Car Insurance Claim Dashboard – Step-by-Step Guide

Comprehensive, ordered instructions to build, document and deploy the Power BI project

---

1. Project Overview

This guide walks through building the Car Insurance Claim Dashboard in Power BI. Follow each numbered step in order. Each step is standalone and contains exact actions to take.

2. Prerequisites

1. Power BI Desktop (latest stable version)
2. Data files (Excel or CSV) or access to SQL Server / Database
3. Basic familiarity with Power Query, DAX, and data modeling concepts
4. Screenshots of the dashboard for reference (optional)

---

3. Data Sources & File Layout

3.1 Collect raw data files: Policies.xlsx, Claims.csv, Vehicles.csv, Customers.csv (or equivalent tables in DB).

3.2 Inspect columns and sample rows to confirm types (Date, Numeric, Text).

3.3 Standardize column names (e.g., PolicyID, ClaimID, CustomerID, VehicleType, VehicleYear, Mileage, ClaimOutcome).

---

4. ETL with Power Query (Step-by-step)

4.1 Open Power BI → Get Data → Choose Excel/CSV/SQL and load your tables (Policies, Claims, Vehicles, Customers).

4.2 In Power Query, for each table: remove unnecessary columns, fix data types, trim whitespace, and remove duplicate rows.

4.3 Create consistent keys: ensure PolicyID, ClaimID, CustomerID are the same format (text or number) across tables.

4.4 Use 'Merge Queries' to join Claims with Customers (on CustomerID) and Vehicles (on VehicleID) where needed. Prefer to keep Claims as the fact table.

4.5 Add computed columns if needed (e.g., ClaimYear = Date.Year([ClaimDate]), MileageBin = Number.RoundDown([AnnualMileage]/1000)*1000).

4.6 Close & Apply to load transformed data to the data model.

---

5. Data Modeling (Star Schema)

5.1 Create a central Fact table: FactClaims. This should contain ClaimID, PolicyID, CustomerID, VehicleID, ClaimAmount, ClaimDate, ClaimOutcome, Mileage, RiskCategory.

5.2 Create Dimension tables: DimCustomer (AgeGroup, Gender, Education, Income), DimVehicle (VehicleType, VehicleYear), DimPolicy (PolicyType, StartDate).

5.3 Set relationships: FactClaims[CustomerID] → DimCustomer[CustomerID], FactClaims[VehicleID] → DimVehicle[VehicleID], FactClaims[PolicyID] → DimPolicy[PolicyID].

6. DAX Measures (KPIs & Calculations)

6.1 Total Policies = DISTINCTCOUNT(DimPolicy[PolicyID])

6.2 Total Claims = COUNTROWS(FactClaims)

6.3 Claim Rate = DIVIDE([Total Claims], [Total Policies], 0)

6.4 High Risk Drivers % = DIVIDE(CALCULATE(COUNTROWS(FactClaims), FactClaims[RiskCategory] = "High"), [Total Drivers], 0)

6.5 Avg Annual Mileage = AVERAGE(FactClaims[AnnualMileage])

Note: Replace column/table names to match your model.

---

7. Visual Building – Overview Page

7.1 Top KPI cards: Create cards for Total Policies, Total Claims, Claim Rate, High-Risk %, Avg Annual Mileage. Use consistent fonts and format (thousands separators).

7.2 Demographic visuals: Donut chart for Gender split, Bar charts for Age Group and Education level. Show exact values as data labels.

7.3 Vehicle insights: Horizontal bar charts for Claims by Vehicle Type and by Vehicle Year.

7.4 Mileage Histogram: Use binned mileage field or histogram visual; show bins such as 0-5k, 5-10k, 10-15k, 15-20k, 20k+.

7.5 Filters & Slicers: Add slicers for Age Group, Gender, Income, and Vehicle Year. Set a synced slicer for all pages if required.

7.6 Design polish: Use consistent color palette (e.g., dark blue + yellow accent), rounded cards & shadows, and spacing between elements.

---

8. Claims Detail Page (Drillthrough)

8.1 Create a new page named 'Claim Details'. Add a table visual and include required fields: ID, Age, Race, Gender, Education, Income, Risk Category, Claim Outcome, Vehicle Type, Vehicle Year.

8.2 Enable drillthrough: Add ClaimID or CustomerID as drillthrough field so users can right-click on visuals from the overview to navigate here.

8.3 Make columns sortable and apply alternate row shading for readability.

8.4 Add a 'Back to Overview' button with Action → Page Navigation to return to the Overview page.

---

9. Testing & Validation

9.1 Validate row counts and aggregates against source files (Total Policies, Total Claims).

9.2 Cross-check Claim Rate calculations against manual calculations in Excel for a sample of records.

9.3 Validate filters & slicers to ensure they apply correctly across pages.

9.4 Test drillthrough and bookmarks (if used) to ensure navigation works as expected.

---

10. Deployment & Sharing

10.1 Save the .pbix file and, if required, publish to Power BI Service (Publish → Select workspace).

10.2 Configure dataset credentials in the service if using live or direct query to databases.

10.3 Set up scheduled refresh in Power BI Service: go to dataset settings → refresh schedule (choose frequency & time).

10.4 Share report: Provide direct link or add report to a workspace app for organizational access.

---

11. Best Practices & Tips

11.1 Keep a consistent and limited color palette to improve readability.

11.2 Document DAX measures in a separate markdown or text file for maintainability.

11.3 Use bookmarks for guided storytelling or to store filter states.

11.4 Avoid heavy visuals that slow rendering; prefer aggregated views with drillthrough for details.

11.5 Add descriptive titles and tooltips to help end-users interpret visuals.

12. Appendix - Example DAX Snippets

Measure	DAX
Total Policies	DISTINCTCOUNT(DimPolicy[PolicyID])
Total Claims	COUNTROWS(FactClaims)
Claim Rate	DIVIDE([Total Claims], [Total Policies], 0)
Avg Annual Mileage	AVERAGE(FactClaims[AnnualMileage])

Author: Loganathan • Contact: loganathanvizasia@gmail.com