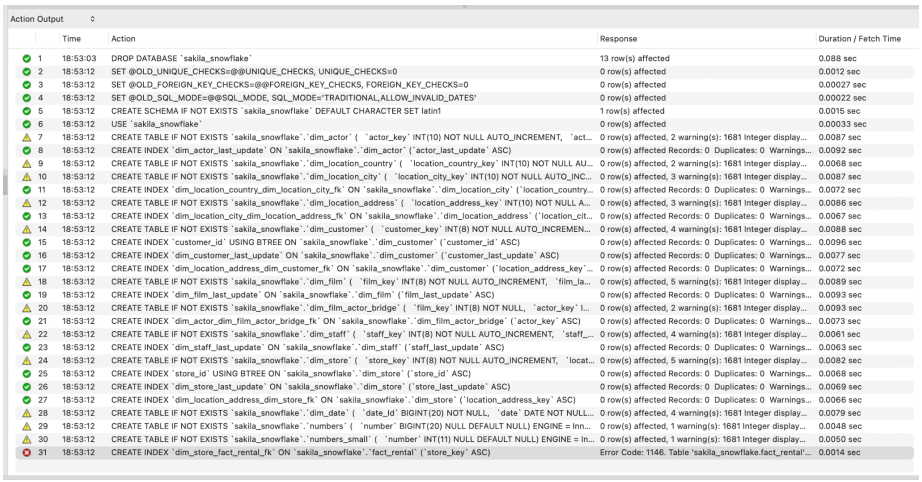
MSCA 31012 Homework3
Zihao Liu

2

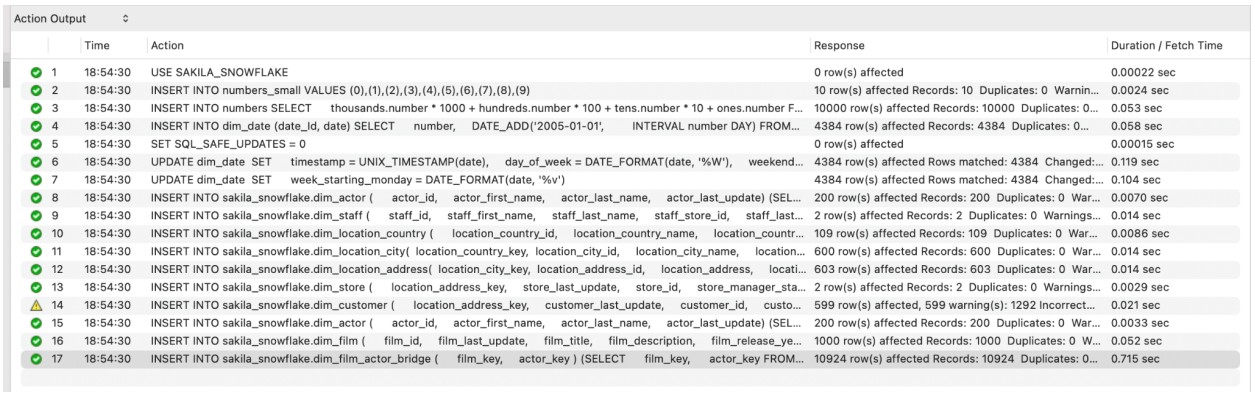
a. Run the DDL & DML provided for the Sakila dimension
DDL



Action Output

	Time	Action	Response	Duration / Fetch Time
1	18:53:03	DROP DATABASE `sakila_snowflake`	13 row(s) affected	0.088 sec
2	18:53:12	SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0	0 row(s) affected	0.0012 sec
3	18:53:12	SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0	0 row(s) affected	0.00027 sec
4	18:53:12	SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES'	0 row(s) affected	0.00022 sec
5	18:53:12	CREATE SCHEMA IF NOT EXISTS `sakila_snowflake` DEFAULT CHARACTER SET latin1	1 row(s) affected	0.0015 sec
6	18:53:12	USE `sakila_snowflake`	0 row(s) affected	0.00033 sec
7	18:53:12	CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`dim_actor` (`actor_key` INT(10) NOT NULL AUTO_INCREMENT, `act...	0 row(s) affected, 2 warning(s): 1681 Integer display...	0.0087 sec
8	18:53:12	CREATE INDEX `dim_actor_last_update` ON `sakila_snowflake`.`dim_actor` (`actor_last_update` ASC)	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.0092 sec
9	18:53:12	CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`dim_location_country` (`location_country_key` INT(10) NOT NULL AU...	0 row(s) affected, 2 warning(s): 1681 Integer display...	0.0068 sec
10	18:53:12	CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`dim_location_city` (`location_city_key` INT(10) NOT NULL AUTO_INC...	0 row(s) affected, 3 warning(s): 1681 Integer display...	0.0087 sec
11	18:53:12	CREATE INDEX `dim_location_country_dim_location_city_fk` ON `sakila_snowflake`.`dim_location_city` (`location_country...	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.0072 sec
12	18:53:12	CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`dim_location_address` (`location_address_key` INT(10) NOT NULL A...	0 row(s) affected, 3 warning(s): 1681 Integer display...	0.0086 sec
13	18:53:12	CREATE INDEX `dim_location_city_dim_location_address_fk` ON `sakila_snowflake`.`dim_location_address` (`location_cit...	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.0067 sec
14	18:53:12	CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`dim_customer` (`customer_key` INT(8) NOT NULL AUTO_INCREMEN...	0 row(s) affected, 4 warning(s): 1681 Integer display...	0.0088 sec
15	18:53:12	CREATE INDEX `customer_id` USING BTREE ON `sakila_snowflake`.`dim_customer` (`customer_id` ASC)	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.0096 sec
16	18:53:12	CREATE INDEX `dim_customer_last_update` ON `sakila_snowflake`.`dim_customer` (`customer_last_update` ASC)	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.0077 sec
17	18:53:12	CREATE INDEX `dim_location_address_dim_customer_fk` ON `sakila_snowflake`.`dim_customer` (`location_address_key`...	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.0072 sec
18	18:53:12	CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`dim_film` (`film_key` INT(8) NOT NULL AUTO_INCREMENT, `film_la...	0 row(s) affected, 5 warning(s): 1681 Integer display...	0.0089 sec
19	18:53:12	CREATE INDEX `dim_film_last_update` ON `sakila_snowflake`.`dim_film` (`film_last_update` ASC)	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.0093 sec
20	18:53:12	CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`dim_film_actor_bridge` (`film_key` INT(8) NOT NULL, `actor_key` I...	0 row(s) affected, 2 warning(s): 1681 Integer display...	0.0093 sec
21	18:53:12	CREATE INDEX `dim_actor_dim_film_actor_bridge_fk` ON `sakila_snowflake`.`dim_film_actor_bridge` (`actor_key` ASC)	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.0073 sec
22	18:53:12	CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`dim_staff` (`staff_key` INT(8) NOT NULL AUTO_INCREMENT, `staff_...	0 row(s) affected, 4 warning(s): 1681 Integer display...	0.0061 sec
23	18:53:12	CREATE INDEX `dim_staff_last_update` ON `sakila_snowflake`.`dim_staff` (`staff_last_update` ASC)	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.0063 sec
24	18:53:12	CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`dim_store` (`store_key` INT(8) NOT NULL AUTO_INCREMENT, `locat...	0 row(s) affected, 5 warning(s): 1681 Integer display...	0.0082 sec
25	18:53:12	CREATE INDEX `store_id` USING BTREE ON `sakila_snowflake`.`dim_store` (`store_id` ASC)	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.0068 sec
26	18:53:12	CREATE INDEX `dim_store_last_update` ON `sakila_snowflake`.`dim_store` (`store_last_update` ASC)	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.0069 sec
27	18:53:12	CREATE INDEX `dim_location_address_dim_store_fk` ON `sakila_snowflake`.`dim_store` (`location_address_key` ASC)	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.0066 sec
28	18:53:12	CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`dim_date` (`date_Id` BIGINT(20) NOT NULL, `date` DATE NOT NULL...	0 row(s) affected, 4 warning(s): 1681 Integer display...	0.0079 sec
29	18:53:12	CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`numbers` (`number` BIGINT(20) NULL DEFAULT NULL) ENGINE = Inn...	0 row(s) affected, 1 warning(s): 1681 Integer display...	0.0048 sec
30	18:53:12	CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`numbers_small` (`number` INT(11) NULL DEFAULT NULL) ENGINE = In...	0 row(s) affected, 1 warning(s): 1681 Integer display...	0.0050 sec
31	18:53:12	CREATE INDEX `dim_store_fact_rental_fk` ON `sakila_snowflake`.`fact_rental` (`store_key` ASC)	Error Code: 1146. Table 'sakila_snowflake.fact_rental'...	0.0014 sec

DML



Action Output

	Time	Action	Response	Duration / Fetch Time
1	18:54:30	USE SAKILA_SNOWFLAKE	0 row(s) affected	0.00022 sec
2	18:54:30	INSERT INTO numbers_small VALUES (0),(1),(2),(3),(4),(5),(6),(7),(8),(9)	10 row(s) affected Records: 10 Duplicates: 0 Warnin...	0.0024 sec
3	18:54:30	INSERT INTO numbers SELECT thousands.number * 1000 + hundreds.number * 100 + tens.number * 10 + ones.number F...	10000 row(s) affected Records: 10000 Duplicates: 0...	0.053 sec
4	18:54:30	INSERT INTO dim_date (date_Id, date) SELECT number, DATE_ADD('2005-01-01', INTERVAL number DAY) FROM...	4384 row(s) affected Records: 4384 Duplicates: 0...	0.058 sec
5	18:54:30	SET SQL_SAFE_UPDATES = 0	0 row(s) affected	0.00015 sec
6	18:54:30	UPDATE dim_date SET timestamp = UNIX_TIMESTAMP(date), day_of_week = DATE_FORMAT(date, '%W'), weekend...	4384 row(s) affected Rows matched: 4384 Changed:...	0.119 sec
7	18:54:30	UPDATE dim_date SET week_starting_monday = DATE_FORMAT(date, '%v')	4384 row(s) affected Rows matched: 4384 Changed:...	0.104 sec
8	18:54:30	INSERT INTO sakila_snowflake.dim_actor (actor_id, actor_first_name, actor_last_name, actor_last_update) (SEL...	200 row(s) affected Records: 200 Duplicates: 0 War...	0.0070 sec
9	18:54:30	INSERT INTO sakila_snowflake.dim_staff (staff_id, staff_first_name, staff_last_name, staff_store_id, staff_last...	2 row(s) affected Records: 2 Duplicates: 0 Warnings...	0.014 sec
10	18:54:30	INSERT INTO sakila_snowflake.dim_location_country (location_country_id, location_country_name, location_countr...	109 row(s) affected Records: 109 Duplicates: 0 War...	0.0086 sec
11	18:54:30	INSERT INTO sakila_snowflake.dim_location_city(location_country_key, location_city_id, location_city_name, location...	600 row(s) affected Records: 600 Duplicates: 0 War...	0.014 sec
12	18:54:30	INSERT INTO sakila_snowflake.dim_location_address(location_city_key, location_address_id, location_address, locati...	603 row(s) affected Records: 603 Duplicates: 0 War...	0.014 sec
13	18:54:30	INSERT INTO sakila_snowflake.dim_store (location_address_key, store_last_update, store_id, store_manager_sta...	2 row(s) affected Records: 2 Duplicates: 0 Warnings...	0.0029 sec
14	18:54:30	INSERT INTO sakila_snowflake.dim_customer (location_address_key, customer_last_update, customer_id, custo...	599 row(s) affected, 599 warning(s): 1292 Incorrect...	0.021 sec
15	18:54:30	INSERT INTO sakila_snowflake.dim_actor (actor_id, actor_first_name, actor_last_name, actor_last_update) (SEL...	200 row(s) affected Records: 200 Duplicates: 0 War...	0.0033 sec
16	18:54:30	INSERT INTO sakila_snowflake.dim_film (film_id, film_last_update, film_title, film_description, film_release_ye...	1000 row(s) affected Records: 1000 Duplicates: 0 W...	0.052 sec
17	18:54:30	INSERT INTO sakila_snowflake.dim_film_actor_bridge (film_key, actor_key) (SELECT film_key, actor_key FROM...	10924 row(s) affected Records: 10924 Duplicates: 0...	0.715 sec

b. For the missing Fact table fact_rental create the DDL and the DML scripts to pull data from the Sakila ER model. Modify/improve the fact table script if you need more metrics for the Tableau report below (For example to include the dollar amount for each rental use rental rate * number of days rented).

#DDL

```
-- Table `sakila_snowflake`.`fact_rental`
-- Write Fact table fact_rental DDL script here
CREATE TABLE IF NOT EXISTS `sakila_snowflake`.`fact_rental` (
  `rental_id` INT(11) NOT NULL,
  `rental_last_update` TIMESTAMP NOT NULL,
  `customer_key` INT(8) NOT NULL,
  `staff_key` INT(8) NOT NULL,
  `film_key` INT(8) NOT NULL,
  `store_key` INT(8) NOT NULL,
  `rental_date_key` BIGINT(20) NOT NULL,
  `return_date_key` BIGINT(20)  NULL,
  `count_returns` INT(10) NULL,
  `count_rentals` INT(10) NULL,
  `rental_duration` INT(10) NULL,
  `dollar_amount` FLOAT NULL,
  PRIMARY KEY (`rental_id`),
  INDEX `fk_fact_rental_dim_customer_idx` (`customer_key` ASC),
  INDEX `fk_fact_rental_dim_staff_idx` (`staff_key` ASC),
  INDEX `fk_fact_rental_dim_film_idx` (`film_key` ASC),
  INDEX `fk_fact_rental_dim_store_idx` (`store_key` ASC),
  INDEX `fk_fact_rental_dim_date_idx`(`rental_date_key` ASC),
  INDEX `fk_fact_return_dim_date_idx`(`return_date_key` ASC),
  CONSTRAINT `fk_fact_rental_dim_customer`
    FOREIGN KEY (`customer_key`)
    REFERENCES `sakila_snowflake`.`dim_customer` (`customer_key`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_fact_rental_dim_staff`
    FOREIGN KEY (`staff_key`)
    REFERENCES `sakila_snowflake`.`dim_staff` (`staff_key`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_fact_rental_dim_film`
    FOREIGN KEY (`film_key`)
    REFERENCES `sakila_snowflake`.`dim_film` (`film_key`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_fact_rental_dim_store`
    FOREIGN KEY (`store_key`)
    REFERENCES `sakila_snowflake`.`dim_store` (`store_key`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
```

```
CONSTRAINT `fk_fact_rental_dim_date`
    FOREIGN KEY (`rental_date_key`)
    REFERENCES `sakila_snowflake`.`dim_date` (`date_Id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_fact_return_dim_date`
    FOREIGN KEY (`return_date_key`)
    REFERENCES `sakila_snowflake`.`dim_date` (`date_ID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;



#DML
-- -----------------------------------------------------
-- Write Fact table fact_rental DML script here
-- -----------------------------------------------------
INSERT INTO sakila_snowflake.fact_rental (
 rental_id,
 rental_last_update,
 customer_key,
 staff_key,
film_key,
store_key,
rental_date_key,
 return_date_key,
count_returns,
count_rentals,
rental_duration,
dollar_amount
)(
SELECT
r.rental_id,
r.last_update,
r.customer_id,
r.staff_id,
i.film_id,
i.store_id,
rental1.date_Id,
return1.date_Id,
case when r.return_date is null then 0 else 1 end ,
case when r.rental_date is null then 0 else 1 end ,
```

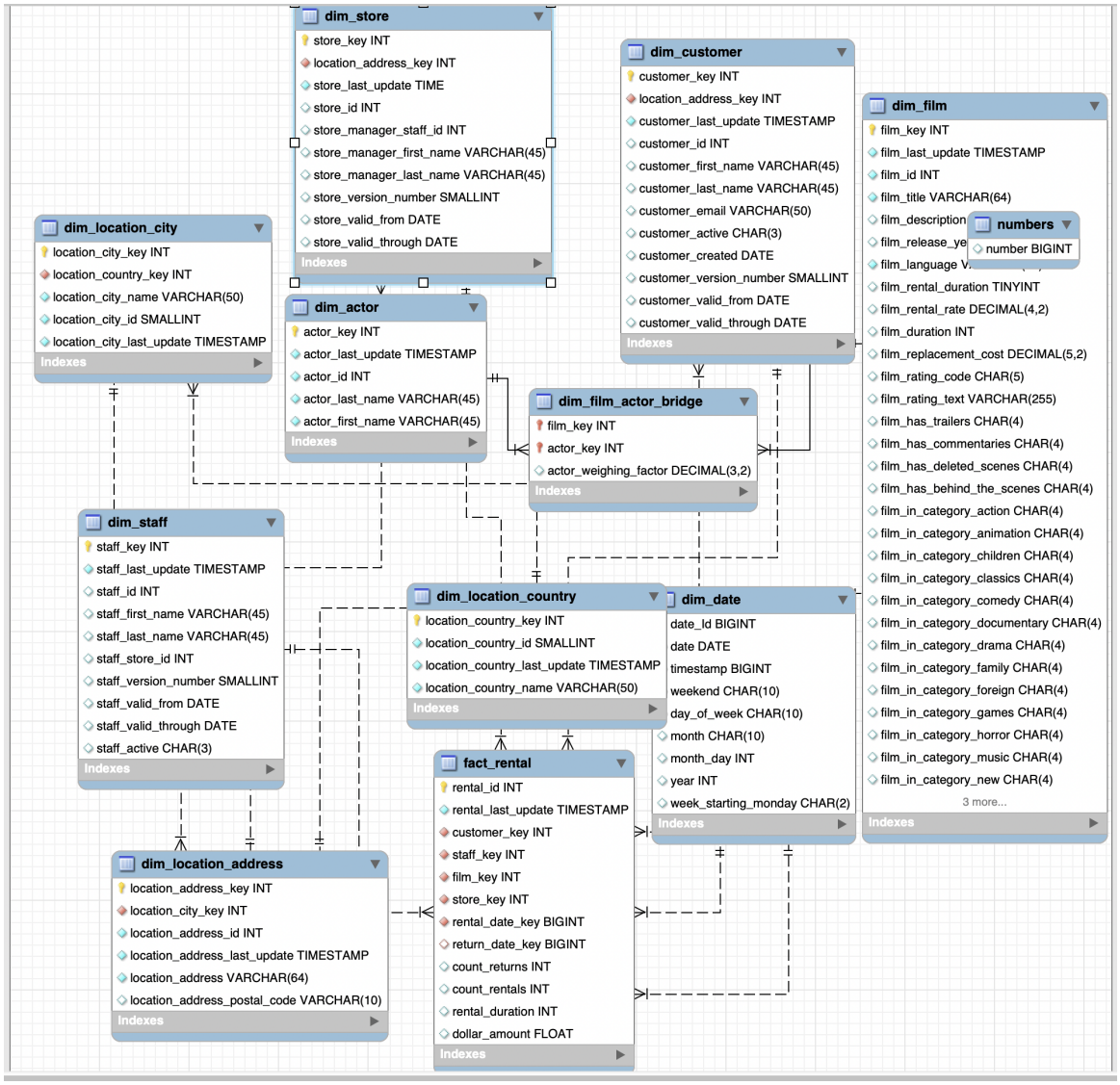
```
case when DATEDIFF(r.return_date, r.rental_date) is null then 0 else
DATEDIFF(r.return_date, r.rental_date) end ,
(case when DATEDIFF(r.return_date, r.rental_date) is null then 0 else
DATEDIFF(r.return_date, r.rental_date) end * f.rental_rate)



FROM sakila.rental as r
left join sakila.inventory as i
on r.inventory_id = i.inventory_id
left join sakila.film as f
on i.film_id = f.film_id,
dim_date as rental1,
dim_date as return1

WHERE
rental1.date = date_format(rental_date,'%Y-%m-%d')
AND return1.date=date_format(return_date,'%Y-%m-%d')
);
```

c. Generate the EER diagram for the final dimensional model.



d. Identify any scripts errors and areas of improvement in the

data model.

1 For dim_actor table, add ORDER BY actor_id

2 For dim_customer, add ORDER BY customer_id

3 Delete the repeated chunk for creating dim_actor

4 For dim_film, order by film_id

5 Change ”scifi“ to “Sci-Fi” for tableau insights

3. Using Tableau and referencing the snowflake dimensional model (Sakila data warehouse) create 5 different reports that provides insights into the sakila dataset. You may use any of the below use cases or you can create your own reporting use cases/Insights:– { 40 Points }

a. Create a report that helps visualize list of movies titles grouped by categories

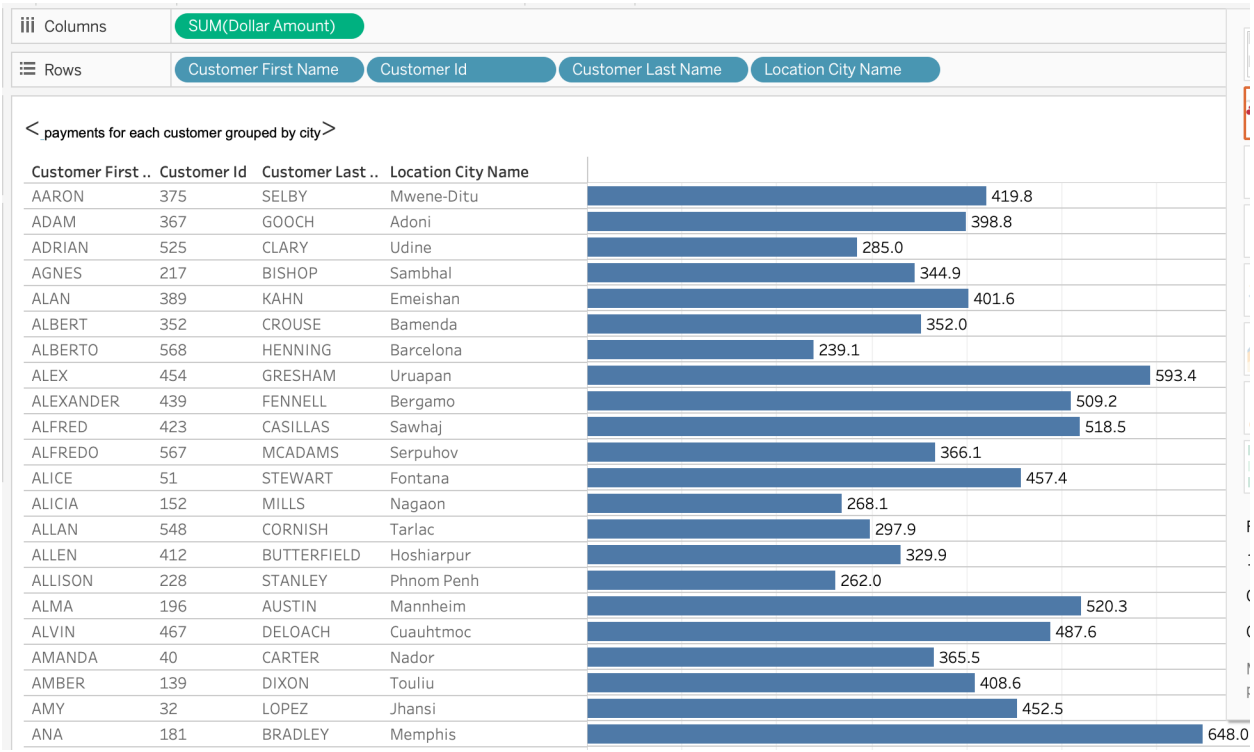
Film Title	Cat
ACADEMY DINOSAUR	Documentary
ACE GOLDFINGER	Horror
ADAPTATION HOLES	Documentary
AFFAIR PREJUDICE	Horror
AFRICAN EGG	Family
AGENT TRUMAN	Foreign
AIRPLANE SIERRA	Comedy
AIRPORT POLLOCK	Horror
ALABAMA DEVIL	Horror
ALADDIN CALENDAR	Sports
ALAMO VIDEOTAPE	Foreign
ALASKA PHANTOM	Music
ALI FOREVER	Horror
ALICE FANTASIA	Classics
ALIEN CENTER	Foreign
ALLEY EVOLUTION	Foreign
ALONE TRIP	Music
ALTER VICTORY	Animation
AMADEUS HOLY	Action
AMELIE HELLFIGHTERS	Music
AMERICAN CIRCUS	Action
AMISTAD MIDSUMMER	NEW
ANACONDA CONFESSIONS	Animation
ANALYZE HOOSIERS	Horror
ANGELS LIFE	NEW

c.Create a report that list the Labels and language for all DVDs

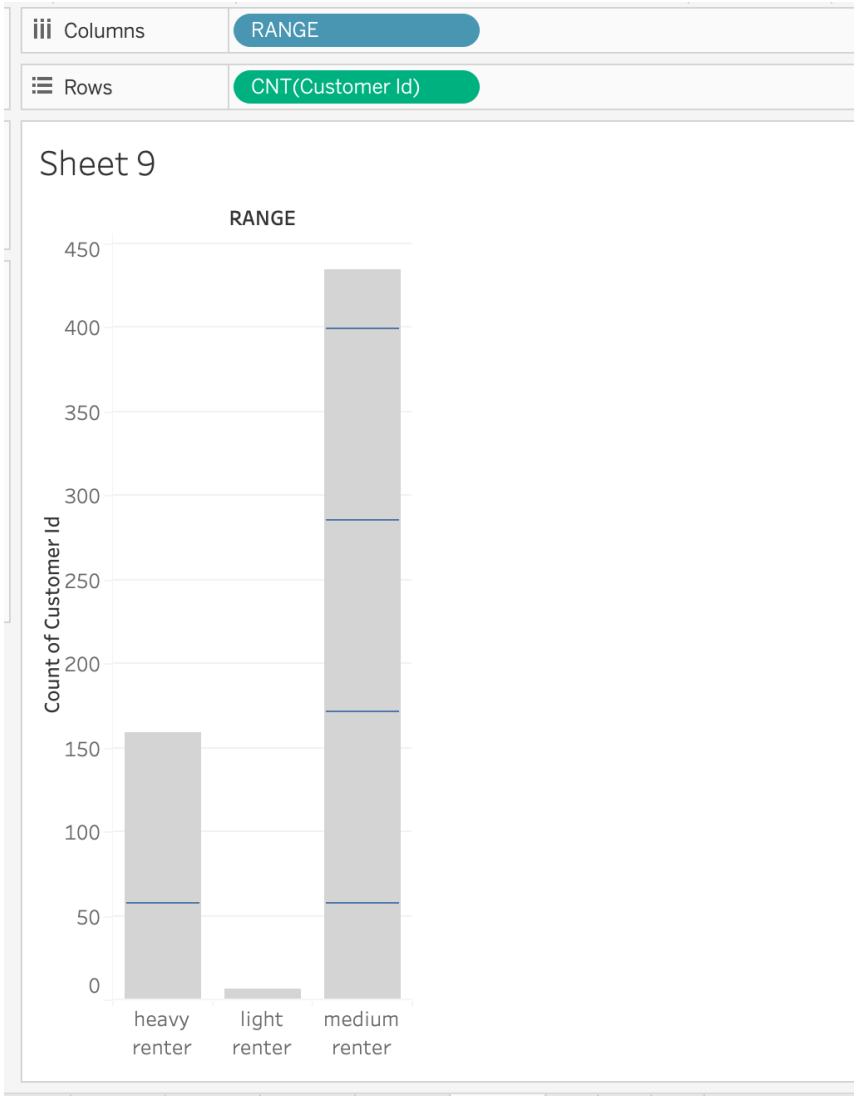
<Labels and language for all DVDs >

Film Title	Film Langu..	Film Description	
ACADEMY DINOSAUR	English	A Epic Drama of a Feminis..	Abc
ACE GOLDFINGER	English	A Astounding Epistle of a ..	Abc
ADAPTATION HOLES	English	A Astounding Reflection o..	Abc
AFFAIR PREJUDICE	English	A Fanciful Documentary o..	Abc
AFRICAN EGG	English	A Fast-Paced Documentar..	Abc
AGENT TRUMAN	English	A Intrepid Panorama of a ..	Abc
AIRPLANE SIERRA	English	A Touching Saga of a Hunt..	Abc
AIRPORT POLLOCK	English	A Epic Tale of a Moose An..	Abc
ALABAMA DEVIL	English	A Thoughtful Panorama o..	Abc
ALADDIN CALENDAR	English	A Action-Packed Tale of a ..	Abc
ALAMO VIDEOTAPE	English	A Boring Epistle of a Butle..	Abc
ALASKA PHANTOM	English	A Fanciful Saga of a Hunte..	Abc
ALI FOREVER	English	A Action-Packed Drama of..	Abc
ALICE FANTASIA	English	A Emotional Drama of a A ..	Abc
ALIEN CENTER	English	A Brilliant Drama of a Cat ..	Abc
ALLEY EVOLUTION	English	A Fast-Paced Drama of a R..	Abc
ALONE TRIP	English	A Fast-Paced Character St..	Abc
ALTER VICTORY	English	A Thoughtful Drama of a C..	Abc
AMADEUS HOLY	English	A Emotional Display of a P..	Abc
AMELIE HELLFIGHTERS	English	A Boring Drama of a Wom..	Abc
AMERICAN CIRCUS	English	A Insightful Drama of a Gi..	Abc
AMISTAD MIDSUMMER	English	A Emotional Character St..	Abc
ANACONDA CONFESSIONS	English	A Lacklusture Display of a ..	Abc
ANALYZE HOOSIERS	English	A Thoughtful Display of a ..	Abc
ANGELS LIFE	English	A Thoughtful Display of a ..	Abc

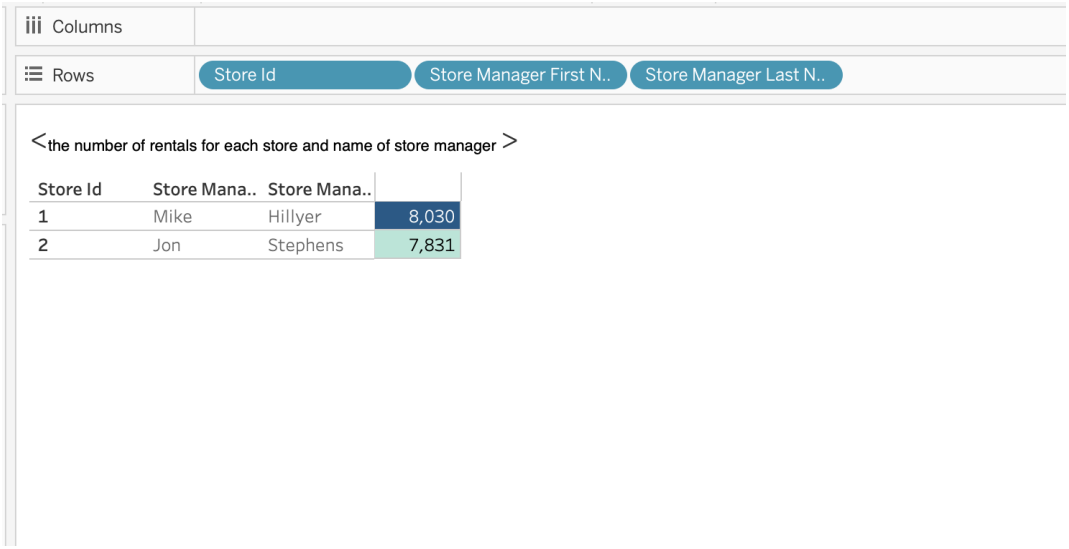
d. Create visualization with payments for each customer grouped by city



e. Create charts that gives insight into the number of customers, ranged by number of rentals made

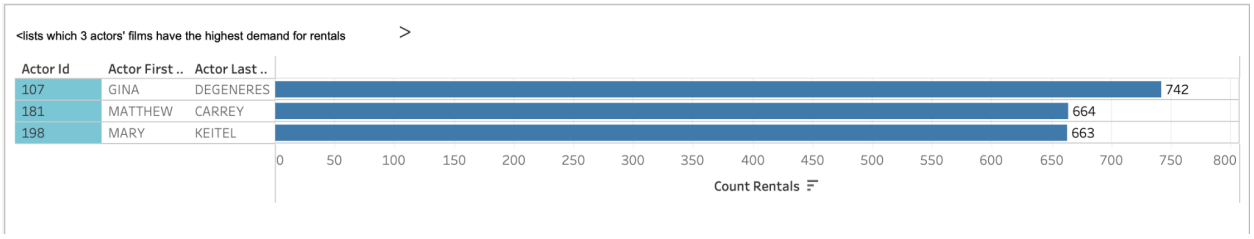


f. Create charts that gives insight into the number of rentals for each store and name of store manager



Store Id	Store Mana..	Store Mana..	
1	Mike	Hillyer	8,030
2	Jon	Stephens	7,831

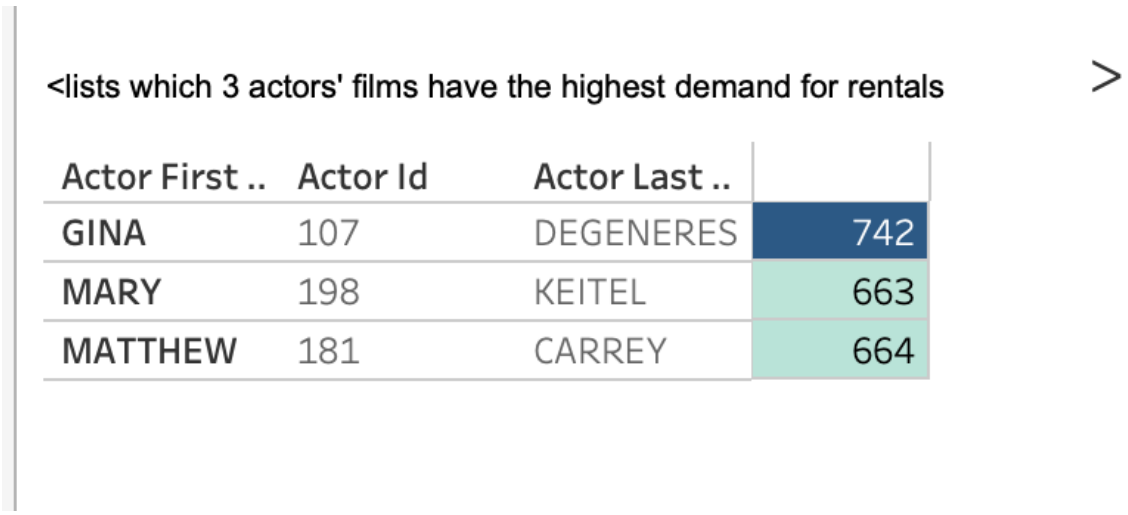
g. Create table that lists which 3 actors' films have the highest demand for rentals



Actor Id	Actor First ..	Actor Last ..	
107	GINA	DEGENERES	742
181	MATTHEW	CARREY	664
198	MARY	KEITEL	663

0 50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800

Count Rentals



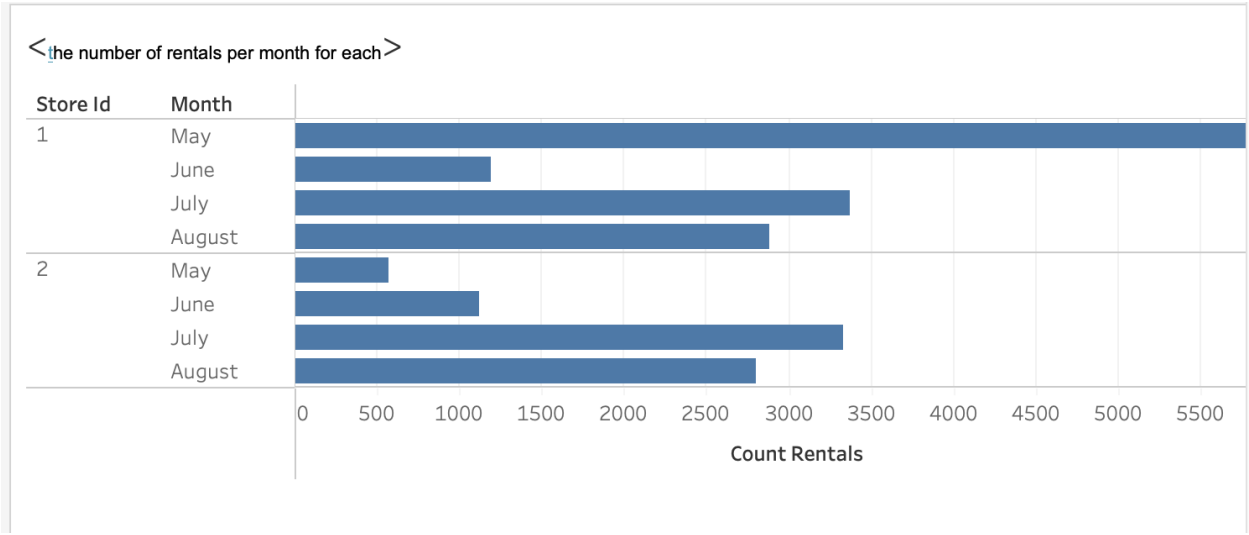
<lists which 3 actors' films have the highest demand for rentals >

Actor First ..	Actor Id	Actor Last ..	
GINA	107	DEGENERES	742
MARY	198	KEITEL	663
MATTHEW	181	CARREY	664

h. Build charts that gives insight into the number of rentals per month for each

<the number of rentals per month for each>

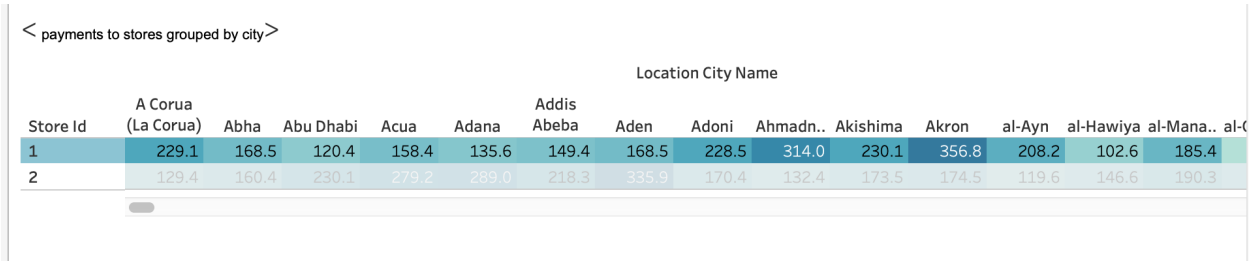
Store Id	Month			
	May	June	July	August
1	5,810	1,190	3,375	2,884
2	575	1,121	3,334	2,801



Additional reports

Some are modified from previous insight（like adding condition）to suit the theme of dashboard

a Payments to Store by City

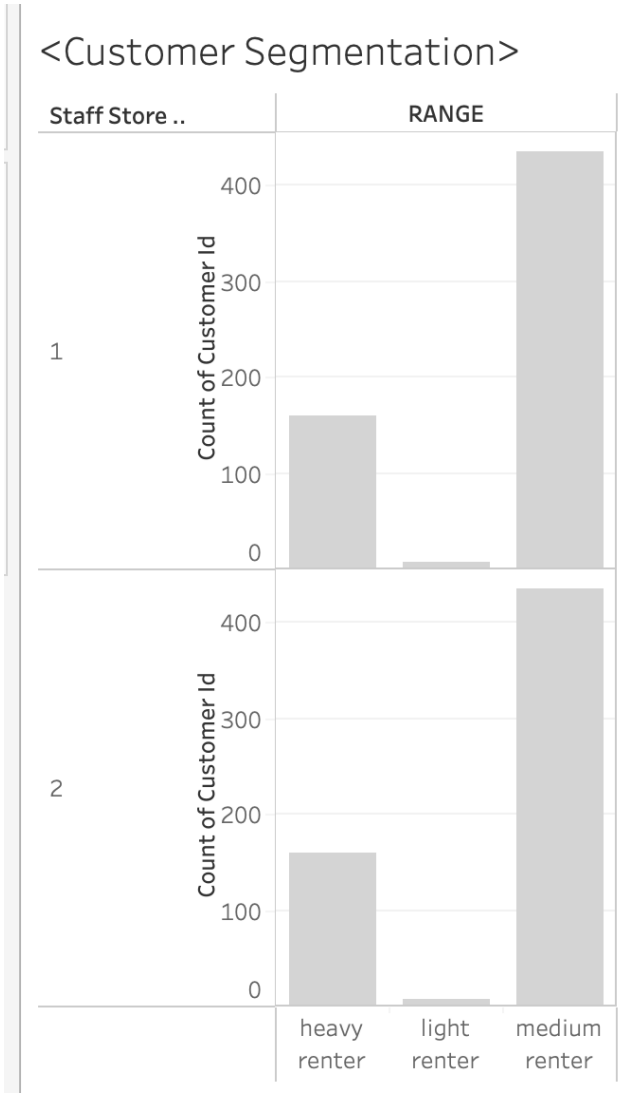


B List the demand of move rentals

<FIlm DEMAND>

Film Id	Film Title	
103	BUCKET BROTHERHOOD	34
738	ROCKETEER MOTHER	33
331	FORWARD TEMPLE	32
382	GRIT CLOCKWORK	32
767	SCALAWAG DUCK	32
31	APACHE DIVINE	31
369	GOODFELLAS SALUTE	31
418	HOBBIT ALIEN	31
489	JUGGLER HARDLY	31
621	NETWORK PEAK	31
730	RIDGEMONT SUBMARINE	31
735	ROBBERS JOON	31
753	RUSH GOODFELLAS	31
891	TIMBERLAND SKY	31
973	WIFE TURN	31
1000	ZORRO ARK	31
109	BUTTERFLY CHOCOLAT	30
239	DOGMA FAMILY	30
285	ENGLISH BULWORTH	30
374	GRAFFITI LOVE	30
450	IDOLS SNATCHERS	30
559	MARRIED GO	30
563	MASSACRE USUAL	30
609	MUSCLE BRIGHT	30
748	RUGRATS SHAKESPEARE	30

C number of customer by range and group by store



D rental by day of week

2. Create a dashboard based on the report set that you have created (For example: Store level dashboard that shows inventory, number of customers per store, revenue, etc.) Note: Make sure the context within the dashboard can be switched using filters.

This dashboard can provid insights on customer, film demand and revenue compositions by time or by location. It can give insights at a store level