**PROJECT ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ 2021-2022**

Στοιχεία ομάδας

Παπαδόπουλος Βασίλειος, ΑΜ : 1080422, 3ο έτος

Κατσικαρέλης Πέτρος, ΑΜ : 1072554, 3ο έτος

Επιλέγοντας Προβολή -> Παράθυρο περιήγησης μπορείτε να περιηγηθείτε στο

έγγραφο με μεγαλύτερη ευκολία. Υπάρχουν επικεφαλίδες - σελιδοδείκτες για τα

create, insert, stored procedures και triggers.

**Μέρος Α**

**1.A)** Παρακάτω ακολουθεί το σχεσιακό διάγραμμα της συνολικής αναθεωρημένης βάσης δεδομένων που υλοποιήσαμε, με τις απαραίτητες επεκτάσεις. Σας παραθέτουμε και το link της εφαρμογής στην οποία το φτιάξαμε, ώστε να μπορέσετε να το δείτε ακριβώς όπως είναι :

<https://lucid.app/lucidchart/97126800-b680-4261-8a6e-261898db3729/edit?invitationId=inv_d8d78a84-52d4-4c7f-9706-64349960e119>

(Λείπει ένα βελάκι στο σχεσιακό διάγραμμα. Ξεκινάει από το address\_id του customer και δείχνει στο adress\_id του address. Ανακαλύψαμε τελευταία στιγμή αυτό το λάθος και δεν προλαβαίναμε να το αλλάξουμε και να αλλάξουμε και το screenshot. Παρακαλούμε πολύ να λειφθεί υπόψην.)

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

**B)** Στη συνέχεια, παραθέτουμε τις **παραδοχές** που θεωρήσαμε :

1) Στον πίνακα category προσθέσαμε επιπλέον 3 κατηγορίες για να καλύψουμε και το εύρος των ειδών των σειρών(History, Coming of age, Mystery).

2) Για κάθε σειρά προσθέσαμε το συνολικό αριθμό επεισοδίων της και όχι τον αριθμό επεισοδίων κάθε κύκλου της. Επίσης, θεωρήσαμε ότι το έτος κυκλοφορίας αναφέρεται στον πρώτο μόνο κύκλο κάθε σειράς και όχι σε κάθε κύκλο ξεχωριστά.

3) Τροποποιήσαμε τους πίνακες rental και payment εισάγωντας ημερομηνίες από το έτος 2021 αντί του 2005-2006, ώστε να βγάζουν νόημα οι αγορές των σειρών, καθώς οι περισσότερες βγήκαν τα τελευταία 10 χρόνια.

4) Προσθέσαμε πεδίο cost\_per\_film στον πίνακα payment και πεδίο cost\_per\_series στον πίνακα series\_payment, ώστε να φαίνεται το κόστος κάθε ταινίας και σειράς.

(Επομένως, δημιουργήσαμε ξανά τον πίνακα payment.)

5) Στον trigger 4.3 θεωρήσαμε ότι το μόνο στοιχείο που θα μπορούσε να αλλάξει ο customer αλλά πρέπει να απαγορεύσουμε μέσω του trigger, είναι το customer\_id του.

6) Έχουμε προσθέσει κάποιες επιπλέον οδούς στον πίνακα address. Είναι οι οδοί στις οποίες κατοικούν οι employees.

7) Δημιουργήσαμε νέους πίνακες series\_rental και series\_payment για να διαχωρίσουμε τις ενοικιάσεις και τις πληρωμές των ταινιών από τις αντίστοιχες των σειρών.

**2)** Για τον ανασχεδιασμό της βάσης, δημιουργήσαμε τους παρακάτω πίνακες :

1) **Administrator** : Περιέχει τα βασικά στοιχεία του διαχειριστή του συστήματος.

2) **Employee** : Περιέχει τα βασικά στοιχεία των υπαλλήλων του παρόχου.

3) **Series** : Περιέχει τα στοιχεία των σειρών που προσθέσαμε. Επειδή όλες οι σειρές βγήκαν στην αγγλική γλώσσα, θεωρούμε τα αγγλικά ως default τιμή στο πεδίο language\_id. Eπιπλέον, όλες οι σειρές έχουν rating ‘pg-13’, καθώς αυτό ισχύει για την πλειοψηφία τους. Τέλος, στην τιμή κάθε επεισοδίου συμπεριλαμβάνεται όλο το επιπέον υλικό που διατίθεται, όπως trailers, deleted scenes κλπ.

4) **Seasons** : Περιέχει τον αριθμό των κύκλων κάθε σειράς.

5) **Episodes** : Περιέχει τον αριθμό και την διάρκεια κάθε επεισοδίου.

6) **Series\_category** : Συνδέονται οι σειρές με την κατάλληλη κατηγορία στην οποία ανήκουν.

7) **Series\_inventory** : Περιέχει τον κατάλογο των σειρών.

8) **Series\_actor** : Αντιστοιχίζονται οι ηθοποιοί με τις σειρές στις οποίες παίζουν.

9) **Series\_rental** : Περιέχει τα στοιχεία κάθε ενοικίασης που πραγματοποιείται.

10) **Series\_payment** : Περιέχει τα στοιχεία πληρωμής κάθε ενοικίασης.

**3)**

**3.1)** Το procedure most\_rented δεν λειτουργεί. Ωστόσο, έχουμε παραθέσει τον κώδικα του.

**3.2)**

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, ηλεκτρονικές συσκευές, υπολογιστής

Περιγραφή που δημιουργήθηκε αυτόματα

Εδώ βλέπουμε ένα παράδειγμα της κλήσης της stored procedure **rentals**. Αρχικά, εμφανίζεται στην οθόνη το id του πελάτη και ο αριθμός των ενοικιάσεων ταινιών που έκανε την ημερομηνία που δώσαμε σαν όρισμα και μετά ο αριθμός των ενοικιάσεων σειρών την ίδια ημερομηνία. Χρησιμοποιήσαμε την συνάρτηση DATE της Sql, η οποία μετατρέπει σε date οποιοδήποτε ημερομηνία σε μορφή datetime της δωθεί.

**3.3)**

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, ηλεκτρονικές συσκευές, υπολογιστής

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, ηλεκτρονικές συσκευές

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, ηλεκτρονικές συσκευές

Περιγραφή που δημιουργήθηκε αυτόματα

Εδώ βλέπουμε ένα παράδειγμα της κλήσης της stored procedure **earnings**. Καλώντας τη, εμφανίζονται τα έσοδα από πληρωμές ταινίων και σειρών ξεχωριστά ανά μήνα.

**3.4.α)**

Εικόνα που περιέχει κείμενο, ηλεκτρονικές συσκευές, στιγμιότυπο οθόνης, εμφάνιση

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, ηλεκτρονικές συσκευές, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, ηλεκτρονικές συσκευές, στιγμιότυπο οθόνης, εμφάνιση

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, ηλεκτρονικές συσκευές, στιγμιότυπο οθόνης, εμφάνιση

Περιγραφή που δημιουργήθηκε αυτόματα

Εδώ βλέπουμε ένα παράδειγμα της κλήσης της stored procedure **lastnames**. Εμφανίζει στην οθόνη το πλήθος και τα ονοματεπώνυμα των ηθοποιών που το ονοματεπώνυμο τους βρίσκεται ανάμεσα στα ορίσματα εισόδου.

**3.4.β)**

Εικόνα που περιέχει κείμενο, ηλεκτρονικές συσκευές, στιγμιότυπο οθόνης, υπολογιστής

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, ηλεκτρονικές συσκευές, εμφάνιση

Περιγραφή που δημιουργήθηκε αυτόματα

**4)**

**4.2)** O trigger sale δεν λειτουργεί. Ωστόσο, έχουμε παραθέσει τον κώδικα του.

**4.3)**

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, ηλεκτρονικές συσκευές

Περιγραφή που δημιουργήθηκε αυτόματα

Εδώ βλέπουμε ένα παράδειγμα εφαρμογής του trigger **not\_allowed**. O trigger ενεργοποιείται όταν προσπαθούμε να ενημερώσουμε το id του πελάτη, κάτι που δεν γίνεται. Για αυτό τον λόγο, εμφανίζεται κατάλληλο μήνυμα λάθους.

Στη συνέχεια ακολουθούν οι κώδικες create, insert, stored procedures και triggers :

# CREATE

CREATE TABLE payment (

payment\_id SMALLINT(5) UNSIGNED NOT NULL AUTO\_INCREMENT,

customer\_id SMALLINT(5) UNSIGNED NOT NULL,

rental\_id INT(11) DEFAULT NULL,

amount DECIMAL(5,2) NOT NULL,

payment\_date DATETIME NOT NULL,

cost\_per\_film VARCHAR(10) NOT NULL DEFAULT ('0.4');

PRIMARY KEY (payment\_id),

CONSTRAINT fk\_payment\_rental FOREIGN KEY (rental\_id) REFERENCES rental (rental\_id) ON DELETE SET NULL ON UPDATE CASCADE,

CONSTRAINT fk\_payment\_customer FOREIGN KEY (customer\_id) REFERENCES customer (customer\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE employee(

employee\_id SMALLINT(5) UNSIGNED NOT NULL AUTO\_INCREMENT,

first\_name VARCHAR(45) NOT NULL,

last\_name VARCHAR(45) NOT NULL,

email VARCHAR(50) DEFAULT NULL,

address\_id SMALLINT(5) UNSIGNED NOT NULL,

PRIMARY KEY (employee\_id),

CONSTRAINT fk\_employee\_address FOREIGN KEY (address\_id) REFERENCES address (address\_id) ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE administrator(

admin\_id SMALLINT(5) UNSIGNED NOT NULL AUTO\_INCREMENT,

first\_name VARCHAR(45) NOT NULL,

last\_name VARCHAR(45) NOT NULL,

email VARCHAR(50) DEFAULT NULL,

PRIMARY KEY (admin\_id)

);

CREATE TABLE series(

series\_id SMALLINT(5) UNSIGNED NOT NULL AUTO\_INCREMENT,

title VARCHAR(128) NOT NULL,

description TEXT DEFAULT NULL,

release\_year YEAR(4) DEFAULT NULL,

language\_id TINYINT(3) UNSIGNED NOT NULL DEFAULT '1',

original\_language\_id TINYINT(3) UNSIGNED DEFAULT NULL,

rating ENUM('G','PG','PG-13','R','NC-17') DEFAULT 'PG-13',

special\_features SET('Trailers','Commentaries','Deleted Scenes','Behind the Scenes', 'All Included') DEFAULT 'All Included',

PRIMARY KEY (series\_id),

CONSTRAINT fk\_series\_language FOREIGN KEY (language\_id) REFERENCES language (language\_id) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT fk\_series\_language\_original FOREIGN KEY (original\_language\_id) REFERENCES language (language\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE seasons(

seasons\_id SMALLINT(5) UNSIGNED NOT NULL AUTO\_INCREMENT,

series\_id SMALLINT(5) UNSIGNED NOT NULL,

number\_of\_seasons SMALLINT(5) UNSIGNED NOT NULL,

PRIMARY KEY(seasons\_id),

CONSTRAINT fk\_series\_id FOREIGN KEY (series\_id) REFERENCES series (series\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE episodes(

episodes\_id SMALLINT(5) UNSIGNED NOT NULL AUTO\_INCREMENT,

series\_id SMALLINT(5) UNSIGNED NOT NULL,

number\_of\_episodes SMALLINT(5) UNSIGNED NOT NULL,

length SMALLINT(5) UNSIGNED DEFAULT NULL,

PRIMARY KEY(episodes\_id),

CONSTRAINT fk\_serepisodes\_id FOREIGN KEY (series\_id) REFERENCES series (series\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE series\_category(

series\_id SMALLINT(5) UNSIGNED NOT NULL,

category\_id TINYINT(3) UNSIGNED NOT NULL,

PRIMARY KEY (series\_id, category\_id),

CONSTRAINT fk\_series\_category\_film FOREIGN KEY (series\_id) REFERENCES series (series\_id) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT fk\_series\_category\_category FOREIGN KEY (category\_id) REFERENCES category (category\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE series\_actor(

actor\_id SMALLINT(5) UNSIGNED NOT NULL,

series\_id SMALLINT(5) UNSIGNED NOT NULL,

PRIMARY KEY (actor\_id,series\_id),

CONSTRAINT fk\_series\_actor\_actor FOREIGN KEY (actor\_id) REFERENCES actor (actor\_id) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT fk\_series\_actor\_series FOREIGN KEY (series\_id) REFERENCES series (series\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE series\_inventory (

series\_inventory\_id MEDIUMINT(8) UNSIGNED NOT NULL AUTO\_INCREMENT,

series\_id SMALLINT(5) UNSIGNED NOT NULL,

PRIMARY KEY (series\_inventory\_id),

CONSTRAINT fk\_inventory\_series FOREIGN KEY (series\_id) REFERENCES series (series\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE series\_rental (

series\_rental\_id INT(11) NOT NULL AUTO\_INCREMENT,

rental\_date DATETIME NOT NULL,

series\_inventory\_id MEDIUMINT(8) UNSIGNED NOT NULL,

customer\_id SMALLINT(5) UNSIGNED NOT NULL,

PRIMARY KEY (series\_rental\_id),

UNIQUE KEY (rental\_date,series\_inventory\_id,customer\_id),

CONSTRAINT fk\_series\_rental\_inventory FOREIGN KEY (series\_inventory\_id) REFERENCES series\_inventory (series\_inventory\_id) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT fk\_series\_rental\_customer FOREIGN KEY (customer\_id) REFERENCES customer (customer\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

CREATE TABLE series\_payment (

series\_payment\_id SMALLINT(5) UNSIGNED NOT NULL AUTO\_INCREMENT,

customer\_id SMALLINT(5) UNSIGNED NOT NULL,

series\_rental\_id INT(11) DEFAULT NULL,

series\_amount DECIMAL(5,2) NOT NULL,

series\_payment\_date DATETIME NOT NULL,

cost\_per\_series VARCHAR(10) NOT NULL DEFAULT '0.2',

PRIMARY KEY (series\_payment\_id),

CONSTRAINT fk\_series\_payment\_rental FOREIGN KEY (series\_rental\_id) REFERENCES series\_rental (series\_rental\_id) ON DELETE RESTRICT ON UPDATE CASCADE,

CONSTRAINT fk\_series\_payment\_customer FOREIGN KEY (customer\_id) REFERENCES customer (customer\_id) ON DELETE RESTRICT ON UPDATE CASCADE

);

# INSERT

INSERT INTO `employee` (`employee\_id`, `first\_name`, `last\_name`, `email`, `address\_id`) VALUES

(324, 'CRISTIANO', 'RONALDO', 'RONALDO@sakilacustomer.org', 237),

(312, 'HUDSON', 'ODOI', 'ODOI@sakilaemployee.org', 45),

(23, 'KAI', 'HAVERTZ', 'HAVERTZ@sakilaemployee.org', 36),

(45, 'TIMO', 'WERNER', 'WERNER@sakilacemployee.org', 70),

(897, 'LIONEL', 'MESSI', 'MESSI@sakilaemployee.org', 68),

(765, 'PHIL', 'JONES', 'JONES@sakilaemployee.org', 44),

(98, 'NEYMAR', 'JR', 'NEYMAR@sakilaemployee.org', 25),

(225, 'KYLIAN', 'MBAPE', 'MBAPE@sakilaemployee.org', 21),

(345, 'LUIS', 'SUAREZ', 'SUAREZ@sakilaemployee.org', 9),

(786, 'GERARD', 'PIQUE', 'PIQUE@sakilaemployee.org', 32),

(896, 'ANTONIO', 'GARCIA', 'GARCIA@sakilaemployee.org', 86),

(666, 'USUF', 'ELARRABI', 'USUF@sakilaemployee.org', 49),

(222, 'RONI', 'LOPEZ', 'LOPEZ@sakilaemployee.org', 30),

(69, 'MADI', 'CAMMARA', 'CAMMARA@sakilaemployee.org', 12),

(55, 'VIRGIL', 'VAN DIJK', 'VIRGIL@sakilaemployee.org', 10),

(124, 'PHIL', 'COUTINHO', 'PHIL@sakilaemployee.org', 8),

(420, 'STEVEN', 'GERARD', 'GERARD@sakilaemployee.org', 422);

INSERT INTO `administrator` (`admin\_id`, `first\_name`, `last\_name`, `email`) VALUES

(1, 'CARLO', 'ANCELOTTI', 'ANCELOTTI@admin.org'),

(2, 'XAVI', 'HERNANDES', 'HERNANDES@admin.org');

INSERT INTO `address` (`address\_id`, `address`, `district`, `city\_id`, `postal\_code`, `phone`) VALUES

(237, 'Gianni Ritsou 70', 'Koukouli', 401, '26335', '66666666666'),

(45, 'High Street 45', 'Yorkshire', 17, '26336', '111111111'),

(36, 'Station road 44', 'Merseyside', 38, '26337', '222222222'),

(70, 'Main Street 43', 'Cornwall', 70, '26338', '3333333333'),

(68, 'Park Road 42', 'Bristall', 88, '26339', '77777777'),

(44, 'Church road 41', 'Dorset', 91, '26340', '5555555555'),

(25, 'Church Street 67', 'Midlands', 92, '26341', '8888888888'),

(21, 'London Road 12', 'Cheshire', 93, '26342', '343433847437'),

(9, 'Victoria Road 89', 'Borrow', 146, '26343', '91983917389'),

(32, 'Green Lane 89', 'Kent', 148, '26344', '2372323783'),

(86, 'Meinour Road 76', 'Sommerset', 160, '26345', '983478984432'),

(49, 'Park Avenue 88', 'Esex', 149, '26346', '93472984634'),

(30, 'Queens Road 43', 'Oxford', 181, '26347', '829479824782893'),

(12, 'Kings Road 44', 'Devon', 297, '26348', '98348723448'),

(10, 'Kings Way 90', 'York', 198, '26349', '3093747438347'),

(8, 'Alexander Road 70', 'Suffolk', 439, '26350', '29834728347'),

(422, 'Broadway 70', 'Hamp', 496, '26351', '329048394834');

INSERT INTO `series` (`series\_id`, `title`, `description`, `release\_year`) VALUES

(100, 'FLASH', 'After being struck by lightning Barry Allen wakes up from his coma to discover he is been given the power of super speed', 2014),

(222, 'PRISON BREAK', 'An innocent man is sent to death row and his only hope is his brother', 2005),

(98, 'EUPHORIA', 'Discovering the life of Rue between family and drugs', 2019),

(11, 'THE OFFICE', 'The documentary of the life of a group of typical office workers', 2005),

(467, 'TED LASSO', 'American college football coach Ted Lasso heads to London to manage AFC Richmond', 2020),

(768, 'BREAKING BAD', 'A high school chemistry teacher selling methamphetamine in order to secure his familys future', 2008),

(254, 'OZARK', 'A financial advisor must launder money to appease a drug boss', 2017),

(44, 'VIKINGS', 'Ragnar Lothbrok a Viking warrior and farmer who yearns to explore - and raid - the distant shores across the ocean', 2013),

(99, 'DAREDEVIL', 'Matt Murdock fights the crime of New York as Daredevil', 2015),

(91, 'SOPRANOS', 'Tony Soprano deals with personal and professional issues in his home and business life', 1999),

(562, 'MONEY HEIST', 'An unusual group of robbers attempt to carry out the most perfect robbery in Spanish history', 2017),

(832, 'STRANGER THINGS', 'When a young boy disappears, his mother and his friends must confront terrifying supernatural forces in order to get him back', 2016),

(823, 'FRIENDS', 'Follows the personal and professional lives of six twenty year old friends living in Manhattan', 1994),

(1, 'THE WIRE', 'The Baltimore drug scene as seen through the eyes of drug dealers and law enforcement', 2002),

(34, 'TOPBOY', 'Two London drug dealers ply their lucrative trade at a public housing estate in East London', 2011),

(434, 'THE WITCHER', 'Geralt of Rivia a solitary monster hunter, struggles to find his place in the world', 2019),

(278, 'BETTER CALL SAUL', 'The trials and tribulations of criminal lawyer Jimmy McGill', 2015),

(209, 'SQUID GAME', 'A survival game that has a whopping 45.6 billion-won prize at stake', 2021),

(88, 'HOW I MET YOUT MOTHER', 'A father recounts to his children the journey he and his four best friends took leading up to him meeting their mother', 2005),

(111, 'DOWNTON ABBEY', 'A chronicle of the lives of the British aristocratic Crawley family', 2010),

(892, 'OUTLANDER', 'An English combat nurse from 1945 is mysteriously swept back in time to 1743', 2014),

(70, 'YOU', 'A dangerously charming intensely obsessive young man goes to extreme measures', 2018),

(45, 'SEX EDUCATION', 'A teenage boy with a sex therapist mother teams up with a high school classmate to set up an underground sex therapy clinic at school', 2019),

(13, 'ELITE', 'When three working-class teenagers begin attending an exclusive private school in Spain', 2018),

(801, 'LUCIFER', 'Lucifer Morningstar has decided hes had enough of being the dutiful servant in Hell and decides to spend some time on Earth', 2016),

(989, 'RIVERDALE', 'Archie and his gang become entangled in dark Riverdale mysteries', 2017),

(999, 'TRUE DETECTIVE', 'Seasonal anthology series in which police investigations unearth the personal and professional secrets of those involved', 2014),

(301, 'THE LAST KINGDOM', 'Uhtred--born a Saxon but raised by Vikings--seeks to claim his ancestral birthright', 2015),

(404, 'OUTER BANKS', 'A group of teenagers stumble upon a treasure map that unearths a long buried secret', 2020),

(505, 'MR ROBOT', 'Elliot a brilliant young cyber-security engineer hacker becomes a key figure in a complex game of global dominance', 2015),

(660, 'KOBRA KAI', 'A middle-aged Daniel LaRusso and Johnny Lawrence find themselves martial-arts rivals again', 2018),

(431, 'NARCOS', 'A chronicled look at the criminal exploits of Colombian drug lord Pablo Escobar', 2018),

(188, 'SHERLOCK', 'A modern update finds the famous sleuth and his doctor partner solving crime in 21st century London', 2010);

INSERT INTO `seasons` (`series\_id`, `number\_of\_seasons`) VALUES

(100, 8),

(222, 5),

(98, 2),

(11, 9),

(467, 2),

(768, 5),

(254, 4),

(44, 6),

(99, 3),

(91, 6),

(562, 5),

(832, 4),

(823, 10),

(1, 5),

(34, 3),

(434, 2),

(278, 5),

(209, 1),

(88, 9),

(111, 6),

(892, 6),

(70, 3),

(45, 3),

(13, 6),

(801, 6),

(989, 6),

(999, 3),

(301, 5),

(404, 3),

(505, 4),

(660, 4),

(431, 3),

(188, 4);

INSERT INTO `episodes` (`series\_id`, `number\_of\_episodes`, `length`) VALUES

(100, 169, 43),

(222, 90, 44),

(98, 18, 55),

(11, 188, 22),

(467, 23, 30),

(768, 62, 49),

(254, 44, 60),

(44, 93, 44),

(99, 39, 54),

(91, 86, 55),

(562, 41, 70),

(832, 34, 51),

(823, 235, 22),

(1, 60, 59),

(34, 19, 60),

(434, 17, 60),

(278, 63, 46),

(209, 10, 60),

(88, 208, 20),

(111, 52, 58),

(892, 87, 60),

(70, 31, 45),

(45, 25, 45),

(13, 41, 60),

(801, 93, 42),

(989, 101, 45),

(999, 24, 55),

(301, 46, 60),

(404, 21, 50),

(505, 45, 55),

(660, 41, 30),

(431, 30, 49),

(188, 15, 88);

INSERT INTO `series\_category` (`series\_id`, `category\_id`) VALUES

(100, 14),

(222, 1),

(98, 7),

(11, 5),

(467, 15),

(768, 7),

(254, 7),

(44, 17),

(99, 14),

(91, 7),

(562, 1),

(832, 14),

(823, 5),

(1, 6),

(34, 1),

(434, 14),

(278, 7),

(209, 10),

(88, 5),

(111, 4),

(892, 14),

(70, 7),

(45, 18),

(13, 18),

(801, 1),

(989, 18),

(999, 19),

(301, 17),

(404, 1),

(505, 7),

(660, 5),

(431, 6),

(188, 19);

INSERT INTO `category` (`category\_id`, `name`) VALUES

(17, 'History'),

(18, 'Coming of age'),

(19, 'Mystery');

INSERT INTO `series\_actor` (`actor\_id`, `series\_id`) VALUES

(666, 1),

(667, 1),

(668, 1),

(669, 11),

(1010, 11),

(2929, 11),

(267, 13),

(9811, 13),

(2234, 13),

(7676, 13),

(111, 34),

(9999, 34),

(1243, 34),

(12, 44),

(1232, 44),

(8898, 44),

(7328, 45),

(1092, 45),

(9000, 45),

(1117, 70),

(8922, 70),

(88, 70),

(45, 88),

(9191, 88),

(655, 88),

(911, 91),

(1211, 91),

(7621, 91),

(1940, 98),

(1821, 98),

(1944, 98),

(2567, 99),

(2109, 99),

(4000, 99),

(4004, 100),

(4008, 100),

(6472, 100),

(2001, 111),

(2002, 111),

(2004, 111),

(4488, 188),

(9911, 188),

(9100, 188),

(5881, 209),

(9999, 209),

(10000, 209),

(1844, 222),

(5009, 222),

(6666, 222),

(6060, 254),

(4040, 254),

(6080, 254),

(9001, 278),

(10003, 278),

(9076, 278),

(9871, 301),

(1230, 301),

(7007, 301),

(8008, 404),

(9009, 404),

(5005, 404),

(413, 431),

(588, 431),

(619, 431),

(620, 434),

(121, 434),

(999, 434),

(1828, 467),

(9991, 467),

(1122, 467),

(4455, 505),

(667, 505),

(771, 505),

(4499, 562),

(9912, 562),

(9101, 562),

(5621, 660),

(8322, 660),

(8233, 660),

(5343, 768),

(3458, 768),

(4367, 768),

(278, 801),

(131, 801),

(69, 801),

(4569, 823),

(3490, 823),

(6709, 823),

(9093, 832),

(21, 832),

(10004, 832),

(9841, 892),

(8491, 892),

(1109, 892),

(1377, 989),

(6003, 989),

(6004, 989),

(6006, 999),

(1923, 999),

(9891, 999);

INSERT INTO series\_inventory (series\_id) VALUES

(1),

(11),

(13),

(34),

(44),

(45),

(70),

(88),

(91),

(98),

(99),

(100),

(111),

(188),

(209),

(222),

(254),

(278),

(301),

(404),

(431),

(434),

(467),

(505),

(562),

(660),

(768),

(801),

(823),

(832),

(892),

(989),

(999);

INSERT INTO `rental` (`rental\_id`, `rental\_date`, `inventory\_id`, `customer\_id`) VALUES

(1194, '2021-06-15 01:25:08', 3774, 402),

(2913, '2021-06-20 03:42:27', 3456, 402),

(6283, '2021-07-11 16:47:32', 3319, 402),

(11920, '2021-08-17 16:10:19', 4253, 402),

(10093, '2021-07-31 20:30:32', 3943, 162),

(1287, '2021-06-15 08:41:38', 4204, 104),

(11700, '2021-08-17 07:12:31', 2365, 104),

(15186, '2021-08-22 15:52:57', 1323, 104),

(424, '2021-05-27 15:34:01', 2815, 35),

(889, '2021-05-30 07:14:53', 1299, 114),

(3924, '2021-07-06 20:38:02', 187, 114),

(7462, '2021-07-27 18:47:47', 878, 114),

(15528, '2021-08-23 03:45:40', 3033, 114),

(1313, '2021-06-15 10:18:34', 1657, 448),

(1823, '2021-06-16 21:48:16', 3871, 448),

(4206, '2021-07-07 11:32:16', 1557, 448),

(6985, '2021-07-27 00:57:42', 1834, 448),

(11798, '2021-08-17 11:21:43', 3974, 448),

(13250, '2021-08-19 16:47:55', 2524, 448),

(9196, '2021-07-30 12:30:19', 4298, 251),

(12666, '2021-08-18 19:11:41', 4156, 251),

(13121, '2021-08-19 11:51:39', 3517, 251),

(1182, '2021-06-15 00:45:21', 1061, 196),

(3342, '2021-06-21 10:46:36', 2884, 196),

(9721, '2021-07-31 08:28:46', 3942, 196),

(12684, '2021-08-18 19:51:27', 978, 196),

(13799, '2021-08-20 12:36:42', 4413, 196),

(7132, '2021-07-27 06:28:34', 3201, 227),

(8417, '2021-07-29 06:53:36', 750, 227),

(12099, '2021-08-17 22:38:54', 1743, 195),

(16040, '2021-08-23 22:19:33', 3524, 195),

(756, '2021-05-29 10:28:45', 3152, 201),

(2047, '2021-06-17 14:40:58', 2371, 201),

(14656, '2021-08-21 19:39:28', 4205, 201),

(6197, '2021-07-11 12:09:51', 4156, 596),

(14417, '2021-08-21 11:13:35', 1768, 596),

(4095, '2021-07-07 06:01:48', 2900, 33),

(13958, '2021-08-20 18:11:44', 2380, 33),

(14623, '2021-08-21 18:29:13', 901, 33),

(2621, '2021-06-19 08:07:31', 1746, 439),

(11042, '2021-08-02 06:04:33', 4149, 439),

(4172, '2021-07-07 09:49:09', 1158, 309),

(4540, '2021-07-08 04:03:28', 4160, 309),

(11964, '2021-08-17 17:37:03', 2225, 309),

(13502, '2021-08-20 01:58:15', 1324, 309),

(14846, '2021-08-22 02:13:48', 3840, 309),

(15422, '2021-08-22 23:58:09', 2601, 309),

(1355, '2021-06-15 13:13:59', 4203, 499),

(4693, '2021-07-08 11:07:36', 1694, 221),

(7531, '2021-07-27 21:19:34', 2393, 221),

(11693, '2021-08-17 06:56:56', 793, 221),

(11802, '2021-08-17 11:32:51', 2815, 221),

(15125, '2021-08-22 12:53:22', 3942, 221),

(3993, '2021-07-06 23:37:06', 1061, 525),

(874, '2021-05-30 05:36:21', 363, 52),

(2862, '2021-06-19 23:47:24', 4260, 52),

(7627, '2021-07-28 00:56:47', 782, 52),

(8686, '2021-07-29 16:17:49', 4160, 52),

(9029, '2021-07-30 06:03:11', 2220, 52),

(12001, '2021-02-14 15:16:03', 4158, 52),

(3025, '2021-06-20 11:46:48', 1309, 293),

(15258, '2021-08-22 18:22:44', 788, 293),

(213, '2021-05-26 08:44:08', 1505, 394),

(6059, '2021-07-11 04:03:54', 1437, 394),

(506, '2021-05-28 02:09:19', 667, 469),

(936, '2021-05-30 13:52:49', 901, 469),

(3526, '2021-07-06 01:03:29', 3871, 469),

(5664, '2021-07-10 08:04:41', 2375, 469),

(8316, '2021-07-29 03:38:49', 4118, 549),

(148, '2021-05-26 00:25:23', 4252, 142),

(3492, '2021-07-05 23:44:37', 980, 142),

(4497, '2021-07-08 01:51:32', 1995, 142),

(4372, '2021-07-07 20:09:01', 2207, 252),

(7168, '2021-07-27 07:51:11', 6, 252),

(2029, '2021-06-17 13:10:59', 914, 512),

(12786, '2021-02-14 15:16:03', 97, 512),

(5462, '2021-07-09 22:56:53', 871, 583),

(10800, '2021-08-01 22:07:44', 507, 583),

(8452, '2021-07-29 07:45:00', 4312, 16),

(772, '2021-05-29 13:08:06', 3319, 556),

(1083, '2021-05-31 11:04:48', 2393, 556),

(2986, '2021-06-20 08:50:28', 478, 556),

(14176, '2021-08-21 03:09:23', 4420, 556),

(882, '2021-05-30 06:16:06', 3841, 477),

(2676, '2021-06-19 11:54:57', 3938, 477),

(4237, '2021-07-07 13:16:55', 3773, 477),

(4283, '2021-07-07 15:29:35', 2807, 477),

(15091, '2021-08-22 11:34:43', 2092, 477),

(2818, '2021-06-19 20:05:52', 2376, 497),

(3696, '2021-07-06 10:04:55', 3244, 497),

(4218, '2021-07-07 12:10:24', 152, 497),

(12123, '2021-08-17 23:22:18', 361, 497),

(12698, '2021-02-14 15:16:03', 3657, 497),

(15919, '2021-08-23 18:01:31', 1746, 497);

INSERT INTO `payment` (`payment\_id`, `customer\_id`, `rental\_id`, `amount`, `payment\_date`) VALUES

(10857, 402, 1194, '4.99', '2021-06-15 01:25:08'),

(10859, 402, 2913, '2.99', '2021-06-20 03:42:27'),

(10867, 402, 6283, '3.99', '2021-07-11 16:47:32'),

(10874, 402, 11920, '0.99', '2021-08-17 16:10:19'),

(4416, 162, 10093, '5.99', '2021-07-31 20:30:32'),

(2802, 104, 1287, '3.99', '2021-06-15 08:41:38'),

(2816, 104, 11700, '2.99', '2021-08-17 07:12:31'),

(2823, 104, 15186, '0.99', '2021-08-22 15:52:57'),

(953, 35, 424, '6.99', '2021-05-27 15:34:01'),

(3060, 114, 889, '2.99', '2021-05-30 07:14:53'),

(3066, 114, 3924, '2.99', '2021-07-06 20:38:02'),

(3074, 114, 7462, '2.99', '2021-07-27 18:47:47'),

(3087, 114, 15528, '2.99', '2021-08-23 03:45:40'),

(12087, 448, 1313, '5.99', '2021-06-15 10:18:34'),

(12088, 448, 1823, '7.99', '2021-06-16 21:48:16'),

(12095, 448, 4206, '2.99', '2021-07-07 11:32:16'),

(12100, 448, 6985, '4.99', '2021-07-27 00:57:42'),

(12103, 448, 11798, '9.99', '2021-08-17 11:21:43'),

(12106, 448, 13250, '3.99', '2021-08-19 16:47:55'),

(6775, 251, 9196, '2.99', '2021-07-30 12:30:19'),

(6780, 251, 12666, '4.99', '2021-08-18 19:11:41'),

(6781, 251, 13121, '2.99', '2021-08-19 11:51:39'),

(5285, 196, 1182, '5.99', '2021-06-15 00:45:21'),

(5292, 196, 3342, '4.99', '2021-06-21 10:46:36'),

(5304, 196, 9721, '6.99', '2021-07-31 08:28:46'),

(5310, 196, 12684, '0.99', '2021-08-18 19:51:27'),

(5312, 196, 13799, '8.99', '2021-08-20 12:36:42'),

(6121, 227, 7132, '7.99', '2021-07-27 06:28:34'),

(6125, 227, 8417, '4.99', '2021-07-29 06:53:36'),

(5276, 195, 12099, '0.99', '2021-08-17 22:38:54'),

(5280, 195, 16040, '11.99', '2021-08-23 22:19:33'),

(5447, 201, 756, '5.99', '2021-05-29 10:28:45'),

(5448, 201, 2047, '1.99', '2021-06-17 14:40:58'),

(5466, 201, 14656, '2.99', '2021-08-21 19:39:28'),

(15967, 596, 6197, '4.99', '2021-07-11 12:09:51'),

(15980, 596, 14417, '0.99', '2021-08-21 11:13:35'),

(910, 33, 4095, '5.99', '2021-07-07 06:01:48'),

(922, 33, 13958, '7.99', '2021-08-20 18:11:44'),

(924, 33, 14623, '0.99', '2021-08-21 18:29:13'),

(11835, 439, 2621, '2.99', '2021-06-19 08:07:31'),

(11852, 439, 11042, '6.99', '2021-08-02 06:04:33'),

(8389, 309, 4172, '4.99', '2021-07-07 09:49:09'),

(8390, 309, 4540, '4.99', '2021-07-08 04:03:28'),

(8406, 309, 11964, '6.99', '2021-08-17 17:37:03'),

(8408, 309, 13502, '0.99', '2021-08-20 01:58:15'),

(8410, 309, 14846, '5.99', '2021-08-22 02:13:48'),

(8411, 309, 15422, '4.99', '2021-08-22 23:58:09'),

(13433, 499, 1355, '2.99', '2021-06-15 13:13:59'),

(5983, 221, 4693, '6.99', '2021-07-08 11:07:36'),

(5988, 221, 7531, '8.99', '2021-07-27 21:19:34'),

(5994, 221, 11693, '4.99', '2021-08-17 06:56:56'),

(5995, 221, 11802, '2.99', '2021-08-17 11:32:51'),

(6002, 221, 15125, '8.99', '2021-08-22 12:53:22'),

(14119, 525, 3993, '6.99', '2021-07-06 23:37:06'),

(1425, 52, 874, '0.99', '2021-05-30 05:36:21'),

(1428, 52, 2862, '2.99', '2021-06-19 23:47:24'),

(1437, 52, 7627, '4.99', '2021-07-28 00:56:47'),

(1438, 52, 8686, '4.99', '2021-07-29 16:17:49'),

(1439, 52, 9029, '4.99', '2021-07-30 06:03:11'),

(1453, 52, 12001, '4.99', '2021-02-14 15:16:03'),

(7929, 293, 3025, '2.99', '2021-06-20 11:46:48'),

(7949, 293, 15258, '4.99', '2021-08-22 18:22:44'),

(10672, 394, 213, '3.99', '2021-05-26 08:44:08'),

(10681, 394, 6059, '4.99', '2021-07-11 04:03:54'),

(12651, 469, 506, '7.99', '2021-05-28 02:09:19'),

(12653, 469, 936, '1.99', '2021-05-30 13:52:49'),

(12658, 469, 3526, '10.99', '2021-07-06 01:03:29'),

(12663, 469, 5664, '6.99', '2021-07-10 08:04:41'),

(14738, 549, 8316, '4.99', '2021-07-29 03:38:49'),

(3832, 142, 148, '0.99', '2021-05-26 00:25:23'),

(3836, 142, 3492, '2.99', '2021-07-05 23:44:37'),

(3837, 142, 4497, '4.99', '2021-07-08 01:51:32'),

(6793, 252, 4372, '0.99', '2021-07-07 20:09:01'),

(6799, 252, 7168, '0.99', '2021-07-27 07:51:11'),

(13782, 512, 2029, '4.99', '2021-06-17 13:10:59'),

(13806, 512, 12786, '0.99', '2021-02-14 15:16:03'),

(15623, 583, 5462, '0.99', '2021-07-09 22:56:53'),

(15633, 583, 10800, '4.99', '2021-08-01 22:07:44'),

(434, 16, 8452, '5.99', '2021-07-29 07:45:00'),

(14891, 556, 772, '5.99', '2021-05-29 13:08:06'),

(14892, 556, 1083, '3.99', '2021-05-31 11:04:48'),

(14895, 556, 2986, '0.99', '2021-06-20 08:50:28'),

(14909, 556, 14176, '2.99', '2021-08-21 03:09:23'),

(12885, 477, 882, '2.99', '2021-05-30 06:16:06'),

(12889, 477, 2676, '4.99', '2021-06-19 11:54:57'),

(12890, 477, 4237, '5.99', '2021-07-07 13:16:55'),

(12891, 477, 4283, '2.99', '2021-07-07 15:29:35'),

(12905, 477, 15091, '2.99', '2021-08-22 11:34:43'),

(13381, 497, 2818, '4.99', '2021-06-19 20:05:52'),

(13382, 497, 3696, '2.99', '2021-07-06 10:04:55'),

(13383, 497, 4218, '7.99', '2021-07-07 12:10:24'),

(13397, 497, 12123, '0.99', '2021-08-17 23:22:18'),

(13404, 497, 12698, '4.99', '2021-02-14 15:16:03'),

(13403, 497, 15919, '0.99', '2021-08-23 18:01:31');

INSERT INTO `series\_rental` ( `rental\_date`, `series\_inventory\_id`, `customer\_id`) VALUES

('2021-04-21 03:42:27', 1, 3),

('2021-04-11 16:47:32', 1, 3),

('2021-04-17 16:10:19', 2, 16),

('2021-04-15 08:41:38', 2, 16),

('2021-04-17 07:12:31', 3, 33),

('2021-04-22 15:52:57', 3, 35),

('2021-05-27 15:34:01', 3, 35),

('2021-05-30 07:14:53', 4, 35),

('2021-05-06 20:38:02', 4, 52),

('2021-05-27 18:47:47', 4, 52),

('2021-05-23 03:45:40', 5, 85),

('2021-05-15 10:18:34', 5, 85),

('2021-05-16 21:48:16', 5, 104),

('2021-05-07 11:32:16', 6, 104),

('2021-06-27 00:57:42', 6, 104),

('2021-06-17 11:21:43', 6, 114),

('2021-06-19 16:47:55', 7, 142),

('2021-06-30 12:30:19', 7, 162),

('2021-06-18 19:11:41', 7, 162),

('2021-06-19 11:51:39', 8, 162),

('2021-06-15 00:45:21', 8, 162),

('2021-06-21 10:46:36', 8, 195),

('2021-07-31 08:28:46', 9, 195),

('2021-07-18 19:51:27', 9, 195),

('2021-07-20 12:36:42', 9, 195),

('2021-07-27 06:28:34', 10, 195),

('2021-07-29 06:53:36', 10, 196),

('2021-07-17 22:38:54', 10, 196),

('2021-07-23 22:19:33', 11, 196),

('2021-07-29 10:28:45', 11, 196),

('2021-08-17 14:40:58', 11, 196),

('2021-08-21 19:39:28', 12, 201),

('2021-08-11 12:09:51', 12, 201),

('2021-08-21 11:13:35', 12, 201),

('2021-08-07 06:01:48', 13, 201),

('2021-08-20 18:11:44', 13, 201),

('2021-08-21 18:29:13', 13, 201),

('2021-08-19 08:07:31', 14, 221),

('2021-09-02 06:04:33', 14, 221),

('2021-09-07 09:49:09', 15, 221),

('2021-09-08 04:03:28', 16, 221),

('2021-09-17 17:37:03', 17, 227),

('2021-09-20 01:58:15', 17, 227),

('2021-09-22 02:13:48', 18, 227),

('2021-09-22 23:58:09', 18, 227),

('2021-09-15 13:13:59', 18, 227),

('2021-10-08 11:07:36', 19, 251),

('2021-10-27 21:19:34', 19, 251),

('2021-10-17 06:56:56', 19, 251),

('2021-10-17 11:32:51', 19, 251),

('2021-10-22 12:53:22', 20, 251),

('2021-10-06 23:37:06', 20, 252),

('2021-10-30 05:36:21', 20, 252),

('2021-10-19 23:47:24', 21, 252),

('2021-11-28 00:56:47', 21, 252),

('2021-11-29 16:17:49', 21, 252),

('2021-11-30 06:03:11', 22, 252),

('2021-11-14 15:16:03', 22, 293),

('2021-11-20 11:46:48', 22, 293),

('2021-11-22 18:22:44', 23, 293),

('2021-11-26 08:44:08', 23, 293),

('2021-11-11 04:03:54', 23, 293),

('2021-12-28 02:09:19', 24, 293),

('2021-12-30 13:52:49', 24, 3),

('2021-12-06 01:03:29', 24, 3),

('2021-12-10 08:04:41', 25, 3),

('2021-12-29 03:38:49', 25, 16),

('2021-12-26 00:25:23', 25, 16),

('2021-12-05 23:44:37', 26, 33),

('2021-12-08 01:51:32', 26, 33),

('2021-04-07 20:09:01', 26, 35),

('2021-05-27 07:51:11', 27, 35),

('2021-06-17 13:10:59', 27, 52),

('2021-07-14 15:16:03', 27, 52),

('2021-08-09 22:56:53', 28, 52),

('2021-09-01 22:07:44', 28, 85),

('2021-10-29 07:45:00', 28, 85),

('2021-11-29 13:08:06', 29, 85),

('2021-12-31 11:04:48', 29, 596),

('2021-04-20 08:50:28', 29, 596),

('2021-05-21 03:09:23', 30, 556),

('2021-06-30 06:16:06', 30, 499),

('2021-07-19 11:54:57', 30, 499),

('2021-08-07 13:16:55', 31, 499),

('2021-09-07 15:29:35', 31, 469),

('2021-10-22 11:34:43', 31, 469),

('2021-11-19 20:05:52', 32, 448),

('2021-12-06 10:04:55', 32, 448),

('2021-04-07 12:10:24', 32, 448),

('2021-05-17 23:22:18', 33, 448),

('2021-06-14 15:16:03', 33, 402),

('2021-07-23 18:01:31', 33, 402),

('2021-09-08 04:03:28', 16, 309),

('2021-09-17 17:37:03', 17, 309),

('2021-09-20 01:58:15', 17, 394),

('2021-09-22 02:13:48', 18, 394),

('2021-09-22 23:58:09', 18, 394),

('2021-09-15 13:13:59', 18, 394),

('2021-10-08 11:07:36', 19, 402),

('2021-10-27 21:19:34', 19, 439),

('2021-10-17 06:56:56', 19, 439),

('2021-10-17 11:32:51', 19, 448),

('2021-10-22 12:53:22', 20, 448),

('2021-10-06 23:37:06', 20, 469),

('2021-10-30 05:36:21', 20, 477),

('2021-10-19 23:47:24', 21, 477),

('2021-11-28 00:56:47', 21, 499),

('2021-11-29 16:17:49', 21, 499),

('2021-11-30 06:03:11', 22, 512),

('2021-11-14 15:16:03', 22, 512),

('2021-11-20 11:46:48', 22, 525),

('2021-11-22 18:22:44', 23, 541),

('2021-11-26 08:44:08', 23, 541),

('2021-11-11 04:03:54', 23, 541),

('2021-12-28 02:09:19', 24, 541),

('2021-12-30 13:52:49', 24, 541),

('2021-12-06 01:03:29', 24, 549),

('2021-12-10 08:04:41', 25, 549),

('2021-12-29 03:38:49', 25, 549),

('2021-12-26 00:25:23', 25, 556),

('2021-12-05 23:44:37', 26, 556),

('2021-12-08 01:51:32', 26, 556),

('2021-04-07 20:09:01', 26, 556),

('2021-05-27 07:51:11', 27, 227),

('2021-06-17 13:10:59', 27, 3),

('2021-07-14 15:16:03', 27, 16),

('2021-08-09 22:56:53', 28, 85),

('2021-09-01 22:07:44', 28, 583),

('2021-10-29 07:45:00', 28, 583),

('2021-11-29 13:08:06', 29, 583),

('2021-12-31 11:04:48', 29, 583),

('2021-04-22 08:50:28', 29, 596),

('2021-05-21 03:09:23', 30, 596),

('2021-06-30 06:16:06', 30, 596),

('2021-07-19 11:54:57', 30, 596),

('2021-08-07 13:16:55', 31, 162),

('2021-09-07 15:29:35', 31, 162),

('2021-10-22 11:34:43', 31, 221),

('2021-11-19 20:05:52', 32, 33),

('2021-12-06 10:04:55', 32, 33),

('2021-04-07 12:10:24', 32, 251),

('2021-05-17 23:22:18', 33, 293),

('2021-06-14 15:16:03', 33, 201),

('2021-07-23 18:01:31', 33, 201),

('2021-08-22 03:03:03', 33, 309);

INSERT INTO `series\_payment` (`customer\_id`, `series\_rental\_id`, `series\_amount`, `series\_payment\_date`) VALUES

(3, 2001, '6.99', '2021-04-21 03:42:27'),

(3, 2002, '3.99', '2021-04-11 16:47:32'),

(16, 2003, '10.99', '2021-04-17 16:10:19'),

(16, 2004, '2.99', '2021-04-15 08:41:38'),

(33, 2005, '3.99', '2021-04-17 07:12:31'),

(35, 2006, '6.99', '2021-04-22 15:52:57'),

(35, 2007, '8.99', '2021-05-27 15:34:01'),

(35, 2008, '9.99', '2021-05-30 07:14:53'),

(52, 2009, '3.99', '2021-05-06 20:38:02'),

(52, 2010, '9.99', '2021-05-27 18:47:47'),

(85, 2011, '0.99', '2021-05-23 03:45:40'),

(85, 2012, '2.99', '2021-05-15 10:18:34'),

(104, 2013, '10.99', '2021-05-16 21:48:16'),

(104, 2014, '2.99', '2021-05-07 11:32:16'),

(104, 2015, '9.99', '2021-06-27 00:57:42'),

(114, 2016, '3.99', '2021-06-17 11:21:43'),

(142, 2017, '3.99', '2021-06-19 16:47:55'),

(162, 2018, '6.99', '2021-06-30 12:30:19'),

(162, 2019, '10.99','2021-06-18 19:11:41'),

(162, 2020, '8.99', '2021-06-19 11:51:39'),

(162, 2021, '8.99', '2021-06-15 00:45:21'),

(195, 2022, '10.99', '2021-06-21 10:46:36'),

(195, 2023, '3.99', '2021-07-31 08:28:46'),

(195, 2024, '2.99', '2021-07-18 19:51:27'),

(195, 2025, '9.99', '2021-07-20 12:36:42'),

(195, 2026, '8.99', '2021-07-27 06:28:34'),

(196, 2027, '6.99', '2021-07-29 06:53:36'),

(196, 2028, '2.99', '2021-07-17 22:38:54'),

(196, 2029, '3.99', '2021-07-23 22:19:33'),

(196, 2030, '9.99', '2021-07-29 10:28:45'),

(196, 2031, '8,99', '2021-08-17 14:40:58'),

(201, 2032, '3.99', '2021-08-21 19:39:28'),

(201, 2033, '2.99', '2021-08-11 12:09:51'),

(201, 2034, '9.99', '2021-08-21 11:13:35'),

(201, 2035, '2.99', '2021-08-07 06:01:48'),

(201, 2036, '3.99', '2021-08-20 18:11:44'),

(201, 2037, '2.99', '2021-08-21 18:29:13'),

(221, 2038, '9.99', '2021-08-19 08:07:31'),

(221, 2039, '2.99', '2021-09-02 06:04:33'),

(221, 2040, '2.99', '2021-09-07 09:49:09'),

(221, 2041, '3.99', '2021-09-08 04:03:28'),

(227, 2042, '9.99', '2021-09-17 17:37:03'),

(227, 2043, '8.99', '2021-09-20 01:58:15'),

(227, 2044, '6.99', '2021-09-22 02:13:48'),

(227, 2045, '2.99', '2021-09-22 23:58:09'),

(227, 2046, '8,99', '2021-09-15 13:13:59'),

(251, 2047, '9.99', '2021-10-08 11:07:36'),

(251, 2048, '2.99', '2021-10-27 21:19:34'),

(251, 2049, '6.99', '2021-10-17 06:56:56'),

(251, 2050, '6.99', '2021-10-17 11:32:51'),

(251, 2051, '2.99', '2021-10-22 12:53:22'),

(252, 2052, '9.99', '2021-10-06 23:37:06'),

(252, 2053, '9.99', '2021-10-30 05:36:21'),

(252, 2054, '14.99', '2021-10-19 23:47:24'),

(252, 2055, '13.99', '2021-11-28 00:56:47'),

(252, 2056, '14.99', '2021-11-29 16:17:49'),

(252, 2057, '9.99', '2021-11-30 06:03:11'),

(293, 2058, '13.99', '2021-11-14 15:16:03'),

(293, 2059, '2.99', '2021-11-20 11:46:48'),

(293, 2060, '14.99', '2021-11-22 18:22:44'),

(293, 2061, '2.99', '2021-11-26 08:44:08'),

(293, 2062, '14.99', '2021-11-11 04:03:54'),

(293, 2063, '9.99', '2021-12-28 02:09:19'),

(3, 2064, '9.99', '2021-12-30 13:52:49'),

(3, 2065, '9.99', '2021-12-06 01:03:29'),

(3, 2066, '14.99', '2021-12-10 08:04:41'),

(16, 2067, '2.99', '2021-12-29 03:38:49'),

(16, 2068, '6.99', '2021-12-26 00:25:23'),

(33, 2069, '6.99', '2021-12-05 23:44:37'),

(33, 2070, '9.99', '2021-12-08 01:51:32'),

(35, 2071, '2.99', '2021-04-07 20:09:01'),

(35, 2072, '14.99', '2021-05-27 07:51:11'),

(52, 2073, '9,99', '2021-06-17 13:10:59'),

(52, 2074, '2.99', '2021-07-14 15:16:03'),

(52, 2075, '6.99', '2021-08-09 22:56:53'),

(85, 2076, '9.99', '2021-09-01 22:07:44'),

(85, 2077, '6.99', '2021-10-29 07:45:00'),

(85, 2078, '6.99', '2021-11-29 13:08:06'),

(596, 2079, '6.99', '2021-12-31 11:04:48'),

(596, 2080, '3.99', '2021-04-20 08:50:28'),

(556, 2081, '3.99', '2021-05-21 03:09:23'),

(499, 2082, '3.99', '2021-06-30 06:16:06'),

(499, 2083, '3.99', '2021-07-19 11:54:57'),

(499, 2084, '3.99', '2021-08-07 13:16:55'),

(469, 2085, '3.99', '2021-09-07 15:29:35'),

(469, 2086, '3.99', '2021-10-22 11:34:43'),

(448, 2087, '4.99', '2021-11-19 20:05:52'),

(448, 2088, '4.99', '2021-12-06 10:04:55'),

(448, 2089, '4.99', '2021-04-07 12:10:24'),

(448, 2090, '4.99', '2021-05-17 23:22:18'),

(402, 2091, '4.99', '2021-06-14 15:16:03'),

(402, 2092, '4.99', '2021-07-23 18:01:31'),

(309, 2093, '5.99', '2021-09-08 04:03:28'),

(309, 2094, '5.99', '2021-09-17 17:37:03'),

(394, 2095, '5.99', '2021-09-20 01:58:15'),

(394, 2096, '5.99', '2021-09-22 02:13:48'),

(394, 2097, '5.99', '2021-09-22 23:58:09'),

(394, 2098, '5.99', '2021-09-15 13:13:59'),

(402, 2099, '7.99', '2021-10-08 11:07:36'),

(439, 2100, '7.99', '2021-10-27 21:19:34'),

(439, 2101, '7.99', '2021-10-17 06:56:56'),

(448, 2102, '7.99', '2021-10-17 11:32:51'),

(448, 2103, '8.99', '2021-10-22 12:53:22'),

(469, 2104, '8.99', '2021-10-06 23:37:06'),

(477, 2105, '8.99', '2021-10-30 05:36:21'),

(477, 2106, '8.99', '2021-10-19 23:47:24'),

(499, 2107, '8.99', '2021-11-28 00:56:47'),

(499, 2108, '10.99', '2021-11-29 16:17:49'),

(512, 2109, '10.99', '2021-11-30 06:03:11'),

(512, 2110, '10.99', '2021-11-14 15:16:03'),

(525, 2111, '10.99', '2021-11-20 11:46:48'),

(541, 2112, '10.99',' 2021-11-22 18:22:44'),

(541, 2113, '10.99', '2021-11-26 08:44:08'),

(541, 2114, '10.99', '2021-11-11 04:03:54'),

(541, 2115, '12.99', '2021-12-28 02:09:19'),

(541, 2116, '12.99', '2021-12-30 13:52:49'),

(549, 2117, '12.99', '2021-12-06 01:03:29'),

(549, 2118, '12.99', '2021-12-10 08:04:41'),

(549, 2119, '12.99', '2021-12-29 03:38:49'),

(556, 2120, '12.99', '2021-12-26 00:25:23'),

(556, 2121, '12.99', '2021-12-05 23:44:37'),

(556, 2122, '12.99', '2021-12-08 01:51:32'),

(556, 2123, '12.99', '2021-04-07 20:09:01'),

(227, 2124, '13.99', '2021-05-27 07:51:11'),

(3, 2125, '13.99', '2021-06-17 13:10:59'),

(16, 2126, '13.99', '2021-07-14 15:16:03'),

(85, 2127, '13.99', '2021-08-09 22:56:53'),

(583, 2128, '13.99', '2021-09-01 22:07:44'),

(583, 2129, '13.99', '2021-10-29 07:45:00'),

(583, 2130, '13.99', '2021-11-29 13:08:06'),

(583, 2131, '13.99', '2021-12-31 11:04:48'),

(596, 2132, '14.99', '2021-04-22 08:50:28'),

(596, 2133, '14.99', '2021-05-21 03:09:23'),

(596, 2134, '14.99','2021-06-30 06:16:06'),

(596, 2135, '14.99', '2021-07-19 11:54:57'),

(162, 2136, '16.99','2021-08-07 13:16:55'),

(162, 2137, '16.99', '2021-09-07 15:29:35'),

(221, 2138, '16.99', '2021-10-22 11:34:43'),

(33, 2139, '16.99', '2021-11-19 20:05:52'),

(33, 2140, '16.99', '2021-12-06 10:04:55'),

(251, 2141, '16.99', '2021-04-07 12:10:24'),

(293, 2142, '16.99', '2021-05-17 23:22:18'),

(201, 2143, '16.99', '2021-06-14 15:16:03'),

(201, 2144, '16.99', '2021-07-23 18:01:31'),

(289, 309, 2145, '6.99', '2021-08-22 03:03:03');

# STORED PROCEDURES

**3.1)**

drop procedure if exists most\_rented;

delimiter $

create procedure most\_rented(choice char,size int, date1 datetime, date2 datetime)

begin

declare filmid smallint(3);

declare mtitle varchar(128);

declare seriesid smallint(3);

declare stitle varchar(128);

declare fflag int;

declare mcursor cursor for

select film.film\_id,film.title from film

inner join inventory on film.film\_id=inventory.film\_id

inner join rental on inventory.inventory\_id=rental.inventory\_id

group by film.film\_id

where rental.rental\_date>=date1 and rental.rental\_date<=date2

having max(count(inventory\_id));

declare scursor cursor for

select series.series\_id,series.title from series

inner join series\_inventory on series.series\_id=series\_inventory.series\_id

inner join series\_rental on series\_inventory.inventory\_id=series\_rental.inventory\_id

group by series.series\_id

where series\_rental.rental\_date>=date1 and series\_rental.rental\_date<=date2

having max(count(series\_inventory\_id));

declare continue handler for not found set fflag=1;

set fflag=0;

case (choice)

when m then

open mcursor;

repeat

fetch mcursor into filmid,mtitle;

if (fflag=0) then

while(count(mtitle<size)) do

select filmid,mtitle from film;

end while;

end if;

until(fflag=1)

end repeat;

close mcursor;

when s then

set fflag=0;

open scursor;

repeat

fetch scursor into seriesid,stitle;

if (fflag=0) then

while(count(stitle<size)) do

select seriesid,stitle from series;

end while;

end if;

until(fflag=1)

end repeat;

close scursor;

end case;

end$

delimiter ;

**3.2)**

drop procedure if exists rentals;

delimiter $

create procedure rentals(cemail varchar(50), date1 datetime)

begin

declare cust\_id int(11);

declare rent\_date date;

declare cust2\_id int(11);

declare rent2\_date date;

declare flag int;

declare rcursor cursor for

select customer.customer\_id,DATE(rental\_date)

from customer

inner join rental on rental.customer\_id=customer.customer\_id

where email=cemail AND DATE(rental\_date)=date1;

declare rcursor2 cursor for

select customer.customer\_id,DATE(series\_rental.rental\_date)

from customer

inner join series\_rental on series\_rental.customer\_id=customer.customer\_id

where email=cemail AND DATE(series\_rental.rental\_date)=date1;

declare continue handler for not found set flag=1;

set flag=0;

open rcursor;

repeat

fetch rcursor into cust\_id,rent\_date;

until (flag=1)

end repeat;

select cust\_id AS'ID pelath',count(rent\_date) AS 'arithmos enoikiasewn tainiwn'

from rental

where customer\_id=cust\_id AND DATE(rental\_date)=date1;

close rcursor;

set flag=0;

open rcursor2;

repeat

fetch rcursor2 into cust2\_id,rent2\_date;

until (flag=1)

end repeat;

select cust2\_id as'ID pelath',count(rent2\_date) as 'arithmos enoikiasewn seirwn'

from series\_rental

where customer\_id=cust2\_id AND DATE(series\_rental.rental\_date)=date1;

close rcursor2;

end$

delimiter ;

**3.3)**

drop procedure if exists earnings;

delimiter $

create procedure earnings()

begin

select sum(amount) as 'Synoliko poso plhrwmwn tainiwn Febrouariou' from payment where payment\_date like'%-02-%';

select sum(series\_amount) as 'Synoliko poso plhrwmwn seirwn Apriliou' from series\_payment where series\_payment\_date like'%-04-%';

select sum(amount) as 'Synoliko poso plhrwmwn tainiwn Maiou' from payment where payment\_date like'%-05-%';

select sum(series\_amount) as 'Synoliko poso plhrwmwn seirwn Maiou' from series\_payment where series\_payment\_date like'%-05-%';

select sum(amount) as 'Synoliko poso plhrwmwn tainiwn Iouniou' from payment where payment\_date like'%-06-%';

select sum(series\_amount) as 'Synoliko poso plhrwmwn seirwn Iouniou' from series\_payment where series\_payment\_date like'%-06-%';

select sum(amount) as 'Synoliko poso plhrwmwn tainiwn Iouliou' from payment where payment\_date like'%-07-%';

select sum(series\_amount) as 'Synoliko poso plhrwmwn seirwn Iouliou' from series\_payment where series\_payment\_date like'%-07-%';

select sum(amount) as 'Synoliko poso plhrwmwn tainiwn Augoustou' from payment where payment\_date like'%-08-%';

select sum(series\_amount) as 'Synoliko poso plhrwmwn seirwn Augoustou' from series\_payment where series\_payment\_date like'%-08-%';

select sum(series\_amount) as 'Synoliko poso plhrwmwn seirwn Septembriou' from series\_payment where series\_payment\_date like'%-09-%';

select sum(series\_amount) as 'Synoliko poso plhrwmwn seirwn Okrobriou' from series\_payment where series\_payment\_date like'%-10-%';

select sum(series\_amount) as 'Synoliko poso plhrwmwn seirwn Noembriou' from series\_payment where series\_payment\_date like'%-11-%';

select sum(series\_amount) as 'Synoliko poso plhrwmwn seirwn Decembriou' from series\_payment where series\_payment\_date like'%-12-%';

end$

delimiter ;

**3.4.α)**

drop procedure if exists lastnames;

delimiter $

create procedure lastnames(lastname1 varchar(45),lastname2 varchar(45))

begin

declare fflag int;

declare fname varchar(45);

declare lname varchar(45);

declare lcursor cursor for

select first\_name,last\_name from actor where last\_name>=lastname1 and last\_name<=lastname2 order by last\_name;

declare continue handler for not found set fflag=1;

open lcursor;

set fflag=0;

repeat

fetch lcursor into fname,lname;

if (fflag=0) then

select fname as 'Onoma',lname as 'Epitheto';

end if;

until(fflag=1)

end repeat;

select count(first\_name) as 'Plithos Hthopoiwn' from actor where last\_name>=lastname1 and last\_name<=lastname2;

close lcursor;

end$

delimiter ;

**3.4.β)**

drop procedure if exists actordata;

delimiter $

create procedure actordata(lastname varchar(45))

begin

declare fflag int;

declare fname varchar(45);

declare lname varchar(45);

declare lcursor cursor for

select first\_name,last\_name from actor where last\_name=lastname order by last\_name;

declare continue handler for not found set fflag=1;

open lcursor;

set fflag=0;

repeat

fetch lcursor into fname,lname;

if (fflag=0) then

select fname as 'Onoma',lname as 'Epitheto';

end if;

until(fflag=1)

end repeat;

select count(first\_name) as 'Plithos Hthopoiwn' from actor where last\_name=lastname;

close lcursor;

end$

delimiter ;

# TRIGGERS

**4.2)**

drop trigger if exists sale;

delimiter $

create trigger sale

before insert on payment

for each row

begin

declare mcount int;

declare mdate datetime;

declare scount int;

declare sdate datetime;

declare flag int;

declare mcursor cursor for

select count(customer\_id),payment\_date

from payment;

declare scursor cursor for

select count(customer\_id),series\_payment\_date

from series\_payment;

declare continue handler for not found set flag=1;

set flag=0;

open mcursor;

repeat

fetch mcursor into mcount,mdate;

if(mcount>=3 and mdate=new.payment\_date) then

set new.cost\_per\_film=cost\_per\_film/2 and new.amount=amount/2;

end if;

until(flag=1)

end repeat;

close mcursor;

set flag=0;

open scursor;

repeat

fetch scursor into scount,sdate;

if(scount>=3 and sdate=new.payment\_date) then

set new.cost\_per\_film=cost\_per\_film/2 and new.amount=amount/2;

end if;

until(flag=1)

end repeat;

close scursor;

end$

delimiter ;

**4.3)**

drop trigger if exists not\_allowed;

delimiter $

create trigger not\_allowed

before update on customer

for each row

begin

if (new.customer\_id IS NOT NULL) then

SIGNAL SQLSTATE VALUE '45000' SET MESSAGE\_TEXT = 'You cannot change your Customer Id.';

end if;

end$

delimiter ;