Here is a column-by-column summary for all results columns in your retireplan project, with definitions and calculation summaries, based on your repo code:

**1. Year**

**Purpose:**  
The projection calendar year for each row.

**Calculation:**  
Starts at the user’s start\_year (input) and increments by 1 for each row.  
*See: engine.py, make\_years and run\_plan()*

**2. Person1\_Age**

**Purpose:**  
Age of the first person (primary) for the given year.

**Calculation:**  
Person1\_Age = Year - birth\_year\_person1  
*Rounded by round\_year utility.*

**3. Person2\_Age**

**Purpose:**  
Age of the second person (spouse/partner) for the given year.

**Calculation:**  
Person2\_Age = Year - birth\_year\_person2  
*Rounded, if present.*

**4. Filing**

**Purpose:**  
Tax filing status for that year ("MFJ" = Married Filing Jointly, or "Single").

**Calculation:**  
Set to "MFJ" if both persons are alive, "Single" otherwise.  
*See: engine.py, run\_plan()*

**5. Lifestyle**

**Purpose:**  
Spending “phase” in retirement (GoGo, SlowGo, NoGo).

**Calculation:**  
Determined by GoGo, SlowGo, NoGo phase durations and survivor status.  
*See: make\_years and phase assignment in engine.py*

**6. Base\_Spend**

**Purpose:**  
Core household spending (excluding taxes and special events).

**Calculation:**  
Base\_Spend = Total\_Spend - Taxes\_Due - Cash\_Events  
*Max with 0. Set in engine.py*

**7. Total\_Spend**

**Purpose:**  
Total outflows for the year, including spending, taxes, events.

**Calculation:**  
For Year 1: from user input (year1.spend).  
For later years: calculated sum of all outflows.  
*See: engine.py*

**8. Taxes\_Due**

**Purpose:**  
Total income taxes for the year.

**Calculation:**  
Computed from taxable income, draw types, Social Security, and filing status.  
*See: taxes.py, compute\_tax\_magi and engine.py*

* **Calculation:**  
  Ensure Taxes Due is based only on the taxable portions of IRA Draws, Roth Conversions, and Social Security in the model.

**9. Cash\_Events**

**Purpose:**  
Special/irregular cash flows (e.g., home sale, inheritance, car purchase).

**Calculation:**  
Sum of amount in cash\_events for the year (can be positive or negative).  
*Deducted from/added to appropriate account.*

Add a “notes” field for each Cash Event in the GUI, and save/load it in the config files.

**10. Social\_Security**

**Purpose:**  
Total Social Security benefits received that year.

**Calculation:**  
Based on start age, annual amount, inflation, and survivor status.  
*See: engine.py and inputs.py for start ages and amounts.*

**11. IRA\_Draw**

**Purpose:**  
Withdrawals from IRA accounts for spending/expenses.

**Calculation:**  
Planned/required draws, determined by need and withdrawal order (excludes RMD).  
*See: engine.py, accounts.py*

**12. Brokerage\_Draw**

**Purpose:**  
Withdrawals from taxable brokerage accounts.

**Calculation:**  
Planned draws to meet spending, per withdrawal order.  
*See: engine.py, accounts.py*

**13. Roth\_Draw**

**Purpose:**  
Withdrawals from Roth IRA accounts.

**Calculation:**  
Planned draws to meet spending, per withdrawal order.  
*See: engine.py, accounts.py*

**14. Roth\_Conversion**

**Purpose:**  
Amount converted from IRA to Roth IRA that year.

**Calculation:**  
User-specified or strategy-driven conversions (for tax management).  
*Reduces IRA, increases Roth. See: engine.py*

**15. RMD**

**Purpose:**  
Required Minimum Distributions (IRS-mandated) from IRAs after a certain age.

**Calculation:**  
IRS formula based on account balance and age, enforced at or after rmd\_start\_age.  
*See: engine.py, likely called via utility or directly calculated.*

**16. IRA\_Balance**

**Purpose:**  
End-of-year IRA account balance.

**Calculation:**  
Prior year balance + growth + inflows (minus draws, conversions, and RMD).  
*See: engine.py, accounts.py*

**17. Brokerage\_Balance**

**Purpose:**  
End-of-year brokerage account balance.

**Calculation:**  
Prior year balance + growth + inflows (minus draws and event outflows).  
*See: engine.py, accounts.py*

**18. Roth\_Balance**

**Purpose:**  
End-of-year Roth IRA balance.

**Calculation:**  
Prior year balance + growth + conversions (minus draws).  
*See: engine.py, accounts.py*

**19. Total\_Assets**

**Purpose:**  
Sum of all account end-of-year balances.

**Calculation:**  
Total\_Assets = IRA\_Balance + Brokerage\_Balance + Roth\_Balance  
*See: engine.py*

**Additional Notes**

* All monetary values are rounded to the nearest dollar by round\_dollar before display.
* The column schema is defined in schema.py as the canonical order and label set.
* Formatting for display (currency, etc.) is handled in results\_display.py.
* The calculations for each column are centralized in the run\_plan function in engine.py, which is the heart of the projection logic. Inputs are loaded and validated in inputs.py.