

ETL 100 - PYTHON / SQL SORGULARI

```
[1]: import pandas as pd
[2]: %load_ext sql
[3]: #sqlite ile veriyi bağlıyoruz.
%sql sqlite:///sakila_master.db
***

[6]: file_path = r"C:\Users\hadim\Downloads\sakila_master.db"

[7]: import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# bir sorgu denemesidir.
cursor = conn.cursor()

# veritabanında ki tablo isimlerini listeleyerek tabloya inceliyoruz.
cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
tables = cursor.fetchall()

# veritabanında ki tablo isimlerini yazdırıldı.
for table in tables:
    print(table[0])

# bu şekilde veritabanı hakkında biraz bilgi edinmiş olduk.

actor
country
city

[69]: import pandas as pd

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# veritabanındaki tabloyu sorguluyoruz 10 tanesini gösterecek şekilde.
df = pd.read_sql_query("SELECT * FROM rental LIMIT 10;", conn)

# Sonuçları yazdırma
print(df)
```

rental_id	rental_date	inventory_id	customer_id	\
0	1 2005-05-24 22:53:30.000	367	130	
1	2 2005-05-24 22:54:33.000	1525	459	
2	3 2005-05-24 23:03:39.000	1711	408	
3	4 2005-05-24 23:04:41.000	2452	333	
4	5 2005-05-24 23:05:21.000	2079	222	
5	6 2005-05-24 23:08:07.000	2792	549	
6	7 2005-05-24 23:11:53.000	3995	269	
7	8 2005-05-24 23:31:46.000	2346	239	
8	9 2005-05-25 00:00:40.000	2580	126	
9	10 2005-05-25 00:02:21.000	1824	399	

```
1) Tüm verileri görmek için;
```

```
Select * from rental
```

```
[72]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# satır ve sütunun düzgün görünmesi için
pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)

# veritabanındaki tabloyu sorguluyoruz 10 tanesini gösterecek şekilde.
df = pd.read_sql_query("SELECT * FROM rental", conn)

print(df)
```

	rental_id	rental_date	inventory_id	customer_id	\
0	1	2005-05-24 22:53:30.000	367	130	
1	2	2005-05-24 22:54:33.000	1525	459	
2	3	2005-05-24 23:03:39.000	1711	408	
3	4	2005-05-24 23:04:41.000	2452	333	
4	5	2005-05-24 23:05:21.000	2079	222	
...
16039	16045	2005-08-23 22:25:26.000	772	14	

```
2) Her filmde ki oyuncuları listelemek istersek;
```

```
SELECT
    film.title,
    actor.first_name,
    actor.last_name
FROM film
JOIN film_actor ON film.film_id = film_actor.film_id
JOIN actor ON film_actor.actor_id = actor.actor_id
```

```
[]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['film', 'actor', 'film_actor']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

oyuncu_listesi = df['film'].merge(df['film_actor'], on='film_id', how='inner') \
    .merge(df['actor'], on='actor_id', how='inner')

sonuc = oyuncu_listesi[['title', 'first_name', 'last_name']]
print(sonuc)

      title first_name last_name
0   ACADEMY DINOSAUR  PENELOPE  GUINNESS
1 ANACONDA CONFESSIONS  PENELOPE  GUINNESS
2       ANGELS LIFE  PENELOPE  GUINNESS
3  BULWORTH COMMANDMENTS  PENELOPE  GUINNESS
4     CHEAPER CLYDE  PENELOPE  GUINNESS
```

```

3) Her filmdeki oyuncu sayısını;

SELECT
    film.title,
    COUNT(actor.actor_id) AS actor_count
FROM film
JOIN film_actor ON film.film_id = film_actor.film_id
JOIN actor ON film_actor.actor_id = actor.actor_id
GROUP BY film.film_id, film.title
ORDER BY actor_count DESC

3]: import pandas as pd
import sqlite3

#sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['film', 'actor', 'film_actor']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

oyuncu_sayisi = df['film'].merge(df['film_actor'], on='film_id', how='inner') \
    .merge(df['actor'], on='actor_id', how='inner')

oyuncu = oyuncu_sayisi.groupby('title').size().reset_index(name='actor_count')

sonuc = oyuncu

print(sonuc)

      title  actor_count
0   ACADEMY DINOSAUR        10
1     ACE GOLDFINGER         4
2   ADAPTATION HOLES         5
3    AFFAIR PREJUDICE         5
4     AFRICAN EGG            5
...       ...

```

```

4) Her oyuncunun oynadığı film sayısını;

SELECT
    actor.first_name,
    actor.last_name,
    COUNT(film.film_id) AS movie_count
FROM actor
JOIN film_actor ON actor.actor_id = film_actor.actor_id
JOIN film ON film_actor.film_id = film.film_id
GROUP BY actor.actor_id, actor.first_name, actor.last_name
ORDER BY movie_count DESC

4]: import pandas as pd
import sqlite3

#sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['actor', 'film', 'film_actor']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

oyun_film = df['actor'].merge(df['film_actor'], on='actor_id', how='inner') \
    .merge(df['film'], on='film_id', how='inner')

sonuc = oyun_film.groupby(['first_name', 'last_name']).size().reset_index(name='movie_count')

a = sonuc

print(a)

      first_name  last_name  movie_count
0        ADAM      GRANT        18
1        ADAM      HOPPER        22
2         AL      GARLAND        26
3        ALAN     DREYFUSS        27
4      ALBERT  JOHANSSON        33

```

5) Envanterde olmayan filmlerin incelemesi ve varsa kaç tanesi olduğunu

```

SELECT COUNT(*) AS not_in_inventory
FROM film
WHERE film_id NOT IN (SELECT film_id FROM inventory)

```

[88]:

```

import pandas as pd
import sqlite3

# sqlite ile veritabanını bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['film', 'inventory']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

```

film_incelemesi = df['film'][~df['film']['film_id'].isin(df['inventory']['film_id'])].shape[0]

sonuc = pd.DataFrame({'film_incelemesi': [film_incelemesi]})

print(sonuc)

film_incelemesi
0 42

6) Kiralananabilir olan her filmin kaç kez kiralandığını ve toplam gelirlerini

```

SELECT
    film.title,
    COUNT(rental.rental_id) AS rental_count,
    SUM(payment.amount) AS total_revenue
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY film.title

```

[89]:

```

import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['film', 'inventory', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

kira = df['film'] \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby('title').agg(
        kiralama_sayisi=('rental_id', 'count'),
        toplam_gelir=('_amount', 'sum'))
    .reset_index()

print(kira)

```

	title	kiralama_sayisi	toplam_gelir
0	ACADEMY DINOSAUR	23	36.77
1	ACE GOLDFINGER	7	52.93
2	ADAPTATION HOLES	12	37.88
3	AFFAIR PREJUDICE	23	91.77
4	AFRICAN EGG	12	51.88
..
953	YOUNG LANGUAGE	7	6.93
954	YOUTH KICK	6	16.94

```
7) Envanterde olmayan filmlerin kira oranları;

SELECT
    title,
    rental_rate
FROM film
WHERE film_id NOT IN (
    SELECT film_id FROM inventory);
```

```
[90]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['film', 'inventory']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

kira_orani = df['film'][~df['film']['film_id'].isin(df['inventory']['film_id'])][['title', 'rental_rate']]

kira_orani.columns = ['filmler', 'kiralama_oranları']

print(kira_orani)

      filmler  kiralama_oranları
13      ALICE FANTASIA          0.99
32      APOLLO TEEN           2.99
35      ARGONAUTS TOWN          0.99
37      ARX RIDGEMONT          0.99
40      ARSENIC INDEPENDENCE        0.99
86      BOONDOCK BALLROOM         0.99
107     BUTCH PANTHER           0.99
127     CATCH AMISTAD            0.99
```

```
8) Birden fazla DVD'yi iade etmeyen müşteriler;
```

```
[74]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

customer_count = df['rental'] \
    .query('return_date.isnull()', engine='python') \
    .customer_id.unique()

iadesiz_musteri = pd.DataFrame({'müşteriler': [customer_count]})

print(iadesiz_musteri)

  müşteriler
0        159
```

```
9) Her müşterinin kiraladığı film sayısı;
```

```
SELECT customer.first_name, customer.last_name, COUNT(rental.rental_id) AS rental_count
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
GROUP BY customer.first_name, customer.last_name
```

```
[50]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['customer', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

kiralama_sayisi = df['customer'] \
    .merge(df['rental'], on='customer_id', how='left') \
    .groupby(['first_name', 'last_name']).agg(
        kiralomesi=('rental_id', 'count')) \
    .reset_index()

print(kiralama_sayisi)

   first_name last_name kiralomesi
0      AARON     SELBY       24
1      ADAM      GOOCH       22
2     ADRIAN     CLARY       19
3     AGNES    BISHOP       23
4      ALAN      KAHN       26
..      ...
594    WILLIE  MARKHAM       25
```

10) Türlerine göre en çok kiralanan filmler ve bunlara ödenen fiyat;

```
SELECT film.title AS film_title, category.name AS genre,
       COUNT(rental.rental_id) AS rental_count,
       SUM(payment.amount) AS total_revenue
FROM film
JOIN film_category ON film.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
JOIN payment ON rentals.rental_id = payment.rental_id
GROUP BY film.title, category.name
ORDER BY rental_count DESC
```

```
5]: import pandas as pd
import sqlite3

# sqlite file veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['film', 'film_category', 'category', 'inventory', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

odeme_fiyati = df['film'] \
    .merge(df['film_category'], on='film_id', suffixes=('', '_cat')) \
    .merge(df['category'], on='category_id', suffixes=('', '_cat2')) \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby(['title', 'name']).agg(
        kiralanmalari=('rental_id', 'count'),
        toplam_geliri=('amount', 'sum'))

    .reset_index().sort_values(by='kiralanmalari', ascending=False)
print(odeme_fiyati)
```

```

11) Tür ve tarihe göre kiralama sayıları ile gelirleri;

SELECT category.name AS genre,
       DATE(rental.rental_date) AS rental_date,
       COUNT(rental.rental_id) AS rental_count,
       SUM(payment.amount) AS total_revenue
FROM category
JOIN film_category ON category.category_id = film_category.category_id
JOIN film ON film_category.film_id = film.film_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY genre, rental_date
ORDER BY rental_date

```

```

66]: import pandas as pd
import sqlite3
#sqlite ile veritabanınıza bağlanıyoruz.
conn = sqlite3.connect(file_path)
#dataframe listesi oluşturma
df = {}
for table_name in ['film', 'film_category', 'category', 'inventory', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

tür_tarih_kiralaması = df['category'] \
    .merge(df['film_category'], on='category_id', suffixes=('', '_cat')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .assign(rental_date=lambda x: pd.to_datetime(x['rental_date'])) \
    .groupby(['name', 'rental_date']).agg(
        kiralama_sayıları=('rental_id', 'count'),
        toplam_gelirleri=('amount', 'sum'))

    ).reset_index().sort_values(by='rental_date')
print(tür_tarih_kiralaması)

```

	name	rental_date	kiralama_sayıları	toplam_gelirleri
7142	Family	2005-05-24 22:53:30	1	2.99
11039	Music	2005-05-24 22:54:33	1	2.99

```

12) Kiralanan filmlerin türlerine göre kaç kere kiralandıkları;

SELECT category.name AS genre, film.title, |
       COUNT(rental.rental_id) AS rental_count
FROM category
JOIN film_category ON category.category_id = film_category.category_id
JOIN film ON film_category.film_id = film.film_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY category.name, film.title
ORDER BY rental_count DESC

```

```

|: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['film', 'film_category', 'category', 'inventory', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
film_kiralaması = df['category'] \
    .merge(df['film_category'], on='category_id', suffixes=('', '_cat')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .groupby(['name', 'title']).agg(
        türlerin_kiralaması=('rental_id', 'count'))

    ).reset_index()
print(film_kiralaması)

      name          title  türlerin_kiralaması
0   Action     ACADEMY DINOSAUR           21
1   Action     AMERICAN CIRCUS           22
2   Action  ANTIKIDNEY TOMATOES           18
3   Action  BAREFOOT MANCHURIAN           18
4   Action    BERETS AGENT           21
..   ...

```

13) En çok rafda bekleyen filmler;

```

SELECT film.title,
       CAST(JULIANDAY('now') - JULIANDAY(rental.rental_date) AS INTEGER) AS days_on_shelf
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
WHERE rental.return_date IS NULL
ORDER BY days_on_shelf DESC

```

```

|: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe sözlüğü oluşturma
df = {}

for table_name in ['film', 'inventory', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

# kiralanmış, iade edilmemiş filmler
rafta_bekleyen_filmser = df['film'] \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id', how='left')

rafta_bekleyen_filmser = rafta_bekleyen_filmser[rafta_bekleyen_filmser['return_date'].isnull()]

# raf hesaplanması
rafta_bekleyen_filmser = rafta_bekleyen_filmser.assign(
    gun_sayisi=lambda x: (pd.to_datetime('now') - pd.to_datetime(x['rental_date'])).dt.days
).sort_values(by='gun_sayisi', ascending=False)

print(rafta_bekleyen_filmser[['title', 'gun_sayisi']])

      title  gun_sayisi
17      ACADEMY DINOSAUR      7173.0
9692    MUMMY CREATURES      6996.0

```

```

14) Erken, geç ve zamanında iade edilen kiralanmış filmler;

SELECT
    SUM(CASE WHEN return_date > DATE(rental_date, '+7 days') THEN 1 ELSE 0 END) AS late_returns,
    SUM(CASE WHEN return_date < DATE(rental_date, '-7 days') THEN 1 ELSE 0 END) AS early_returns,
    SUM(CASE WHEN return_date = DATE(rental_date, '+7 days') THEN 1 ELSE 0 END) AS on_time_returns
FROM rental
WHERE return_date IS NOT NULL

```

```

108]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

rental_data = df['rental']

rental_data['rental_date'] = pd.to_datetime(rental_data['rental_date'])
rental_data['return_date'] = pd.to_datetime(rental_data['return_date'])

gecikenler = rental_data[rental_data['return_date'] > (rental_data['rental_date'] + pd.Timedelta(days=7))].shape[0]
erken_iade_edenler = rental_data[rental_data['return_date'] < (rental_data['rental_date'] + pd.Timedelta(days=7))].shape[0]
zamaninda_iade_edenler = rental_data[rental_data['return_date'] == (rental_data['rental_date'] + pd.Timedelta(days=7))].shape[0]

print(f"Gecikmeli iade eden kişiler: {gecikenler}")
print(f"Erken iade eden kişiler: {erken_iade_edenler}")
print(f"Zamanında iade eden kişiler: {zamaninda_iade_edenler}")

```

Gecikmeli iade eden kişiler: 4494
 Erken iade eden kişiler: 11364

15) En çok DVD kiralayan müşteri;

```

SELECT customer.first_name, customer.last_name, COUNT(rental.rental_id) AS rental_count
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
GROUP BY customer.first_name, customer.last_name
ORDER BY rental_count DESC
LIMIT 1

```

```

107]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['customer', 'rental', 'inventory']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_dvd_kiralayan_musteri = df['customer'] \
    .merge(df['rental'], on='customer_id', how='left') \
    .merge(df['inventory'], on='inventory_id') \
    .groupby(['first_name', 'last_name']).agg(
        kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='kiralama_sayisi', ascending=False).head(1)

print(max_dvd_kiralayan_musteri)

```

	first_name	last_name	kiralama_sayisi
175	ELEANOR	HUNT	46

```

16) En popüler film kategorisi;

SELECT category.name, COUNT(rental.rental_id) AS rental_count
FROM category
JOIN film_category ON category.category_id = film_category.category_id
JOIN film ON film_category.film_id = film.film_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY category.name
ORDER BY rental_count DESC
LIMIT 1

11]: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}

for table_name in ['category', 'film_category', 'film', 'inventory', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

populer_kategori = df['category'] \
    .merge(df['film_category'], on='category_id', suffixes=('_category', '_film_category')) \
    .merge(df['film'], on='film_id', suffixes=('_film_category', '_film')) \
    .merge(df['inventory'], on='film_id', suffixes=('_film', '_inventory')) \
    .merge(df['rental'], on='inventory_id', suffixes=('_inventory', '_rental')) \
    .groupby('name').agg(
        kiralama_sayisi=('rental_id', 'count')

    ).reset_index() \
    .sort_values(by='kiralama_sayisi', ascending=False).head(1)
print(populer_kategori)

      name   kiralama_sayisi
14  Sports           1179

17) En çok kiralama işlemi yapan çalışan;

SELECT staff.first_name, staff.last_name, COUNT(rental.rental_id) AS rental_count
FROM staff
JOIN rental ON staff.staff_id = rental.staff_id
GROUP BY staff.first_name, staff.last_name
ORDER BY rental_count DESC
LIMIT 1

16]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['staff', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_kiralama_yapan_kisi = df['staff'] \
    .merge(df['rental'], on='staff_id', how='left') \
    .groupby(['first_name', 'last_name']).agg(
        kiralama_sayisi=('rental_id', 'count'))
    .reset_index() \
    .sort_values(by='kiralama_sayisi', ascending=False).head(1)

print(max_kiralama_yapan_kisi)

      first_name last_name   kiralama_sayisi
1       Mike      Hiller          8848

```

```

18) En çok gelir getiren film;

SELECT film.title, SUM(payment.amount) AS total_revenue
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY film.title
ORDER BY total_revenue DESC
LIMIT 1

```

```

124]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['film', 'inventory', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_gelir = df['film'] \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inventory')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby('title').agg(toplam_gelir = ('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='toplam_gelir', ascending=False) \
    .head(1)

print(max_gelir)

```

	title	toplam_gelir
841	TELEGRAPH VOYAGE	231.73

```

19) Her müşterinin toplam harcama miktarı;

SELECT customer.first_name AS musteri_ad,
       customer.last_name AS musteri_soyad,
       SUM(payment.amount) AS toplam_harcama
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY customer.first_name, customer.last_name

```

```

2]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['customer', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

toplam_harcama_miktari = df['customer'] \
    .merge(df['rental'], on='customer_id') \
    .merge(df['payment'], on='rental_id') \
    .groupby(['first_name', 'last_name']).agg(toplam_harcama=('amount', 'sum')) \
    .reset_index()

print(toplam_harcama)

```

	first_name	last_name	toplam_harcama
0	AARON	SELBY	110.76
1	ADAM	GOOCH	101.78
2	ADRIAN	CLARY	74.81

```

20) Kategorilerin toplam kiralananmasi ve gelirleri;

SELECT category.name AS category_name,
       COUNT(rental.rental_id) AS rental_count,
       SUM(payment.amount) AS total_payment
FROM category
JOIN film_category ON category.category_id = film_category.category_id
JOIN film ON film_category.film_id = film.film_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY category.name

```

```

[1]: import pandas as pd
import sqlite3
# sqlite ile veritabanina baglaniyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi olusturma
df = {}
for table_name in ['category', 'film_category', 'film', 'inventory', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

kategorilerin_toplam_kirasi = df['category'] \
    .merge(df['film_category'], on='category_id', suffixes=('', '_film_category')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inventory')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby('name').agg(
        kiralanma_sayilari = ('rental_id', 'count'),
        toplam_gelirleri = ('amount', 'sum')
    ).reset_index()
print(kategorilerin_toplam_kirasi)

      name kiralanma_sayilari  toplam_gelirleri
0   Action           1112        4375.85
1 Animation          1166        4656.30
2 Children            945        3655.55

```

```

21) En uzun süre kirada kalmış filmler;
SELECT film.title,
       MAX(julianday(rental.return_date) - julianday(rental.rental_date)) AS max_rental_duration
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
WHERE rental.return_date IS NOT NULL
GROUP BY film.title
ORDER BY max_rental_duration DESC
LIMIT 10

```

```

3]: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['film', 'inventory', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(),
                                   columns=[column[0] for column in conn.execute(query).description])
# kiralanma süresi
film_kiralama = df['film'] \
    .merge(df['inventory'], on='film_id', suffixes=('_', '_inventory')) \
    .merge(df['rental'], on='inventory_id', suffixes=('_', '_rental'))
# iade edilenler
film_kiralama = film_kiralama[film_kiralama['return_date'].notnull()]
# kiralama süresini güne göre hesaplama
film_kiralama['kiralama_suresi'] = (
    pd.to_datetime(film_kiralama['return_date']) - pd.to_datetime(film_kiralama['rental_date']))
).dt.total_seconds() / 86400
# maksimum kiralama süresi
film_kiralama = film_kiralama.groupby('title').agg(
    max_kiralama_suresi=('kiralama_suresi', 'max')
).reset_index()
# max kiralanan 10 film
film_kiralama = film_kiralama.sort_values(by='max_kiralama_suresi', ascending=False).head(10)
print(film_kiralama)

      title  max_kiralama_suresi
400 HOLOCAUST HIGHBALL          9.249306

```

```
22) En az kiralanan 5 film;

SELECT film.title,
       COUNT(rental.rental_id) AS rental_count
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY film.title
ORDER BY rental_count ASC
LIMIT 5
```

```
137]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe Listesi oluşturma
df = {}

for table_name in ['film', 'inventory', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(),
                                   columns=[column[0] for column in conn.execute(query).description])

min_kiralananlar = df['film'] \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id') \
    .groupby('title').agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='kiralama_sayisi') \
    .head(5)

print(min_kiralananlar)
```

	title	kiralama_sayisi
558	MIXED DOORS	4
866	TRAIN BUNCH	4
378	HARDLY ROBBERS	4
585	MUSSOLINI SPOILERS	5

```
24) En fazla kazanç sağlayan 5 müsteri;

SELECT customer.first_name,
       customer.last_name,
       SUM(payment.amount) AS total_spent
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY customer.first_name, customer.last_name
ORDER BY total_spent DESC
LIMIT 5
```

```
[141]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}

for table_name in ['customer', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(),
                                   columns=[column[0] for column in conn.execute(query).description])

max_kazanc = df['customer'] \
    .merge(df['rental'], on='customer_id') \
    .merge(df['payment'], on='rental_id') \
    .groupby(['first_name', 'last_name']).agg(toplam_harcaması=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='toplam_harcaması', ascending=False) \
    .head(5)

print(max_kazanc)
```

	first_name	last_name	toplam_harcaması
318	KARL	SEAL	221.55
175	ELEANOR	HUNT	216.54
105	CLARA	SHAW	195.58
474	RHONDA	KENNEDY	194.61
389	MARION	SNYDER	194.61

```

25) Her filmin ortalama kiralama süresi;
SELECT film.title,
       AVG(julianday(rental.return_date) - julianday(rental.rental_date)) AS avg_rental_duration
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
WHERE rental.return_date IS NOT NULL
GROUP BY film.title

```

```

: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['film', 'inventory', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(),
                                   columns=[column[0] for column in conn.execute(query).description])
# ortalama kiralama süresi
merged_df = df['film'] \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id')
merged_df = merged_df[merged_df['return_date'].notnull()]
# kiralama süresinin günü
merged_df['kiralama_suresi'] = (
    pd.to_datetime(merged_df['return_date']) - pd.to_datetime(merged_df['rental_date']))
).dt.total_seconds() / 86400
# film adına göre ortalama kiralama süresi
ortalama_kiralama_suresi = merged_df.groupby('title').agg(
    avg_rental_duration=('kiralama_suresi', 'mean'))
).reset_index()
print(ortalama_kiralama_suresi.head())

```

	title	avg_rental_duration
0	ACADEMY DINOSAUR	4.997128
1	ACE GOLDFINGER	5.642708
2	ADAPTATION HOLES	3.465625
3	AFFAIR PREJUDICE	4.758333

```

26) Her türde ki en popüler film;
WITH RankedFilms AS (
    SELECT category.name AS genre,
        film.title,
        COUNT(rental.rental_id) AS rental_count,
        ROW_NUMBER() OVER (PARTITION BY category.name ORDER BY COUNT(rental.rental_id) DESC) AS row_num
    FROM category
    JOIN film_category ON category.category_id = film_category.category_id
    JOIN film ON film_category.film_id = film.film_id
    JOIN inventory ON film.film_id = inventory.film_id
    JOIN rental ON inventory.inventory_id = rental.inventory_id
    GROUP BY genre, film.title)
SELECT genre, title, rental_count
FROM RankedFilms
WHERE row_num = 1

```

```

import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['category', 'film_category', 'film', 'inventory', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(),
                                    columns=[column[0] for column in conn.execute(query).description])
# birleştirme
merged = df['category'] \
    .merge(df['film_category'], on='category_id') \
    .merge(df['film'], on='film_id') \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id')
# kiralanma sayıları
populer_film = merged.groupby(['name', 'title']).agg(
    rental_count=('rental_id', 'count'))
.popreset_index().rename(columns={'name': 'genre'})
# her türde en popüler filmler için
en_populer_filmler = populer_film.loc[populer_film.groupby('genre')['rental_count'].idxmax()].reset_index(drop=True)
print(en_populer_filmler)

      genre          title  rental_count
0   Action  RUGRATS SHAKESPEARE        30

```

```

27) Her türün en fazla gelir sağlayan filmi;
WITH RankedFilms AS (
    SELECT category.name AS genre,
        film.title,
        SUM(payment.amount) AS total_revenue,
        ROW_NUMBER() OVER (PARTITION BY category.name ORDER BY SUM(payment.amount) DESC) AS row_num
    FROM category
    JOIN film_category ON category.category_id = film_category.category_id
    JOIN film ON film_category.film_id = film.film_id
    JOIN inventory ON film.film_id = inventory.film_id
    JOIN rental ON inventory.inventory_id = rental.inventory_id
    JOIN payment ON rental.rental_id = payment.rental_id
    GROUP BY genre, film.title)
    SELECT genre, title, total_revenue
    FROM RankedFilms
    WHERE row_num = 1

```

```

import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['category', 'film_category', 'film', 'inventory', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(),
                                    columns=[column[0] for column in conn.execute(query).description])
merged = df['category'] \
    .merge(df['film_category'], on='category_id') \
    .merge(df['film'], on='film_id') \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id') \
    .merge(df['payment'], on='rental_id')
# film her türde ne kadar gelir getirdiği
revenue_by_film = merged.groupby(['name', 'title']).agg(
    total_revenue=('amount', 'sum'))
.reset_index().rename(columns={'name': 'genre'})
# her türün en yüksek gelirli filmleri
en_yuksek_gelirli_film = revenue_by_film.loc[
    revenue_by_film.groupby('genre')['total_revenue'].idxmax()]
.reset_index(drop=True)

```

```

    [revenue_by_film.groupby('genre')['total_revenue'].idxmax()]
].reset_index(drop=True)
print(en_yuksek_gelirli_film)

```

	genre	title	total_revenue
0	Action	FOOL MOCKINGBIRD	175.77
1	Animation	DOGMA FAMILY	178.70
2	Children	BACKLASH UNDEFEATED	158.81
3	Classics	STEEL SANTA	141.77
4	Comedy	ZORRO ARK	214.69
5	Documentary	WIFE TURN	223.69
6	Drama	TORQUE BOUND	198.72

```

28) DVD, en çok iade etmeyen müsteri; --> dvd en az iade eden müsteri
SELECT customer.first_name, customer.last_name, COUNT(rental.rental_id) AS unreturned_count
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
WHERE rental.return_date IS NULL
GROUP BY customer.first_name, customer.last_name
ORDER BY unreturned_count DESC
LIMIT 1

: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['customer', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(),
                                    columns=[column[0] for column in conn.execute(query).description])

iade_eden_musteriler = df['customer'] \
    .merge(df['rental'], on='customer_id') \
    .loc[df['rental']['return_date'].notnull()] # return_date null olmayanları seç
min_iade_eden_musteri = iade_eden_musteriler.groupby(['first_name', 'last_name']).agg(
    returned_count=('rental_id', 'count'))
.reset_index() \
    .sort_values(by='returned_count', ascending=True) \
    .head(1)

print(min_iade_eden_musteri)

```

	first_name	last_name	returned_count
71	BRIAN	WYMAN	12

29) En fazla kiralama yapan 5 çalışan;

```
SELECT staff.first_name, staff.last_name, COUNT(rental.rental_id) AS rental_count
FROM staff
JOIN rental ON staff.staff_id = rental.staff_id
GROUP BY staff.first_name, staff.last_name
ORDER BY rental_count DESC
LIMIT 5
```

```
import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(),
                                    columns=[column[0] for column in conn.execute(query).description])

# çalışanlar ve kiralamalar tablosunu bireleştir
max_kiralama = df['staff'] \
    .merge(df['rental'], on='staff_id') \
    .groupby(['first_name', 'last_name']).agg(
        total_rentals=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='total_rentals', ascending=False) \
    .head(5)

print(max_kiralama)

### en fazla 5 tane istendi ama 5 tane maksimum kiralama yapan olmadığı için 2 tane çıkmaktadır.
```

	first_name	last_name	total_rentals
1	Mike	Hillyer	8040
0	Jon	Stephens	8004

30) En fazla kiralama yapan 5 müşterinin hangi şubeden kiralama yaptığı;

```
'9]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

query = "PRAGMA table_info(store);" # Bu sorgu store tablosundaki sütunları getirir
columns = pd.read_sql_query(query, conn)

print(columns)
```

	cid	name	type	notnull	dflt_value	pk
0	0	store_id	INT	1	None	1
1	1	manager_staff_id	SMALLINT	1	None	0
2	2	address_id	INT	1	None	0
3	3	last_update	TIMESTAMP	1	None	0

```
[177]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe Listesi oluşturma
df = {}
for table_name in ['customer', 'rental', 'store', 'address']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(),
                                   columns=[column[0] for column in conn.execute(query).description])

max_kiralama_yapanlar = df['customer'] \
    .merge(df['rental'], on='customer_id') \
    .merge(df['store'], on='store_id') \
    .groupby(['first_name', 'last_name', 'store_id']).agg(
        total_rentals=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='total_rentals', ascending=False) \
    .head(5)

print(max_kiralama_yapanlar)
```

	first_name	last_name	store_id	total_rentals
175	ELEANOR	HUNT	1	46
318	KARL	SEAL	2	45
379	MARCIA	DEAN	1	42
105	CLARA	SHAW	1	42
536	TAMMY	SANDERS	2	41

31) Her türde en az kiralanan filmler;

```
[180]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe Listesi oluşturma
df = {}
for table_name in ['category', 'film_category', 'film', 'inventory', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

min_kiralama = df['category'] \
    .merge(df['film_category'], on='category_id') \
    .merge(df['film'], on='film_id') \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inventory')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .groupby(['name', 'title']).agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='kiralama_sayisi', ascending=True) \
    .head(5) \
    .rename(columns={'name': 'kategorisi', 'title': 'filmin_ismi', 'kiralama_sayisi': 'kiralama_sayisi'})

print("Her Türde En Az Kiralanınan Filmler:")
print(min_kiralama.to_string(index=False))
```

Her Türde En Az Kiralanınan Filmler:		
kategorisi	filmin_ismi	kiralama_sayisi
Documentary	HARDLY ROBBERS	4
Foreign	MIXED DOORS	4
Horror	TRAIN BUNCH	4
New	MANNEQUIN WORST	5
Games	FEVER EMPIRE	5

32) En çok kiralama yapan 5 müşterinin şehirleri;

```
SELECT
    city.city,
    customer.first_name,
    customer.last_name,
    COUNT(rental.rental_id) AS rental_count
FROM customer
JOIN address ON customer.address_id = address.address_id
JOIN city ON address.city_id = city.city_id
JOIN rental ON customer.customer_id = rental.customer_id
GROUP BY customer.customer_id, city.city
ORDER BY rental_count DESC
LIMIT 5
```

```
import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['customer', 'address', 'city', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
max_kiralama_yapan_musteri = df['customer'] \
    .merge(df['address'], on='address_id', suffixes=('', '_address')) \
    .merge(df['city'], on='city_id', suffixes=('', '_city')) \
    .merge(df['rental'], on='customer_id', suffixes=('', '_rental')) \
    .groupby(['city', 'first_name', 'last_name']).agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='kiralama_sayisi', ascending=False) \
    .head(5) \
    .rename(columns={'city': 'şehri', 'first_name': 'isimi', 'last_name': 'soyisimi'})
print("En Çok Kiralama Yapan 5 Müşteri:")
print(max_kiralama_yapan_musteri)
```

En Çok Kiralama Yapan 5 Müşteri:

	şehri	isimi	soyisimi	kiralama_sayisi
433	Saint-Denis	ELEANOR	HUNT	46
96	Cape Coral	KARL	SEAL	45
333	Molodetno	CLARA	SHAW	42

33) En çok kazanç sağlayan 5 müşterinin şehirleri;

```
SELECT
    city.city,
    customer.first_name,
    customer.last_name,
    SUM(payment.amount) AS total_spent
FROM customer
JOIN address ON customer.address_id = address.address_id
JOIN city ON address.city_id = city.city_id
JOIN rental ON customer.customer_id = rental.customer_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY customer.customer_id, city.city
ORDER BY total_spent DESC
LIMIT 5
```

```
import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['customer', 'address', 'city', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
max_kazanc = df['customer'] \
    .merge(df['address'], on='address_id', suffixes=('', '_address')) \
    .merge(df['city'], on='city_id', suffixes=('', '_city')) \
    .merge(df['rental'], on='customer_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby(['city', 'first_name', 'last_name']) \
    .agg(toplam_harcama=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='toplam_harcama', ascending=False) \
    .head(5)
print("Maksimum kazanç sağlayan 5 müşterinin şehirleri:")
print(max_kazanc[['city', 'first_name', 'last_name', 'toplam_harcama']])
```

Maksimum kazanç sağlayan 5 müşterinin şehirleri:

	city	first_name	last_name	toplam_harcama
96	Cape Coral	KARL	SEAL	221.55
433	Saint-Denis	ELEANOR	HUNT	216.54

```

34) En çok kiralanan filmlerin şehirleri;
SELECT
    city.city,
    film.title,
    COUNT(rental.rental_id) AS rental_count
FROM rental
JOIN customer ON rental.customer_id = customer.customer_id
JOIN address ON customer.address_id = address.address_id
JOIN city ON address.city_id = city.city_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
GROUP BY city.city, film.title
ORDER BY rental_count DESC
LIMIT 5

```

```

import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['city', 'address', 'customer', 'rental', 'inventory', 'film']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0] for column in conn.execute(query).description])
max_kiralananların_sehirleri = df['city'] \
    .merge(df['address'], on='city_id', suffixes=('', '_address')) \
    .merge(df['customer'], on='address_id', suffixes=('', '_customer')) \
    .merge(df['rental'], on='customer_id', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='rental_id', suffixes=('', '_inventory')) \
    .merge(df['film'], on='inventory_id', suffixes=('', '_film')) \
    .groupby(['city', 'title']).agg(rental_count=('rental_id', 'count')).reset_index() \
    .sort_values(by='rental_count', ascending=False) \
    .head(7)
print(max_kiralananların_sehirleri)

```

	city	title	rental_count
14182	Trshavn	FLATLINERS KILLER	3
12735	Sorocaba	CADDYSHACK JEDI	3
7308	Kurgan	DETECTIVE VISION	3

```

35) En az kiralanan 5 filmin şehirleri;
SELECT
    city.city,
    film.title,
    COUNT(rental.rental_id) AS rental_count
FROM rental
JOIN customer ON rental.customer_id = customer.customer_id
JOIN address ON customer.address_id = address.address_id
JOIN city ON address.city_id = city.city_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
GROUP BY city.city, film.title
ORDER BY rental_count ASC
LIMIT 5

```

```

import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['city', 'address', 'customer', 'rental', 'inventory', 'film']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0] for column in conn.execute(query).description])
min_kiralananların_sehirleri = df['city'] \
    .merge(df['address'], on='city_id', suffixes=('', '_address')) \
    .merge(df['customer'], on='address_id', suffixes=('', '_customer')) \
    .merge(df['rental'], on='customer_id', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='rental_id', suffixes=('', '_inventory')) \
    .merge(df['film'], on='inventory_id', suffixes=('', '_film')) \
    .groupby(['city', 'title']).agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='kiralama_sayisi', ascending=True) \
    .head(5) \
    .rename(columns={'city': 'sehri', 'title': 'filmin_adi', 'kiralama_sayisi': 'kiralama_sayisi'})
print("En Az Kiralanan 5 Filmin Şehirleri :")
print(min_kiralananların_sehirleri)

```

En Az Kiralanan 5 Filmin Şehirleri :

	sehri	filmin_adi	kiralama_sayisi
0	A Coruña (La Coruña)	BLADE POLISH	1
10513	Phnom Penh	Tourist PELICAN	1
10514	Phnom Penh	TWISTED PIRATES	1

36) En çok kazanç sağlayan 5 filmin şehirleri;

```
: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['city', 'address', 'customer', 'rental', 'inventory', 'film', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_kazancli_filmelerin_sehri = df['city'] \
    .merge(df['address'], on='city_id', suffixes=('', '_address')) \
    .merge(df['customer'], on='address_id', suffixes=('', '_customer')) \
    .merge(df['rental'], on='customer_id', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', suffixes=('', '_inventory')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby(['city', 'title']).agg(
        total_revenue=('amount', 'sum'))
    .reset_index() \
    .sort_values(by='total_revenue', ascending=False) \
    .head(5)

print("En Çok Kazanç Sağlayan 5 Filmin Şehirleri :")
print(max_kazancli_filmelerin_sehri)
```

En Çok Kazanç Sağlayan 5 Filmin Şehirleri :

	city	title	total_revenue
10752	Probolinggo	MINDS TRUMAN	18.98
10565	Plock	CALIFORNIA BIRDS	18.98
6650	Kamyin	ROSES TREASURE	17.98
11854	Santa Barbara dOeste	WIFE TURN	17.98
3520	Datong	EAGLES PANKY	16.98

37) En az kazançlı 5 filmin şehirleri;

```
: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['city', 'address', 'customer', 'rental', 'inventory', 'film', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

film_gelir_sehir = df['city'] \
    .merge(df['address'], on='city_id', suffixes=('', '_address')) \
    .merge(df['customer'], on='address_id', suffixes=('', '_customer')) \
    .merge(df['rental'], on='customer_id', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', suffixes=('', '_inventory')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby(['city', 'title']).agg(
        total_revenue=('amount', 'sum'))
    .reset_index() \
    .sort_values(by='total_revenue', ascending=False) \
    .head(5)

print("En az kazançlı 5 filmin şehirleri:")
print(film_gelir_sehir.to_string(index=False))
```

En az kazançlı 5 filmin şehirleri:

	city	title	total_revenue
	Probolinggo	MINDS TRUMAN	18.98
	Plock	CALIFORNIA BIRDS	18.98
	Kamyin	ROSES TREASURE	17.98
	Santa Barbara dOeste	WIFE TURN	17.98
	Datong	EAGLES PANKY	16.98

```

38) En fazla kiralama yapan müşterinin kiraladığı filmler;
SELECT
    customer.first_name,
    customer.last_name,
    film.title,
    COUNT(rental.rental_id) AS rental_count
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
WHERE customer.customer_id = (
    SELECT customer_id
    FROM rental
    GROUP BY customer_id
    ORDER BY COUNT(*) DESC
    LIMIT 1)
GROUP BY film.film_id
ORDER BY rental_count DESC

9]: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['customer', 'rental', 'inventory', 'film']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
max_kiralamaYapan_musteri = df['customer'] \
    .merge(df['rental'], on='customer_id', suffixes=('', '_rental')) \
    .groupby('customer_id').agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='kiralama_sayisi', ascending=False) \
    .head(3)
max_kiralamaYapan_musterinin_filmeleri = max_kiralamaYapan_musteri \
    .merge(df['customer'], on='customer_id', suffixes=('', '_customer')) \
    .merge(df['rental'], on='customer_id', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', suffixes=('', '_inventory')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .groupby(['first_name', 'last_name', 'title']).agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .rename(columns={'first_name': 'ad', 'last_name': 'soyad', 'title': 'film_adi', 'kiralama_sayisi': 'kiralama_sayisi'}) \
    .sort_values(by='film_adi', ascending=False) \
    .head(3)
print("En Fazla Kiralama Yapan Müşterinin Kiraladığı Filmler: ")
print(max_kiralamaYapan_musterinin_filmeleri)

En Fazla Kiralama Yapan Müşterinin Kiraladığı Filmler:
      ad soyad          film_adi kiralama_sayisi
41   CLARA SHAW      WOLVES DESIRE           1
87   ELEANOR HUNT    WIZARD COLDBLOODED        1
131  KARL SEAL       WEDDING APOLLO          2

40) Kazancı en çok sağlayan müşterinin kiraladığı filmler;
SELECT
    customer.first_name,
    customer.last_name,
    film.title,
    SUM(payment.amount) AS total_spent
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
JOIN payment ON rental.rental_id = payment.rental_id
WHERE customer.customer_id = (
    SELECT customer.customer_id
    FROM customer
    JOIN rental ON customer.customer_id = rental.customer_id
    JOIN payment ON rental.rental_id = payment.rental_id
    GROUP BY customer.customer_id
    ORDER BY SUM(payment.amount) DESC
    LIMIT 1
)
GROUP BY film.film_id
ORDER BY total_spent DESC

```

```
[82]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['customer', 'rental', 'inventory', 'film', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_kazancli_musteri_id = df['customer'] \
    .merge(df['rental'], on='customer_id', how='inner', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', how='inner', suffixes=('', '_payment')) \
    .groupby('customer_id').agg(total_income=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='total_income', ascending=False) \
    .head(1)['customer_id'].values[0]

musterinin_filmeleri = df['customer'] \
    .query("customer_id == @max_kazancli_musteri_id") \
    .merge(df['rental'], on='customer_id', how='inner', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', how='inner', suffixes=('', '_inventory')) \
    .merge(df['film'], on='film_id', how='inner', suffixes=('', '_film')) \
    .merge(df['payment'], on='rental_id', how='inner', suffixes=('', '_payment')) \
    .groupby('title').agg(total_spent=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='total_spent', ascending=False) \
    .head(7)

print("Kazancı En Çok Sağlayan Müşterinin Kiraladığı Filmler:")
print(musterinin_filmeleri)
```

Kazancı En Çok Sağlayan Müşterinin Kiraladığı Filmler:

	title	total_spent
12	FOOL MOCKINGBIRD	18.99
41	STING PERSONAL	9.99

```
41) Kazancı en az sağlayan müşterinin kiraladığı filmleri;
SELECT
    customer.first_name,
    customer.last_name,
    film.title,
    SUM(payment.amount) AS total_spent
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
JOIN payment ON rental.rental_id = payment.rental_id
WHERE customer.customer_id = (
    SELECT customer.customer_id
    FROM customer
    JOIN rental ON customer.customer_id = rental.customer_id
    JOIN payment ON rental.rental_id = payment.rental_id
    GROUP BY customer.customer_id
    ORDER BY SUM(payment.amount) ASC
    LIMIT 1)
GROUP BY film.title
ORDER BY total_spent ASC
```

```
[87]: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['customer', 'rental', 'inventory', 'film', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
```

```

for column in conn.execute(query).description]
min_kazancli_musteri_id = df['customer'] \
    .merge(df['rental'], on='customer_id', how='inner', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', how='inner', suffixes=('', '_payment')) \
    .groupby('customer_id').agg(total_income=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='total_income') \
    .head(1)[['customer_id']].values[0]
musterinin_filmleri = df['customer'] \
    .query("customer_id == @min_kazancli_musteri_id") \
    .merge(df['rental'], on='customer_id', how='inner', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', how='inner', suffixes=('', '_inventory')) \
    .merge(df['film'], on='film_id', how='inner', suffixes=('', '_film')) \
    .merge(df['payment'], on='rental_id', how='inner', suffixes=('', '_payment')) \
    .groupby('title').agg(total_spent=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='total_spent', ascending=True) \
    .head(7)
print("Kazancı En Az Sağlayan Müşterinin Kiraladığı Filmler :")
print(musterinin_filmleri)

Kazancı En Az Sağlayan Müşterinin Kiraladığı Filmler :
      title  total_spent
0     ARMAGEDDON LOST      0.99
1     CAMELOT VACATION      0.99
3     EMPIRE MALKOVICH      0.99
7     ILLUSION AMELIE      0.99

```

42) Kazancı en az sağlayan müşterinin en fazla kiraladığı film türleri;

```

SELECT
    customer.first_name,
    customer.last_name,
    category.name AS genre,
    COUNT(rental.rental_id) AS rental_count
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
JOIN film_category ON film.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
WHERE customer.customer_id = (
    SELECT customer.customer_id
    FROM customer
    JOIN rental ON customer.customer_id = rental.customer_id
    JOIN payment ON rental.rental_id = payment.rental_id
    GROUP BY customer.customer_id
    ORDER BY SUM(payment.amount) ASC
    LIMIT 1
)
GROUP BY category.name
ORDER BY rental_count DESC
LIMIT 1

```

```

04]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['customer', 'rental', 'inventory', 'film', 'film_category', 'category', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
        for column in conn.execute(query).description])

```

```

min_kazancli_musteri_id = df['customer'] \
    .merge(df['rental'], on='customer_id', how='inner', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', how='inner', suffixes=('', '_payment')) \
    .groupby('customer_id').agg(total_income=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='total_income') \
    .head(1)['customer_id'].values[0]

musterinin_film_turu = df['customer'] \
    .query("customer_id == @min_kazancli_musteri_id") \
    .merge(df['rental'], on='customer_id', how='inner', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', how='inner', suffixes=('', '_inventory')) \
    .merge(df['film'], on='film_id', how='inner', suffixes=('', '_film')) \
    .merge(df['film_category'], on='film_id', how='inner', suffixes=('', '_film_category')) \
    .merge(df['category'], on='category_id', how='inner', suffixes=('', '_category')) \
    .groupby('name').agg(rental_count=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='rental_count', ascending=False) \
    .head(1)

print("Kazancı en az sağlayan müşterinin en fazla kiraladığı film türleri: ")
print(musterinin_film_turu)

```

Kazancı en az sağlayan müşterinin en fazla kiraladığı film türleri:

name	rental_count
Sci-Fi	4

43) En çok kiralanan filmi hangi çalışan kiraladı;

```

SELECT
    staff.first_name,
    staff.last_name,
    film.title,
    COUNT(rental.rental_id) AS rental_count
FROM staff
JOIN rental ON staff.staff_id = rental.staff_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
WHERE film.film_id = (
    SELECT film.film_id
    FROM film
    JOIN inventory ON film.film_id = inventory.film_id
    JOIN rental ON inventory.inventory_id = rental.inventory_id
    GROUP BY film.film_id
    ORDER BY COUNT(rental.rental_id) DESC
    LIMIT 1
)
GROUP BY staff.first_name, staff.last_name, film.title

```

```

107]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['customer', 'rental', 'inventory', 'film', 'film_category', 'category', 'payment', 'staff']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
        for column in conn.execute(query).description])

```

```

max_kiralanan_film = df['film'] \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id') \
    .groupby('film_id').agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='kiralama_sayisi', ascending=False) \
    .head(1)

kiralayan_calisan = max_kiralanan_film \
    .merge(df['inventory'], on='film_id', suffixes=('_film', '_inventory')) \
    .merge(df['rental'], on='inventory_id', suffixes=('_inventory', '_rental')) \
    .merge(df['staff'], on='staff_id', suffixes=('_rental', '_staff')) \
    .merge(df['film'], on='film_id', suffixes=('_staff', '_film')) \
    .groupby(['first_name', 'last_name', 'title']).agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .rename(columns={'first_name': 'ad', 'last_name': 'soyad', 'title': 'film_adi', 'kiralama_sayisi': 'kiralama_sayisi'})

print("En Çok Kiralanan Filmi Kiralayan Çalışanlar:")
print(kiralayan_calisan)

```

En Çok Kiralanan Filmi Kiralayan Çalışanlar:

	ad	soyad	film_adi	kiralama_sayisi
0	Jon	Stephens	BUCKET BROTHERHOOD	18
1	Mike	Hillyer	BUCKET BROTHERHOOD	16

44) En az kiralanan filmi hangi çalışan kiraladı;

```

SELECT
    staff.first_name,
    staff.last_name,
    film.title,
    COUNT(rental.rental_id) AS rental_count
FROM staff
JOIN rental ON staff.staff_id = rental.staff_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
WHERE film.film_id = (
    SELECT film.film_id
    FROM film
    JOIN inventory ON film.film_id = inventory.film_id
    JOIN rental ON inventory.inventory_id = rental.inventory_id
    GROUP BY film.film_id
    ORDER BY COUNT(rental.rental_id) ASC
    LIMIT 1)
GROUP BY staff.first_name, staff.last_name, film.title

```

```

112]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                     for column in conn.execute(query).description])

min_kiralanan_film = df['film'] \

```

```

min_kiralanan_film = df['film'] \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inventory')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby('film_id').agg(total_gelir=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='total_gelir', ascending=True) \
    .head(1)

kiralayan_calisan = df['staff'] \
    .merge(df['rental'], on='staff_id', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', suffixes=('', '_inventory')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .loc[lambda x: x['film_id'] == en_cok_gelir_getiren_film['film_id'].values[0]] \
    .groupby(['first_name', 'last_name', 'title']).agg(total_revenue=('amount', 'sum')) \
    .reset_index()

print("En Az Kiralanan Film Ve Kiralayan Çalışanlar: ")
print(kiralayan_calisan)

```

```

En Az Kiralanan Film Ve Kiralayan Çalışanlar:
   first_name last_name          title  total_revenue
0      Jon Stephens TELEGRAPH VOYAGE      147.83
1     Mike Hillyer TELEGRAPH VOYAGE       83.90

```

45) En kazançlı filmi hangi çalışanın kiraladığı;

```

SELECT
    staff.first_name,
    staff.last_name,
    film.title,
    SUM(payment.amount) AS total_revenue
FROM staff
JOIN rental ON staff.staff_id = rental.staff_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
JOIN payment ON rental.rental_id = payment.rental_id
WHERE film.film_id = (
    SELECT film.film_id
    FROM film
    JOIN inventory ON film.film_id = inventory.film_id
    JOIN rental ON inventory.inventory_id = rental.inventory_id
    JOIN payment ON rental.rental_id = payment.rental_id
    GROUP BY film.film_id
    ORDER BY SUM(payment.amount) DESC
    LIMIT 1
)
GROUP BY staff.first_name, staff.last_name, film.title
ORDER BY total_revenue DESC

```

```

[19]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

```

```

en_fazla_kazancli_film = df['film'] \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby('film_id')['amount'].sum().idxmax()

kiralayan_calisan = df['staff'] \
    .merge(df['rental'], on='staff_id', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', suffixes=('', '_inv')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .loc[lambda x: x['film_id'] == en_fazla_kazancli_film] \
    .groupby(['first_name', 'last_name', 'title']) \
    .agg(toplam_gelir=('amount', 'sum')) \
    .reset_index() \
    .rename(columns={'first_name': 'ad', 'last_name': 'soyad', 'title': 'film_adi', 'toplam_gelir': 'toplam_gelir'})

print("En kazançlı filmi hangi çalışanın kiraladığı: ")
print(kiralayan_calisan)

```

```

En kazançlı filmi hangi çalışanın kiraladığı:
      ad      soyad      film_adi      toplam_gelir
0 Jon Stephens TELEGRAPH VOYAGE        147.83
1 Mike Hillyer TELEGRAPH VOYAGE        83.90

```

46) En az kazançlı filmi hangi çalışanın kiraladığı;

```

SELECT
    staff.first_name,
    staff.last_name,
    film.title,
    SUM(payment.amount) AS total_revenue
FROM staff
JOIN rental ON staff.staff_id = rental.staff_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
JOIN payment ON rental.rental_id = payment.rental_id
WHERE film.film_id = (
    SELECT film.film_id
    FROM film
    JOIN inventory ON film.film_id = inventory.film_id
    JOIN rental ON inventory.inventory_id = rental.inventory_id
    JOIN payment ON rental.rental_id = payment.rental_id
    GROUP BY film.film_id
    ORDER BY SUM(payment.amount) ASC
    LIMIT 1
)
GROUP BY staff.first_name, staff.last_name, film.title
ORDER BY total_revenue ASC

```

```

0]: import pandas as pd
import sqlite3

# sqlite ile veritabanınıza bağlanıyoruz.
conn = sqlite3.connect(file_path)

# DataFrame listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

```

```

    for column in conn.execute(query).description]

en_az_kazancli_film = df['film'] \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby('film_id')['amount'].sum().idxmin()

kiralayan_calisan = df['staff'] \
    .merge(df['rental'], on='staff_id', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', suffixes=('', '_inv')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .loc[lambda x: x['film_id'] == en_az_kazancli_film] \
    .groupby(['first_name', 'last_name', 'title']) \
    .agg(toplam_gelir=('amount', 'sum')) \
    .reset_index() \
    .rename(columns={'first_name': 'ad', 'last_name': 'soyad', 'title': 'film_adi', 'toplam_gelir': 'toplam_gelir'})

print("En az kazançlı filmi hangi çalışanın kiraladığı: ")
print(kiralayan_calisan)

En az kazançlı filmi hangi çalışanın kiraladığı:
      ad      soyad      film_adi      toplam_gelir
0  Jon Stephens  OKLAHOMA JUMANJI       1.98
1  Mike Hillyer  OKLAHOMA JUMANJI       3.96

```

47) En çok kiralanan filmin hangi mağzadan kiralandığı;

```

SELECT
    store.store_id,
    film.title,
    COUNT(rental.rental_id) AS rental_count
FROM store
JOIN staff ON store.store_id = staff.store_id
JOIN rental ON staff.staff_id = rental.staff_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
WHERE film.film_id = (
    SELECT film.film_id
    FROM film
    JOIN inventory ON film.film_id = inventory.film_id
    JOIN rental ON inventory.inventory_id = rental.inventory_id
    GROUP BY film.film_id
    ORDER BY COUNT(rental.rental_id) DESC
    LIMIT 1
)
GROUP BY store.store_id, film.title
ORDER BY rental_count DESC

```

```

]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_kiralanan_film = df['film'] \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .groupby(['film_id', 'store_id']).count().idxmax()

```

```

filmin_magaza_id = df['store'] \
    .merge(df['staff'], on='store_id', suffixes=('', '_staff')) \
    .merge(df['rental'], on='staff_id', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', suffixes=('', '_inv')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .loc[lambda x: x['film_id'] == max_kiralanan_film] \
    .groupby(['store_id', 'title']) \
    .agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .rename(columns={'store_id': 'magaza_id', 'title': 'film_adi', 'kiralama_sayisi': 'kiralama_sayisi'})

print("En çok kiralanan filmin hangi mağzadan kiralandığı: ")
print(filmin_magaza_id)

```

En çok kiralanan filmin hangi mağzadan kiralandığı:

magaza_id	film_adi	kiralama_sayısı
0	1 BUCKET BROTHERHOOD	16
1	2 BUCKET BROTHERHOOD	18

48) En az kiralanan filmin hangi mağazada kiralandığı;

```

SELECT
    store.store_id,
    film.title,
    COUNT(rental.rental_id) AS rental_count
FROM store
JOIN staff ON store.store_id = staff.store_id
JOIN rental ON staff.staff_id = rental.staff_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
WHERE film.film_id = (
    SELECT film.film_id
    FROM film
    JOIN inventory ON film.film_id = inventory.film_id
    JOIN rental ON inventory.inventory_id = rental.inventory_id
    GROUP BY film.film_id
    ORDER BY COUNT(rental.rental_id) ASC
    LIMIT 1
)

```

```

GROUP BY store.store_id, film.title
ORDER BY rental_count ASC

[123]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

min_kiralanan_film = df['film'] \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .groupby('film_id')['rental_id'].count().idxmin()

filmin_magaza_id = df['store'] \
    .merge(df['staff'], on='store_id', suffixes=('', '_staff')) \
    .merge(df['rental'], on='staff_id', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', suffixes=('', '_inv')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .loc[lambda x: x['film_id'] == min_kiralanan_film] \
    .groupby(['store_id', 'title']) \
    .agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .rename(columns={'store_id': 'magaza_id', 'title': 'film_adi', 'kiralama_sayisi': 'kiralama_sayisi'})

print("En az kiralanan filmin hangi mağazada kiralandığı: ")
print(filmin_magaza_id)

```

En az kiralanan filmin hangi mağazada kiralandığı:

magaza_id	film_adi	kiralama_sayisi
0	1 HARDLY ROBBERS	1
1	2 HARDLY ROBBERS	3

49) En kazançlı filmin hangi mağazadan kiralandığı;

```

SELECT
    store.store_id,
    film.title,
    SUM(payment.amount) AS total_revenue
FROM store
JOIN staff ON store.store_id = staff.store_id
JOIN rental ON staff.staff_id = rental.staff_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
JOIN payment ON rental.rental_id = payment.rental_id
WHERE film.film_id = (
    SELECT film.film_id
    FROM film
    JOIN inventory ON film.film_id = inventory.film_id
    JOIN rental ON inventory.inventory_id = rental.inventory_id
    JOIN payment ON rental.rental_id = payment.rental_id
    GROUP BY film.film_id
    ORDER BY SUM(payment.amount) DESC
    LIMIT 1
)
GROUP BY store.store_id, film.title
ORDER BY total_revenue DESC

```

```

4]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

```

```

max_kazancli_film = df['film'] \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby('film_id')['amount'].sum().idxmax()

filmin_magaza_id = df['store'] \
    .merge(df['staff'], on='store_id', suffixes=('', '_staff')) \
    .merge(df['rental'], on='staff_id', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', suffixes=('', '_inv')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .loc[lambda x: x['film_id'] == max_kazancli_film] \
    .groupby(['store_id', 'title']) \
    .agg(toplam_gelir=('amount', 'sum')) \
    .reset_index() \
    .rename(columns={'store_id': 'magaza_id', 'title': 'film_adi', 'toplam_gelir': 'toplum_gelir'})

print("En kazançlı filmin hangi mağazadan kiralandığı: ")
print(filmin_magaza_id)

```

En kazançlı filmin hangi mağazadan kiralandığı:

	magaza_id	film_adi	toplum_gelir
0	1	TELEGRAPH VOYAGE	83.90
1	2	TELEGRAPH VOYAGE	147.83

50) En az kazançlı filmin hangi mağazadan kiralandığı;

```

SELECT
    store.store_id,
    film.title,
    SUM(payment.amount) AS total_revenue
FROM store
JOIN staff ON store.store_id = staff.store_id
JOIN rental ON staff.staff_id = rental.staff_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
JOIN payment ON rental.rental_id = payment.rental_id
WHERE film.film_id = (
    SELECT film.film_id
    FROM film
    JOIN inventory ON film.film_id = inventory.film_id
    JOIN rental ON inventory.inventory_id = rental.inventory_id
    JOIN payment ON rental.rental_id = payment.rental_id
    GROUP BY film.film_id
    ORDER BY SUM(payment.amount) ASC
    LIMIT 1)
GROUP BY store.store_id, film.title
ORDER BY total_revenue ASC

```

```

[25]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

```

```

min_kazancli_film = df['film'] \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby('film_id')['amount'].sum().idxmin()

filmin_magaza_id = df['store'] \
    .merge(df['staff'], on='store_id', suffixes=('', '_staff')) \
    .merge(df['rental'], on='staff_id', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', suffixes=('', '_inv')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .loc[lambda x: x['film_id'] == min_kazancli_film] \
    .groupby(['store_id', 'title']) \
    .agg(toplam_gelir=('amount', 'sum')) \
    .reset_index() \
    .rename(columns={'store_id': 'magaza_id', 'title': 'film_adi', 'toplam_gelir': 'toplasm_gelir'})

print("En az kazançlı filmin hangi mağazadan kiralandığı: ")
print(filmin_magaza_id)

```

En az kazançlı filmin hangi mağazadan kiralandığı:

	magaza_id	film_adi	toplasm_gelir
0	1	OKLAHOMA JUMANJI	3.96
1	2	OKLAHOMA JUMANJI	1.98

51) Müşterilerin kiraladıkları filmlerin toplam kiralama süresi;

```

SELECT
    customer.first_name,
    customer.last_name,
    SUM(julianday(rental.return_date) - julianday(rental.rental_date)) AS total_rental_duration
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
WHERE rental.return_date IS NOT NULL
GROUP BY customer.first_name, customer.last_name

```

```
(111): import pandas as pd
import sqlite3

#sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

#dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'customer', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0] for column in conn.execute(query).description])

#dönüşüm işlemi yapılıyor
df['rental'][['rental_date']] = pd.to_datetime(df['rental'][['rental_date']])
df['rental'][['return_date']] = pd.to_datetime(df['rental'][['return_date']])

toplasm_kiralama_suresi = df['customer'] \
    .merge(df['rental'], on='customer_id') \
    .groupby(['first_name', 'last_name']) \
    .agg(total_rental_duration=('rental_id', lambda x: (df['rental'][['return_date']][x.index] - df['rental'][['rental_date']][x.index]).sum().days)) \
    .reset_index()

```

Toplam Kiralama Süresi:

	first_name	last_name	total_rental_duration
0	AARON	SELBY	121
1	ADAM	GOOCH	182
2	ADRIAN	CLARY	95
3	AGNES	BISHOP	96
4	ALAN	KAHN	128
...
594	WILLIE	MARRAHAN	136

```

52) En çok kiralanan film türleri;

SELECT
    category.name AS genre,
    film.title,
    COUNT(rental.rental_id) AS rental_count
FROM film
JOIN film_category ON film.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY category.name, film.title
ORDER BY rental_count DESC

[136]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_kiralanan_tur = df['film'] \
    .merge(df['film_category'], on='film_id', suffixes=('', '_fc')) \
    .merge(df['category'], on='category_id', suffixes=('', '_c')) \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_r')) \
    .groupby(['name', 'title']) \
    .agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='kiralama_sayisi', ascending=False) \
    .head(3)

print("En çok kiralanan film türleri: ")
print(max_kiralanan_tur)

```

```

En çok kiralanan film türleri:
      name          title  kiralama_sayisi
0   Thriller  Thriller  1000
1   Thriller  Thriller  999
2   Thriller  Thriller  998

```

```

53) En az kiralanan film türleri;

SELECT
    category.name AS genre,
    film.title,
    COUNT(rental.rental_id) AS rental_count
FROM film
JOIN film_category ON film.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY genre, film.title
ORDER BY rental_count ASC

```

```

[135]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

min_kiralanan_tur = df['film'] \
    .merge(df['film_category'], on='film_id', suffixes=('', '_fc')) \
    .merge(df['category'], on='category_id', suffixes=('', '_c')) \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_r')) \
    .groupby(['name', 'title']) \
    .agg(rental_count=('rental_id', 'count')) \
    .reset_index() \
    .rename(columns={'name': 'genre'}) \
    .sort_values(by='rental_count', ascending=True) \
    .head(3)

```

```

print("En az kiralanan film türleri: ")
print(min_kiralanan_tur)

En az kiralanan film türleri:
   genre          title  rental_count
315 Documentary  HARDLY ROBBERS        4
525 Foreign      MIXED DOORS         4
656 Horror       TRAIN BUNCH        4

54) Her müşterinin toplam ödeme miktarı;

SELECT
    customer.first_name,
    customer.last_name,
    SUM(payment.amount) AS total_payment
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY customer.first_name, customer.last_name

```

```

138]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

```

```

toplam_odeme = df['customer'] \
    .merged(df['rental'], on='customer_id') \
    .merge(df['payment'], on='rental_id') \
    .groupby(['first_name', 'last_name']) \
    .agg(toplam_odeme=('amount', 'sum')) \
    .reset_index()

print("Her müşterinin toplam ödeme miktarı: ")
print(toplam_odeme)

```

Her müşterinin toplam ödeme miktarı:

	first_name	last_name	toplam_odeme
0	AARON	SELBY	110.76
1	ADAM	GOCCH	101.78
2	ADRIAN	CLARY	74.81
3	AGNES	BISHOP	98.77
4	ALAN	KAHN	124.74
..
594	WILLIE	MARKHAM	101.75
595	WILMA	RICHARDS	91.80
596	YOLANDA	WEAVER	110.73
597	YVONNE	WATKINS	92.79
598	ZACHARY	HITE	146.69

[599 rows x 3 columns]

55) En uzun süre kiralanan filmler;

```

SELECT
    film.title,
    MAX(julianday(rental.return_date) - julianday(rental.rental_date)) AS max_rental_days
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY film.title
ORDER BY max_rental_days DESC

```

```

146]: import pandas as pd
      import sqlite3

      # sqlite ile veritabanına bağlanıyoruz.
      conn = sqlite3.connect(file_path)

      # dataframe listesi oluşturma
      df = {}
      for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
          query = f"SELECT * FROM {table_name}"
          df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
              for column in conn.execute(query).description])

      # dönüştürme işlemi yapılıyor
      df['rental'][['rental_date']] = pd.to_datetime(df['rental'][['rental_date']])
      df['rental'][['return_date']] = pd.to_datetime(df['rental'][['return_date']])

      # kiralama süresinin hesabı
      df['rental'][['rental_days']] = (df['rental'][['return_date']] - df['rental'][['rental_date']]).dt.days

      max_kiralanan_film = df[['film'] \
          .merge(df[['inventory']], on='film_id') \
          .merge(df[['rental']], on='inventory_id') \
          .groupby('title') \
          .agg(max_rental_days=('rental_days', 'max')) \
          .reset_index() \
          .sort_values(by='max_rental_days', ascending=False)

      print("En uzun süre kiralanan filmler:")
      print(max_kiralanan_film)

```

```

En uzun süre kiralanan filmler:
   title  max_rental_days
0    ACADEMY DINOSAUR      9.0
655     PITY BOUND      9.0
551     MILE MULAN      9.0
552    MILLION ACE      9.0

```

56) En kısa süre kiralanan filmler;

```

SELECT
    film.title,
    MIN(julianday(rental.return_date) - julianday(rental.rental_date)) AS min_rental_days
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY film.title
ORDER BY min_rental_days ASC

```

```

import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
        for column in conn.execute(query).description])

# dönüştürme işlemi yapılıyor
df['rental'][['rental_date']] = pd.to_datetime(df['rental'][['rental_date']])
df['rental'][['return_date']] = pd.to_datetime(df['rental'][['return_date']])
# kiralama süresinin hesabı
df['rental'][['rental_days']] = (df['rental'][['return_date']] - df['rental'][['rental_date']]).dt.days
# En kısa süre kiralanan film
min_kiralanan_film = df[['film']] \
    .merge(df[['inventory']], on='film_id') \
    .merge(df[['rental']], on='inventory_id') \
    .groupby('title') \
    .agg(min_rental_days=('rental_days', 'min')) \
    .reset_index() \
    .sort_values(by='min_rental_days', ascending=True)
print("En kısa süre kiralanan filmler:")
print(min_kiralanan_film)

```

```

En kısa süre kiralanan filmler:
   title  min_rental_days
0    ACADEMY DINOSAUR      0.0
544    MERMAID INSECTS      0.0
546    METROPOLIS COMA      0.0

```

57) Müşterilerin kiraladığı filmler için ortalama ödeme miktarı;

```
SELECT
    customer.first_name,
    customer.last_name,
    AVG(payment.amount) AS avg_payment
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY customer.first_name, customer.last_name
```

58]: import pandas as pd
import sqlite3

```
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

ort_odeme = df['customer'] \
    .merge(df['rental'], on='customer_id') \
    .merge(df['payment'], on='rental_id') \
    .groupby(['first_name', 'last_name'], as_index=False) \
    .agg(avg_payment=('amount', 'mean'))

ort_odeme['avg_payment'] = ort_odeme['avg_payment'].round(5)

print("Müşterilerin kiraladığı filmler için ortalama ödeme miktarı: ")
print(ort_odeme)
```

Müşterilerin kiraladığı filmler için ortalama ödeme miktarı:

first_name	last_name	avg_payment
AARON	SELBY	4.61500
ADAM	GODCH	4.62636

58) En çok kiralanan filmlerin ortalama kiralama süresi;

```
SELECT
    film.title,
    AVG(julianday(rental.return_date) - julianday(rental.rental_date)) AS avg_rental_days
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY film.title
ORDER BY COUNT(rental.rental_id) DESC
LIMIT 10
```

57]: import pandas as pd

```
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

# kiralama tarihleri için dönüştürme
df['rental']['rental_date'] = pd.to_datetime(df['rental']['rental_date'])
df['rental']['return_date'] = pd.to_datetime(df['rental']['return_date'])
# kiralama süresinin hesabı
df['rental']['rental_days'] = (df['rental']['return_date'] - df['rental']['rental_date']).dt.days
max_kiralanan_film = df['film'] \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_r')) \
    .assign(rental_days=lambda x: (x['return_date'] - x['rental_date']).dt.days) \
    .groupby('title') \
    .agg(avg_rental_days=('rental_days', 'mean'), rental_count=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='rental_count', ascending=False) \
    .head(17)
print("En çok kiralanan filmlerin ortalama kiralama süresi:")
print(max_kiralanan_film[['title', 'avg_rental_days', 'rental_count']])
```

En çok kiralanan filmlerin ortalama kiralama süresi:

title	avg_rental_days	rental_count
BUCKET BROTHERHOOD	4.411765	34
ROCKETEER MOTHER	4.039394	33
RIDGEHONT SUBMARINE	5.580645	32

```

59) En az kiralanan filmlerin ortalama kiralama süresi;
SELECT
    film.title,
    AVG(julianday(rental.return_date) - julianday(rental.rental_date)) AS avg_rental_days
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY film.title
ORDER BY COUNT(rental.rental_id) ASC
LIMIT 10

```

```

import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
# kiralama tarihleri için dönüştürme
df['rental']['rental_date'] = pd.to_datetime(df['rental']['rental_date'])
df['rental']['return_date'] = pd.to_datetime(df['rental']['return_date'])
# kiralama süresinin hesabı
df['rental']['rental_days'] = (df['rental']['return_date'] - df['rental']['rental_date']).dt.days
min_kiralanan_film = df['film'] \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inv')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_r')) \
    .assign(rental_days=lambda x: (x['return_date'] - x['rental_date']).dt.days) \
    .groupby('title') \
    .agg(avg_rental_days=('rental_days', 'mean'), rental_count=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='rental_count', ascending=True) \
    .head(17)
print("En az kiralanan filmlerin ortalama kiralama süresi:")
print(min_kiralanan_film[['title', 'avg_rental_days', 'rental_count']])

```

En az kiralanan filmlerin ortalama kiralama süresi:

	title	avg_rental_days	rental_count
558	MIXED DOORS	5.75	4
866	TRAIN BUNCH	3.00	4
378	HARDLY ROBBERS	6.50	4
585	MUSSOLINI SPOILERS	3.60	5
245	FREEBORN SIEGEATRIS	2.40	5

```

60) En çok kazanç sağlayan müşterilerin ortalama ödeme miktarı;
SELECT
    customer.first_name,
    customer.last_name,
    AVG(payment.amount) AS avg_payment
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY customer.customer_id
ORDER BY SUM(payment.amount) DESC
LIMIT 12

```

```

: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
ort_odeme = df[['customer']] \
    .merge(df['rental'], on='customer_id', suffixes=('', '_r')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_p')) \
    .groupby(['first_name', 'last_name']) \
    .agg(avg_payment=('amount', 'mean'), total_payment=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='total_payment', ascending=False) \
    .head(12)
print("En çok kazanç sağlayan müşterilerin ortalama ödeme miktarı: ")
print(ort_odeme[['first_name', 'last_name', 'avg_payment']])

```

En çok kazanç sağlayan müşterilerin ortalama ödeme miktarı:

	first_name	last_name	avg_payment
318	KARL	SEAL	4.923333
175	ELEANOR	HUNT	4.707391
105	CLARA	SHAW	4.656667

61) En az kazanç sağlayan müşterilerin ortalama ödeme miktarı;

```

SELECT
    customer.first_name,
    customer.last_name,
    AVG(payment.amount) AS avg_payment
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY customer.customer_id
ORDER BY SUM(payment.amount) ASC
LIMIT 12

```

```

: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
ort_odeme = df[['customer']] \
    .merge(df['rental'], on='customer_id', suffixes=('', '_r')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_p')) \
    .groupby(['first_name', 'last_name']) \
    .agg(avg_payment=('amount', 'mean'), total_payment=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='total_payment', ascending=True) \
    .head(12)
print("En az kazanç sağlayan müşterilerin ortalama ödeme miktarı: ")
print(ort_odeme[['first_name', 'last_name', 'avg_payment']])

```

En az kazanç sağlayan müşterilerin ortalama ödeme miktarı:

	first_name	last_name	avg_payment
83	CAROLINE	BOWMAN	3.390000
351	LEONA	OBRIEN	3.632857
71	BRIAN	WYMAN	4.406667
296	JOHNNY	TURPIN	3.042632

```

62) Mağazalardaki toplam kiralama süresi;

SELECT
    store.store_id,
    SUM(julianday(rental.return_date) - julianday(rental.rental_date)) AS total_rental_days
FROM store
JOIN staff ON store.store_id = staff.store_id
JOIN rental ON staff.staff_id = rental.staff_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
GROUP BY store.store_id

```

```

: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
# dönüştürme işlemi正在进行中
df['rental']['rental_date'] = pd.to_datetime(df['rental']['rental_date'])
df['rental']['return_date'] = pd.to_datetime(df['rental']['return_date'])

# kiralama süresinin hesabı
df['rental']['rental_duration'] = (df['rental']['return_date'] - df['rental']['rental_date']).dt.days
toplam_sure = (df['store']
               .merge(df['staff'], on='store_id', how='inner')
               .merge(df['rental'], on='staff_id', how='inner')
               .groupby('store_id')
               .agg(total_rental_days=('rental_duration', 'sum'))
               .reset_index())
print("Mağazalardaki toplam kiralama süresi: ")
print(toplam_sure)

```

```

Mağazalardaki toplam kiralama süresi:
   store_id  total_rental_days
0         1            35897.0
1         2            35889.0

```

```

SELECT
    store.store_id,
    film.title,
    MAX(julianday(rental.return_date) - julianday(rental.rental_date)) AS max_rental_days
FROM store
JOIN staff ON store.store_id = staff.store_id
JOIN rental ON staff.staff_id = rental.staff_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
GROUP BY store.store_id, film.title
ORDER BY max_rental_days DESC
LIMIT 1

```

```

import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0] for column in conn.execute(query).description])
# dönüştürme işlemi yapılıyor
df['rental']['rental_date'] = pd.to_datetime(df['rental']['rental_date'])
df['rental']['return_date'] = pd.to_datetime(df['rental']['return_date'])
# kiralama süresinin hesabı
df['rental']['rental_duration'] = (df['rental']['return_date'] - df['rental']['rental_date']).dt.days
max_kiralanan_film = (df['store']
    .merge(df['staff'], on='store_id', how='inner', suffixes=('', '_staff'))
    .merge(df['rental'], on='staff_id', how='inner', suffixes=('', '_rental'))
    .merge(df['inventory'], on='inventory_id', how='inner', suffixes=('', '_inventory'))
    .merge(df['film'], on='film_id', how='inner', suffixes=('', '_film'))
    .groupby(['store_id', 'title'])
    .agg(max_rental_days=('rental_duration', 'max'))
    .reset_index()
    .sort_values(by='max_rental_days', ascending=False)
    .head(1))
print("En uzun süre kiralanan filmin hangi mağazada kiralandığı:")
print(max_kiralanan_film)

```

En uzun süre kiralanan filmin hangi mağazada kiralandığı:

store_id	title	max_rental_days
0	ACADEMY DINOSAUR	9.0

64) En kısa süre kiralanan film hangi mağazada kiralandığı;

```

100]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0] for column in conn.execute(query).description])

# dönüştürme işlemi yapılıyor
df['rental']['rental_date'] = pd.to_datetime(df['rental']['rental_date'])
df['rental']['return_date'] = pd.to_datetime(df['rental']['return_date'])

# kiralama süresinin hesabı
df['rental']['rental_duration'] = (df['rental']['return_date'] - df['rental']['rental_date']).dt.days

min_kiralanan_film = (df['store']
    .merge(df['staff'], on='store_id', how='inner', suffixes=('', '_staff'))
    .merge(df['rental'], on='staff_id', how='inner', suffixes=('', '_rental'))
    .merge(df['inventory'], on='inventory_id', how='inner', suffixes=('', '_inventory'))
    .merge(df['film'], on='film_id', how='inner', suffixes=('', '_film'))
    .groupby(['store_id', 'title'])
    .agg(min_rental_duration=('rental_duration', 'min'))
    .reset_index()
    .sort_values(by='min_rental_duration', ascending=True)
    .head(1))

print("En Kısa Süre Kiralanan Filmin Hangi Mağazada Kiralandığı:")
print(min_kiralanan_film[['store_id', 'title', 'min_rental_duration']])

```

En Kısa Süre Kiralanan Filmin Hangi Mağazada Kiralandığı:

store_id	title	min_rental_duration
0	ACADEMY DINOSAUR	0.0

```

65) Her filmin ortalama kiralama süresi;
SELECT film.title, AVG(julianday(rental.return_date) - julianday(rental.rental_date)) AS average_rental_days
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY film.title

184]: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
# dönüştürme işlemi正在进行中
df['rental']['rental_date'] = pd.to_datetime(df['rental']['rental_date'])
df['rental']['return_date'] = pd.to_datetime(df['rental']['return_date'])
# kiralama süresinin hesabı
df['rental']['rental_duration'] = (df['rental']['return_date'] - df['rental']['rental_date']).dt.days
ort_kiralama_suresi = df['film'] \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id') \
    .assign(kiralama_suresi=lambda x: (pd.to_datetime(x['return_date']) - pd.to_datetime(x['rental_date'])).dt.days) \
    .groupby('title') \
    .agg(ortalama_kiralama_gunleri=('kiralama_suresi', 'mean')) \
    .reset_index()
print("Her filmin ortalama kiralama süresi: ")
print(ort_kiralama_suresi)

Her filmin ortalama kiralama süresi:
   title  ortalama_kiralama_gunleri
0  ACADEMY DINOSAUR          4.545455
1    ACE GOLDFINGER          5.333333

```

```

66) En çok kiralanan film kategorileri;

SELECT category.name AS genre, film.title, COUNT(rental.rental_id) AS rental_count
FROM category
JOIN film_category ON category.category_id = film_category.category_id
JOIN film ON film_category.film_id = film.film_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY genre, film.title
ORDER BY rental_count DESC

05]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

populer_film_kategorisi = (df['category'] \
    .merge(df['film_category'], on='category_id', how='inner', suffixes=('', '_film_category')) \
    .merge(df['film'], on='film_id', how='inner', suffixes=('', '_film')) \
    .merge(df['inventory'], on='film_id', how='inner', suffixes=('', '_inventory')) \
    .merge(df['rental'], on='inventory_id', how='inner', suffixes=('', '_rental')) \
    .groupby(['name', 'title']) \
    .agg(rental_count=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='rental_count', ascending=False))

print("En çok kiralanan film kategorileri: ")
print(populer_film_kategorisi)

En çok kiralanan film kategorileri:
   name          title  rental_count
989    Travel    BUCKET BROTHERHOOD      34
536    Foreign   ROCKTEER MOTHER       33

```

67) En az kiralanan film kategorisi;

```
SELECT category.name AS genre, film.title, COUNT(rental.rental_id) AS rental_count
FROM category
JOIN film_category ON category.category_id = film_category.category_id
JOIN film ON film_category.film_id = film.film_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY genre, film.title
ORDER BY rental_count ASC
```

```
import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

min_film_kategorisi = (df['category'] \
    .merge(df['film_category'], on='category_id', how='inner', suffixes=('', '_film_category')) \
    .merge(df['film'], on='film_id', how='inner', suffixes=('', '_film')) \
    .merge(df['inventory'], on='film_id', how='inner', suffixes=('', '_inventory')) \
    .merge(df['rental'], on='inventory_id', how='inner', suffixes=('', '_rental')) \
    .groupby(['name', 'title']) \
    .agg(rental_count=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='rental_count', ascending=True))

print("En az kiralanan film kategorileri: ")
print(min_film_kategorisi)
```

En az kiralanan film kategorileri:

	name	title	rental_count
315	Documentary	HARDLY ROBBERS	4

68) En çok film kiralanan mağzalar;

```
: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_kiralanan_magazalar = df['category'] \
    .merge(df['film_category'], on='category_id', suffixes=('', '_film_category')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inventory')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .groupby('name') \
    .agg(toplam_kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='toplam_kiralama_sayisi', ascending=False)

print("En çok film kiralanan mağzalara ait türler ve kiralama sayıları: ")
print(max_kiralanan_magazalar)
```

En çok film kiralanan mağzalara ait türler ve kiralama sayıları:

	name	toplam_kiralama_sayisi
14	Sports	1179
1	Animation	1166
0	Action	1112
13	Sci-Fi	1101
7	Family	1096

```

69) En az film kiralanan mağazalar;

SELECT film.title, COUNT(rental.rental_id) AS toplam_kiralama_sayisi
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY film.title
ORDER BY toplam_kiralama_sayisi DESC

]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

min_kiralanan_magazalar = df['film'] \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id') \
    .groupby('title') \
    .agg(toplam_kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='toplam_kiralama_sayisi', ascending=True)

print("En az film kiralanan mağzaların filmleri ve kiralanma sayıları: ")
print(min_kiralanan_magazalar)

```

En az film kiralanan mağzaların filmleri ve kiralanma sayıları:

	title	toplam_kiralama_sayisi
558	MIXED DOORS	4
866	TRAIN BUNCH	4
378	HARDLY ROBBERS	4
585	MUSSOLINI SPOILERS	5
315	FREEDOM CLEOPATRA	5
...

70) En çok filimde rol almış aktörler;

```

SELECT actor.first_name, actor.last_name, COUNT(film_actor.film_id) AS film_count
FROM actor
JOIN film_actor ON actor.actor_id = film_actor.actor_id
GROUP BY actor.first_name, actor.last_name
ORDER BY film_count DESC

```

```

]: import pandas as pd
import sqlite3

# sqlite ile veritabanınıza bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category', 'actor', 'film_actor']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_rol_aktor = (df['actor'] \
    .merge(df['film_actor'], on='actor_id', how='inner') \
    .groupby(['first_name', 'last_name']) \
    .agg(film_count=('film_id', 'count')) \
    .reset_index() \
    .sort_values(by='film_count', ascending=False))

print("En çok filimde rol almış aktörler")
print(max_rol_aktor)

```

En çok filimde rol almış aktörler

	first_name	last_name	file_count
188	SUSAN	DAVIS	54
65	SINA	DEGENERES	42
190	WALTER	TORN	41
123	MARY	KEITEL	40
125	MATTHEW	CARREY	39
...
177	SISSY	SOBIESKI	38

```

71) En az filimde rol almış aktörler;

SELECT actor.first_name, actor.last_name, COUNT(film_actor.film_id) AS film_count
FROM actor
JOIN film_actor ON actor.actor_id = film_actor.actor_id
GROUP BY actor.first_name, actor.last_name
ORDER BY film_count ASC

[37]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma.
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category', 'actor', 'film_actor']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0] for column in conn.execute(query).description])

min_rol_aktor = (df['actor']
                  .merge(df['film_actor'], on='actor_id', how='inner')
                  .groupby(['first_name', 'last_name'])
                  .agg(film_count=('film_id', 'count'))
                  .reset_index()
                  .sort_values(by='film_count', ascending=True))

print("En çok filimde rol almış aktörler")
print(min_rol_aktor )

En çok filimde rol almış aktörler
   first_name last_name  film_count
51      EMILY       DEE          14
99      JUDY       DEAN          15
101     JULIA     FAWCETT         15
103     JULIA  ZELLWEGER          16
177     SISSY     SOBIESKI          18

```

```

72) Aktörlerin oynadığı en çok kiralanmış filmler;

[201]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category',
                   'actor', 'film_actor']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0] for column in conn.execute(query).description])

aktorler_ve_kiralama = (df['actor']
                         .merge(df['film_actor'], on='actor_id', how='inner')
                         .merge(df['film'], on='film_id', how='inner')
                         .merge(df['inventory'], on='film_id', how='inner')
                         .merge(df['rental'], on='inventory_id', how='inner')
                         .groupby(['actor_id', 'first_name', 'last_name', 'title'])
                         .agg(toplam_kiralama_sayisi=('rental_id', 'count'))
                         .reset_index()
                         .sort_values(by='toplam_kiralama_sayisi', ascending=False)
                         .head(10))

print("Aktörlerin Oynadığı En Çok Kiralanan Filmler Ve Aktörleri:")
print(aktorler_ve_kiralama[['first_name', 'last_name', 'title', 'toplam_kiralama_sayisi']])

```

```

C:\Users\hadim\anaconda3\lib\site-packages\pandas\core\frame.py:9190: FutureWarning: Passing 'suffixes' which cause dupli-
cor in a future version.

return merge(
Aktörlerin Oynadığı En Çok Kiralanan Filmler Ve Aktörleri:
   first_name last_name           title  toplam_kiralama_sayisi
1283      GARY   PHOENIX  BUCKET BROTHERHOOD            34
784       TIM   HACKMAN  BUCKET BROTHERHOOD            34
2274  CHARLIZE    DENCH  BUCKET BROTHERHOOD            34
2352   KIRSTEN   AKROYD  BUCKET BROTHERHOOD            34
5042     BURT   TEMPLE  BUCKET BROTHERHOOD            34
629       RIP  CRAWFORD  BUCKET BROTHERHOOD            34

```

```

73) Aktörlerin oynadığı en az kiralanan filmler:
[207]: import pandas as pd
import sqlite3

#sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

#dataframe listesi oluşturma
df = []
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category', 'actor', 'film_actor']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

aktorler_ve_kiralama = (df[['actor']]
                        .merge(df[['film_actor']], on='actor_id', how='inner')
                        .merge(df[['film']], on='film_id', how='inner')
                        .merge(df[['inventory']], on='film_id', how='inner')
                        .merge(df[['rental']], on='inventory_id', how='inner')
                        .groupby(['actor_id', 'first_name', 'last_name', 'title'])
                        .agg(toplam_kiralama_sayisi={'rental_id', 'count'})
                        .reset_index()
                        .sort_values(by='toplam_kiralama_sayisi', ascending=True)
                        .head(10))

# Sonucu yazdırma
print("Aktörlerin Oynadığı En Az Kiralanan Filmler Ve Aktörleri:")
print(aktorler_ve_kiralama[['first_name', 'last_name', 'title', 'toplam_kiralama_sayisi']))

C:\Users\hadim\anaconda3\lib\site-packages\pandas\core\frame.py:9190: FutureWarning: Passing 'suffixes' which cause duplicate columns ('last_update' in a future version.
return merge(
Aktörlerin Oynadığı En Az Kiralanan Filmler Ve Aktörleri:
   first_name  last_name      title  toplam_kiralama_sayisi
4127     GRETA       MALDEN  HARDLY ROBBERS          4
3474     ADAM        HOPPER    TRAIN BUNCH          4
2798  PENELOPE      CRONYN    TRAIN BUNCH          4
1796  MICHELLE  MCCONAUGHEY    MIXED DOORS          4

```

```

74) En fazla kategoride oynamış aktörler:
SELECT category.name AS genre, COUNT(DISTINCT film_actor.actor_id) AS actor_count
FROM category
JOIN film_category ON category.category_id = film_category.category_id
JOIN film ON film_category.film_id = film.film_id
JOIN film_actor ON film.film_id = film_actor.film_id
GROUP BY genre
ORDER BY actor_count DESC

```

```

[208]: import pandas as pd
import sqlite3

#sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
#dataframe listesi oluşturma
df = []
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category', 'actor', 'film_actor']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

kategori_aktor = df['category'] \
    .merge(df[['film_category']], on='category_id', suffixes=('', '_film_category')) \
    .merge(df[['film']], on='film_id', suffixes=('', '_film')) \
    .merge(df[['film_actor']], on='film_id', suffixes=('', '_film_actor')) \
    .groupby('name') \
    .agg(aktor_sayisi={'actor_id', 'nunique'}) \
    .reset_index() \
    .sort_values(by='aktor_sayisi', ascending=False)

print("En fazla kategoride oynamış aktörler:")
print(kategori_aktor)

```

```

En fazla kategoride oynamış aktörler:
   name  aktor_sayisi
14    Sports        182
  8  Foreign        179

```

```

75) En az kategoride oynayan aktörler;

SELECT category.name AS genre, COUNT(DISTINCT film_actor.actor_id) AS actor_count
FROM category
JOIN film_category ON category.category_id = film_category.category_id
JOIN film ON film_category.film_id = film.film_id
JOIN film_actor ON film.film_id = film_actor.film_id
GROUP BY genre
ORDER BY actor_count ASC

[109]: import pandas as pd
import sqlite3

#sqlite ile veritabanınıza bağlanıyoruz.
conn = sqlite3.connect(file_path)

#dataframe listesi oluşturma
df = {}
for table_name in ['staff', 'rental', 'inventory', 'film', 'payment', 'store', 'customer', 'category', 'film_category', 'actor', 'film_actor']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0] for column in conn.execute(query).description])

kategori_aktor = df['category'] \
    .merge(df['film_category'], on='category_id', suffixes=('', '_film_category')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['film_actor'], on='film_id', suffixes=('', '_film_actor')) \
    .groupby('name') \
    .agg(aktor_sayisi=('actor_id', 'nunique')) \
    .reset_index() \
    .sort_values(by='aktor_sayisi', ascending=True)

print("En az kategoride oynayan aktörleri:")
print(kategori_aktor)

En az kategoride oynayan aktörler:
      name  aktor_sayisi
11     Music        144
4      Comedy        147
9      Games        150
10     Horror        156
3    Classics        162
6      Drama        162

```

```

76) Filimde en çok kiralanan oyuncu kategorileri;

SELECT category.name AS genre, COUNT(rental.rental_id) AS rental_count
FROM category
JOIN film_category ON category.category_id = film_category.category_id
JOIN film ON film_category.film_id = film.film_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY category.name
ORDER BY rental_count DESC

[110]: import pandas as pd
import sqlite3

#sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

#dataframe listesi oluşturma
df = {}
for table_name in ['category', 'film_category', 'film', 'inventory', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0] for column in conn.execute(query).description])

pop_oyun_kategorisi = df['category'] \
    .merge(df['film_category'], on='category_id', suffixes=('', '_film_category')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inventory')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .groupby('name') \
    .agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='kiralama_sayisi', ascending=False)

print("Filmde en çok kiralanan oyuncu kategorileri:")
print(pop_oyun_kategorisi)

Filmde en çok kiralanan oyuncu kategorileri:
      name  kiralama_sayisi
14    Sports        1179
1   Animation       1166

```

```

77) Film de en az kiralanan oyun kategorileri;

SELECT category.name AS genre, COUNT(rental.rental_id) AS rental_count
FROM category
JOIN film_category ON category.category_id = film_category.category_id
JOIN film ON film_category.film_id = film.film_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY genre
ORDER BY rental_count ASC

[12]: import pandas as pd
import sqlite3

# sqlite ile veritabanina baglaniyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi olusturma
df = {}
for table_name in ['category', 'film_category', 'film', 'inventory', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

oyun_kategorisi = df['category'] \
    .merge(df['film_category'], on='category_id', suffixes=('', '_film_category')) \
    .merge(df['film'], on='film_id', suffixes=('', '_film')) \
    .merge(df['inventory'], on='film_id', suffixes=('', '_inventory')) \
    .merge(df['rental'], on='inventory_id', suffixes=('', '_rental')) \
    .groupby('name') \
    .agg(kiralama_sayisi=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='kiralama_sayisi', ascending=True)

print("Film de en az kiralanan oyun kategorileri: ")
print(oyun_kategorisi)

Film de en az kiralanan oyun kategorileri:
      name  kiralama_sayisi
11     Music           830
15    Travel           837
18   Horror           846
..       ...

```

```

78) En fazla kazanç sağlayan kategoriler;
SELECT category.name AS genre, SUM(payment.amount) AS total_revenue
FROM category
JOIN film_category ON category.category_id = film_category.category_id
JOIN film ON film_category.film_id = film.film_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY genre
ORDER BY total_revenue DESC

[13]: import pandas as pd
import sqlite3
# sqlite ile veritabanina baglaniyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi olusturma
df = {}
for table_name in ['category', 'film_category', 'film', 'inventory', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_kazanc = (df['category'] \
    .merge(df['film_category'], on='category_id', how='inner', suffixes=('', '_film_category')) \
    .merge(df['film'], on='film_id', how='inner', suffixes=('', '_film')) \
    .merge(df['inventory'], on='film_id', how='inner', suffixes=('', '_inventory')) \
    .merge(df['rental'], on='inventory_id', how='inner', suffixes=('', '_rental')) \
    .groupby('name') \
    .agg(total_revenue=('rental_rate', lambda x: x.sum() * len(x))) \
    .reset_index() \
    .sort_values(by='total_revenue', ascending=False))
print("En fazla kazanç sağlayan kategoriler: ")
print(max_kazanc)

En fazla kazanç sağlayan kategoriler:
      name  total_revenue
14     Sports        4264690.59
1   Animation        3752584.44
..       ...

```

```

79) En az kazanç sağlayan kategoriler;
SELECT category.name AS genre, SUM(payment.amount) AS total_revenue
FROM category
JOIN film_category ON category.category_id = film_category.category_id
JOIN film ON film_category.film_id = film.film_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY genre
ORDER BY total_revenue ASC;

```

```

: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['category', 'film_category', 'film', 'inventory', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
min_kazanc = (df['category']
               .merge(df['film_category'], on='category_id', how='inner', suffixes=('', '_film_category'))
               .merge(df['film'], on='film_id', how='inner', suffixes=('', '_film'))
               .merge(df['inventory'], on='film_id', how='inner', suffixes=('', '_inventory'))
               .merge(df['rental'], on='inventory_id', how='inner', suffixes=('', '_rental'))
               .groupby('name')
               .agg(total_revenue=('rental_rate', lambda x: x.sum() * len(x)))
               .reset_index()
               .sort_values(by='total_revenue', ascending=True))
print("En az kazanç sağlayan kategoriler: ")
print(min_kazanc)

```

En az kazanç sağlayan kategoriler:

	name	total_revenue
11	Music	2109611.00
10	Horror	2219514.84

```

80) En çok film kiralayan müşterilerin ortalama ödeme miktarları;
SELECT customer.first_name, customer.last_name, AVG(payment.amount) AS avg_payment
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY customer.first_name, customer.last_name
ORDER BY COUNT(rental.rental_id) DESC
LIMIT 10

```

```

: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['customer', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
max_odeme = (df['customer']
               .merge(df['rental'], on='customer_id', how='inner')
               .merge(df['payment'], on='rental_id', how='inner')
               .groupby(['first_name', 'last_name'])
               .agg(
                   avg_payment=('amount', 'mean'),
                   rental_count=('rental_id', 'count')
               )
               .reset_index()
               .sort_values(by='rental_count', ascending=False)
               .head(10))
print("En çok film kiralayan müşterilerin ortalama ödeme miktarları: ")
print(max_odeme)

```

En çok film kiralayan müşterilerin ortalama ödeme miktarları:

	first_name	last_name	avg_payment	rental_count
175	ELEANOR	HUNT	4.707391	46
318	KARL	SEAL	4.923333	45

```
81) En az film kiralayan müşterilerin ortalama ödeme miktarı;

SELECT customer.first_name, customer.last_name, AVG(payment.amount) AS avg_payment
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY customer.first_name, customer.last_name
ORDER BY COUNT(rental.rental_id) ASC
LIMIT 10
```

```
[24]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['customer', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

min_odeme = (df['customer']
              .merge(df['rental'], on='customer_id', how='inner')
              .merge(df['payment'], on='rental_id', how='inner')
              .groupby(['first_name', 'last_name'])
              .agg(
                  avg_payment=('amount', 'mean'),
                  rental_count=('rental_id', 'count'))
              .reset_index()
              .sort_values(by='rental_count', ascending=True)
              .head(17))

print("En az film kiralayan müşterilerin ortalama ödeme miktarı: ")
print(min_odeme)
```

```
En az film kiralayan müşterilerin ortalama ödeme miktarı:
   first_name last_name  avg_payment  rental_count
71      BRIAN     WYMAN      4.406667         12
550    TIFFANY     JORDAN      4.275714         14
```

```
82) Mağazalarda en çok kiralanın kategorileri;

SELECT store.store_id, category.name AS genre, COUNT(rental.rental_id) AS rental_count
FROM store
JOIN staff ON store.store_id = staff.store_id
JOIN rental ON staff.staff_id = rental.staff_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id |
JOIN film_category ON film.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
GROUP BY store.store_id, genre
ORDER BY rental_count DESC
```

```
[25]: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyor.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['store', 'staff', 'rental', 'inventory', 'film', 'film_category', 'category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

magaza_kategori = (df['store']
                    .merge(df['staff'], on='store_id', how='inner', suffixes=('', '_staff'))
                    .merge(df['rental'], on='staff_id', how='inner', suffixes=('', '_rental'))
                    .merge(df['inventory'], on='inventory_id', how='inner', suffixes=('', '_inventory'))
                    .merge(df['film'], on='film_id', how='inner', suffixes=('', '_film'))
                    .merge(df['film_category'], on='film_id', how='inner', suffixes=('', '_film_category'))
                    .merge(df['category'], on='category_id', how='inner', suffixes=('', '_category'))
                    .groupby(['store_id', 'name'])
                    .agg(rental_count=('rental_id', 'count'))
                    .reset_index()
                    .sort_values(by='rental_count', ascending=False))
print("Mağazalarda en çok kiralanın kategorileri: ")
print(magaza_kategori)
```

```
Mağazalarda en çok kiralanın kategorileri:
   store_id      name  rental_count
30          2    Sports        614
17          2  Animation       584
1           1  Animation       582
```

```

83) Mağazalarda en az kiralanan kategoriler;
SELECT store.store_id, category.name AS genre, COUNT(rental.rental_id) AS rental_count
FROM store
JOIN staff ON store.store_id = staff.store_id
JOIN rental ON staff.staff_id = rental.staff_id
JOIN inventory ON rental.inventory_id = inventory.inventory_id
JOIN film ON inventory.film_id = film.film_id
JOIN film_category ON film.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
GROUP BY store.store_id, genre
ORDER BY rental_count ASC

```

```

: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['store', 'staff', 'rental', 'inventory', 'film', 'film_category', 'category']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
magaza_kategori = (df['store'] \
    .merge(df['staff'], on='store_id', how='inner', suffixes=('', '_staff')) \
    .merge(df['rental'], on='staff_id', how='inner', suffixes=('', '_rental')) \
    .merge(df['inventory'], on='inventory_id', how='inner', suffixes=('', '_inventory')) \
    .merge(df['film'], on='film_id', how='inner', suffixes=('', '_film')) \
    .merge(df['film_category'], on='film_id', how='inner', suffixes=('', '_film_category')) \
    .merge(df['category'], on='category_id', how='inner', suffixes=('', '_category')) \
    .groupby(['store_id', 'name']) \
    .agg(rental_count=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='rental_count', ascending=True))
print("Mağazalarda en az kiralanan kategoriler: ")
print(magaza_kategori)

Mağazalarda en az kiralanan kategoriler:
  store_id      name  rental_count
27          2     Music         398
15          1   Travel         399

```

```

84) En fazla sayıda film kiralayan mağaza ve toplam kiralama sayısını;
SELECT store.store_id, COUNT(rental.rental_id) AS total_rentals
FROM store
JOIN staff ON store.store_id = staff.store_id
JOIN rental ON staff.staff_id = rental.staff_id
GROUP BY store.store_id |
ORDER BY total_rentals DESC
LIMIT 1

```

```

: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['store', 'staff', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

magaza_top_kirasi = df['store'] \
    .merge(df['staff'], on='store_id') \
    .merge(df['rental'], on='staff_id') \
    .groupby('store_id') \
    .agg(total_rentals=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='total_rentals', ascending=False) \
    .head(1)
print("En fazla sayıda film kiralayan mağaza ve toplam kiralama sayısı: ")
print(magaza_top_kirasi)

En fazla sayıda film kiralayan mağaza ve toplam kiralama sayısı:
  store_id  total_rentals
0          1           8040

```

```
85) En az sayıda film kiralayan mağaza ve toplam kiralama sayısı;  
  
SELECT store.store_id, COUNT(rental.rental_id) AS total_rentals  
FROM store  
JOIN staff ON store.store_id = staff.store_id  
JOIN rental ON staff.staff_id = rental.staff_id  
GROUP BY store.store_id  
ORDER BY total_rentals ASC  
LIMIT 1
```

```
[30]:  
import pandas as pd  
import sqlite3  
  
# sqlite ile veritabanına bağlanıyoruz.  
conn = sqlite3.connect(file_path)  
  
# dataframe listesi oluşturma  
df = {}  
for table_name in ['store', 'staff', 'rental']:  
    query = f"SELECT * FROM {table_name}"  
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]  
                                                                           for column in conn.execute(query).description])  
  
magaza_top_kirasi = df['store'] \  
    .merge(df['staff'], on='store_id') \  
    .merge(df['rental'], on='staff_id') \  
    .groupby('store_id') \  
    .agg(total_rentals=('rental_id', 'count')) \  
    .reset_index() \  
    .sort_values(by='total_rentals', ascending=True) \  
    .head(1)  
  
print("En fazla sayıda film kiralayan mağaza ve toplam kiralama sayısı: ")  
print(magaza_top_kirasi)  
  
En fazla sayıda film kiralayan mağaza ve toplam kiralama sayısı:  
store_id  total_rentals  
1           2          8004
```

```
86) En fazla sayıda kiralama yapan müsteri ve kiralama sayısı;  
  
SELECT customer.first_name, customer.last_name, COUNT(rental.rental_id) AS total_rentals  
FROM customer  
JOIN rental ON customer.customer_id = rental.customer_id  
GROUP BY customer.first_name, customer.last_name  
ORDER BY total_rentals DESC  
LIMIT 1
```

```
[31]:  
import pandas as pd  
import sqlite3  
  
# sqlite ile veritabanına bağlanıyoruz.  
conn = sqlite3.connect(file_path)  
  
# dataframe listesi oluşturma  
df = {}  
for table_name in ['store', 'staff', 'rental', 'customer']:  
    query = f"SELECT * FROM {table_name}"  
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]  
                                                                           for column in conn.execute(query).description])  
  
max_kira = df['customer'] \  
    .merge(df['rental'], on='customer_id') \  
    .groupby(['first_name', 'last_name']) \  
    .agg(total_rentals=('rental_id', 'count')) \  
    .reset_index() \  
    .sort_values(by='total_rentals', ascending=False) \  
    .head(1)  
  
print("En fazla sayıda kiralama yapan müsteri ve kiralama sayısı: ")  
print(max_kira)
```

```
En fazla sayıda kiralama yapan müsteri ve kiralama sayısı:  
first_name last_name  total_rentals  
175   ELEANOR      HUNT        46
```

```
87) En az sayıda kiralama yapan müsteri ve kiralama sayısı;  
  
SELECT customer.first_name, customer.last_name, COUNT(rental.rental_id) AS total_rentals  
FROM customer  
JOIN rental ON customer.customer_id = rental.customer_id  
GROUP BY customer.first_name, customer.last_name  
ORDER BY total_rentals ASC  
LIMIT 1
```

```
[34]: import pandas as pd  
import sqlite3  
  
# sqlite ile veritabanına bağlanıyoruz.  
conn = sqlite3.connect(file_path)  
  
# dataframe listesi oluşturma  
df = {}  
for table_name in ['store', 'staff', 'rental', 'customer']:  
    query = f"SELECT * FROM {table_name}"  
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]  
                                                                           for column in conn.execute(query).description])  
  
min_kira = df['customer'] \  
    .merge(df['rental'], on='customer_id') \  
    .groupby(['first_name', 'last_name']) \  
    .agg(total_rentals=('rental_id', 'count')) \  
    .reset_index() \  
    .sort_values(by='total_rentals', ascending=True) \  
    .head(1)  
  
print("En fazla sayıda kiralama yapan müsteri ve kiralama sayısı: ")  
print(min_kira)
```

```
En fazla sayıda kiralama yapan müsteri ve kiralama sayısı:  
first_name last_name total_rentals  
71 BRIAN WYMAN 12
```

88) En fazla kiralanan film ve toplam kirası;

```
SELECT film.title, COUNT(rental.rental_id) AS total_rentals  
FROM film  
JOIN inventory ON film.film_id = inventory.film_id  
JOIN rental ON inventory.inventory_id = rental.inventory_id  
GROUP BY film.title  
ORDER BY total_rentals DESC  
LIMIT 1
```

```
[40]: import pandas as pd  
import sqlite3  
  
# sqlite ile veritabanına bağlanıyoruz.  
conn = sqlite3.connect(file_path)  
  
# dataframe listesi oluşturma  
df = {}  
for table_name in ['rental', 'film', 'inventory']:  
    query = f"SELECT * FROM {table_name}"  
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]  
                                                                           for column in conn.execute(query).description])  
  
max_kirali_film = df['film'] \  
    .merge(df['inventory'], on='film_id') \  
    .merge(df['rental'], on='inventory_id') \  
    .groupby('title') \  
    .agg(total_rentals=('rental_id', 'count')) \  
    .reset_index() \  
    .sort_values(by='total_rentals', ascending=False) \  
    .head(1)  
  
print("En fazla kiralanan film ve toplam kirası: ")  
print(max_kirali_film)
```

```
En fazla kiralanan film ve toplam kirası:  
title total_rentals  
96 BUCKET BROTHERHOOD 34
```

```

89) En az kiralanan film ve toplam kirasi;

SELECT film.title, COUNT(rental.rental_id) AS total_rentals
FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
GROUP BY film.title
ORDER BY total_rentals ASC
LIMIT 1

[41]: import pandas as pd
import sqlite3

# sqlite ile veritabanina baglaniyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi olusturma
df = {}
for table_name in ['rental', 'film', 'inventory']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

min_kirali_film = df['film'] \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id') \
    .groupby('title') \
    .agg(total_rentals=('rental_id', 'count')) \
    .reset_index() \
    .sort_values(by='total_rentals', ascending=True) \
    .head(1)

print("En fazla kiralanan film ve toplam kirasi:")
print(min_kirali_film)

```

En fazla kiralanan film ve toplam kirasi:

title	total_rentals
MIXED DOORS	4

```

90) En fazla kazanc saglayan musteri ve toplam kazancı;

SELECT customer.first_name, customer.last_name, SUM(payment.amount) AS total_revenue
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY customer.first_name, customer.last_name
ORDER BY total_revenue DESC
LIMIT 1

[43]: import pandas as pd
import sqlite3

# sqlite ile veritabanina baglaniyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi olusturma
df = {}
for table_name in ['rental', 'customer', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_kazanc = df['customer'] \
    .merge(df['rental'], on='customer_id') \
    .merge(df['payment'], on='rental_id') \
    .groupby(['first_name', 'last_name']) \
    .agg(total_revenue=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='total_revenue', ascending=False) \
    .head(1)

print("En fazla kazanc saglayan musteri ve toplam kazancı:")
print(max_kazanc)

```

En fazla kazanc saglayan musteri ve toplam kazancı:

first_name	last_name	total_revenue
KARL	SEAL	221.55

```
91) En az kazanç sağlayan müsteri ve toplam kazancı;

SELECT customer.first_name, customer.last_name, SUM(payment.amount) AS total_revenue
FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY customer.first_name, customer.last_name
ORDER BY total_revenue ASC
LIMIT 1
```

```
[44]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['rental', 'customer', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

min_kazanc = df['customer'] \
    .merge(df['rental'], on='customer_id') \
    .merge(df['payment'], on='rental_id') \
    .groupby(['first_name', 'last_name']) \
    .agg(total_revenue=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='total_revenue', ascending=True) \
    .head(1)

print("En az kazanç sağlayan müsteri ve toplam kazancı: ")
print(min_kazanc)
```

En az kazanç sağlayan müsteri ve toplam kazancı:

first_name	last_name	total_revenue
B3	CAROLINE	50.85

```
92) En çok kazanç sağlayan film ve kategorisi;
SELECT film.title, category.name AS genre, SUM(payment.amount) AS total_revenue
FROM film
JOIN film_category ON film.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY film.title, category.name
ORDER BY total_revenue DESC
LIMIT 5
```

```
[1]: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['film', 'film_category', 'category', 'inventory', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_kazancli_film_kategorisi = df['film'] \
    .merge(df['film_category'], on='film_id', suffixes=('', '_fc')) \
    .merge(df['category'], on='category_id', suffixes=('', '_cat')) \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id') \
    .merge(df['payment'], on='rental_id', suffixes=('', '_pay')) \
    .groupby(['title', 'name']) \
    .agg(total_revenue=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='total_revenue', ascending=False) \
    .head(5)

print("En çok kazanç sağlayan film ve kategorisi: ")
print(max_kazancli_film_kategorisi)
```

En çok kazanç sağlayan film ve kategorisi:

	title	name	total_revenue
841	TELEGRAPH VOYAGE	Music	231.73
930	WIFE TURN	Documentary	223.69
957	ZORRO ARK	Comedy	214.69

```

93) En az kazanç sağlayan film ve kategorisi;

SELECT film.title, category.name AS genre, SUM(payment.amount) AS total_revenue
FROM film
JOIN film_category ON film.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY film.title, genre
ORDER BY total_revenue ASC
LIMIT 5

:
import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['film', 'film_category', 'category', 'inventory', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])
min_kazancli_film_kategorisi = df['film'] \
    .merge(df['film_category'], on='film_id', suffixes=('', '_fc')) \
    .merge(df['category'], on='category_id', suffixes=('', '_cat')) \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id') \
    .merge(df['payment'], on='rental_id', suffixes=('', '_pay')) \
    .groupby(['title', 'name']) \
    .agg(total_revenue=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='total_revenue', ascending=True) \
    .head(5)
print("En az kazanç sağlayan film ve kategorisi: ")
print(min_kazancli_film_kategorisi)

En az kazanç sağlayan film ve kategorisi:
      title      name  total_revenue
847   TEXAS WATCH     Horror       5.94
608 OKLAHOMA JUMANJI      New       5.94
315  FREEDOM CLEOPATRA   Comedy       5.95
244 DUFFEL APOCALYPSE Documentary     6.93

```

```
94) En fazla kazanç sağlayan mağaza ve toplam kazancı;
```

```
SELECT store.store_id, SUM(payment.amount) AS total_revenue
FROM store
JOIN staff ON store.store_id = staff.store_id
JOIN rental ON staff.staff_id = rental.staff_id
JOIN payment ON rental.rental_id = payment.rental_id
GROUP BY store.store_id
ORDER BY total_revenue DESC
LIMIT 1
```

```
[54]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['store', 'staff', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

max_kazancli_magaza = df['store'] \
    .merge(df['staff'], on='store_id', suffixes=('', '_staff')) \
    .merge(df['rental'], on='staff_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby('store_id') \
    .agg(toplam_kazanc=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='toplam_kazanc', ascending=False) \
    .head(1)

print("En fazla kazanç sağlayan mağaza ve toplam kazancı:")
print(max_kazancli_magaza)
```

```
En fazla kazanç sağlayan mağaza ve toplam kazancı:
```

```
store_id      toplam_kazanc
1              33881.94
```

```
95) En az kazanç sağlayan mağaza ve toplam kazancı;
```

```
SELECT
    store.store_id,
    SUM(payment.amount) AS total_revenue
FROM
    store
JOIN
    staff ON store.store_id = staff.store_id
JOIN
    rental ON staff.staff_id = rental.staff_id
JOIN
    payment ON rental.rental_id = payment.rental_id
GROUP BY
    store.store_id
ORDER BY
    total_revenue ASC
LIMIT 1
```

```
i]: import pandas as pd
import sqlite3
# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)
# dataframe listesi oluşturma
df = {}
for table_name in ['store', 'staff', 'rental', 'payment']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

min_kazancli_magaza = df['store'] \
    .merge(df['staff'], on='store_id', suffixes=('', '_staff')) \
    .merge(df['rental'], on='staff_id', suffixes=('', '_rental')) \
    .merge(df['payment'], on='rental_id', suffixes=('', '_payment')) \
    .groupby('store_id') \
    .agg(toplam_kazanc=('amount', 'sum')) \
    .reset_index() \
    .sort_values(by='toplam_kazanc', ascending=True) \
    .head(1)
```

```

print("En az kazanç sağlayan mağaza ve toplam kazancı:")
print(min_kazancli_magaza)

En az kazanç sağlayan mağaza ve toplam kazancı:
  store_id  toplam_kazanc
0          1      33524.62

96) En fazla sayıda farklı film kiralayan müsteri;

SELECT
    customer.first_name,
    customer.last_name,
    COUNT(DISTINCT inventory.film_id) AS unique_films_rented
FROM
    customer
JOIN
    rental ON customer.customer_id = rental.customer_id
JOIN
    inventory ON rental.inventory_id = inventory.inventory_id
GROUP BY
    customer.first_name, customer.last_name
ORDER BY
    unique_films_rented DESC
LIMIT 1

57]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['rental', 'customer']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

farkli_film = df['customer'] \\\n
    .merge(df['rental'], on='customer_id') \\
    .groupby(['first_name', 'last_name']) \\
    .agg(farkli_kiralama_sayisi=('inventory_id', 'nunique')) \\
    .reset_index() \\
    .sort_values(by='farkli_kiralama_sayisi', ascending=False) \\
    .head(1)

print("En fazla sayıda farklı film kiralayan müsteri:")
print(farkli_film)

En fazla sayıda farklı film kiralayan müsteri:
  first_name last_name  farkli_kiralama_sayisi
175   ELEANOR     HUNT            46

97) En fazla sayıda farklı film kiralayan müsteri;

SELECT
    customer.first_name,
    customer.last_name,
    COUNT(DISTINCT inventory.film_id) AS unique_films_rented
FROM
    customer
JOIN
    rental ON customer.customer_id = rental.customer_id
JOIN
    inventory ON rental.inventory_id = inventory.inventory_id
GROUP BY
    customer.first_name, customer.last_name
ORDER BY
    unique_films_rented ASC
LIMIT 1

```

```
[50]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['rental', 'customer']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

farkli_film = df['customer'] \
    .merge(df['rental'], on='customer_id') \
    .groupby(['first_name', 'last_name']) \
    .agg(farkli_kiralama_sayisi=('inventory_id', 'nunique')) \
    .reset_index() \
    .sort_values(by='farkli_kiralama_sayisi', ascending=True) \
    .head(1)

print("En az sayıda farklı film kiralayan müsteri:")
print(farkli_film)

En az sayıda farklı film kiralayan müsteri:
  first_name last_name  farkli_kiralama_sayisi
71      BRIAN      WYMAN                 12
```

98) En fazla sayıda farklı müsteri tarafından kiralanan film;

```
SELECT
    film.title,
    COUNT(DISTINCT rental.customer_id) AS unique_customers
FROM
    film
JOIN
    inventory ON film.film_id = inventory.film_id
JOIN
    rental ON inventory.inventory_id = rental.inventory_id
GROUP BY
    film.title
ORDER BY
    unique_customers DESC
LIMIT 5
```

```
[51]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['rental', 'film' , 'inventory']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

farkli_musteri_film = df['film'] \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id') \
    .groupby('title') \
    .agg(farkli_musteri_sayisi=('customer_id', 'nunique')) \
    .reset_index() \
    .sort_values(by='farkli_musteri_sayisi', ascending=False) \
    .head(5)

print("En fazla sayıda farklı müsteri tarafından kiralanan film:")
print(farkli_musteri_film)
```

```
print("En fazla sayıda farklı müşteri tarafından kiralanan film:")
print(farklı_musteri_film)
```

```
En fazla sayıda farklı müşteri tarafından kiralanan film:
    title  farklı_musteri_sayısı
96     BUCKET BROTHERHOOD           33
312    FORWARD TEMPLE              32
361    GRIT CLOCKWORK              32
697    RIDGEMONT SUBMARINE          32
733    SCALAWAG DUCK                32
```

```
99) En fazla sayıda farklı müşteri tarafından kiralanan film;
```

```
SELECT
    film.title,
    COUNT(DISTINCT rental.customer_id) AS unique_customers
FROM
    film
JOIN
    inventory ON film.film_id = inventory.film_id
JOIN
    rental ON inventory.inventory_id = rental.inventory_id
GROUP BY
    film.title
ORDER BY
    unique_customers ASC
| TMTT 5
```

```
[63]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

# dataframe listesi oluşturma
df = {}
for table_name in ['rental', 'film' , 'inventory']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                               for column in conn.execute(query).description])

farklı_musteri_film = df['film'] \
    .merge(df['inventory'], on='film_id') \
    .merge(df['rental'], on='inventory_id') \
    .groupby('title') \
    .agg(farklı_musteri_sayıısı=('customer_id', 'nunique')) \
    .reset_index() \
    .sort_values(by='farklı_musteri_sayıısı', ascending=True) \
    .head(5)

print("En az sayıda farklı müşteri tarafından kiralanan film: ")
print(farklı_musteri_film)
```

```
En az sayıda farklı müşteri tarafından kiralanan film:
    title  farklı_musteri_sayıısı
866    TRAIN BUNCH                 4
417    HUNTER ALTER                 4
558    MIXED DOORS                  4
378    HAROLDY ROBBERS              4
747    SEVEN SWARM                  5
```

```

100) En fazla kazanç sağlanan mağaza ve kategorisi;

SELECT
    store.store_id,
    category.name AS genre,
    SUM(payment.amount) AS total_revenue
FROM
    store
JOIN
    staff ON store.store_id = staff.store_id
JOIN
    rental ON staff.staff_id = rental.staff_id
JOIN
    inventory ON rental.inventory_id = inventory.inventory_id
JOIN
    film ON inventory.film_id = film.film_id
JOIN
    film_category ON film.film_id = film_category.film_id
JOIN
    category ON film_category.category_id = category.category_id
JOIN
    payment ON rental.rental_id = payment.rental_id
GROUP BY
    store.store_id, category.name
ORDER BY
    total_revenue DESC
LIMIT 3

```

```

7]: import pandas as pd
import sqlite3

# sqlite ile veritabanına bağlanıyoruz.
conn = sqlite3.connect(file_path)

```

```

# dataframe listesi oluşturma
df = {}
for table_name in ['inventory', 'film', 'film_category', 'category', 'payment', 'store', 'staff', 'rental']:
    query = f"SELECT * FROM {table_name}"
    df[table_name] = pd.DataFrame(conn.execute(query).fetchall(), columns=[column[0]
                                                                           for column in conn.execute(query).description])

magaza_kategori = (df['store']
                    .merge(df['staff'], on='store_id', how='inner', suffixes=('', '_staff'))
                    .merge(df['rental'], on='staff_id', how='inner', suffixes=('', '_rental'))
                    .merge(df['inventory'], on='inventory_id', how='inner', suffixes=('', '_inventory'))
                    .merge(df['film'], on='film_id', how='inner', suffixes=('', '_film'))
                    .merge(df['film_category'], on='film_id', how='inner', suffixes=('', '_film_category'))
                    .merge(df['category'], on='category_id', how='inner', suffixes=('', '_category'))
                    .merge(df['payment'], on='rental_id', how='inner', suffixes=('', '_payment'))
                    .groupby(['store_id', 'name']) # 'name' burada kategori adıdır
                    .agg(total_revenue=('amount', 'sum')) # Toplam geliri hesaplama
                    .reset_index()
                    .sort_values(by='total_revenue', ascending=False)
                    .head(3))

print("En fazla kazanç sağlanan mağaza ve kategorisi: ")
print(magaza_kategori)

### J tane göstererek kategoriyi daha iyi gözeletemek istedim. Spor kategorisi mağazalar arasında
## en fazla kazancı ilk olarak 2. mağazoda sonrasta da 1. mağazoda getirerek spor kategorisinin
## en fazla kazancı sağladığı netlettirmiştir. Mağaza olarak da 2. mağazadır.

```

```

En fazla kazanç sağlanan mağaza ve kategorisi:
   store_id      name  total_revenue
30          2  Sports        2761.85
14          1  Sports        2552.36
13          1  Sci-Fi        2385.47

```

Kübra Nur Babacan