GEN AI Mapping Agent

Prompt Given for Mapping Agent to Gemini to Map and Create a Transformation Query from Source to Target. Source Definition and Target Definition Stored in Json Object.

```
Prompt = f'''
You are an accurate sql mapping assistant
write a big query sql that uses the sources and maps to the target definition.
The Jsons contain the metadata of the sources and targets.
<EXAMPLE>
Json for the Sources = {json_data_sources1}
Json for the Targets = {json_data_target1}
Query :
-- CTE for Latest Department
WITH LatestDepartment AS (
  SELECT ee.ID AS EmployeeID,
        ed.DEPID AS DepartmentID,
        d.name AS DepartmentName,
.....................
<EXAMPLE>
Json for the Sources = {json_data_sources2}
Json for the Targets = {json_data_target2}
'''
```

GEN AI Mapping Agent

Sources Profile - json_data_sources2

SourceTableName	SourceColumnName	SourceColumnDataType	UniqueDataValues	TableDescription	Nullable
Ptnt101	ID	int	1,2	patient table	N
Ptnt101	firstname	varchar	Raj,Kamal	patient table	Y
Ptnt101	lastname	varchar	Kumar,Hassan	patient table	Y
PTINS101	ptntid	int	1,2	patient insurance bridge	N
PTINS101	insid	int	1,2	patient insurance bridge	N
PTINS101	ValidFrom	Date	1,2	patient insurance bridge	N
ADD101	ID	int	1,2	address table	N
ADD101	Address Line 1	varchar	504 church street,101 gated road	address table	Y
ADD101	City	varchar	mumbai,pune	address table	Y
ADD101	State	varchar	maharastra,maharastra	address table	Y
ADD101	Zip	varchar	400050 , 400093	address table	Y
ADD101	ValidFrom	Date	10-10-2024,11-10-2024	address table	N
ADD101	AddType	varchar	H,I	address table	N
ADD101	INSID	int	1,2	address table	Y
ADD101	ptntid	int	1,2	address table	N
INS101	Id	int	1,2	insurance table	N
INS101	name	varchar	HomeInsuranceforAll,Grownups	insurance table	N

Target Profile - json_data_target2

TargetTableName	TargetColumnName	TargetExplanation	TargetColumnDataT
PATIENT	PatientId	unique patient id	varchar
PATIENT	PatientName	patient full name	varchar
PATIENT	InsuranceName	latest Insurance using valid from date from patient insurance bridge table	varchar
PATIENT	InsuranceId	latest Insurance using valid from date from patient insurance bridge table	varchar
PATIENT	InsuranceAddressLine1	Latest Insurance Address using valid from date for each patient , insurance	varchar
PATIENT	InsuranceCity	Latest Insurance Address using valid from date for each patient , insurance	varchar
PATIENT	InsuranceState	Latest Insurance Address using valid from date for each patient , insurance	varchar
PATIENT	InsuranceZip	Latest Insurance Address using valid from date for each patient , insurance	varchar
PATIENT	AddressLine1	Latest home address from the address table using valid from date disregarding the insurance.	varchar
PATIENT	City	Latest home address from the address table using valid from date disregarding the insurance.	varchar
PATIENT	State	Latest home address from the address table using valid from date disregarding the insurance.	varchar
PATIENT	Zip	Latest home address from the address table using valid from date disregarding the insurance.	varchar

GEN AI Mapping Agent

Sample Sources Profile - json_data_sources1

SourceTableName	SourceColumnName	SourceColumnDataType	UniqueDataValues	TableDescription	Nullable
EE	ID	int	1,2	Employee table	N
EE	firstname	varchar	Raj,Kamal	Employee table	Y
EE	lastname	varchar	Kumar,Hassan	Employee table	Y
Dep	ID	int	1,2	Department Table	N
Dep	name	varchar	Finance,Sales	Department Table	N
ED	EMPID	int	1,2	Employee Department table	N
ED	DEPID	int	1,2	Employee Department table	N
ED	ValidFrom	Date	01-10-2020,01-01-2018	Employee Department table	Y
AD	City	varchar	Ahmedabad,Goa	address table	Y
AD	State	varchar	Gujarat,maharastra	address table	Y
AD	Zip	varchar	400050 , 400093	address table	Y
AD	ValidFrom	Date	01-10-2000,01-01-2005	address table	N
AD	DEPID	int	1,2	address table	N

Sample Target Profile - json_data_target

TargetTableName	TargetColumnName	TargetExplanation	TargetColumnDataT
Employee	EmployeeID	unique patient id	varchar
Employee	EmployeeName	patient full name	varchar
Employee	DeptName	Latest Department Name using valid from	varchar
Employee	DeptID	Latest Department ID using valid from	varchar
Employee	DeptAddressCity	Latest Address for Latest Department using valid from	varchar
Employee	DeptAddressState	Latest Address for Latest Department using valid from	varchar
Employee	DeptAddressZip	Latest Address for Latest Department using valid from	varchar

GEN AI Mapping Agent

Sample Query Given for few shot learning.

```
-- CTE for Latest Department
WITH LatestDepartment AS (
  SELECT ee.ID AS EmployeeID,
        ed.DEPID AS DepartmentID,
        d.name AS DepartmentName,
        ROW_NUMBER() OVER (PARTITION BY ee.ID ORDER BY ed.ValidFrom
DESC) AS rn
  FROM EE ee
  LEFT JOIN ED ed ON ee.ID = ed.EMPID
  LEFT JOIN Dep d ON ed.DEPID = d.ID
),

-- CTE for Latest Department Address
LatestDepartmentAddress AS (
  SELECT ld.EmployeeID,
        ld.DepartmentID,
        ld.DepartmentName,
        a.City AS DepartmentAddressCity,
        a.State AS DepartmentAddressState,
        a.Zip AS DepartmentAddressZip,
        ROW_NUMBER() OVER (PARTITION BY ld.EmployeeID, ld.DepartmentID
ORDER BY a.ValidFrom DESC) AS rn
  FROM LatestDepartment ld
  LEFT JOIN AD a ON ld.DepartmentID = a.DEPID
  WHERE ld.rn = 1
),
```

```
-- Final Query
SELECT ld.EmployeeID,
      CONCAT(CAST(ee.firstname AS VARCHAR(100)), ' ',
CAST(ee.lastname AS VARCHAR(100))) AS EmployeeName,
      lda.DepartmentName,
      CAST(lda.DepartmentID AS VARCHAR(20)) AS DeptID,
      lda.DepartmentAddressCity,
      lda.DepartmentAddressState,
      lda.DepartmentAddressZip
FROM LatestDepartment ld
INNER JOIN EE ee ON ld.EmployeeID = ee.ID
LEFT JOIN LatestDepartmentAddress lda ON ld.EmployeeID =
lda.EmployeeID AND ld.DepartmentID = lda.DepartmentID
WHERE ld.rn = 1;
```

GEN AI Mapping Agent

Final Query Generated by Agent

```
WITH LatestInsurance AS (
  SELECT ptnt.ID AS PatientID,
        ptins.insid AS InsuranceID,
        ins.name AS InsuranceName,
        ROW_NUMBER() OVER (PARTITION BY ptnt.ID ORDER BY ptins.ValidFrom DESC) AS
rn
  FROM Ptnt101 ptnt
  LEFT JOIN PTINS101 ptins ON ptnt.ID = ptins.ptntid
  LEFT JOIN INS101 ins ON ptins.insid = ins.Id
),

LatestInsuranceAddress AS (
  SELECT li.PatientID,
        li.InsuranceID,
        li.InsuranceName,
        a.ID AS AddressId,
        a."Address Line 1" AS InsuranceAddressLine1,
        a.City AS InsuranceCity,
        a.State AS InsuranceState,
        a.Zip AS InsuranceZip,
        ROW_NUMBER() OVER (PARTITION BY li.PatientID, li.InsuranceID ORDER BY
a.ValidFrom DESC) AS rn
  FROM LatestInsurance li
  LEFT JOIN ADD101 a ON li.InsuranceID = a.INSID
),

LatestAddress AS (
  SELECT ptnt.ID AS PatientID,
        a.ID AS AddressId,
        a."Address Line 1" AS AddressLine1,
        a.City AS City,
        a.State AS State,
        a.Zip AS Zip,
        ROW_NUMBER() OVER (PARTITION BY ptnt.ID ORDER BY a.ValidFrom DESC) AS rn
  FROM Ptnt101 ptnt
  LEFT JOIN ADD101 a ON ptnt.ID = a.ptntid
  WHERE a.AddType = 'H'
),
-- Final Query
SELECT ptnt.ID AS PatientId,
      CONCAT(CAST(ptnt.firstname AS VARCHAR(100)), ' ', CAST(ptnt.lastname AS
VARCHAR(100))) AS PatientName,
      lia.InsuranceName,
      CAST(lia.InsuranceID AS VARCHAR(20)) AS InsuranceId,
      lia.InsuranceAddressLine1,
      lia.InsuranceCity,
      lia.InsuranceState,
      lia.InsuranceZip,
      la.AddressLine1,
      la.City,
      la.State,
      la.Zip
FROM Ptnt101 ptnt
LEFT JOIN LatestInsurance lia ON ptnt.ID = lia.PatientID
LEFT JOIN LatestInsuranceAddress lia ON lia.PatientID = lia.PatientID AND
lia.InsuranceID = lia.InsuranceID
LEFT JOIN LatestAddress la ON ptnt.ID = la.PatientID
WHERE lia.rn = 1 AND la.rn = 1;
```

Conclusion

- Claude Performed better than Gemini on this task and even gave the result with zero shot prompt but was not consistent.
- We do not need to provide the actual data for the code to be generated.
- Zero shot prompts did not work very well in this task.
- Few Shot prompting gave good results with Claude and Gemini both.