SAS Concepts and Vocabulary:

1. DATA Step in SAS

- **Definition**: A `DATA` step is used to create or modify datasets in SAS.
- **Example**

```
data HRRM.Jan_Insured_SELECTED;
  set HRRM.Jan_insured_act;
  keep ID DOB EE_Retiree_Age Last_Hire_Date;
run;
```

`data`: Creates a new dataset.
`set`: Reads an existing dataset.
`keep`: Selects specific columns to retain.
`run`: Executes the data step.

2. PROC SQL in SAS

- **Definition**: `PROC SQL` is used to query, manipulate, and summarize data using SQL syntax in SAS.
- **Example**

```
proc sql;
  select
    sum(case when Benefit_program = 'Base' then 1 else 0 end) as total_base,
    sum(case when Benefit_program = 'Plus' then 1 else 0 end) as total_plus
  from matched_tcds;
quit;
```

Purpose: Count how many records have `Benefit_program = 'Base'` and `Benefit_program = 'Plus'`.

Key SQL Functions in SAS:

- **SELECT**: Picks the columns or expressions you want to retrieve.
- **FROM**: Specifies the dataset from which data is pulled.
- **SUM**: Totals numeric values.
- **COUNT**: Counts the number of rows or non-missing values.
- **AVG**: Calculates the average of numeric values.

- **MAX** / **MIN**: Finds the highest or lowest value in a column.
- **CASE WHEN**: Applies conditional logic.
- **GROUP BY**: Groups data by categories.
- **WHERE**: Filters the data.
- **HAVING**: Filters data after grouping.
- **AS**: Renames or labels a result.

3. Merging Datasets in SAS

- **Definition**: Combining two datasets based on a common key.
- **Example**

```
data HRRM.Final_combine_data;
  merge HRRM.Jan_Insured_Act (in=a keep=ID_name Gender DOB)
      HRRM.June_TCDRS (in=b keep=ID_name Co TCDRS_Earning);
  by ID_name;
  if a and b;
run;
```

`merge`: Combines two datasets.
`by`: Specifies the common key (`ID_name`).
`if a and b;`: Keeps only records that exist in both datasets.

4. PROC PRINT in SAS

- **Definition**: Used to display the contents of a dataset or specific variables.
- Example:

```
proc print data=HRRM.final_combine_data;
  var ID_name Gender DOB Last_Hire_Date;
run;
```

`var`: Specifies which columns to display.

5. Conditional Aggregation with SQL in SAS

- **Definition**: Using `SUM(CASE WHEN ...)` to calculate totals based on specific conditions.
- **Example**:

```
sum(case when Benefit_program = 'Base' then 1 else 0 end) as total_base
```

Purpose: Count how many records meet a specific condition (e.g., `Benefit_program = 'Base'`).

SQL Functions in SAS:

- **SELECT**
- **FROM**
- **SUM**
- **COUNT**
- **AVG**
- **MAX**
- **MIN**
- **CASE WHEN**
- **GROUP BY**
- **WHERE**
- **HAVING**
- **AS**