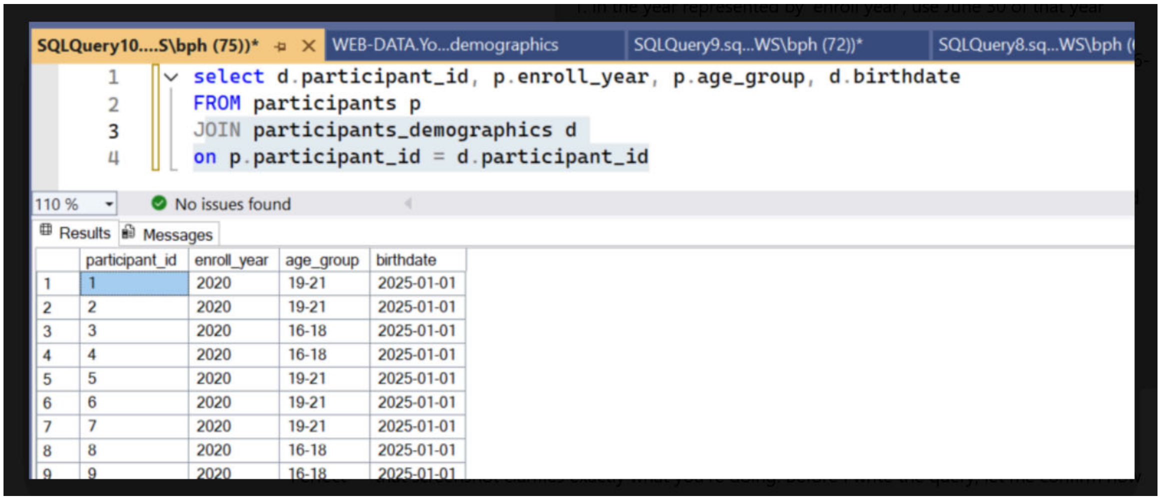
SQLQuery10....S\bph (75))*
WEB-DATA.Yo...demographics
SQLQuery9.sq...WS\bph (72))*
SQLQuery8.sq...WS\bph (
110 %
No issues found
Results
Messages

```sql
select d.participant_id, p.enroll_year, p.age_group, d.birthdate
FROM participants p
JOIN participants_demographics d
on p.participant_id = d.participant_id
```

	participant_id	enroll_year	age_group	birthdate
1	1	2020	19-21	2025-01-01
2	2	2020	19-21	2025-01-01
3	3	2020	16-18	2025-01-01
4	4	2020	16-18	2025-01-01
5	5	2020	19-21	2025-01-01
6	6	2020	19-21	2025-01-01
7	7	2020	19-21	2025-01-01
8	8	2020	16-18	2025-01-01
9	9	2020	16-18	2025-01-01

- Here is an example of the resulting combined data. I would now like to update the birthdate (in participants_demographic) to a random date as follows: 1. in the year represented by 'enroll year', use June 30 of that year (midway through the year). 2. refer to the column 'age group' and wich will be either '19-21' or '16-18', 3. then create a random date that is within that range For example if the year is 2020 and the age group is 16-18 the birthdate would make this imaginary participant 16, 17 or 18 years old on june 30, 2020. lets do this as a select first, then when it looks right we can make that an update. Ask any clarifying questions that may be needed, first.

ChatGPT said:

Perfect — that screenshot clarifies exactly what you're doing. Before I write the query, let me confirm how you'd like to handle one subtle but important point:

When you say "create a random date that is within that range," we have two possible interpretations:

1. **Literal birthdays:**
   For each participant, the resulting birthdate should be sometime between **June 30 of** `(enroll_year - max_age)` and **June 30 of** `(enroll_year - min_age)`.
   Example: if enroll_year = 2020 and age_group = '16-18', then any date between **2002-07-01 and 2004-06-30**.
2. **Exactly their age on June 30 of the enroll year:**
   So, for each participant we randomly choose an integer age (e.g., 16, 17, or 18), and their birthday is **June 30 minus that number of years**, possibly plus/minus a few random days within that year.

Could you confirm which version you want?

If it's #1 (full random date within the range), I'll give you a clean `SELECT` that shows a reproducible random date for each record using `RAND(CHECKSUM(NEWID()))`.

If it's #2 (age as of June 30 of enroll_year), I'll instead generate one of the three valid ages and pick a random day within that year — both doable directly in T-SQL.

Once you confirm which you want, I'll show you:

- a safe `SELECT` **preview** (so you can see the dates),

- and then a matching `UPDATE` statement.

You said:

it should be #1 unless by "plus/minus a few random days" you meant +/- 180 days

ChatGPT said:

Perfect — that’s clear. ✅
We’ll go with **#1**: a *truly random birthday* within the valid range, i.e.:

- For **‘16–18’**: random date between **June 30 – 18 years** and **June 30 – 16 years**
- For **‘19–21’**: random date between **June 30 – 21 years** and **June 30 – 19 years**

Here’s a **preview query** you can safely run first (it only selects — no updates yet):

```
SELECT
    d.participant_id,
    p.enroll_year,
    p.age_group,
    -- calculate lower & upper bounds based on enroll_year and age_group
    CASE
        WHEN p.age_group = '16-18' THEN DATEADD(YEAR, -18, DATEFROMPARTS(p.enroll_year, 6, 30))
        WHEN p.age_group = '19-21' THEN DATEADD(YEAR, -21, DATEFROMPARTS(p.enroll_year, 6, 30))
    END AS min_birthdate,
    CASE
        WHEN p.age_group = '16-18' THEN DATEADD(YEAR, -16, DATEFROMPARTS(p.enroll_year, 6, 30))
        WHEN p.age_group = '19-21' THEN DATEADD(YEAR, -19, DATEFROMPARTS(p.enroll_year, 6, 30))
    END AS max_birthdate,
    -- random birthday within the range
    DATEADD(
        DAY,
        CAST(RAND(CHECKSUM(NEWID())) *
             DATEDIFF(
                 DAY,
                 CASE
                    WHEN p.age_group = '16-18' THEN DATEADD(YEAR, -18, DATEFROMPARTS(p.enroll_year, 6, 30))
                    WHEN p.age_group = '19-21' THEN DATEADD(YEAR, -21, DATEFROMPARTS(p.enroll_year, 6, 30))
                 END,
                 CASE
                    WHEN p.age_group = '16-18' THEN DATEADD(YEAR, -16, DATEFROMPARTS(p.enroll_year, 6, 30))
                    WHEN p.age_group = '19-21' THEN DATEADD(YEAR, -19, DATEFROMPARTS(p.enroll_year, 6, 30))
                 END
             ) AS int)
        ,
        CASE
            WHEN p.age_group = '16-18' THEN DATEADD(YEAR, -18, DATEFROMPARTS(p.enroll_year, 6, 30))
            WHEN p.age_group = '19-21' THEN DATEADD(YEAR, -21, DATEFROMPARTS(p.enroll_year, 6, 30))
        END
    ) AS random_birthdate
FROM participants p
JOIN participants_demographics d
    ON p.participant_id = d.participant_id;
```

🧩 How it works

- `DATEFROMPARTS(p.enroll_year, 6, 30)` → June 30 of that enroll year.
- For example: `2020` → `2020-06-30`.
- We then shift back 16–21 years to get min/max birth years.
- `RAND(CHECKSUM(NEWID()))` creates a **unique random value per row**.
- `DATEADD(DAY, random × range, min_birthdate)` picks a day in that range.