Aggregate Table Analytics Documentation:

Description

In PRJ1 the data stage was set with the original tables from the LA Crimes Data. In PRJ4 that same data is taken and aggregated on the day level to allow for proper application of windows function. This generated rolling metrics on partitions relating to date intervals to expand on general, weapons based, violent and domestic violence crimes. Data was populated into the following three steps for all aggregate tables. First is creating the table, second is creating the CTE's to transform the data and third is inserting the data into the table created in step one.

Table Creation

- The table creation step followed standard SQL format for all tables and included the same windows function based fields that were slightly adjusted for each specific section's unique data.
- The primary difference between tables aside from topic based differences is that general and weapons aggregate tables are done using windows functions on crime_tally and crime_case_density. The arrest, violent crime and domestic violence aggregate tables use custom made fields to reflect rate based windows functions.
- The field names are not abbreviated to give a clear explanation of the purpose of the values. Abbreviation is possible but for the sake of digesting the code they are descriptive.

General

```
CREATE TABLE overall_crimes(
    date_rptd date,
    month_year varchar(14),
    month varchar(9),
    year integer,
    crime_tally numeric,
    crime_case_density numeric,
    rolling_crime_tally numeric,
    rolling_crime_tally_bymonth numeric,
    rolling_crime_tally_by_month_year numeric,
    rolling_crime_tally_by_year numeric,
    crime_case_density_bymonth numeric,
    crime_case_density_by_month_year numeric,
    crime_case_density_by_year numeric,
    avg_rolling_crime_tally numeric,
    avg_crime_tally_by_month numeric,
    avg_crime_tally_by_month_year numeric,
    avg_crime_tally_by_year numeric,
    avg_crime_case_density numeric,
    avg_crime_case_density_by_month numeric,
    avg_crime_case_density_by_month_year numeric,
    avg_crime_case_density_by_year numeric,
    "90day_AVG_rolling_crime_tally" numeric,
    "90day_AVG_rolling_crime_case_density" numeric
```

```
);
```

- The general aggregate table is created to get the overall view of how crime count and density move and fluctuate within multiple date intervals.

Weapons

```
CREATE TABLE w_overall_crimes (
    w_date DATE,
    w_month_year VARCHAR(14),
    w_month VARCHAR(9),
    w_year INTEGER,
    weapon_type VARCHAR(50),
    weapon_description VARCHAR(255),
    w_crime_tally NUMERIC,
    w_rolling_crime_tally_for_weapon NUMERIC,
    w_rolling_crime_tally_bymonth_for_weapon NUMERIC,
    w_rolling_crime_tally_by_month_year_for_weapon NUMERIC,
    w_rolling_crime_tally_by_year_for_weapon NUMERIC,
    w_rolling_crime_case_density_for_weapon NUMERIC,
    w_crime_case_density_bymonth_for_weapon NUMERIC,
    w_crime_case_density_by_month_year_for_weapon NUMERIC,
    w_crime_case_density_by_year_for_weapon NUMERIC,
    w_rolling_crime_tally_for_weapon_type NUMERIC,
    w_rolling_crime_tally_bymonth_for_weapon_type NUMERIC,
    w_rolling_crime_tally_by_month_year_for_weapon_type NUMERIC,
    w_rolling_crime_tally_by_year_for_weapon_type NUMERIC,
    w_rolling_crime_case_density_for_weapon_type NUMERIC,
    w_crime_case_density_bymonth_for_weapon_type NUMERIC,
    w_crime_case_density_by_month_year_for_weapon_type NUMERIC,
    w_crime_case_density_by_year_for_weapon_type NUMERIC,
    w_avg_rolling_crime_tally_for_weapon NUMERIC,
    w_avg_rolling_crime_tally_bymonth_for_weapon NUMERIC,
    w_avg_rolling_crime_tally_by_month_year_for_weapon NUMERIC,
    w_avg_rolling_crime_tally_by_year_for_weapon NUMERIC,
    w_avg_crime_case_density_for_weapon NUMERIC,
    w_avg_crime_case_density_bymonth_for_weapon NUMERIC,
    w_avg_crime_case_density_by_month_year_for_weapon NUMERIC,
    w_avg_crime_case_density_by_year_for_weapon NUMERIC,
    w_90day_AVG_rolling_crime_tally_for_weapon NUMERIC,
    w_90day_AVG_rolling_crime_case_density_for_weapon NUMERIC,
    w_avg_rolling_crime_tally_for_weapon_type NUMERIC,
    w_avg_rolling_crime_tally_bymonth_for_weapon_type NUMERIC,
    w_avg_rolling_crime_tally_by_month_year_for_weapon_type NUMERIC,
    w_avg_rolling_crime_tally_by_year_for_weapon_type NUMERIC,
    w_avg_crime_case_density_for_weapon_type NUMERIC,
    w_avg_crime_case_density_bymonth_for_weapon_type NUMERIC,
    w_avg_crime_case_density_by_month_year_for_weapon_type NUMERIC,
    w_avg_crime_case_density_by_year_for_weapon_type NUMERIC
);
```

- The weapons aggregate table was created to get a better understanding of the individual weapon and weapon type trends within crimes. Tracking what crimes include weapons give a clear picture of the level of danger a single crime in Los Angeles could involve.

Arrests

```
CREATE TABLE arrest_agg (
    report_date DATE,
    arrests INT,
    arrest_rate NUMERIC,
    rolling_arrests INT,
    rolling_arrests_bymonth INT,
    rolling_arrests_by_month_year INT,
    rolling_arrests_by_year INT,
    avg_rolling_arrests NUMERIC,
    avg_arrests_by_month NUMERIC,
    avg_arrests_by_month_year NUMERIC,
    avg_arrests_by_year NUMERIC,
    avg_arrest_rate NUMERIC,
    avg_arrest_rate_by_month NUMERIC,
    avg_arrest_rate_by_month_year NUMERIC,
    avg_arrest_rate_by_year NUMERIC,
    "90day_avg_rolling_arrests" NUMERIC,
    "90day_avg_rolling_arrest_rate" NUMERIC
);
```

- Tracking Arrests give a good idea on crime severity and the level of efficiency the LAPD is running at. This can create a visual of both the seasonality of crime and performance for the LAPD.

Violent Crime

```
CREATE TABLE violent_crimes_agg (
    report_date DATE,
    violent_crimes INT,
    violent_crime_rate NUMERIC,
    rolling_violent_crimes INT,
    rolling_violent_crimes_bymonth INT,
    rolling_violent_crimes_by_month_year INT,
    rolling_violent_crimes_by_year INT,
    avg_rolling_violent_crimes NUMERIC,
    avg_violent_crimes_by_month NUMERIC,
    avg_violent_crimes_by_month_year NUMERIC,
    avg_violent_crimes_by_year NUMERIC,
    avg_violent_crime_rate NUMERIC,
    avg_violent_crime_rate_by_month NUMERIC,
    avg_violent_crime_rate_by_month_year NUMERIC,
    avg_violent_crime_rate_by_year NUMERIC,
    "90day_avg_rolling_violent_crimes" NUMERIC,
    "90day_avg_rolling_violent_crime_rate" NUMERIC
);
```

- Violent crime monitoring is another way to depict the level of harm and severity of crime that is occuring in the Los Angeles area. Not only giving the public a better view of general crime but also allowing the police force to anticipate the likelihood they will be dealing with a violent offender.

Domestic Violence

```
CREATE TABLE domestic_violence_agg (
    report_date DATE,
    domestic_violence_count INT,
    domestic_violence_rate NUMERIC,
    rolling_domestic_violence INT,
    rolling_domestic_violence_bymonth INT,
    rolling_domestic_violence_by_month_year INT,
    rolling_domestic_violence_by_year INT,
    avg_rolling_domestic_violence NUMERIC,
    avg_domestic_violence_by_month NUMERIC,
    avg_domestic_violence_by_month_year NUMERIC,
    avg_domestic_violence_by_year NUMERIC,
    avg_domestic_violence_rate NUMERIC,
    avg_domestic_violence_rate_by_month NUMERIC,
    avg_domestic_violence_rate_by_month_year NUMERIC,
    avg_domestic_violence_rate_by_year NUMERIC,
    "90day_avg_rolling_domestic_violence" NUMERIC,
    "90day_avg_rolling_domestic_violence_rate" NUMERIC
);
```

- Domestic Violence crimes are another standard way to track the state of crime in a location. By tracking domestic violence it can give a great view of how safe the average household is.

CTE's

- Every set of CTE's involves taking the original data and aggregating it to the day level. This is followed by the application of the windows functions on day to year date intervals of both rolling SUMs and AVGs.

General

```
WITH OC AS (
    SELECT
        d.date::date AS date,
        d.month_year::varchar(14),
        d.month::varchar(9),
        d.year::integer,
        c.crime_tally::numeric AS crime_tally,
        c.crime_case_density1::numeric AS crime_case_density1,
        SUM(c.crime_tally) OVER(ORDER BY d.date) AS rolling_crime_tally,
        SUM(c.crime_tally) OVER(PARTITION BY d.month ORDER BY d.date) AS
rolling_crime_tally_bymonth,
        SUM(c.crime_tally) OVER(PARTITION BY d.month_year ORDER BY d.date) AS
rolling_crime_tally_by_month_year,
        SUM(c.crime_tally) OVER(PARTITION BY d.year ORDER BY d.date) AS
rolling_crime_tally_by_year,
        SUM(c.crime_case_density1) OVER(ORDER BY d.date) AS
rolling_crime_case_density,
        SUM(c.crime_case_density1) OVER(PARTITION BY d.month ORDER BY d.date) AS
rolling_crime_case_density_bymonth,
```

```
        SUM(c.crime_case_density1) OVER(PARTITION BY d.month_year ORDER BY
d.date) AS rolling_crime_case_density_by_month_year,
        SUM(c.crime_case_density1) OVER(PARTITION BY d.year ORDER BY d.date) AS
rolling_crime_case_density_by_year,
        ROUND(AVG(c.crime_tally) OVER(ORDER BY d.date), 2) AS
avg_rolling_crime_tally,
        ROUND(AVG(c.crime_tally) OVER(PARTITION BY d.month ORDER BY d.date), 2)
AS avg_crime_tally_by_month,
        ROUND(AVG(c.crime_tally) OVER(PARTITION BY d.month_year ORDER BY d.date),
2) AS avg_crime_tally_by_month_year,
        ROUND(AVG(c.crime_tally) OVER(PARTITION BY d.year ORDER BY d.date), 2) AS
avg_crime_tally_by_year,
        ROUND(AVG(c.crime_case_density1) OVER(ORDER BY d.date), 2) AS
avg_crime_case_density,
        ROUND(AVG(c.crime_case_density1) OVER(PARTITION BY d.month ORDER BY
d.date), 2) AS avg_crime_case_density_by_month,
        ROUND(AVG(c.crime_case_density1) OVER(PARTITION BY d.month_year ORDER BY
d.date), 2) AS avg_crime_case_density_by_month_year,
        ROUND(AVG(c.crime_case_density1) OVER(PARTITION BY d.year ORDER BY
d.date), 2) AS avg_crime_case_density_by_year,
        ROUND(AVG(c.crime_tally) OVER(ORDER BY d.date ROWS BETWEEN 89 PRECEDING
AND CURRENT ROW), 2) AS "90day_AVG_rolling_crime_tally",
        ROUND(AVG(c.crime_case_density1) OVER(ORDER BY d.date ROWS BETWEEN 89
PRECEDING AND CURRENT ROW), 2) AS "90day_AVG_rolling_crime_case_density"
    FROM
    (
        SELECT
            date_rptd,
            SUM(crime_tally) AS crime_tally,
            SUM(crime_case_density) AS crime_case_density1
        FROM crime_codes
        GROUP BY date_rptd
    ) c
    LEFT JOIN date_table d ON c.date_rptd = d.date
)
```

Weapons

```
WITH WOC AS (
    SELECT
        lc.date_rptd :: DATE AS report_date,
        w.weapon_type,
        w.weapon_description,
        SUM(cc.crime_tally) AS crime_tally_weapon,
        SUM(cc.crime_case_density) AS crime_case_density1_weapon
    FROM la_crimes lc
    LEFT JOIN crime_codes cc ON cc.division_records_num = lc.division_records_num
    LEFT JOIN crime_weapons w ON w.weapon_used_code = lc.weapon_used_code
    GROUP BY lc.date_rptd AS report_date, w.weapon_type, w.weapon_description
),
WOC2 AS (
    SELECT
```

```
        date_rptd,
        weapon_type,
        weapon_description,
        crime_tally_weapon,
        crime_case_density1_weapon,
        SUM(crime_tally_weapon) OVER (PARTITION BY weapon_type,
weapon_description ORDER BY date_rptd) AS crime_tally_weapontype,
        SUM(crime_case_density1_weapon) OVER (PARTITION BY weapon_type,
weapon_description ORDER BY date_rptd) AS crime_case_density1_weapontype
    FROM WOC
),
WOC3 AS (
    SELECT
        d.date::date AS w_date,
        d.month_year::varchar(14) AS w_month_year,
        d.month::varchar(9) AS w_month,
        d.year::integer AS w_year,
        c.weapon_type,
        c.weapon_description,
        c.crime_tally_weapon AS w_crime_tally,
        SUM(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_description ORDER
BY d.date) AS w_rolling_crime_tally_for_weapon,
        SUM(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_description,
d.month ORDER BY d.date) AS w_rolling_crime_tally_bymonth_for_weapon,
        SUM(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_description,
d.month_year ORDER BY d.date) AS w_rolling_crime_tally_by_month_year_for_weapon,
        SUM(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_description, d.year
ORDER BY d.date) AS w_rolling_crime_tally_by_year_for_weapon,
        SUM(c.crime_case_density1_weapon) OVER (PARTITION BY c.weapon_description
ORDER BY d.date) AS w_rolling_crime_case_density_for_weapon,
        SUM(c.crime_case_density1_weapon) OVER (PARTITION BY
c.weapon_description, d.month ORDER BY d.date) AS
w_crime_case_density_bymonth_for_weapon,
        SUM(c.crime_case_density1_weapon) OVER (PARTITION BY
c.weapon_description, d.month_year ORDER BY d.date) AS
w_crime_case_density_by_month_year_for_weapon,
        SUM(c.crime_case_density1_weapon) OVER (PARTITION BY
c.weapon_description, d.year ORDER BY d.date) AS
w_crime_case_density_by_year_for_weapon,
        SUM(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_type ORDER BY
d.date) AS w_rolling_crime_tally_for_weapon_type,
        SUM(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_type, d.month ORDER
BY d.date) AS w_rolling_crime_tally_bymonth_for_weapon_type,
        SUM(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_type, d.month_year
ORDER BY d.date) AS w_rolling_crime_tally_by_month_year_for_weapon_type,
        SUM(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_type, d.year ORDER
BY d.date) AS w_rolling_crime_tally_by_year_for_weapon_type,
        SUM(c.crime_case_density1_weapon) OVER (PARTITION BY c.weapon_type ORDER
BY d.date) AS w_rolling_crime_case_density_for_weapon_type,
        SUM(c.crime_case_density1_weapon) OVER (PARTITION BY c.weapon_type,
d.month ORDER BY d.date) AS w_crime_case_density_bymonth_for_weapon_type,
```

```
        SUM(c.crime_case_density1_weapon) OVER (PARTITION BY c.weapon_type,
d.month_year ORDER BY d.date) AS
w_crime_case_density_by_month_year_for_weapon_type,
        SUM(c.crime_case_density1_weapon) OVER (PARTITION BY c.weapon_type,
d.year ORDER BY d.date) AS w_crime_case_density_by_year_for_weapon_type,
        ROUND(AVG(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_description
ORDER BY d.date), 2) AS w_avg_rolling_crime_tally_for_weapon,
        ROUND(AVG(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_description,
d.month ORDER BY d.date), 2) AS w_avg_rolling_crime_tally_bymonth_for_weapon,
        ROUND(AVG(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_description,
d.month_year ORDER BY d.date), 2) AS
w_avg_rolling_crime_tally_by_month_year_for_weapon,
        ROUND(AVG(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_description,
d.year ORDER BY d.date), 2) AS w_avg_rolling_crime_tally_by_year_for_weapon,
        ROUND(AVG(c.crime_case_density1_weapon) OVER (PARTITION BY
c.weapon_description ORDER BY d.date), 2) AS w_avg_crime_case_density_for_weapon,
        ROUND(AVG(c.crime_case_density1_weapon) OVER (PARTITION BY
c.weapon_description, d.month ORDER BY d.date), 2) AS
w_avg_crime_case_density_bymonth_for_weapon,
        ROUND(AVG(c.crime_case_density1_weapon) OVER (PARTITION BY
c.weapon_description, d.month_year ORDER BY d.date), 2) AS
w_avg_crime_case_density_by_month_year_for_weapon,
        ROUND(AVG(c.crime_case_density1_weapon) OVER (PARTITION BY
c.weapon_description, d.year ORDER BY d.date), 2) AS
w_avg_crime_case_density_by_year_for_weapon,
        ROUND(AVG(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_description
ORDER BY d.date ROWS BETWEEN 89 PRECEDING AND CURRENT ROW), 2) AS
w_90day_AVG_rolling_crime_tally_for_weapon,
        ROUND(AVG(c.crime_case_density1_weapon) OVER (PARTITION BY
c.weapon_description ORDER BY d.date ROWS BETWEEN 89 PRECEDING AND CURRENT ROW),
2) AS w_90day_AVG_rolling_crime_case_density_for_weapon,
        ROUND(AVG(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_type ORDER BY
d.date), 2) AS w_avg_rolling_crime_tally_for_weapon_type,
        ROUND(AVG(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_type, d.month
ORDER BY d.date), 2) AS w_avg_rolling_crime_tally_bymonth_for_weapon_type,
        ROUND(AVG(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_type,
d.month_year ORDER BY d.date), 2) AS
w_avg_rolling_crime_tally_by_month_year_for_weapon_type,
        ROUND(AVG(c.crime_tally_weapon) OVER (PARTITION BY c.weapon_type, d.year
ORDER BY d.date), 2) AS w_avg_rolling_crime_tally_by_year_for_weapon_type,
        ROUND(AVG(c.crime_case_density1_weapon) OVER (PARTITION BY c.weapon_type
ORDER BY d.date), 2) AS w_avg_crime_case_density_for_weapon_type,
        ROUND(AVG(c.crime_case_density1_weapon) OVER (PARTITION BY c.weapon_type,
d.month ORDER BY d.date), 2) AS w_avg_crime_case_density_bymonth_for_weapon_type,
        ROUND(AVG(c.crime_case_density1_weapon) OVER (PARTITION BY c.weapon_type,
d.month_year ORDER BY d.date), 2) AS
w_avg_crime_case_density_by_month_year_for_weapon_type,
        ROUND(AVG(c.crime_case_density1_weapon) OVER (PARTITION BY c.weapon_type,
d.year ORDER BY d.date), 2) AS w_avg_crime_case_density_by_year_for_weapon_type
    FROM WOC2 c
    JOIN date_table d ON c.date_rptd = d.date
)
```

- The above shows an extensive number of windows functions. The inclusion of both the weapon level and weapon type calls for three levels to allow for the rolling values on both the weapon and weapon type level.

Rate CTE's

- The rate tables include calculations for both the count and rate fields unique to them are done in the first CTE.
- Windows functions are then applied using the count and the rate fields.

Arrests

```
WITH AOC AS (

SELECT

        lc.date_rptd::DATE AS report_date,

        SUM(cc.crime_tally) AS crime_tally,

        ROUND(

            CAST(SUM(

                CASE

                    WHEN lc.status_description IN ('Adult Arrest', 'Juv Arrest')
THEN 1

                    ELSE 0

                END

            ) AS NUMERIC)

            / NULLIF(SUM(cc.crime_tally), 0), 2

        ) AS arrest_rate,

        SUM(

            CASE

                WHEN lc.status_description IN ('Adult Arrest', 'Juv Arrest') THEN
1
```

```
                ELSE 0

            END

        ) AS arrests

    FROM

        la_crimes lc

    LEFT JOIN

        crime_codes cc

        ON cc.division_records_num = lc.division_records_num

    GROUP BY

        lc.date_rptd

)

, AOC2 AS (

    SELECT

        c.report_date,

        c.arrests,

        c.arrest_rate,

        SUM(c.arrests) OVER (ORDER BY c.report_date) AS rolling_arrests,

        SUM(c.arrests) OVER (PARTITION BY d.month ORDER BY c.report_date) AS
rolling_arrests_bymonth,

        SUM(c.arrests) OVER (PARTITION BY d.month_year ORDER BY c.report_date) AS
rolling_arrests_by_month_year,

        SUM(c.arrests) OVER (PARTITION BY d.year ORDER BY c.report_date) AS
rolling_arrests_by_year,
```

```
        ROUND(AVG(c.arrests) OVER (ORDER BY c.report_date), 2) AS
avg_rolling_arrests,

        ROUND(AVG(c.arrests) OVER (PARTITION BY d.month ORDER BY c.report_date),
2) AS avg_arrests_by_month,

        ROUND(AVG(c.arrests) OVER (PARTITION BY d.month_year ORDER BY
c.report_date), 2) AS avg_arrests_by_month_year,

        ROUND(AVG(c.arrests) OVER (PARTITION BY d.year ORDER BY c.report_date),
2) AS avg_arrests_by_year,

        ROUND(AVG(c.arrest_rate) OVER (ORDER BY c.report_date), 2) AS
avg_arrest_rate,

        ROUND(AVG(c.arrest_rate) OVER (PARTITION BY d.month ORDER BY
c.report_date), 2) AS avg_arrest_rate_by_month,

        ROUND(AVG(c.arrest_rate) OVER (PARTITION BY d.month_year ORDER BY
c.report_date), 2) AS avg_arrest_rate_by_month_year,

        ROUND(AVG(c.arrest_rate) OVER (PARTITION BY d.year ORDER BY
c.report_date), 2) AS avg_arrest_rate_by_year,

        ROUND(AVG(c.arrests) OVER (ORDER BY c.report_date ROWS BETWEEN 89
PRECEDING AND CURRENT ROW), 2) AS "90day_avg_rolling_arrests",

        ROUND(AVG(c.arrest_rate) OVER (ORDER BY c.report_date ROWS BETWEEN 89
PRECEDING AND CURRENT ROW), 2) AS "90day_avg_rolling_arrest_rate"

    FROM

        AOC c

    LEFT JOIN

        date_table d

        ON c.report_date = d.date

)
```

Violent Crime

```
WITH VOC AS (
```

```
    SELECT
        lc.date_rptd::DATE AS report_date,
        SUM(cc.crime_tally) AS crime_tally,
        ROUND(
            CAST(SUM(
                CASE
                    WHEN lc.crime_code_description IN (
                        'AGGRAVATED ASSAULT', 'SIMPLE ASSAULT', 'ROBBERY',
'MURDER', 'RAPE'
                    ) THEN 1
                    ELSE 0
                END
            ) AS NUMERIC)
            / NULLIF(SUM(cc.crime_tally), 0), 2
        ) AS violent_crime_rate,
        SUM(
            CASE
                WHEN lc.crime_code_description IN (
                    'AGGRAVATED ASSAULT', 'SIMPLE ASSAULT', 'ROBBERY', 'MURDER',
'RAPE'
                ) THEN 1
                ELSE 0
            END
        ) AS violent_crimes
    FROM
        la_crimes lc
    LEFT JOIN
        crime_codes cc
        ON cc.division_records_num = lc.division_records_num
    GROUP BY
```

```
        lc.date_rptd

)

, VOC2 AS (

    SELECT

        v.report_date,

        v.violent_crimes,

        v.violent_crime_rate,

        SUM(v.violent_crimes) OVER (ORDER BY v.report_date) AS
rolling_violent_crimes,

        SUM(v.violent_crimes) OVER (PARTITION BY d.month ORDER BY v.report_date)
AS rolling_violent_crimes_bymonth,

        SUM(v.violent_crimes) OVER (PARTITION BY d.month_year ORDER BY
v.report_date) AS rolling_violent_crimes_by_month_year,
        SUM(v.violent_crimes) OVER (PARTITION BY d.year ORDER BY v.report_date)
AS rolling_violent_crimes_by_year,

        ROUND(AVG(v.violent_crimes) OVER (ORDER BY v.report_date), 2) AS
avg_rolling_violent_crimes,

        ROUND(AVG(v.violent_crimes) OVER (PARTITION BY d.month ORDER BY
v.report_date), 2) AS avg_violent_crimes_by_month,

        ROUND(AVG(v.violent_crimes) OVER (PARTITION BY d.month_year ORDER BY
v.report_date), 2) AS avg_violent_crimes_by_month_year,

        ROUND(AVG(v.violent_crimes) OVER (PARTITION BY d.year ORDER BY
v.report_date), 2) AS avg_violent_crimes_by_year,

        ROUND(AVG(v.violent_crime_rate) OVER (ORDER BY v.report_date), 2) AS
avg_violent_crime_rate,

        ROUND(AVG(v.violent_crime_rate) OVER (PARTITION BY d.month ORDER BY
v.report_date), 2) AS avg_violent_crime_rate_by_month,

        ROUND(AVG(v.violent_crime_rate) OVER (PARTITION BY d.month_year ORDER BY
v.report_date), 2) AS avg_violent_crime_rate_by_month_year,

        ROUND(AVG(v.violent_crime_rate) OVER (PARTITION BY d.year ORDER BY
v.report_date), 2) AS avg_violent_crime_rate_by_year,

        ROUND(AVG(v.violent_crimes) OVER (ORDER BY v.report_date ROWS BETWEEN 89
PRECEDING AND CURRENT ROW), 2) AS "90day_avg_rolling_violent_crimes",

        ROUND(AVG(v.violent_crime_rate) OVER (ORDER BY v.report_date ROWS BETWEEN
89 PRECEDING AND CURRENT ROW), 2) AS "90day_avg_rolling_violent_crime_rate"

    FROM
```

```
        VOC v

    LEFT JOIN

        date_table d

        ON v.report_date = d.date

)
```

Domestic Violence

```
WITH DVOC AS (
    SELECT
        lc.date_rptd::DATE AS report_date,
        SUM(cc.crime_tally) AS crime_tally,
        ROUND(
            CAST(SUM(
                CASE
                    WHEN lc.crime_code_description IN('DOMESTIC BATTERY',
                        'DOMESTIC VIOLENCE', 'INTIMATE PARTNER - AGGRAVATED
ASSAULT',
                        'INTIMATE PARTNER - SIMPLE ASSAULT') THEN 1
                    ELSE 0
                END
            ) AS NUMERIC)
            / NULLIF(SUM(cc.crime_tally), 0), 2
        ) AS domestic_violence_rate,
        SUM(
            CASE
                WHEN lc.crime_code_description IN('DOMESTIC BATTERY',
                        'DOMESTIC VIOLENCE', 'INTIMATE PARTNER - AGGRAVATED
ASSAULT',
                        'INTIMATE PARTNER - SIMPLE ASSAULT') THEN 1
                ELSE 0
            END
        ) AS domestic_violence
    FROM
        la_crimes lc
    LEFT JOIN
        crime_codes cc
        ON cc.division_records_num = lc.division_records_num
    GROUP BY
        lc.date_rptd
)
, DVOC2 AS (
    SELECT
        d.report_date,
        d.domestic_violence,
        d.domestic_violence_rate,
        SUM(d.domestic_violence) OVER (ORDER BY d.report_date) AS
rolling_domestic_violence,
```

```
        SUM(d.domestic_violence) OVER (PARTITION BY dt.month ORDER BY
d.report_date) AS rolling_domestic_violence_bymonth,
        SUM(d.domestic_violence) OVER (PARTITION BY dt.month_year ORDER BY
d.report_date) AS rolling_domestic_violence_by_month_year,
        SUM(d.domestic_violence) OVER (PARTITION BY dt.year ORDER BY
d.report_date) AS rolling_domestic_violence_by_year,
        ROUND(AVG(d.domestic_violence) OVER (ORDER BY d.report_date), 2) AS
avg_rolling_domestic_violence,
        ROUND(AVG(d.domestic_violence) OVER (PARTITION BY dt.month ORDER BY
d.report_date), 2) AS avg_domestic_violence_by_month,
        ROUND(AVG(d.domestic_violence) OVER (PARTITION BY dt.month_year ORDER BY
d.report_date), 2) AS avg_domestic_violence_by_month_year,
        ROUND(AVG(d.domestic_violence) OVER (PARTITION BY dt.year ORDER BY
d.report_date), 2) AS avg_domestic_violence_by_year,
        ROUND(AVG(d.domestic_violence_rate) OVER (ORDER BY d.report_date), 2) AS
avg_domestic_violence_rate,
        ROUND(AVG(d.domestic_violence_rate) OVER (PARTITION BY dt.month ORDER BY
d.report_date), 2) AS avg_domestic_violence_rate_by_month,
        ROUND(AVG(d.domestic_violence_rate) OVER (PARTITION BY dt.month_year
ORDER BY d.report_date), 2) AS avg_domestic_violence_rate_by_month_year,
        ROUND(AVG(d.domestic_violence_rate) OVER (PARTITION BY dt.year ORDER BY
d.report_date), 2) AS avg_domestic_violence_rate_by_year,
        ROUND(AVG(d.domestic_violence) OVER (ORDER BY d.report_date ROWS BETWEEN
89 PRECEDING AND CURRENT ROW), 2) AS "90day_avg_rolling_domestic_violence",
        ROUND(AVG(d.domestic_violence_rate) OVER (ORDER BY d.report_date ROWS
BETWEEN 89 PRECEDING AND CURRENT ROW), 2) AS
"90day_avg_rolling_domestic_violence_rate"
    FROM
        DVOC d
    LEFT JOIN
        date_table dt
        ON d.report_date = dt.date
)
```

CTE Insertion into Tables

- The insertion of all of the data into the tables happens using the last CTE to select fields and load the data into the tables made in the beginning. Do not forget all CTEs relevant to the insertion need to also be highlighted when running the insertion query.

General

```
INSERT INTO overall_crimes (
    date_rptd, month_year, month, year, crime_tally, crime_case_density,
    rolling_crime_tally, rolling_crime_tally_bymonth,
rolling_crime_tally_by_month_year,
    rolling_crime_tally_by_year, crime_case_density_bymonth,
crime_case_density_by_month_year,
    crime_case_density_by_year, avg_rolling_crime_tally,
avg_crime_tally_by_month,
```

```
    avg_crime_tally_by_month_year, avg_crime_tally_by_year,
avg_crime_case_density,
    avg_crime_case_density_by_month, avg_crime_case_density_by_month_year,
avg_crime_case_density_by_year,
    "90day_AVG_rolling_crime_tally", "90day_AVG_rolling_crime_case_density"
)
SELECT
    date, month_year, month, year, crime_tally, crime_case_density1 AS
crime_case_density,
    rolling_crime_tally, rolling_crime_tally_bymonth,
rolling_crime_tally_by_month_year,
    rolling_crime_tally_by_year, rolling_crime_case_density_bymonth,
rolling_crime_case_density_by_month_year,
    rolling_crime_case_density_by_year, avg_rolling_crime_tally,
avg_crime_tally_by_month,
    avg_crime_tally_by_month_year, avg_crime_tally_by_year,
avg_crime_case_density,
    avg_crime_case_density_by_month, avg_crime_case_density_by_month_year,
avg_crime_case_density_by_year,
    "90day_AVG_rolling_crime_tally", "90day_AVG_rolling_crime_case_density"
FROM OC;
```

Weapons

```
INSERT INTO w_overall_crimes (
    w_date,
    w_month_year,
    w_month,
    w_year,
    weapon_type,
    weapon_description,
    w_crime_tally,
    w_rolling_crime_tally_for_weapon,
    w_rolling_crime_tally_bymonth_for_weapon,
    w_rolling_crime_tally_by_month_year_for_weapon,
    w_rolling_crime_tally_by_year_for_weapon,
    w_rolling_crime_case_density_for_weapon,
    w_crime_case_density_bymonth_for_weapon,
    w_crime_case_density_by_month_year_for_weapon,
    w_crime_case_density_by_year_for_weapon,
    w_rolling_crime_tally_for_weapon_type,
    w_rolling_crime_tally_bymonth_for_weapon_type,
    w_rolling_crime_tally_by_month_year_for_weapon_type,
    w_rolling_crime_tally_by_year_for_weapon_type,
    w_rolling_crime_case_density_for_weapon_type,
    w_crime_case_density_bymonth_for_weapon_type,
    w_crime_case_density_by_month_year_for_weapon_type,
    w_crime_case_density_by_year_for_weapon_type,
    w_avg_rolling_crime_tally_for_weapon,
    w_avg_rolling_crime_tally_bymonth_for_weapon,
    w_avg_rolling_crime_tally_by_month_year_for_weapon,
```

```
    w_avg_rolling_crime_tally_by_year_for_weapon,
    w_avg_crime_case_density_for_weapon,
    w_avg_crime_case_density_bymonth_for_weapon,
    w_avg_crime_case_density_by_month_year_for_weapon,
    w_avg_crime_case_density_by_year_for_weapon,
    w_90day_AVG_rolling_crime_tally_for_weapon,
    w_90day_AVG_rolling_crime_case_density_for_weapon,
    w_avg_rolling_crime_tally_for_weapon_type,
    w_avg_rolling_crime_tally_bymonth_for_weapon_type,
    w_avg_rolling_crime_tally_by_month_year_for_weapon_type,
    w_avg_rolling_crime_tally_by_year_for_weapon_type,
    w_avg_crime_case_density_for_weapon_type,
    w_avg_crime_case_density_bymonth_for_weapon_type,
    w_avg_crime_case_density_by_month_year_for_weapon_type,
    w_avg_crime_case_density_by_year_for_weapon_type
)
SELECT
    w_date,
    w_month_year,
    w_month,
    w_year,
    weapon_type,
    weapon_description,
    w_crime_tally,
    w_rolling_crime_tally_for_weapon,
    w_rolling_crime_tally_bymonth_for_weapon,
    w_rolling_crime_tally_by_month_year_for_weapon,
    w_rolling_crime_tally_by_year_for_weapon,
    w_rolling_crime_case_density_for_weapon,
    w_crime_case_density_bymonth_for_weapon,
    w_crime_case_density_by_month_year_for_weapon,
    w_crime_case_density_by_year_for_weapon,
    w_rolling_crime_tally_for_weapon_type,
    w_rolling_crime_tally_bymonth_for_weapon_type,
    w_rolling_crime_tally_by_month_year_for_weapon_type,
    w_rolling_crime_tally_by_year_for_weapon_type,
    w_rolling_crime_case_density_for_weapon_type,
    w_crime_case_density_bymonth_for_weapon_type,
    w_crime_case_density_by_month_year_for_weapon_type,
    w_crime_case_density_by_year_for_weapon_type,
    w_avg_rolling_crime_tally_for_weapon,
    w_avg_rolling_crime_tally_bymonth_for_weapon,
    w_avg_rolling_crime_tally_by_month_year_for_weapon,
    w_avg_rolling_crime_tally_by_year_for_weapon,
    w_avg_crime_case_density_for_weapon,
    w_avg_crime_case_density_bymonth_for_weapon,
    w_avg_crime_case_density_by_month_year_for_weapon,
    w_avg_crime_case_density_by_year_for_weapon,
    w_90day_AVG_rolling_crime_tally_for_weapon,
    w_90day_AVG_rolling_crime_case_density_for_weapon,
    w_avg_rolling_crime_tally_for_weapon_type,
    w_avg_rolling_crime_tally_bymonth_for_weapon_type,
```

```
    w_avg_rolling_crime_tally_by_month_year_for_weapon_type,
    w_avg_rolling_crime_tally_by_year_for_weapon_type,
    w_avg_crime_case_density_for_weapon_type,
    w_avg_crime_case_density_bymonth_for_weapon_type,
    w_avg_crime_case_density_by_month_year_for_weapon_type,
    w_avg_crime_case_density_by_year_for_weapon_type
FROM WOC3;
```

Arrests

```
INSERT INTO arrest_agg (
    report_date, arrests, arrest_rate,
    rolling_arrests, rolling_arrests_bymonth, rolling_arrests_by_month_year,
    rolling_arrests_by_year, avg_rolling_arrests, avg_arrests_by_month,
    avg_arrests_by_month_year, avg_arrests_by_year, avg_arrest_rate,
    avg_arrest_rate_by_month, avg_arrest_rate_by_month_year,
avg_arrest_rate_by_year,
    "90day_avg_rolling_arrests", "90day_avg_rolling_arrest_rate"
)
SELECT
    report_date, arrests, arrest_rate,
    rolling_arrests, rolling_arrests_bymonth, rolling_arrests_by_month_year,
    rolling_arrests_by_year, avg_rolling_arrests, avg_arrests_by_month,
    avg_arrests_by_month_year, avg_arrests_by_year, avg_arrest_rate,
    avg_arrest_rate_by_month, avg_arrest_rate_by_month_year,
avg_arrest_rate_by_year,
    "90day_avg_rolling_arrests", "90day_avg_rolling_arrest_rate"
FROM AOC2;
```

Violent Crime

```
INSERT INTO violent_crimes_agg (
    report_date, violent_crimes, violent_crime_rate,
    rolling_violent_crimes, rolling_violent_crimes_bymonth,
rolling_violent_crimes_by_month_year,
    rolling_violent_crimes_by_year, avg_rolling_violent_crimes,
avg_violent_crimes_by_month,
    avg_violent_crimes_by_month_year, avg_violent_crimes_by_year,
avg_violent_crime_rate,
    avg_violent_crime_rate_by_month, avg_violent_crime_rate_by_month_year,
avg_violent_crime_rate_by_year,
    "90day_avg_rolling_violent_crimes", "90day_avg_rolling_violent_crime_rate"
)
SELECT
    report_date, violent_crimes, violent_crime_rate,
    rolling_violent_crimes, rolling_violent_crimes_bymonth,
rolling_violent_crimes_by_month_year,
    rolling_violent_crimes_by_year, avg_rolling_violent_crimes,
avg_violent_crimes_by_month,
    avg_violent_crimes_by_month_year, avg_violent_crimes_by_year,
avg_violent_crime_rate,
    avg_violent_crime_rate_by_month, avg_violent_crime_rate_by_month_year,
avg_violent_crime_rate_by_year,
    "90day_avg_rolling_violent_crimes", "90day_avg_rolling_violent_crime_rate"
FROM VOC2;
```

Domestic Violence

```
INSERT INTO domestic_violence_agg (
    report_date, domestic_violence_count, domestic_violence_rate,
    rolling_domestic_violence, rolling_domestic_violence_bymonth,
rolling_domestic_violence_by_month_year,
    rolling_domestic_violence_by_year, avg_rolling_domestic_violence,
avg_domestic_violence_by_month,
    avg_domestic_violence_by_month_year, avg_domestic_violence_by_year,
avg_domestic_violence_rate,
    avg_domestic_violence_rate_by_month,
avg_domestic_violence_rate_by_month_year, avg_domestic_violence_rate_by_year,
    "90day_avg_rolling_domestic_violence",
"90day_avg_rolling_domestic_violence_rate"
)
SELECT
    report_date, domestic_violence AS domestic_violence_count,
domestic_violence_rate,
    rolling_domestic_violence, rolling_domestic_violence_bymonth,
rolling_domestic_violence_by_month_year,
    rolling_domestic_violence_by_year, avg_rolling_domestic_violence,
avg_domestic_violence_by_month,
    avg_domestic_violence_by_month_year, avg_domestic_violence_by_year,
avg_domestic_violence_rate,
```

```
    avg_domestic_violence_rate_by_month,
avg_domestic_violence_rate_by_month_year, avg_domestic_violence_rate_by_year,
    "90day_avg_rolling_domestic_violence",
"90day_avg_rolling_domestic_violence_rate"
FROM DVOC2;
```