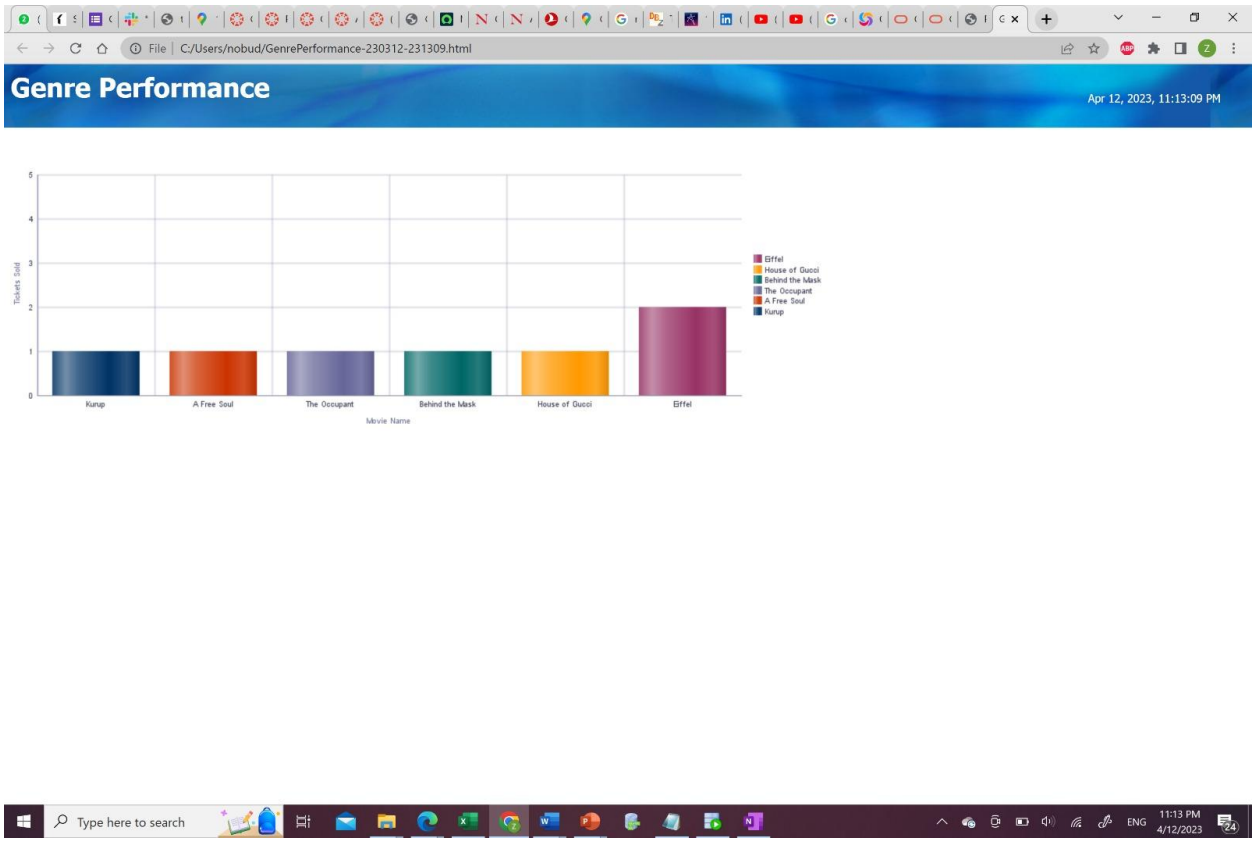
PROJECT REPORT

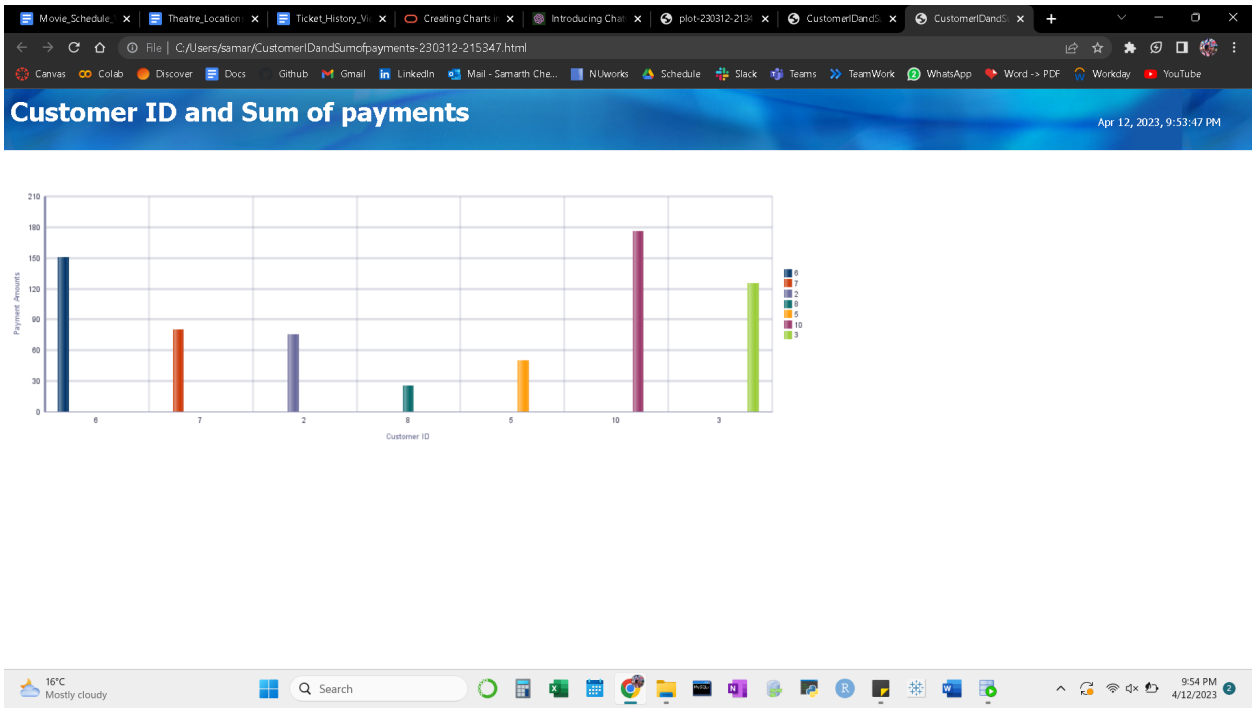
GENRE-WISE PERFORMANCE REPORT



The above plot was created using the User Defined Report feature of the Oracle SQL Developer. This plot gives us information regarding the genre wise performance based on the number of movie names. We are using the below code to obtain this plot:

```
SELECT movie_name, movie_name, Count(movie_name) as total_sales
FROM ticket join scheduled_show on ticket.show_id = scheduled_show.show_id
join movie on scheduled_show.movie_id = movie.movie_id
GROUP BY movie_name
```

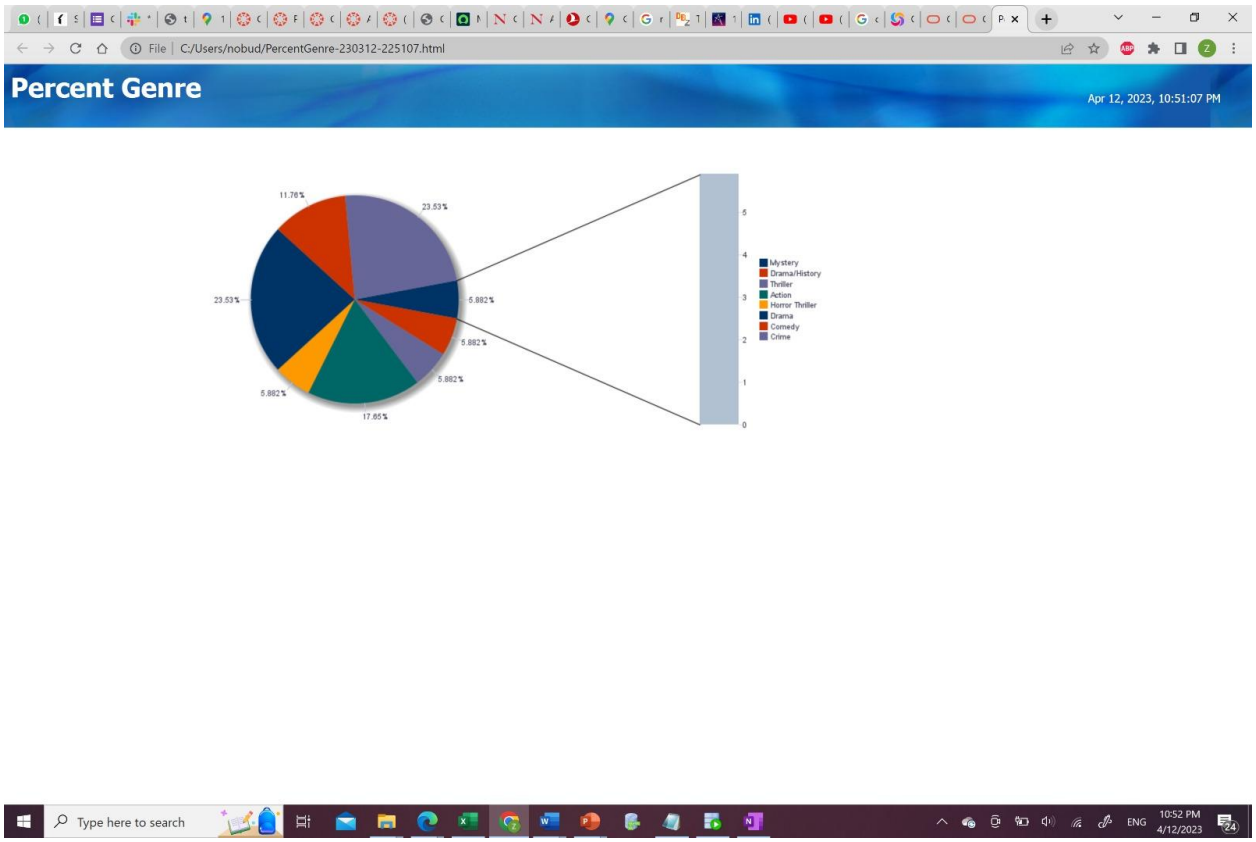
CUSTOMER ID VS SUM OF PAYMENTS REPORT



This indicates the basic chart which is produced when we want to display the customer ID and their respective sum of payments with a minimum payment amount of 20 and is grouped according to the customer ID. We have used the below query to generate this plot:

```
select p.customer_id, sum(p.payment_amount)
from payment p
join customer c on p.customer_id = c.customer_id
where p.payment_amount > 20
group by p.customer_id
```

MOVIES AND THEIR RESPECTIVE GENRES REPORT



The above pie chart represents what genre a particular movie belongs to and what is the percentage-wise distribution of every genre. We have used the following code to implement this:

```
SELECT genre, genre, COUNT() * 100.0 / (SELECT COUNT() FROM movie) AS genre_percent
FROM movie
GROUP BY genre, genre;
```

TICKET HISTORY VIEW REPORT

Below is the code which can be used to create a view that has all the information related to that particular ticket booking and its history:

```
CREATE or REPLACE VIEW tickets_history AS
SELECT t.ticket_id,t.customer_id, m.movie_name, ss.start_date_time, ss.end_date_time,
sc.screen_id, t.seat_list, p.payment_amount, p.payment_status, LISTAGG(a.addon_name, ', ')
WITHIN GROUP (ORDER BY a.addon_id) AS addon_names,
      SUM(a.price * ca.addon_quantity) AS addon_total_price
FROM ticket t
INNER JOIN scheduled_show ss ON t.show_id = ss.show_id
INNER JOIN movie_screen sc ON ss.screen_id = sc.screen_id
INNER JOIN movie m ON ss.movie_id = m.movie_id
INNER JOIN payment p ON t.payment_id = p.payment_id
LEFT JOIN customer_addon ca ON t.ticket_id = ca.ticket_id
LEFT JOIN addon a ON ca.addon_id = a.addon_id
GROUP BY t.ticket_id,t.customer_id, m.movie_name, ss.start_date_time, ss.end_date_time,
sc.screen_id, t.seat_list, p.payment_amount, p.payment_status;

select *from movieadmin.tickets_history;
```

The above code generates the following output:

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.313 seconds

	TICKET_ID	CUSTOMER_ID	MOVIE_NAME	START_DATE_TIME	END_DATE_TIME	SCREEN_ID	SEAT_LIST	PAYMENT_AMOUNT	PAYMENT_STATUS	ADDON_NAMES	ADDON_TOTAL_PRICE
1	1	11	The Occupant	6/2/2021 08:00:00	6/2/2021 11:30:00	3	25F	176	Successful	PopCorm, Fries, Candy	26
2	4	11	The Occupant	6/2/2021 08:00:00	6/2/2021 11:30:00	3	25A	176	Successful	PopCorm, Fries, Candy	26
3	5	11	The Occupant	6/2/2021 08:00:00	6/2/2021 11:30:00	3	25B	176	Successful	PopCorm, Fries, Candy	26
4	6	11	The Occupant	6/2/2021 08:00:00	6/2/2021 11:30:00	3	25C	176	Successful	PopCorm, Fries, Candy	26

MOVIE SCHEDULE VIEW REPORT

Below is the code which can be used to create a view that has all the information related to the movie schedule:

```
CREATE or REPLACE VIEW movie_schedule AS
SELECT m.movie_name,m.Genre,m.movie_language,ss.show_id, ss.start_date_time,
ss.end_date_time, sc.screen_id
FROM scheduled_show ss
INNER JOIN movie_screen sc ON ss.screen_id = sc.screen_id
INNER JOIN movie m ON ss.movie_id = m.movie_id;

SELECT * FROM movieadmin.movie_schedule;
```

The above code generates the following output:

Query Result

SQL | All Rows Fetched: 8 in 0.871 seconds

	MOVIE_NAME	GENRE	MOVIE_LANGUAGE	SHOW_ID	START_DATE_TIME	END_DATE_TIME	SCREEN_ID
1	A Free Soul	Crime	English	3	11/1/2021 11:00:00	11/1/2021 02:30:00	2
2	A Free Soul	Crime	English	8	2023-05-01 18:30:00	2023-05-01 21:00:00	1
3	A Free Soul	Crime	English	9	2023-05-01 18:30:00	2023-05-01 21:00:00	1
4	Iron Man	Drama	English	4	11/1/2021 17:00:00	11/1/2021 19:30:00	2
5	Behind the Mask	Crime	English	5	11/1/2021 20:00:00	11/1/2021 22:30:00	2
6	The Occupant	Thriller	Spanish	7	6/2/2021 08:00:00	6/2/2021 11:30:00	3
7	House of Gucci	Crime	English	6	5/2/2021 10:00:00	5/2/2021 00:30:00	2
8	Eiffel	Drama	French	2	9/12/2021 10:30:00	9/12/2021 01:30:00	1

THEATRE LOCATIONS VIEW REPORT

Below is the code which can be used to create a view that has all the information related to the different theater locations:

```
CREATE or REPLACE VIEW theatre_locations_view AS
SELECT t.theatre_id, t.theatre_name, l.theatre_city, l.theatre_state
FROM theatre t
INNER JOIN theatre_location l ON t.location_id = l.location_id;
/

SELECT * FROM movieadmin.theatre_locations_view where theatre_city = 'Albany';
```

The above code generates the following output:

Query Result

SQL | All Rows Fetched: 17 in 0.14 seconds

	THEATRE_ID	THEATRE_NAME	THEATRE_CITY	THEATRE_STATE
1	1	AMC	Boston	MA
2	18	Orion Mall	Boston	MA
3	2	Regal Entertainment Group	Albany	NY
4	8	AMC Anaheim Gardenwalk 6	Albany	NY
5	9	7th Street Theatre	Albany	NY
6	10	777 Theatre	Albany	NY
7	13	Harkins Theatres	Albany	NY
8	7	Stages Theatre	Philadelphia	PA
9	15	Regal Entertainment Group	Philadelphia	PA
10	16	MC Dine-In Rio Cinemas 18	Philadelphia	PA
11	3	Artcraft Theatre	Phoenix	AZ
12	4	Cinemark Theatres	Phoenix	AZ
13	6	Marcus Corporation	Columbus	OH
14	11	AMC 34th Street 14	Columbus	OH
15	12	AMC Broadstreet 7	Columbus	OH
16	5	Astor Theater	Chicago	IL
17	14	AMC Classic Findlay 12	Baltimore	MD

SEAT AVAILABILITY VIEW

Below is the code which can be used to create a view that has all the information related to the number of available seats:

```
CREATE or replace VIEW seats_available AS
SELECT m.movie_name,ss.start_date_time, sc.screen_id,s.seat_number,s.seat_status
FROM seat s
INNER JOIN scheduled_show ss ON s.screen_id = ss.screen_id
INNER JOIN movie_screen sc ON ss.screen_id = sc.screen_id
INNER JOIN movie m ON ss.movie_id = m.movie_id
where s.seat_status = 'Y';
/


select * from movieadmin.seats_available;
```

The above code generates the following output:

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3

SQL | All Rows Fetched: 24 in 0.29 seconds

	MOVIE_NAME	START_DATE_TIME	SCREEN_ID	SEAT_NUMBER	SEAT_STATUS
1	A Free Soul	2023-05-01 18:30:00	1	5B	Y
2	A Free Soul	2023-05-01 18:30:00	1	5B	Y
3	Eiffel	9/12/2021 10:30:00	1	5B	Y
4	A Free Soul	2023-05-01 18:30:00	1	7C	Y
5	A Free Soul	2023-05-01 18:30:00	1	7C	Y
6	Eiffel	9/12/2021 10:30:00	1	7C	Y
7	A Free Soul	11/1/2021 11:00:00	2	10A	Y
8	Iron Man	11/1/2021 17:00:00	2	10A	Y
9	Behind the Mask	11/1/2021 20:00:00	2	10A	Y
10	House of Gucci	5/2/2021 10:00:00	2	10A	Y
11	A Free Soul	11/1/2021 11:00:00	2	10B	Y
12	Iron Man	11/1/2021 17:00:00	2	10B	Y
13	Behind the Mask	11/1/2021 20:00:00	2	10B	Y
14	House of Gucci	5/2/2021 10:00:00	2	10B	Y
15	A Free Soul	2023-05-01 18:30:00	1	10D	Y
16	A Free Soul	2023-05-01 18:30:00	1	10D	Y
17	Eiffel	9/12/2021 10:30:00	1	10D	Y
18	A Free Soul	2023-05-01 18:30:00	1	10E	Y
19	A Free Soul	2023-05-01 18:30:00	1	10E	Y
20	Eiffel	9/12/2021 10:30:00	1	10E	Y
21	A Free Soul	2023-05-01 18:30:00	1	20A	Y
22	A Free Soul	2023-05-01 18:30:00	1	20A	Y
23	Eiffel	9/12/2021 10:30:00	1	20A	Y
24	The Occupant	6/2/2021 08:00:00	3	25D	Y

cloudwallet/dmmd_ticketmasters/userpackage-05.sql | Line 60 C