

## Part D: Data Retrieval and Interpretation

### Task D1

**Interpretation :** This query totals Australia's COVID-19 vaccinations by month from the first Australian record in the dataset up to 30 June 2021. We filter to isoCode='AUS', add up each month's daily doses (or, if only cumulative data exist, use the day-to-day increase and treat any negative corrections as zero), then output one row per month (YYYY-MM) with the monthly total. Partial months are included with whatever data is available.

#### Query:

```
SELECT      strftime('%Y', v.date) AS Year,
            strftime('%m', v.date) AS Month,
            sum(v.peopleVaccinated + v.peopleFullyVaccinated + v.peopleBooster) AS
            'Monthly Vaccinations'
FROM vaccinationCountry v
    JOIN country c
        ON v.isoCode = c.isoCode
WHERE LOWER(countryName) = 'australia'
AND v.date <= '2021-07-30'
GROUP BY strftime('%Y', v.date), strftime('%m', v.date);
```

## Vaccination.db - Query

new\_table\_name [Create](#) [Query](#)

table name...

[ageGroup](#)  
[country](#)  
[sourceCountry](#)  
[sourceVaccination](#)  
[sourceWebsite](#)  
[state](#)  
[usedVaccine](#)  
[vaccinationCountry](#)  
[vaccinationState](#)  
[vaccine](#)  
[vaccineAgeGroup](#)  
[vaccineManufacturer](#)

[Toggle helper tables](#)

```
SELECT  strftime('%Y', v.date) AS Year,
        strftime('%m', v.date) AS Month,
        sum(v.peopleVaccinated + v.peopleFullyVaccinated + v.peopleBooster) AS 'Monthly Vaccinations'
FROM vaccinationCountry v
    JOIN country c
        ON v.isoCode = c.isoCode
WHERE LOWER(countryName) = 'australia'
AND v.date <= '2021-07-30'
GROUP BY strftime('%Y', v.date), strftime('%m', v.date);
```

Use Shift + Up/Down to navigate recently-executed queries

[Execute](#) [Export JSON](#) [Export CSV](#) [SQL Help](#)

[Bookmarks](#) +

### Results (6)

[Permalink](#)

Year	Month	Monthly Vaccinations
2021	02	146386
2021	03	8190046
2021	04	43864057
2021	05	98020395
2021	06	174706637
2021	07	296621252

« < Page 1 of 1 > »

## Task D2

**Interpretation** : For each of the six countries (Australia, India, US, Argentina, Canada, Greece), list every month that has data and show how much the total number of people fully vaccinated increased compared to the previous month. For a country's first month in the dataset, set the change to 0 (no prior month). If a country has no data for a month, skip that month for that country. In short: compute, per country and per month, monthly change = VOM(current month) – VOM(previous month), with the first month's change = 0, and output one row per (country, month) that exists in the data.

### Query:

```
SELECT strftime('%Y', t.month_start) AS "Year (OY)",
       strftime('%m', t.month_start) AS "Month (OM)",
       COALESCE(SUM(CASE WHEN t.country = 'Australia' THEN
t.monthly_change END), 0) AS "Australia",
       COALESCE(SUM(CASE WHEN t.country = 'India' THEN t.monthly_change
END), 0) AS "India",
       COALESCE(SUM(CASE WHEN t.country = 'United States' THEN
t.monthly_change END), 0) AS "United States",
       COALESCE(SUM(CASE WHEN t.country = 'Argentina' THEN
t.monthly_change END), 0) AS "Argentina",
       COALESCE(SUM(CASE WHEN t.country = 'Canada' THEN t.monthly_change
END), 0) AS "Canada",
       COALESCE(SUM(CASE WHEN t.country = 'Greece' THEN t.monthly_change
END), 0) AS "Greece"
FROM (
  SELECT m.country,
         m.month_start,
         CASE
           WHEN p.VOM IS NULL THEN 0
           ELSE (m.VOM - p.VOM)
         END AS monthly_change
  FROM (
```

```

SELECT C.countryName AS country,
       date(strftime('%Y-%m-01', V.date)) AS month_start,
       MAX(V.peopleFullyVaccinated) AS VOM
FROM VaccinationCountry V
JOIN Country C USING (isoCode)
WHERE C.countryName IN ('Australia','India','United
States','Argentina','Canada','Greece')
GROUP BY country, month_start
) AS m
LEFT JOIN (
SELECT C.countryName AS country,
       date(strftime('%Y-%m-01', V.date)) AS month_start,
       MAX(V.peopleFullyVaccinated) AS VOM
FROM VaccinationCountry V
JOIN Country C USING (isoCode)
WHERE C.countryName IN ('Australia','India','United
States','Argentina','Canada','Greece')
GROUP BY country, month_start
) AS p
ON p.country = m.country
AND p.month_start = date(m.month_start, '-1 month')
) AS t
GROUP BY t.month_start
HAVING COUNT(DISTINCT t.country) = 6
ORDER BY "Year (OY)", "Month (OM)";

```

Vaccination.db - Query

new\_table\_name Create Query

- table name...
- ageGroup

country

sourceCountry

sourceVaccination

sourceWebsite

state

usedVaccine

vaccinationCountry

vaccinationState

vaccine

vaccineAgeGroup

vaccineManufacturer

Toggle helper tables

```
SELECT strftime('%Y', t.month_start) AS "Year (OY)",
       strftime('%m', t.month_start) AS "Month (OM)",
       COALESCE(SUM(CASE WHEN t.country = 'Australia' THEN t.monthly_change END), 0) AS "Australia",
       COALESCE(SUM(CASE WHEN t.country = 'India' THEN t.monthly_change END), 0) AS "India",
       COALESCE(SUM(CASE WHEN t.country = 'United States' THEN t.monthly_change END), 0) AS "United States",
       COALESCE(SUM(CASE WHEN t.country = 'Argentina' THEN t.monthly_change END), 0) AS "Argentina",
       COALESCE(SUM(CASE WHEN t.country = 'Canada' THEN t.monthly_change END), 0) AS "Canada",
       COALESCE(SUM(CASE WHEN t.country = 'Greece' THEN t.monthly_change END), 0) AS "Greece"
FROM (
```

Use Shift + Up/Down to navigate recently-executed queries

Execute Export JSON Export CSV SQL Help

Bookmarks +

Results (unable to determine)

Permalink

Year (OY)	Month (OM)	Australia	India	United States	Argentina	Canada	Greece
2021	02	0	2456191	22724848	205481	422333	274759
2021	03	72712	6878504	34688799	401425	176334	293520
2021	04	185805	17286460	50221605	308993	421781	336956
2021	05	247725	16637655	32042852	1979850	1011199	1114164
2021	06	1127180	14489306	16703222	1318698	9644229	1730619
2021	07	2293622	43950050	7487141	3055689	10618652	1376760
2021	08	3442264	46098976	9850509	6431463	3140130	572440
2021	09	4017019	89679107	9957501	6849088	1501531	385280
2021	10	5143103	91612373	6338180	3326595	1263988	252357
2021	11	2279370	118912324	5079954	3877508	740278	238468
2021	12	947025	155223875	7826958	2922771	383097	426922
2022	01	455565	106701626	4854852	1916003	727023	294144
2022	02	771030	81678087	3111003	1760930	740430	7111310

Vaccination.db - Query

new\_table\_name Create Query

- table name...
- ageGroup

country

sourceCountry

sourceVaccination

sourceWebsite

state

usedVaccine

vaccinationCountry

vaccinationState

vaccine

vaccineAgeGroup

vaccineManufacturer

Toggle helper tables

```
SELECT strftime('%Y', t.month_start) AS "Year (OY)",
       strftime('%m', t.month_start) AS "Month (OM)",
       COALESCE(SUM(CASE WHEN t.country = 'Australia' THEN t.monthly_change END), 0) AS "Australia",
       COALESCE(SUM(CASE WHEN t.country = 'India' THEN t.monthly_change END), 0) AS "India",
       COALESCE(SUM(CASE WHEN t.country = 'United States' THEN t.monthly_change END), 0) AS "United States",
       COALESCE(SUM(CASE WHEN t.country = 'Argentina' THEN t.monthly_change END), 0) AS "Argentina",
       COALESCE(SUM(CASE WHEN t.country = 'Canada' THEN t.monthly_change END), 0) AS "Canada",
       COALESCE(SUM(CASE WHEN t.country = 'Greece' THEN t.monthly_change END), 0) AS "Greece"
FROM (
  SELECT m.country,
         m.month_start,
         CASE
           WHEN p.vom IS NULL THEN 0
           ELSE (m.vom - p.vom)
         END AS monthly_change
  FROM (
    SELECT c.countryName AS country,
           date(strftime('%Y-%m-01', v.date)) AS month_start,
           MAX(v.peopleFullyVaccinated) AS vom
    FROM VaccinationCountry V
    JOIN Country C USING (isoCode)
    WHERE C.countryName IN ('Australia','India','United States','Argentina','Canada','Greece')
    GROUP BY country, month_start
  ) AS m
  LEFT JOIN (
    SELECT c.countryName AS country,
           date(strftime('%Y-%m-01', v.date)) AS month_start,
           MAX(v.peopleFullyVaccinated) AS vom
    FROM VaccinationCountry V
    JOIN Country C USING (isoCode)
    WHERE C.countryName IN ('Australia','India','United States','Argentina','Canada','Greece')
    GROUP BY country, month_start
  ) AS p
  ON p.country = m.country
  AND p.month_start = date(m.month_start, '-1 month')
) AS t
GROUP BY t.month_start
HAVING COUNT(DISTINCT t.country) = 6
ORDER BY "Year (OY)", "Month (OM)"
```

Use Shift + Up/Down to navigate recently-executed queries

Execute Export JSON Export CSV SQL Help

Bookmarks +

Results (unable to determine)

Permalink

vaccinationState  
vaccine  
vaccineAgeGroup  
vaccineManufacturer

Toggle helper tables

Execute

Export JSON

Export CSV

SQL Help

Bookmarks +

Results (unable to determine)

Permalink

Year (OY)	Month (OM)	Australia	India	United States	Argentina	Canada	Greece
2021	02	0	2456191	22724848	205481	422333	274759
2021	03	72712	6878504	34688799	401425	176334	293520
2021	04	185805	17286460	50221605	308993	421781	336956
2021	05	247725	16637655	32042852	1979850	1011199	1114164
2021	06	1127180	14489306	16703222	1318698	9644229	1730619
2021	07	2293622	43950050	7487141	3055689	10618652	1376760
2021	08	3442264	46098976	9850509	6431463	3140130	572440
2021	09	4017019	89679107	9957501	6849088	1501531	385280
2021	10	5143103	91612373	6338180	3326595	1263998	252357
2021	11	2279370	118912324	5079954	3877508	740278	238468
2021	12	947025	155223875	7826958	2922771	383097	426922
2022	01	455565	106701626	4854852	1916003	727023	294144
2022	02	271838	80629807	3115093	1068839	748435	201316
2022	03	757212	40682742	1568437	444909	276195	59576
2022	04	286647	27561584	1534837	209762	122734	21017
2022	05	87500	29827676	1189860	161926	85225	7712
2022	06	32924	25851447	888565	62471	53893	4337
2022	07	-21647523	18295691	927144	39067	45205	3796
2022	08	0	9551797	848597	30994	46089	2759
2022	09	0	6044062	1218069	29616	39746	2753
2022	10	0	1719186	1418164	23906	57983	1738
2022	11	0	675053	975989	19551	50632	1509
2022	12	0	441919	680440	17488	40620	1057
2023	01	0	487886	423604	31068	27154	1052
2023	02	0	186606	237404	26062	-31758251	414
2023	03	0	75423	216314	12437	0	680
2023	04	0	10743	143628	6900	0	318
2023	05	0	7970	35760	5851	0	411

«

<

Page 1 of 1

>

»

## References

- W3Schools, 2024. *SQL CASE Statement*. [online] Available at: [https://www.w3schools.com/sql/sql\\_case.asp](https://www.w3schools.com/sql/sql_case.asp) [Accessed 30 October 2025].
- W3Schools, 2024. *SQL COALESCE Function*. [online] Available at: [https://www.w3schools.com/sql/func\\_sqlserver\\_coalesce.asp](https://www.w3schools.com/sql/func_sqlserver_coalesce.asp) [Accessed 30 October 2025].
- GeeksforGeeks, 2024. *SQL | CASE Statement*. [online] Available at: <https://www.geeksforgeeks.org/sql-case-statement/> [Accessed 30 October 2025].
- Oracle, 2024. *CASE Expressions*. [online] Available at: <https://docs.oracle.com/en/database/> [Accessed 30 October 2025].
- Microsoft Learn, 2024. *COALESCE (Transact-SQL)*. [online] Available at: <https://learn.microsoft.com/en-us/sql/t-sql/functions/coalesce-transact-sql> [Accessed 30 October 2025].