

```
-- #####
-- SQL CODE USED:
-- Note: Used postgresQL ver 16.2 using personal server to store schemas.
-- Code is shown in rough order with brief desc.
-- Brisbane CBD > South Brisbane > Woolloong
-- Memorial Park > Auckland Point > South Gladstone
```

Datasets (Warehouse):

```
public."Woolloongabba_Air_Quality"
public."Brisbane_CBD_Air_Quality"
```

```
-- #####
-- 01. Insert Datetime ID into composite files
-- #####
INSERT INTO public."brisbane_composite_air_quality_raw"("datetime_id")
(SELECT "datetime_id" FROM public."brisbane_cbd_air_quality_raw" );
```

```
-- #####
-- 02. Insert, non empty files
-- #####
UPDATE
  brisbane_composite_air_quality_raw
SET
  datevalue = brisbane_cbd_air_quality_raw.datevalue
FROM
  brisbane_cbd_air_quality_raw
WHERE
  brisbane_composite_air_quality_raw.datetime_id= brisbane_cbd_air_quality_raw.datetime_id;
```

```
-- #####
-- 03. Insert CBD - - > Compound dataset.
-- #####
UPDATE
  brisbane_composite_air_quality_raw
SET
  timevalue = brisbane_cbd_air_quality_raw.timevalue,
  "wind_direction_(degN)" = brisbane_cbd_air_quality_raw."wind_direction_(degN)",
  "wind_speed_(m/s)" = brisbane_cbd_air_quality_raw."wind_speed_(m/s)",
  "wind_deviation_(deg)" = brisbane_cbd_air_quality_raw."wind_deviation_(deg)",
  "wind_speed_deviation_(m/s)" = brisbane_cbd_air_quality_raw."wind_speed_deviation_(m/s)",
  "wind_temperature_(degC)" = brisbane_cbd_air_quality_raw."wind_temperature_(degC)",
  "humidity_(%)" = brisbane_cbd_air_quality_raw."humidity_(%)",
  "rainfall_(mm)" = brisbane_cbd_air_quality_raw."rainfall_(mm)",
  "barometric_pressure_(hPa)" = brisbane_cbd_air_quality_raw."barometric_pressure_(hPa)",
  "PM10_(ug/m3)" = brisbane_cbd_air_quality_raw."PM10_(ug/m3)",
  "PM2.5_(ug/m3)" = brisbane_cbd_air_quality_raw."PM2.5_(ug/m3)",
  "visibility_reducing_particles_(Mm-1)" =
  brisbane_cbd_air_quality_raw."visibility_reducing_particles_(Mm-1)"
FROM
  brisbane_cbd_air_quality_raw
WHERE
  brisbane_composite_air_quality_raw.datetime_id= brisbane_cbd_air_quality_raw.datetime_id;
```

```

-- #####
-- 04. Insert Auckland - - > Compound dataset as Auckland has more values.
-- No overlap between Memorial/Auckland => Interchangeable.
-- #####
UPDATE
  gladstone_composite_air_quality_raw
SET
  "wind_direction_(degN)" = auckland_point_air_quality_raw."wind_direction_(degN)",
  "wind_speed_(m/s)" = auckland_point_air_quality_raw."wind_speed_(m/s)",
  "wind_deviation_(deg)" = auckland_point_air_quality_raw."wind_deviation_(deg)",
  "wind_speed_deviation_(m/s)" = auckland_point_air_quality_raw."wind_speed_deviation_(m/s)",
  "wind_temperature_(degC)" = auckland_point_air_quality_raw."wind_temperature_(degC)",
  "humidity_(%)" = auckland_point_air_quality_raw."humidity_(%)",
  "rainfall_(mm)" = auckland_point_air_quality_raw."rainfall_(mm)",
  "barometric_pressure_(hPa)" = auckland_point_air_quality_raw."barometric_pressure_(hPa)",
  "PM10_(ug/m3)" = auckland_point_air_quality_raw."PM10_(ug/m3)"
FROM
  public.auckland_point_air_quality_raw
WHERE
  gladstone_composite_air_quality_raw.datetime_id = auckland_point_air_quality_raw.datetime_id;
SELECT * FROM public.gladstone_composite_air_quality_raw ORDER BY datetime_id ASC;

-- #####
-- 05. Insert South - - > Compound dataset, adding missing values.
-- #####
UPDATE
  gladstone_composite_air_quality_raw
SET
  "nitrogen_oxide_(ppm)" = public.south_gladstone_air_quality_raw."nitrogen_oxide_(ppm)",
  "nitrogen_oxides_(ppm)" = public.south_gladstone_air_quality_raw."nitrogen_oxides_(ppm)",
  "PM2.5_(ug/m3)" = public.south_gladstone_air_quality_raw."PM2.5_(ug/m3)",
  "visibility-reducing_particles_(Mm-1)" = public.south_gladstone_air_quality_raw."visibility-
reducing_particles_(Mm-1)"
FROM
  public.south_gladstone_air_quality_raw
WHERE
  gladstone_composite_air_quality_raw.datetime_id =
  public.south_gladstone_air_quality_raw.datetime_id;

-- #####
-- 06. Brisbane, creating clone composite table, using raw as backup:
-- #####
brisbane_composite_air_quality_raw_filled or
gladstone_composite_air_quality_raw_filled
CREATE TABLE brisbane_composite_air_quality_raw_filled AS
SELECT * FROM public.brisbane_composite_air_quality_raw;

-- Checks for missing values.
SELECT * FROM public.brisbane_composite_air_quality_raw_filled
WHERE "wind_direction_(degN)" IS NULL
ORDER BY datetime_id ASC;

```

```

-- #####
-- 07. Update function version 1 for Gladstone:
-- #####
--UPDATE gladstone_composite_air_quality_raw_filled
--SET
--"wind_direction_(degN)" = public.south_gladstone_air_quality_raw."wind_direction_(degN)"
--FROM
--public.south_gladstone_air_quality_raw
--WHERE
--(public.gladstone_composite_air_quality_raw_filled."wind_direction_(degN)" IS NULL)
--AND
--(public.south_gladstone_air_quality_raw."wind_direction_(degN)" IS NOT NULL)
--AND
--(public.gladstone_composite_air_quality_raw_filled.datetime_id =
public.south_gladstone_air_quality_raw.datetime_id);

-- #####
-- 08. Checking if Gladstone vars are NULL:
-- #####
-- SELECT datetime_id, "wind_direction_(degN)", "wind_speed_(m/s)",
-- "wind_deviation_(deg)", "wind_speed_deviation_(m/s)",
-- "wind_temperature_(degC)", "humidity_(%)",
-- "rainfall_(mm)", "barometric_pressure_(hPa)", "PM10_(ug/m3)"
-- FROM public.south_gladstone_air_quality_raw
-- WHERE "wind_speed_(m/s)" IS NULL
-- OR "wind_direction_(degN)" IS NULL
-- OR "wind_deviation_(deg)" IS NULL
-- OR "wind_speed_deviation_(m/s)" IS NULL
-- OR "wind_temperature_(degC)" IS NULL
-- OR "humidity_(%)" IS NULL
-- OR "rainfall_(mm)" IS NULL
-- OR "barometric_pressure_(hPa)" IS NULL
-- OR "PM10_(ug/m3)" IS NULL
-- ORDER BY datetime_id ASC;

-- #####
-- 09. Filling NULLS, Brisbane version:
-- #####
--UPDATE brisbane_composite_air_quality_raw_filled
--SET
--"carbon_monoxide_(ppm)" = public.south_brisbane_air_quality_raw."carbon_monoxide_(ppm)"
--FROM
--public.south_brisbane_air_quality_raw
--WHERE
--(public.brisbane_composite_air_quality_raw_filled."carbon_monoxide_(ppm)" IS NULL)
--AND
--(public.south_brisbane_air_quality_raw."carbon_monoxide_(ppm)" IS NOT NULL)
--AND
--(public.brisbane_composite_air_quality_raw_filled.datetime_id =
public.south_brisbane_air_quality_raw.datetime_id);
--TLDR wind_direction to PM 2.5 Wool (CBD base) %%, then Nit Oxi, Dio, Mono South (as
Wool base),

```

```

-- #####
-- 10. NULL checker Brisbane
-- #####
SELECT datetime_id, "wind_direction_(degN)", "wind_speed_(m/s)",
-- "wind_deviation_(deg)", "wind_speed_deviation_(m/s)",
-- "wind_temperature_(degC)", "humidity_(%)",
-- "rainfall_(mm)", "barometric_pressure_(hPa)" , "PM10_(ug/m3)", "PM2.5_(ug/m3)",
-- "nitrogen_oxide_(ppm)", "nitrogen_dioxide_(ppm)", "nitrogen_oxides_(ppm)"
-- , "carbon_monoxide_(ppm)"
-- FROM public.brisbane_composite_air_quality_raw_filled
-- WHERE "wind_direction_(degN)" IS NULL
-- OR "wind_speed_(m/s)" IS NULL
-- OR "wind_deviation_(deg)" IS NULL
-- OR "wind_speed_deviation_(m/s)" IS NULL
-- OR "wind_temperature_(degC)" IS NULL
-- OR "humidity_(%)" IS NULL
-- OR "rainfall_(mm)" IS NULL
-- OR "barometric_pressure_(hPa)" IS NULL
-- OR "PM10_(ug/m3)" IS NULL
-- OR "PM2.5_(ug/m3)" IS NULL
-- OR "nitrogen_oxide_(ppm)" IS NULL
-- OR "nitrogen_dioxide_(ppm)" IS NULL
-- OR "nitrogen_oxides_(ppm)" IS NULL
-- OR "carbon_monoxide_(ppm)" IS NULL
-- ORDER BY datetime_id ASC;

-- #####
-- 11. Checks filled Brisbane dataset for NULLs:
-- #####
-- SELECT *
-- FROM public.brisbane_composite_air_quality_raw_filled
-- WHERE "wind_direction_(degN)" IS NULL
-- OR "wind_speed_(m/s)" IS NULL
-- OR "wind_deviation_(deg)" IS NULL
-- OR "wind_speed_deviation_(m/s)" IS NULL
-- OR "wind_temperature_(degC)" IS NULL
-- OR "humidity_(%)" IS NULL
-- OR "rainfall_(mm)" IS NULL
-- OR "barometric_pressure_(hPa)" IS NULL
-- OR "PM10_(ug/m3)" IS NULL
-- OR "PM2.5_(ug/m3)" IS NULL
-- OR "nitrogen_oxide_(ppm)" IS NULL
-- OR "nitrogen_dioxide_(ppm)" IS NULL
-- OR "nitrogen_oxides_(ppm)" IS NULL
-- OR "carbon_monoxide_(ppm)" IS NULL
-- OR "visibility_reducing_particles_(Mm-1)" IS NULL
-- ORDER BY datetime_id ASC;

```

```

-- #####
-- 12. Gladstone version:
-- #####
-- SELECT *
-- FROM public.gladstone_composite_air_quality_raw_filled
-- WHERE "wind_direction_(degN)" IS NULL
-- OR "wind_speed_(m/s)" IS NULL
-- OR "wind_deviation_(deg)" IS NULL
-- OR "wind_speed_deviation_(m/s)" IS NULL
-- OR "wind_temperature_(degC)" IS NULL
-- OR "humidity_(%)" IS NULL
-- OR "rainfall_(mm)" IS NULL
-- OR "barometric_pressure_(hPa)" IS NULL
-- OR "PM10_(ug/m3)" IS NULL--ALL good till here
-- OR "PM2.5_(ug/m3)" IS NULL -- 483 - bit too big
-- OR "nitrogen_dioxide_(ppm)" IS NULL -- None
-- OR "nitrogen_oxides_(ppm)" IS NULL -- 51, Good!
-- OR "nitrogen_oxide_(ppm)" IS NULL --51, Good!
-- OR "visibility-reducing_particles_(Mm-1)" IS NULL -- 90, Solid!
-- OR "ozone_(ppm)" IS NULL -- Within 5% band, maybe call 1%?
-- OR "sulfur_dioxide_(ppm)" IS NULL -- Within 5% band, maybe call 1%?
-- OR "xylenes_(ppb)" IS NULL -- In 5% band
-- OR "benzene_(ppb)" IS NULL -- Too much
-- OR "toulene_(ppb)" IS NULL -- Too much
-- OR "formaldehyde_(ppb)" IS NULL -- Too much
-- OR "visibility-reducing_particles_(Mm-1)" IS NULL
-- ORDER BY datetime_id ASC;

-- #####
-- 13. Created Brisbane table for filled dataset:
-- #####
-- CREATE TABLE clean_brisbane_composite_air_quality AS
-- SELECT * FROM public.brisbane_composite_air_quality_raw_filled;
-- SELECT * FROM public. clean_brisbane_composite_air_quality
-- ORDER BY datetime_id ASC;

-- #####
--14. Created Gladstone table for completely filled dataset (bar dropped values)
-- #####
CREATE TABLE clean_gladstone_composite_air_quality AS
SELECT * FROM public.gladstone_composite_air_quality_raw_filled;
SELECT * FROM public. clean_gladstone_composite_air_quality
ORDER BY datetime_id ASC;

```

```

-- #####
-- 15. Query process to create and store a table with front-fillable data for brisbane:
-- #####
-- CREATE TABLE temp_table
-- AS SELECT datetime_id,
-- COALESCE("carbon_monoxide_(ppm)",
-- FIRST_VALUE("carbon_monoxide_(ppm)") OVER(
-- PARTITION BY generic_group ORDER BY datetime_id ASC)) AS data_filled
-- FROM (
-- SELECT datetime_id, "carbon_monoxide_(ppm)",
-- COUNT("carbon_monoxide_(ppm)") OVER (ORDER BY datetime_id ASC) AS generic_group
-- FROM public.brisbane_clean_data
-- ORDER BY
-- datetime_id ASC);

-- #####
-- 16. Update function
-- #####
-- UPDATE public.brisbane_clean_data
-- SET
-- "carbon_monoxide_(ppm)" = public.temp_table.data_filled
-- FROM
-- public.temp_table
-- WHERE
-- (public.brisbane_clean_data.datetime_id = temp_table.datetime_id);
--## Checker
-- SELECT datetime_id,"carbon_monoxide_(ppm)"
-- FROM public.brisbane_clean_data
-- WHERE "carbon_monoxide_(ppm)" IS NULL
-- OR brisbane_clean_data.datetime_id IN(6200,6201,6202,6203,6468,6469,6470)
-- ORDER BY datetime_id ASC;

-- #####
-- 17. Null checker for highlighted nulls, used to test if process works (Bris):
-- #####
-- SELECT * FROM public.brisbane_clean_data
-- WHERE "wind_direction_(degN)" IS NULL
-- OR "wind_speed_(m/s)" IS NULL
-- OR "wind_deviation_(deg)" IS NULL
-- OR "wind_speed_deviation_(m/s)" IS NULL
-- OR "wind_temperature_(degC)" IS NULL
-- OR "humidity_%" IS NULL
-- OR "rainfall_(mm)" IS NULL
-- OR "barometric_pressure_(hPa)" IS NULL
-- OR "PM10_(ug/m3)" IS NULL
-- OR "PM2.5_(ug/m3)" IS NULL
-- OR "nitrogen_oxide_(ppm)" IS NULL
-- OR "nitrogen_dioxide_(ppm)" IS NULL
-- OR "nitrogen_oxides_(ppm)" IS NULL
-- OR "carbon_monoxide_(ppm)" IS NULL
-- OR public.brisbane_clean_data.datetime_id IN(6200,6201,6202,6203,6468,6469,6470)
-- ORDER BY datetime_id ASC;

```

```

-- #####
-- 18. Create New Table for Gladstone
-- #####
-- CREATE TABLE gladstone_clean_data AS
-- SELECT * FROM public.gladstone_composite_air_quality_raw_filled;

-- #####
-- 19. Query to check number of nulls per category for Gladstone
-- #####
-- SELECT
-- SUM(CASE WHEN "wind_direction_(degN)" IS NULL THEN 1 ELSE 0 END) AS
"wind_direction_nulls",
-- SUM(CASE WHEN "wind_speed_(m/s)" IS NULL THEN 1 ELSE 0 END) AS
"wind_speed_nulls",
-- SUM(CASE WHEN "wind_deviation_(deg)" IS NULL THEN 1 ELSE 0 END) AS
"wind_deviation_nulls",
-- SUM(CASE WHEN "wind_speed_deviation_(m/s)" IS NULL THEN 1 ELSE 0 END) AS
"wind_speed_deviation_nulls",
-- SUM(CASE WHEN "wind_temperature_(degC)" IS NULL THEN 1 ELSE 0 END) AS
"wind_temperature_nulls",
-- SUM(CASE WHEN "humidity_(%)" IS NULL THEN 1 ELSE 0 END) AS "humidity_nulls",
-- SUM(CASE WHEN "rainfall_(mm)" IS NULL THEN 1 ELSE 0 END) AS "rainfall_nulls",
-- SUM(CASE WHEN "barometric_pressure_(hPa)" IS NULL THEN 1 ELSE 0 END) AS
"bp_nulls",
-- SUM(CASE WHEN "PM10_(ug/m3)" IS NULL THEN 1 ELSE 0 END) AS "PM10_nulls",
-- SUM(CASE WHEN "ozone_(ppm)" IS NULL THEN 1 ELSE 0 END) AS "ozone_nulls",
-- SUM(CASE WHEN "nitrogen_dioxide_(ppm)" IS NULL THEN 1 ELSE 0 END) AS
"nitrogen_dioxide_nulls",
-- SUM(CASE WHEN "sulfur_dioxide_(ppm)" IS NULL THEN 1 ELSE 0 END) AS
"sulfur_dioxide_nulls",
-- SUM(CASE WHEN "benzene_(ppb)" IS NULL THEN 1 ELSE 0 END) AS "benzene_nulls",
-- SUM(CASE WHEN "toulene_(ppb)" IS NULL THEN 1 ELSE 0 END) AS "toulene_nulls",
-- SUM(CASE WHEN "xylenes_(ppb)" IS NULL THEN 1 ELSE 0 END) AS "xylenes_nulls",
-- SUM(CASE WHEN "formaldehyde_(ppb)" IS NULL THEN 1 ELSE 0 END) AS
"formaldehyde_nulls",
-- SUM(CASE WHEN "nitrogen_oxide_(ppm)" IS NULL THEN 1 ELSE 0 END) AS
"nitrogen_oxide_nulls",
-- SUM(CASE WHEN "nitrogen_oxides_(ppm)" IS NULL THEN 1 ELSE 0 END) AS
"nitrogen_oxides_nulls",
-- SUM(CASE WHEN "PM2.5_(ug/m3)" IS NULL THEN 1 ELSE 0 END) AS "PM2.5_nulls",
-- SUM(CASE WHEN "visibility-reducing_particles_(Mm-1)" IS NULL THEN 1 ELSE 0 END)
AS "vrp_nulls"
-- FROM public.gladstone_composite_air_quality_raw_filled;
-- -- TLDR "PM10_(ug/m3)", "ozone_(ppm)", "sulfur_dioxide_(ppm)", "xylenes_(ppb)",
"nitrogen_oxide_(ppm)",
-- -- "nitrogen_oxides_(ppm)", "visibility-reducing_particles_(Mm-1)" are under 5%,
-- -- the rest are not. Focus on completing these!
-- -- Others over 5%: "benzene_(ppb)", "PM2.5_(ug/m3)", "formaldehyde_(ppb)", "toulene_(ppb)"
-- -- Largest: formaldehyde w/ 16.639% missing values

```

```

-- #####
-- 20. Query to generate forward-filled data for Gladstone:
-- #####
-- DROP TABLE IF EXISTS temp_table CASCADE;
-- CREATE TABLE temp_table
-- AS SELECT datetime_id,
-- COALESCE("PM2.5_(ug/m3)",
-- FIRST_VALUE("PM2.5_(ug/m3)") OVER(
-- PARTITION BY generic_group ORDER BY datetime_id ASC)) AS data_filled
-- FROM (
-- SELECT datetime_id, "PM2.5_(ug/m3)",
-- COUNT("PM2.5_(ug/m3)") OVER (ORDER BY datetime_id ASC) AS generic_group
-- FROM public.gladstone_composite_air_quality_raw_filled
-- ORDER BY
-- datetime_id ASC);

-- #####
-- 21. Checks if there is nulls in Colms (nums)
-- #####
-- SELECT temp_table.datetime_id, data_filled, public.gladstone_clean_data."PM2.5_(ug/m3)"
-- FROM temp_table FULL JOIN public.gladstone_clean_data
-- ON temp_table.datetime_id = public.gladstone_clean_data.datetime_id
-- WHERE public.gladstone_clean_data."PM2.5_(ug/m3)" IS NULL
-- OR public.gladstone_clean_data.datetime_id IN(250,251,252,253,254)
-- ORDER BY datetime_id asc;

-- #####
-- 22. Gladstone Update function
-- #####
-- UPDATE public.gladstone_clean_data
-- SET
-- "PM2.5_(ug/m3)" = public.temp_table.data_filled
-- FROM
-- public.temp_table
-- WHERE
-- (public.gladstone_clean_data.datetime_id = temp_table.datetime_id);
-- --## Checker
-- SELECT datetime_id,"PM2.5_(ug/m3)"
-- FROM public.gladstone_clean_data
-- WHERE "nitrogen_oxides_(ppm)" IS NULL
-- OR public.gladstone_clean_data.datetime_id IN(1139,1140,1141,1142)
-- ORDER BY datetime_id ASC;

```



```

-- #####
-- 23. Simpler Gladstone Null checker, done for computation speed and visual feedback:
-- --Checker for Nulls in gladstone clean
-- #####
-- SELECT * FROM public.gladstone_clean_data
-- WHERE "ozone_(ppm)" IS NULL
-- OR "sulfur_dioxide_(ppm)" IS NULL
-- --OR "benzene_(ppb)" IS NULL
-- --OR "toulene_(ppb)" IS NULL
-- OR "xylenes_(ppb)" IS NULL
-- --OR "formaldehyde_(ppb)" IS NULL
-- OR "visibility-reducing_particles_(Mm-1)" IS NULL
-- OR "PM10_(ug/m3)" IS NULL
-- OR "PM2.5_(ug/m3)" IS NULL
-- OR "nitrogen_oxide_(ppm)" IS NULL
-- OR "nitrogen_dioxide_(ppm)" IS NULL
-- OR "nitrogen_oxides_(ppm)" IS NULL
-- --OR public.gladstone_clean_data.datetime_id IN(250,251,252,253,254)
-- ORDER BY datetime_id ASC;
-- -- Others over 5%: "benzene_(ppb)","PM2.5_(ug/m3)","formaldehyde_(ppb)","toulene_(ppb)"
-- ins: IN(2628,2629,2630) IN(1630,1631,1632,1633,1634), IN(1139,1140,1141,1142),
IN(250,251,252,253,254) –Values to check

-- #####
-- 24. Dropping Gladstone columns with large unfilled data:
-- #####
-- CLEANING GLADSTONE COLUMNS:
-- ALTER TABLE public.gladstone_clean_data
-- DROP COLUMN "benzene_(ppb)",
-- DROP COLUMN "formaldehyde_(ppb)",
-- DROP COLUMN "toulene_(ppb)";
-- SELECT * FROM public.gladstone_clean_data
-- ORDER BY datetime_id asc;

```

```

-- #####
-- 25. QUERY: PM10 & 2.5
-- #####
-- DROP TABLE IF EXISTS "Query: Air Quality Data" CASCADE;
-- CREATE TABLE "Query: Air Quality Data" AS
-- (SELECT
-- public.brisbane_clean_data.datetime_id AS "Datetime ID",
-- public.brisbane_clean_data.datevalue "Date (YYYY/MM/DD)",
-- public.brisbane_clean_data.timevalue AS "Hour (24:00)",
-- public.brisbane_clean_data."PM10_(ug/m3)" AS "brisbane PM10 (ug/m3)",
-- public.brisbane_clean_data."PM2.5_(ug/m3)" AS "brisbane PM2.5 (ug/m3)",
-- public.brisbane_clean_data."visibility_reducing_particles_(Mm-1)" AS "brisbane VRPs (Mm-1)",
-- public.gladstone_clean_data."PM10_(ug/m3)" AS "gladstone PM10 (ug/m3)",
-- public.gladstone_clean_data."PM2.5_(ug/m3)" AS "gladstone PM2.5 (ug/m3)",
-- public.gladstone_clean_data."visibility-reducing_particles_(Mm-1)" AS "gladstone VRPs (Mm-1)"
-- FROM public.brisbane_clean_data FULL JOIN public.gladstone_clean_data
-- ON public.brisbane_clean_data.datetime_id = public.gladstone_clean_data.datetime_id
-- ORDER BY "Datetime ID" ASC);
-- SELECT * FROM "Query: Air Quality Data" ORDER BY "Datetime ID" ASC;

-- #####
-- 26. QUERY: H T, BP & R
-- #####
-- DROP TABLE IF EXISTS "Query: Athmosphere metrics" CASCADE;
-- CREATE TABLE "Query: Athmosphere metrics" AS
-- -- "Rain, Humidity, Temp. & Barometric Pressure"
-- (SELECT
-- public.brisbane_clean_data.datetime_id AS "Datetime ID",
-- public.brisbane_clean_data.datevalue "Date (YYYY/MM/DD)",
-- public.brisbane_clean_data.timevalue AS "Hour (24:00)",
-- public.brisbane_clean_data."humidity_(%)" AS "brisbane humidity (%)",
-- public.brisbane_clean_data."rainfall_(mm)" AS "brisbane rainfall (mm)",
-- public.brisbane_clean_data."wind_temperature_(degC)" AS "brisbane temperature (degC)",
-- public.brisbane_clean_data."barometric_pressure_(hPa)" AS "brisbane Barometric Pressure (hPa)",
-- public.gladstone_clean_data."humidity_(%)" AS "gladstone humidity (%)",
-- public.gladstone_clean_data."rainfall_(mm)" AS "gladstone rainfall (mm)",
-- public.gladstone_clean_data."wind_temperature_(degC)" AS "gladstone temperature (degC)",
-- public.gladstone_clean_data."barometric_pressure_(hPa)" AS "gladstone Barometric Pressure (hPa)"
-- FROM public.brisbane_clean_data FULL JOIN public.gladstone_clean_data
-- ON public.brisbane_clean_data.datetime_id = public.gladstone_clean_data.datetime_id
-- ORDER BY "Datetime ID" ASC);
-- SELECT * FROM "Query: Athmosphere metrics" ORDER BY "Datetime ID" ASC;

```

```

-- #####
-- 27. QUERY: Chemical Measures
-- #####
-- DROP TABLE IF EXISTS "Query: Air Molecule Measurements" CASCADE;
-- --QUERY: Chemical Measures
-- CREATE TABLE "Query: Air Molecule Measurements" AS
-- (SELECT
-- public.brisbane_clean_data.datetime_id AS "Datetime ID",
-- public.brisbane_clean_data.datevalue "Date (YYYY/MM/DD)",
-- public.brisbane_clean_data.timevalue AS "Hour (24:00)",
-- public.brisbane_clean_data."nitrogen_oxide_(ppm)" AS "brisbane nitrogen oxide (ppm)",
-- public.brisbane_clean_data."nitrogen_dioxide_(ppm)" AS "brisbane nitrogen dioxide (ppm)",
-- public.brisbane_clean_data."nitrogen_oxides_(ppm)" AS "brisbane nitrogen oxides (ppm)",
-- public.brisbane_clean_data."carbon_monoxide_(ppm)" AS "brisbane carbon monoxide (ppm)",
-- public.gladstone_clean_data."nitrogen_oxide_(ppm)" AS "gladstone nitrogen oxide (ppm)",
-- public.gladstone_clean_data."nitrogen_dioxide_(ppm)" AS "gladstone nitrogen dioxide (ppm)",
-- public.gladstone_clean_data."nitrogen_oxides_(ppm)" AS "gladstone nitrogen oxides (ppm)",
-- public.gladstone_clean_data."sulfur_dioxide_(ppm)" AS "gladstone sulfur dioxide (ppm)",
-- public.gladstone_clean_data."ozone_(ppm)" AS "gladstone ozone (ppm)",
-- public.gladstone_clean_data."xylenes_(ppb)" AS "gladstone xylenes (ppb)"
-- FROM public.brisbane_clean_data FULL JOIN public.gladstone_clean_data
-- ON public.brisbane_clean_data.datetime_id = public.gladstone_clean_data.datetime_id
-- ORDER BY "Datetime ID" ASC);
-- SELECT * FROM "Query: Air Molecule Measurements" ORDER BY "Datetime ID" ASC;

```

```

-- #####
-- 28. QUERY: PM10 & 2.5 BINS: Safe vs unsafe levels
-- #####
-- DROP TABLE IF EXISTS "Query: Air Quality Guidelines" CASCADE;

-- Needed two commands - one to generate table and one to fill it.
-- CREATE TABLE "Query: Air Quality Guidelines"(
--     "Day ID" INTEGER PRIMARY KEY,
--     "DateStamp" DATE,
--     "Brisbane PM10 Daily Average" REAL,
--     "Brisbane PM10 Guidline Safety Value" VARCHAR(10),
--     "Gladstone PM10 Daily Average" REAL,
--     "Gladstone PM10 Guidline Safety Value" VARCHAR(10),
--     "Brisbane PM2.5 Daily Average" REAL,
--     "Brisbane PM2.5 Guidline Safety Value" VARCHAR(10),
--     "Gladstone PM2.5 Daily Average" REAL,
--     "Gladstone PM2.5 Guidline Safety Value" VARCHAR(10),
--     "Brisbane VRPs Daily Average" REAL,
--     "Brisbane VRPs Guidline Safety Value" VARCHAR(10),
--     "Gladstone VRPs Daily Average" REAL,
--     "Gladstone VRPs Guidline Safety Value" VARCHAR(10),
--     "Brisbane NO2 Daily Average" REAL,
--     "Brisbane NO2 Guidline Safety Value" VARCHAR(10),
--     "Gladstone NO2 Daily Average" REAL,
--     "Gladstone NO2 Guidline Safety Value" VARCHAR(10),
--     "Brisbane CO Daily Average" REAL,
--     "Brisbane CO Guidline Safety Value" VARCHAR(10),
--     "Gladstone O3 Daily Average" REAL,
--     "Gladstone O3 Guidline Safety Value" VARCHAR(10),
--     "Gladstone SO2 Daily Average" REAL,
--     "Gladstone SO2 Guidline Safety Value" VARCHAR(10)
--     --These are daily metrics that need to be fixed first.
-- )

-- #####
-- 29. Query to generate Air quality with WHO guidelines
-- #####
--CREATE TABLE "Query: Air Quality Guidelines" AS(
SELECT ROW_NUMBER() OVER (ORDER BY public."Query: Air Molecule
Measurements"."Date (YYYY/MM/DD)" ASC) AS "Date ID", -- is ID value
public."Query: Air Molecule Measurements"."Date (YYYY/MM/DD)" AS "Date
(YYYY/MM/DD)",
AVG(public."Query: Athmosphere metrics"."brisbane temperature (degC)") AS "Brisbane Average
Temperature",
AVG(public."Query: Athmosphere metrics"."gladstone temperature (degC)") AS "Gladstone
Average Temperature",
AVG("brisbane PM10 (ug/m3)") AS "Brisbane PM10 Daily Average",
AVG("gladstone PM10 (ug/m3)") AS "Gladstone PM10 Daily Average",
AVG("brisbane PM2.5 (ug/m3)") AS "Brisbane PM2.5 Daily Average",
AVG("gladstone PM2.5 (ug/m3)") AS "Gladstone PM2.5 Daily Average",
AVG("brisbane nitrogen dioxide (ppm)") AS "Brisbane NO2 Daily Average",
AVG("gladstone nitrogen dioxide (ppm)") AS "Gladstone NO2 Daily Average",

```

```

AVG("brisbane carbon monoxide (ppm)") AS "Brisbane CO Daily Average",
AVG("gladstone ozone (ppm)") AS "Gladstone O3 Daily Average",
AVG("gladstone sulfur dioxide (ppm)") AS "Gladstone SO2 Daily Average",
CASE WHEN AVG("brisbane PM10 (ug/m3)") > 45 THEN 'Unsafe' ELSE 'Safe' END AS "PM10
Safety Levels Brisbane",
CASE WHEN AVG("gladstone PM10 (ug/m3)") > 45 THEN 'Unsafe' ELSE 'Safe' END AS "PM10
Safety Levels Gladstone",
CASE WHEN AVG("brisbane PM2.5 (ug/m3)") > 15 THEN 'Unsafe' ELSE 'Safe' END AS "PM2.5
Safety Levels Brisbane",
CASE WHEN AVG("gladstone PM2.5 (ug/m3)") > 15 THEN 'Unsafe' ELSE 'Safe' END AS
"PM2.5 Safety Levels Gladstone",
-- CASE WHEN AVG("brisbane nitrogen dioxide (ppm)") > 25*0.0217367677426*24.45 THEN
'Unsafe' ELSE 'Safe' END AS "NO2 Safety Levels Brisbane", -- 46.005g/mol
-- CASE WHEN AVG("gladstone nitrogen dioxide (ppm)") > 25*0.0217367677426*24.45 THEN
'Unsafe' ELSE 'Safe' END AS "NO2 Safety Levels Gladstone",
-- CASE WHEN AVG("brisbane carbon monoxide (ppm)") > 15*24.45*0.035701535166 THEN
'Unsafe' ELSE 'Safe' END AS "CO Safety Levels Brisbane", -- 28.010 g/mol
-- CASE WHEN AVG("gladstone ozone (ppm)") > 100*24.45*0.021277953912 THEN 'Unsafe'
ELSE 'Safe' END AS "O3 Safety Levels Gladstone", -- 47.997 g/mol
-- CASE WHEN AVG("gladstone sulfur dioxide (ppm)") > 40*24.45*0.0156089033185 THEN
'Unsafe' ELSE 'Safe' END AS "SO2 Safety Levels Gladstone" -- 64.066 g/mol
-- Adjusted (Simple)
CASE WHEN AVG("brisbane nitrogen dioxide (ppm)") > 25*(273.16+ MAX(public."Query:
Athmosphere metrics"."brisbane temperature (degC)")*62.4/(1*46.005) THEN 'Unsafe' ELSE
'Safe' END AS "NO2 Safety Levels Brisbane", -- 46.005g/mol
CASE WHEN AVG("gladstone nitrogen dioxide (ppm)") > 25*(273.16+ MAX(public."Query:
Athmosphere metrics"."gladstone temperature (degC)")*62.4/(1*46.005) THEN 'Unsafe' ELSE
'Safe' END AS "NO2 Safety Levels Gladstone",
CASE WHEN AVG("brisbane carbon monoxide (ppm)") > 15*(273.16+ MAX(public."Query:
Athmosphere metrics"."brisbane temperature (degC)")*62.4/(1*28.010) THEN 'Unsafe' ELSE
'Safe' END AS "CO Safety Levels Brisbane", -- 28.010 g/mol
CASE WHEN AVG("gladstone ozone (ppm)") > 100*(273.16+ MAX(public."Query: Athmosphere
metrics"."gladstone temperature (degC)")*62.4/(1*47.997) THEN 'Unsafe' ELSE 'Safe' END AS
"O3 Safety Levels Gladstone", -- 47.997 g/mol
CASE WHEN AVG("gladstone sulfur dioxide (ppm)") > 40*(273.16+ MAX(public."Query:
Athmosphere metrics"."gladstone temperature (degC)")*62.4/(1*64.066) THEN 'Unsafe' ELSE
'Safe' END AS "SO2 Safety Levels Gladstone" -- 64.066 g/mol

-- Note: mg/m3 to ppm/ppb scale to fix to temperature to temp average, then checked outliers with
max/mins.

```

```
-- #####
-- 30. Scratch 5 Query 2 Wind Ranges whisker plot (see additional procedures at bottom of file)
-- #####
--Query 2: Wind speed deviation for whisker plot
DROP TABLE IF EXISTS "Query: Wind Speed Deviations" CASCADE;
```

```
CREATE TABLE "Query: Wind Speed Deviations" AS(
SELECT ROW_NUMBER() OVER (ORDER BY public.brisbane_clean_data."datetime_id" ASC)
AS "Date ID", -- is ID value
public.brisbane_clean_data."datevalue" AS "Date (YYYY/MM/DD)",
public.brisbane_clean_data."timevalue" AS "Hour (24:00)",
public.brisbane_clean_data."wind_speed_(m/s)" AS "Brisbane Wind Speed Km/h",
public.gladstone_clean_data."wind_speed_(m/s)" AS "Gladstone Wind Speed Km/h",
3.6*public.brisbane_clean_data."wind_speed_(m/s)" AS "Brisbane Wind Std Deviation Km/h", --
Scaling factor: 1 m/s = 3.6 km/h
3.6*public.gladstone_clean_data."wind_speed_(m/s)" AS "Gladstone Wind Std Deviation Km/h"
FROM public.brisbane_clean_data
FULL JOIN public.gladstone_clean_data
ON public.brisbane_clean_data."datetime_id" = public.gladstone_clean_data."datetime_id"
ORDER BY "Date ID" ASC);
```

```
SELECT * FROM "Query: Wind Speed Deviations" ORDER BY "Date ID" ASC
```

```
-- #####
-- 31. Wind speed deviation for whisker plot (first correction, m/s to km/h)
-- #####
DROP TABLE IF EXISTS "Query: Wind Speed Deviations" CASCADE;
```

```
CREATE TABLE "Query: Wind Speed Deviations" AS(
SELECT ROW_NUMBER() OVER (ORDER BY public.brisbane_clean_data."datetime_id" ASC)
AS "Date ID", -- is ID value
public.brisbane_clean_data."datevalue" AS "Date (YYYY/MM/DD)",
public.brisbane_clean_data."timevalue" AS "Hour (24:00)",
public.brisbane_clean_data."wind_direction_(degN)" AS "Brisbane Wind Direction (degN)",
public.gladstone_clean_data."wind_direction_(degN)" AS "Gladstone Wind Direction (degN)",
3.6*public.brisbane_clean_data."wind_speed_(m/s)" AS "Brisbane Wind Speed Km/h", --Scaling
factor: 1 m/s = 3.6 km/h
3.6*public.gladstone_clean_data."wind_speed_(m/s)" AS "Gladstone Wind Speed Km/h",
3.6*public.brisbane_clean_data."wind_speed_deviation_(m/s)" AS "Brisbane Wind Std Deviation
Km/h",
3.6*public.gladstone_clean_data."wind_speed_deviation_(m/s)" AS "Gladstone Wind Std Deviation
Km/h"
FROM public.brisbane_clean_data
FULL JOIN public.gladstone_clean_data
ON public.brisbane_clean_data."datetime_id" = public.gladstone_clean_data."datetime_id"
ORDER BY "Date ID" ASC);
```

```
SELECT * FROM "Query: Wind Speed Deviations" ORDER BY "Date ID" ASC
```

```

-- #####
-- 32. Wind Speed Deviation Addition: Injected stand deviation results into a compound result:
-- Note: Was relabelled to a placeholder file, so table name irrelevant.
-- #####
-- DROP TABLE IF EXISTS "Wind Direction Query T" CASCADE;
-- Create Table "Wind Direction Query T" AS
-- (SELECT
-- --SELECT
-- public."Query: Wind Speed Deviations T"."Datetime ID", "Date (YYYY/MM/DD)", "Hour
(24:00)",
-- 5*public."Query: Wind Speed Deviations T"."Datetime ID"-4 AS "DTIDA",
-- 5*public."Query: Wind Speed Deviations T"."Datetime ID"-3 AS "DTIDB",
-- 5*public."Query: Wind Speed Deviations T"."Datetime ID"-2 AS "DTIDC",
-- 5*public."Query: Wind Speed Deviations T"."Datetime ID"-1 AS "DTIDD",
-- 5*public."Query: Wind Speed Deviations T"."Datetime ID" AS "DTIDE",
-- public."Query: Wind Speed Deviations T"."Brisbane Wind Speed Km/h" - 2*public."Query:
Wind Speed Deviations T"."Brisbane Wind Std Deviation Km/h" AS "Brisbane WS0",
-- public."Query: Wind Speed Deviations T"."Brisbane Wind Speed Km/h" - public."Query: Wind
Speed Deviations T"."Brisbane Wind Std Deviation Km/h" AS "Brisbane WS1",
-- public."Query: Wind Speed Deviations T"."Brisbane Wind Speed Km/h" AS "Brisbane WS2",
-- public."Query: Wind Speed Deviations T"."Brisbane Wind Speed Km/h" + public."Query: Wind
Speed Deviations T"."Brisbane Wind Std Deviation Km/h" AS "Brisbane WS3",
-- public."Query: Wind Speed Deviations T"."Brisbane Wind Speed Km/h" + 2*public."Query:
Wind Speed Deviations T"."Brisbane Wind Std Deviation Km/h" AS "Brisbane WS4",
-- public."Query: Wind Speed Deviations T"."Gladstone Wind Speed Km/h" - 2*public."Query:
Wind Speed Deviations T"."Gladstone Wind Std Deviation Km/h" AS "Gladstone WS0",
-- public."Query: Wind Speed Deviations T"."Gladstone Wind Speed Km/h" - public."Query: Wind
Speed Deviations T"."Gladstone Wind Std Deviation Km/h" AS "Gladstone WS1",
-- public."Query: Wind Speed Deviations T"."Gladstone Wind Speed Km/h" AS "Gladstone WS2",
-- public."Query: Wind Speed Deviations T"."Gladstone Wind Speed Km/h" + public."Query:
Wind Speed Deviations T"."Gladstone Wind Std Deviation Km/h" AS "Gladstone WS3",
-- public."Query: Wind Speed Deviations T"."Gladstone Wind Speed Km/h" + 2*public."Query:
Wind Speed Deviations T"."Gladstone Wind Std Deviation Km/h" AS "Gladstone WS4"
-- FROM public."Query: Wind Speed Deviations T" ORDER BY "Datetime ID" ASC
-- );
-- SELECT
-- (SELECT "DTIDA" FROM public."AllWind" AS "DT ID"
-- UNION ALL
-- SELECT "DTIDB" FROM public."AllWind"
-- UNION ALL
-- SELECT "DTIDC" FROM public."AllWind"
-- UNION ALL
-- SELECT "DTIDD" FROM public."AllWind"
-- UNION ALL
-- SELECT "DTIDE" FROM public."AllWind"
-- ),
-- (SELECT "Brisbane WS0" FROM public."AllWind" AS "Brisbane Winds km/h" UNION ALL
-- SELECT "Brisbane WS1" FROM public."AllWind" UNION ALL
-- SELECT "Brisbane WS2" FROM public."AllWind" UNION ALL
-- SELECT "Brisbane WS3" FROM public."AllWind" UNION ALL
-- SELECT "Brisbane WS4" FROM public."AllWind"
-- ),

```

```

-- (SELECT "Gladstone WS0" FROM public."AllWind" AS "Gladstone Winds km/h" UNION ALL
-- SELECT "Gladstone WS1" FROM public."AllWind" UNION ALL
-- SELECT "Gladstone WS2" FROM public."AllWind" UNION ALL
-- SELECT "Gladstone WS3" FROM public."AllWind" UNION ALL
-- SELECT "Gladstone WS4" FROM public."AllWind"
-- )
-- FROM public."AllWind"
-- --ORDER BY "DT ID" ASC;
-- --SELECT * FROM public."AllWind";

-- #####
-- 33. Generating final Wind Speed composite, and renamed:
-- #####
DROP TABLE IF EXISTS "WDTrue" CASCADE;
CREATE TABLE "WDTrue" AS
(SELECT "DTIDA" AS "Datetime ID", "Brisbane WS0" AS "Brisbane Windspeed", "Gladstone
WS0" AS "Gladstone Windspeed", "Date (YYYY/MM/DD)", "Hour (24:00)"
FROM public."AllWind");
INSERT INTO "WDTrue"("Datetime ID", "Brisbane Windspeed", "Gladstone Windspeed", "Date
(YYYY/MM/DD)", "Hour (24:00)")
SELECT "DTIDB" AS "Datetime ID", "Brisbane WS1" AS "Brisbane Windspeed", "Gladstone
WS1" AS "Gladstone Windspeed", "Date (YYYY/MM/DD)", "Hour (24:00)" FROM
public."AllWind";

INSERT INTO "WDTrue"("Datetime ID", "Brisbane Windspeed", "Gladstone Windspeed", "Date
(YYYY/MM/DD)", "Hour (24:00)")
SELECT "DTIDC" AS "Datetime ID", "Brisbane WS2" AS "Brisbane Windspeed", "Gladstone
WS2" AS "Gladstone Windspeed", "Date (YYYY/MM/DD)", "Hour (24:00)" FROM
public."AllWind";

INSERT INTO "WDTrue"("Datetime ID", "Brisbane Windspeed", "Gladstone Windspeed", "Date
(YYYY/MM/DD)", "Hour (24:00)")
SELECT "DTIDD" AS "Datetime ID", "Brisbane WS3" AS "Brisbane Windspeed", "Gladstone
WS3" AS "Gladstone Windspeed", "Date (YYYY/MM/DD)", "Hour (24:00)" FROM
public."AllWind";

INSERT INTO "WDTrue"("Datetime ID", "Brisbane Windspeed", "Gladstone Windspeed", "Date
(YYYY/MM/DD)", "Hour (24:00)")
SELECT "DTIDE" AS "Datetime ID", "Brisbane WS4" AS "Brisbane Windspeed", "Gladstone
WS4" AS "Gladstone Windspeed", "Date (YYYY/MM/DD)", "Hour (24:00)" FROM
public."AllWind";

SELECT * FROM "WDTrue" ORDER BY "Datetime ID";

```



```

-- #####
-- NOTE: TYPOS OF METEOROLOGICAL AND ASTRONOMICAL WERE FIXED BY
-- POSTGRESQL RENAMING
-- #####
-- 34. Query 3: Monthly/Seasonal Metrics for Metero/Astronomical Seasons
-- #####
-- DROP TABLE IF EXISTS "Query: Monthly/Seasonal Temperature Metrics" CASCADE;

-- CREATE TABLE "Query: Monthly/Seasonal Temperature Metrics" AS(
-- SELECT public."Query: Athmosphere metrics"."Datetime ID" AS "Date ID", -- is ID value
-- ROW_NUMBER() OVER (ORDER BY public."Query: Athmosphere metrics"."Datetime ID"
-- ASC)
-- public."Query: Athmosphere metrics"."Date (YYYY/MM/DD)" AS "Date (YYYY/MM/DD)",
-- public."Query: Athmosphere metrics"."Hour (24:00)" AS "Hour (24:00)",
-- TO_CHAR(public."Query: Athmosphere metrics"."Date (YYYY/MM/DD)", 'Month') AS
-- "Month",
-- EXTRACT(month FROM public."Query: Athmosphere metrics"."Date (YYYY/MM/DD)") AS
-- "Month Value",
-- CASE
--     WHEN EXTRACT(month FROM public."Query: Athmosphere metrics"."Date
-- (YYYY/MM/DD)") IN(12,1,2) THEN 'Summer'
--     WHEN EXTRACT(month FROM public."Query: Athmosphere metrics"."Date
-- (YYYY/MM/DD)") IN(3,4,5) THEN 'Autumn'
--     WHEN EXTRACT(month FROM public."Query: Athmosphere metrics"."Date
-- (YYYY/MM/DD)") IN(6,7,8) THEN 'Winter'
--     ELSE 'Spring' END "Season (Meterological)",
-- CASE
--     WHEN EXTRACT(month FROM public."Query: Athmosphere metrics"."Date
-- (YYYY/MM/DD)") IN(12,1,2) THEN 1
--     WHEN EXTRACT(month FROM public."Query: Athmosphere metrics"."Date
-- (YYYY/MM/DD)") IN(3,4,5) THEN 2
--     WHEN EXTRACT(month FROM public."Query: Athmosphere metrics"."Date
-- (YYYY/MM/DD)") IN(6,7,8) THEN 3
--     ELSE 4 END "Season (Meterological) Value",
-- CASE
--     WHEN public."Query: Athmosphere metrics"."Datetime ID" < 1899 OR public."Query:
-- Athmosphere metrics"."Datetime ID" >= 8529 THEN 'Summer'
--     WHEN public."Query: Athmosphere metrics"."Datetime ID" >= 1899 AND public."Query:
-- Athmosphere metrics"."Datetime ID" < 4124 THEN 'Autumn'
--     WHEN public."Query: Athmosphere metrics"."Datetime ID" >= 4124 AND public."Query:
-- Athmosphere metrics"."Datetime ID" < 6372 THEN 'Winter'
--     WHEN public."Query: Athmosphere metrics"."Datetime ID" >= 6372 AND public."Query:
-- Athmosphere metrics"."Datetime ID" < 8529 THEN 'Spring' END "Season (Astronomical)",
-- CASE
--     WHEN public."Query: Athmosphere metrics"."Datetime ID" < 1899 OR public."Query:
-- Athmosphere metrics"."Datetime ID" >= 8529 THEN 1
--     WHEN public."Query: Athmosphere metrics"."Datetime ID" >= 1899 AND public."Query:
-- Athmosphere metrics"."Datetime ID" < 4124 THEN 2
--     WHEN public."Query: Athmosphere metrics"."Datetime ID" >= 4124 AND public."Query:
-- Athmosphere metrics"."Datetime ID" < 6372 THEN 3
--     WHEN public."Query: Athmosphere metrics"."Datetime ID" >= 6372 AND public."Query:
-- Athmosphere metrics"."Datetime ID" < 8529 THEN 4 END "Season (Astronomical) Value",

```

```
-- public."Query: Athmosphere metrics"."brisbane humidity (%)" AS "Brisbane Humidity (%)",
-- public."Query: Athmosphere metrics"."gladstone humidity (%)" AS "Gladstone Humidity (%)",

-- public."Query: Athmosphere metrics"."brisbane rainfall (mm)" AS "Brisbane Rainfall (mm)",
-- public."Query: Athmosphere metrics"."gladstone rainfall (mm)" AS "Gladstone Rainfall (mm)",

-- public."Query: Athmosphere metrics"."brisbane temperature (degC)" AS "Brisbane Temperature
(degC)",
-- public."Query: Athmosphere metrics"."gladstone temperature (degC)" AS "Gladstone
Temperature (degC)",

-- public."Query: Athmosphere metrics"."brisbane Barometric Pressure (hPa)" AS "Brisbane
Barometric Pressure (hPa)",
-- public."Query: Athmosphere metrics"."gladstone Barometric Pressure (hPa)" AS "Gladstone
Barometric Pressure (hPa)"

-- FROM public."Query: Athmosphere metrics"
-- ORDER BY "Date ID" ASC);

-- SELECT * FROM public."Query: Monthly/Seasonal Temperature Metrics" ORDER BY "Date
ID" ASC
```

```

-- #####
--35. SEASONS METEROLOGICAL
-- #####
-- DROP TABLE IF EXISTS "SubQuery: Season Metrics (Meterological)" CASCADE;

-- CREATE TABLE "SubQuery: Season Metrics (Meterological)" AS(
-- SELECT
-- ROW_NUMBER() OVER (ORDER BY public."Query: Monthly/Seasonal Temperature
Metrics"."Season (Meterological)" ASC) AS "Row ID", -- is ID value
-- public."Query: Monthly/Seasonal Temperature Metrics"."Season (Meterological)" AS
"Meterological Season",

-- AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Brisbane Humidity (%)") AS
"Metero Brisbane Humidity (%)",
-- AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Gladstone Humidity (%)")AS
"Metero Gladstone Humidity (%)",

-- AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Brisbane Rainfall (mm)") AS
"Metero Brisbane Rainfall (mm)",
-- AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Gladstone Rainfall (mm)") AS
"Metero Gladstone Rainfall (mm)",

-- AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Brisbane Temperature (degC)")
AS "Metero Brisbane Temperature (degC)",
-- AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Gladstone Temperature (degC)")
AS "Metero Gladstone Temperature (degC)",

-- AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Brisbane Barometric Pressure
(hPa)") AS "Metero Brisbane Barometric Pressure (hPa)",
-- AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Gladstone Barometric Pressure
(hPa)") AS "Metero Gladstone Barometric Pressure (hPa)"

-- FROM public."Query: Monthly/Seasonal Temperature Metrics"
-- GROUP BY public."Query: Monthly/Seasonal Temperature Metrics"."Season (Meterological)"
-- ORDER BY "Row ID" ASC);

-- SELECT * FROM "SubQuery: Season Metrics (Meterological)" ORDER BY "Row ID" ASC;

```

```

-- #####
-- 36. SEASONS ASTRONOMICAL
-- #####
DROP TABLE IF EXISTS "SubQuery: Season Metrics (Astronomical)" CASCADE;

CREATE TABLE "SubQuery: Season Metrics (Astronomical)" AS(
SELECT
ROW_NUMBER() OVER (ORDER BY public."Query: Monthly/Seasonal Temperature
Metrics"."Season (Astronomical)" ASC) AS "Row ID", -- is ID value
public."Query: Monthly/Seasonal Temperature Metrics"."Season (Astronomical)" AS
"Astronomical Season",

AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Brisbane Humidity (%)") AS
"Astro Brisbane Humidity (%)",
AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Gladstone Humidity (%)")AS
"Astro Gladstone Humidity (%)",

AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Brisbane Rainfall (mm)") AS
"Astro Brisbane Rainfall (mm)",
AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Gladstone Rainfall (mm)") AS
"Astro Gladstone Rainfall (mm)",

AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Brisbane Temperature (degC)") AS
"Astro Brisbane Temperature (degC)",
AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Gladstone Temperature (degC)")
AS "Astro Gladstone Temperature (degC)",

AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Brisbane Barometric Pressure
(hPa)") AS "Astro Brisbane Barometric Pressure (hPa)",
AVG(public."Query: Monthly/Seasonal Temperature Metrics"."Gladstone Barometric Pressure
(hPa)") AS "Astro Gladstone Barometric Pressure (hPa)"

FROM public."Query: Monthly/Seasonal Temperature Metrics"
GROUP BY public."Query: Monthly/Seasonal Temperature Metrics"."Season (Astronomical)"
ORDER BY "Row ID" ASC);

SELECT * FROM "SubQuery: Season Metrics (Astronomical)" ORDER BY "Row ID" ASC;

-- #####
-- ADDITIONAL PROCEDURES HERE:
-- #####

```

```

-- #####
-- 37 Air Molecule Measurements
-- #####
SET TIME ZONE 'Australia/Brisbane';
DROP TABLE IF EXISTS "Query: Air Molecule Measurements T" CASCADE;
CREATE TABLE "Query: Air Molecule Measurements T" AS
(SELECT
  "Datetime ID",
  TO_TIMESTAMP( to_char("Date (YYYY/MM/DD)"::date, 'YYYY-MM-DD')||' '|| to_char("Hour
(24:00)"::time without time zone, 'HH24:MI'), 'YYYY-MM-DD HH24:MI TZH')::timestamp AS
  "Date-Time",
  "Date (YYYY/MM/DD)",
  "Hour (24:00)",
  "brisbane nitrogen oxide (ppm)",
  "brisbane nitrogen dioxide (ppm)",
  "brisbane nitrogen oxides (ppm)",
  "brisbane carbon monoxide (ppm)",
  "gladstone nitrogen oxide (ppm)",
  "gladstone nitrogen dioxide (ppm)",
  "gladstone nitrogen oxides (ppm)",
  "gladstone sulfur dioxide (ppm)",
  "gladstone ozone (ppm)",
  "gladstone xylenes (ppb)"
FROM public."Query: Air Molecule Measurements" ORDER BY "Datetime ID" ASC);

SELECT * FROM public."Query: Air Molecule Measurements T" ORDER BY "Datetime ID"
ASC;

-- #####
-- 38 Air Quality Data
-- #####
SET TIME ZONE 'Australia/Brisbane';
DROP TABLE IF EXISTS public."Query: Air Quality Data T" CASCADE;
CREATE TABLE "Query: Air Quality Data T" AS
(SELECT
  "Datetime ID",
  TO_TIMESTAMP( to_char("Date (YYYY/MM/DD)"::date, 'YYYY-MM-DD')||' '|| to_char("Hour
(24:00)"::time without time zone, 'HH24:MI'), 'YYYY-MM-DD HH24:MI TZH')::timestamp AS
  "Date-Time",
  "Date (YYYY/MM/DD)",
  "Hour (24:00)",
    "brisbane PM10 (ug/m3)",
    "brisbane PM2.5 (ug/m3)",
    "brisbane VRPs (Mm-1)",
    "gladstone PM10 (ug/m3)",
    "gladstone PM2.5 (ug/m3)",
    "gladstone VRPs (Mm-1)"
FROM public."Query: Air Quality Data" ORDER BY "Datetime ID" ASC);
SELECT * FROM public."Query: Air Quality Data T" ORDER BY "Datetime ID" ASC;

```

```

-- #####
-- 39 Atmospheric metrics
-- #####
SET TIME ZONE 'Australia/Brisbane';
DROP TABLE IF EXISTS "Query: Atmosphere metrics T" CASCADE;
CREATE TABLE "Query: Atmosphere metrics T" AS
(SELECT
  "Datetime ID",
  TO_TIMESTAMP( to_char("Date (YYYY/MM/DD) "::date, 'YYYY-MM-DD') || ' ' || to_char("Hour
(24:00) "::time without time zone, 'HH24:MI'), 'YYYY-MM-DD HH24:MI TZh')::timestamp AS
  "Date-Time",
  "Date (YYYY/MM/DD)",
  "Hour (24:00)",
  "brisbane humidity (%)",
  "brisbane rainfall (mm)",
  "brisbane temperature (degC)",
  "brisbane Barometric Pressure (hPa)",
  "gladstone humidity (%)",
  "gladstone rainfall (mm)",
  "gladstone temperature (degC)",
  "gladstone Barometric Pressure (hPa)"

FROM public."Query: Atmosphere metrics" ORDER BY "Datetime ID" ASC);

SELECT * FROM public."Query: Atmosphere metrics T" ORDER BY "Datetime ID" ASC;

-- #####
-- 40 Wind Speed Deviations
-- #####
SET TIME ZONE 'Australia/Brisbane';
DROP TABLE IF EXISTS public."Query: Wind Speed Deviations T" CASCADE;
CREATE TABLE "Query: Wind Speed Deviations T" AS
(SELECT
  ROW_NUMBER() OVER (ORDER BY public."Query: Wind Speed Deviations"."Date ID"
ASC) AS "Datetime ID",
  TO_TIMESTAMP( to_char("Date (YYYY/MM/DD) "::date, 'YYYY-MM-DD') || ' ' ||
to_char("Hour (24:00) "::time without time zone, 'HH24:MI'), 'YYYY-MM-DD HH24:MI
TZh')::timestamp AS "Date-Time",
  "Date (YYYY/MM/DD)",
  "Hour (24:00)",
  "Brisbane Wind Direction (degN)",
  "Gladstone Wind Direction (degN)",
  "Brisbane Wind Speed Km/h",
  "Gladstone Wind Speed Km/h",
  "Brisbane Wind Std Deviation Km/h",
  "Gladstone Wind Std Deviation Km/h"

FROM public."Query: Wind Speed Deviations" ORDER BY "Datetime ID" ASC);
SELECT * FROM "Query: Wind Speed Deviations T";

```

```

-- #####
-- 41 Air Quality Guidelines
-- #####
SET TIME ZONE 'Australia/Brisbane';
DROP TABLE IF EXISTS public."Query: Air Quality Guidelines T" CASCADE;

CREATE TABLE public."Query: Air Quality Guidelines T" AS
(SELECT *
FROM public."Query: Air Quality Guidelines" ORDER BY "Date ID" ASC);

SELECT * FROM public."Query: Air Quality Guidelines" ORDER BY "Date ID" ASC;

-- #####
-- 42 Monthly Seasonal Temperature Metrics
-- #####
SET TIME ZONE 'Australia/Brisbane';
DROP TABLE IF EXISTS public."Query: Monthly/Seasonal Temperature Metrics T" CASCADE;

CREATE TABLE "Query: Monthly/Seasonal Temperature Metrics T" AS
(SELECT
  "Date ID" AS "Datetime ID",
  TO_TIMESTAMP( to_char("Date (YYYY/MM/DD) "::date, 'YYYY-MM-DD') || ' ' || to_char("Hour
(24:00) "::time without time zone, 'HH24:MI'), 'YYYY-MM-DD HH24:MI TZH')::timestamp AS
  "Date-Time",
  "Date (YYYY/MM/DD)",
  "Hour (24:00)",
  "Month",
  "Month Value",
  "Season (Meterological)",
  "Season (Meterological) Value",
  "Season (Astronomical)",
  "Season (Astronomical) Value",
  "Brisbane Humidity (%)",
  "Gladstone Humidity (%)",
  "Brisbane Rainfall (mm)",
  "Gladstone Rainfall (mm)",
  "Brisbane Temperature (degC)",
  "Gladstone Temperature (degC)",
  "Brisbane Barometric Pressure (hPa)",
  "Gladstone Barometric Pressure (hPa)"
FROM public."Query: Monthly/Seasonal Temperature Metrics" ORDER BY "Date ID" ASC);

SELECT * FROM public."Query: Monthly/Seasonal Temperature Metrics T" ORDER BY
"Datetime ID" ASC;

```

```

-- #####
-- 43 Monthly Seasonal Temperature Metrics 2
-- #####
SET TIME ZONE 'Australia/Brisbane';
DROP TABLE IF EXISTS public."Query: Monthly/Seasonal Temperature Metrics T" CASCADE;

CREATE TABLE "Query: Monthly/Seasonal Temperature Metrics T" AS
(SELECT
  "Date ID" AS "Datetime ID",
  TO_TIMESTAMP( to_char("Date (YYYY/MM/DD)"::date, 'YYYY-MM-DD')||' '|| to_char("Hour
(24:00)"::time without time zone, 'HH24:MI'), 'YYYY-MM-DD HH24:MI TZH')::timestamp AS
  "Date-Time",
  "Date (YYYY/MM/DD)",
  "Hour (24:00)",
  "Month",
  "Month Value",
  "Season (Meterological)" AS "Season (Meteorological)",
  "Season (Meterological) Value" AS "Season (Meteorological) Value",
  "Season (Astronomical)",
  "Season (Astronomical) Value",
  "Brisbane Humidity (%)",
  "Gladstone Humidity (%)",
  "Brisbane Rainfall (mm)",
  "Gladstone Rainfall (mm)",
  "Brisbane Temperature (degC)",
  "Gladstone Temperature (degC)",
  "Brisbane Barometric Pressure (hPa)",
  "Gladstone Barometric Pressure (hPa)"
FROM public."Query: Monthly/Seasonal Temperature Metrics" ORDER BY "Date ID" ASC);

SELECT * FROM public."Query: Monthly/Seasonal Temperature Metrics T" ORDER BY
"Datetime ID" ASC;

```



```

-- #####
-- 44. Substitution order ver2 for Brisbane using COALESCE
-- #####
SELECT
coalesce(ta.datetime_id, tb.datetime_id, tc.datetime_id) as datetime_id,
coalesce(ta.datevalue, tb.datevalue, tc.datevalue) as datevalue,
coalesce(ta.timevalue, tb.timevalue, tc.timevalue) as timevalue,
coalesce(ta."wind_direction_(degN)", tb."wind_direction_(degN)", tc."wind_direction_(degN)") as
"wind_direction_(degN)",
coalesce(ta."wind_speed_(m/s)", tb."wind_speed_(m/s)", tc."wind_speed_(m/s)") as
"wind_speed_(m/s)",
coalesce(ta."wind_deviation_(deg)", tb."wind_deviation_(deg)", tc."wind_deviation_(deg)") as
"wind_deviation_(deg)",
coalesce(ta."wind_speed_deviation_(m/s)", tb."wind_speed_deviation_(m/s)",
tc."wind_speed_deviation_(m/s)") as "wind_speed_deviation_(m/s)",
coalesce(ta."wind_temperature_(degC)", tb."wind_temperature_(degC)",
tc."wind_temperature_(degC)") as "wind_temperature_(degC)",
coalesce(ta."humidity_(%)", tc."humidity_(%)") as "humidity_(%)",
coalesce(ta."rainfall_(mm)", tc."rainfall_(mm)") as "rainfall_(mm)",
coalesce(ta."barometric_pressure_(hPa)", tc."barometric_pressure_(hPa)") as
"barometric_pressure_(hPa)",
coalesce(ta."PM10_(ug/m3)", tb."PM10_(ug/m3)", tc."PM10_(ug/m3)") as "PM10_(ug/m3)",
coalesce(ta."PM2.5_(ug/m3)", tb."PM2.5_(ug/m3)", tc."PM2.5_(ug/m3)") as "PM2.5_(ug/m3)",
coalesce(ta."visibility_reducing_particles_(Mm-1)") as "visibility_reducing_particles_(Mm-1)",
coalesce(tb."nitrogen_oxide_(ppm)", tc."nitrogen_oxide_(ppm)") as "nitrogen_oxide_(ppm)",
coalesce(tb."nitrogen_dioxide_(ppm)", tc."nitrogen_dioxide_(ppm)") as "nitrogen_dioxide_(ppm)",
coalesce(tb."nitrogen_oxides_(ppm)", tc."nitrogen_oxides_(ppm)") as "nitrogen_oxides_(ppm)",
coalesce(tb."carbon_monoxide_(ppm)", tc."carbon_monoxide_(ppm)") as
"carbon_monoxide_(ppm)"
FROM
public.brisbane_cbd_air_quality_raw AS ta,
public.south_brisbane_air_quality_raw AS tb,
public.woolloomgabbba_air_quality_raw AS tc
WHERE
ta.datetime_id = tb.datetime_id
AND
tb.datetime_id = tc.datetime_id
ORDER BY datetime_id ASC

-- Repeat w/ wrapped table
CREATE TABLE public."Brisbane_Fixed_table_1" AS (SELECT
coalesce(ta.datetime_id, tb.datetime_id, tc.datetime_id) as datetime_id,
coalesce(ta.datevalue, tb.datevalue, tc.datevalue) as datevalue,
coalesce(ta.timevalue, tb.timevalue, tc.timevalue) as timevalue,
coalesce(ta."wind_direction_(degN)", tb."wind_direction_(degN)", tc."wind_direction_(degN)") as
"wind_direction_(degN)",
coalesce(ta."wind_speed_(m/s)", tb."wind_speed_(m/s)", tc."wind_speed_(m/s)") as
"wind_speed_(m/s)",
coalesce(ta."wind_deviation_(deg)", tb."wind_deviation_(deg)", tc."wind_deviation_(deg)") as
"wind_deviation_(deg)",
coalesce(ta."wind_speed_deviation_(m/s)", tb."wind_speed_deviation_(m/s)",
tc."wind_speed_deviation_(m/s)") as "wind_speed_deviation_(m/s)",

```

```

coalesce(ta."wind_temperature_(degC)", tb."wind_temperature_(degC)",
tc."wind_temperature_(degC)") as "wind_temperature_(degC)",
coalesce(ta."humidity_(%)", tc."humidity_(%)") as "humidity_(%)",
coalesce(ta."rainfall_(mm)", tc."rainfall_(mm)") as "rainfall_(mm)",
coalesce(ta."barometric_pressure_(hPa)", tc."barometric_pressure_(hPa)") as
"barometric_pressure_(hPa)",
coalesce(ta."PM10_(ug/m3)", tb."PM10_(ug/m3)", tc."PM10_(ug/m3)") as "PM10_(ug/m3)",
coalesce(ta."PM2.5_(ug/m3)", tb."PM2.5_(ug/m3)", tc."PM2.5_(ug/m3)") as "PM2.5_(ug/m3)",
coalesce(ta."visibility_reducing_particles_(Mm-1)") as "visibility_reducing_particles_(Mm-1)",
coalesce(tb."nitrogen_oxide_(ppm)", tc."nitrogen_oxide_(ppm)") as "nitrogen_oxide_(ppm)",
coalesce(tb."nitrogen_dioxide_(ppm)", tc."nitrogen_dioxide_(ppm)") as "nitrogen_dioxide_(ppm)",
coalesce(tb."nitrogen_oxides_(ppm)", tc."nitrogen_oxides_(ppm)") as "nitrogen_oxides_(ppm)",
coalesce(tb."carbon_monoxide_(ppm)", tc."carbon_monoxide_(ppm)") as
"carbon_monoxide_(ppm)"
FROM
public.brisbane_cbd_air_quality_raw AS ta,
public.south_brisbane_air_quality_raw AS tb,
public.woolloomgabbba_air_quality_raw AS tc
WHERE
ta.datetime_id = tb.datetime_id
AND
tb.datetime_id = tc.datetime_id
ORDER BY datetime_id ASC);

-- #####
-- 45. Brisbane NULL checker final
-- #####
-- Null checker
-- SELECT *
-- FROM public."Brisbane_Fixed_table_1"
-- WHERE "wind_direction_(degN)" IS NULL
-- OR "wind_speed_(m/s)" IS NULL
-- OR "wind_deviation_(deg)" IS NULL
-- OR "wind_speed_deviation_(m/s)" IS NULL
-- OR "wind_temperature_(degC)" IS NULL
-- OR "humidity_(%)" IS NULL
-- OR "rainfall_(mm)" IS NULL
-- OR "barometric_pressure_(hPa)" IS NULL
-- OR "PM10_(ug/m3)" IS NULL
-- OR "PM2.5_(ug/m3)" IS NULL
-- OR "nitrogen_oxide_(ppm)" IS NULL
-- OR "nitrogen_dioxide_(ppm)" IS NULL
-- OR "nitrogen_oxides_(ppm)" IS NULL
-- OR "carbon_monoxide_(ppm)" IS NULL
-- --OR "visibility_reducing_particles_(Mm-1)" IS NULL
-- OR datetime_id IN(1342,1343,1344,1345)
-- ORDER BY datetime_id ASC;

```