

```

SELECT * FROM system_schema.keyspaces;

CREATE KEYSPACE movies WITH replication = {'class': 'SimpleStrategy',
'replication_factor': '1'} AND durable_writes = true;

DESCRIBE KEYSPACE movies;

SELECT * FROM system_schema.keyspaces;

DROP TABLE IF EXISTS movies.movie;
CREATE TABLE movies.movie (movie_id int,
    title text,          // title
    release_year int,    // year
    running_time int,     // runtimes
    languages set<text>, // language codes
    genres set<text>,     // genres
    plot_outline text,    // plot outline
    cover_url text,       // cover url
    top250_rank int,      // top 250 rank
    PRIMARY KEY (movie_id)
);

DROP TABLE IF EXISTS movies.actor;
CREATE TABLE movies.actor (actor_id int,
    name text,           // name
    headshot_url text,   // headshot
    mini_biography text, // mini biography
    birth_date text,     // birth date
    trade_mark list<text>, // trade mark
    PRIMARY KEY (actor_id)
);

DESCRIBE TABLE movie;

DESCRIBE TABLE actor;

// insert "The Matrix" - 0133093
INSERT INTO movies.movie (movie_id, title, release_year, running_time, languages,
    genres, plot_outline, cover_url, top250_rank)
VALUES (0133093,
    'The Matrix',
    1999,
    136,
    {'en'},
    {'Action', 'Sci-Fi'},
    $$Thomas A. Anderson is a man living two lives. By day he is an average computer programmer and by night a hacker known as Neo. Ne
    'https://m.media-amazon.com/images/M/MV5BNzQzOTk3OTAtNDQ0Zi00ZTVkLWI0MTetMD11ZjNkYzNjNTc4L2ltYWdlXkEyXkFqcGdeQXVyNjU0OTQ0OTY@._V1_
    19);

// insert "Pulp Fiction" - 0110912
INSERT INTO movies.movie (movie_id, title, release_year, running_time, languages,
    genres, plot_outline, cover_url, top250_rank)
VALUES (0110912,
    'Pulp Fiction',
    1994,
    154,
    {'en', 'es', 'fr'},
    {'Crime', 'Drama'},
    $$Jules Winnfield (Samuel L. Jackson) and Vincent Vega (John Travolta) are two hit men who are out to retrieve a suitcase stolen
    'https://m.media-amazon.com/images/M/MV5BNhMDIzTU00MTRlLWFjM2ItYzViMjE3YzI5MjIjXkEyXkFqcGdeQXVyNzkwMjQ5NzM@._V1_SY150_
    8);

// insert "Speed" - 0111257
INSERT INTO movies.movie (movie_id, title, release_year, running_time, languages,
    genres, plot_outline, cover_url, top250_rank)
VALUES (0111257,
    'Speed',
    1994,
    116,
    {'en'},
    {'Action', 'Adventure', 'Crime', 'Thriller'},
    $$Bomber extortionist's elevator plan backfires, so he rigs a bomb to a LA city bus. The stipulation is: once armed, the bus mu
    'https://m.media-amazon.com/images/M/MV5BYjc0MjYyN2EtZGRhMy00NzJlLWI2Y2QtYzh1YTU3NzAxNzg4XkEyXkFqcGdeQXVyMTQxNzMzNDI@._V1_SY150_
    null);

```

```

// insert "Bruce Willis" - 0000246
INSERT INTO movies.actor (actor_id, name, headshot_url, birth_date, trade_mark)
VALUES (0000246,
    'Bruce Willis',
    'https://m.media-amazon.com/images/M/MV5BMjA0MjMTE5OF5BMl5BanBnXkFtZTcwMzQ2ODE3Mw@@._V1_UY98_CR8,0,67,98_AL_.jpg',
    '1955-03-19',
    ['Frequently plays a man who suffered a tragedy, had lost something or had a crisis of confidence or conscience.',
    'Frequently plays likeable wisecracking heroes with a moral centre',
    'Headlines action-adventures, often playing a policeman, hitman or someone in the military',
    'Often plays men who get caught up in situations far beyond their control',
    'Sardonic one-liners',
    'Shaven head',
    'Distinctive, gravelly voice',
    'Smirky grin.',
    'Known for playing cynical anti-heroes with unhappy personal lives']);

// insert "John Travolta" - 0000237
INSERT INTO movies.actor (actor_id, name, headshot_url, birth_date, trade_mark)
VALUES (0000237,
    'John Travolta',
    'https://m.media-amazon.com/images/M/MV5BMTUwNjQ0ODkxN15BMl5BanBnXkFtZTcwMDc5NjQwNw@@._V1_UY98_CR3,0,67,98_AL_.jpg',
    '1954-02-18',
    ['Cleft chin and razor-sharp cheekbones',
    'Often works some sort of dance into his roles',
    'New Jersey accent',
    'Black hair and blue eyes']);

// insert "Sandra Bullock" - 0000113
INSERT INTO movies.actor (actor_id, name, headshot_url, birth_date, trade_mark)
VALUES (0000113,
    'Sandra Bullock',
    'https://m.media-amazon.com/images/M/MV5BMTI5NDY5NjU3NF5BMl5BanBnXkFtZTcwMzQ0MTMyMw@@._V1_UX67_CR0,0,67,98_AL_.jpg',
    '1964-07-26',
    null);

// insert "Samuel L. Jackson" - 0000168
INSERT INTO movies.actor (actor_id, name, headshot_url, birth_date, trade_mark)
VALUES (0000168,
    'Samuel L. Jackson',
    'https://m.media-amazon.com/images/M/MV5BMTQ1NTQwMTYxN15BMl5BanBnXkFtZTYwMjA1MzY1._V1_UX67_CR0,0,67,98_AL_.jpg',
    '1948-12-21',
    ['Deep authoritative voice',
    'Rebellious characters who are disliked or considered strange by others in the story',
    'Often plays police officers or government officials. Both prone to intimidation or violence',
    'Often plays very wise and intelligent characters with great capacities for violence',
    'Frequently plays tough characters who swear a lot',
    'Frequent swearing',
    'Often sports a moustache or goatee in his films',
    'Shaven head',
    'Kangol hats',
    'Often plays hotheaded characters with a fiery temper',
    'Often shouts the word "motherf*****" at some point in a film',
    'Frequently cast by Quentin Tarantino']);

// insert "Uma Thurman" - 0000235
INSERT INTO movies.actor (actor_id, name, headshot_url, birth_date, trade_mark)
VALUES (0000235,
    'Uma Thurman',
    'https://m.media-amazon.com/images/M/MV5BMjMxNzk1MTQyMl5BMl5BanBnXkFtZTgwMDIzMDEyMTE@._V1_UX67_CR0,0,67,98_AL_.jpg',
    '1970-04-29',
    ['Long blond hair and blue eyes', 'Statuesque, model-like figure']);

// insert "Keanu Reeves" - 0000206
INSERT INTO movies.actor (actor_id, name, headshot_url, birth_date, trade_mark)
VALUES (0000206,
    'Keanu Reeves',
    'https://m.media-amazon.com/images/M/MV5BNjUxNDcwMTg4Ml5BMl5BanBnXkFtZTcwMjU4NDYyOA@@._V1_UY98_CR4,0,67,98_AL_.jpg',
    '1964-09-02',
    ['Intense contemplative gaze',
    'Deep husky voice',
    'Known for playing stoic reserved characters']);

// insert "Quentin Tarantino" - 0000233
INSERT INTO movies.actor (actor_id, name, headshot_url, birth_date, trade_mark)
VALUES (0000233,
    'Quentin Tarantino',
    'https://m.media-amazon.com/images/M/MV5BMTgyMjI3ODAzN15BMl5BanBnXkFtZTcwNzY2MDYxOQ@@._V1_UX67_CR0,0,67,98_AL_.jpg',
    '1963-03-27',
    ['Lead characters usually drive General Motors vehicles, particularly Chevrolet and Cadillac, such as Jules 1974 Nova and Vince',
    'Briefcases and suitcases play an important role in Pulp Fiction (1994), Reservoir Dogs (1992), Jackie Brown (1997), True Romi

```

```
'Makes references to cult movies and television',
'Frequently works with Harvey Keitel, Tim Roth, Michael Madsen, Uma Thurman, Michael Bowen, Samuel L. Jackson, Michael Parks ;
'His films usually have a shot from inside an automobile trunk',
'He always has a Dutch element in his films: The opening tune, Little Green Bag, in Reservoir Dogs (1992) was performed by Ge
'[The Mexican Standoff] All his movies (including True Romance (1993), which he only wrote and did not direct) feature a scene
'Often uses an unconventional storytelling device in his films, such as retrospect (Reservoir Dogs (1992)), non-linear (Pulp F:
'His films will often include one long, unbroken take where a character is followed around somewhere.']);
```

```
SELECT movie_id, title
FROM movies.movie;
```

```
SELECT *
FROM movies.actor;
```

```
UPDATE movies.actor
SET name = 'Bruce Walter Willis'
WHERE actor_id = 0000246;
```

```
SELECT *
FROM movies.actor
WHERE actor_id = 0000246;
```

```
DROP TABLE IF EXISTS movies.movies_by_actor;
CREATE TABLE movies.movies_by_actor (actor_id int,
                                     movie_id int,
                                     title text,
                                     PRIMARY KEY (actor_id, movie_id)
);
```

```
DROP TABLE IF EXISTS movies.actors_by_movie;
CREATE TABLE movies.actors_by_movie (movie_id int,
                                     title text STATIC,
                                     actor_id int,
                                     name text,
                                     PRIMARY KEY (movie_id, actor_id)
);
```

```
// Фильмы для актера "Bruce Willis"
```

```
INSERT INTO movies.movies_by_actor (actor_id, movie_id, title)
VALUES (0000246, 0110912, 'Pulp Fiction');
```

```
INSERT INTO movies.movies_by_actor (actor_id, movie_id, title)
VALUES (0000246, 1606378, 'A Good Day to Die Hard');
```

```
INSERT INTO movies.movies_by_actor (actor_id, movie_id, title)
VALUES (0000246, 0217869, 'Unbreakable');
```

```
INSERT INTO movies.movies_by_actor (actor_id, movie_id, title)
VALUES (0000246, 0377917, 'The Fifth Element');
```

```
INSERT INTO movies.movies_by_actor (actor_id, movie_id, title)
VALUES (0000246, 0112864, 'Die Hard: With a Vengeance');
```

```
// Фильмы для актера "Keanu Reeves"
```

```
INSERT INTO movies.movies_by_actor (actor_id, movie_id, title)
VALUES (0000206, 0133093, 'The Matrix');
```

```
INSERT INTO movies.movies_by_actor (actor_id, movie_id, title)
VALUES (0000206, 0234215, 'The Matrix Reloaded');
```

```
INSERT INTO movies.movies_by_actor (actor_id, movie_id, title)
VALUES (0000206, 0111257, 'Speed');
```

```
// Фильмы для актера "Sandra Bullock"
```

```
INSERT INTO movies.movies_by_actor (actor_id, movie_id, title)
VALUES (0000113, 0111257, 'Speed');
```

```
// Актеры для фильма "The Matrix"
```

```
INSERT INTO movies.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0133093, 'The Matrix', 0000206, 'Keanu Reeves');
```

```
INSERT INTO movies.actors_by_movie (movie_id, title, actor_id, name)
```

```

VALUES (0133093, 'The Matrix', 0000401, 'Laurence Fishburne');

INSERT INTO movies.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0133093, 'The Matrix', 0005251, 'Carrie-Anne Moss');

INSERT INTO movies.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0133093, 'The Matrix', 0915989, 'Hugo Weaving');

// Актеры для фильма "Pulp Fiction"

INSERT INTO movies.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0110912, 'Pulp Fiction', 0000237, 'John Travolta');

INSERT INTO movies.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0110912, 'Pulp Fiction', 0000168, 'Samuel L. Jackson');

INSERT INTO movies.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0110912, 'Pulp Fiction', 0000246, 'Bruce Willis');

SELECT title
FROM movies.movies_by_actor
WHERE actor_id = 0000246;

SELECT name, title
FROM movies.actors_by_movie
WHERE movie_id = 0110912;

DROP TABLE IF EXISTS movies.rating_by_movie;
CREATE TABLE movies.rating_by_movie (movie_id int,
    one_star counter,
    two_star counter,
    three_star counter,
    four_star counter,
    five_star counter,
    PRIMARY KEY (movie_id)
);

UPDATE movies.rating_by_movie
SET five_star = five_star + 1
WHERE movie_id = 0110912;

UPDATE movies.rating_by_movie
SET four_star = four_star + 1
WHERE movie_id = 0110912;

UPDATE movies.rating_by_movie
SET five_star = five_star + 1
WHERE movie_id = 0110912;

UPDATE movies.rating_by_movie
SET five_star = five_star + 1
WHERE movie_id = 0110912;

UPDATE movies.rating_by_movie
SET two_star = two_star + 1
WHERE movie_id = 0110912;

SELECT * FROM movies.rating_by_movie WHERE movie_id = 0110912;

DROP TABLE IF EXISTS movies.movie_viewed_by_time;
CREATE TABLE movies.movie_viewed_by_time (movie_id int,
    year int,
    month int,
    male counter,
    female counter,
    PRIMARY KEY (movie_id, year, month)
) WITH CLUSTERING ORDER BY (year DESC, month DESC);

// Pulp Fiction Views 2019/03
UPDATE movies.movie_viewed_by_time
SET male = male + 1
WHERE movie_id = 0110912 AND year = 2019 and month = 03;

UPDATE movies.movie_viewed_by_time
SET male = male + 1

```

```
WHERE movie_id = 0110912 AND year = 2019 and month = 03;
```

```
UPDATE movies.movie_viewed_by_time  
SET female = female + 1  
WHERE movie_id = 0110912 AND year = 2019 and month = 03;
```

```
// Pulp Fiction Views 2019/04  
UPDATE movies.movie_viewed_by_time  
SET female = female + 1  
WHERE movie_id = 0110912 AND year = 2019 and month = 04;
```

```
UPDATE movies.movie_viewed_by_time  
SET male = male + 1  
WHERE movie_id = 0110912 AND year = 2019 and month = 04;
```

```
UPDATE movies.movie_viewed_by_time  
SET female = female + 1  
WHERE movie_id = 0110912 AND year = 2019 and month = 04;
```

```
UPDATE movies.movie_viewed_by_time  
SET female = female + 1  
WHERE movie_id = 0110912 AND year = 2019 and month = 04;
```

```
// Pulp Fiction Views 2019/05  
UPDATE movies.movie_viewed_by_time  
SET male = male + 1  
WHERE movie_id = 0110912 AND year = 2019 and month = 05;
```

```
UPDATE movies.movie_viewed_by_time  
SET male = male + 1  
WHERE movie_id = 0110912 AND year = 2019 and month = 05;
```

```
UPDATE movies.movie_viewed_by_time  
SET female = female + 1  
WHERE movie_id = 0110912 AND year = 2019 and month = 05;
```

```
UPDATE movies.movie_viewed_by_time  
SET female = female + 1  
WHERE movie_id = 0110912 AND year = 2019 and month = 05;
```

```
SELECT *  
FROM movies.movie_viewed_by_time  
WHERE movie_id = 0110912;
```

```
SELECT *  
FROM movies.movie_viewed_by_time  
WHERE movie_id = 0110912 AND year = 2019 AND month = 05;
```

```
SELECT *  
FROM movies.movie_viewed_by_time  
WHERE movie_id = 0110912 AND year = 2019 AND month >= 01 AND month <= 5;
```

```
//Индивидуальное задание. Вариант 13
```

```
CREATE KEYSPACE library WITH replication =  
{'class':'SimpleStrategy','replication_factor':1};  
DESCRIBE KEYSPACE library;
```

```
DROP TABLE IF EXISTS library.books;  
CREATE TABLE library.books (  
    book_id int,  
    title text,  
    author text,  
    year_published int,  
    genre text,  
    isbn text,  
    PRIMARY KEY (book_id)  
);
```

```
-- Классика  
INSERT INTO library.books (book_id, title, author, year_published, genre, isbn)  
VALUES (1, 'To Kill a Mockingbird', 'Harper Lee', 1960, 'Classic', '978-0061120084');
```

```
-- Фантастика  
INSERT INTO library.books (book_id, title, author, year_published, genre, isbn)  
VALUES (2, 'Dune', 'Frank Herbert', 1965, 'Science Fiction', '978-0441172719');
```

```
-- Программирование
INSERT INTO library.books (book_id, title, author, year_published, genre, isbn)
VALUES (3, 'Clean Code', 'Robert C. Martin', 2008, 'Programming', '978-0132350884');

-- Детектив
INSERT INTO library.books (book_id, title, author, year_published, genre, isbn)
VALUES (4, 'The Girl with the Dragon Tattoo', 'Stieg Larsson', 2005, 'Mystery', '978-0307269751');

SELECT * FROM library.books;

UPDATE library.books
SET year_published = 1950
WHERE book_id = 2;

SELECT title, author, year_published
FROM library.books
WHERE book_id = 2;
```