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

Copyright & Disclaimer

This report is provided for **educational and professional demonstration purposes**. The MIMIC-III dataset remains the property of its respective owners and is used in accordance with applicable data usage agreements.

No patient-identifiable information is disclosed or altered in this document.

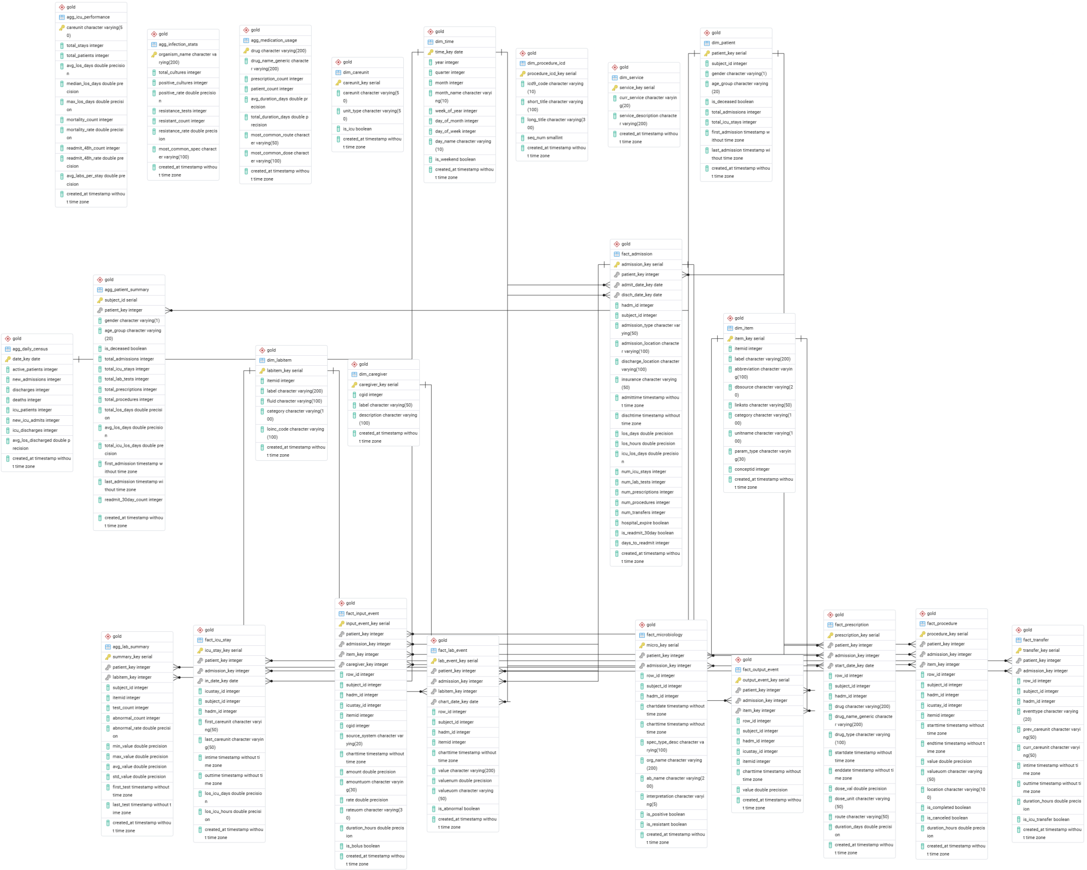
🥇 Gold Layer Analysis

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

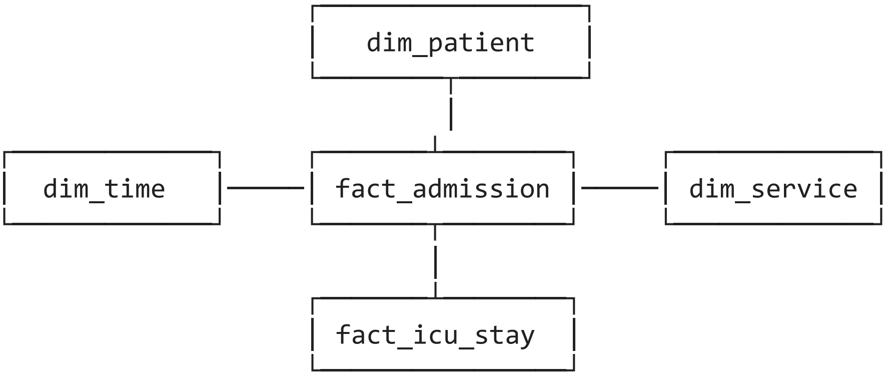
What is the Gold Layer?

The Gold layer contains **analytics-ready data** organized as a dimensional model:

- **Dimension Tables**: Reference data for filtering/grouping
- **Fact Tables**: Transactional data with measures
- **Aggregate Tables**: Pre-computed summaries



Star Schema Design



In [ ]:

```python
# 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)
```

In [ ]:

```python
# Helper functions
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}")).
```

DIMENSION TABLES

Dimension tables provide the **context** for fact data - the “who, what, when, where” of analysis.

1. DIM_PATIENT

Purpose: Master patient dimension with lifetime statistics.

Column	Type	Description
patient_key	INTEGER	**Surrogate PK** (auto-increment)
subject_id	INTEGER	**Natural key** - Patient ID
gender	VARCHAR	M/F
dob	TIMESTAMP	Date of birth
dod	TIMESTAMP	Date of death
is_deceased	BOOLEAN	Mortality flag
total_admissions	INTEGER	Lifetime admissions count
total_icu_stays	INTEGER	Lifetime ICU stays
first_admission	TIMESTAMP	First hospital visit
last_admission	TIMESTAMP	Most recent visit

In [3]:

```
print(f"dim_patient: {table_count('gold', 'dim_patient'):,} records")
query_df("SELECT * FROM gold.dim_patient LIMIT 5")
```

```
2026-01-01 13:41:01,161 INFO sqlalchemy.engine.Engine select pg_catalog.version()
2026-01-01 13:41:01,164 INFO sqlalchemy.engine.Engine [raw sql] {}
2026-01-01 13:41:01,182 INFO sqlalchemy.engine.Engine select current_schema()
2026-01-01 13:41:01,184 INFO sqlalchemy.engine.Engine [raw sql] {}
2026-01-01 13:41:01,205 INFO sqlalchemy.engine.Engine show standard_conforming_strings
2026-01-01 13:41:01,209 INFO sqlalchemy.engine.Engine [raw sql] {}
2026-01-01 13:41:01,218 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:01,222 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gold.dim_patient
2026-01-01 13:41:01,224 INFO sqlalchemy.engine.Engine [generated in 0.00206s] {}
2026-01-01 13:41:01,269 INFO sqlalchemy.engine.Engine COMMIT
dim_patient: 100 records
2026-01-01 13:41:01,281 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:01,283 INFO sqlalchemy.engine.Engine SELECT * FROM gold.dim_patient LIMIT 5
2026-01-01 13:41:01,288 INFO sqlalchemy.engine.Engine [generated in 0.00418s] {}
2026-01-01 13:41:01,341 INFO sqlalchemy.engine.Engine COMMIT
```

Out[3]:

	patient_key	subject_id	gender	age_group	is_deceased	total_admissions	total_icu_stays	first_admission	last_admission	created_at
0	4	10017	F	None	True	1	1	2149-05-26 17:19:00	2149-05-26 17:19:00	2025-12-26 00:08:42.452185
1	5	10019	M	None	True	1	1	2163-05-14 20:43:00	2163-05-14 20:43:00	2025-12-26 00:08:42.452185
2	6	10026	F	None	True	1	1	2195-05-17 07:39:00	2195-05-17 07:39:00	2025-12-26 00:08:42.452185
3	7	10027	F	None	True	1	1	2190-07-13 07:15:00	2190-07-13 07:15:00	2025-12-26 00:08:42.452185
4	8	10029	M	None	True	1	1	2139-09-22 10:58:00	2139-09-22 10:58:00	2025-12-26 00:08:42.452185

2. DIM_TIME

Purpose: Date dimension for time-based analysis.

Column	Type	Description
time_key	DATE	**PK** - Calendar date
year	INTEGER	Year (e.g., 2100)
quarter	INTEGER	Quarter (1-4)

Column	Type	Description
month	INTEGER	Month (1-12)
month_name	VARCHAR	Month name
week_of_year	INTEGER	ISO week number
day_of_month	INTEGER	Day (1-31)
day_of_week	INTEGER	Weekday (0=Mon)
day_name	VARCHAR	Day name
is_weekend	BOOLEAN	Saturday/Sunday

In [4]:

```
print(f"dim_time: {table_count('gold', 'dim_time'):,} records")
query_df("SELECT * FROM gold.dim_time LIMIT 5")
```

```
2026-01-01 12:28:16,795 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 12:28:16,797 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.dim_time
2026-01-01 12:28:16,798 INFO sqlalchemy.engine.Engine [generated in 0.00100s]
{}
2026-01-01 12:28:16,809 INFO sqlalchemy.engine.Engine COMMIT
dim_time: 38,715 records
2026-01-01 12:28:16,813 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 12:28:16,814 INFO sqlalchemy.engine.Engine SELECT * FROM gold.dim_ti
me LIMIT 5
2026-01-01 12:28:16,816 INFO sqlalchemy.engine.Engine [generated in 0.00146s]
{}
2026-01-01 12:28:16,823 INFO sqlalchemy.engine.Engine COMMIT
```

Out[4]:

	time_key	year	quarter	month	month_name	week_of_year	day_of_month	day_of_week	day_name	is_weekend	created_at
0	2100-01-01	2100	1	1	January	53	1	4	Friday	False	2025-12-26 00:07:01.315552
1	2100-01-02	2100	1	1	January	53	2	5	Saturday	True	2025-12-26 00:07:01.315552
2	2100-01-03	2100	1	1	January	53	3	6	Sunday	True	2025-12-26 00:07:01.315552
3	2100-01-04	2100	1	1	January	1	4	0	Monday	False	2025-12-26 00:07:01.315552
4	2100-01-05	2100	1	1	January	1	5	1	Tuesday	False	2025-12-26 00:07:01.315552

3. DIM_CAREUNIT

Purpose: ICU care unit dimension.

Column	Type	Description
careunit	VARCHAR	**PK** - Unit code (e.g., MICU)
careunit_name	VARCHAR	Full name
unit_type	VARCHAR	Type classification

In [5]:

```
print(f"dim_careunit: {table_count('gold', 'dim_careunit'):,} records")
query_df("SELECT * FROM gold.dim_careunit", limit=None)
```

```
2026-01-01 12:28:41,232 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 12:28:41,233 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.dim_careunit
2026-01-01 12:28:41,234 INFO sqlalchemy.engine.Engine [generated in 0.00060s]
{}
2026-01-01 12:28:41,241 INFO sqlalchemy.engine.Engine COMMIT
dim_careunit: 5 records
2026-01-01 12:28:41,245 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 12:28:41,246 INFO sqlalchemy.engine.Engine SELECT * FROM gold.dim_ca
reunit
2026-01-01 12:28:41,247 INFO sqlalchemy.engine.Engine [generated in 0.00079s]
{}
2026-01-01 12:28:41,251 INFO sqlalchemy.engine.Engine COMMIT
```

Out[5]:

	careunit_key	careunit	unit_type	is_icu	created_at
0	1	MICU	ICU	True	2025-12-26 00:08:43.702866
1	2	CCU	ICU	True	2025-12-26 00:08:43.702866
2	3	TSICU	ICU	True	2025-12-26 00:08:43.702866
3	4	SICU	ICU	True	2025-12-26 00:08:43.702866
4	5	CSRU	ICU	True	2025-12-26 00:08:43.702866

4. DIM_ITEM

Purpose: Item dictionary (procedures, inputs, outputs).

Column	Type	Description
item_key	INTEGER	**Surrogate PK**
itemid	INTEGER	Natural key
label	VARCHAR	Item name
abbreviation	VARCHAR	Short name
category	VARCHAR	Category
unitname	VARCHAR	Unit of measure

In [6]:

```
print(f"dim_item: {table_count('gold', 'dim_item'):,} records")
query_df("SELECT item_key, itemid, label, category FROM gold.dim_item LIMIT 10"
```

```
2026-01-01 12:28:52,295 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 12:28:52,296 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.dim_item
2026-01-01 12:28:52,297 INFO sqlalchemy.engine.Engine [generated in 0.00095s]
{}
2026-01-01 12:28:52,309 INFO sqlalchemy.engine.Engine COMMIT
dim_item: 12,487 records
2026-01-01 12:28:52,312 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 12:28:52,313 INFO sqlalchemy.engine.Engine SELECT item_key, itemid,
label, category FROM gold.dim_item LIMIT 10
2026-01-01 12:28:52,314 INFO sqlalchemy.engine.Engine [generated in 0.00103s]
{}
2026-01-01 12:28:52,320 INFO sqlalchemy.engine.Engine COMMIT
```

Out[6]:

	item_key	itemid	label	category
0	1	1435	Sustained Nystamus	None
1	2	1436	Tactile Disturbances	None
2	3	1437	Tremor	None
3	4	1438	Ulnar Pulse [Right]	None
4	5	1439	Visual Disturbances	None
5	6	1447	Transpulmonary Pres	None
6	7	1448	Vd/Vt:	None
7	8	1449	Arterial BP(Rad)	None
8	9	1450	level one	None
9	10	1451	L girth size	None

5. DIM_LABITEM

Purpose: Lab test dictionary.

Column	Type	Description
labitem_key	INTEGER	**Surrogate PK**
itemid	INTEGER	Natural key
label	VARCHAR	Test name
fluid	VARCHAR	Specimen type
category	VARCHAR	Test category
loinc_code	VARCHAR	LOINC code

In [7]:

```
print(f"dim_labitem: {table_count('gold', 'dim_labitem'):,} records")
```

```
query_df("SELECT labitem_key, itemid, label, fluid, category FROM gold.dim_labi
```

```
2026-01-01 12:29:04,463 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 12:29:04,464 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.dim_labitem
2026-01-01 12:29:04,465 INFO sqlalchemy.engine.Engine [generated in 0.00103s]
{}
2026-01-01 12:29:04,473 INFO sqlalchemy.engine.Engine COMMIT
dim_labitem: 753 records
2026-01-01 12:29:04,477 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 12:29:04,477 INFO sqlalchemy.engine.Engine SELECT labitem_key, itemi
d, label, fluid, category FROM gold.dim_labitem LIMIT 10
2026-01-01 12:29:04,478 INFO sqlalchemy.engine.Engine [generated in 0.00056s]
{}
2026-01-01 12:29:04,481 INFO sqlalchemy.engine.Engine COMMIT
```

Out[7]:

	labitem_key	itemid	label	fluid	category
0	1	50800	SPECIMEN TYPE	BLOOD	BLOOD GAS
1	2	50801	Alveolar-arterial Gradient	Blood	Blood Gas
2	3	50802	Base Excess	Blood	Blood Gas
3	4	50803	Calculated Bicarbonate, Whole Blood	Blood	Blood Gas
4	5	50804	Calculated Total CO2	Blood	Blood Gas
5	6	50805	Carboxyhemoglobin	Blood	Blood Gas
6	7	50806	Chloride, Whole Blood	Blood	Blood Gas
7	8	50807	Comments	Blood	Blood Gas
8	9	50808	Free Calcium	Blood	Blood Gas
9	10	50809	Glucose	Blood	Blood Gas

FACT TABLES

Fact tables contain the **measurable events** - admissions, lab tests, prescriptions, etc.

6. FACT_ADMISSION

Purpose: Hospital admissions with enriched metrics.

Column	Type	Description
admission_key	INTEGER	**Surrogate PK**
hadm_id	INTEGER	**NK** - Admission ID
patient_key	INTEGER	FK to dim_patient
admit_date_key	DATE	FK to dim_time

Column	Type	Description
disch_date_key	DATE	FK to dim_time
admission_type	VARCHAR	EMERGENCY, ELECTIVE, etc.
los_days	FLOAT	Length of stay (days)
los_hours	FLOAT	Length of stay (hours)
num_icu_stays	INTEGER	**Aggregated** ICU stays
num_lab_tests	INTEGER	**Aggregated** lab tests
num_prescriptions	INTEGER	**Aggregated** prescriptions
hospital_expire	BOOLEAN	Died in hospital

In [8]:

```
print(f"fact_admission: {table_count('gold', 'fact_admission'):,} records")
query_df("""
    SELECT hadm_id, admission_type, los_days,
           num_icu_stays, num_lab_tests, num_prescriptions, hospital_expire
    FROM gold.fact_admission
    ORDER BY num_lab_tests DESC
    LIMIT 5
""")
```

```
2026-01-01 12:29:17,174 INFO sqlalchemy.engine.Engine BEGIN
(implicit) 2026-01-01 12:29:17,175 INFO sqlalchemy.engine.Engine SELECT
COUNT(*) FROM gold.fact_admission 2026-01-01 12:29:17,177 INFO
sqlalchemy.engine.Engine [generated in 0.00133s] {} 2026-01-01
12:29:17,186 INFO sqlalchemy.engine.Engine COMMIT fact_admission: 129
records 2026-01-01 12:29:17,190 INFO sqlalchemy.engine.Engine BEGIN
(implicit) 2026-01-01 12:29:17,191 INFO sqlalchemy.engine.Engine SELECT
hadm_id, admission_type, los_days, num_icu_stays, num_lab_tests,
num_prescriptions, hospital_expire FROM gold.fact_admission ORDER BY
num_lab_tests DESC LIMIT 5
```

2026-01-01 12:29:17,192 INFO sqlalchemy.engine.Engine [generated in 0.00075s] {} 2026-01-01 12:29:17,198 INFO sqlalchemy.engine.Engine COMMIT

Out[8]:

	hadm_id	admission_type	los_days	num_icu_stays	num_lab_tests	num_prescriptions	hospital_expire
0	186361	EMERGENCY	123.98	1	6880	784	False
1	160445	EMERGENCY	35.25	1	4967	437	True
2	163189	EMERGENCY	36.01	1	3093	209	False
3	145203	EMERGENCY	25.00	1	1841	314	False
4	170883	EMERGENCY	28.94	3	1478	244	True

In [10]:

```
# Admission analysis with dimensions
query_df("""
    SELECT
        fa.admission_type,
        COUNT(*) as admissions,
        ROUND(AVG(fa.los_days)::numeric, 2) AS avg_los,
        SUM(fa.num_icu_stays) as total_icu_stays,
        SUM(fa.num_lab_tests) as total_labs
    FROM gold.fact_admission fa
    GROUP BY fa.admission_type
    ORDER BY admissions DESC
""", limit=None)
```

```
2026-01-01 12:31:31,561 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 12:31:31,562 INFO sqlalchemy.engine.Engine
    SELECT
        fa.admission_type,
        COUNT(*) as admissions,
        ROUND(AVG(fa.los_days)::numeric, 2) AS avg_los,
        SUM(fa.num_icu_stays) as total_icu_stays,
        SUM(fa.num_lab_tests) as total_labs
    FROM gold.fact_admission fa
    GROUP BY fa.admission_type
    ORDER BY admissions DESC

2026-01-01 12:31:31,563 INFO sqlalchemy.engine.Engine [generated in 0.00103s]
{}
2026-01-01 12:31:31,572 INFO sqlalchemy.engine.Engine COMMIT
```

Out[10]:

	admission_type	admissions	avg_los	total_icu_stays	total_labs
0	EMERGENCY	119	9.23	126	58159
1	ELECTIVE	8	11.67	8	3139
2	URGENT	2	6.26	2	514

7. FACT_ICU_STAY

Purpose: ICU stay facts with dimensional keys.

Column	Type	Description
icu_stay_key	INTEGER	**Surrogate PK**
icustay_id	INTEGER	**NK** - ICU stay ID
patient_key	INTEGER	FK to dim_patient
in_date_key	DATE	FK to dim_time
first_careunit	VARCHAR	First ICU unit
last_careunit	VARCHAR	Last ICU unit
los_icu_days	FLOAT	ICU length of stay
los_icu_hours	FLOAT	ICU LOS (hours)

In [11]:

```
print(f"fact_icu_stay: {table_count('gold', 'fact_icu_stay'):,} records")
query_df("""
    SELECT icustay_id, first_careunit, last_careunit, los_icu_days, los_icu_hou
    FROM gold.fact_icu_stay
    ORDER BY los_icu_days DESC
    LIMIT 5
""")
```

```
2026-01-01 12:31:42,520 INFO sqlalchemy.engine.Engine BEGIN
(implicit) 2026-01-01 12:31:42,521 INFO sqlalchemy.engine.Engine SELECT
COUNT(*) FROM gold.fact_icu_stay 2026-01-01 12:31:42,523 INFO
sqlalchemy.engine.Engine [generated in 0.00126s] {} 2026-01-01
12:31:42,534 INFO sqlalchemy.engine.Engine COMMIT fact_icu_stay: 136
records 2026-01-01 12:31:42,538 INFO sqlalchemy.engine.Engine BEGIN
(implicit) 2026-01-01 12:31:42,540 INFO sqlalchemy.engine.Engine SELECT
icustay_id, first_careunit, last_careunit, los_icu_days, los_icu_hours
FROM gold.fact_icu_stay ORDER BY los_icu_days DESC LIMIT 5
```

2026-01-01 12:31:42,541 INFO sqlalchemy.engine.Engine [generated in 0.00093s] {} 2026-01-01 12:31:42,545 INFO sqlalchemy.engine.Engine COMMIT

Out[11]:

	icustay_id	first_careunit	last_careunit	los_icu_days	los_icu_hours
0	249805	SICU	SICU	35.41	849.76
1	239396	MICU	MICU	31.12	746.96
2	250305	SICU	SICU	29.26	702.24
3	223177	CCU	CCU	25.00	599.92
4	271544	TSICU	TSICU	22.39	537.35

8. FACT_LAB_EVENT

Purpose: Lab test results with dimension keys.

Column	Type	Description
lab_event_key	INTEGER	**Surrogate PK**
row_id	INTEGER	Original row ID
patient_key	INTEGER	FK to dim_patient
labitem_key	INTEGER	FK to dim_labitem
chart_date_key	DATE	FK to dim_time
valuenum	FLOAT	Numeric result
valueuom	VARCHAR	Unit of measure
is_abnormal	BOOLEAN	Abnormal flag

In [12]:

```
print(f"fact_lab_event: {table_count('gold', 'fact_lab_event'):,} records")
query_df("""
    SELECT row_id, itemid, charttime, valuenum, valueuom, is_abnormal
    FROM gold.fact_lab_event
    WHERE valuenum IS NOT NULL
    LIMIT 5
""")
```

```
2026-01-01 12:31:52,708 INFO sqlalchemy.engine.Engine BEGIN
(implicit) 2026-01-01 12:31:52,710 INFO sqlalchemy.engine.Engine SELECT
COUNT(*) FROM gold.fact_lab_event 2026-01-01 12:31:52,712 INFO
sqlalchemy.engine.Engine [generated in 0.00155s] {} 2026-01-01
12:31:52,734 INFO sqlalchemy.engine.Engine COMMIT fact_lab_event: 76,074
records 2026-01-01 12:31:52,738 INFO sqlalchemy.engine.Engine BEGIN
(implicit) 2026-01-01 12:31:52,739 INFO sqlalchemy.engine.Engine SELECT
row_id, itemid, charttime, valuenum, valueuom, is_abnormal FROM
gold.fact_lab_event WHERE valuenum IS NOT NULL LIMIT 5
```

2026-01-01 12:31:52,740 INFO sqlalchemy.engine.Engine [generated in 0.00071s] {} 2026-01-01 12:31:52,747 INFO sqlalchemy.engine.Engine COMMIT

Out[12]:

	row_id	itemid	charttime	valuenum	valueuom	is_abnormal
0	6245566	50912	2164-11-16 05:50:00	2.1	mg/dL	True
1	6245567	50931	2164-11-16 05:50:00	162.0	mg/dL	True
2	6245568	50960	2164-11-16 05:50:00	1.0	mg/dL	True
3	6245569	50970	2164-11-16 05:50:00	3.6	mg/dL	False
4	6245570	50971	2164-11-16 05:50:00	2.6	mEq/L	True

In [13]:

```
# Lab analysis with dimension join
query_df("""
    SELECT
        dl.label as test_name,
        dl.fluid,
        COUNT(*) as test_count,
        SUM(CASE WHEN fl.is_abnormal THEN 1 ELSE 0 END) as abnormal_count,
        ROUND(100.0 * SUM(CASE WHEN fl.is_abnormal THEN 1 ELSE 0 END) / COUNT(*
    FROM gold.fact_lab_event fl
    JOIN gold.dim_labitem dl ON fl.labitem_key = dl.labitem_key
    GROUP BY dl.label, dl.fluid
    ORDER BY test_count DESC
    LIMIT 10
""")
```

```
2026-01-01 12:32:03,507 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 12:32:03,508 INFO sqlalchemy.engine.Engine
    SELECT
        dl.label as test_name,
        dl.fluid,
        COUNT(*) as test_count,
        SUM(CASE WHEN fl.is_abnormal THEN 1 ELSE 0 END) as abnormal_count,
        ROUND(100.0 * SUM(CASE WHEN fl.is_abnormal THEN 1 ELSE 0 END) / COUNT
(*), 1) as abnormal_pct
    FROM gold.fact_lab_event fl
    JOIN gold.dim_labitem dl ON fl.labitem_key = dl.labitem_key
    GROUP BY dl.label, dl.fluid
    ORDER BY test_count DESC
    LIMIT 10

2026-01-01 12:32:03,509 INFO sqlalchemy.engine.Engine [generated in 0.00058s]
{}
2026-01-01 12:32:03,569 INFO sqlalchemy.engine.Engine COMMIT
```

Out[13]:

	test_name	fluid	test_count	abnormal_count	abnormal_pct
0	Glucose	Blood	2404	1719	71.5
1	Hematocrit	Blood	2317	2192	94.6
2	Potassium	Blood	2279	279	12.2
3	Sodium	Blood	2185	471	21.6
4	Creatinine	Blood	2175	1025	47.1
5	Chloride	Blood	2160	715	33.1
6	Urea Nitrogen	Blood	2158	1262	58.5
7	Bicarbonate	Blood	2151	638	29.7
8	Hemoglobin	Blood	2150	2039	94.8
9	Anion Gap	Blood	2134	140	6.6

AGGREGATE TABLES

Pre-computed summaries for fast dashboard queries.

9. AGG_PATIENT_SUMMARY

Purpose: Per-patient lifetime statistics.

Column	Type	Description
subject_id	INTEGER	**PK** - Patient ID
total_admissions	INTEGER	Lifetime admission count

Column	Type	Description
total_icu_stays	INTEGER	Lifetime ICU count
total_lab_tests	INTEGER	Lifetime lab count
total_prescriptions	INTEGER	Lifetime prescriptions
total_los_days	FLOAT	Total days in hospital

In [14]:

```
print(f"agg_patient_summary: {table_count('gold', 'agg_patient_summary'):,} rec
query_df("""
    SELECT subject_id, total_admissions, total_icu_stays,
           total_lab_tests, total_prescriptions, total_los_days
    FROM gold.agg_patient_summary
    ORDER BY total_lab_tests DESC
    LIMIT 5
""")
```

```
2026-01-01 12:32:39,062 INFO sqlalchemy.engine.Engine BEGIN
(implicit) 2026-01-01 12:32:39,063 INFO sqlalchemy.engine.Engine SELECT
COUNT(*) FROM gold.agg_patient_summary 2026-01-01 12:32:39,063 INFO
sqlalchemy.engine.Engine [generated in 0.00063s] {} 2026-01-01
12:32:39,072 INFO sqlalchemy.engine.Engine COMMIT agg_patient_summary:
100 records 2026-01-01 12:32:39,076 INFO sqlalchemy.engine.Engine BEGIN
(implicit) 2026-01-01 12:32:39,077 INFO sqlalchemy.engine.Engine SELECT
subject_id, total_admissions, total_icu_stays, total_lab_tests,
total_prescriptions, total_los_days FROM gold.agg_patient_summary ORDER
BY total_lab_tests DESC LIMIT 5
```

2026-01-01 12:32:39,078 INFO sqlalchemy.engine.Engine [generated in 0.00103s] {} 2026-01-01 12:32:39,082 INFO sqlalchemy.engine.Engine COMMIT

Out[14]:

	subject_id	total_admissions	total_icu_stays	total_lab_tests	total_prescriptions	total_los_days
0	40310	2	2	7262	893	131.00
1	10126	1	1	5009	437	35.25
2	41976	15	15	4495	1158	84.40
3	44212	1	1	3093	209	36.01
4	42135	2	2	2522	282	33.69

10. AGG_ICU_PERFORMANCE

Purpose: ICU unit performance metrics.

Column	Type	Description
careunit	VARCHAR	**PK** - ICU unit
total_stays	INTEGER	Total ICU stays

Column	Type	Description
total_patients	INTEGER	Unique patients
avg_los_days	FLOAT	Average LOS
max_los_days	FLOAT	Maximum LOS

In [15]:

```
print(f"agg_icu_performance: {table_count('gold', 'agg_icu_performance'):,} rec
query_df("SELECT * FROM gold.agg_icu_performance ORDER BY total_stays DESC", li
```

```
2026-01-01 12:32:55,565 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 12:32:55,566 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gold.agg_icu_performance
2026-01-01 12:32:55,567 INFO sqlalchemy.engine.Engine [generated in 0.00086s] {}
2026-01-01 12:32:55,573 INFO sqlalchemy.engine.Engine COMMIT
agg_icu_performance: 5 records
2026-01-01 12:32:55,577 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 12:32:55,579 INFO sqlalchemy.engine.Engine SELECT * FROM gold.agg_icu_performance ORDER BY total_stays DESC
2026-01-01 12:32:55,580 INFO sqlalchemy.engine.Engine [generated in 0.00092s] {}
2026-01-01 12:32:55,584 INFO sqlalchemy.engine.Engine COMMIT
```

Out[15]:

	careunit	total_stays	total_patients	avg_los_days	median_los_days	max_los_days	mortality_count	mortality_rate	readmit_48h_count	readmit_48h_rate	avg_labs_per_stay	created_at
0	MICU	77	56	3.955455	None	31.12	0	None	0	None	None	2025-12-26 00:09:07.320716
1	SICU	23	20	5.669130	None	35.41	0	None	0	None	None	2025-12-26 00:09:07.320716
2	CCU	19	18	5.753684	None	25.00	0	None	0	None	None	2025-12-26 00:09:07.320716
3	TSICU	11	11	3.590909	None	22.39	0	None	0	None	None	2025-12-26 00:09:07.320716
4	CSRU	6	6	3.631667	None	8.14	0	None	0	None	None	2025-12-26 00:09:07.320716

11. AGG_MEDICATION_USAGE

Purpose: Drug usage statistics.

Column	Type	Description
drug	VARCHAR	**PK** - Drug name
prescription_count	INTEGER	Total prescriptions
patient_count	INTEGER	Unique patients

In [ ]:

```
print(f"agg_medication_usage: {table_count('gold', 'agg_medication_usage'):,} r
query_df("""
    SELECT drug, prescription_count, patient_count
    FROM gold.agg_medication_usage
    ORDER BY prescription_count DESC
    LIMIT 10
""")
```

12. AGG_INFECTION_STATS

Purpose: Infection/organism statistics.

Column	Type	Description
organism_name	VARCHAR	**PK** - Organism
total_cultures	INTEGER	Culture count
positive_cultures	INTEGER	Positive count
resistant_count	INTEGER	Resistant cultures

In [16]:

```
print(f"agg_infection_stats: {table_count('gold', 'agg_infection_stats'):,} rec
query_df("""
    SELECT organism_name, total_cultures, positive_cultures, resistant_count
    FROM gold.agg_infection_stats
    ORDER BY total_cultures DESC
    LIMIT 10
""")
```

```
2026-01-01 12:33:10,665 INFO sqlalchemy.engine.Engine BEGIN
(implicit) 2026-01-01 12:33:10,666 INFO sqlalchemy.engine.Engine SELECT
COUNT(*) FROM gold.agg_infection_stats 2026-01-01 12:33:10,667 INFO
sqlalchemy.engine.Engine [generated in 0.00107s] {} 2026-01-01
12:33:10,671 INFO sqlalchemy.engine.Engine COMMIT agg_infection_stats:
46 records 2026-01-01 12:33:10,673 INFO sqlalchemy.engine.Engine BEGIN
(implicit) 2026-01-01 12:33:10,674 INFO sqlalchemy.engine.Engine SELECT
organism_name, total_cultures, positive_cultures, resistant_count FROM
gold.agg_infection_stats ORDER BY total_cultures DESC LIMIT 10
```

2026-01-01 12:33:10,675 INFO sqlalchemy.engine.Engine [generated in 0.00058s] {} 2026-01-01 12:33:10,678 INFO sqlalchemy.engine.Engine COMMIT

Out[16]:

	organism_name	total_cultures	positive_cultures	resistant_count
0	ESCHERICHIA COLI	234	234	32
1	STAPH AUREUS COAG +	172	172	68
2	PROTEUS MIRABILIS	107	107	43
3	KLEBSIELLA PNEUMONIAE	101	101	23
4	PSEUDOMONAS AERUGINOSA	98	98	31
5	ACINETOBACTER BAUMANNII COMPLEX	58	58	11
6	STAPHYLOCOCCUS, COAGULASE NEGATIVE	53	53	22
7	YEAST	44	44	0
8	SERRATIA MARCESCENS	31	31	7
9	ENTEROCOCCUS SP.	27	27	13

Gold Layer Summary

In [4]:

```
# Complete Gold layer summary
gold_tables = {
    'Dimensions': ['dim_patient', 'dim_time', 'dim_careunit', 'dim_item', 'dim_
                   'dim_service', 'dim_procedure_icd', 'dim_caregiver'],
    'Facts': ['fact_admission', 'fact_icu_stay', 'fact_lab_event', 'fact_prescr
              'fact_transfer', 'fact_input_event', 'fact_output_event',
              'fact_procedure', 'fact_microbiology'],
    'Aggregates': ['agg_patient_summary', 'agg_daily_census', 'agg_icu_performa
                   'agg_lab_summary', 'agg_medication_usage', 'agg_infection_st
}

summary = []
for category, tables in gold_tables.items():
    for table in tables:
        try:
            count = table_count('gold', table)
            summary.append({'category': category, 'table': table, 'count': coun
        except:
            summary.append({'category': category, 'table': table, 'count': 'N/A

pd.DataFrame(summary)
```

```
2026-01-01 13:41:13,604 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:13,605 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.dim_patient
2026-01-01 13:41:13,605 INFO sqlalchemy.engine.Engine [cached since 12.38s ago]
{}
2026-01-01 13:41:13,610 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:13,613 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:13,615 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.dim_time
2026-01-01 13:41:13,616 INFO sqlalchemy.engine.Engine [generated in 0.00113s]
{}
2026-01-01 13:41:13,682 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:13,686 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:13,687 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.dim_careunit
2026-01-01 13:41:13,687 INFO sqlalchemy.engine.Engine [generated in 0.00065s]
{}
2026-01-01 13:41:13,766 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:13,771 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:13,773 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.dim_item
2026-01-01 13:41:13,774 INFO sqlalchemy.engine.Engine [generated in 0.00091s]
{}
2026-01-01 13:41:13,890 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:13,893 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:13,895 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.dim_labitem
2026-01-01 13:41:13,896 INFO sqlalchemy.engine.Engine [generated in 0.00100s]
{}
2026-01-01 13:41:13,914 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:13,918 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:13,920 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.dim_service
2026-01-01 13:41:13,920 INFO sqlalchemy.engine.Engine [generated in 0.00086s]
{}
2026-01-01 13:41:13,928 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:13,933 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:13,934 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.dim_procedure_icd
2026-01-01 13:41:13,936 INFO sqlalchemy.engine.Engine [generated in 0.00118s]
{}
2026-01-01 13:41:13,944 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:13,947 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:13,949 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.dim_caregiver
2026-01-01 13:41:13,950 INFO sqlalchemy.engine.Engine [generated in 0.00092s]
{}
2026-01-01 13:41:13,991 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:13,995 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:13,996 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.fact_admission
2026-01-01 13:41:13,998 INFO sqlalchemy.engine.Engine [generated in 0.00132s]
{}
2026-01-01 13:41:14,012 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:14,015 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,017 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.fact_icu_stay
2026-01-01 13:41:14,018 INFO sqlalchemy.engine.Engine [generated in 0.00102s]
{}
2026-01-01 13:41:14,034 INFO sqlalchemy.engine.Engine COMMIT
```

```
2026-01-01 13:41:14,039 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,040 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.fact_lab_event
2026-01-01 13:41:14,041 INFO sqlalchemy.engine.Engine [generated in 0.00106s]
{}
2026-01-01 13:41:14,233 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:14,240 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,242 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.fact_prescription
2026-01-01 13:41:14,243 INFO sqlalchemy.engine.Engine [generated in 0.00090s]
{}
2026-01-01 13:41:14,289 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:14,293 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,294 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.fact_transfer
2026-01-01 13:41:14,295 INFO sqlalchemy.engine.Engine [generated in 0.00117s]
{}
2026-01-01 13:41:14,315 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:14,319 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,321 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.fact_input_event
2026-01-01 13:41:14,323 INFO sqlalchemy.engine.Engine [generated in 0.00113s]
{}
2026-01-01 13:41:14,379 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:14,382 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,383 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.fact_output_event
2026-01-01 13:41:14,384 INFO sqlalchemy.engine.Engine [generated in 0.00081s]
{}
2026-01-01 13:41:14,426 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:14,429 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,430 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.fact_procedure
2026-01-01 13:41:14,431 INFO sqlalchemy.engine.Engine [generated in 0.00085s]
{}
2026-01-01 13:41:14,449 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:14,454 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,456 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.fact_microbiology
2026-01-01 13:41:14,457 INFO sqlalchemy.engine.Engine [generated in 0.00108s]
{}
2026-01-01 13:41:14,485 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:14,490 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,491 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.agg_patient_summary
2026-01-01 13:41:14,492 INFO sqlalchemy.engine.Engine [generated in 0.00116s]
{}
2026-01-01 13:41:14,498 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:14,501 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,503 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.agg_daily_census
2026-01-01 13:41:14,504 INFO sqlalchemy.engine.Engine [generated in 0.00168s]
{}
2026-01-01 13:41:14,569 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:14,574 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,575 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gol
d.agg_icu_performance
2026-01-01 13:41:14,576 INFO sqlalchemy.engine.Engine [generated in 0.00128s]
{}
2026-01-01 13:41:14,580 INFO sqlalchemy.engine.Engine COMMIT
```

```
2026-01-01 13:41:14,584 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,586 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gold.agg_lab_summary
2026-01-01 13:41:14,588 INFO sqlalchemy.engine.Engine [generated in 0.00191s]
{}
2026-01-01 13:41:14,623 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:14,628 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,630 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gold.agg_medication_usage
2026-01-01 13:41:14,631 INFO sqlalchemy.engine.Engine [generated in 0.00105s]
{}
2026-01-01 13:41:14,639 INFO sqlalchemy.engine.Engine COMMIT
2026-01-01 13:41:14,644 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2026-01-01 13:41:14,646 INFO sqlalchemy.engine.Engine SELECT COUNT(*) FROM gold.agg_infection_stats
2026-01-01 13:41:14,647 INFO sqlalchemy.engine.Engine [generated in 0.00110s]
{}
2026-01-01 13:41:14,651 INFO sqlalchemy.engine.Engine COMMIT
```

Out[4]:

	category	table	count
0	Dimensions	dim_patient	100
1	Dimensions	dim_time	38715
2	Dimensions	dim_careunit	5
3	Dimensions	dim_item	12487
4	Dimensions	dim_labitem	753
5	Dimensions	dim_service	3
6	Dimensions	dim_procedure_icd	64
7	Dimensions	dim_caregiver	7567
8	Facts	fact_admission	129
9	Facts	fact_icu_stay	136
10	Facts	fact_lab_event	228222
11	Facts	fact_prescription	31194
12	Facts	fact_transfer	1572
13	Facts	fact_input_event	48023
14	Facts	fact_output_event	22640
15	Facts	fact_procedure	1506
16	Facts	fact_microbiology	4006
17	Aggregates	agg_patient_summary	100
18	Aggregates	agg_daily_census	129
19	Aggregates	agg_icu_performance	5
20	Aggregates	agg_lab_summary	18530
21	Aggregates	agg_medication_usage	590
22	Aggregates	agg_infection_stats	46