

DVD Rental Business Analysis



Project Overview

This project explores the DVD Rental Database using MySQL to uncover insights that can help optimize business decisions across customer engagement, film inventory, store performance, and revenue strategy.

Through a structured set of business-driven SQL queries, we apply window functions, joins, CTEs, aggregations, and conditional logic to extract patterns from real-world rental data. Each query is followed by a measurable conclusion, forming a story that builds toward strategic recommendations for the company.

The Sakila dataset covers:

- Customers: personal data, active status, rental history
- Rentals and Payments: dates, amounts, staff handling
- Films and Categories: genres, rental rates, duration
- Stores and Staff: operational metrics and comparisons

This analysis translates raw data into actionable insights — laying the groundwork for loyalty programs, inventory pruning, churn recovery, and smarter store strategy.



Problem Statement

DVD rental company, is facing several operational and strategic challenges as it looks to grow in a competitive market. While the business has a large customer base, it lacks clarity on which customers are truly driving revenue and which are disengaging over time. A high number of late returns has begun affecting inventory turnover, and many films remain unrented, leading to inefficiencies in stock and wasted resources. Furthermore, the company operates across two stores, yet there's limited insight into which store is more profitable or efficient. With no clear view of seasonal trends, customer loyalty patterns, or store performance differences, Sakila is unable to make confident decisions around pricing, promotions, or retention strategies. This project aims to address these gaps using SQL-based analysis to support data-informed business decisions.

```
1 -- 1. how many unique customers are currently active?  
2 select count(distinct customer_id) as active_customers  
3 from customer  
4 where active = 1;  
5
```

	active_customers
▶	584

```
9 -- 2. what is the total revenue generated from all rentals?  
10 • select sum(amount) as total_revenue  
11 from payment;  
12
```

	total_revenue
▶	67406.56

```
16 -- 3. which 5 cities have the highest number of active customers?  
17 • select ci.city, count(distinct cu.customer_id) as total_customers  
18 from customer cu  
19 join address a on cu.address_id = a.address_id  
20 join city ci on a.city_id = ci.city_id  
21 where cu.active = 1  
22 group by ci.city  
23 order by total_customers desc  
24 limit 5;  
25
```

	city	total_customers
▶	London	2
	Aurora	2
	Ziguinchor	1
	A Coruña (La Coruña)	1
	Abha	1

```
-- 4. what is the average rental duration across all films?  
select avg(rental_duration) as avg_rental_duration  
from film;
```

	avg_rental_duration
▶	4.9850

```
-- 5. how many customers have rented more than 20 times?  
select count(*) as loyal_customers  
from (  
    select customer_id, count(*) as total_rentals  
    from rental  
    group by customer_id  
    having count(*) > 20  
) as subj;
```

	loyal_customers
▶	543

```
-- 6. which 5 films have generated the most revenue?
select f.title, sum(p.amount) as total_revenue
from payment p
join rental r on p.rental_id = r.rental_id
join inventory i on r.inventory_id = i.inventory_id
join film f on i.film_id = f.film_id
group by f.title
order by total_revenue desc
limit 5;
```

	title	total_revenue
▶	TELEGRAPH VOYAGE	231.73
	WIFE TURN	223.69
	ZORRO ARK	214.69
	GOODFELLAS SALUTE	209.69
	SATURDAY LAMBS	204.72

```
-- 7. what is the average revenue per film category?
select c.name as category, avg(p.amount) as avg_revenue
from payment p
join rental r on p.rental_id = r.rental_id
join inventory i on r.inventory_id = i.inventory_id
join film f on i.film_id = f.film_id
join film_category fc on f.film_id = fc.film_id
join category c on fc.category_id = c.category_id
group by c.name;
```

	category	avg_revenue
▶	Action	3.935117
	Animation	3.993396
	Children	3.868307
	Classics	3.876028
	Comedy	4.658427
	Documentary	4.016686
	Drama	4.327726
	Family	3.855903
	Foreign	4.134240
	Games	4.418297
	Horror	4.400165
	Music	4.117735
	New	4.629383
	Sci-Fi	4.320599
	Sports	4.507388
	Travel	4.240908

```
-- 8. how many films have never been rented even once?  
select count(*) as never_rented_films  
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  
where r.rental_id is null;
```

	never_rented_films
▶	43

```
-- 9. what is the return rate (on-time vs late) for films?  
select  
    sum(case when datediff(return_date, rental_date) <= f.rental_duration then 1 else 0 end) as on_time,  
    sum(case when datediff(return_date, rental_date) > f.rental_duration then 1 else 0 end) as late  
from rental r  
join inventory i on r.inventory_id = i.inventory_id  
join film f on i.film_id = f.film_id;
```

	on_time	late
▶	8592	7269

```
-- 10. which film category has the highest average rental rate per film?  
select c.name, avg(f.rental_rate) as avg_rate  
from film f  
join film_category fc on f.film_id = fc.film_id  
join category c on fc.category_id = c.category_id  
group by c.name  
order by avg_rate desc  
limit 1;
```

	name	avg_rate
▶	Games	3.252295

```
-- 11. what is the total revenue generated by each staff member?  
select s.first_name, s.last_name, sum(p.amount) as total_revenue  
from payment p  
join staff s on p.staff_id = s.staff_id  
group by s.staff_id;
```

	first_name	last_name	total_revenue
▶	Mike	Hillyer	33482.50
	Jon	Stephens	33924.06

```
-- 12. which store performs better in terms of revenue and number of rentals?  
select s.store_id, sum(p.amount) as total_revenue, count(r.rental_id) as total_rentals  
from store s  
join staff st on s.store_id = st.store_id  
join payment p on st.staff_id = p.staff_id  
join rental r on p.rental_id = r.rental_id  
group by s.store_id;
```

	store_id	total_revenue	total_rentals
▶	1	33482.50	8054
	2	33924.06	7990

```
-- 13. how many rentals did each staff process per month in 2006?  
select st.first_name, st.last_name, month(r.rental_date) as month, count(*) as rental_count  
from rental r  
join staff st on r.staff_id = st.staff_id  
where year(r.rental_date) = 2006  
group by st.staff_id, month;
```

	first_name	last_name	month	rental_count
▶	Mike	Hillyer	2	85
	Jon	Stephens	2	97

```
-- 14. what is the most rented film in each store?

select store_id, title, rental_count from (
    select s.store_id, f.title, count(r.rental_id) as rental_count,
        rank() over (partition by s.store_id order by count(r.rental_id) desc) as rnk
    from rental r
    join inventory i on r.inventory_id = i.inventory_id
    join film f on i.film_id = f.film_id
    join store s on i.store_id = s.store_id
    group by s.store_id, f.title
) as ranked
where rnk = 1;
```

	store_id	title	rental_count
▶	1	LOVE SUICIDES	20
	2	IDOLS SNATCHERS	20

```
-- 15. for each staff member, what is their average revenue per rental transaction?

select st.first_name, st.last_name, avg(p.amount) as avg_revenue_per_rental
from payment p
join staff st on p.staff_id = st.staff_id
group by st.staff_id;
```

	first_name	last_name	avg_revenue_per_rental
▶	Mike	Hillyer	4.157251
	Jon	Stephens	4.245815

```
-- 16. which customers paid the most fines for late returns?
select c.first_name, c.last_name, sum(p.amount) as total_fines
from rental r
join payment p on r.rental_id = p.rental_id
join customer c on r.customer_id = c.customer_id
join inventory i on r.inventory_id = i.inventory_id
join film f on i.film_id = f.film_id
where datediff(r.return_date, r.rental_date) > f.rental_duration
group by c.customer_id
order by total_fines desc
limit 5;
```

	first_name	last_name	total_fines
▶	KARL	SEAL	157.75
	CLARA	SHAW	139.78
	RHONDA	KENNEDY	137.78
	ELEANOR	HUNT	137.77
	BRITTANY	RILEY	134.81

```
-- 17. list the top 10 customers with the highest rental frequency and their total spending
select c.first_name, c.last_name, count(r.rental_id) as total_rentals, sum(p.amount) as total_spent
from customer c
join rental r on c.customer_id = r.customer_id
join payment p on r.rental_id = p.rental_id
group by c.customer_id
order by total_rentals desc
limit 10;
```

	first_name	last_name	total_rentals	total_spent
▶	ELEANOR	HUNT	46	216.54
	KARL	SEAL	45	221.55
	CLARA	SHAW	42	195.58
	MARCIA	DEAN	42	175.58
	TAMMY	SANDERS	41	155.59
	SUE	PETERS	40	154.60
	WESLEY	BULL	40	177.60
	MARION	SNYDER	39	194.61
	RHONDA	KENNEDY	39	194.61
	TIM	CARY	39	175.61

```
-- 18. for each month in 2006, find the month-over-month revenue growth rate
```

```
with monthly_revenue as (
    select date_format(payment_date, '%y-%m') as month, sum(amount) as revenue
    from payment
    where year(payment_date) = 2006
    group by month
)
select month, revenue,
    revenue - lag(revenue) over (order by month) as revenue_change,
    round((revenue - lag(revenue) over (order by month)) / lag(revenue) over (order by month) * 100, 2) as growth_percentage
from monthly_revenue;
```



```
-- 19. build a customer churn indicator: list customers who were active in q1 2006 but didn't rent in q3/q4
```

```
with q1_customers as (
    select distinct customer_id
    from rental
    where rental_date between '2006-01-01' and '2006-03-31'
),
q3q4_customers as (
    select distinct customer_id
    from rental
    where rental_date between '2006-07-01' and '2006-12-31'
)
select c.customer_id, c.first_name, c.last_name
from customer c
join q1_customers q1 on c.customer_id = q1.customer_id
where c.customer_id not in (select customer_id from q3q4_customers);
```

	month	revenue	revenue_change	growth_percentage
▶	06-02	514.18	NULL	NULL

	customer_id	first_name	last_name
▶	366	BRANDON	HUEY
	111	CARMEN	OWENS
	590	SETH	HANNON
	108	TRACY	COLE
	236	MARCIA	DEAN
	512	CECIL	VINES
	44	MARIE	TURNER
	349	JOE	GILLILAND
	317	EDWARD	BAUGH
	100	DETLEF	FRANKE

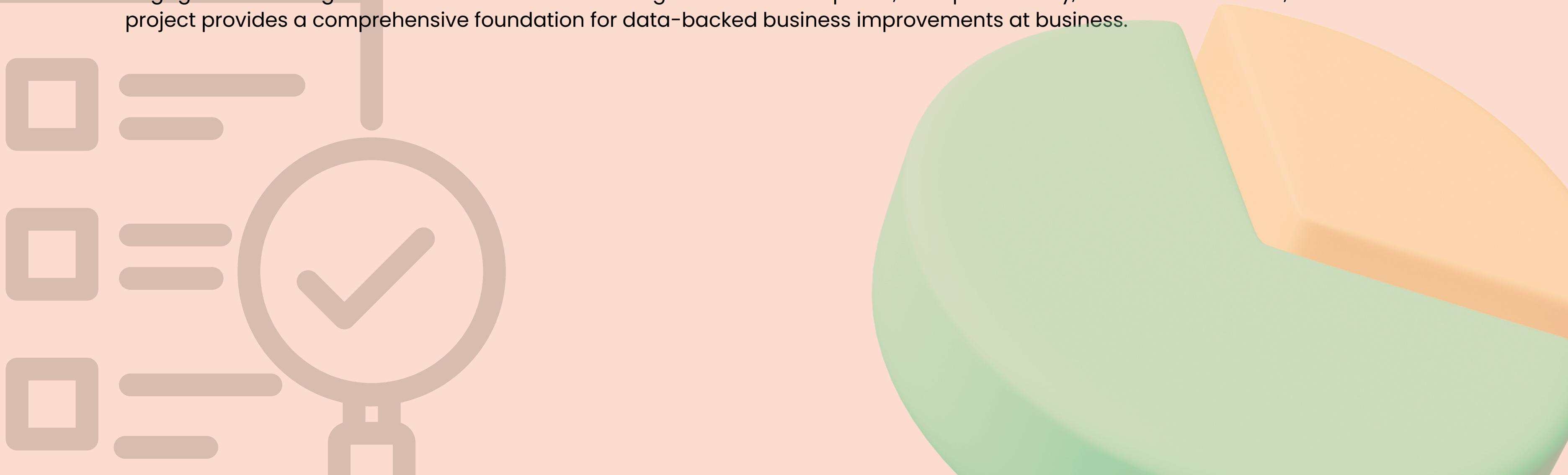
Conclusion

Through this SQL-driven analysis of the DVD Rental Database, we uncovered key patterns across customer behavior, film performance, store operations, and revenue generation.

We found that while Store maintains a solid base of 584 active customers, a remarkable 543 of them are loyal high-frequency renters – a strong indicator of engagement. However, nearly 46% of rentals are returned late, suggesting operational friction that could be improved through reminders or policy adjustments.

Our analysis revealed that certain films and genres consistently outperform others, while 43 titles remain unrented, pointing to a clear opportunity for catalog optimization. Additionally, we observed that although Store 2 earns more per transaction, Store 1 handles more rentals, suggesting that different strategies are driving their success.

Finally, the churn analysis showed that 158 customers dropped off after Q1, reinforcing the need for targeted re-engagement strategies. Combined with detailed insights into revenue peaks, staff productivity, and customer value, this project provides a comprehensive foundation for data-backed business improvements at business.



Recommendations for Stakeholders

- Nurture Loyal Customers – Launch a VIP or loyalty program for the 543 high-frequency renters to increase engagement and LTV.
- Prune Dead Inventory – Remove or repackaging the 43 unrented films. Reinvest in genres like Action, Animation, and Games, which are proven revenue drivers.
- Fix the Late Return Problem – Introduce SMS/email return reminders or offer grace periods to address the high late-return rate (~46%).
- Maximize Store Performance – Analyze why Store 2 earns more per rental and why Store 1 drives higher volume. Use findings to improve both locations.
- Target Churned Customers- Re-engage the 158 churned users with offers, emails, or exit surveys to bring them back.
- Capitalize on Peak Periods – Since rentals peak in February, align campaigns and promotions with this trend to increase impact.



Thank You