<table>
<tr><th>Question</th><th>Query</th><th>Interpretation</th></tr>
<tr><td>1. What is the company's revenue per month? Compare each month value with the average revenue per month.</td><td>

```
SELECT strftime('%m', p.payment_date) as month,
   sum(p.amount) as revenue_per_month,
   count (distinct customer_id) as nbr_customers,
   count (distinct payment_id) as nbr_transactions,
   CASE WHEN sum(p.amount) < avg_revenue_per_month THEN 'below'
      WHEN sum(p.amount) = avg_revenue_per_month THEN 'equal'
      else 'higher' end as compare_with_average
FROM payment p,
   (SELECT sum(p.amount)/count(distinct strftime('%m', p.payment_date)) as
avg_revenue_per_month
   FROM payment p)
GROUP BY 1
ORDER BY 1;
```

</td><td>The result shows that the company has seen an increase in the number of customers and revenue per month across four months. Compared to the average revenue per month, the company improved revenue thanks to attracting numerous customers. However, the ratio of sales per customer decreased in August due to a decrease in revenue.</td></tr>
<tr><td></td><td>

	month	revenue per month	nbr customers	nbr transactions	compare with average
1	02	514.180000000001	158	182	below
2	05	4824.429999999861	520	1157	below
3	06	9631.87999999961	590	2312	below
4	07	28373.890000003783	599	6711	higher
5	08	24072.130000002162	599	5687	higher

</td><td></td></tr>
<tr><td>2.How many customers does the company have?</td><td>

```
SELECT 'Total: ' as Notes, count(*) as Number_of_customers
FROM customer
UNION
SELECT 'Active customers: ', count(*)
FROM customer
WHERE active = 1
UNION
SELECT 'Non-active customers: ', count(*)
FROM customer
WHERE active = 0;
```

</td><td>The company has a total of 599 customers, of which there are 584 active customers and 15 customers who currently were not interested in renting DVDs from the company.</td></tr>
<tr><td></td><td>

Notes	Number of customer:
Active customers:	584
Non-active customers:	15
Total:	599

</td><td></td></tr>
<tr><td>3.The objective of this question is to separate customers</td><td>

```
SELECT frequency, count(*) as nbr_customers,
sum(number_of_purchases)/count(frequency) as avg_number_of_purchases
FROM (select p.customer_id,
   count(*) as number_of_purchases,
   CASE WHEN count(*) > avg_total_purchases and active = 1 THEN 'regular'
```

</td><td>The company has 312 non-regular customers and 287 regular customers. The average number of purchases</td></tr>
</table>

based on the number of purchases and active. If the time of purchases of a customer is greater than the average purchases per customer, then the customer is regular, else the customer is non-regular. How may customer in its type? Calculate average purchases per type.

```
   ELSE 'non-regular' END as frequency
   FROM payment p, customer cr,
   (SELECT count(*)/count(distinct p.customer_id) as avg_total_purchases
   from payment p)
WHERE p.customer_id = cr.customer_id
GROUP BY 1)
GROUP BY 1;
```

frequency	nbr customers	avg number of purchases
non-regular	312	23
regular	287	30

for each type is 23 and 30, respectively.

4.Create a new column called loyalty to divide customers into three groups based on how long the customers have been with the company (rely on '2016-01-01' and the first payment date of each customer). If the tenure is less than eight months and the customer still rents DVD, then the customer is

```
SELECT loyalty, tenure, count(distinct customer_id) as nuber_of_customers
FROM (select p.customer_id, active,
   (strftime('%m', '2006-01-01') - min(strftime('%m',p.payment_date))+ 12) as tenure,
   CASE WHEN (strftime('%m', '2006-01-01') - min(strftime('%m',p.payment_date))+ 12) <= 8  and cr.active = 1 THEN 'new'
      WHEN (strftime('%m', '2006-01-01') - min(strftime('%m',p.payment_date))+ 12) > 8  and cr.active = 1 THEN 'loyal'
      else 'disloyal' end as loyalty
      FROM payment p, customer cr
      WHERE p.customer_id = cr.customer_id
      GROUP BY 1)
GROUP BY 1, 2;
```

	loyalty	tenure	nuber of customers
1	disloyal	7	1
2	disloyal	8	10
3	disloyal	11	4
4	loyal	11	154
5	new	6	1
6	new	7	58
7	new	8	371

The tenure of a customer ranges from 6 months to 11 months. Overall, there are 15 disloyal customers, 154 loyal customers and 430 new customers.

new; if the term is greater than eight months and the customer still rents DVDs, the customer is loyal. If the customer is no longer rents DVDs, then the customer is disloyal. Show the number of customers for each type of loyalty and tenure.

5.What is the total revenue, number of customers, average revenue, and percentage of revenue of 10 countries with the highest sales? Order the result by total revenue in descending.

```
SELECT c.country, sum(p.amount) as revenue_per_country,
      count (distinct p.customer_id) as nbr_customers,
      sum(p.amount)/total as pct_of_revenue
FROM payment p, customer cr, address a, city cy, country c,
   (SELECT sum(p.amount) as total FROM payment p)
WHERE p.customer_id = cr.customer_id and
   cr.address_id = a.address_id and
   a.city_id = cy.city_id and
   cy.country_id = c.country_id
GROUP BY 1
ORDER BY 2 desc
LIMIT 10;
```

The top 10 countries are India, China, the US, Japan, Mexico, Brazil, Russian, Philippines, Turkey, Nigeria account for more than 50% of the revenue of Sakila company. Of which, India has the highest total revenue accounts for 9.8% of Sakila's revenue.

	country	revenue_per_country	nbr_customers	pct_of_revenue
1	India	6630.26999999977	60	0.09834786760692
2	China	5802.729999999806	53	0.08607283290103
3	United States	4110.3199999999	36	0.06096904155971
4	Japan	3471.7399999999293	31	0.05149688110524
5	Mexico	3307.039999999941	30	0.04905385935879
6	Brazil	3200.5199999999454	28	0.04747383096515
7	Russian Federation	3045.8699999999517	28	0.04517988249466
8	Philippines	2381.3199999999874	20	0.03532250482857
9	Turkey	1662.120000000003	15	0.02465449487077
10	Nigeria	1511.4800000000027	13	0.02242002737905

6.What is the total revenue, number of customers, average revenue, and percentage of revenue of each market? Order the result by total revenue in descending.	**CREATE TABLE** continent(continent_id text primary key, continent_name text not null; **INSERT INTO** continent(continent_id, continent_name) values ('AF', 'Africa'), ('AS', 'Asia'), ('EU', 'Europe'), ('NA', 'North America'), ('SA', 'South America'), ('OC', 'Oceania'), ('AN', 'Antarctica'); **CREATE TABLE** market(country_id SMALLINT NOT NULL primary key, country VARCHAR (50) NOT NULL, continent_code text); **INSERT INTO MARKET**(country_id, country, continent_code) values (1, 'Afghanistan', 'AS'), (2, 'Algeria', 'AF'), (3, 'American Samoa', 'OC'), (4, 'Angola', 'AF'), …… (107, 'Yemen', 'AS'), (108, 'Yugoslavia', 'EU'), (109, 'Zambia', 'AF'); **SELECT** co.continent_name, sum(p.amount) as revenue_per_market, count (distinct p.customer_id) as nbr_customers, sum(p.amount)/count(distinct p.customer_id) as avg_revenue_per_customer, sum(p.amount)/total as pct_of_revenue **FROM** payment p, customer cr, address a, city cy, continent co, market m, (**SELECT** sum(p.amount) as total **FROM** payment p) **WHERE** p.customer_id = cr.customer_id and cr.address_id = a.address_id and a.city_id = cy.city_id and cy.country_id = m.country_id and m.continent_code = co.continent_id **GROUP BY** 1 **ORDER BY** 2 desc;	The biggest market of Sakila is Asia, the revenue of this market is 30,610, represents for 45% of total revenue. Asia also has the highest number of customers. Europe follows Asia with the revenue of 12,120, accounts for 18% of total revenue. North America, South America, Africa, Oceania sort in third, fourth, fifth and sixth position respectively. However, Africa has the highest revenue per customer, which is 114.8. The second, third, fourth, fifth and sixth position are North America with 113, Asia with 112.5, Europe with 113, South America with 111.1 and Oceania with 106.

	continent name	revenue per market	nbr customers	avg revenue per customer	pct of revenue
1	Asia	30610.500000004628	272	112.53860294119349	0.4540504989061
2	Europe	12120.389999999485	108	112.22583333332857	0.17978370580146
3	North America	9047.34999999964	80	113.0918749999955	0.13420080630102
4	South America	7777.69999999971	70	111.10999999999585	0.11536788243712
5	Africa	7117.299999999744	62	114.79516129031845	0.10557206239244
6	Oceania	743.2700000000011	7	106.18142857142873	0.01102504416203

7.What is the total sales of the most selling film category and the least selling film category?

```
SELECT 'The most selling film category: ' || category as Category,
max(total_sales) as sales
FROM (SELECT c.name as category, sum(p.amount) as total_sales
   FROM payment p, rental r, inventory i, film f, film_category fc, category c
   WHERE p.rental_id = r.rental_id  and
   r.inventory_id = i.inventory_id and
   i.film_id = f.film_id and
   f.film_id = fc.film_id and
   fc.category_id = c.category_id
GROUP BY c.name)
UNION
SELECT 'The least selling film category: ' || category as Category,
min(total_sales)
FROM (SELECT c.name as category, sum(p.amount) as total_sales
   FROM payment p, rental r, inventory i, film f, film_category fc, category c
   WHERE p.rental_id = r.rental_id  and
   r.inventory_id = i.inventory_id and
   i.film_id = f.film_id and
   f.film_id = fc.film_id and
   fc.category_id = c.category_id
GROUP BY c.name)
```

The most selling category film is Sport, with the total sales is 5314. The least selling category film is Music, with the total sales is 3417.

	Category	sales
1	The least selling film category: Music	3417.7199999999325
2	The most selling film category: Sports	5314.209999999847

8.What are the total sales for each rating of the month that has the highest sales?

```
SELECT f.rating, SUM(p.amount) as total_sales
   FROM payment p, rental r, inventory i, film f
   WHERE p.rental_id = r.rental_id  and
   r.inventory_id = i.inventory_id and
   i.film_id = f.film_id
GROUP BY 1
HAVING strftime('%m', p.payment_date) = (SELECT month
                                 FROM(select strftime('%m', p.payment_date) as
month,
                                    sum(p.amount) as revenue_per_month
                                    from payment p
                                    GROUP BY month
```

The total sales for rating G in month that has the highest sales is approximately 11,664. There is only one type of rating in this month.

<table>
<tr><td></td><td>

```
ORDER BY 2 desc
LIMIT 1));
```

</td><td></td></tr>
<tr><td></td><td>

rating	total_sales
G	11664.229999999505

</td><td></td></tr>
<tr><td rowspan="2">9.Categorize actor based on the ratio between total sales per actor and the average sales per actor. If the ratio greater than 1 then the actor has good performing, if the ratio equals 1 then the actor has average performing, else the actor has low performing. Calculate the number of actors per each type.</td><td>

```
SELECT performing, count (*) as nbr_actors
FROM (SELECT a.actor_id, a.first_name || ' ' || a.last_name as actor,
sum(p.amount) as total_sales,
   CASE WHEN round(sum(p.amount)/avg_sales_per_actor, 1) > 1 THEN
'good'
      WHEN round(sum(p.amount)/avg_sales_per_actor, 1) = 1 THEN 'average'
      ELSE 'low' end as performing
FROM payment p, rental r, inventory i, film f, film_actor fa, actor a,
(SELECT sum(p.amount)/count(distinct a.actor_id) as avg_sales_per_actor
   FROM payment p, rental r, inventory i, film f, film_actor fa, actor a
   WHERE p.rental_id = r.rental_id  and
   r.inventory_id = i.inventory_id and
   i.film_id = f.film_id and
   f.film_id = fa.film_id and
   fa.actor_id = a.actor_id) as a
   where p.rental_id = r.rental_id  and
   r.inventory_id = i.inventory_id and
   i.film_id = f.film_id and
   f.film_id = fa.film_id and
   fa.actor_id = a.actor_id
GROUP BY 1)
GROUP BY 1
ORDER BY 2 desc;
```

</td><td rowspan="2">There are 78 actors with low performing, 50 actors with average performing and 72 actors with good performing.</td></tr>
<tr><td>

	performing	nbr_actors
1	low	78
2	good	72
3	average	50

</td></tr>
<tr><td>10.Are there movies that have no sales? Are there any characteristics related to these movies?</td><td>

```
SELECT f.title, f.rating, c.name as category, l.name as language, a.first_name
|| ' ' || a.last_name as actor
FROM film f
LEFT JOIN inventory i on f.film_id = i.film_id
LEFT JOIN rental r on i.inventory_id = r.inventory_id
LEFT JOIN payment p on r.rental_id = p.rental_id
INNER JOIN film_category fc on f.film_id = fc.film_id
INNER JOIN category c on fc.category_id = c.category_id
INNER JOIN language l on f.language_id = l.language_id
INNER JOIN film_actor fa on f.film_id = fa.film_id
INNER JOIN actor a on a.actor_id = fa.actor_id
GROUP BY 1
HAVING sum(p.amount) IS NULL;
```

</td><td>There are 42 movies that have no sales. All movies have the same language, which is English.</td></tr>
</table>

Total rows loaded: 42

	title	rating	category	language	actor
1	ALICE FANTASIA	NC-17	Classics	English	WOODY HOFFMAN
2	APOLLO TEEN	PG-13	Drama	English	MAE HOFFMAN
3	ARGONAUTS TOWN	PG-13	Animation	English	JULIA BARRYMORE
4	ARK RIDGEMONT	NC-17	Action	English	PARKER GOLDBERG
5	ARSENIC INDEPENDENCE	PG	Travel	English	CUBA ALLEN
6	BOONDOCK BALLROOM	NC-17	Travel	English	ED CHASE
7	BUTCH PANTHER	PG-13	New	English	CUBA OLIVIER
8	CATCH AMISTAD	G	Foreign	English	JOHNNY CAGE
9	CHINATOWN GLADIATOR	PG	New	English	UMA WOOD
10	CHOCOLATE DUCK	R	Foreign	English	JOE SWANK

11.Calculate the total sales for each store. The result should include store_id, store city and store country, manager name and total sales.

```
SELECT s.store_id,
   c.city||', '||cy.country as store,
   sf.first_name||' '||sf.last_name as manager,
   sum(p.amount) as total_sales
FROM payment p, rental r, inventory i, store s, address a, city c, country cy,
staff sf
WHERE p.rental_id = r.rental_id and
   r.inventory_id = i.inventory_id and
   i.store_id = s.store_id and
   s.address_id = a.address_id and
   a.city_id = c.city_id and
   c.country_id = cy.country_id and
   s.manager_staff_id = sf.staff_id
GROUP BY  1, 2, 3;
```

The total sales of the first store is approximately 33,679.79, lower than the total sales of the second store, which is 33,726.77.

	store_id	store	manager	total_sales
1	1	Lethbridge, Canada	Mike Hillyer	33679.790000004956
2	2	Woodridge, Australia	Jon Stephens	33726.770000005024