01_bronze_layer_analysis

January 2, 2026

0.1 About the Author

This report was authored by Ayoub Majjid, a fifth-year computer engineering student at EMSI with a background in Experimental Sciences. His academic journey has provided a strong foundation in mathematics, physics, and chemistry, and has shaped a growing expertise in technology, system design, and data engineering.

Ayoub currently serves as Tech Lead and Entrepreneur at Intellcap, where he leads three innovation projects focused on building impactful and scalable startup solutions. His work emphasizes transforming ideas into robust technical systems, with a particular interest in data platforms, system architecture, and end-to-end engineering workflows.

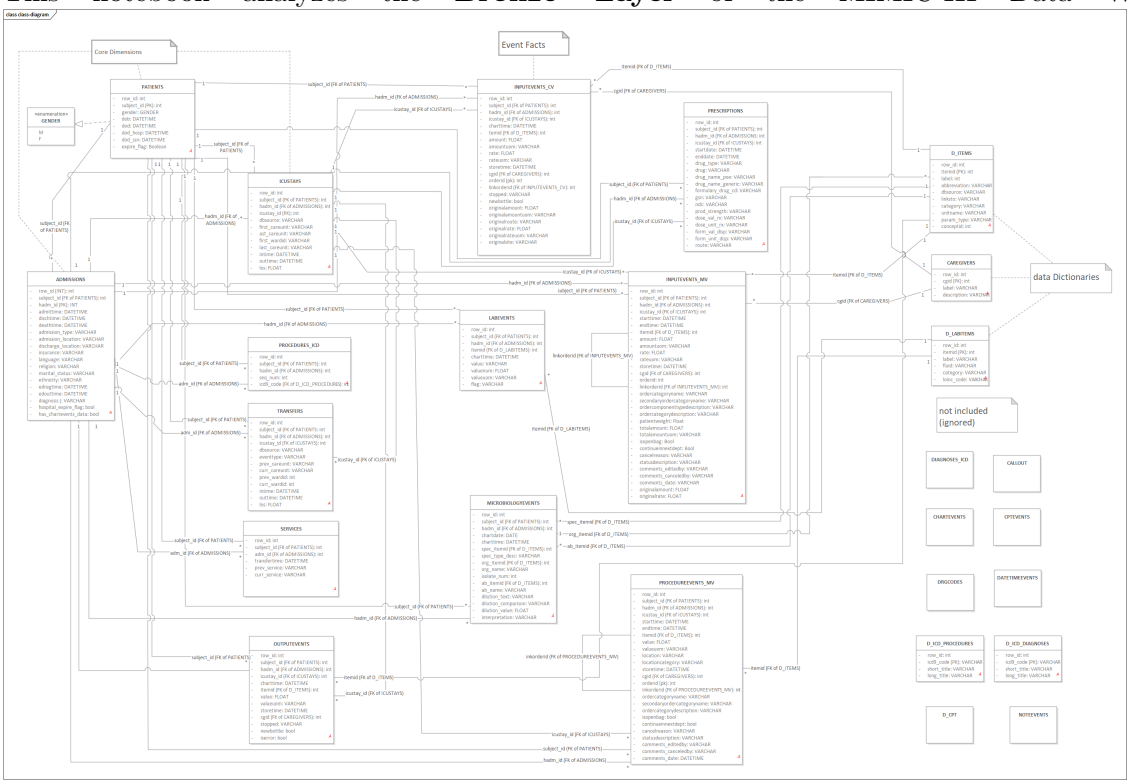
Email: ayoub@majjid.com
Website: https://majjid.com

0.2 Copyright & Disclaimer

---

1 Bronze Layer Analysis

This notebook analyzes the **Bronze Layer** of the MIMIC-III Data Warehouse.



##

What is the Bronze Layer?

The Bronze layer contains **raw data** loaded directly from CSV files with minimal transformation.

- Data is preserved in its original format
- No data cleaning or validation
- Source of truth for raw data

[ ]:
```
# Setup
import sys
sys.path.insert(0, '../../..')

import pandas as pd
from sqlalchemy import text
from app.shared import get_db

pd.set_option('display.max_columns', None)
pd.set_option('display.width', None)
```

[ ]:
```
# Helper function to query and display data
def query_df(sql, limit=10):
    with get_db() as session:
```

```
        result = session.execute(text(sql))
        df = pd.DataFrame(result.fetchall(), columns=result.keys())
    return df.head(limit) if limit else df

def table_count(schema, table):
    with get_db() as session:
        return session.execute(text(f"SELECT COUNT(*) FROM {schema}.{table}")).
 ↪scalar()
```

1.1 1. PATIENTS Table

Description: Core patient demographics table containing one row per patient.

```
| Column | Type | Description |
|--------|------|-------------|
| row_id | INTEGER | Unique row identifier |
| subject_id | INTEGER | **Primary key** - Unique patient identifier |
| gender | VARCHAR | Patient gender (M/F) |
| dob | TIMESTAMP | Date of birth |
| dod | TIMESTAMP | Date of death (if applicable) |
| dod_hosp | TIMESTAMP | Date of death in hospital |
| dod_ssn | TIMESTAMP | Date of death from SSA records |
| expire_flag | INTEGER | 1 if patient deceased, 0 otherwise |
```

[ ]:
```
print(f"Total Patients: {table_count('bronze', 'patients'):,}")
query_df("SELECT * FROM bronze.patients ORDER BY subject_id LIMIT 5")
```

```
2026-01-01 14:33:03,239 INFO sqlalchemy.engine.Engine select
pg_catalog.version()
2026-01-01 14:33:03,241 INFO sqlalchemy.engine.Engine [raw sql] {}
2026-01-01 14:33:03,251 INFO sqlalchemy.engine.Engine select current_schema()
2026-01-01 14:33:03,252 INFO sqlalchemy.engine.Engine [raw sql] {}

2026-01-01 14:33:03,272 INFO sqlalchemy.engine.Engine show
standard_conforming_strings
2026-01-01 14:33:03,276 INFO sqlalchemy.engine.Engine [raw sql] {}
2026-01-01 14:33:03,281 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:33:03,285 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.patients
2026-01-01 14:33:03,290 INFO sqlalchemy.engine.Engine [generated in 0.00604s] {}
2026-01-01 14:33:03,304 INFO sqlalchemy.engine.Engine COMMIT
Total Patients: 100
2026-01-01 14:33:03,312 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:33:03,314 INFO sqlalchemy.engine.Engine SELECT * FROM
bronze.patients ORDER BY subject_id LIMIT 5
2026-01-01 14:33:03,316 INFO sqlalchemy.engine.Engine [generated in 0.00171s] {}
2026-01-01 14:33:03,339 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:

	row_id	subject_id	gender	dob	dod	dod_hosp	dod_ssn	expire_flag	created_at	updated_at
0	9467	10006	F	2094-03-05	2165-08-12	2165-08-12	2165-08-12	True	2025-12-25 12:14:53.276330+00:00	2025-12-25 12:14:53.276330+00:00
1	9472	10011	F	2090-06-05	2126-08-28	2126-08-28	NaT	True	2025-12-25 12:14:53.276330+00:00	2025-12-25 12:14:53.276330+00:00
2	9474	10013	F	2038-09-03	2125-10-07	2125-10-07	2125-10-07	True	2025-12-25 12:14:53.276330+00:00	2025-12-25 12:14:53.276330+00:00
3	9478	10017	F	2075-09-21	2152-09-12	NaT	2152-09-12	True	2025-12-25 12:14:53.276330+00:00	2025-12-25 12:14:53.276330+00:00
4	9479	10019	M	2114-06-20	2163-05-15	2163-05-15	2163-05-15	True	2025-12-25 12:14:53.276330+00:00	2025-12-25 12:14:53.276330+00:00

[ ]:
```
# Gender distribution
query_df("""
    SELECT gender, COUNT(*) as count,
           ROUND(100.0 * COUNT(*) / SUM(COUNT(*)) OVER(), 1) as percentage
    FROM bronze.patients
    GROUP BY gender
""", limit=None)
```

```
2026-01-01 14:33:03,457 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:33:03,460 INFO sqlalchemy.engine.Engine
    SELECT gender, COUNT(*) as count,
           ROUND(100.0 * COUNT(*) / SUM(COUNT(*)) OVER(), 1) as percentage
    FROM bronze.patients
    GROUP BY gender

2026-01-01 14:33:03,462 INFO sqlalchemy.engine.Engine [generated in 0.00198s] {}
2026-01-01 14:33:03,479 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:

	gender	count	percentage
0	M	45	45.0
1	F	55	55.0

1.2 2. ADMISSIONS Table

Description: Hospital admission records. One row per hospital visit.

```
| Column | Type | Description |
|--------|------|-------------|
| hadm_id | INTEGER | **Primary key** - Unique admission ID |
| subject_id | INTEGER | Patient ID (FK to patients) |
| admittime | TIMESTAMP | Admission date/time |
| dischtime | TIMESTAMP | Discharge date/time |
| deathtime | TIMESTAMP | Time of death (if in hospital) |
| admission_type | VARCHAR | EMERGENCY, ELECTIVE, URGENT, NEWBORN |
| admission_location | VARCHAR | Where patient was admitted from |
| discharge_location | VARCHAR | Where patient was discharged to |
| insurance | VARCHAR | Patient's insurance type |
| diagnosis | TEXT | Preliminary diagnosis |
```

[ ]:
```
print(f"Total Admissions: {table_count('bronze', 'admissions'):,}")
query_df("SELECT hadm_id, subject_id, admittime, dischtime, admission_type,␣
 ↪insurance FROM bronze.admissions LIMIT 5")
```

```
2026-01-01 14:33:03,544 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:33:03,546 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.admissions
2026-01-01 14:33:03,547 INFO sqlalchemy.engine.Engine [generated in 0.00134s] {}
2026-01-01 14:33:03,570 INFO sqlalchemy.engine.Engine COMMIT
Total Admissions: 129
2026-01-01 14:33:03,580 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:33:03,582 INFO sqlalchemy.engine.Engine SELECT hadm_id,
subject_id, admittime, dischtime, admission_type, insurance FROM
bronze.admissions LIMIT 5
2026-01-01 14:33:03,583 INFO sqlalchemy.engine.Engine [generated in 0.00108s] {}
2026-01-01 14:33:03,589 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:
```
   hadm_id  subject_id           admittime           dischtime admission_type  \
0   142345       10006 2164-10-23 21:09:00 2164-11-01 17:15:00      EMERGENCY
1   105331       10011 2126-08-14 22:32:00 2126-08-28 18:59:00      EMERGENCY
2   165520       10013 2125-10-04 23:36:00 2125-10-07 15:13:00      EMERGENCY
3   199207       10017 2149-05-26 17:19:00 2149-06-03 18:42:00      EMERGENCY
4   177759       10019 2163-05-14 20:43:00 2163-05-15 12:00:00      EMERGENCY

  insurance
0  Medicare
1   Private
2  Medicare
3  Medicare
4  Medicare
```

[ ]:
```
# Admission type distribution
query_df("""
    SELECT admission_type, COUNT(*) as count
    FROM bronze.admissions
```

```
    GROUP BY admission_type
    ORDER BY count DESC
""", limit=None)
```

```
2026-01-01 14:33:03,636 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:33:03,640 INFO sqlalchemy.engine.Engine
    SELECT admission_type, COUNT(*) as count
    FROM bronze.admissions
    GROUP BY admission_type
    ORDER BY count DESC

2026-01-01 14:33:03,642 INFO sqlalchemy.engine.Engine [generated in 0.00327s] {}
2026-01-01 14:33:03,649 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:

	admission_type	count
0	EMERGENCY	119
1	ELECTIVE	8
2	URGENT	2

1.3 3. ICUSTAYS Table

Description: ICU stay records. One row per ICU visit.

Column	Type	Description
icustay_id	INTEGER	**Primary key** - Unique ICU stay ID
subject_id	INTEGER	Patient ID
hadm_id	INTEGER	Hospital admission ID
dbsource	VARCHAR	Data source (carevue/metavision)
first_careunit	VARCHAR	First ICU unit
last_careunit	VARCHAR	Last ICU unit
first_wardid	INTEGER	First ward ID
last_wardid	INTEGER	Last ward ID
intime	TIMESTAMP	ICU admission time
outtime	TIMESTAMP	ICU discharge time
los	FLOAT	Length of stay in days

[ ]:
```
print(f"Total ICU Stays: {table_count('bronze', 'icustays'):,}")
query_df("SELECT icustay_id, subject_id, hadm_id, first_careunit, intime,␣
 ↪outtime, los FROM bronze.icustays LIMIT 5")
```

```
2026-01-01 14:33:03,680 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:33:03,682 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.icustays
2026-01-01 14:33:03,684 INFO sqlalchemy.engine.Engine [generated in 0.00135s] {}
2026-01-01 14:33:03,705 INFO sqlalchemy.engine.Engine COMMIT
Total ICU Stays: 136
2026-01-01 14:33:03,717 INFO sqlalchemy.engine.Engine BEGIN (implicit)
```

```
2026-01-01 14:33:03,719 INFO sqlalchemy.engine.Engine SELECT icustay_id,
subject_id, hadm_id, first_careunit, intime, outtime, los FROM bronze.icustays
LIMIT 5
2026-01-01 14:33:03,721 INFO sqlalchemy.engine.Engine [generated in 0.00197s] {}
2026-01-01 14:33:03,735 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:
```
   icustay_id  subject_id  hadm_id first_careunit              intime  \
0      206504       10006   142345           MICU 2164-10-23 21:10:15
1      232110       10011   105331           MICU 2126-08-14 22:34:00
2      264446       10013   165520           MICU 2125-10-04 23:38:00
3      204881       10017   199207            CCU 2149-05-29 18:52:29
4      228977       10019   177759           MICU 2163-05-14 20:43:56

              outtime      los
0 2164-10-25 12:21:07   1.6325
1 2126-08-28 18:59:00  13.8507
2 2125-10-07 15:13:52   2.6499
3 2149-05-31 22:19:17   2.1436
4 2163-05-16 03:47:04   1.2938
```

[ ]:
```
# ICU unit distribution
query_df("""
    SELECT first_careunit, COUNT(*) as stays,
           ROUND(AVG(los)::numeric, 2) as avg_los_days
    FROM bronze.icustays
    GROUP BY first_careunit
    ORDER BY stays DESC
""", limit=None)
```

```
2026-01-01 14:34:04,896 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:04,897 INFO sqlalchemy.engine.Engine
    SELECT first_careunit, COUNT(*) as stays,
           ROUND(AVG(los)::numeric, 2) as avg_los_days
    FROM bronze.icustays
    GROUP BY first_careunit
    ORDER BY stays DESC

2026-01-01 14:34:04,898 INFO sqlalchemy.engine.Engine [generated in 0.00102s] {}
2026-01-01 14:34:04,903 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:
```
  first_careunit  stays avg_los_days
0           MICU     77         3.96
1           SICU     23         5.67
2            CCU     19         5.75
3          TSICU     11         3.59
4           CSRU      6         3.63
```

1.4 4. LABEVENTS Table

Description: Laboratory test results. One row per lab test.

Column	Type	Description
row_id	INTEGER	**Primary key**
subject_id	INTEGER	Patient ID
hadm_id	INTEGER	Hospital admission ID
itemid	INTEGER	Lab test ID (FK to d_labitems)
charttime	TIMESTAMP	Time of measurement
value	VARCHAR	Test result (text)
valuenum	FLOAT	Test result (numeric)
valueuom	VARCHAR	Unit of measurement
flag	VARCHAR	Abnormal flag (abnormal/delta)

[ ]:
```
print(f"Total Lab Events: {table_count('bronze', 'labevents'):,}")
query_df("SELECT row_id, subject_id, itemid, charttime, value, valuenum,␣
 ↪valueuom, flag FROM bronze.labevents LIMIT 5")
```

```
2026-01-01 14:34:08,780 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:08,781 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.labevents
2026-01-01 14:34:08,783 INFO sqlalchemy.engine.Engine [generated in 0.00139s] {}
2026-01-01 14:34:08,840 INFO sqlalchemy.engine.Engine COMMIT
Total Lab Events: 76,074
2026-01-01 14:34:08,845 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:08,846 INFO sqlalchemy.engine.Engine SELECT row_id, subject_id,
itemid, charttime, value, valuenum, valueuom, flag FROM bronze.labevents LIMIT 5
2026-01-01 14:34:08,847 INFO sqlalchemy.engine.Engine [generated in 0.00095s] {}
2026-01-01 14:34:08,855 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:

	row_id	subject_id	itemid	charttime	value	valuenum	valueuom	flag
0	6244563	10006	50868	2164-09-24 20:21:00	19	19.0	mEq/L	None
1	6244564	10006	50882	2164-09-24 20:21:00	27	27.0	mEq/L	None
2	6244565	10006	50893	2164-09-24 20:21:00	10.0	10.0	mg/dL	None
3	6244566	10006	50902	2164-09-24 20:21:00	97	97.0	mEq/L	None
4	6244567	10006	50912	2164-09-24 20:21:00	7.0	7.0	mg/dL	abnormal

[ ]:
```
# Most common lab tests
query_df("""
    SELECT l.itemid, d.label, COUNT(*) as test_count
```

```
    FROM bronze.labevents l
    JOIN bronze.d_labitems d ON l.itemid = d.itemid
    GROUP BY l.itemid, d.label
    ORDER BY test_count DESC
    LIMIT 10
""")
```

```
2026-01-01 14:34:11,253 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:11,255 INFO sqlalchemy.engine.Engine
    SELECT l.itemid, d.label, COUNT(*) as test_count
    FROM bronze.labevents l
    JOIN bronze.d_labitems d ON l.itemid = d.itemid
    GROUP BY l.itemid, d.label
    ORDER BY test_count DESC
    LIMIT 10

2026-01-01 14:34:11,256 INFO sqlalchemy.engine.Engine [generated in 0.00080s] {}
2026-01-01 14:34:11,362 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:

	itemid	label	test_count
0	51221	Hematocrit	2317
1	50971	Potassium	2279
2	50983	Sodium	2185
3	50912	Creatinine	2175
4	50902	Chloride	2160
5	51006	Urea Nitrogen	2158
6	50882	Bicarbonate	2151
7	50868	Anion Gap	2134
8	50931	Glucose	2121
9	51265	Platelet Count	2088

1.5 5. PRESCRIPTIONS Table

Description: Medication prescriptions. One row per prescription. `text | Column | Type | Description | |--------|------|-------------| | row_id | INTEGER | **Primary key** | | subject_id | INTEGER | Patient ID | | hadm_id | INTEGER | Hospital admission ID | | icustay_id | INTEGER | ICU stay ID | | startdate | TIMESTAMP | Prescription start | | enddate | TIMESTAMP | Prescription end | | drug_type | VARCHAR | Type of drug | | drug | VARCHAR | Drug name | | drug_name_generic | VARCHAR | Generic drug name | | dose_val_rx | VARCHAR | Dose value | | dose_unit_rx | VARCHAR | Dose unit | | route | VARCHAR | Administration route |text`

[ ]:
```
print(f"Total Prescriptions: {table_count('bronze', 'prescriptions'):,}")
query_df("SELECT row_id, subject_id, drug, drug_name_generic, dose_val_rx,␣
 ↪dose_unit_rx, route FROM bronze.prescriptions LIMIT 5")
```

```
2026-01-01 14:34:15,701 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:15,703 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.prescriptions
2026-01-01 14:34:15,705 INFO sqlalchemy.engine.Engine [generated in 0.00200s] {}
2026-01-01 14:34:15,726 INFO sqlalchemy.engine.Engine COMMIT
Total Prescriptions: 10,398
2026-01-01 14:34:15,729 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:15,730 INFO sqlalchemy.engine.Engine SELECT row_id, subject_id,
drug, drug_name_generic, dose_val_rx, dose_unit_rx, route FROM
bronze.prescriptions LIMIT 5
2026-01-01 14:34:15,730 INFO sqlalchemy.engine.Engine [generated in 0.00072s] {}
2026-01-01 14:34:15,738 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:

	row_id	subject_id	drug	drug_name_generic	dose_val_rx	dose_unit_rx	route
0	32600	42458	Pneumococcal Vac Polyvalent	PNEUMOcoccal Vac Polyvalent	0.5	mL	IM
1	32601	42458	Bisacodyl	Bisacodyl	10	mg	PO
2	32602	42458	Bisacodyl	Bisacodyl (Rectal)	10	mg	PR
3	32603	42458	Senna	Senna	1	TAB	PO
4	32604	42458	Docusate Sodium (Liquid)	Docusate Sodium (Liquid)	100	mg	PO

[ ]:
```
# Most prescribed drugs
query_df("""
    SELECT drug, COUNT(*) as prescription_count,
           COUNT(DISTINCT subject_id) as unique_patients
    FROM bronze.prescriptions
    WHERE drug IS NOT NULL
    GROUP BY drug
    ORDER BY prescription_count DESC
    LIMIT 10
""")
```

```
2026-01-01 14:34:18,785 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:18,786 INFO sqlalchemy.engine.Engine
    SELECT drug, COUNT(*) as prescription_count,
           COUNT(DISTINCT subject_id) as unique_patients
    FROM bronze.prescriptions
    WHERE drug IS NOT NULL
    GROUP BY drug
    ORDER BY prescription_count DESC
    LIMIT 10
```

```
2026-01-01 14:34:18,787 INFO sqlalchemy.engine.Engine [generated in 0.00076s] {}
2026-01-01 14:34:18,926 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:

	drug	prescription_count	unique_patients
0	Potassium Chloride	529	66
1	D5W	439	68
2	0.9% Sodium Chloride	409	39
3	NS	362	56
4	Furosemide	346	52
5	Insulin	300	67
6	Iso-Osmotic Dextrose	265	60
7	5% Dextrose	256	32
8	SW	244	51
9	Magnesium Sulfate	206	59

1.6 6. D_LABITEMS (Dictionary Table)

Description: Lab item dictionary. Defines all lab tests.

Column	Type	Description
row_id	INTEGER	Row identifier
itemid	INTEGER	**Primary key** - Lab test ID
label	VARCHAR	Human-readable test name
fluid	VARCHAR	Specimen type (Blood, Urine, etc.)
category	VARCHAR	Test category
loinc_code	VARCHAR	LOINC code

[ ]:
```
print(f"Total Lab Items: {table_count('bronze', 'd_labitems'):,}")
query_df("SELECT itemid, label, fluid, category, loinc_code FROM bronze.
 ↪d_labitems LIMIT 10")
```

```
2026-01-01 14:34:23,657 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:23,658 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.d_labitems
2026-01-01 14:34:23,659 INFO sqlalchemy.engine.Engine [generated in 0.00133s] {}
2026-01-01 14:34:23,663 INFO sqlalchemy.engine.Engine COMMIT
Total Lab Items: 753
2026-01-01 14:34:23,667 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:23,668 INFO sqlalchemy.engine.Engine SELECT itemid, label,
fluid, category, loinc_code FROM bronze.d_labitems LIMIT 10
2026-01-01 14:34:23,669 INFO sqlalchemy.engine.Engine [generated in 0.00101s] {}
2026-01-01 14:34:23,673 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:

	itemid	label	fluid	category	loinc_code
0	50800	SPECIMEN TYPE	BLOOD	BLOOD GAS	None
1	50801	Alveolar-arterial Gradient	Blood	Blood Gas	19991-9

```
2   50802                          Base Excess  Blood  Blood Gas    11555-0
3   50803  Calculated Bicarbonate, Whole Blood  Blood  Blood Gas     1959-6
4   50804                  Calculated Total CO2  Blood  Blood Gas    34728-6
5   50805                     Carboxyhemoglobin  Blood  Blood Gas    20563-3
6   50806                 Chloride, Whole Blood  Blood  Blood Gas     2069-3
7   50807                              Comments  Blood  Blood Gas       None
8   50808                          Free Calcium  Blood  Blood Gas     1994-3
9   50809                               Glucose  Blood  Blood Gas     2339-0
```

1.7 7. D_ITEMS (Dictionary Table)

Description: Item dictionary for procedures, inputs, outputs.

```
| Column | Type | Description |
|--------|------|-------------|
| itemid | INTEGER | **Primary key** - Item ID |
| label | VARCHAR | Human-readable item name |
| abbreviation | VARCHAR | Short name |
| dbsource | VARCHAR | Data source |
| linksto | VARCHAR | Related table |
| category | VARCHAR | Item category |
| unitname | VARCHAR | Unit of measurement |
```

[ ]:
```
print(f"Total Items: {table_count('bronze', 'd_items'):,}")
query_df("SELECT itemid, label, category, dbsource, linksto FROM bronze.d_items␣
 ↪LIMIT 10")
```

```
2026-01-01 14:34:28,657 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:28,659 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.d_items
2026-01-01 14:34:28,660 INFO sqlalchemy.engine.Engine [generated in 0.00105s] {}
2026-01-01 14:34:28,778 INFO sqlalchemy.engine.Engine COMMIT
Total Items: 12,487
2026-01-01 14:34:28,783 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:28,784 INFO sqlalchemy.engine.Engine SELECT itemid, label,
category, dbsource, linksto FROM bronze.d_items LIMIT 10
2026-01-01 14:34:28,786 INFO sqlalchemy.engine.Engine [generated in 0.00123s] {}
2026-01-01 14:34:28,790 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:
```
   itemid                label category dbsource      linksto
0    1435    Sustained Nystamus     None  carevue  chartevents
1    1436  Tactile Disturbances     None  carevue  chartevents
2    1437               Tremor     None  carevue  chartevents
3    1438   Ulnar Pulse [Right]     None  carevue  chartevents
4    1439   Visual Disturbances     None  carevue  chartevents
5    1447   Transpulmonary Pres     None  carevue  chartevents
6    1448               Vd/Vt:     None  carevue  chartevents
```

```
7    1449      Arterial BP(Rad)     None  carevue  chartevents
8    1450             level one     None  carevue  chartevents
9    1451          L girth size     None  carevue  chartevents
```

1.8 Bronze Layer Summary

The Bronze layer contains raw MIMIC-III data with the following characteristics: - **No transformations** applied - **Original data types** preserved - **All columns** from source CSVs - Ready for Silver layer transformation

[ ]:
```
# Bronze layer summary
bronze_tables = ['patients', 'admissions', 'icustays', 'labevents',␣
 ↪'prescriptions',
                 'transfers', 'inputevents_mv', 'outputevents',␣
 ↪'procedureevents_mv',
                 'microbiologyevents', 'd_items', 'd_labitems']

summary = []
for table in bronze_tables:
    try:
        count = table_count('bronze', table)
        summary.append({'table': table, 'count': count})
    except:
        summary.append({'table': table, 'count': 'N/A'})

pd.DataFrame(summary)
```

```
2026-01-01 14:34:32,641 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:32,642 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.patients
2026-01-01 14:34:32,643 INFO sqlalchemy.engine.Engine [cached since 89.36s ago]
{}
2026-01-01 14:34:32,646 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 14:34:32,651 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:32,652 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.admissions
2026-01-01 14:34:32,653 INFO sqlalchemy.engine.Engine [cached since 89.11s ago]
{}
2026-01-01 14:34:32,656 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 14:34:32,659 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:32,661 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.icustays
2026-01-01 14:34:32,662 INFO sqlalchemy.engine.Engine [cached since 88.98s ago]
{}
2026-01-01 14:34:32,665 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 14:34:32,668 INFO sqlalchemy.engine.Engine BEGIN (implicit)
```

```
2026-01-01 14:34:32,670 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.labevents
2026-01-01 14:34:32,670 INFO sqlalchemy.engine.Engine [cached since 23.89s ago]
{}
2026-01-01 14:34:32,678 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 14:34:32,682 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:32,683 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.prescriptions
2026-01-01 14:34:32,684 INFO sqlalchemy.engine.Engine [cached since 16.98s ago]
{}
2026-01-01 14:34:32,687 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 14:34:32,690 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:32,691 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.transfers
2026-01-01 14:34:32,692 INFO sqlalchemy.engine.Engine [generated in 0.00089s] {}
2026-01-01 14:34:32,712 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 14:34:32,716 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:32,717 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.inputevents_mv
2026-01-01 14:34:32,718 INFO sqlalchemy.engine.Engine [generated in 0.00086s] {}
2026-01-01 14:34:32,746 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 14:34:32,748 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:32,749 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.outputevents
2026-01-01 14:34:32,749 INFO sqlalchemy.engine.Engine [generated in 0.00071s] {}
2026-01-01 14:34:32,761 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 14:34:32,764 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:32,765 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.procedureevents_mv
2026-01-01 14:34:32,766 INFO sqlalchemy.engine.Engine [generated in 0.00096s] {}
2026-01-01 14:34:32,778 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 14:34:32,781 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:32,782 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.microbiologyevents
2026-01-01 14:34:32,782 INFO sqlalchemy.engine.Engine [generated in 0.00066s] {}
2026-01-01 14:34:32,788 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 14:34:32,791 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:32,793 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.d_items
2026-01-01 14:34:32,793 INFO sqlalchemy.engine.Engine [cached since 4.135s ago]
{}
2026-01-01 14:34:32,796 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 14:34:32,798 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 14:34:32,799 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM
bronze.d_labitems
2026-01-01 14:34:32,800 INFO sqlalchemy.engine.Engine [cached since 9.142s ago]
{}
2026-01-01 14:34:32,803 INFO sqlalchemy.engine.Engine COMMIT
```

[ ]:

	table	count
0	patients	100
1	admissions	129
2	icustays	136
3	labevents	76074
4	prescriptions	10398
5	transfers	524
6	inputevents_mv	13224
7	outputevents	11320
8	procedureevents_mv	753
9	microbiologyevents	2003
10	d_items	12487
11	d_labitems	753