DATA ANALYST INTERNSHIP REPORT WEEK-1

TASK 1:Onboarding & Analytics Foundations

• Introduction to data analytics, types of data, and analytics lifecycle.
• Install and set up: Python (Anaconda/Jupyter), Excel, Power BI or Tableau Public.
• Complete a beginner course or YouTube playlist (Khan Academy / IBM / Analytics Vidhya).
• Task: Analyze a simple CSV file using Excel – calculate averages, use pivot tables, Charts.

1. Calculating an Average with the AVERAGE Function

To compute the arithmetic mean of a numeric column (for example, the “mental_health_score” column in your CSV), you’d:

1. Open the CSV in Excel.
2. Suppose your scores run from cell D2 down to D101. In an empty cell (e.g., D102), enter:
3. =AVERAGE(D2:D101)

---

2. Creating a PivotTable to Summarize Data

PivotTables let you slice and dice your data without writing formulas. To create one:

1. Select any cell in your dataset (or the entire table).
2. Go to the **Insert** tab in the Ribbon and click **PivotTable**
3. In the **Create PivotTable** dialog, choose whether to place it on a new worksheet or in an existing one, then click **OK**.
4. You’ll see a **PivotTable Fields** pane. Drag fields like:
    - **Institution Type** into **Rows**
    - **Gender** into **Columns** (optional)
    - **mental_health_score** into **Values** (it will default to “Sum of mental_health_score”)
5. Click on the dropdown in **Values**, choose **Value Field Settings**, and select **Average** instead of **Sum**
6. Click **OK**, and your PivotTable will show the average mental health score by institution type (and gender, if used).

3. Adding Charts for Visualization

Once your PivotTable is in place:

1. Click anywhere inside the PivotTable.

2. Go to the **PivotTable Analyze** (or **Options**) tab → **PivotChart**.
3. Pick a chart type (e.g., Column or Bar).
4. Click **OK** to insert the chart, which will update automatically as you change the PivotTable

D7024 =AVERAGE(D1:D7023)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Age	Course	Gender	CGPA	Stress_Leve	Depression	Anxiety_Sc	Sleep_Qual	Physical_Ac	Diet_Qualit	Social_Supp	Relationshi	Substance_	Counseling_	Family_Hist	Chronic_Illn	Financial_S	Extracurric	Semester_(	Residence_Typ
7003	30	Others	Male	3.48	0	4	0	Good	Low	Average	Moderate	In a Relatio	Never	Frequently	Yes	No	3	High	28	Off-Campus
7004	24	Engineerin	Male	3.57	2	1	0	Good	Moderate	Average	High	Married	Never	Never	No	No	3	Low	20	On-Campus
7005	26	Business	Female	3.01	1	3	3	Average	Low	Good	Moderate	Single	Never	Occasionall	Yes	No	4	High	17	On-Campus
7006	33	Engineerin	Female	3.63	1	1	5	Average	Moderate	Average	Low	Single	Never	Occasionall	Yes	No	0	Low	24	On-Campus
7007	24	Others	Male	3.82	5	0	0	Average	Moderate	Good	High	In a Relatio	Never	Never	No	No	3	Low	16	With Family
7008	29	Medical	Female		1	2	2	Average	High	Average	Low	Single	Never	Frequently	No	No	2	Moderate	20	Off-Campus
7009	24	Computer S	Female	3.63	3	4	1	Good	High	Good	Low	In a Relatio	Occasionall	Occasionall	No	No	5	Moderate	17	On-Campus
7010	18	Medical	Male	3.58	3	3	1	Poor	Low	Average	Moderate	Single	Never	Never	No	No	3	Low	27	On-Campus
7011	18	Computer S	Female	3.67	1	3	4	Good	High	Average	Moderate	Single	Never	Never	No	No	3	Low	17	On-Campus
7012	23	Others	Male	3.53	4	1	1	Good	Low	Average	Low	Single	Never	Never	No	No	3	Low	25	With Family
7013	21	Medical	Female	3.26	4	3	3	Average	High	Average	Moderate	Single	Never	Never	No	No	2	High	15	On-Campus
7014	22	Computer S	Male		1	4	0	Average	High	Poor	High	Single	Never	Occasionall	No	No	2	High	29	On-Campus
7015	24	Law	Male	3.98	0	0	3	Poor	Moderate	Poor	High	Single		Occasionall	No	No	2	Low	22	Off-Campus
7016	26	Medical	Male		0	2	1	Average	Moderate	Average	High	Single	Never	Occasionall	No	No	5	High	24	On-Campus
7017	18	Engineerin	Female	3.64	5	2	2	Average	High	Average	Low	Single	Frequently	Never	Yes	No	2	Low	20	On-Campus
7018	20	Law	Female	3.33	1	1	1	Average	Low	Average	Moderate	In a Relatio	Never	Occasionall	No	No	3	Moderate	29	Off-Campus
7019	20	Law	Female	3.69	3	1	5	Good	Low	Poor	High	Single	Never	Frequently	Yes	No	1	High	26	Off-Campus
7020	24	Medical	Female	3.73	3	4	1	Good	Moderate	Average	High	Single	Never	Occasionall	No	No	3	Low	15	Off-Campus
7021	26	Others	Male	3.65	4	5	1	Good	High	Poor	Moderate	Married	Never	Occasionall	No	Yes	4	Moderate	17	Off-Campus
7022	24	Medical	Male	3.65	4	3	4	Average	High	Poor	Moderate	Single	Never	Never	No	Yes	4	Moderate	18	Off-Campus
7023	22	Medical	Female		3	5	0	Average	High	Average	Moderate	In a Relatio	Never	Never	No	No	2	Low	17	With Family
7024				3.49127	2.427941	2.254486	2.300484										2.453005		22.01054	

Fig 1. AVERAGE Calculation in Excel

Count of Course	Column Labels				
Row Labels	Average	Good	Poor	(blank)	Grand Total
Business	**257**	**394**	**66**		**717**
Female	129	200	37		366
Male	128	194	29		351
Computer Science	**385**	**527**	**116**		**1028**
Female	172	272	62		506
Male	213	255	54		522
Engineering	**443**	**519**	**110**		**1072**
Female	235	271	51		557
Male	208	248	59		515
Law	**549**	**686**	**150**		**1385**
Female	261	355	77		693
Male	288	331	73		692
Medical	**833**	**1086**	**186**		**2105**
Female	408	517	88		1013
Male	425	569	98		1092
Others	**268**	**377**	**70**		**715**
Female	138	168	34		340
Male	130	209	36		375
(blank)					
(blank)					
Grand Total	**2735**	**3589**	**698**		**7022**

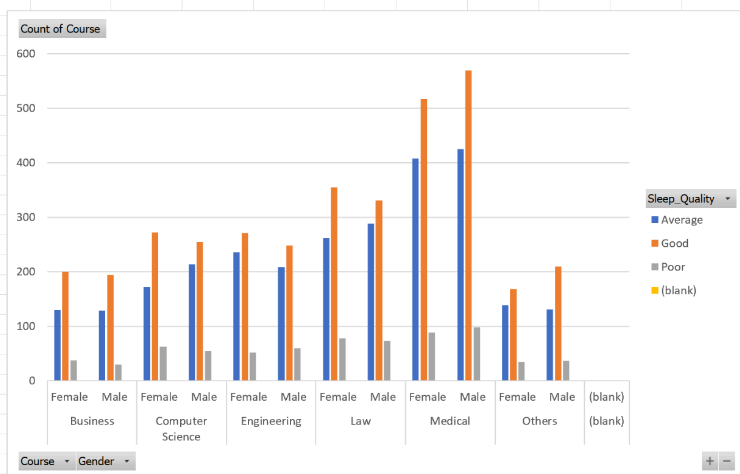


Fig 2. Pivot Table Creation and Chart Formation

TASK 2:Data Cleaning & Manipulation in Excel & Python

• Learn about missing data, duplicates, data formatting.
• Perform data cleaning on a messy Excel dataset.
• Use Python + Pandas to load, clean, and summarize datasets.
• Task: Clean and explore a public dataset (Titanic Dataset Used).

Colab Link : Internship_Task 2_22IT157

TASK 3:Exploratory Data Analysis (EDA)

• Learn descriptive statistics and visualizations.
• Use Python (matplotlib, seaborn, pandas_profiling) to perform EDA.
• Task: Perform EDA on a real-world dataset (Netflix Dataset used).
• Create a Jupyter Notebook report with insights and graphs.

Colab Link : Internship_Task 3_22IT157

TASK 4:Data Visualization Tools

• Introduction to Power BI or Tableau.
• Import data, create dashboards, slicers, and KPIs.
• Task: Build a dashboard showing regional sales, profit trends, or student
Performance.

Certainly! Below is a **detailed explanation of the steps followed to create the Student Performance Dashboard**, based on the provided image, written in a definitive format suitable for a report:

STUDENT PERFORMANCE DASHBOARD

Steps Followed to Create the Dashboard

1. Data Import

The dataset was imported into Power BI. The dataset contained fields such as:

- gender
- race/ethnicity
- parental level of education
- lunch

- test preparation course
- math score
- reading score
- writing score

2. Data Cleaning and Transformation

- Ensured all column headers were appropriately named and formatted.
- Verified that numerical columns (math score, reading score, writing score) were recognized as numeric data types.
- Cleaned any missing or inconsistent entries if present.

Created a new **Average Score** column using the formula:

Average Score = (math score + reading score + writing score) / 3

3. Dashboard Design and Visual Creation

A. Title and KPI Metrics

- Added a **title**: "Student Performance Dashboard".
- Placed **KPI cards** at the top displaying:
    - **Average Score** (67.77)
    - **Average Math Score** (66.09)
    - **Average Reading Score** (69.17)
    - **Average Writing Score** (68.05)

B. Bar Chart – Average Score by Gender

- Visual Type: Horizontal Bar Chart.

- Axis: gender on Y-axis, Average Score on X-axis.
- Insight: Allows comparison of performance between male and female students.

C. Donut Chart – Average of Student by Race/Ethnicity

- Visual Type: Donut Chart.
- Fields used: race/ethnicity and corresponding Average Score.
- Color-coded each group from A to E and displayed both value and percentage distribution.

D. Stacked Bar Chart – Score by Parental Level of Education

- Visual Type: Stacked Bar Chart.
- X-axis: parental level of education
- Y-axis: Sum of math score, reading score, writing score (grouped by subject).
- Legend: Color-coded for each subject score.
- Insight: Demonstrates the relationship between parents' education levels and student performance.

E. Slicer Filters (Right Panel)

Five interactive slicers were added to filter data across all visualizations:

- **Gender**
- **Race/Ethnicity**
- **Parental Level of Education**
- **Lunch Type**
- **Test Preparation Course**

Key Insights from the Dashboard

1. Overall Student Performance

- The **average score across all students is 67.77**, indicating moderate overall performance.

2. Gender-Based Performance

- **Female students** have a slightly higher average score than **male students**.

3. Race/Ethnicity Group Performance

- **Group E** has the highest average score (72.8), suggesting better performance than other groups.
- **Group A** has the lowest average (63.0), indicating a performance gap that may need attention.

4. Impact of Parental Education

- Students whose parents have a **master's or bachelor's degree** perform significantly better across all subjects compared to those whose parents have **only high school education**.
- There is a clear positive correlation between **parental education level** and student performance.

5. Subject-Wise Trends

- Among the three subjects:
    - **Reading scores** tend to be slightly higher on average.
    - **Math scores** are comparatively lower.

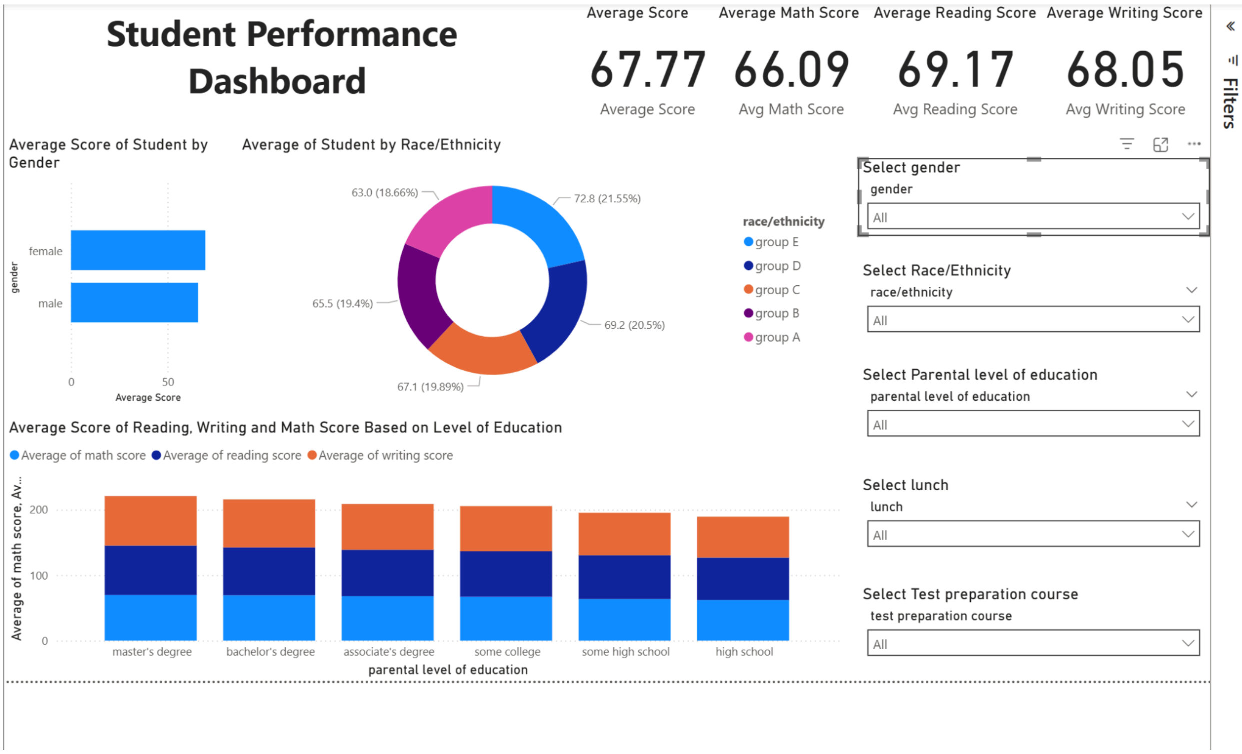


Fig 3. Student Performance Dashboard

TASK 5:SQL For Data Analytics

- Learn basic to intermediate SQL: SELECT, WHERE, JOIN, GROUP BY, etc.
- Practice on Mode Analytics, Hackerrank, or SQLZoo.
- Task: Solve 10 real-world business queries on a sample dataset.
- Bonus: Connect SQL with Power BI and visualize query results.

SQL Queries Run:

1. Total Revenue by Store

```
SELECT
  s.store_id,
  CONCAT('Store ', s.store_id) AS store_name,
  ROUND(SUM(p.amount), 2) AS total_revenue
FROM payment p
JOIN rental r ON p.rental_id = r.rental_id
JOIN inventory i ON r.inventory_id = i.inventory_id
JOIN store s     ON i.store_id    = s.store_id
GROUP BY s.store_id;
```

2. Top 5 Customers by Total Spend

```
SELECT
  c.customer_id,
  CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
  ROUND(SUM(p.amount), 2) AS total_spent
FROM customer c
JOIN payment p ON c.customer_id = p.customer_id
GROUP BY c.customer_id
ORDER BY total_spent DESC
LIMIT 5;
```

3. Most Rented Films (Top 10)

```
SELECT
  f.film_id,
  f.title,
  COUNT(r.rental_id) AS times_rented
FROM film f
JOIN inventory i ON f.film_id = i.film_id
JOIN rental r   ON i.inventory_id = r.inventory_id
GROUP BY f.film_id
ORDER BY times_rented DESC
LIMIT 10;
```

4. Highest-Grossing Film Categories

```
SELECT
  cat.category_id,
  cat.name AS category,
  ROUND(SUM(p.amount),2) AS revenue
FROM category cat
JOIN film_category fc ON cat.category_id = fc.category_id
JOIN film f           ON fc.film_id       = f.film_id
JOIN inventory i      ON f.film_id        = i.film_id
JOIN rental r         ON i.inventory_id   = r.inventory_id
JOIN payment p        ON r.rental_id      = p.rental_id
GROUP BY cat.category_id
ORDER BY revenue DESC;
```

5. Current Overdue Rentals

```
SELECT
  r.rental_id,
  c.customer_id,
  CONCAT(c.first_name,' ',c.last_name) AS customer,
  r.rental_date,
  r.return_date
FROM rental r
JOIN customer c ON r.customer_id = c.customer_id
WHERE r.return_date IS NULL
  AND r.rental_date < NOW() - INTERVAL 7 DAY;
```

6. Inventory Count by City

```
SELECT
  ci.city_id,
  ci.city,
  COUNT(i.inventory_id) AS inventory_count
FROM inventory i
JOIN store s    ON i.store_id     = s.store_id
JOIN address a  ON s.address_id   = a.address_id
JOIN city ci    ON a.city_id      = ci.city_id
GROUP BY ci.city_id;
```

7. Films Never Rented

```
SELECT
  f.film_id,
  f.title
FROM film f
LEFT JOIN inventory i ON f.film_id      = i.film_id
LEFT JOIN rental r    ON i.inventory_id = r.inventory_id
WHERE r.rental_id IS NULL;
```

8. Average Rental Duration by Store

```
SELECT
  s.store_id,
  ROUND(AVG(TIMESTAMPDIFF(DAY, r.rental_date, r.return_date)),2) AS avg_rental_days
FROM rental r
JOIN inventory i ON r.inventory_id = i.inventory_id
JOIN store s     ON i.store_id     = s.store_id
```

```
WHERE r.return_date IS NOT NULL
GROUP BY s.store_id;
```

9. Revenue by Staff Member

```
SELECT
  st.staff_id,
  CONCAT(st.first_name,' ',st.last_name) AS staff_name,
  ROUND(SUM(p.amount),2) AS total_revenue
FROM staff st
JOIN payment p ON st.staff_id = p.staff_id
GROUP BY st.staff_id;
```

10. Customers with No Rentals (Inactive)

```
SELECT
  c.customer_id,
  CONCAT(c.first_name,' ',c.last_name) AS customer
FROM customer c
LEFT JOIN rental r ON c.customer_id = r.customer_id
WHERE r.rental_id IS NULL;
```

Connecting SQL with PowerBI:

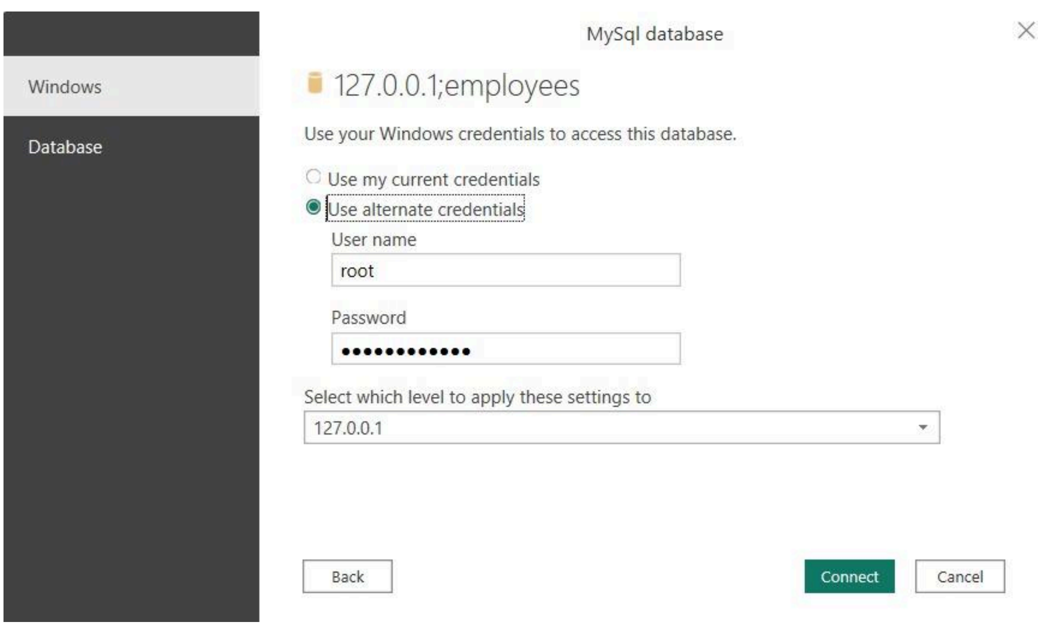


Fig 4. Connecting MySQL Workbench server with PowerBI

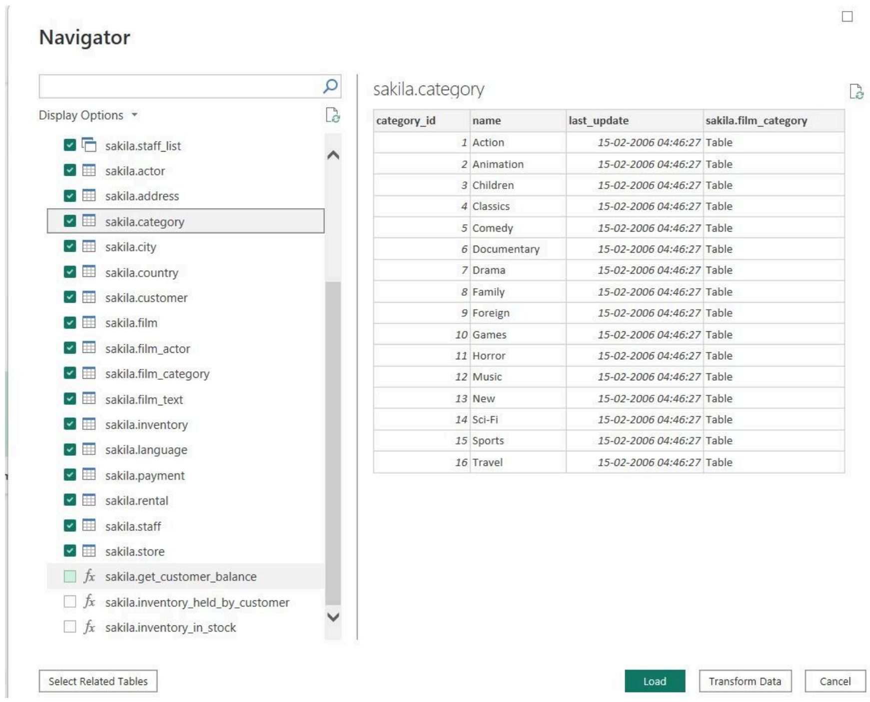


Fig 5. Selecting appropriate SQL files for Dashboard Creation

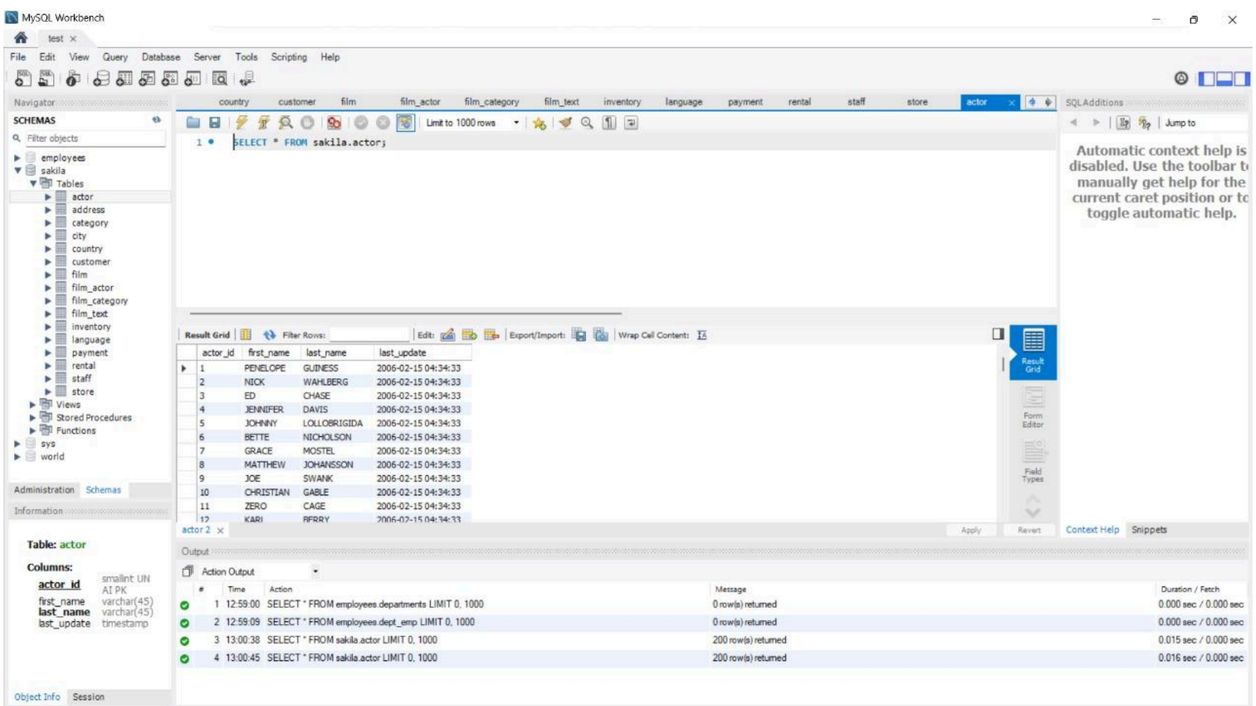


Fig 6. Running Queries from Workbench

Dashboard 1:

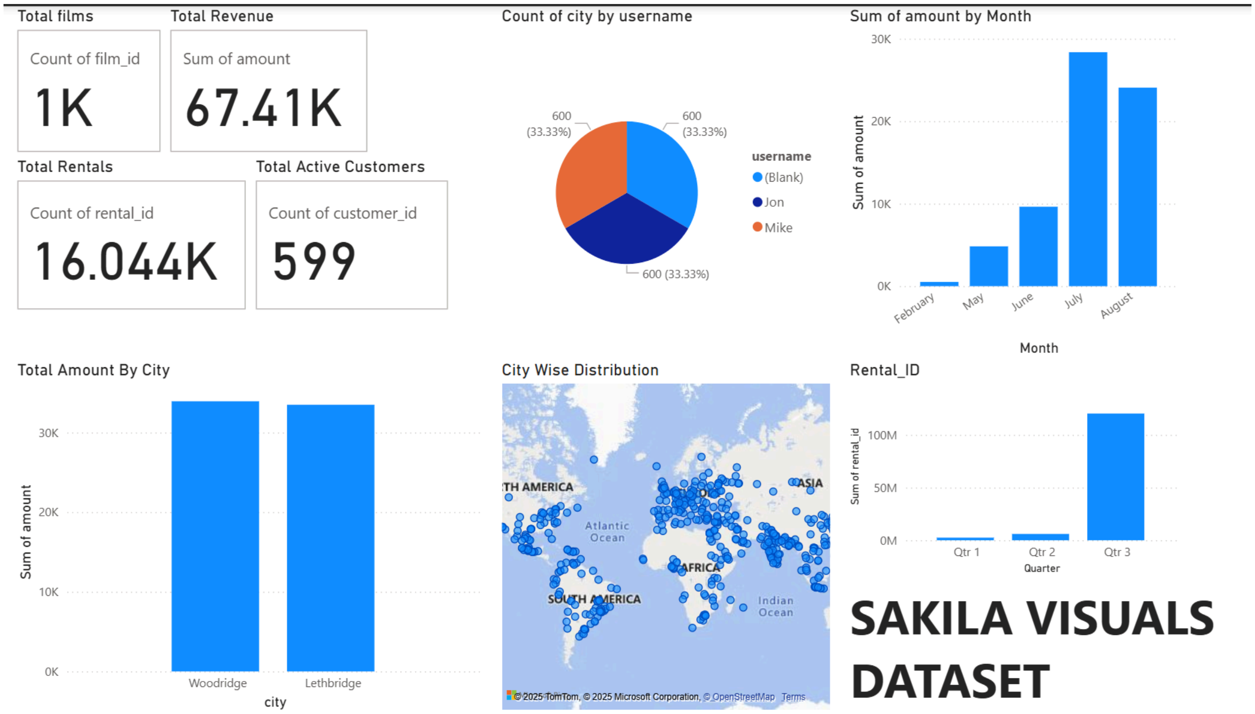
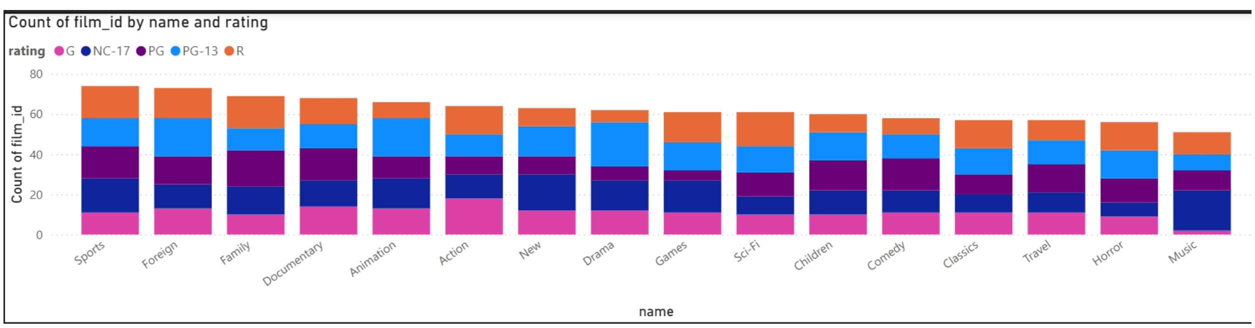


Fig 7. Sakila Dataset Dashboard 1

Dashboard 2:



title	Sum of length	rating	Sum of release_year	Sum of rental_rate
ACADEMY DINOSAUR	86	PG	2006	0.99
ACE GOLDFINGER	48	G	2006	4.99
ADAPTATION HOLES	50	NC-17	2006	2.99
AFFAIR PREJUDICE	117	G	2006	2.99
AFRICAN EGG	130	G	2006	2.99
AGENT TRUMAN	169	PG	2006	2.99
AIRPLANE SIERRA	62	PG-13	2006	4.99
AIRPORT POLLOCK	54	R	2006	4.99
ALABAMA DEVIL	114	PG-13	2006	2.99
ALADDIN CALENDAR	63	NC-17	2006	4.99
ALAMO VIDEOTAPE	126	G	2006	0.99
ALASKA PHANTOM	136	PG	2006	0.99
Total	**115272**		**2006000**	**2,980.00**

Fig 8. Sakila Dataset Dashboard 2

Dashboard #1: “Sakila Visuals Dataset”

1. Data Preparation

- **Import** the core Sakila tables:
  film, rental, payment, customer, inventory, store, address, city, plus any user-login or username mapping table.

2. KPI Cards

Placed at the top-left to give at-a-glance metrics:

1. **Total Films** = COUNT(DISTINCT film_id) → **1 K**
2. **Total Revenue** = SUM(payment_amount) → **67.41 K**
3. **Total Rentals** = COUNT(rental_id) → **16.044 K**
4. **Total Active Customers** = COUNT(DISTINCT customer_id) → **599**

3. Pie Chart: Count of City by Username

- **Group by** username and count **distinct cities** or simply count rows by username.
- Shows three equal slices (**600**, 33.3% each) for (Blank), Jon, and Mike.

4. Column Chart: Sum of Amount by Month

- **X-axis**: Month (ordered Feb → Aug)
- **Y-axis**: SUM(payment_amount)
- Bars reveal a ramp-up from near zero in February to a peak in **July (~28 K)**, then a small drop in August.

5. Column Chart: Total Amount by City

- **X-axis**: city (Woodridge, Lethbridge)

- **Y-axis**: SUM(payment_amount)
- Both cities tie at roughly **35 K** revenue each.

6. Map: City-Wise Distribution

- **Latitude/Longitude** from the city table plotted as points.
- Bubble size (or uniform) marks every store's customer/rental location worldwide.

7. Column Chart: Rental_ID by Quarter

- **X-axis**: Quarter (Q1, Q2, Q3)
- **Y-axis**: SUM(rental_id) (or count rentals)
- Q3 dwarfs Q1/Q2, indicating bulk of activity in the third quarter.

Insights from Dashboard #1

1. **Strong Q3 Seasonality**
   The vast majority of rentals (and associated revenue) occur in **Q3**, suggesting a summer-peak demand for DVDs.
2. **Rapid Month-Over-Month Growth**
   Revenue climbs from almost zero in February to **~28 K in July**, revealing either a promotional campaign or seasonal customer behavior.
3. **Geographic Concentration**
   Two cities—**Woodridge** and **Lethbridge**—generate virtually all revenue, indicating focus markets or store locations.
4. **User Engagement Split**
   Three user accounts (including blank/anonymous) contribute equally to city counts, suggesting three major channels or staff members handling transactions.
5. **Customer Base Depth**
   Nearly **600 active customers** generating over **16 K rentals** underscores solid repeat usage.

Dashboard #2: "Film Inventory & Ratings Analysis"

1. Data Preparation

- **Import**: film, film_category → category, plus rating and length fields.
- **Join** so each row has:
  film_id, title, name (category), rating, length, release_year, rental_rate

2. Stacked Column Chart: Count of Film_ID by Category and Rating

- **X-axis**: category name
- **Y-axis**: COUNT(film_id)
- **Stack**: rating (G, PG, PG-13, R, NC-17)
- Visually compares how many films each category has at each rating level.

3. Pie Chart: Count of Film_ID by Rating

- **Slices** for each rating, showing both absolute counts and percentages:
  - **PG-13**: 223 (22.3%)
  - **NC-17**: 210 (21%)
  - **R**: 195 (19.5%)
  - **PG**: 194 (19.4%)
  - **G**: 178 (17.8%)

4. Detail Table: Film Attributes

- **Columns**: title, SUM(length), rating, SUM(release_year), SUM(rental_rate)
- **Totals** row at bottom:
  - **Total Length**: 115,272 minutes
  - **Total Release Years**: 2,006,000 (sum of 1,000 films × 2006)

- **Total Rental Rate**: 2,980.00

Insights from Dashboard #2

1. **Category-Rating Profiles**
   - **Sports** and **Foreign** lead in overall film counts (≈75 each), heavily weighted toward PG-13 and R.
   - **Action** has the single largest PG slice (≈20 films).
2. **Rating Distribution Is Balanced**
   No single rating dominates—each hovers between **17–22%** of total films, ensuring a diverse library.
3. **Library Depth & Pricing**
   - Average film length ≈115 minutes.
   - Uniform release year (2006), indicating a snapshot in time.
   - Rental rates cluster at common price points (0.99, 2.99, 4.99).
4. **Targeting & Licensing**
   - A strong PG-13/R presence suggests targeting teen/adult demographics.
   - NC-17 and G titles are fewer—niche or specialty content.

TASK 6: Capstone Project & Final Report

• Choose or receive a real-world dataset (from Kaggle, Google, or organization).
• Perform:
  o Data cleaning (Excel or Python)
  o EDA (Python)
  o Visualization Dashboard (Power BI or Tableau)
  o Summary of insights and recommendations.

1. Data Cleaning & Preparation

1. **Load the raw CSV** into your Python/Excel environment.

Assess missingness

```
print(df.isna().sum())
```

2. – Drop or impute any columns that are more than ~30 % missing.

Deduplicate

```
df.drop_duplicates(inplace=True)
```

3. **Feature engineering**
    - **country_count**: how many countries each title is available in
    - **title_length**: number of characters in the title
    - **title_age**: 2025 − releaseYear
4. **Clean multi-valued fields**
   – Split the comma-separated genres into one row per genre for genre-level aggregations.

2. Exploratory Data Analysis (Python)

1. **Distribution of IMDb Ratings**
   – Histogram + KDE to see if ratings cluster or are uniform.
2. **Rating spread by Type** (movie vs. tv)
   – Boxplots of imdbAverageRating to compare variance and medians.
3. **Top 10 Genres by Title Count**
   – Bar chart of exploded-and-counted genres.
4. **Pairwise Relationships** (releaseYear, imdbAverageRating, imdbNumVotes)
   – Pairplot (with regression lines) plus a correlation heatmap.
5. **Time Trends**

- **Annual Title Releases**: line plot of count by releaseYear.
- **Annual Average Rating**: line plot of mean imdbAverageRating by year.

3. Dashboard Design & Visuals

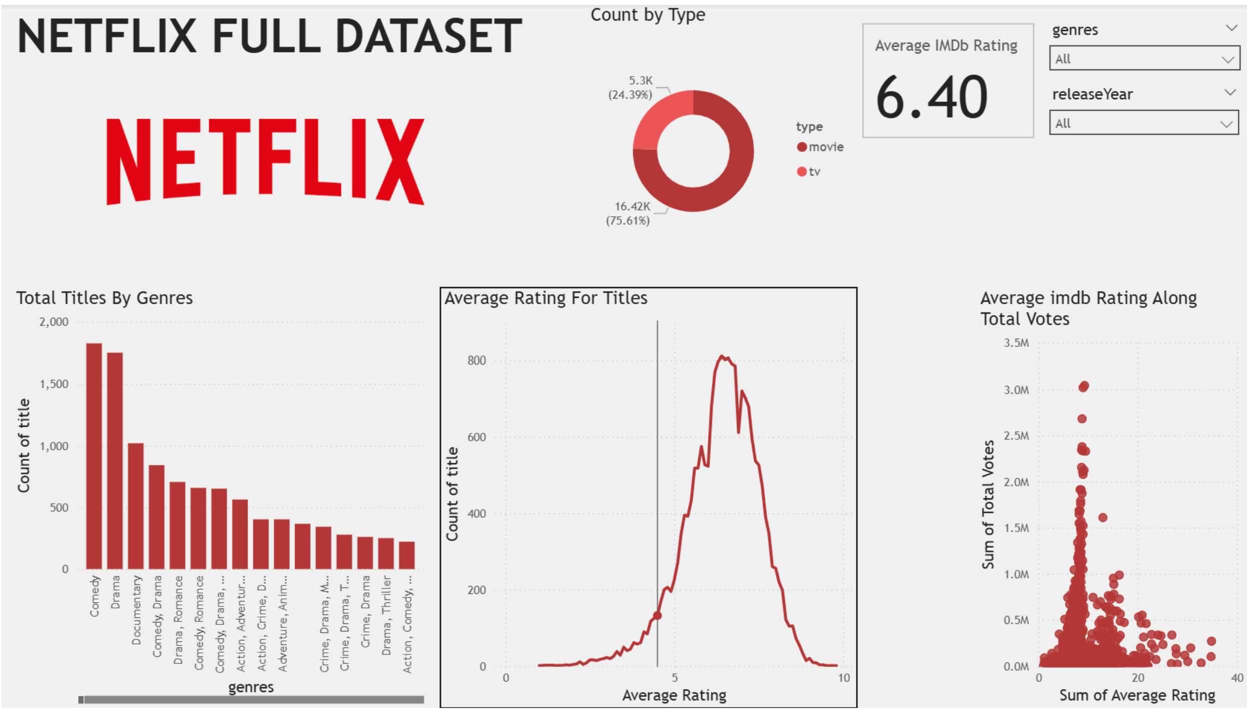


Fig 9. Netflix Clean Dataset Dashboard

4. Key Insights & Recommendations

1. **Catalog Composition**
   - **Comedy** (≈1,800 titles) and **Drama** (≈1,700) dominate.
   - Niche genres (e.g., "Action/Comedy," "Documentary") tail off below 500 titles.
2. **Rating Distribution**
   - IMDb ratings cluster around **6–8**, with very few < 3 or > 9.

- **TV shows** tend to have slightly higher median ratings than **movies**.

3. **Temporal Trends**

   - Title additions **accelerate** sharply from 2015 onward—reflecting Netflix's ramp-up in originals.

   - **Average ratings** dip slightly after 2018, suggesting mixed reception to aggressive content expansion.

4. **Engagement Signal**

   - There's a **moderate positive correlation** between imdbNumVotes and imdbAverageRating—higher-voted titles tend to score better.

   - A small cluster of low-vote, high-rating outliers suggests hidden gems that could be promoted.

Next Steps

- **Content Gaps**: Identify genres with high viewer ratings but low title counts to guide acquisitions (e.g., "Sci-Fi/Kids").

- **Quality Over Quantity**: Monitor the post-2018 dip in average ratings—consider more selective commissioning of originals.

- **Personalization Signals**: Use the scatter's high-vote high-rating titles to seed recommendation algorithms.