**Column-by-column explanation (with data type & typical values)**

**A) Organization & meta**

* **Questionnaire** *(nvarchar)*  
  CDP questionnaire edition for the row (e.g., Cities 2023).  
  *Use:* context/versioning.
* **Organization\_Number** *(int)*  
  CDP’s numeric ID for the reporting city (e.g., 31153).  
  *Use:* joins/unique ID.
* **Organization\_Name** *(nvarchar)*  
  Formal reporting entity name (e.g., City-State of Berlin).  
  *Use:* labeling.
* **City** *(nvarchar)*  
  City name (e.g., Berlin).  
  *Use:* filters, labeling.
* **Country** *(nvarchar)*  
  Country name (e.g., Germany).  
  *Use:* filters, labeling.
* **CDP\_Region** *(nvarchar)*  
  CDP regional grouping (e.g., Europe).  
  *Use:* regional analysis.
* **Access** *(nvarchar)*  
  Data access level (usually public).  
  *Use:* governance, rarely used in visuals.
* **Number\_of\_times\_reporting** *(int)*  
  How many years the city has reported to CDP (e.g., 12).  
  *Use:* maturity/context.
* **C40\_City** *(bit / nvarchar → imported as text in CSV)*  
  Whether the city is part of C40 (e.g., True/False).  
  *Use:* segmentation. (If text, convert to bit if needed.)
* **GCoM\_City** *(bit / nvarchar)*  
  Whether the city is part of GCoM (e.g., True/False).  
  *Use:* segmentation.

**B) Inventory configuration (applies to the entire inventory year)**

* **Primary\_protocol\_framework\_used\_to\_compile\_main\_inventory** *(nvarchar)*  
  Main protocol/standard used (e.g., GPC, IPCC, etc.).  
  *Use:* methodology notes.
* **Gases\_included\_in\_main\_inventory** *(nvarchar)*  
  List/description of gases included (e.g., CO2, CH4, N2O or a narrative).  
  *Use:* scope of gases, documentation.
* **Year\_covered\_by\_main\_inventory** *(int)*  
  Inventory year (e.g., 2019).  
  *Use:* time axis.
* **Boundary\_of\_main\_inventory\_relative\_to\_jurisdiction\_boundary** *(nvarchar)*  
  Boundary choice description (e.g., “same as jurisdiction boundary”).  
  *Use:* context.
* **Population\_in\_year\_covered\_by\_main\_inventory** *(nvarchar)*  
  Population for that year (often textified numbers).  
  *Use:* per-capita metrics (convert to numeric).
* **Tool\_used\_to\_compile\_main\_inventory** *(nvarchar)*  
  Software/tool used to compile (e.g., ClearPath, spreadsheets).  
  *Use:* documentation.
* **City\_Location** *(nvarchar)*  
  Location info; often coordinate-like text or “City, Country”.  
  *Use:* mapping (clean/parse as needed).
* **Last\_update** *(nvarchar / date-as-text)*  
  Last update timestamp as text.  
  *Use:* data currency (parse to date if needed).

**C) Emissions “table” structure (like columns/rows of a matrix)**

* **Emissions\_Question\_Number** *(nvarchar)* and **Emissions\_Question\_Name** *(nvarchar)*  
  CDP matrix position info (question metadata).  
  *Use:* tracing back to original questionnaire.
* **Emissions\_Column\_Number** *(int)* and **Emissions\_Column\_Name** *(nvarchar)*  
  The **column** within the CDP emissions table. The **name** encodes *scope-like meaning*:
  + Direct emissions (metric tonnes CO2e)^ → **Scope 1**
  + Indirect emissions from the use of grid-supplied electricity, heat, steam and/or cooling (metric tonnes CO2e)^ → **Scope 2**
  + Emissions occurring outside the jurisdiction boundary as a result of in-jurisdiction activities (metric tonnes CO2e) → **Out-of-boundary** (Scope 3–like)  
    *Use:* **map to Scope** by pattern on Emissions\_Column\_Name.
* **Emissions\_Row\_Number** *(int)* and **Emissions\_Row\_Name** *(nvarchar)*  
  The **row** within the table; acts like *Sector/Subsector*. Format:  
  Sector > Subsector (e.g., Stationary energy > Residential buildings^).  
  The trailing ^ is a note marker—strip it for analytics.  
  *Use:* split into sector and subsector using ' > ' delimiter.

**D) Emissions values & quality flags**

* **Emissions\_Response\_Answer** *(nvarchar → numeric-as-text)*  
  The **numeric emissions value** in **metric tonnes CO2e**, but stored as text with thousands separators (e.g., 536,000).  
  *Use:* convert via TRY\_CONVERT(decimal(18,2), REPLACE([Emissions\_Response\_Answer], ',', '')).
* **Emissions\_Notation\_Key** *(nvarchar)*  
  Data flags similar to IPCC/GPC notation:
  + IE = Included Elsewhere (value may be 0 here; counted in another row).
  + NO = Not Occurring.
  + NE = Not Estimated.  
    (Blank = reported numeric.)  
    *Use:* **filter logic**—commonly exclude NO and NE; keep IE depending on your summing logic to avoid double counting.
* **Emissions\_Description** *(nvarchar)*  
  Free-text description of the row’s source/process (e.g., Industrial process - Direct).  
  *Use:* tooltips, documentation.
* **Emissions\_Estimated** *(int / nvarchar)*  
  Binary-like indicator if the value is estimated (often 1/0).  
  *Use:* quality indicator; segment visuals (Estimated vs Reported).
* **Emissions\_Data\_Group** *(nvarchar)*  
  Line **type** within the emissions table. Typical values:
  + SubSector, GridSubSector → **granular** rows (use in the fact view)
  + SectorTotal, SourceTotal, GridSectorTotal → **totals** rows (exclude from granular fact; useful for cross-checks)  
    *Use:* filtering for granular vs total lines.
* **Emissions\_Rank** *(int)*  
  Display order ranking from the source table.  
  *Use:* optional for ordering rows in detailed tables.

**E) Housekeeping you added**

* **row\_id** *(int, IDENTITY, PK)*  
  Your technical surrogate key in stg.cdp\_city\_emissions\_raw.  
  *Use:* row-level traceability.

**Practical modeling notes (how to use them)**

* **Scope mapping:**  
  Build CASE from Emissions\_Column\_Name → Scope:
  + LIKE 'Direct emissions (%' → Scope 1
  + LIKE 'Indirect emissions from the use of grid-supplied%' → Scope 2
  + LIKE 'Emissions occurring outside the jurisdiction boundary%' → Out-of-boundary
* **Sector/Subsector:**  
  Split Emissions\_Row\_Name on ' > ':
  + sector = LEFT(…, CHARINDEX(' > ', …)-1)
  + subsector = SUBSTRING(… after ' > ')  
    Then **strip trailing carets ^** from both.
* **Numeric conversion:**  
  TRY\_CONVERT(decimal(18,2), REPLACE([Emissions\_Response\_Answer], ',', '')) AS emissions\_tCO2e
* **Filter for “fact” rows:**  
  Emissions\_Data\_Group IN ('SubSector','GridSubSector')  
  and **exclude** rows where Emissions\_Notation\_Key IN ('NO','NE').
* **Totals logic:**  
  Aggregate the fact view by year + scope, then UNION an “All scopes” row for convenience.

What is GCoM? What are the protocols mentioned in the data set (e.g., GPC, IPCC, etc.), and what do they mean? What is CIRIS in Tool\_used\_to\_compile\_main\_inventory? What is meant by 'Provide a breakdown of your community-wide emissions in the format of the Common Reporting Framework.' in Emissions\_Question\_Name, and what is Emissions\_Question\_Number? What is the binary ( 1 and 0) in Emissions\_Estimated, and what does it mean ? What are SubSector, GridSubSector, SectorTotal, SourceTotal, GridSectorTotal in Emissions\_Data\_Group, and what do they mean? And what is Emissions\_Rank, which ranks high ( 0 is lowest and going up increases the rank)?

**GCoM**

**Global Covenant of Mayors for Climate & Energy.**  
A large international alliance of cities committing to measure, report, and act on climate and energy. If GCoM\_City = True, Berlin participates and aligns with GCoM’s MRV (measurement-reporting-verification) expectations.

**Protocols you’ll see (e.g., GPC, IPCC)**

* **GPC** — *Global Protocol for Community-Scale GHG Inventories* (WRI/C40/ICLEI).  
  The de-facto standard for city inventories: defines **sectors/subsectors**, **Scopes 1–3**, boundary rules, and reporting options (**BASIC/BASIC+**). Your sector/subsector lines and scope mapping in the dataset come straight from GPC.
* **IPCC Guidelines** — Methods/category logic from the *2006 IPCC Guidelines* (with refinements).  
  Cities often use IPCC emission factors/methods under the GPC umbrella.
* (You may also encounter in other cities’ files) **ISO 14064-1** (org-level GHG accounting), **PAS 2070** (city-level assessment), or “Other (please specify)”. In your file the “Primary\_protocol\_framework…” column states which one Berlin used.

**CIRIS (in Tool\_used\_to\_compile\_main\_inventory)**

**City Inventory Reporting and Information System.**  
A GPC-aligned Excel workbook/tool (widely used by cities) to structure activity data, apply factors, and produce a GPC-compliant inventory. Seeing CIRIS there just means Berlin built the inventory with that template/tool.

**“Provide a breakdown … in the format of the Common Reporting Framework.”**

This is the **question text** (you’ll see it in Emissions\_Question\_Name) asking the city to submit emissions **by sector/subsector and scope** using the **Common Reporting Framework (CRF)** of the CDP-ICLEI questionnaire.  
Practically: that’s why your rows look like Sector > Subsector and your columns look like “Direct / Indirect / Out-of-boundary”. It’s the tabular GPC layout CDP expects.

**Emissions\_Question\_Number**

The internal **question ID/code** for that CRF table in the CDP questionnaire (e.g., “3.1”, “3.1a”). It’s a reference key, not an amount.

**Emissions\_Estimated (binary 1/0)**

* **1** = the reported value is **estimated** (e.g., modeled, proxy factors).
* **0** = **not estimated** (more directly measured or well-specified calculation).  
  Use it to flag data quality or segment visuals (e.g., show share of estimated vs. measured).

**Emissions\_Data\_Group (line-type)**

This tells you what each row *is* in the CDP table:

* **SubSector** — A granular GPC line inside a sector (e.g., *Stationary energy > Residential buildings*).  
  *Use these in your fact view.*
* **GridSubSector** — Granular lines specifically within **Generation of grid-supplied energy** (e.g., *Electricity-only generation*, *CHP*, *Heat/cold*).  
  *Also use in your fact view.*
* **SectorTotal** — The **sum for a sector** (e.g., total Stationary energy).  
  *Skip in fact; useful for QA cross-checks.*
* **GridSectorTotal** — The **sum for the grid-generation sector**.  
  *Skip in fact; use for QA if needed.*
* **SourceTotal** — A higher-level **roll-up total** across related sources/rows.  
  *Skip in fact; use for QA / reconciliation.*

**Rule of thumb for modeling:** keep SubSector and GridSubSector as **facts**; treat the \*Total groups as **validation totals** (don’t sum them with the granular lines or you’ll double count).

**Emissions\_Rank**

A **display/sort key** from CDP—**not** a quality score.

* **0** is the **first/lowest** rank (appears earliest).
* Larger numbers appear **later** in the table.  
  Use it only if you want to preserve CDP’s original row ordering in a table visual.

While creating v\_city\_emissions\_fact you only added rows with 'SubSector','GridSubSector' in Emissions\_Data\_Group, and Emissions\_Notation\_Key is Null or IE. (in th line of code ' WHERE s.[Emissions\_Data\_Group] IN ('SubSector','GridSubSector') -- keep only granular lines

AND (s.[Emissions\_Notation\_Key] IS NULL OR s.[Emissions\_Notation\_Key] NOT IN ('NO','NE'))' from which i understood). Why?

**1) Keep only SubSector and GridSubSector**

* Those are the **atomic (granular)** GPC lines (e.g., Stationary energy > Residential buildings).
* Rows labeled SectorTotal, SourceTotal, GridSectorTotal are **pre-summed** by CDP.  
  If we mix totals with granular rows, you’ll **double count** in visuals.
* In Power BI we can always **recalculate totals** from granular facts (and it’s safer).

**TL;DR:** Facts = only the lowest level; totals are derived in BI (or kept in a separate QA view).

**2) Exclude NE and NO (notation key)**

* NE = **Not Estimated** → no reliable number to aggregate.
* NO = **Not Occurring** → should not contribute to totals.  
  Keeping them in the numeric fact table creates noise or misleading zeros. You can still analyze completeness via a **separate QA table/view** if you want.

**3) Why allow NULL and IE to pass the filter?**

* We **didn’t explicitly filter out IE** (Included Elsewhere) because, under GPC practice, IE rows **usually carry 0 or blank** values—their emissions are **booked in another row** to avoid double counting.
* Our numeric column is built with:
* TRY\_CONVERT(decimal(18,2), REPLACE([Emissions\_Response\_Answer], ',', ''))

so blanks stay **NULL**, and zeros remain **0**—both are harmless in sums.

* If a dataset **ever** puts a positive number on an IE line (rare, but it happens), including it would risk double counting. Two safe options if you want to be stricter:
  + **Option A (exclude IE entirely):**
  + AND (s.[Emissions\_Notation\_Key] IS NULL)
  + **Option B (force IE to zero at source):**
  + CASE WHEN s.[Emissions\_Notation\_Key] = 'IE' THEN 0
  + ELSE TRY\_CONVERT(decimal(18,2), REPLACE(s.[Emissions\_Response\_Answer], ',', ''))
  + END AS emissions\_tCO2e

I generally prefer **A** for the fact view and keep IE counts in a QA view.

**Recommended pattern (clean + QA)**

* **v\_city\_emissions\_fact** → granular only (SubSector,GridSubSector), **exclude NE, NO**, and (optionally) **exclude IE** to be extra safe.
* **v\_city\_emissions\_totals** → totals **rederived** from the fact.
* **v\_city\_emissions\_qa** (optional) → includes SectorTotal/SourceTotal/GridSectorTotal and counts of NE/NO/IE so you can reconcile and show data completeness.

If you want, I can give you a **one-line edit** to the view to **exclude IE** now, and a tiny QA view to track how many rows were NE/NO/IE.