DESCRIPTIVE ANALYSIS:

```
-- descriptive stats of film (3.6.2)
SELECT
                    MIN(release_year) AS min_release_year,
                    MAX(release_year) AS max_release_year,
         ROUND(AVG(release_year),0) AS avg_release_year,
                    MIN(rental_duration) AS min_rental_duration,
                    MAX(rental_duration) AS max_rental_duration,
         ROUND(AVG(rental_duration),0) AS avg_rental_duration,
                    MIN(length) AS min_length,
                    MAX(length) AS max_length,
         ROUND(AVG(length),0) AS avg_length,
                    MIN(rental_rate) AS min_rental_rate,
                    MAX(rental_rate) AS max_rental_rate,
         ROUND(AVG(rental_rate),0)AS avg_rental_rate,
                    MIN(replacement_cost) AS min_replacement_cost,
                    MAX(replacement_cost) AS max_replacement_cost,
         ROUND(AVG(replacement_cost),0) AS avg_replacement_cost
FROM film
```

```
-- descriptive stats of film (3.6.2) -- mode for rating
SELECT MODE() WITHIN GROUP (ORDER BY rating) AS mode_rating
FROM film
```

```
-- count of total inventory
SELECT COUNT(distinct inventory_id)
FROM inventory
```

```
-- how many countries does RB serve? 109
SELECT DISTINCT country
FROM payment A
LEFT JOIN customer B       ON A.customer_id =         B.customer_id
RIGHT JOIN address C       ON B.address_id =          C.address_id
LEFT JOIN city D           ON C.city_id =             D.city_id
LEFT JOIN country E        ON D.country_id =          E.country_id
ORDER BY 1
```

ANALYSIS:

```
-- number of transactions (ttl customers, ttl revenue, ttl transaction) in each country
SELECT
        E.country,
        COUNT(DISTINCT B.customer_id) AS count_customers,
        SUM(amount) AS sum_amount,
        COUNT(DISTINCT payment_id) AS count_transaction
FROM payment A
INNER JOIN customer B   ON A.customer_id =    B.customer_id
INNER JOIN address C    ON B.address_id =     C.address_id
INNER JOIN city D       ON C.city_id =        D.city_id
INNER JOIN country E    ON D.country_id =     E.country_id
GROUP BY
        E.country
ORDER BY 3 desc
```

```
-- which countries are served + what are their revenues?
SELECT
        DISTINCT country,
        SUM(amount)
FROM payment A
LEFT JOIN customer B    ON A.customer_id =    B.customer_id
RIGHT JOIN address C    ON B.address_id =     C.address_id
LEFT JOIN city D        ON C.city_id =        D.city_id
LEFT JOIN country E     ON D.country_id =     E.country_id
GROUP BY country
ORDER BY sum desc
```

```
-- TOP 10 movies by revenue
SELECT
        title,
        description,
        rating,
        SUM(A.amount) AS sum_amount
FROM payment A
LEFT JOIN rental B      ON A.rental_id =      B.rental_id
LEFT JOIN inventory C   ON B.inventory_id =   C.inventory_id
LEFT JOIN film D        ON C.film_id =        D.film_id
GROUP BY
        title,
        description,
        rating
ORDER BY sum_amount desc
LIMIT 10
```

```
-- BOTTOM 10 movies by revenue
SELECT
        title,
        description,
        rating,
        SUM(A.amount) AS sum_amount
FROM payment A
LEFT JOIN rental B          ON A.rental_id =          B.rental_id
LEFT JOIN inventory C       ON B.inventory_id =       C.inventory_id
LEFT JOIN film D            ON C.film_id =            D.film_id
GROUP BY
        title,
        description,
        rating
ORDER BY sum_amount
LIMIT 10

-- genres calc compiled (thriller outlier)
SELECT
        F.name,
        SUM(A.amount) AS sum_amount,
        COUNT(distinct D.film_id) AS number_of_films,
        COUNT(distinct C.inventory_id) AS inventory,
        MIN(D.rental_duration) AS min_rental_duration,
        MAX(D.rental_duration) AS min_rental_duration,
        ROUND(AVG(D.rental_duration),0) AS avg_rental_duration
FROM payment A
LEFT JOIN rental B          ON A.rental_id =          B.rental_id
LEFT JOIN inventory C       ON B.inventory_id =       C.inventory_id
LEFT JOIN film D            ON C.film_id =            D.film_id
LEFT JOIN film_category E ON D.film_id =              E.film_id
LEFT JOIN category F        ON E.category_id =        F.category_id
GROUP BY
        F.name
ORDER BY 2 desc

-- TOP 10 highest paying customers
SELECT
        B.customer_id,
        B.first_name,
        B.last_name,
        SUM(A.amount) AS total_amount,
        D.city,
        E.country
FROM payment A
INNER JOIN customer B ON A.customer_id = B.customer_id
INNER JOIN address C        ON B.address_id =         C.address_id
INNER JOIN city D           ON C.city_id =            D.city_id
INNER JOIN country E        ON D.country_id =            E.country_id
GROUP BY
            D.city,
            E.country,
            B.customer_id
ORDER BY total_amount desc
LIMIT 10
```

BACKGROUND INFO:

```
-- country + store IDs for two store locations
SELECT
        A.store_id,
        D.country_id,
        D.country
FROM store A
INNER JOIN address B    ON A.address_id =       B.address_id
INNER JOIN city C       ON B.city_id =          C.city_id
INNER JOIN country D    ON C.country_id =       D.country_id
GROUP BY
        A.store_id,
        D.country_id
```

```
-- payment.staff_id = store_id
SELECT *
FROM payment A
LEFT JOIN staff B       ON A.staff_id =   B.staff_id
LEFT JOIN store C       ON B.store_id =   C.store_id
```

```
-- what languages are rented by region? (English is the ONLY language rented)
SELECT *
FROM payment A
LEFT JOIN rental B      ON A.rental_id =        B.rental_id
LEFT JOIN inventory C   ON B.inventory_id =     C.inventory_id
LEFT JOIN film D        ON C.film_id =          D.film_id
LEFT JOIN language E    ON D.language_id =      E.language_id
ORDER BY name
```

```
-- how many countries are served by each store? (1=CAN=81; 2=AUS=82; overlapping countries for a total of 163 total)
SELECT
        COUNT(country),
        store_id
FROM
        (SELECT
                DISTINCT country,
                store_id
        FROM payment A
        LEFT JOIN customer B    ON A.customer_id =      B.customer_id
        RIGHT JOIN address C    ON B.address_id =       C.address_id
        LEFT JOIN city D        ON C.city_id =          D.city_id
        LEFT JOIN country E     ON D.country_id =       E.country_id
        ORDER BY store_id desc) AS regional_stores
GROUP BY store_id
```

NOT USING:

```
-- top 10 countries with most amount of customers (3.7.1)
SELECT
        D.country,
        COUNT(A.customer_id) AS total_customers
FROM customer A
INNER JOIN address B     ON A.address_id =        B.address_id
INNER JOIN city C        ON B.city_id =           C.city_id
INNER JOIN country D     ON C.country_id =        D.country_id
GROUP BY
        D.country
ORDER BY total_customers desc
LIMIT 10

-- top 10 cities within top 10 countries with most customers (3.7.2)
SELECT
        D.country,
        C.city,
        COUNT(A.customer_id) AS total_customers
FROM customer A
INNER JOIN address B     ON A.address_id =        B.address_id
INNER JOIN city C        ON B.city_id =           C.city_id
INNER JOIN country D     ON C.country_id =        D.country_id
WHERE country IN (
        'India',
        'China',
        'United States',
        'Japan',
        'Mexico',
        'Brazil',
        'Russian Federation',
        'Philippines',
        'Turkey',
        'Indonesia'
        )
GROUP BY
        D.country,
        C.city
ORDER BY total_customers desc
LIMIT 10

-- top 5 customers in top 10 cities who paid highest amount (3.7.3)
SELECT
        B.customer_id,
        B.first_name,
        B.last_name,
        SUM(A.amount) AS total_amount,
        D.city,
        E.country
FROM payment A
INNER JOIN customer B   ON A.customer_id =      B.customer_id
INNER JOIN address C    ON B.address_id =       C.address_id
INNER JOIN city D       ON C.city_id =          D.city_id
INNER JOIN country E    ON D.country_id =       E.country_id
WHERE city IN (
        'Aurora',
        'Acua',
        'Citrus Heights',
        'Iwaki',
        'Ambattur',
        'Shanwei',
        'So Leopoldo',
        'Teboksary',
        'Tianjin',
        'Cianjur'
        )
GROUP BY
        D.city,
        E.country,
        B.customer_id
ORDER BY total_amount desc
LIMIT 5

-- rental duration between ratings (min/max same; avg of NC-17 slightly higher than rest)
SELECT
        rating,
        MIN(rental_duration) AS min_rental_duration,
        MAX(rental_duration) AS min_rental_duration,
        AVG(rental_duration) AS avg_rental_duration
FROM payment A
LEFT JOIN rental B      ON A.rental_id =        B.rental_id
LEFT JOIN inventory C   ON B.inventory_id =     C.inventory_id
LEFT JOIN film D        ON C.film_id =          D.film_id
LEFT JOIN language E    ON D.language_id =      E.language_id
GROUP BY 1
ORDER BY 1

-- BOTTOM 5 movies by revenue of region 1=CAN
SELECT
        title,
        description,
        rating,
        SUM(A.amount) AS sum_amount
FROM payment A
LEFT JOIN rental B      ON A.rental_id =        B.rental_id
LEFT JOIN inventory C     ON B.inventory_id =       C.inventory_id
LEFT JOIN film D          ON C.film_id =            D.film_id
WHERE A.staff_id = 1
GROUP BY
        title,
        description,
        rating
ORDER BY sum_amount
LIMIT 5

-- BOTTOM 5 movies by revenue of region 2=AUS
SELECT
        title,
        description,
        rating,
        SUM(A.amount) AS sum_amount
FROM payment A
LEFT JOIN rental B        ON A.rental_id =          B.rental_id
LEFT JOIN inventory C     ON B.inventory_id =       C.inventory_id
LEFT JOIN film D          ON C.film_id =            D.film_id
WHERE A.staff_id = 2
GROUP BY
        title,
        description,
        rating
ORDER BY sum_amount
LIMIT 5

-- TOP 5 movies by revenue of region 1=CAN
SELECT
        title,
        description,
        rating,
        SUM(A.amount) AS sum_amount
FROM payment A
LEFT JOIN rental B        ON A.rental_id =          B.rental_id
LEFT JOIN inventory C     ON B.inventory_id =       C.inventory_id
LEFT JOIN film D          ON C.film_id =            D.film_id
WHERE A.staff_id = 1
GROUP BY
        title,
        description,
        rating
ORDER BY sum_amount desc
LIMIT 5
```

```
-- TOP 5 movies by revenue of region 2=AUS
SELECT
        title,
        description,
        rating,
        SUM(A.amount) AS sum_amount
FROM payment A
LEFT JOIN rental B          ON A.rental_id =          B.rental_id
LEFT JOIN inventory C       ON B.inventory_id =       C.inventory_id
LEFT JOIN film D            ON C.film_id =            D.film_id
WHERE A.staff_id = 2
GROUP BY
        title,
        description,
        rating
ORDER BY sum_amount desc
LIMIT 5
```

```
-- which countries are served by each store location? (1=CAN; 2=AUS)
SELECT
        DISTINCT country,
        store_id
FROM payment A
LEFT JOIN customer B        ON A.customer_id =        B.customer_id
RIGHT JOIN address C        ON B.address_id =         C.address_id
LEFT JOIN city D            ON C.city_id =            D.city_id
LEFT JOIN country E         ON D.country_id =         E.country_id
ORDER BY store_id desc
```

```
-- how much $ coming in from each store location (staff_id = store_id)
-- not much difference (1=CAN=30,252.12; 2=AUS=31,059.92)
SELECT
        staff_id,
        SUM(amount)
FROM payment
GROUP BY staff_id
```

```
-- what rating is rented the most/least?
SELECT
        rating,
        COUNT(rating) AS count_of_rating
FROM payment A
LEFT JOIN rental B          ON A.rental_id =          B.rental_id
LEFT JOIN inventory C       ON B.inventory_id =       C.inventory_id
LEFT JOIN film D            ON C.film_id =            D.film_id
LEFT JOIN language E        ON D.language_id =        E.language_id
GROUP BY 1
ORDER BY 2 desc;
```

```
-- count of movies per genre
SELECT
        F.name,
        COUNT(distinct D.film_id) AS number_of_films,
        COUNT(distinct C.inventory_id) AS inventory
FROM payment A
LEFT JOIN rental B              ON A.rental_id =          B.rental_id
LEFT JOIN inventory C           ON B.inventory_id =       C.inventory_id
LEFT JOIN film D                ON C.film_id =            D.film_id
LEFT JOIN film_category E       ON D.film_id =            E.film_id
LEFT JOIN category F            ON E.category_id =        F.category_id
GROUP BY
        F.name
ORDER BY 2 desc

-- genres by revenue
SELECT
        F.name,
        SUM(A.amount) AS sum_amount
FROM payment A
LEFT JOIN rental B              ON A.rental_id =          B.rental_id
LEFT JOIN inventory C           ON B.inventory_id =       C.inventory_id
LEFT JOIN film D                ON C.film_id =            D.film_id
LEFT JOIN film_category E       ON D.film_id =            E.film_id
LEFT JOIN category F            ON E.category_id =        F.category_id
GROUP BY
        F.name
ORDER BY sum_amount desc

-- inventory count by genre
SELECT
        F.name,
        COUNT(distinct C.inventory_id) AS inventory
FROM payment A
LEFT JOIN rental B              ON A.rental_id =          B.rental_id
LEFT JOIN inventory C           ON B.inventory_id =       C.inventory_id
LEFT JOIN film D                ON C.film_id =            D.film_id
LEFT JOIN film_category E       ON D.film_id =            E.film_id
LEFT JOIN category F            ON E.category_id =        F.category_id
GROUP BY
        F.name

-- rental duration between 2 regions (min/max same 3-7; avg 4.9)
SELECT
        A.staff_id,
        MIN(rental_duration) AS min_rental_duration,
        MAX(rental_duration) AS min_rental_duration,
        AVG(rental_duration) AS avg_rental_duration
```

```
FROM payment A
LEFT JOIN rental B          ON A.rental_id =          B.rental_id
LEFT JOIN inventory C       ON B.inventory_id =       C.inventory_id
LEFT JOIN film D            ON C.film_id =            D.film_id
LEFT JOIN language E        ON D.language_id =        E.language_id
GROUP BY 1
ORDER BY 1

-- what rating is rented the most/least in region 1 / CAN?
SELECT
        A.staff_id,
        rating,
        COUNT(rating) AS count_of_rating
FROM payment A
LEFT JOIN rental B          ON A.rental_id =          B.rental_id
LEFT JOIN inventory C       ON B.inventory_id =       C.inventory_id
LEFT JOIN film D            ON C.film_id =            D.film_id
LEFT JOIN language E        ON D.language_id =        E.language_id
WHERE A.staff_id = 1
GROUP BY 1, 2
ORDER BY 3 desc;

-- what rating is rented the most/least in region 2 / AUS?
SELECT
        A.staff_id,
        rating,
        COUNT(rating) AS count_of_rating
FROM payment A
LEFT JOIN rental B          ON A.rental_id =          B.rental_id
LEFT JOIN inventory C       ON B.inventory_id =       C.inventory_id
LEFT JOIN film D            ON C.film_id =            D.film_id
LEFT JOIN language E        ON D.language_id =        E.language_id
WHERE A.staff_id = 2
GROUP BY 1, 2
ORDER BY 3 desc
```

** not much difference between two regions –
Region 1: PG13, NC17, R, PG, G
Region 2: PG13, NC17, PG, R, G

```
-- ALL genres calc compiled (thriller outlier) for region 1=CAN
SELECT
        F.name,
        SUM(A.amount) AS sum_amount,
        COUNT(distinct D.film_id) AS number_of_films,
        COUNT(distinct C.inventory_id) AS inventory,
        MIN(D.rental_duration) AS min_rental_duration,
        MAX(D.rental_duration) AS min_rental_duration,
         ROUND(AVG(D.rental_duration),0) AS avg_rental_duration
FROM payment A
LEFT JOIN rental B         ON A.rental_id =           B.rental_id
LEFT JOIN inventory C      ON B.inventory_id =        C.inventory_id
LEFT JOIN film D           ON C.film_id =             D.film_id
LEFT JOIN film_category E  ON D.film_id =             E.film_id
LEFT JOIN category F       ON E.category_id =         F.category_id
WHERE A.staff_id = 1
GROUP BY
         F.name
ORDER BY 2 desc

-- ALL genres calc compiled (thriller outlier) for region 2=AUS
SELECT
         F.name,
         SUM(A.amount) AS sum_amount,
         COUNT(distinct D.film_id) AS number_of_films,
         COUNT(distinct C.inventory_id) AS inventory,
         MIN(D.rental_duration) AS min_rental_duration,
         MAX(D.rental_duration) AS min_rental_duration,
         ROUND(AVG(D.rental_duration),0) AS avg_rental_duration
FROM payment A
LEFT JOIN rental B                  ON A.rental_id =           B.rental_id
LEFT JOIN inventory C               ON B.inventory_id =        C.inventory_id
LEFT JOIN film D                    ON C.film_id =             D.film_id
LEFT JOIN film_category E           ON D.film_id =             E.film_id
LEFT JOIN category F                ON E.category_id =         F.category_id
WHERE A.staff_id = 2
GROUP BY
         F.name
ORDER BY 2 desc

-- relationship btwn rating + genre (none)
SELECT
         rating,
         name
FROM film D
LEFT JOIN film_category E  ON D.film_id =             E.film_id
LEFT JOIN category F       ON E.category_id =         F.category_id
ORDER BY rating, name

-- number of transactions per rental rate
SELECT
         rental_rate,
         COUNT(payment_id) AS count_transaction
FROM payment A
INNER JOIN rental B        ON A.customer_id =         B.customer_id
INNER JOIN inventory C     ON B.inventory_id =        C.inventory_id
INNER JOIN film D         ON C.film_id =            D.film_id
GROUP BY
          rental_rate
ORDER BY count_transaction desc

-- number of transactions per rental rate in region 1=CAN
SELECT
          A.staff_id,
          rental_rate,
          COUNT(payment_id) AS count_transaction
FROM payment A
INNER JOIN rental B       ON A.customer_id =        B.customer_id
INNER JOIN inventory C    ON B.inventory_id =       C.inventory_id
INNER JOIN film D         ON C.film_id =            D.film_id
WHERE A.staff_id = 1
GROUP BY
          A.staff_id,
          rental_rate
ORDER BY count_transaction desc

-- number of transactions per rental rate in region 2=AUS
SELECT
          A.staff_id,
          rental_rate,
          COUNT(payment_id) AS count_transaction
FROM payment A
INNER JOIN rental B       ON A.customer_id =        B.customer_id
INNER JOIN inventory C    ON B.inventory_id =       C.inventory_id
INNER JOIN film D         ON C.film_id =            D.film_id
WHERE A.staff_id = 2
GROUP BY
          A.staff_id,
          rental_rate
ORDER BY count_transaction desc
```

Tableau URL:

https://public.tableau.com/app/profile/akiko.oguchi/viz/revenuextransactions/Sheet1
https://public.tableau.com/app/profile/akiko.oguchi/viz/bottom10wordcount/Sheet1
https://public.tableau.com/app/profile/akiko.oguchi/viz/topbottom10movies/Sheet1
https://public.tableau.com/app/profile/akiko.oguchi/viz/top10customers_16886141905570/Sheet1
https://public.tableau.com/app/profile/akiko.oguchi/viz/genreranking/Sheet1
https://public.tableau.com/app/profile/akiko.oguchi/viz/wordcount_16886052839560/Sheet1
https://public.tableau.com/app/profile/akiko.oguchi/viz/countriesserved/Sheet1
https://public.tableau.com/app/profile/akiko.oguchi/viz/globalrevenue_16885877346130/Sheet1