

Calgary Gas and Permit Inspections

*Enhancing Operational Oversight Through Data-Driven Gas Permit and
Inspection*

VISHAL JAVVAJI





INDEX

S. No.	Topic	Page No.
1	Introduction	3
2	Project Overview	3
3	Purpose of the System	3
4	Database Design	4
5	Key Tables and Their Roles	5 – 7
6	Relationships between Tables	7-8
7	Data Integrity Rules	8-10
8	Field Design	10
9	Executive Dashboard	11-12
10	Managerial Dashboard	13-16
11	Analyst Dashboard	16-19
12	Key Learnings	20
13	Realistic Use Case	20
14	Benefits	20
15	DAX Measures used	21-22
16	Future Enhancements	22
17	Conclusion	22



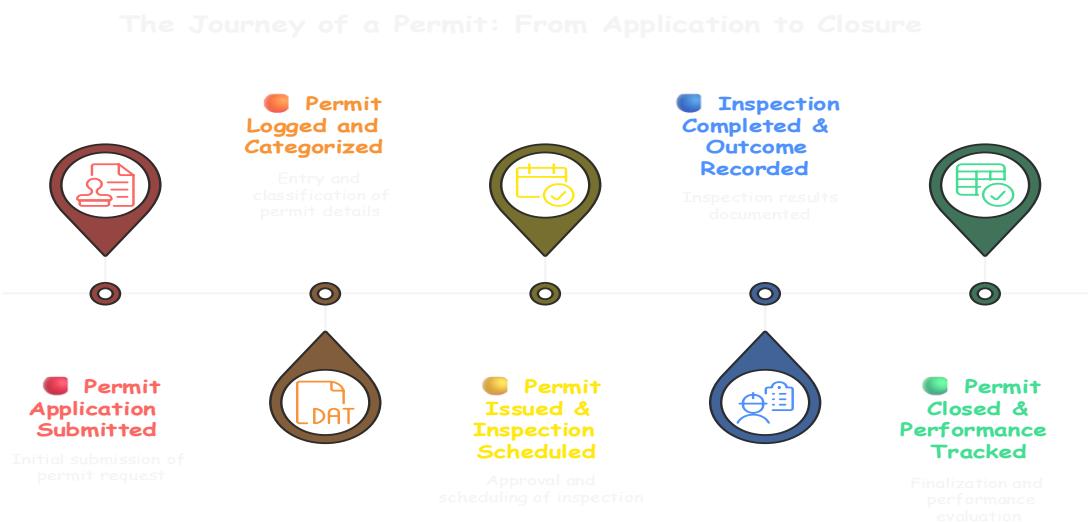
1. Introduction

This project explores a gas permits and inspections dataset and builds three Power BI dashboards customized for three roles: **Executive**, **Manager**, and **Analyst**. Each dashboard provides relevant visualizations and metrics for its target audience. The goal is to enable data-driven decisions through intuitive visual storytelling.

Source: Calgary Open Data Portal

2. Project Overview

- Focused on analyzing gas permits and inspections data from the City of Calgary.
- Created three dashboards: Executive, Managerial, and Analyst, each for specific user roles.
- Purpose is to track trends, evaluate performance, and identify operational issues.



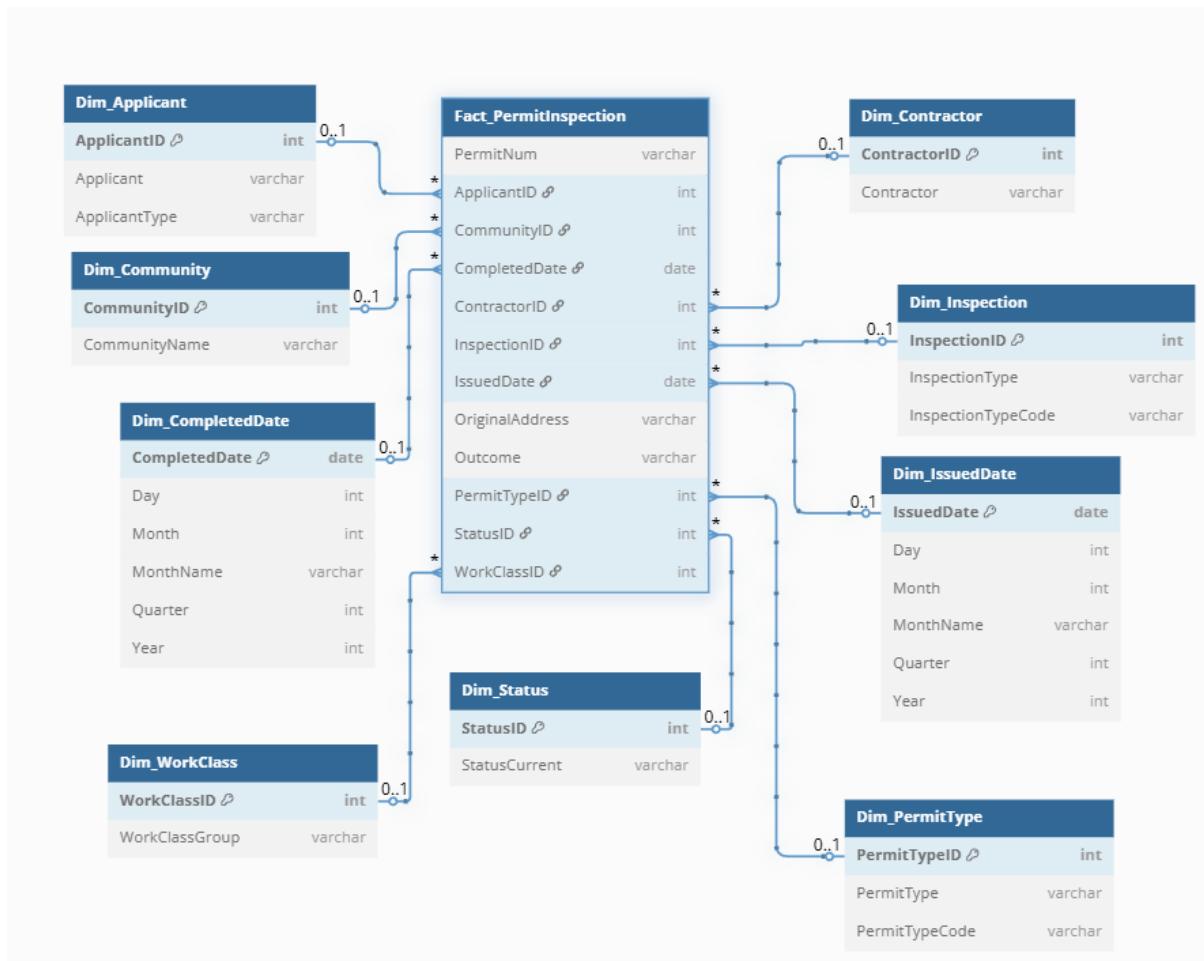
3. Purpose of the System

- Improve permit and inspection efficiency.
- Ensure transparency and public safety.
- Provide decision-makers with actionable insights.
- Detect bottlenecks or delays in permit processing.



4. Database Design

A relational database structure was used for this system. Tables are logically connected using primary and foreign keys to maintain data consistency and integrity.





5. Key Tables and Their Roles

I. Applicant

Field Name	Data Type	Description
ApplicantID	INT (PK)	Unique Identifier for applicant
Applicant	VARCHAR(150)	Name or label of the applicant
ApplicantType	VARCHAR(50)	Type/Category (e.g., Contractor, Individual)

II. Community

Field Name	Data Type	Description
CommunityID	INT (PK)	Unique identifier for each community
CommunityName	VARCHAR(50)	Name of the community/region

III. Date

Field Name	Data Type	Description
DatID	INT (PK)	Different Identifier for every date
CompletedDate	DATE	Full completed date
Day	INT	Day portion of the date
Month	INT	Month as a number (1-12)
MonthName	VARCHAR(20)	Full month name
Quarter	INT	Quarter of the year (1-4)
Year	INT	Year portion (e.g., 2023,2024)

IV. Contractor

Field Name	Data Type	Description
ContractorID	INT (PK)	Unique identifier for contractor.
Contractor Name	VARCHAR(100)	Name of the contractor.



V. Inspection

Field Name	Data Type	Description
InspectionID	INT (PK)	Unique identifier for inspection
InspectionType	VARCHAR(30)	Type of inspection (e.g., Final, Rough)
InspectionTypeCode	VARCHAR	Abbreviation or code for inspection

VI. PermitType

Field Name	Data Type	Description
PermitTypeID	INT (PK)	Unique identifier for type of permit
PermitType	VARCHAR(30)	Name of the permit type
PermitTypeCode	VARCHAR(20)	Abbreviated permit code

VII. Status

Field Name	Data Type	Description
StatusID	INT (PK, Auto)	Unique identifier for status
StatusCurrent	VARCHAR(50)	Label for current permit status (e.g., Completed, On Hold).

VIII. WorkClass

Field Name	Data Type	Description
WorkClassID	INT (PK)	Unique Identifier for work class
WorkClassGroup	VARCHAR(50)	Category-group of the work class.



IX. PermitInspection

Field Name	Data Type	Description
PermitNum	VARCHAR	Unique identifier for each permit
ApplicantID	INT (FK)	Links to Dim_Applicant, represents who applied.
CommunityID	INT (FK)	Links to Dim_Community, location info.
CompletedDate	DATE	The date the permit process was completed.
ContractorID	INT (FK)	Links to Dim_Contractor, who carried out the work.
InspectionID	INT (FK)	Links to Dim_Inspection, type of inspection performed.
IssuedDate	DATE	When the permit was officially issued.
OriginalAddress	VARCHAR	Address where the permit applies.
Outcome	VARCHAR	Result of the inspection or permit.
PermitTypeID	INT	Links to Dim_PermitType, category of permit.
StatusID	INT	Links to Dim_Status, shows the permit's current status.
WorkClassID	INT	Links to Dim_WorkClass, categorizes type of work.

6. Relationships between tables

- **Dim_Applicant → Fact_PermitInspection**
One applicant can be associated with many permits and inspections.
- **Dim_Community → Fact_PermitInspection**
One community can have multiple permit and inspection records.
- **Dim_CompletedDate → Fact_PermitInspection**
One completed date can be linked to many permit records.
- **Dim_Contractor → Fact_PermitInspection**
One contractor may handle multiple permits and inspections.
- **Dim_Inspection → Fact_PermitInspection**
One inspection type can relate to many permit inspection records.
- **Dim_IssuedDate → Fact_PermitInspection**
One issued date can correspond to many permit records.
- **Dim_PermitType → Fact_PermitInspection**
One permit type can be associated with multiple permit inspection entries.



- **Dim_Status → Fact_PermitInspection**
One status value can apply to many permits.
 - **Dim_WorkClass → Fact_PermitInspection**
One work class can categorize multiple permits.
-

7. Data Integrity Rules

1. Primary Key (PK) Assignments

Each dimension table has a unique identifier:

- Dim_Applicant → ApplicantID
- Dim_Community → CommunityID
- Dim_Contractor → ContractorID
- Dim_IssuedDate → IssuedDate
- Dim_CompletedDate → CompletedDate
- Dim_WorkClass → WorkClassID
- Dim_Status → StatusID
- Dim_Inspection → InspectionID
- Dim_PermitType → PermitTypeID

These fields act as primary keys, ensuring every record in these tables is uniquely identified.

2. Foreign Key (FK) Constraints in Fact Table

Fact_PermitInspection table includes foreign keys pointing to each dimension:

- ApplicantID → Dim_Applicant
- CommunityID → Dim_Community
- ContractorID → Dim_Contractor
- InspectionID → Dim_Inspection
- IssuedDate → Dim_IssuedDate
- CompletedDate → Dim_CompletedDate
- WorkClassID → Dim_WorkClass
- StatusID → Dim_Status
- PermitTypeID → Dim_PermitType



This maintains referential integrity and links each fact record to detailed dimensional attributes.

3. Data Type Rules

To ensure consistency and validation:

- All ID fields are integers (e.g., ApplicantID, WorkClassID).
- Date fields (IssuedDate, CompletedDate) are of date type.
- Text descriptors (e.g., Applicant, CommunityName, PermitType, InspectionType, Outcome, OriginalAddress) are varchar.
- Metrics (like PermitNum) are numeric or varchar based on format used.

4. Relationship Types

- **One-to-Many:** All dimension tables relate 1-to-many with the Fact_PermitInspection table. One record in dimension → many in fact.
- **Direction:** Relationships flow from dimensions to fact, supporting filtering and slicers.
- **Active Relationships:** Only one active relationship per field (no ambiguity in time/date joins).

5. Surrogate Key Usage

All dimensions use surrogate keys (e.g., PermitTypeID) that are independent of business logic, simplifying joins and improving performance.

6. Field Constraints

- PermitNum, IssuedDate, and CompletedDate should not be null.
- ApplicantID, ContractorID, etc., must match existing dimension records.
- Each combination of PermitNum and IssuedDate should be unique to ensure one record per permit issued.

7. Naming Conventions

- Prefixes used (Dim_-, Fact_) distinguish dimension from fact tables.
- Suffixes like ID clearly indicate primary/foreign keys.

8. Normalization and Star Schema



- Dimensions hold descriptive data (who, what, where, when).
- Fact table holds transactional data (permits, outcomes, dates).
- Ensures minimal redundancy and efficient slicing/dicing of data.

9. Column Consistency

- Related fields (**e.g., Month, Year**) across Dim_IssuedDate and Dim_CompletedDate follow the same structure.
- All dimensional values (**e.g., StatusCurrent, WorkClassGroup**) are stored consistently without duplicates.

10. Data Validation Logic

- For each permit entry:
 - Must have a valid Applicant, Contractor, Community, and Inspection.
 - Must include at least an IssuedDate (and preferably a CompletedDate).
- Outcome and StatusCurrent values should match allowed business domain values (e.g., “Completed”, “Hold”).

8. Field Design

Fields were selected based on their real-world application in Gas and Permit Inspection. Final fields were chosen after eliminating unrelated or redundant fields.

PermitNum, WorkClassGroup, StatusCurrent, OriginalAddress, IssuedDate, CompletedDate, InspectionType, Outcome, ID, Applicant, ApplicantType, CommunityName, Contractor, InspectionTypeCode, PermitType, PermitTypeCode, ProcessID

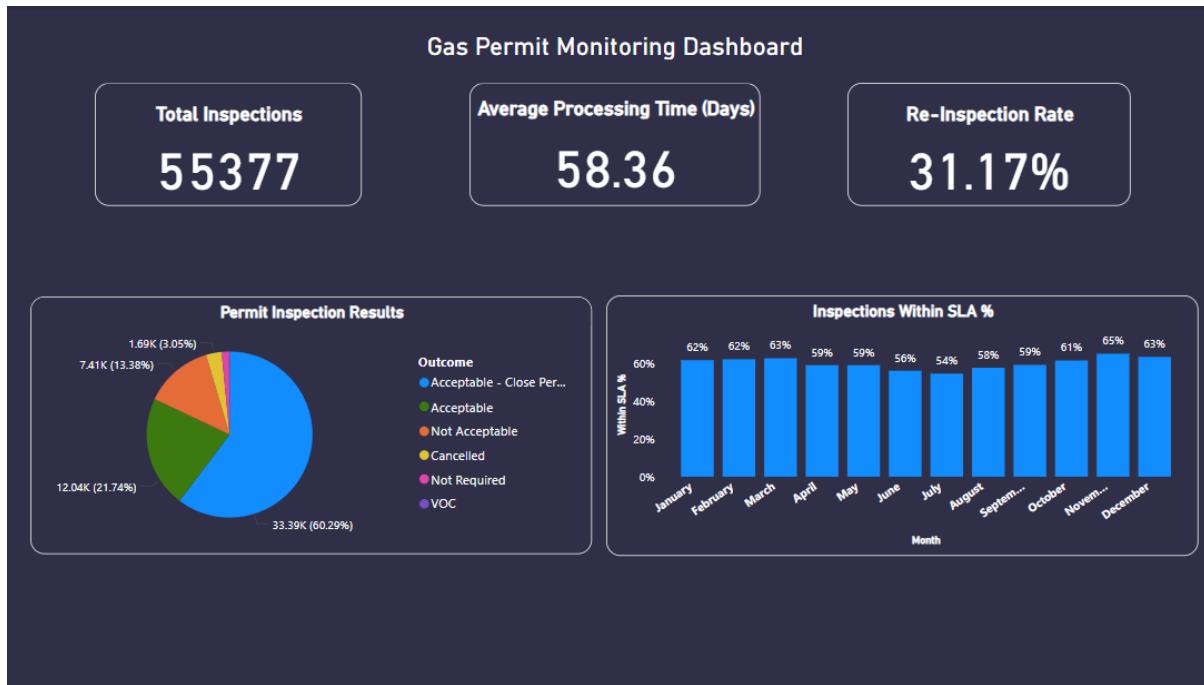
Final Field List (after normalization):

- Created surrogate IDs for dimensions.
- Split date fields into separate tables.
- Removed or merged redundant fields like PermitNum and ProcessID.



9. Executive Dashboard

Purpose: Provide top-level summary for leadership.



Visuals and Explanation:

A. KPI Cards



- Inspection Pass Rate:** Percentage of inspections marked as "Passed."
- Average Processing Time:** Average days between permit issuance and completion.
- Re-inspection Rate:** Percentage of inspections that needed to be repeated.

These cards give executives an instant pulse of operational efficiency and compliance.

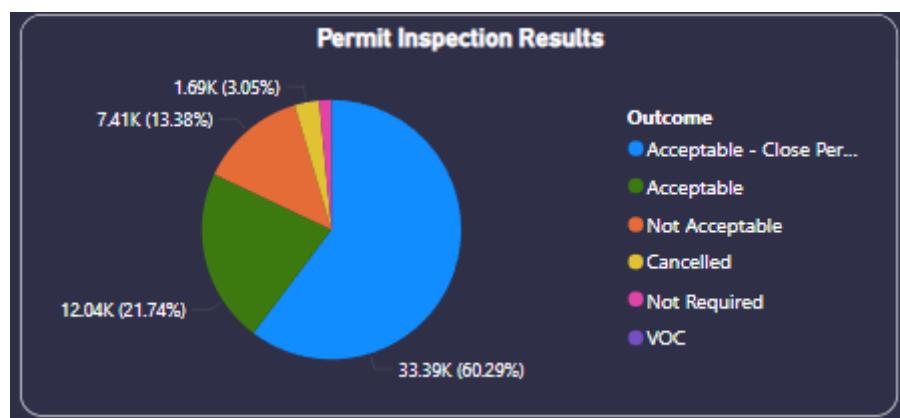


B. Monthly Inspection Trend (Line Chart) (Within SLA)



- **X-Axis:** Month from CompletedDate
- **Y-Axis:** Count of Inspections
- Purpose: Shows overall volume and seasonal patterns in inspections.

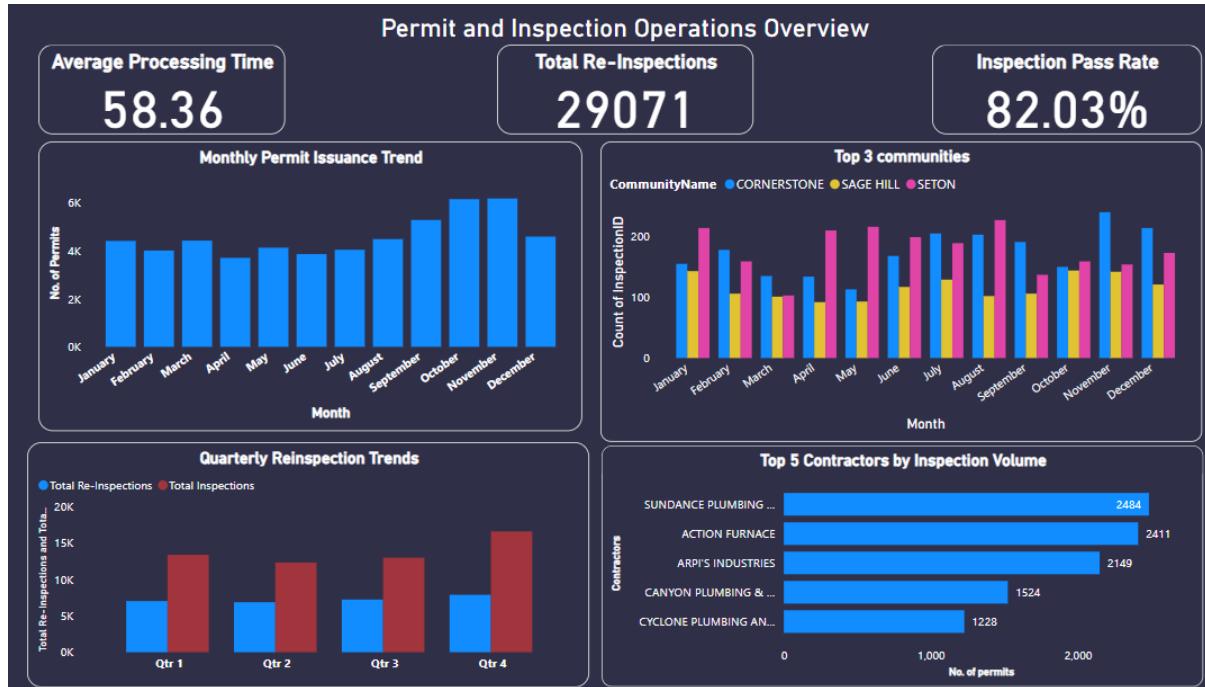
C. Permits by Outcome (Donut Chart)



- Slices: Passed, Failed, Cancelled, etc.
- Helps identify quality and effectiveness at a glance.



10. Managerial Dashboard



Purpose: Help managers monitor workload and optimize processes.

Visuals and Explanation:

A. Monthly Permit Issuance Trend (Column Chart)



- **X-Axis:** Month
- **Y-Axis:** No. of Permits



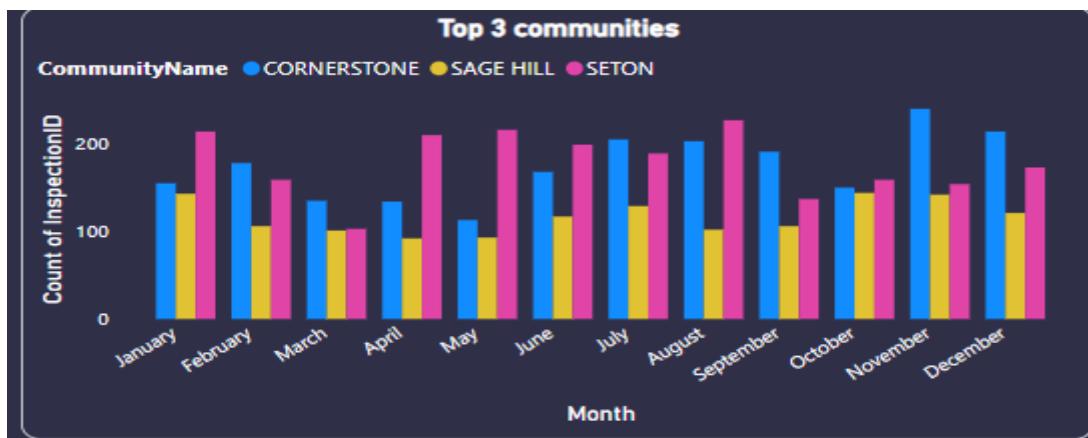
- **Explanation:**

This column chart tracks how many gas permits were issued month by month throughout the year.

It helps identify seasonal trends or peaks in permitting activity.

For example, there's a noticeable rise in permit issuance from September to November, likely due to seasonal work schedules or construction activity.

B. Top 3 Communities by Inspection (Clustered Column Chart)



- **X-Axis:** Month

- **Y-Axis:** Count of InspectionID

- **Legend:** Community Name (Cornerstone, Sage Hill, Seton)

- **Explanation:**

This chart compares the number of inspections conducted across the top 3 communities for each month.

It helps detect which communities have higher inspection demand over time, enabling better allocation of inspectors or support teams.



C. Quarterly Re-Inspection Trends (Clustered Column Chart)



- X-Axis:** Quarter (Q1 to Q4)
- Y-Axis:** Number of Inspections
- Legend:** Total Re-Inspections and Total Inspections
- Explanation:**
This bar chart compares the total number of inspections versus re-inspections in each quarter.
A higher re-inspection rate may indicate quality control issues, contractor performance problems, or complex permit types needing more follow-up.

D. Top 5 Contractors by Inspection Volume (Horizontal Bar Chart)



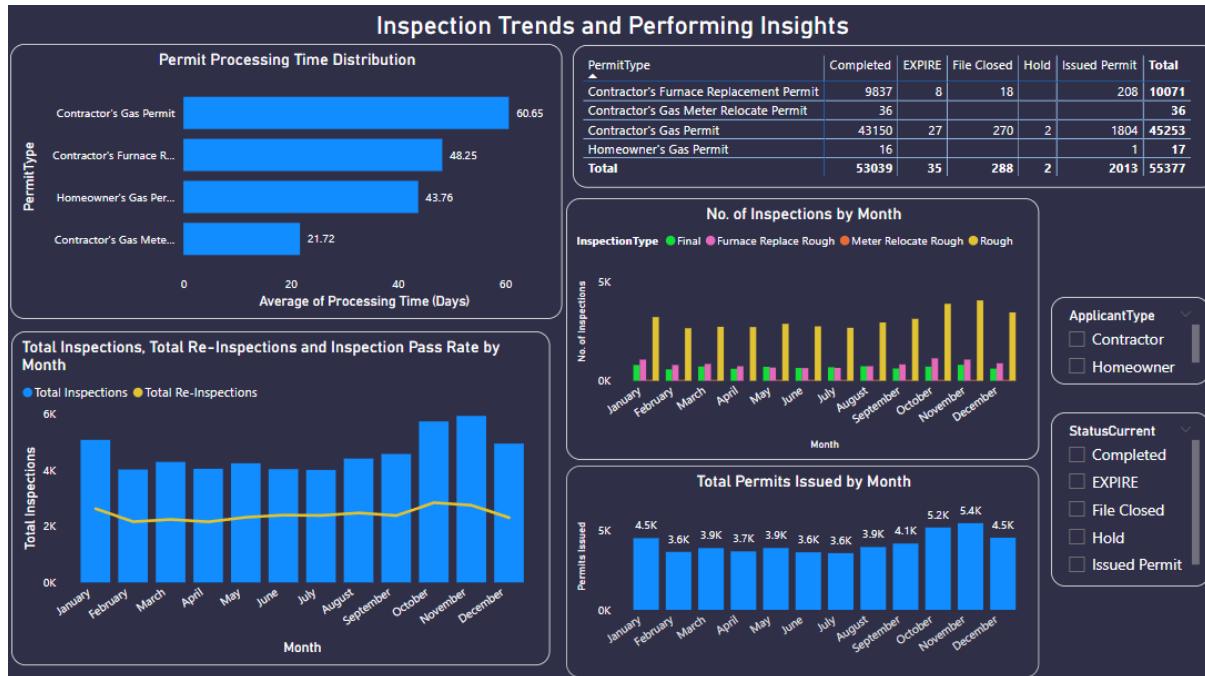
- X-Axis:** No. of Permits
- Y-Axis:** Contractor Names



- **Explanation:**

This visual lists the top 5 contractors based on the number of inspections they handled. It helps identify which contractors are the most active, which can guide performance reviews, audits, or even targeted communications for training and updates.

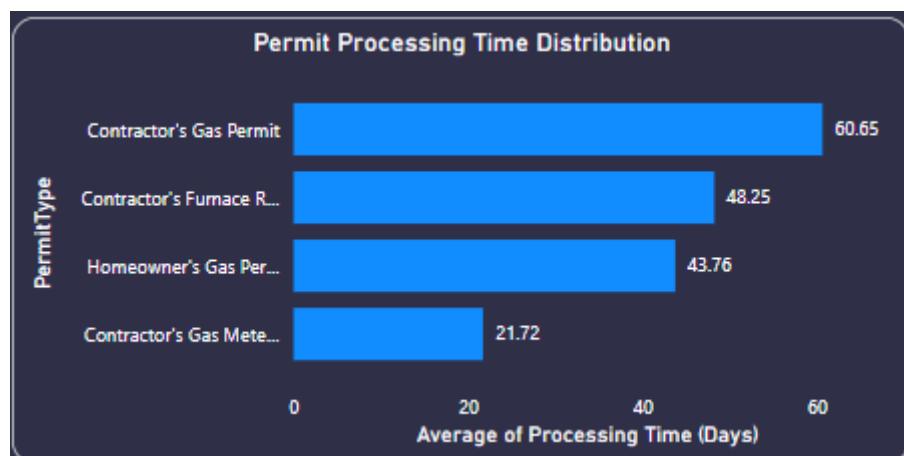
11. Analyst Dashboard



Purpose: Provide deep insights and pattern recognition for analysts.

Visuals and Explanation:

A. Permit Processing Time Distribution (Horizontal Bar Chart)





- X-Axis: **Average Processing Time (Days)**

- Y-Axis: **Permit Type**

- **Explanation:**

This bar chart shows the average number of days it takes to process different types of permits.

It clearly reveals that Contractor's Gas Permit has the highest average processing time (60.65 days), while Contractor's Gas Meter Relocate Permit is processed much faster (21.72 days).

Useful for identifying bottlenecks in the permitting workflow.

B. Total Inspections, Total Re-Inspections, and Inspection Pass Rate by Month (Combo Chart)



- X-Axis: **Month**

- Y-Axis (Left): **Total Inspections (Bar)**

- Y-Axis (Right): **Re-Inspections (Bar) + Inspection Pass Rate (Line)**

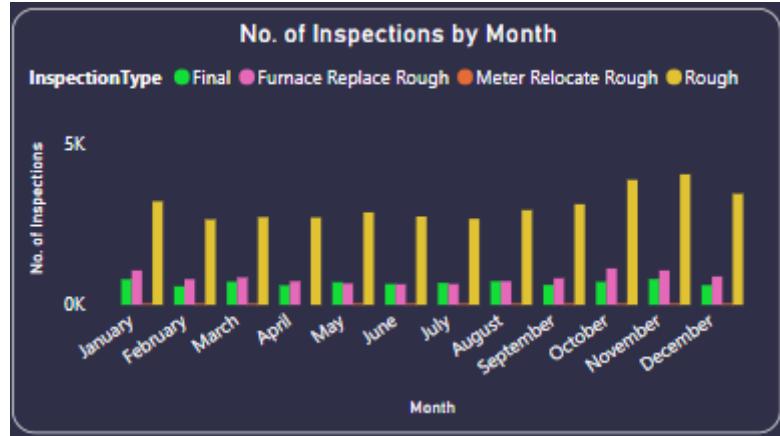
- **Explanation:**

This chart combines bar and line visuals to compare the total number of inspections, the number of re-inspections, and the inspection pass rate.

This view is helpful for detecting quality trends over time. A rising re-inspection rate might suggest a dip in first-pass success.

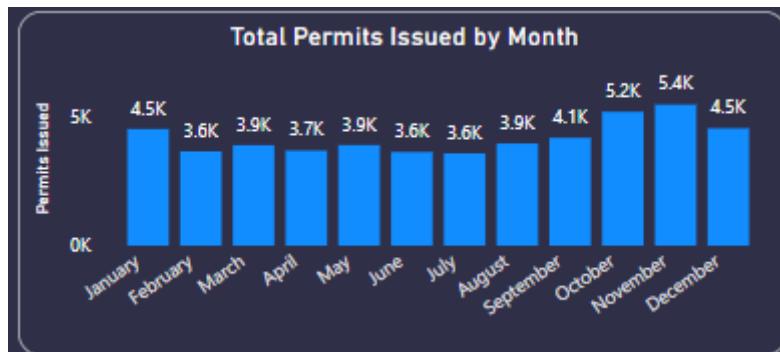


C. No. of Inspections by Month (Clustered Column Chart)



- **X-Axis: Month**
- **Y-Axis: Count of Inspections**
- **Legend: Inspection Types (Final, Rough, Meter Relocate Rough, etc.)**
- **Explanation:**
This clustered column chart displays how many inspections were carried out each month, broken down by inspection type.
It helps to understand which inspection types are most common and how they fluctuate throughout the year.

D. Total Permits Issued by Month (Column Chart)



- **X-Axis: Month**
- **Y-Axis: Number of Permits Issued**
- **Explanation:**
This simple bar chart shows how many permits were issued each month.
For example, issuance peaks in October and November, possibly due to seasonal demand.



E. Permit Status Overview Table (Matrix Visual)

PermitType	Completed	EXPIRE	File Closed	Hold	Issued Permit	Total
Contractor's Furnace Replacement Permit	9837	8	18		208	10071
Contractor's Gas Meter Relocate Permit	36					36
Contractor's Gas Permit	43150	27	270	2	1804	45253
Homeowner's Gas Permit	16				1	17
Total	53039	35	288	2	2013	55377

- Rows:** Permit Type
- Columns:** Statuses (Completed, Expired, File Closed, Hold, Issued, Total)
- Explanation:**
The matrix provides a detailed count of permits under various statuses across permit types.
It serves as a quick summary to identify trends and backlogs (e.g., large numbers of permits completed vs. on hold).

◆ Slicers Used in Dashboard

- Applicant Type**
 - Allows filtering visuals by Contractor or Homeowner applicants.

ApplicantType
<input type="checkbox"/> Contractor
<input type="checkbox"/> Homeowner

StatusCurrent
<input type="checkbox"/> Completed
<input type="checkbox"/> EXPIRE
<input type="checkbox"/> File Closed
<input type="checkbox"/> Hold
<input type="checkbox"/> Issued Permit

- Status Current**
 - Lets users filter the entire dashboard based on permit status such as:
 - Completed
 - EXPIRE
 - File Closed
 - Hold
 - Issued Permit



12. Key Learnings

- Slicers must be relevant and not too overwhelming.
 - Avoid overcrowding charts with high-cardinality fields.
 - Title clarity, axis labels, and tooltips improve usability.
 - Analyst dashboards should avoid KPI cards and focus on detailed patterns.
-

13. Realistic Use Case

Scenario

The City of Calgary's Building Services Department is responsible for issuing gas permits, tracking inspection outcomes, and ensuring contractors comply with safety and construction standards. The department handles thousands of permits and inspection records every year across different communities, contractors, and permit types.

Problem

- **Inefficiency in Permit Processing:** Permits are delayed due to bottlenecks in inspections and documentation.
 - **No Central Monitoring System:** Managers and executives lack a **central dashboard** to oversee operations or identify processing delays.
 - **Low Transparency:** Applicants and contractors don't receive timely updates on permit statuses.
 - **No Trend Analysis:** Historical data is not analyzed effectively to predict workloads or recurring issues.
-

14. Benefits

- **Operational Efficiency:** Faster decisions using real-time data.
 - **Accountability:** Clear tracking of contractor and inspector performance.
 - **Planning & Forecasting:** Better understanding of seasonal trends in permit requests.
 - **Transparency:** Easy reporting for compliance audits and public disclosures.
-



15. DAX Measures Used

Average Processing Time =

```
AVERAGEX(Fact_Permits, DATEDIFF(Fact_Permits[IssuedDate],  
Fact_Permits[CompletedDate], DAY))
```

Inspection Pass Rate =

```
DIVIDE(  
    CALCULATE(COUNTROWS(Fact_Permits), Fact_Permits[Outcome] = "Passed"),  
    COUNTROWS(Fact_Permits))
```

SLA Compliance =

```
DIVIDE(  
    CALCULATE(COUNTROWS(Fact_Permits), Fact_Permits[ProcessingDays] <= SLA_Target),  
    COUNTROWS(Fact_Permits))
```

Re-inspection Rate =

```
DIVIDE(  
    CALCULATE(COUNTROWS(Fact_Permits), Fact_Permits[Outcome] = "Re-Inspection"),  
    COUNTROWS(Fact_Permits))
```

YoY Growth =

```
VAR CurrentYear = YEAR(TODAY())  
VAR PreviousYear = CurrentYear - 1  
RETURN  
DIVIDE(  
    CALCULATE(COUNTROWS(Fact_Permits), YEAR(Fact_Permits[CompletedDate]) =  
    CurrentYear) -  
    CALCULATE(COUNTROWS(Fact_Permits), YEAR(Fact_Permits[CompletedDate]) =  
    PreviousYear),
```



CALCULATE(COUNTROWS(Fact_Permits), YEAR(Fact_Permits[CompletedDate]) = PreviousYear))

16. Future Enhancements

- Add automated notifications.
 - Develop a front-end UI for non-technical users.
 - Enable mobile access for technicians.
-

17. Conclusion

This Hotel Maintenance Management System serves as a practical, scalable, and easy-to-use solution for managing guest complaints and room issues. It reflects real industry workflows and enforces data quality throughout.
