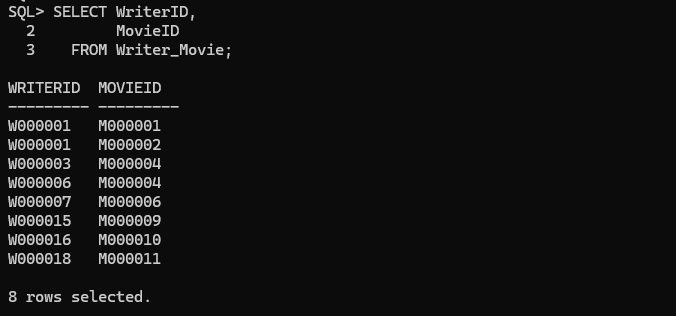
**SELECT \* FROM ACTOR\_TV\_SHOW:**

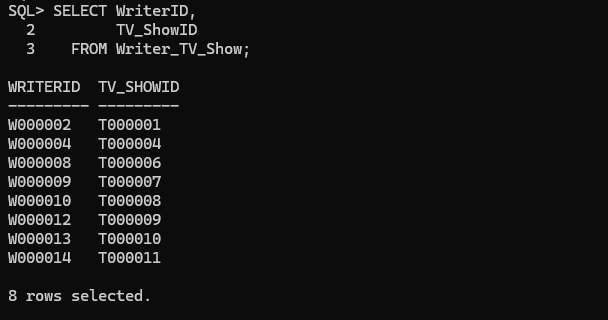
**SELECT \* FROM WRITER\_CHILDREN:**

****

**SELECT \* FROM WRITER\_PARENTS:**

**SELECT \* FROM WRITER\_MEDIA:**

**SELECT \* FROM WRITER\_MOVIE:**

**SELECT \* FROM WRITER\_TV\_SHOW:**

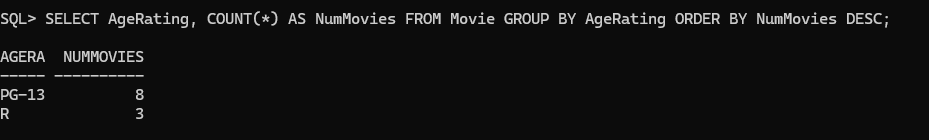
**Part 2 Queries:**

* Query 1 (one table):
* **English Description:** This query groups all movies in the movie table by their age rating, then counts how many movies fall into those categories. Then, it sorts the list where the age rating with the highest number of movies appears first.
* SELECT AgeRating, COUNT(\*) AS NumMovies

FROM Movie

GROUP BY AgeRating

ORDER BY NumMovies DESC;



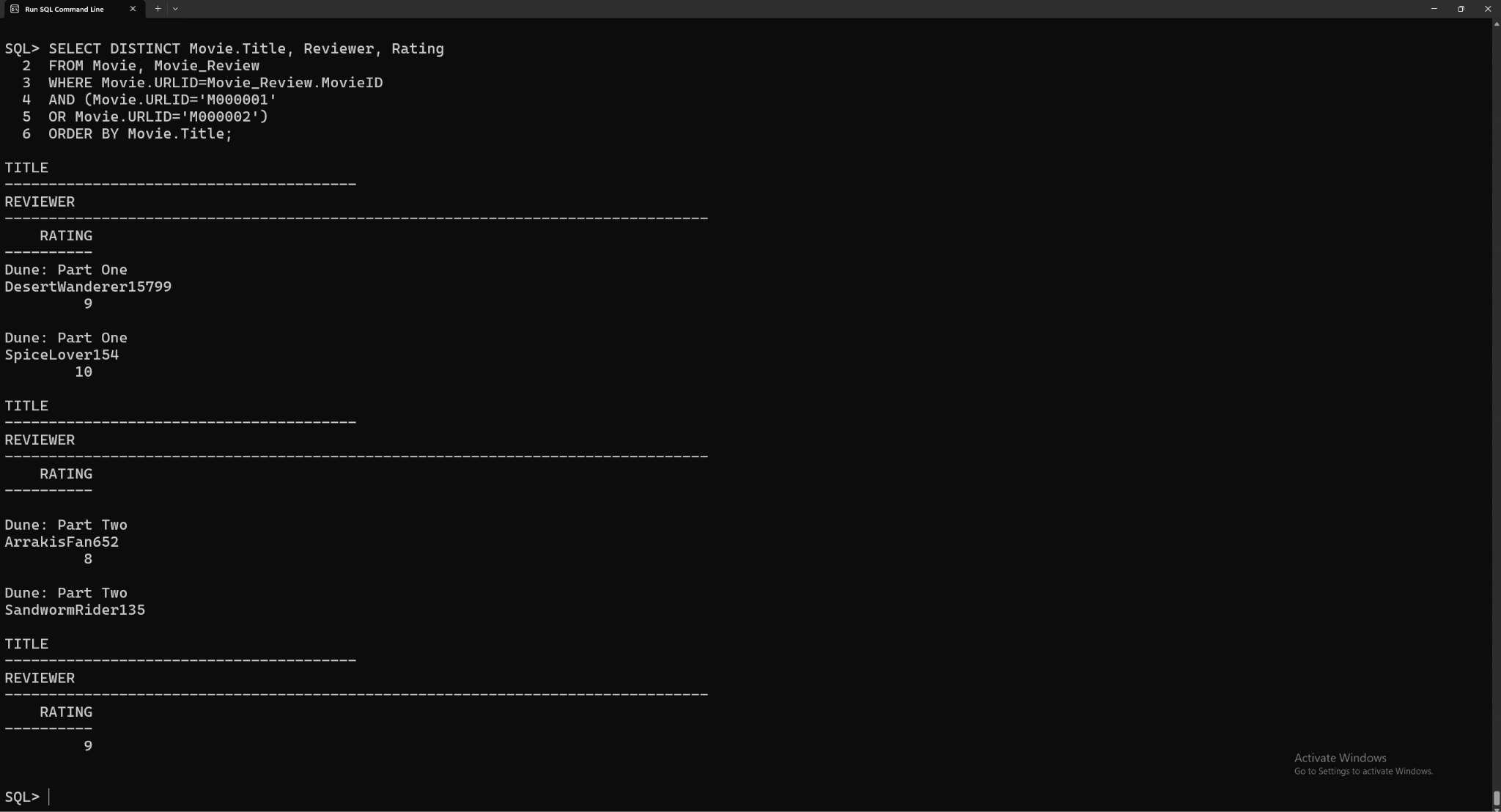
* Query 2 (two tables):
* **English Description:** This query joins the movie table and the movie review table. It then filters the results so you only get the movies with the two IDs, which are Dune: Part One and Dune: Part Two. It uses distinct to make sure there are no duplicates, and finally, it sorts the list alphabetically by movie title.
* SELECT DISTINCT Movie.Title, Reviewer, Rating

FROM Movie, Movie\_Review

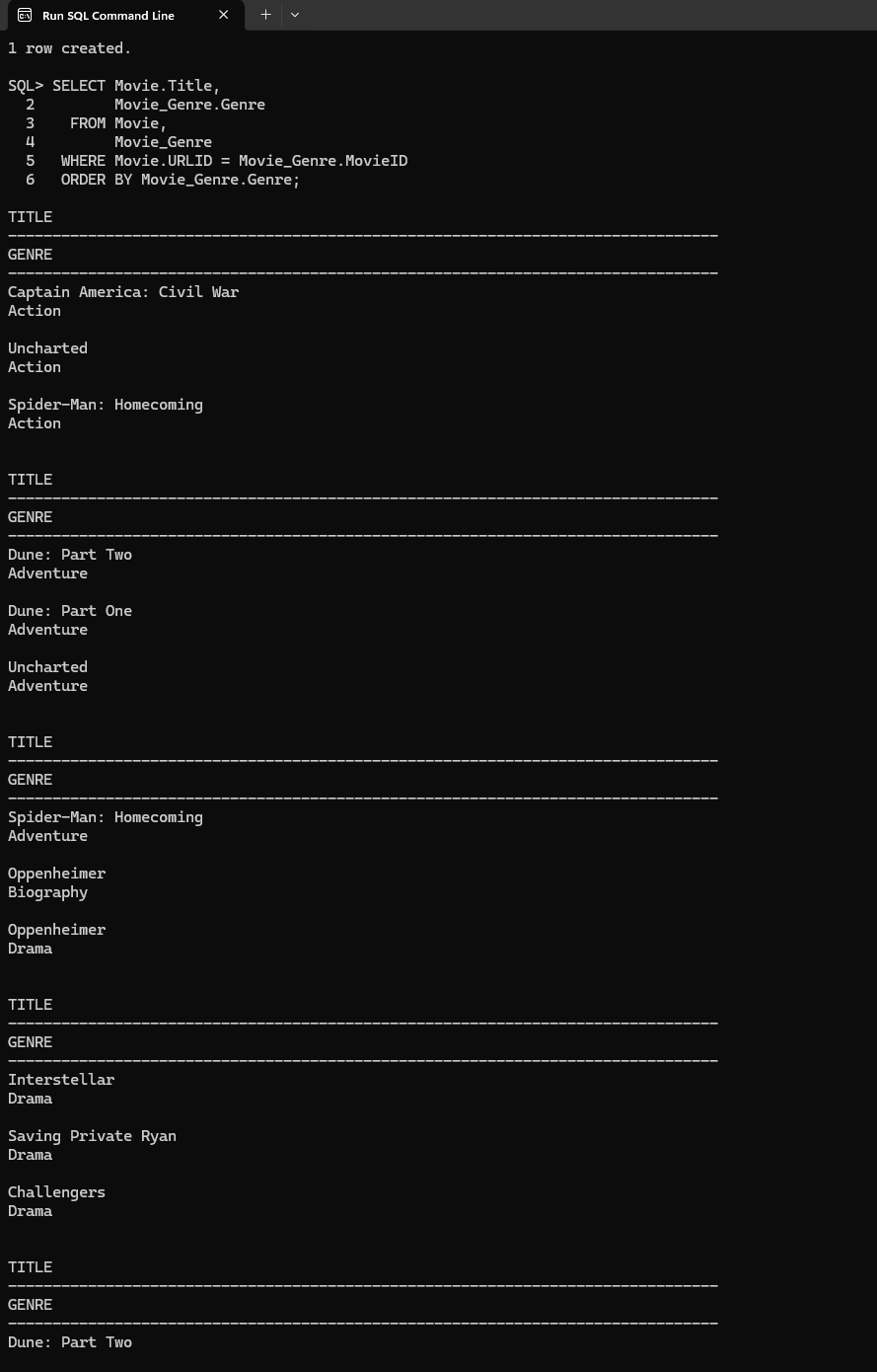
WHERE Movie.URLID=Movie\_Review.MovieID

AND (Movie.URLID='M000001' OR Movie.URLID='M000002')

ORDER BY Movie.Title;



* Query 3 (two tables):
* **English Description:** This query joins the movie table with the movie’s genre. Then it selects the movie title along with its genre. Lastly, it orders the list alphabetically by genre so you will see the movies grouped together by genre.
* SELECT Movie.Title, Movie\_Genre.Genre
* FROM Movie, Movie\_Genre
* WHERE Movie.URLID = Movie\_Genre.MovieID
* ORDER BY Movie\_Genre.Genre;



* Query 4 (three tables):
* **English Description:** This query retrieves the movie title, director's name, and the average rating of the movie.
* SELECT M.Title AS MovieTitle, D.FirstName, D.LastName AS DirectorName, MAR.AverageRating

FROM Movie M

JOIN Director\_Movie DM ON M.URLID = DM.MovieID

JOIN Director D ON DM.DirectorID = D.URLID

LEFT JOIN Movie\_Average\_Rating MAR ON M.URLID = MAR.MovieID

ORDER BY MAR.AverageRating DESC;



* Query 5 (three tables):
* **English Description:** This query retrieves the TV show title, writer's name, and the streaming platforms where the TV show is available.
* SELECT A.FirstName ' ' A.LastName AS ActorName, TV.Title AS TVShowTitle, COUNT(E.EpisodeNumber) AS EpisodeCount

FROM Actor A

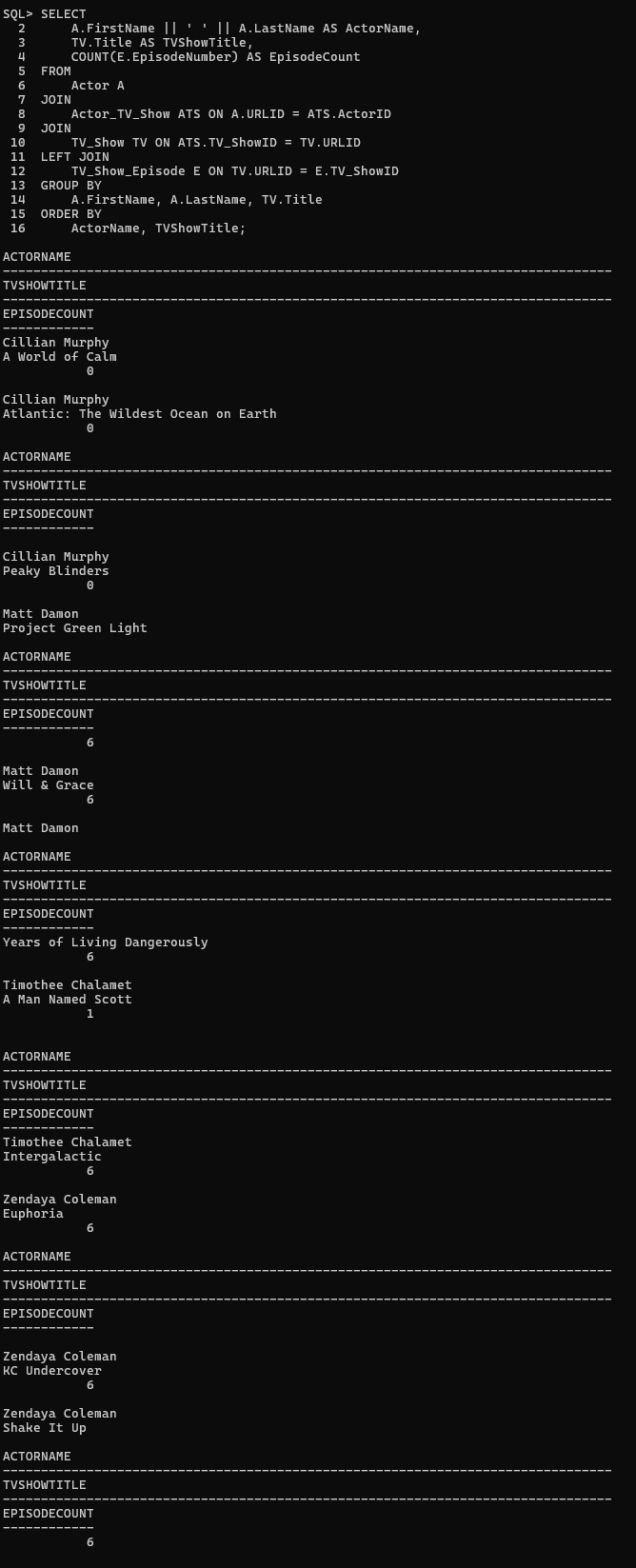
JOIN Actor\_TV\_Show ATS ON A.URLID = ATS.ActorID

JOIN TV\_Show TV ON ATS.TV\_ShowID = TV.URLID

LEFT JOIN TV\_Show\_Episode E ON TV.URLID = E.TV\_ShowID

GROUP BY A.FirstName, A.LastName, TV.Title

ORDER BY ActorName, TVShowTitle;



* Query 6 (four tables):
* **English Description:** this is a query that joins four tables to retrieve information about TV shows, their genres, the directors who directed them, and the streaming platforms where they are available:
* SELECT T.Title AS TVShowTitle, T.ReleaseDate,TG.Genre,

D.FirstName || ' ' || D.LastName AS DirectorName,

TS.Streaming AS StreamingPlatform

FROM TV\_Show T

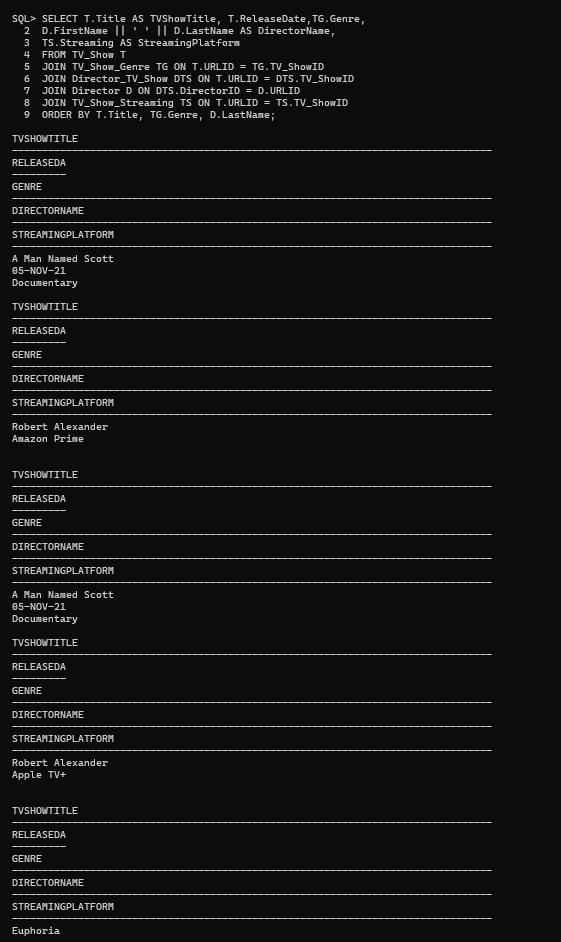
JOIN TV\_Show\_Genre TG ON T.URLID = TG.TV\_ShowID

JOIN Director\_TV\_Show DTS ON T.URLID = DTS.TV\_ShowID

JOIN Director D ON DTS.DirectorID = D.URLID

JOIN TV\_Show\_Streaming TS ON T.URLID = TS.TV\_ShowID

ORDER BY T.Title, TG.Genre, D.LastName;



* Query 7 (use set theory):
* **English Description:** Retrieve the names of actors who have appeared in both movies and TV shows, along with the count of movies and TV shows they have acted in, and is then ordered alphabetically.
* SELECT FirstName, LastName, count(DISTINCT Actor\_Movie.MovieID) MovieCount,

count(DISTINCT Actor\_TV\_Show.TV\_ShowID) AS TVShowCount

FROM Actor, Actor\_Movie, Actor\_TV\_Show

WHERE Actor.URLID=Actor\_Movie.ActorID

AND URLID = Actor\_TV\_Show.ActorID

AND URLID IN (

SELECT ActorID FROM Actor\_Movie

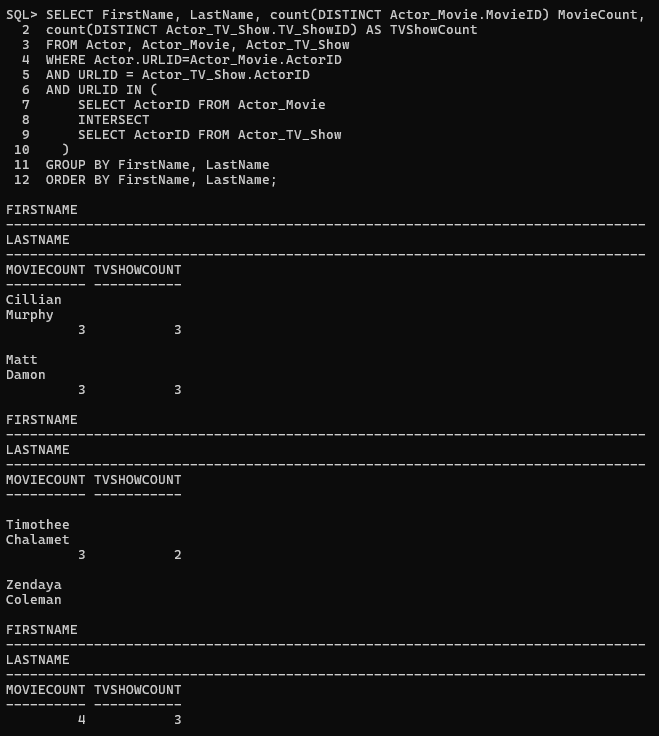
INTERSECT

SELECT ActorID FROM Actor\_TV\_Show

)

GROUP BY FirstName, LastName

ORDER BY FirstName, LastName;



* Query 8 (use set theory):
* **English Description:** Find all director IDs, first names, and last names of those who have directed movies but have not directed any TV shows.
* CREATE TABLE Director\_Movies\_Only AS

SELECT URLID AS DirectorID, FirstName, LastName

FROM Director, Director\_Movie

WHERE Director.URLID = Director\_Movie.DirectorID;

* SELECT DirectorID, FirstName, LastName

FROM Director\_Movies\_Only

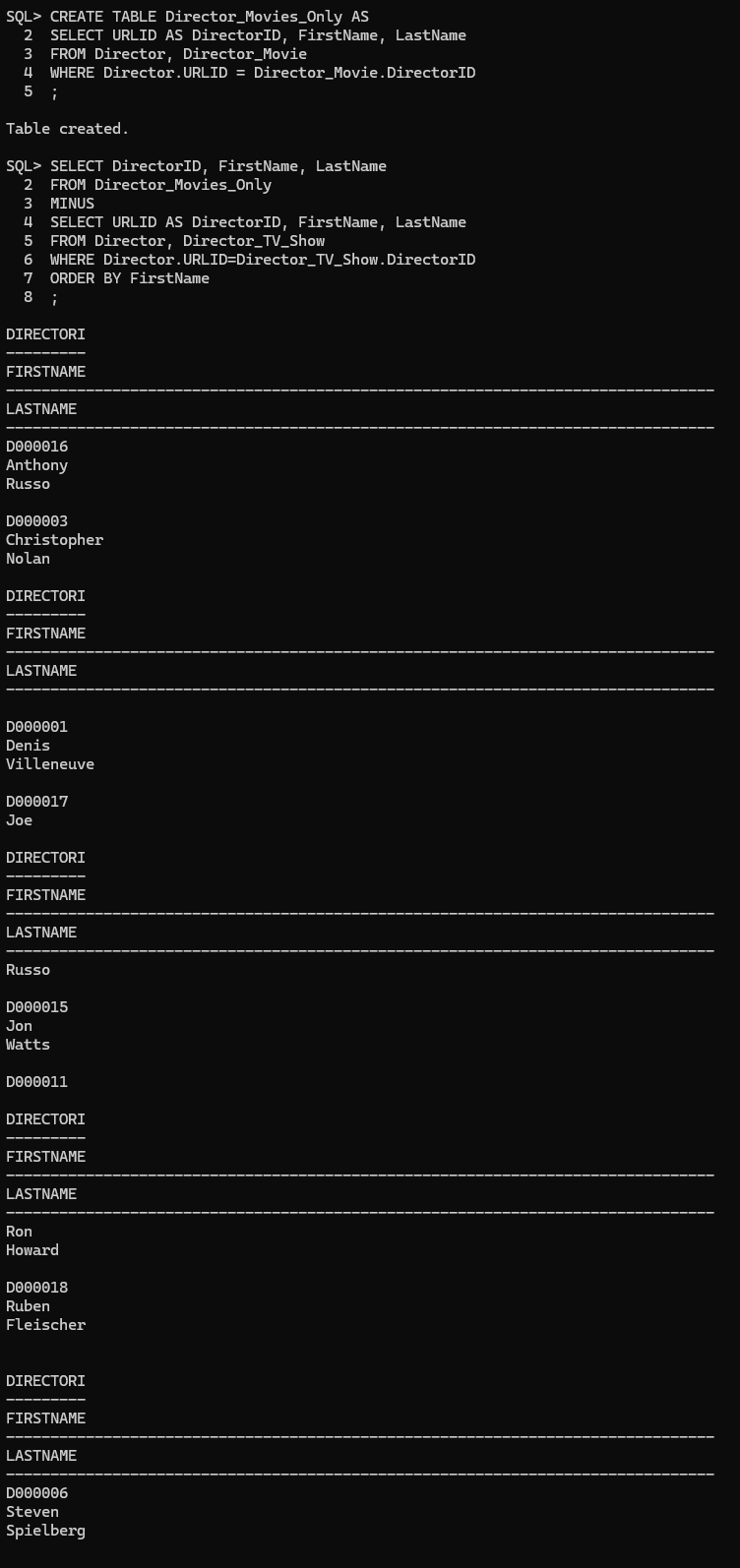
MINUS

SELECT URLID AS DirectorID, FirstName, LastName

FROM Director, Director\_TV\_Show

WHERE Director.URLID=Director\_TV\_Show.DirectorID

ORDER BY FirstName;



* Query 9 (use aggregate and grouping):
* **English Description:** Retrieve the most popular genres by calculating the average rating of movies grouped by their genre and ordered from highest to lowest.
* SELECT Genre,

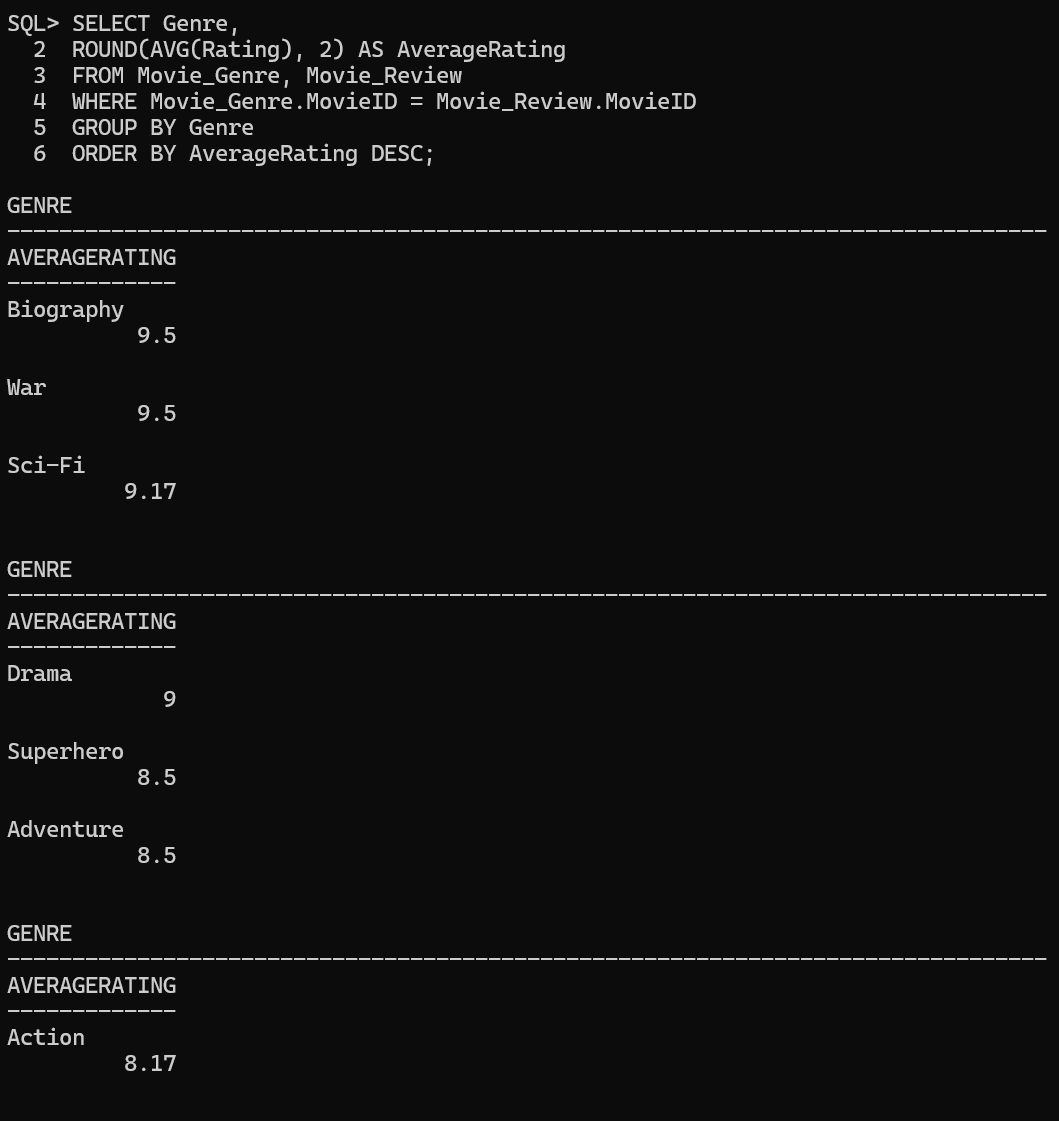
ROUND(AVG(Rating), 2) AS AverageRating

FROM Movie\_Genre, Movie\_Review

WHERE Movie\_Genre.MovieID = Movie\_Review.MovieID

GROUP BY Genre

ORDER BY AverageRating DESC;



* Query 10 (use aggregate and grouping):
* Retrieve the youngest and oldest ages of writers who have won an award, grouped by the award.
* CREATE TABLE Writer\_Age\_Award AS

SELECT Award, FirstName, LastName, DateOfBirth

FROM Writer

WHERE Award IS NOT NULL;

* SELECT Award,

MIN(FLOOR(MONTHS\_BETWEEN(SYSDATE, DateOfBirth) / 12)) AS YoungestAge,

MAX(FLOOR(MONTHS\_BETWEEN(SYSDATE, DateOfBirth) / 12)) AS OldestAge

FROM Writer\_Age\_Award

GROUP BY Award

ORDER BY Award;

