HAP720: PROJECT

RASHMI KONDAKINDI

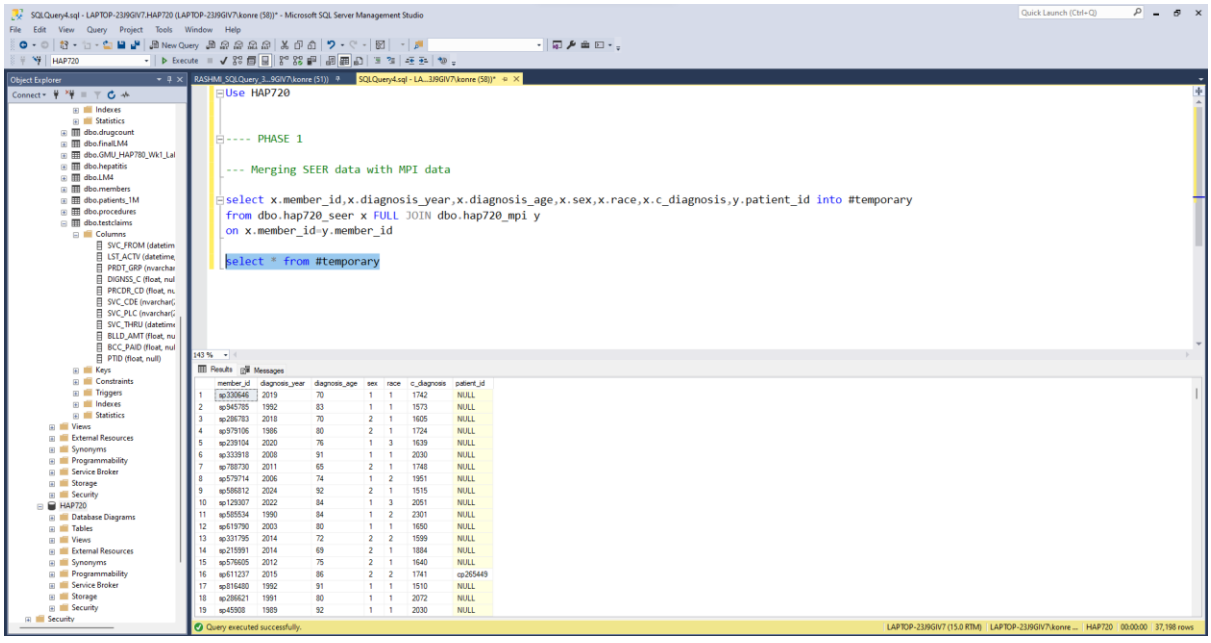
PHASE 1

In this phase, please write SQL, Python, or other programming languages to integrate records from two source files (SEER and Claims) based on the MPI.

After you finish the integration, please answer the following three questions according to your results.

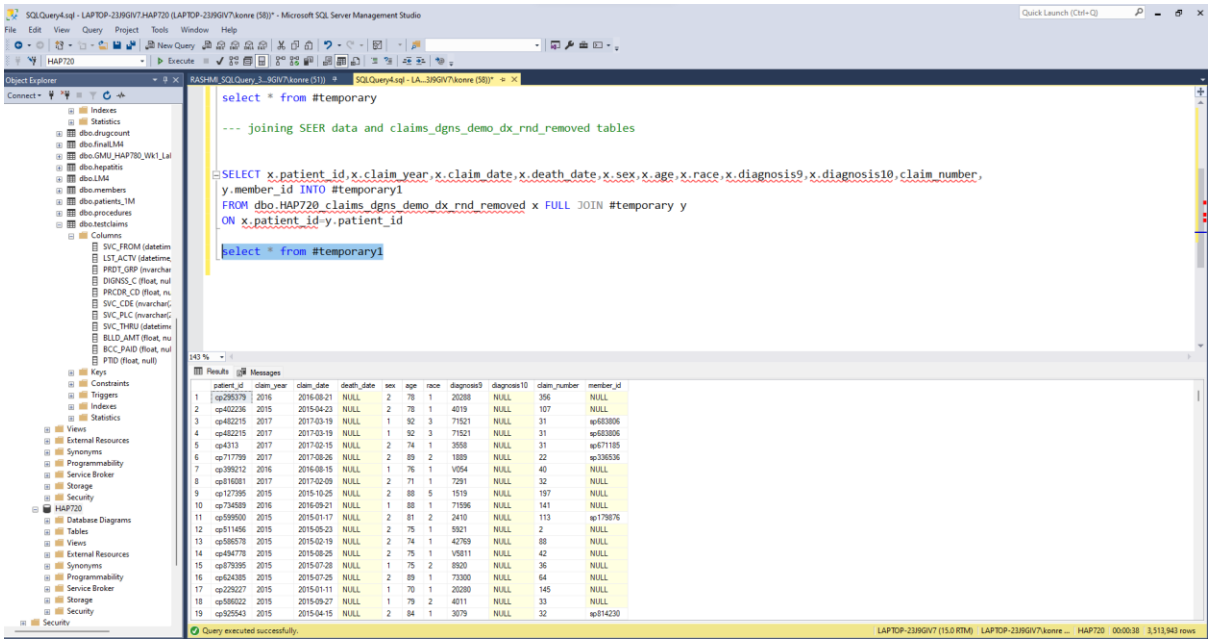
1. **How many patients in the SEER do not have matching records in the Claims?**
2. **How many patients in the Claims do not have matching records in the SEER?**
3. **What does this mean? (i.e., why do patients not exist in SEER?)**

```
---- PHASE 1

--- Merging SEER data with MPI data

select
x.member_id,x.diagnosis_year,x.diagnosis_age,x.sex,x.race,x.c_diagnosis,y.patient_id
into #temporary
from dbo.hap720_seer x FULL JOIN dbo.hap720_mpi y
on x.member_id=y.member_id

select * from #temporary
```

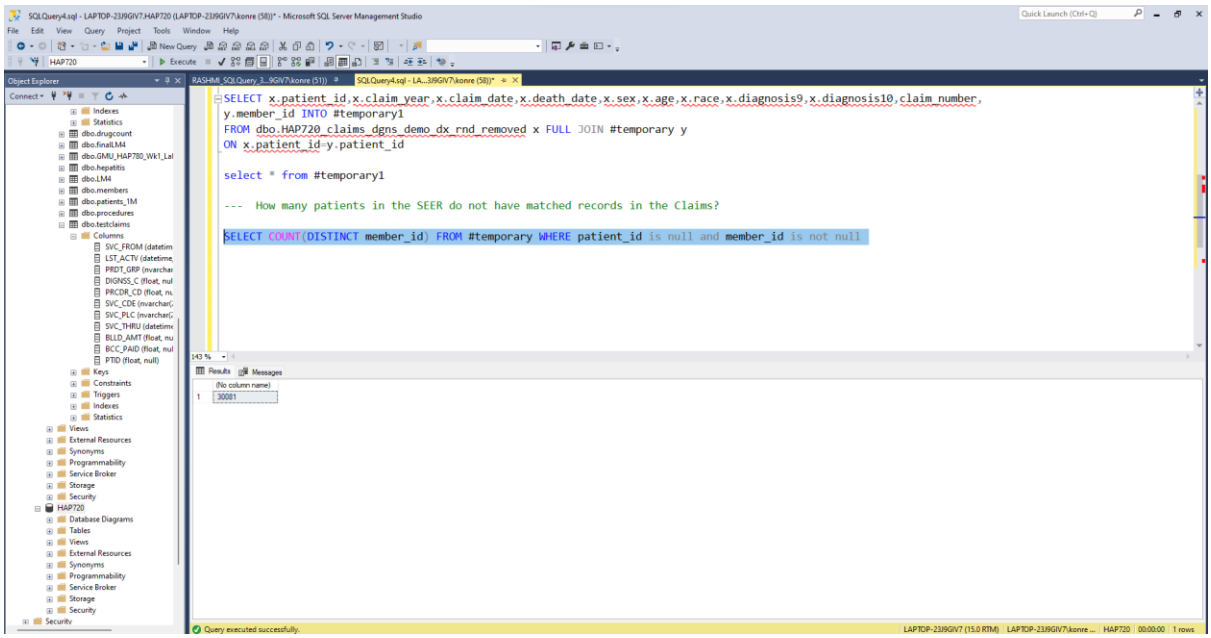



```
--- joining SEER data and claims_dgns_demo_dx_rnd_removed tables
```

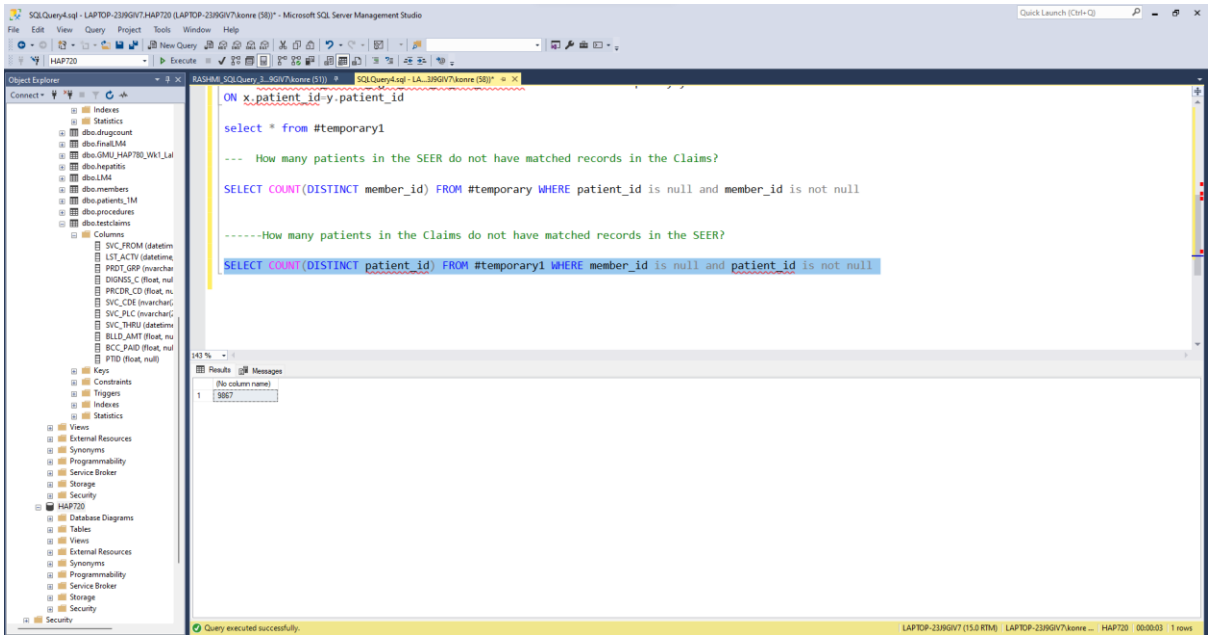
```
SELECT
x.patient_id,x.claim_year,x.claim_date,x.death_date,x.sex,x.age,x.race,x.diagnosis9,x.
diagnosis10,claim_number,
y.member_id INTO #temporary1
FROM dbo.HAP720_claims_dgns_demo_dx_rnd_removed x FULL JOIN #temporary y
ON x.patient_id=y.patient_id

select * from #temporary1
```




```
-------How many patients in the SEER do not have matched records in the Claims?

SELECT COUNT(DISTINCT member_id) FROM #temporary WHERE patient_id is null and
member_id is not null
```



```
------How many patients in the Claims do not have matched records in the SEER?

SELECT COUNT(DISTINCT patient_id) FROM #temporary1 WHERE member_id is null and
patient_id is not null
```



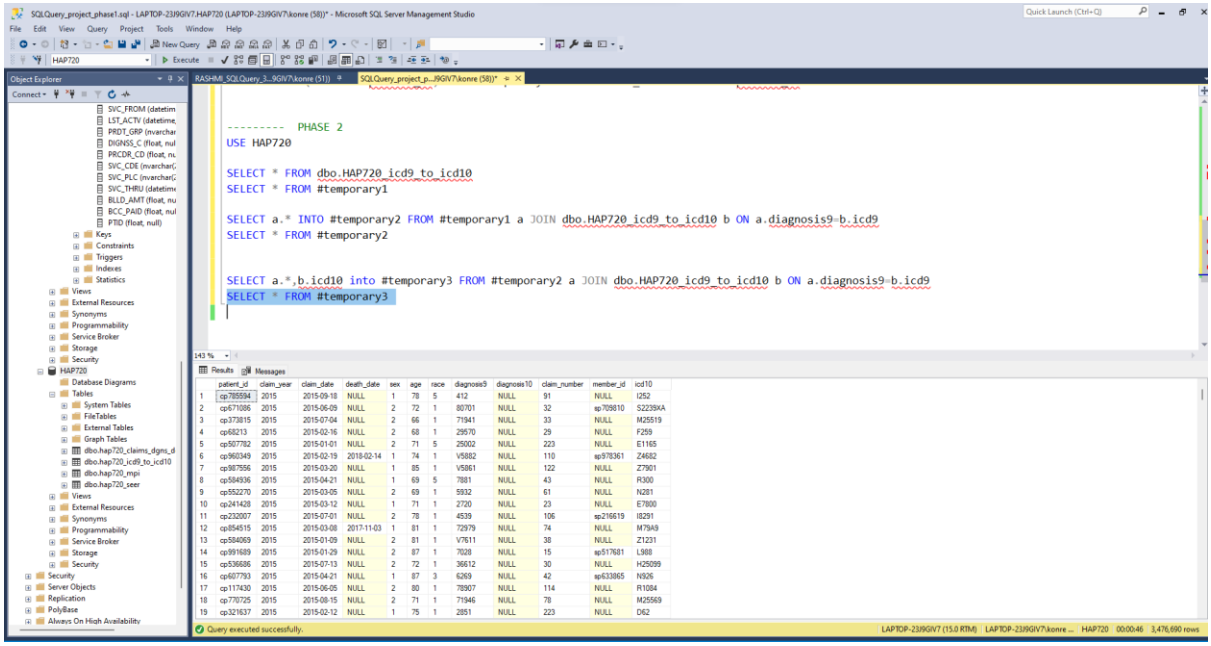
----------What does this mean? (i.e., why do patients not exist in SEER?)

The patient may not be listed in SEER for a number of reasons; the cancer registry was established to compile information on all cancer patients in the SEER database. It's possible that the patient didn't have cancer or didn't match the criteria for inclusion if they aren't recorded in the SEER database. Other factors included errors or omissions during data collection or coding, which resulted in inaccurate or missing data.

PHASE 2

In this phase, please write SQL, Python, or other programming languages to map ICD9CM codes to ICD10CM in your merged file from Phase One. You can find the mappings between ICD9CM and ICD10 in the attached file.

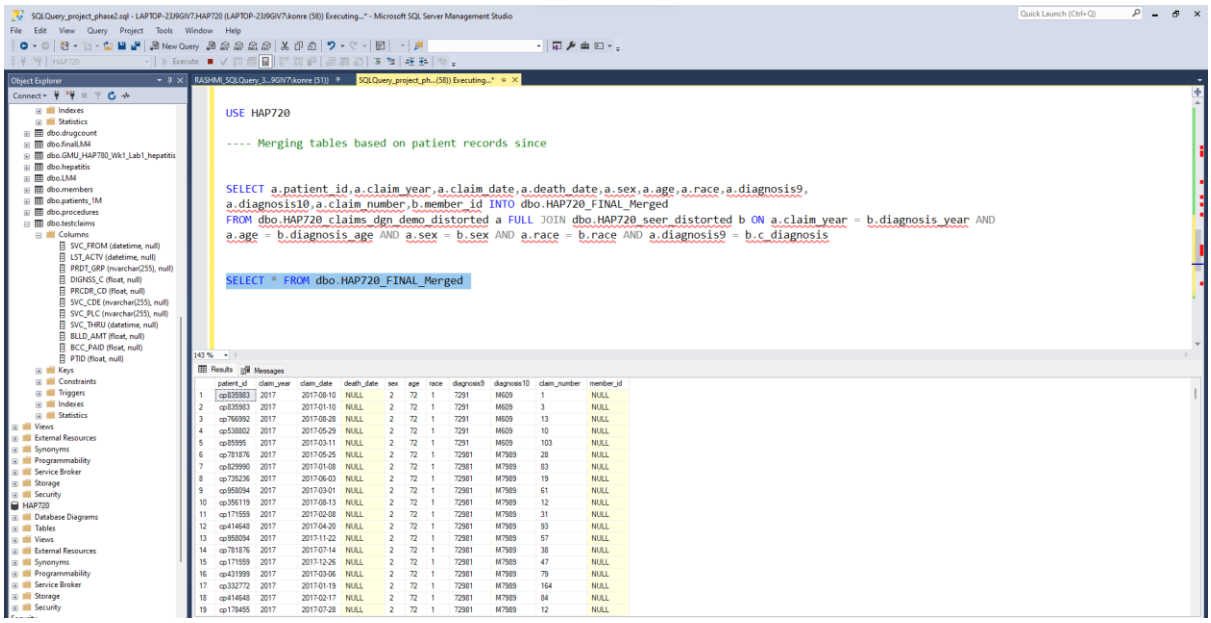
```
---------  PHASE 2
USE HAP720

SELECT * FROM dbo.HAP720_icd9_to_icd10
SELECT * FROM #temporary1

SELECT a.* INTO #temporary2 FROM #temporary1 a JOIN dbo.HAP720_icd9_to_icd10 b ON
a.diagnosis9=b.icd9
SELECT * FROM #temporary2


SELECT a.*,b.icd10 into #temporary3 FROM #temporary2 a JOIN dbo.HAP720_icd9_to_icd10 b
ON a.diagnosis9=b.icd9
SELECT * FROM #temporary3
```



PHASE 3

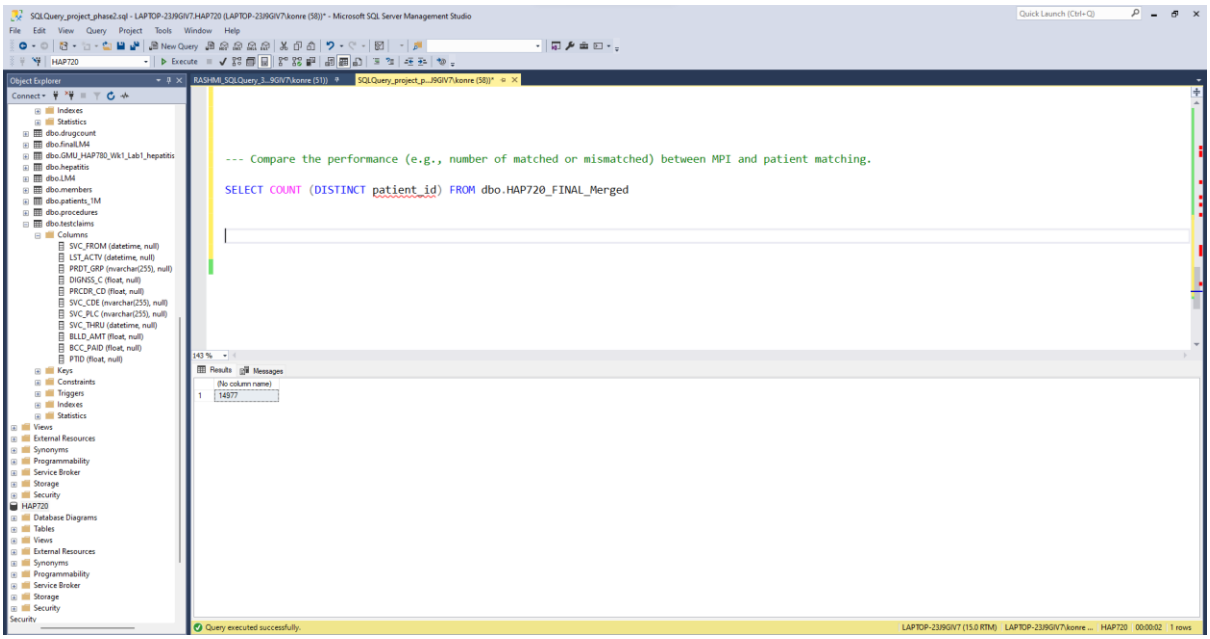
In this phase, we assume that we do not have MPI then we need to apply patient-matching algorithms. Please write SQL, Python, or other programming languages to integrate records from two source files (SEER and Claims) based on the patient's demographic (e.g., age, gender, etc.) and clinical information (e.g., ICD).

```
---- PHASE 3

USE HAP720

---- Merging tables based on patient records since


SELECT
a.patient_id,a.claim_year,a.claim_date,a.death_date,a.sex,a.age,a.race,a.diagnosis9,
a.diagnosis10,a.claim_number,b.member_id INTO dbo.HAP720_FINAL_Merged
FROM dbo.HAP720_claims_dgn_demo_distorted a FULL JOIN dbo.HAP720_seer_distorted b ON
a.claim_year = b.diagnosis_year AND
a.age = b.diagnosis_age AND a.sex = b.sex AND a.race = b.race AND a.diagnosis9 =
b.c_diagnosis

SELECT * FROM dbo.HAP720_FINAL_Merged
```



1. **Compare the performance (e.g., number of matched or mismatched) between MPI and patient matching.**

```
--- Compare the performance (e.g., number of matched or mismatched) between MPI and
patient matching.

SELECT COUNT (DISTINCT patient_id) FROM dbo.HAP720_FINAL_Merged
```



2. **Explain why the matching is not perfect?**

   Since neither table had a common unique ID, the matching was insufficient, thus we linked the patient characteristics tables. By first connecting the SEER database with MPI, and then the output table with the claims table, we were able to connect flawlessly in the first instance since we had an MPI table with columns for patient_id and member_id.