# Course 2 Module 5
# Programming Assignment

## Assignment is to ETL MIMIC data into the OMOP CONDITION_OCCURRENCE table

**Detailed instructions with Slide Notes**

# Step 1: Understand source/target data models

**Paste one or more MIMIC table(s) from the previous two slides that contain data for ETL into OMOP CONDITION_OCCURRENCE here!**

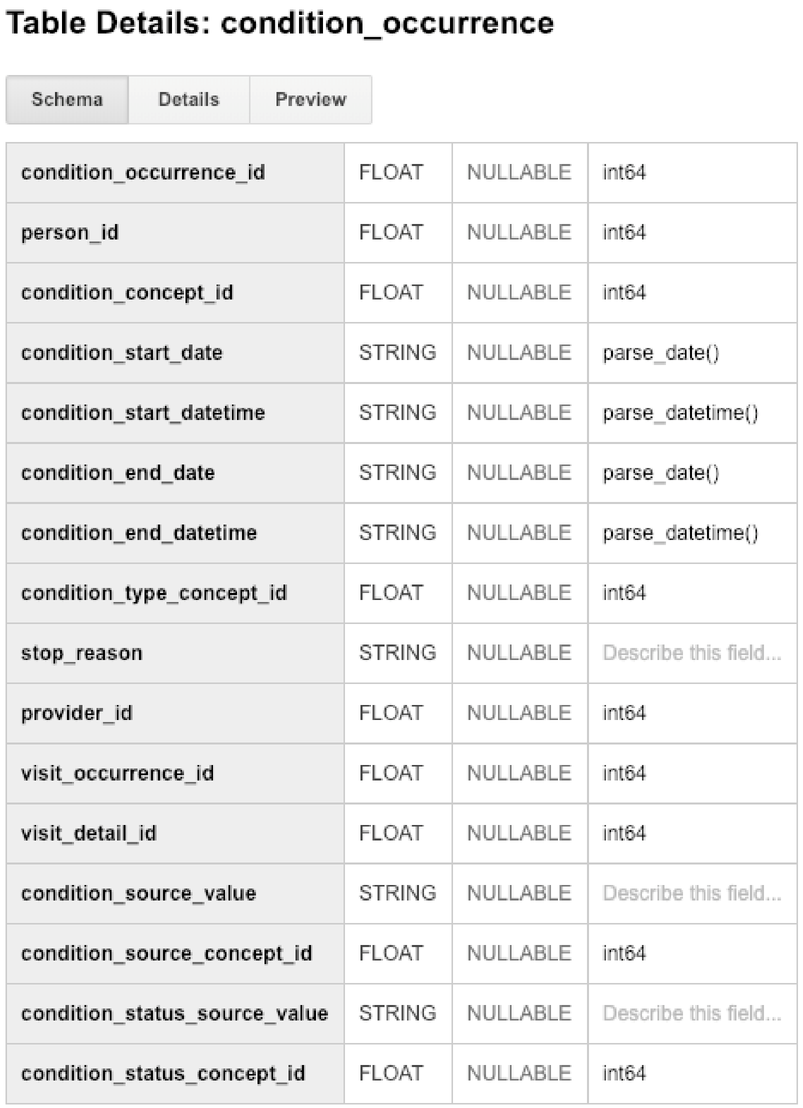


Table Details: condition_occurrence

| | | | |
|---|---|---|---|
| condition_occurrence_id | FLOAT | NULLABLE | int64 |
| person_id | FLOAT | NULLABLE | int64 |
| condition_concept_id | FLOAT | NULLABLE | int64 |
| condition_start_date | STRING | NULLABLE | parse_date() |
| condition_start_datetime | STRING | NULLABLE | parse_datetime() |
| condition_end_date | STRING | NULLABLE | parse_date() |
| condition_end_datetime | STRING | NULLABLE | parse_datetime() |
| condition_type_concept_id | FLOAT | NULLABLE | int64 |
| stop_reason | STRING | NULLABLE | Describe this field... |
| provider_id | FLOAT | NULLABLE | int64 |
| visit_occurrence_id | FLOAT | NULLABLE | int64 |
| visit_detail_id | FLOAT | NULLABLE | int64 |
| condition_source_value | STRING | NULLABLE | Describe this field... |
| condition_source_concept_id | FLOAT | NULLABLE | int64 |
| condition_status_source_value | STRING | NULLABLE | Describe this field... |
| condition_status_concept_id | FLOAT | NULLABLE | int64 |

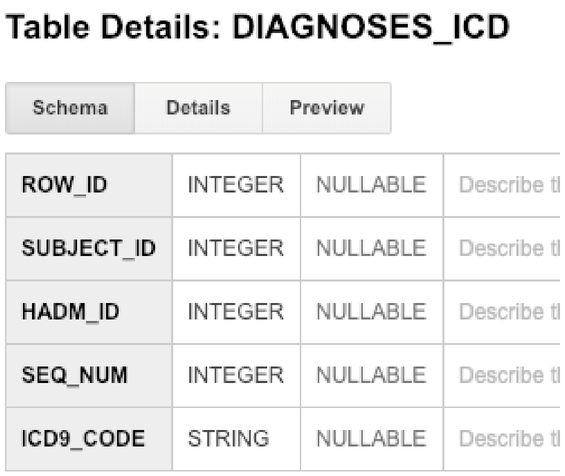


Table Details: DIAGNOSES_ICD

| | | | |
|---|---|---|---|
| ROW_ID | INTEGER | NULLABLE | Describe tl |
| SUBJECT_ID | INTEGER | NULLABLE | Describe tl |
| HADM_ID | INTEGER | NULLABLE | Describe tl |
| SEQ_NUM | INTEGER | NULLABLE | Describe tl |
| ICD9_CODE | STRING | NULLABLE | Describe tl |

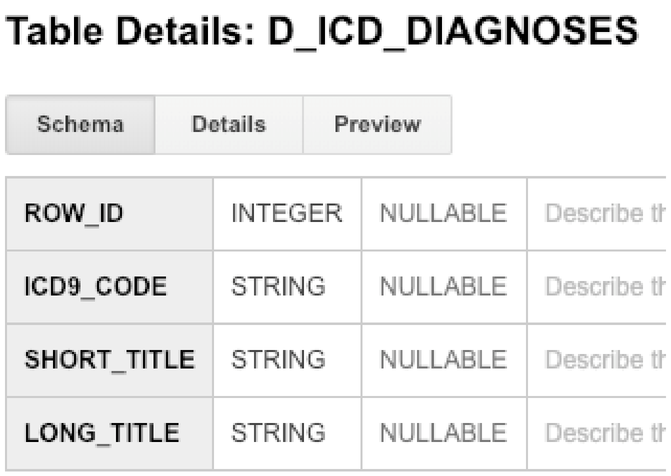
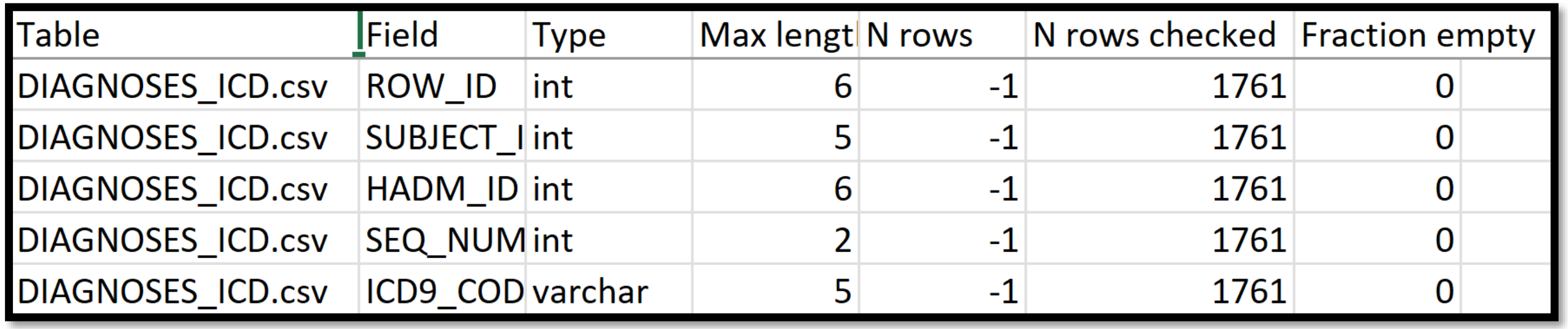


Table Details: D_ICD_DIAGNOSES

| | | | |
|---|---|---|---|
| ROW_ID | INTEGER | NULLABLE | Describe th |
| ICD9_CODE | STRING | NULLABLE | Describe th |
| SHORT_TITLE | STRING | NULLABLE | Describe th |
| LONG_TITLE | STRING | NULLABLE | Describe th |

# Step 2: Profile source table or tables

**Using the White Rabbit profiling data from the 100 patient MIMIC database provided in the Assessment to comment on the distribution of the SUBJECT_ID field from one of the MIMIC tables selected in Step 1**



| Table | Field | Type | Max lengt | N rows | N rows checked | Fraction empty |
|---|---|---|---|---|---|---|
| DIAGNOSES_ICD.csv | ROW_ID | int | 6 | -1 | 1761 | 0 |
| DIAGNOSES_ICD.csv | SUBJECT_I | int | 5 | -1 | 1761 | 0 |
| DIAGNOSES_ICD.csv | HADM_ID | int | 6 | -1 | 1761 | 0 |
| DIAGNOSES_ICD.csv | SEQ_NUM | int | 2 | -1 | 1761 | 0 |
| DIAGNOSES_ICD.csv | ICD9_COD | varchar | 5 | -1 | 1761 | 0 |

## DIAGNOSES_ICD

The White Rabbit profiling data shows that there are 1761 subject_id rows in the table which is far more than the 100 subjects that are expected to be in the sample data. Using the query-
select distinct subject_id from mimic3_demo.DIAGNOSES_ICD though gives a result with only 100 rows as expected. This indicates that each unique patient has had multiple diagnoses assigned to them.

# Step 3: Create ETL mappings

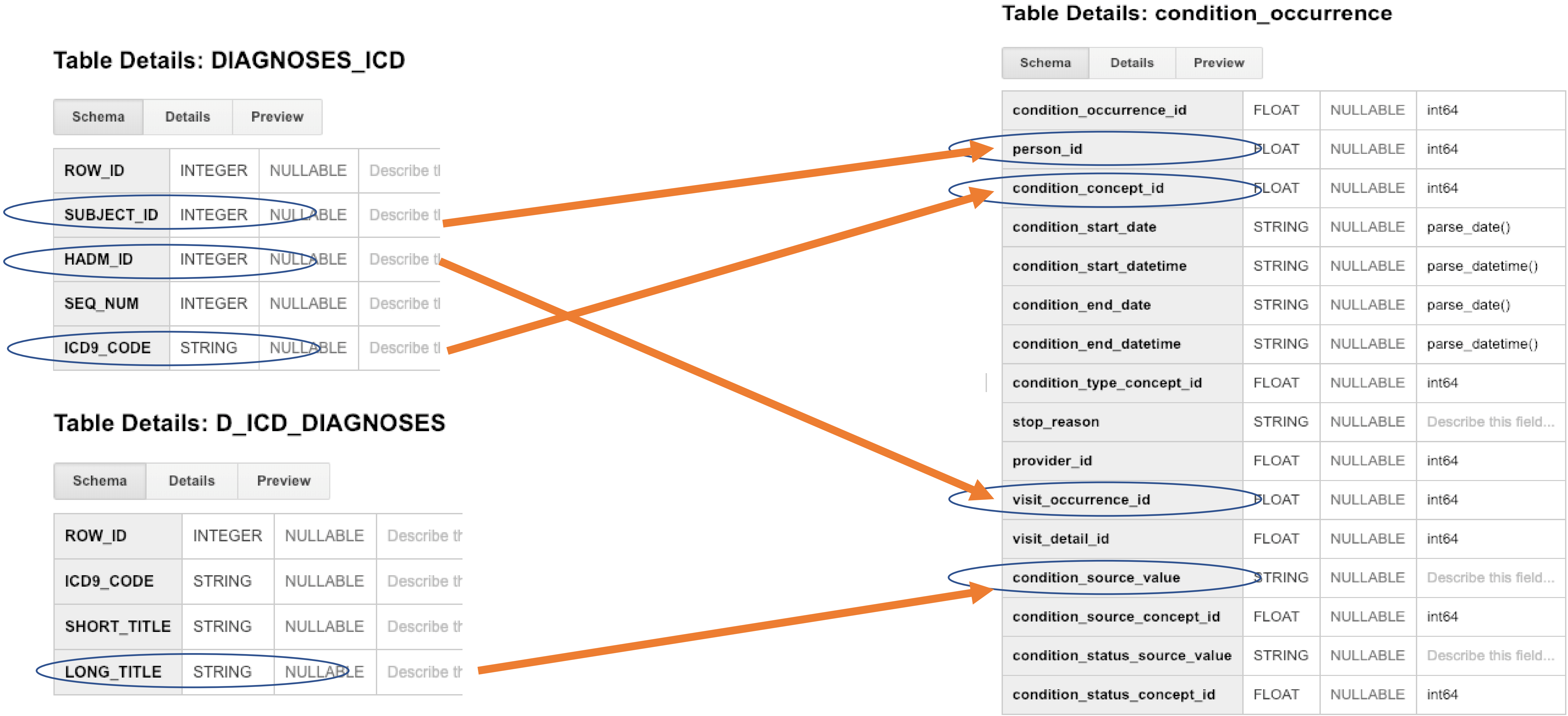


**Table Details: DIAGNOSES_ICD**

| | | | |
|---|---|---|---|
| ROW_ID | INTEGER | NULLABLE | Describe t |
| SUBJECT_ID | INTEGER | NULLABLE | Describe t |
| HADM_ID | INTEGER | NULLABLE | Describe t |
| SEQ_NUM | INTEGER | NULLABLE | Describe t |
| ICD9_CODE | STRING | NULLABLE | Describe t |

**Table Details: D_ICD_DIAGNOSES**

| | | | |
|---|---|---|---|
| ROW_ID | INTEGER | NULLABLE | Describe th |
| ICD9_CODE | STRING | NULLABLE | Describe th |
| SHORT_TITLE | STRING | NULLABLE | Describe th |
| LONG_TITLE | STRING | NULLABLE | Describe th |

**Table Details: condition_occurrence**

| | | | |
|---|---|---|---|
| condition_occurrence_id | FLOAT | NULLABLE | int64 |
| person_id | FLOAT | NULLABLE | int64 |
| condition_concept_id | FLOAT | NULLABLE | int64 |
| condition_start_date | STRING | NULLABLE | parse_date() |
| condition_start_datetime | STRING | NULLABLE | parse_datetime() |
| condition_end_date | STRING | NULLABLE | parse_date() |
| condition_end_datetime | STRING | NULLABLE | parse_datetime() |
| condition_type_concept_id | FLOAT | NULLABLE | int64 |
| stop_reason | STRING | NULLABLE | Describe this field... |
| provider_id | FLOAT | NULLABLE | int64 |
| visit_occurrence_id | FLOAT | NULLABLE | int64 |
| visit_detail_id | FLOAT | NULLABLE | int64 |
| condition_source_value | STRING | NULLABLE | Describe this field... |
| condition_source_concept_id | FLOAT | NULLABLE | int64 |
| condition_status_source_value | STRING | NULLABLE | Describe this field... |
| condition_status_concept_id | FLOAT | NULLABLE | int64 |

# Explanation of mappings

- SUBJECT_ID → person_id

The unique identifier of a patient, subject_id in the MIMIC DIAGNOSES_ICD table is used to populate the unique identifier, person_id in the OMOP CONDITION_OCCURENCE table

- HADM_ID → visit_occurrence_id

HADM_ID, which is a unique identifier of each hospital stay is used to populate the visit_occurrence_id, which identifies the visit during which the condition occurred.

- ICD9_CODE → condition_concept_id

The ICD9_CODE, which is a code corresponding to the diagnosis assigned to the patient, is used to populate condition_concept_id.

- LONG_TITLE → condition_source_value

The condition_source_value maps to the condition_concept_id. So, the original value of ICD9_CODE from the source, LONG_TITLE, is used to populate the condition_source_value table.

# Step 4: Write transformation code

```
WITH occur1 as (select distinct d.subject_id as person_id, d.hadm_id as visit_occurence_id, d.icd9_code as condition_concept_id  from mimic3_demo.DIAGNOSES_ICD d),

        occur as (select distinct o1.person_id, o1. visit_occurence_id, o1.condition_concept_id, di.long_title as condition_source_value from occur1 o1 join mimic3_demo.D_ICD_DIAGNOSES di on o1.condition_concept_id= di.ICD9_CODE)

select * from occur
```

**Paste the SQL statements that transform data from one or more MIMIC tables into the three OMOP CONDITION_OCCURRENCE fields (patient-id, visit_occurrence_id, condition_source_value) into the Coursera Submission Site**

# Step 5: Execute transformation code

**Execute the ETL code from Step 4 but do not submit the output table.**

**Use the output table for Step 6.**

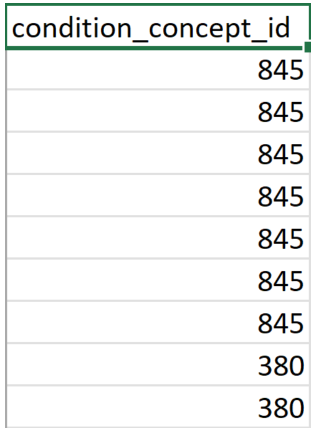
**There is no submission for this Step.**

# Step 6: Perform data quality assessment

I used the following SQL code to check if there are any ICD9 codes that did not get mapped (indicated by a value of zero) to the condition_occurrence table during the ETL process-

```
WITH occur1 as (select distinct d.subject_id as person_id, d.hadm_id as visit_occurence_id, d.icd9_code as condition_concept_id  from mimic3_demo.DIAGNOSES_ICD d),

occur as (select distinct o1.person_id, o1. visit_occurence_id, o1.condition_concept_id, di.long_title as condition_source_value from occur1 o1 join mimic3_demo.D_ICD_DIAGNOSES di on o1.condition_concept_id= di.ICD9_CODE)

select condition_concept_id

from occur

order by condition_concept_id
```

The result showed that there were no unmapped codes (no zeroes) and hence, the ETL process proceeded successfully-



| condition_concept_id |
|---:|
| 845 |
| 845 |
| 845 |
| 845 |
| 845 |
| 845 |
| 845 |
| 380 |
| 380 |

# Step 7: Package documentation

- Congratulations! The materials in the previous slides constitute a complete ETL package.

**There is no submission for this Step.**