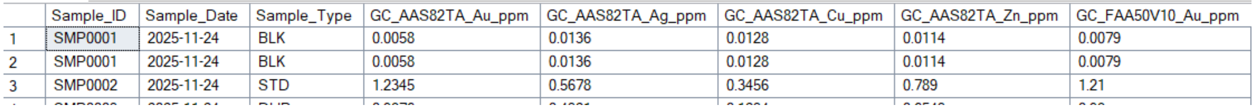
Step 1: Define the Table Structure

```sql
CREATE TABLE FireAssay (
    Sample_ID VARCHAR(20),
    Sample_Date DATE,
    Sample_Type VARCHAR(10),
    GC_AAS82TA_Au_ppm FLOAT,
    GC_AAS82TA_Ag_ppm FLOAT,
    GC_AAS82TA_Cu_ppm FLOAT,
    GC_AAS82TA_Zn_ppm FLOAT,
    GC_FAA50V10_Au_ppm FLOAT);

INSERT INTO FireAssay (
    Sample_ID, Sample_Date, Sample_Type,
    GC_AAS82TA_Au_ppm, GC_AAS82TA_Ag_ppm,
    GC_AAS82TA_Cu_ppm, GC_AAS82TA_Zn_ppm,
    GC_FAA50V10_Au_ppm
)
VALUES
('SMP0001', '2025-11-24', 'BLK', 0.0058, 0.0136, 0.0128, 0.0114, 0.0079),
('SMP0002', '2025-11-24', 'STD', 1.2345, 0.5678, 0.3456, 0.7890, 1.2100),
('SMP0003', '2025-11-24', 'DUP', 0.9876, 0.4321, 0.1234, 0.6543, 0.9800),
('SMP0004', '2025-11-24', 'RPT', 0.1122, 0.2233, 0.3344, 0.4455, 0.1100),
('SMP0005', '2025-11-24', 'UNK', 0.0001, 0.0002, 0.0003, 0.0004, 0.0001),
('SMP0006', '2025-11-24', 'STD', 2.3456, 1.2345, 0.9876, 1.1111, 2.3000),
('SMP0007', '2025-11-24', 'DUP', 1.1111, 0.9999, 0.8888, 0.7777, 1.1000),
('SMP0008', '2025-11-24', 'BLK', 0.0060, 0.0140, 0.0130, 0.0120, 0.0081),
('SMP0009', '2025-11-24', 'RPT', 0.2233, 0.3344, 0.4455, 0.5566, 0.2200),
('SMP0010', '2025-11-24', 'STD', 3.4567, 2.3456, 1.2345, 2.2222, 3.4000);
```



	Sample_ID	Sample_Date	Sample_Type	GC_AAS82TA_Au_ppm	GC_AAS82TA_Ag_ppm	GC_AAS82TA_Cu_ppm	GC_AAS82TA_Zn_ppm	GC_FAA50V10_Au_ppm
1	SMP0001	2025-11-24	BLK	0.0058	0.0136	0.0128	0.0114	0.0079
2	SMP0001	2025-11-24	BLK	0.0058	0.0136	0.0128	0.0114	0.0079
3	SMP0002	2025-11-24	STD	1.2345	0.5678	0.3456	0.789	1.21

Enrich the dataset with some variety. Here's a batch of **10 more rows** that includes:

- **Outliers**: unusually high values to simulate mineral-rich samples
- **NULLs**: missing data to reflect real-world lab gaps
- **Mixed types**: more DUP, RPT, UNK, and BLK samples

```sql
INSERT INTO FireAssay (
    Sample_ID, Sample_Date, Sample_Type,
    GC_AAS82TA_Au_ppm, GC_AAS82TA_Ag_ppm,
    GC_AAS82TA_Cu_ppm, GC_AAS82TA_Zn_ppm,
    GC_FAA50V10_Au_ppm
)
VALUES
('SMP0011', '2025-11-25', 'DUP', 0.8765, 0.6543, 0.4321, 0.3210, 0.8500),
('SMP0012', '2025-11-25', 'RPT', 0.0000, 0.0000, 0.0000, 0.0000, 0.0000),
('SMP0013', '2025-11-25', 'UNK', NULL, NULL, NULL, NULL, NULL),
('SMP0014', '2025-11-25', 'STD', 5.6789, 4.5678, 3.4567, 2.3456, 5.6000),
('SMP0015', '2025-11-25', 'BLK', 0.0055, 0.0125, 0.0115, 0.0105, 0.0075),
('SMP0016', '2025-11-25', 'DUP', 1.2345, NULL, 0.3456, NULL, 1.2000),
('SMP0017', '2025-11-25', 'RPT', NULL, 0.2233, NULL, 0.4455, NULL),
('SMP0018', '2025-11-25', 'STD', 6.7890, 5.6789, 4.5678, 3.4567, 6.7000),
('SMP0019', '2025-11-25', 'UNK', 0.1111, 0.2222, 0.3333, 0.4444, 0.1100),
('SMP0020', '2025-11-25', 'BLK', 0.0062, 0.0142, 0.0132, 0.0122, 0.0083);
```

Step 2: Insert Complex Rows

1. Duplicate Sample IDs

Sometimes samples are accidentally logged twice. This can cause integrity issues if `Sample_ID` is a primary key.

- We'll insert a row with an existing `Sample_ID` (e.g., `'SMP0002'`) to simulate a conflict.

2. Invalid Dates

Samples might be entered with future dates or malformed entries.

- We'll use a date like `'2099-01-01'` to simulate a suspicious future entry.

3. Extreme Outliers

Some samples show unusually high concentrations — either due to rich mineralization or lab error.

- We'll add values like `999.999` to simulate this.

4. NULLs in Critical Fields

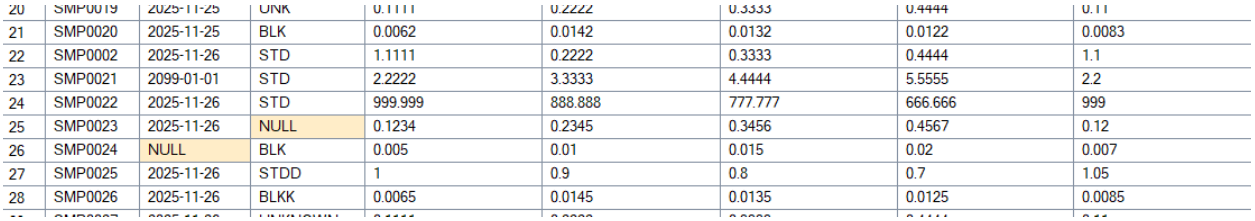
Missing values in key columns can affect analysis.

- We'll insert rows with `NULL` in `Sample_Type` or `Sample_Date`.

5. Inconsistent Sample Types

Typos or non-standard codes like `'STDD'`, `'BLKK'`, `'UNKNOWN'`.

```sql
INSERT INTO FireAssay (
    Sample_ID, Sample_Date, Sample_Type,
    GC_AAS82TA_Au_ppm, GC_AAS82TA_Ag_ppm,
    GC_AAS82TA_Cu_ppm, GC_AAS82TA_Zn_ppm,
    GC_FAA50V10_Au_ppm
)
VALUES
-- Duplicate Sample_ID
('SMP0002', '2025-11-26', 'STD', 1.1111, 0.2222, 0.3333, 0.4444, 1.1000),

-- Invalid future date
('SMP0021', '2099-01-01', 'STD', 2.2222, 3.3333, 4.4444, 5.5555, 2.2000),

-- Extreme outlier
('SMP0022', '2025-11-26', 'STD', 999.999, 888.888, 777.777, 666.666, 999.000),

-- NULL in Sample_Type
('SMP0023', '2025-11-26', NULL, 0.1234, 0.2345, 0.3456, 0.4567, 0.1200),

-- NULL in Sample_Date
('SMP0024', NULL, 'BLK', 0.0050, 0.0100, 0.0150, 0.0200, 0.0070),

-- Typo in Sample_Type
('SMP0025', '2025-11-26', 'STDD', 1.0000, 0.9000, 0.8000, 0.7000, 1.0500),
('SMP0026', '2025-11-26', 'BLKK', 0.0065, 0.0145, 0.0135, 0.0125, 0.0085),
('SMP0027', '2025-11-26', 'UNKNOWN', 0.1111, 0.2222, 0.3333, 0.4444, 0.1100);
```



20	SMP0019	2025-11-25	UNK	0.1111	0.2222	0.3333	0.4444	0.11
21	SMP0020	2025-11-25	BLK	0.0062	0.0142	0.0132	0.0122	0.0083
22	SMP0002	2025-11-26	STD	1.1111	0.2222	0.3333	0.4444	1.1
23	SMP0021	2099-01-01	STD	2.2222	3.3333	4.4444	5.5555	2.2
24	SMP0022	2025-11-26	STD	999.999	888.888	777.777	666.666	999
25	SMP0023	2025-11-26	NULL	0.1234	0.2345	0.3456	0.4567	0.12
26	SMP0024	NULL	BLK	0.005	0.01	0.015	0.02	0.007
27	SMP0025	2025-11-26	STDD	1	0.9	0.8	0.7	1.05
28	SMP0026	2025-11-26	BLKK	0.0065	0.0145	0.0135	0.0125	0.0085

Advanced Data Issues to Simulate

6. Duplicate Dates

Multiple samples logged with the exact same timestamp — could be batch entries or copy-paste errors.

- We'll reuse `'2025-11-26'` across several rows.

7. Mismatched Units

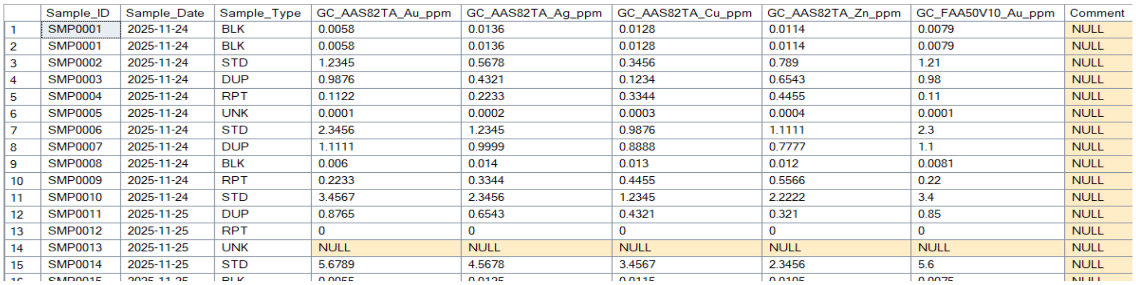
Values entered in the wrong unit (e.g., mg/kg instead of ppm). We'll simulate this by inserting unusually large or small values that don't match the expected range.

- Example: `0.000001` or `10000.0` for Au.

8. Flagged Samples with Comments

Some samples are flagged for review due to anomalies. We'll add a `Comment` column to hold notes like `"Possible contamination"` or `"Retest required"`.

```
--add a Comment column to table:
ALTER TABLE FireAssay
ADD Comment TEXT;
--insert more complex rows:
INSERT INTO FireAssay (
    Sample_ID, Sample_Date, Sample_Type,
    GC_AAS82TA_Au_ppm, GC_AAS82TA_Ag_ppm,
    GC_AAS82TA_Cu_ppm, GC_AAS82TA_Zn_ppm,
    GC_FAA50V10_Au_ppm, Comment
)
VALUES
-- Duplicate date
('SMP0028', '2025-11-26', 'STD', 1.2345, 0.5678, 0.3456, 0.7890, 1.2100, 'Batch
entry'),
('SMP0029', '2025-11-26', 'DUP', 0.9876, 0.4321, 0.1234, 0.6543, 0.9800, 'Same
timestamp as SMP0028'),

-- Mismatched units (too small)
('SMP0030', '2025-11-26', 'STD', 0.000001, 0.000002, 0.000003, 0.000004, 0.000001,
'Check units'),

-- Mismatched units (too large)
('SMP0031', '2025-11-26', 'STD', 10000.0, 9000.0, 8000.0, 7000.0, 9500.0,
'Suspiciously high values'),

-- Flagged for contamination
('SMP0032', '2025-11-26', 'RPT', 0.1234, 0.2345, 0.3456, 0.4567, 0.1200, 'Possible
contamination'),

-- Flagged for retest
('SMP0033', '2025-11-26', 'DUP', 1.1111, 0.9999, 0.8888, 0.7777, 1.1000, 'Retest
required');
```



	Sample_ID	Sample_Date	Sample_Type	GC_AAS82TA_Au_ppm	GC_AAS82TA_Ag_ppm	GC_AAS82TA_Cu_ppm	GC_AAS82TA_Zn_ppm	GC_FAA50V10_Au_ppm	Comment
1	SMP0001	2025-11-24	BLK	0.0058	0.0136	0.0128	0.0114	0.0079	NULL
2	SMP0001	2025-11-24	BLK	0.0058	0.0136	0.0128	0.0114	0.0079	NULL
3	SMP0002	2025-11-24	STD	1.2345	0.5678	0.3456	0.789	1.21	NULL
4	SMP0003	2025-11-24	DUP	0.9876	0.4321	0.1234	0.6543	0.98	NULL
5	SMP0004	2025-11-24	RPT	0.1122	0.2233	0.3344	0.4455	0.11	NULL
6	SMP0005	2025-11-24	UNK	0.0001	0.0002	0.0003	0.0004	0.0001	NULL
7	SMP0006	2025-11-24	STD	2.3456	1.2345	0.9876	1.1111	2.3	NULL
8	SMP0007	2025-11-24	DUP	1.1111	0.9999	0.8888	0.7777	1.1	NULL
9	SMP0008	2025-11-24	BLK	0.006	0.014	0.013	0.012	0.0081	NULL
10	SMP0009	2025-11-24	RPT	0.2233	0.3344	0.4455	0.5566	0.22	NULL
11	SMP0010	2025-11-24	STD	3.4567	2.3456	1.2345	2.2222	3.4	NULL
12	SMP0011	2025-11-25	DUP	0.8765	0.6543	0.4321	0.321	0.85	NULL
13	SMP0012	2025-11-25	RPT	0	0	0	0	0	NULL
14	SMP0013	2025-11-25	UNK	NULL	NULL	NULL	NULL	NULL	NULL
15	SMP0014	2025-11-25	STD	5.6789	4.5678	3.4567	2.3456	5.6	NULL
16	SMP0015	2025-11-25	BLK	0.0055	0.0135	0.0115	0.0105	0.0075	NULL

1. Compare Au values between GC_AAS82TA and GC_FAA50V10

We want to **compare gold assay values** from two different methods:

- `GC_AAS82TA_Au_ppm` (cyanide digestion)
- `GC_FAA50V10_Au_ppm` (fire assay)

And **flag samples** where the difference between these two methods is **greater than 20%**, which may indicate:

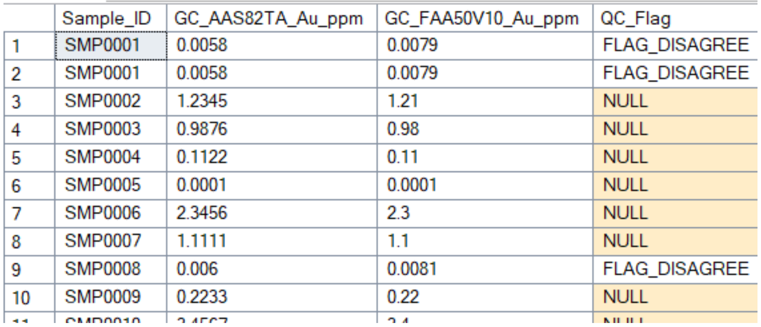
- Method disagreement
- Lab error
- Sample heterogeneity

```
SELECT Sample_ID,
       GC_AAS82TA_Au_ppm,
       GC_FAA50V10_Au_ppm,
       CASE
             WHEN ABS(GC_AAS82TA_Au_ppm - GC_FAA50V10_Au_ppm) /
             NULLIF(GC_AAS82TA_Au_ppm, 0) > 0.2
                THEN 'FLAG_DISAGREE'
           ELSE NULL
       END AS QC_Flag
FROM [dbo].[FireAssay]
WHERE GC_AAS82TA_Au_ppm IS NOT NULL AND GC_FAA50V10_Au_ppm IS NOT NULL;
```

- `ABS(...)`: Calculates the absolute difference between the two Au values.
- `NULLIF(GC_AAS82TA_Au_ppm, 0)`: Prevents division by zero.
- `> 0.2`: Flags if the difference is more than 20%.

`'FLAG_DISAGREE'`: the difference is large enough to trigger the flag

- `CASE ... THEN 'FLAG_DISAGREE'`: Adds a QC flag if the condition is met.
- `WHERE ... IS NOT NULL`: Filters out rows with missing Au values.



	Sample_ID	GC_AAS82TA_Au_ppm	GC_FAA50V10_Au_ppm	QC_Flag
1	SMP0001	0.0058	0.0079	FLAG_DISAGREE
2	SMP0001	0.0058	0.0079	FLAG_DISAGREE
3	SMP0002	1.2345	1.21	NULL
4	SMP0003	0.9876	0.98	NULL
5	SMP0004	0.1122	0.11	NULL
6	SMP0005	0.0001	0.0001	NULL
7	SMP0006	2.3456	2.3	NULL
8	SMP0007	1.1111	1.1	NULL
9	SMP0008	0.006	0.0081	FLAG_DISAGREE
10	SMP0009	0.2233	0.22	NULL
11	SMP0010	3.4567	3.4	NULL

2. We want to flag samples **where either gold value is** below the detection limit **— in this case,** 0.01 ppm**.**

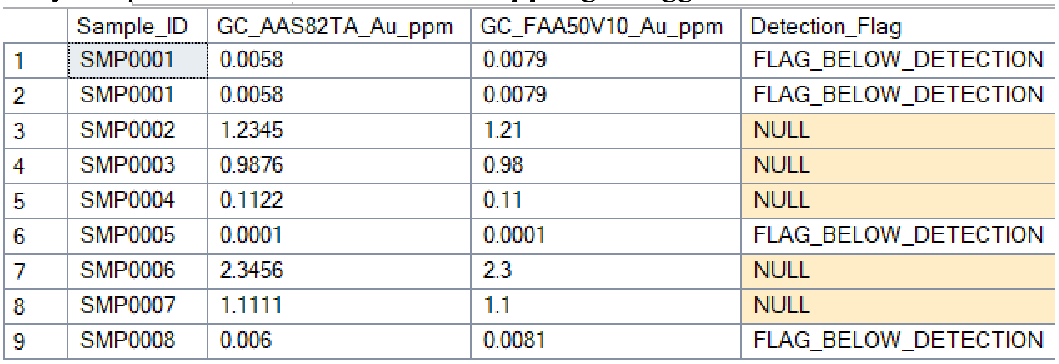
This helps identify:

- Non-detects
- Possibly unreliable measurements
- Samples that may need re-analysis or special handling

```
SELECT Sample_ID,
       GC_AAS82TA_Au_ppm,
       GC_FAA50V10_Au_ppm,
       CASE
           WHEN GC_AAS82TA_Au_ppm < 0.01 OR GC_FAA50V10_Au_ppm < 0.01
                THEN 'FLAG_BELOW_DETECTION'
           ELSE NULL
       END AS Detection_Flag
FROM [dbo].[FireAssay]
WHERE GC_AAS82TA_Au_ppm IS NOT NULL AND GC_FAA50V10_Au_ppm IS NOT NULL;
```

- `GC_AAS82TA_Au_ppm < 0.01`: Checks if cyanide assay is below detection.
- `GC_FAA50V10_Au_ppm < 0.01`: Checks if fire assay is below detection.
- `OR`: Flags if **either** value is below 0.01 ppm.
- `CASE ... THEN 'FLAG_BELOW_DETECTION'`: Adds a detection flag.
- `WHERE ... IS NOT NULL`: Ensures both values exist before evaluating.

Only samples with **either value < 0.01 ppm** get flagged.



	Sample_ID	GC_AAS82TA_Au_ppm	GC_FAA50V10_Au_ppm	Detection_Flag
1	SMP0001	0.0058	0.0079	FLAG_BELOW_DETECTION
2	SMP0001	0.0058	0.0079	FLAG_BELOW_DETECTION
3	SMP0002	1.2345	1.21	NULL
4	SMP0003	0.9876	0.98	NULL
5	SMP0004	0.1122	0.11	NULL
6	SMP0005	0.0001	0.0001	FLAG_BELOW_DETECTION
7	SMP0006	2.3456	2.3	NULL
8	SMP0007	1.1111	1.1	NULL
9	SMP0008	0.006	0.0081	FLAG_BELOW_DETECTION

2-1- combines both flags into a single output.

-We want to:

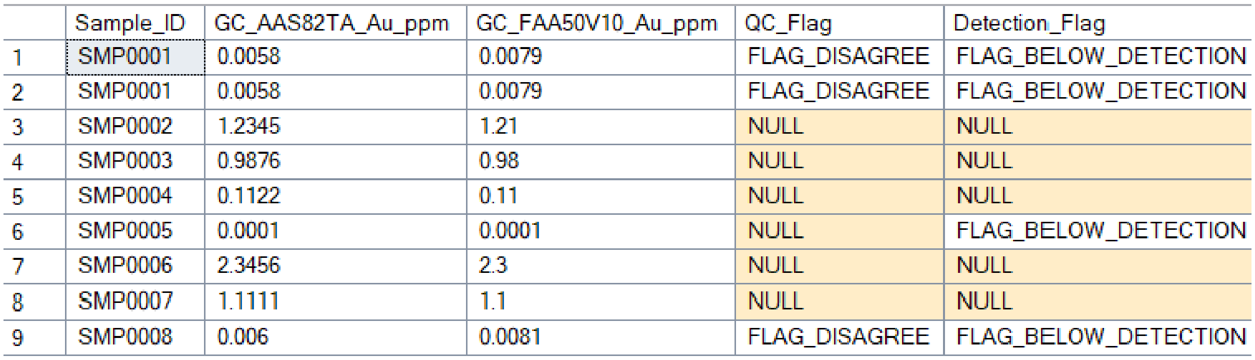
- Compare assay disagreement (from Query #1)
- Check detection limits (from Query #2)
- And report both flags in one result set

This gives a full QC picture for each sample.

```
SELECT Sample_ID,
       GC_AAS82TA_Au_ppm,
       GC_FAA50V10_Au_ppm,
       CASE
           WHEN ABS(GC_AAS82TA_Au_ppm - GC_FAA50V10_Au_ppm) /
NULLIF(GC_AAS82TA_Au_ppm, 0) > 0.2
                THEN 'FLAG_DISAGREE'
           ELSE NULL
       END AS QC_Flag,
       CASE
```

```
           WHEN GC_AAS82TA_Au_ppm < 0.01 OR GC_FAA50V10_Au_ppm < 0.01
                THEN 'FLAG_BELOW_DETECTION'
           ELSE NULL
       END AS Detection_Flag
FROM [dbo].[FireAssay]
WHERE GC_AAS82TA_Au_ppm IS NOT NULL AND GC_FAA50V10_Au_ppm IS NOT NULL;
```

- First CASE: Flags disagreement (>20% difference)
- Second CASE: Flags values below detection (<0.01 ppm)
- Both flags are shown in separate columns
- WHERE: Filters out nulls to ensure valid comparisons



	Sample_ID	GC_AAS82TA_Au_ppm	GC_FAA50V10_Au_ppm	QC_Flag	Detection_Flag
1	SMP0001	0.0058	0.0079	FLAG_DISAGREE	FLAG_BELOW_DETECTION
2	SMP0001	0.0058	0.0079	FLAG_DISAGREE	FLAG_BELOW_DETECTION
3	SMP0002	1.2345	1.21	NULL	NULL
4	SMP0003	0.9876	0.98	NULL	NULL
5	SMP0004	0.1122	0.11	NULL	NULL
6	SMP0005	0.0001	0.0001	NULL	FLAG_BELOW_DETECTION
7	SMP0006	2.3456	2.3	NULL	NULL
8	SMP0007	1.1111	1.1	NULL	NULL
9	SMP0008	0.006	0.0081	FLAG_DISAGREE	FLAG_BELOW_DETECTION

3. Identify Outliers in GC_AAS82TA_Au_ppm

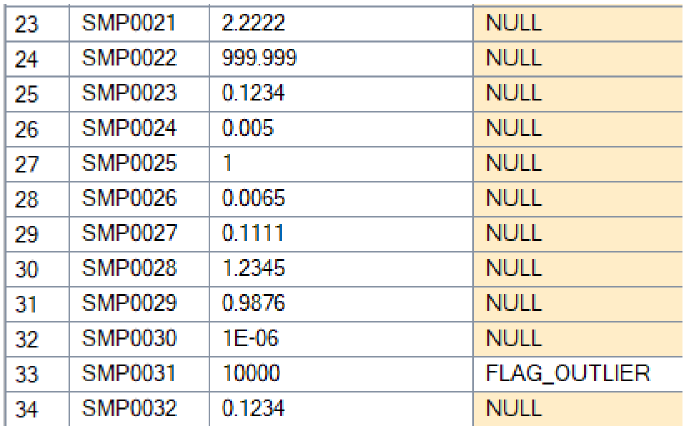
Flag samples where GC_AAS82TA_Au_ppm is **greater than 3 standard deviations above the mean**, which may indicate:

- Sample contamination
- Unexpected mineralization
- Lab error

```
WITH Stats AS (
    SELECT
        AVG(GC_AAS82TA_Au_ppm) AS Mean_Au,
        STDEV(GC_AAS82TA_Au_ppm) AS SD_Au
    FROM [dbo].[FireAssay]
    WHERE GC_AAS82TA_Au_ppm IS NOT NULL
)
SELECT f.Sample_ID,
       f.GC_AAS82TA_Au_ppm,
       CASE
           WHEN f.GC_AAS82TA_Au_ppm > s.Mean_Au + 3 * s.SD_Au THEN 'FLAG_OUTLIER'
           ELSE NULL
       END AS Outlier_Flag
FROM [dbo].[FireAssay] f
CROSS JOIN Stats s;
```

- WITH Stats AS (...): This is a Common Table Expression (CTE) — a temporary result set you can reference later.
- AVG(...): Calculates the average gold concentration.
- STDEV(...): Calculates the standard deviation, which measures how spread out the values are.
- WHERE GC_AAS82TA_Au_ppm IS NOT NULL: Ensures we only use valid (non-missing) values in the calculation.
- Second SELECT: Pulls out each sample's ID and gold value.

- `CASE WHEN ... THEN 'FLAG_OUTLIER'`: Flags samples where the gold value is **more than 3 standard deviations above the mean**.
- `CROSS JOIN Stats`: This joins every row in `FireAssay` with the single-row result from `Stats`, so we can use the calculated mean and SD in every comparison.
- `ELSE NULL`: If the value isn't an outlier, no flag is applied.



23	SMP0021	2.2222	NULL
24	SMP0022	999.999	NULL
25	SMP0023	0.1234	NULL
26	SMP0024	0.005	NULL
27	SMP0025	1	NULL
28	SMP0026	0.0065	NULL
29	SMP0027	0.1111	NULL
30	SMP0028	1.2345	NULL
31	SMP0029	0.9876	NULL
32	SMP0030	1E-06	NULL
33	SMP0031	10000	FLAG_OUTLIER
34	SMP0032	0.1234	NULL

Using 3 sigma to flag high Au values helps:

- Detect **sample contamination**
- Highlight **potential mineralization zones**
- Catch **lab errors or mislabeling**

Upper and **lower bound checks** using the 3 sigma rule:

```
WITH Stats AS (
    SELECT
        AVG(GC_AAS82TA_Au_ppm) AS Mean_Au,
        STDEV(GC_AAS82TA_Au_ppm) AS SD_Au
    FROM [dbo].[FireAssay]
    WHERE GC_AAS82TA_Au_ppm IS NOT NULL
)
SELECT f.Sample_ID,
       f.GC_AAS82TA_Au_ppm,
       CASE
             WHEN f.GC_AAS82TA_Au_ppm > s.Mean_Au + 3 * s.SD_Au THEN
             'FLAG_HIGH_OUTLIER'
             WHEN f.GC_AAS82TA_Au_ppm < s.Mean_Au - 3 * s.SD_Au THEN
             'FLAG_LOW_OUTLIER'
           ELSE NULL
       END AS Outlier_Flag
FROM [dbo].[FireAssay] f

 CROSS JOIN Stats s;
```

4. Flag Missing GC_AAS82TA_Au_ppm Values

Identify samples where the gold value is **missing** (i.e., `NULL`) so they can be:

- Investigated
- Reanalyzed
- Excluded from certain calculations

```
SELECT Sample_ID,
       GC_AAS82TA_Au_ppm,
       CASE
           WHEN GC_AAS82TA_Au_ppm IS NULL THEN 'FLAG_MISSING_AU'
           ELSE NULL
       END AS Missing_Flag

 FROM [dbo].[FireAssay];
```

- `GC_AAS82TA_Au_ppm IS NULL`: Checks for missing values
- `CASE`: Flags those rows with `'FLAG_MISSING_AU'`
- `ELSE NULL`: Leaves other rows unflagged

Missing values can skew:

- Averages
- Standard deviations
- Geostatistical models

Flagging them helps ensure data integrity and proper handling in downstream analysis.

5. Duplicate Reproducibility Check

This step is all about evaluating how well **duplicate samples** agree with each other. In geochemistry QA/QC, duplicates are used to assess **precision** — if two samples from the same location yield very different results, that's a red flag.

For each pair of duplicates:

- Compare **original vs. duplicate** assay values
- Calculate **absolute difference** and **relative percent difference (RPD)**
- Flag any pairs that exceed acceptable thresholds
  - For each pair of duplicates:
  - Compare original vs. duplicate assay values
  - Calculate absolute difference and relative percent difference (RPD)
  - Flag any pairs that exceed acceptable thresholds

```
SELECT
    Original.Sample_ID AS Original_ID,
    Duplicate.Sample_ID AS Duplicate_ID,
    Original.GC_AAS82TA_Au_ppm AS Original_Au,
    Duplicate.GC_AAS82TA_Au_ppm AS Duplicate_Au,
    ABS(Original.GC_AAS82TA_Au_ppm - Duplicate.GC_AAS82TA_Au_ppm) AS Abs_Diff,
    CASE
        WHEN Original.GC_AAS82TA_Au_ppm + Duplicate.GC_AAS82TA_Au_ppm = 0 THEN NULL
        ELSE ABS(Original.GC_AAS82TA_Au_ppm - Duplicate.GC_AAS82TA_Au_ppm) * 100.0 /
```

```
             ((Original.GC_AAS82TA_Au_ppm + Duplicate.GC_AAS82TA_Au_ppm) / 2)
    END AS RPD,
    CASE
        WHEN ABS(Original.GC_AAS82TA_Au_ppm - Duplicate.GC_AAS82TA_Au_ppm) > 0.1
THEN 'RED_FLAG'
        WHEN ABS(Original.GC_AAS82TA_Au_ppm - Duplicate.GC_AAS82TA_Au_ppm) > 0.05
THEN 'YELLOW_FLAG'
        ELSE 'GREEN'
    END AS QC_Status
FROM
    FireAssay AS Original
JOIN
    FireAssay AS Duplicate
    ON Duplicate.Sample_ID = Original.Sample_ID + '_DUP'
WHERE
    Original.Sample_Type = 'Original' AND Duplicate.Sample_Type = 'Duplicate';
```

Description:

```
Original.GC_AAS82TA_Au_ppm AS Original_Au,
    Duplicate.GC_AAS82TA_Au_ppm AS Duplicate_Au,
```

- Pulls the gold assay values (`Au_ppm`) from both original and duplicate samples.
- Renames them for easier reading.

```
ABS(Original.GC_AAS82TA_Au_ppm - Duplicate.GC_AAS82TA_Au_ppm) AS Abs_Diff,
```

- Calculates the **absolute difference** between original and duplicate values.
- Useful for spotting large deviations.

```
CASE
        WHEN Original.GC_AAS82TA_Au_ppm + Duplicate.GC_AAS82TA_Au_ppm = 0
THEN NULL
```

- Prevents division by zero when both values are zero.

```
ELSE ABS(Original.GC_AAS82TA_Au_ppm - Duplicate.GC_AAS82TA_Au_ppm) * 100.0 /
             ((Original.GC_AAS82TA_Au_ppm + Duplicate.GC_AAS82TA_Au_ppm) / 2)
    END AS RPD,
```

- Calculates **Relative Percent Difference (RPD)**:

$$\mathrm{RPD} = \frac{|\mathrm{Original} - \mathrm{Duplicate}|}{\mathrm{Average\ of\ both}} \times 100$$

- This shows how far apart the values are in percentage terms.

```
CASE
        WHEN ABS(Original.GC_AAS82TA_Au_ppm - Duplicate.GC_AAS82TA_Au_ppm) >
0.1 THEN 'RED_FLAG'
```

```
        WHEN ABS(Original.GC_AAS82TA_Au_ppm - Duplicate.GC_AAS82TA_Au_ppm) >
0.05 THEN 'YELLOW_FLAG'
        ELSE 'GREEN'
    END AS QC_Status
```

- Flags the result based on how big the difference is:
    - `> 0.1 ppm`: serious issue → `RED_FLAG`
    - `0.05–0.1 ppm`: moderate issue → `YELLOW_FLAG`
    - `≤ 0.05 ppm`: acceptable → `GREEN`

```
FROM
    FireAssay AS Original
JOIN
    FireAssay AS Duplicate
    ON Duplicate.Sample_ID = Original.Sample_ID + '_DUP'
```

- Joins the table to itself:
    - `Original` and `Duplicate` are aliases for the same table.
    - Matches samples where the duplicate has the same ID plus `'_DUP'`.

```
WHERE
    Original.Sample_Type = 'Original' AND Duplicate.Sample_Type =
'Duplicate';
```

- Filters the rows:
    - Ensures you're only comparing true original–duplicate pairs.

Summary

This query:

- Matches original and duplicate samples
- Calculates differences
- Flags issues with intuitive labels

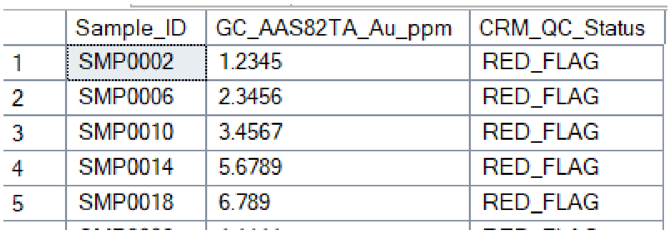
6. Certified Reference Material (CRM) Check

Goal:

- Compare measured values from CRM samples to their **certified values** and flag any deviations beyond tolerance.
- CRM samples are labeled with `Sample_Type = 'STD'`
- Certified value for Au = `1.00 ppm` (example)
- Acceptable tolerance = ±10%

```
SELECT Sample_ID,
       GC_AAS82TA_Au_ppm,
       CASE
           WHEN GC_AAS82TA_Au_ppm IS NULL THEN 'RED_FLAG'
           WHEN ABS(GC_AAS82TA_Au_ppm - 1.00) / 1.00 > 0.10 THEN 'RED_FLAG'
           WHEN ABS(GC_AAS82TA_Au_ppm - 1.00) / 1.00 > 0.05 THEN 'YELLOW_FLAG'
           ELSE 'GREEN'
       END AS CRM_QC_Status
FROM [dbo].[FireAssay]

      WHERE Sample_Type = 'STD';
```

- Flags CRM samples based on how far they deviate from the certified value
- You can adjust the certified value and tolerance per batch or CRM type
  - ➢ GC_AAS82TA_Au_ppm: This is the measured gold concentration from your sample.
  - ➢ 1.00: This is the certified value from the CRM (Certified Reference Material).
  - ➢ GC_AAS82TA_Au_ppm - 1.00: Calculates the difference between the measured and certified value.
  - ➢ ABS(...): Takes the absolute value of that difference — so it doesn't matter if the result is above or below the certified value.
  - ➢ / 1.00: Normalizes the difference by dividing by the certified value — this gives the **relative error**.
  - ➢ > 0.10: Checks if the relative error is greater than **10%**.
  - ➢ THEN 'RED_FLAG': If the error exceeds 10%, the sample fails the QC check and gets flagged.



	Sample_ID	GC_AAS82TA_Au_ppm	CRM_QC_Status
1	SMP0002	1.2345	RED_FLAG
2	SMP0006	2.3456	RED_FLAG
3	SMP0010	3.4567	RED_FLAG
4	SMP0014	5.6789	RED_FLAG
5	SMP0018	6.789	RED_FLAG

This check ensures your lab results are **accurate** and **within acceptable limits** compared to a trusted standard. If the deviation is too large, it could indicate:

- Instrument drift
- Sample prep error
- Calibration issues

7. Blank Sample Check

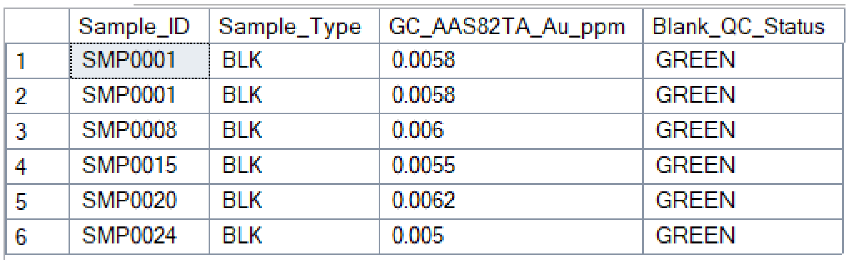
Goal:

Ensure blank samples show **no detectable gold** — values should be **near zero**.

Assumptions:

- Blank samples are labeled with `Sample_Type = 'BLK'`
- Detection limit = `0.01 ppm`

```
SELECT Sample_ID,
        Sample_Type,
        GC_AAS82TA_Au_ppm,
        CASE
            WHEN GC_AAS82TA_Au_ppm IS NULL THEN 'RED_FLAG'
            WHEN GC_AAS82TA_Au_ppm > 0.01 THEN 'RED_FLAG'
            ELSE 'GREEN'
        END AS Blank_QC_Status
FROM [dbo].[FireAssay]

WHERE Sample_Type = 'BLK';
```



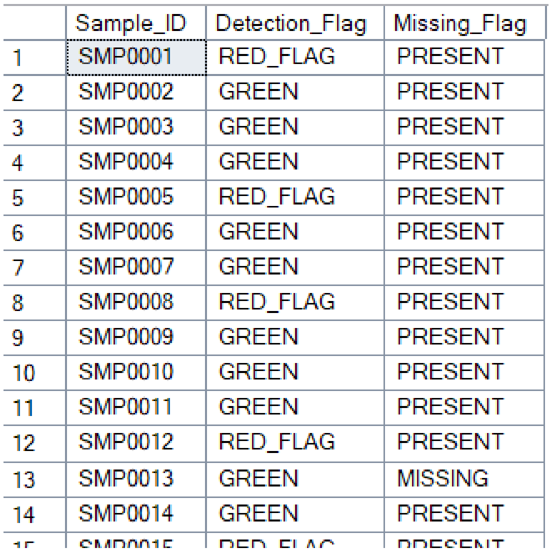
	Sample_ID	Sample_Type	GC_AAS82TA_Au_ppm	Blank_QC_Status
1	SMP0001	BLK	0.0058	GREEN
2	SMP0001	BLK	0.0058	GREEN
3	SMP0008	BLK	0.006	GREEN
4	SMP0015	BLK	0.0055	GREEN
5	SMP0020	BLK	0.0062	GREEN
6	SMP0024	BLK	0.005	GREEN

Wrap all together

Query to View Cleaned Sample Records

```
-- This query builds two CTEs:
-- DetectionCheck: Flags samples with low gold values (< 0.01 ppm) as RED_FLAG
-- MissingCheck: Flags samples with NULL gold values as MISSING
-- Then joins them and removes duplicate rows

WITH DetectionCheck AS (
    SELECT
        Sample_ID,
        CASE
            WHEN GC_AAS82TA_Au_ppm < 0.01 THEN 'RED_FLAG'
            ELSE 'GREEN'
        END AS Detection_Flag
    FROM dbo.FireAssay
),
MissingCheck AS (
    SELECT
        Sample_ID,
        CASE
            WHEN GC_AAS82TA_Au_ppm IS NULL THEN 'MISSING'
            ELSE 'PRESENT'
        END AS Missing_Flag
    FROM dbo.FireAssay
)
SELECT DISTINCT
    d.Sample_ID,
    d.Detection_Flag,
    m.Missing_Flag
FROM DetectionCheck d
```

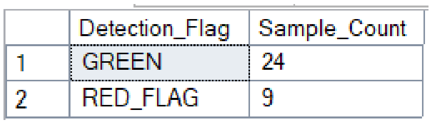
```
JOIN MissingCheck m ON d.Sample_ID = m.Sample_ID;
```



	Sample_ID	Detection_Flag	Missing_Flag
1	SMP0001	RED_FLAG	PRESENT
2	SMP0002	GREEN	PRESENT
3	SMP0003	GREEN	PRESENT
4	SMP0004	GREEN	PRESENT
5	SMP0005	RED_FLAG	PRESENT
6	SMP0006	GREEN	PRESENT
7	SMP0007	GREEN	PRESENT
8	SMP0008	RED_FLAG	PRESENT
9	SMP0009	GREEN	PRESENT
10	SMP0010	GREEN	PRESENT
11	SMP0011	GREEN	PRESENT
12	SMP0012	RED_FLAG	PRESENT
13	SMP0013	GREEN	MISSING
14	SMP0014	GREEN	PRESENT
15	SMP0015	RED_FLAG	PRESENT

Query to Count Samples by Detection Flag

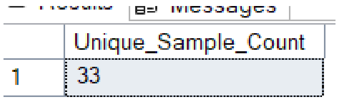
```
-- This query counts how many samples fall under each Detection_Flag category
-- RED_FLAG = gold value < 0.01 ppm
-- GREEN = gold value >= 0.01 ppm

WITH DetectionCheck AS (
    SELECT
        Sample_ID,
        CASE
            WHEN GC_AAS82TA_Au_ppm < 0.01 THEN 'RED_FLAG'
            ELSE 'GREEN'
        END AS Detection_Flag
    FROM dbo.FireAssay
),
MissingCheck AS (
    SELECT
        Sample_ID,
        CASE
            WHEN GC_AAS82TA_Au_ppm IS NULL THEN 'MISSING'
            ELSE 'PRESENT'
        END AS Missing_Flag
    FROM dbo.FireAssay
)
SELECT
    d.Detection_Flag,
    COUNT(DISTINCT d.Sample_ID) AS Sample_Count
FROM DetectionCheck d
JOIN MissingCheck m ON d.Sample_ID = m.Sample_ID

GROUP BY d.Detection_Flag;
```



	Detection_Flag	Sample_Count
1	GREEN	24
2	RED_FLAG	9

Query to Count Unique Samples

```
-- This query counts the total number of unique samples
-- It joins DetectionCheck and MissingCheck and counts distinct Sample_IDs

WITH DetectionCheck AS (
    SELECT
        Sample_ID,
        CASE
            WHEN GC_AAS82TA_Au_ppm < 0.01 THEN 'RED_FLAG'
            ELSE 'GREEN'
        END AS Detection_Flag
    FROM dbo.FireAssay
),
MissingCheck AS (
    SELECT
        Sample_ID,
        CASE
            WHEN GC_AAS82TA_Au_ppm IS NULL THEN 'MISSING'
            ELSE 'PRESENT'
        END AS Missing_Flag
    FROM dbo.FireAssay
)
SELECT COUNT(DISTINCT d.Sample_ID) AS Unique_Sample_Count
FROM DetectionCheck d

JOIN MissingCheck m ON d.Sample_ID = m.Sample_ID;
```



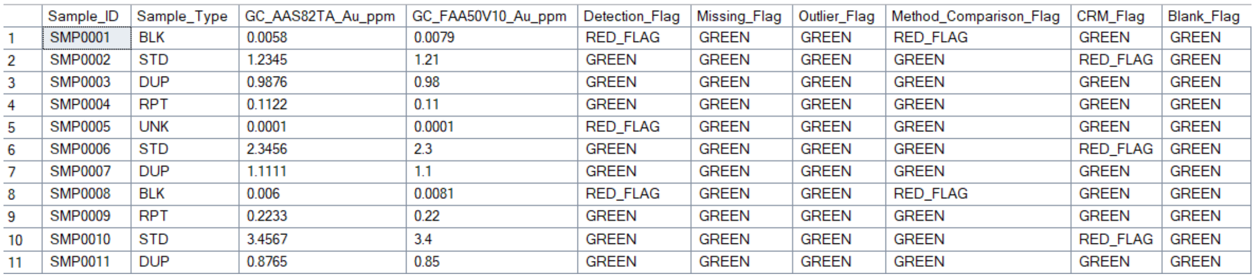
	Unique_Sample_Count
1	33

Full Query with Deduplication

```
-- Step 0: Deduplicate base data by Sample_ID
WITH BaseSamples AS (
    SELECT
        Sample_ID,
        Sample_Type,
        GC_AAS82TA_Au_ppm,
        GC_FAA50V10_Au_ppm
    FROM (
        SELECT *,
               ROW_NUMBER() OVER (PARTITION BY Sample_ID ORDER BY Sample_ID) AS rn
        FROM dbo.FireAssay
    ) AS numbered
    WHERE rn = 1  -- Keep only the first occurrence of each Sample_ID
),

-- Step 1: Flag samples below detection limit
```

```sql
DetectionCheck AS (
    SELECT Sample_ID,
           CASE
               WHEN GC_AAS82TA_Au_ppm < 0.01 OR GC_FAA50V10_Au_ppm < 0.01 THEN
'RED_FLAG'
               ELSE 'GREEN'
           END AS Detection_Flag
    FROM BaseSamples
),

-- Step 2: Flag samples with missing values
MissingCheck AS (
    SELECT Sample_ID,
           CASE
               WHEN GC_AAS82TA_Au_ppm IS NULL THEN 'RED_FLAG'
               ELSE 'GREEN'
           END AS Missing_Flag
    FROM BaseSamples
),

-- Step 3: Flag statistical outliers
OutlierCheck AS (
    SELECT Sample_ID,
           CASE
               WHEN GC_AAS82TA_Au_ppm > 5.0 THEN 'RED_FLAG'
               ELSE 'GREEN'
           END AS Outlier_Flag
    FROM BaseSamples
),

-- Step 4: Flag method disagreement (>20% difference)
MethodComparison AS (
    SELECT Sample_ID,
           CASE
               WHEN ABS(GC_AAS82TA_Au_ppm - GC_FAA50V10_Au_ppm) /
NULLIF(GC_FAA50V10_Au_ppm, 0) > 0.20 THEN 'RED_FLAG'
               ELSE 'GREEN'
           END AS Method_Comparison_Flag
    FROM BaseSamples
),

-- Step 5: Flag CRM samples outside ±10% of expected value (1.00 ppm)
CRMCheck AS (
    SELECT Sample_ID,
           CASE
               WHEN Sample_Type = 'STD' AND ABS(GC_AAS82TA_Au_ppm - 1.00) / 1.00 >
0.10 THEN 'RED_FLAG'
               ELSE 'GREEN'
           END AS CRM_Flag
    FROM BaseSamples
),

-- Step 6: Flag blank samples with contamination (>0.01 ppm)
BlankCheck AS (
    SELECT Sample_ID,
           CASE
               WHEN Sample_Type = 'BLK' AND GC_AAS82TA_Au_ppm > 0.01 THEN 'RED_FLAG'
               ELSE 'GREEN'
```

```sql
            END AS Blank_Flag
    FROM BaseSamples
)

-- Final step: Combine all flags into one dashboard table
SELECT
    b.Sample_ID,
    b.Sample_Type,
    b.GC_AAS82TA_Au_ppm,
    b.GC_FAA50V10_Au_ppm,
    d.Detection_Flag,
    m.Missing_Flag,
    o.Outlier_Flag,
    mc.Method_Comparison_Flag,
    crm.CRM_Flag,
    bl.Blank_Flag
FROM BaseSamples b
LEFT JOIN DetectionCheck d ON b.Sample_ID = d.Sample_ID
LEFT JOIN MissingCheck m ON b.Sample_ID = m.Sample_ID
LEFT JOIN OutlierCheck o ON b.Sample_ID = o.Sample_ID
LEFT JOIN MethodComparison mc ON b.Sample_ID = mc.Sample_ID
LEFT JOIN CRMCheck crm ON b.Sample_ID = crm.Sample_ID

LEFT JOIN BlankCheck bl ON b.Sample_ID = bl.Sample_ID;
```



	Sample_ID	Sample_Type	GC_AAS82TA_Au_ppm	GC_FAA50V10_Au_ppm	Detection_Flag	Missing_Flag	Outlier_Flag	Method_Comparison_Flag	CRM_Flag	Blank_Flag
1	SMP0001	BLK	0.0058	0.0079	RED_FLAG	GREEN	GREEN	RED_FLAG	GREEN	GREEN
2	SMP0002	STD	1.2345	1.21	GREEN	GREEN	GREEN	GREEN	RED_FLAG	GREEN
3	SMP0003	DUP	0.9876	0.98	GREEN	GREEN	GREEN	GREEN	GREEN	GREEN
4	SMP0004	RPT	0.1122	0.11	GREEN	GREEN	GREEN	GREEN	GREEN	GREEN
5	SMP0005	UNK	0.0001	0.0001	RED_FLAG	GREEN	GREEN	GREEN	GREEN	GREEN
6	SMP0006	STD	2.3456	2.3	GREEN	GREEN	GREEN	GREEN	RED_FLAG	GREEN
7	SMP0007	DUP	1.1111	1.1	GREEN	GREEN	GREEN	GREEN	GREEN	GREEN
8	SMP0008	BLK	0.006	0.0081	RED_FLAG	GREEN	GREEN	RED_FLAG	GREEN	GREEN
9	SMP0009	RPT	0.2233	0.22	GREEN	GREEN	GREEN	GREEN	GREEN	GREEN
10	SMP0010	STD	3.4567	3.4	GREEN	GREEN	GREEN	GREEN	RED_FLAG	GREEN
11	SMP0011	DUP	0.8765	0.85	GREEN	GREEN	GREEN	GREEN	GREEN	GREEN