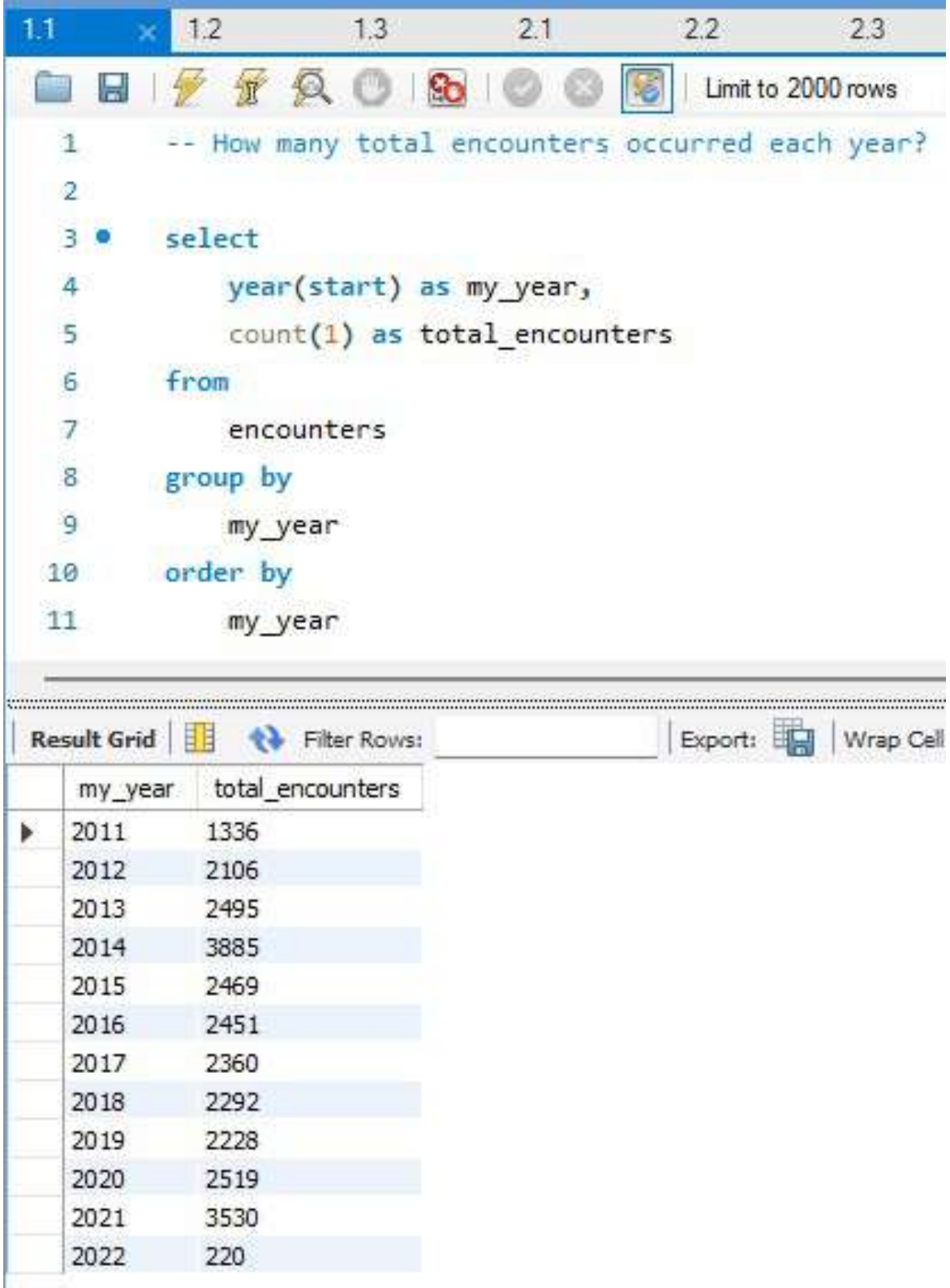

1.1
1.2
1.3
2.1
2.2
2.3
Limit to 2000 rows
1 -- How many total encounters occurred each year?
2
3 select
4 year(start) as my_year,
5 count(1) as total_encounters
6 from
7 encounters
8 group by
9 my_year
10 order by
11 my_year
Result Grid
Filter Rows:
Export:
Wrap Cell
my_year total_encounters
2011 1336
2012 2106
2013 2495
2014 3885
2015 2469
2016 2451
2017 2360
2018 2292
2019 2228
2020 2519
2021 3530
2022 220

1.1 | 1.2 | 1.3 | 2.1 | 2.2 | 2.3 | 2.4 | 3.1 | 3.2 | 3.3

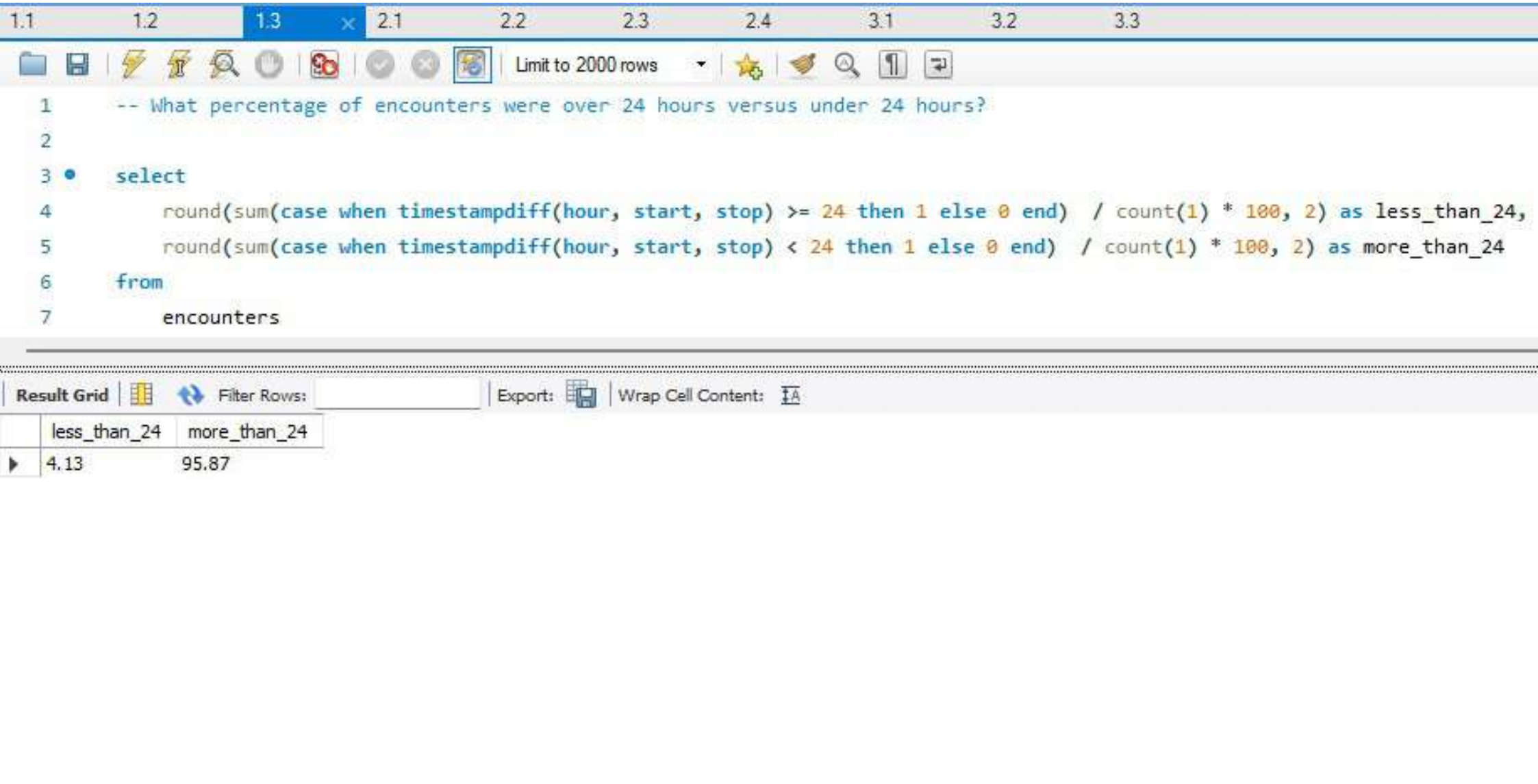
Limit to 2000 rows

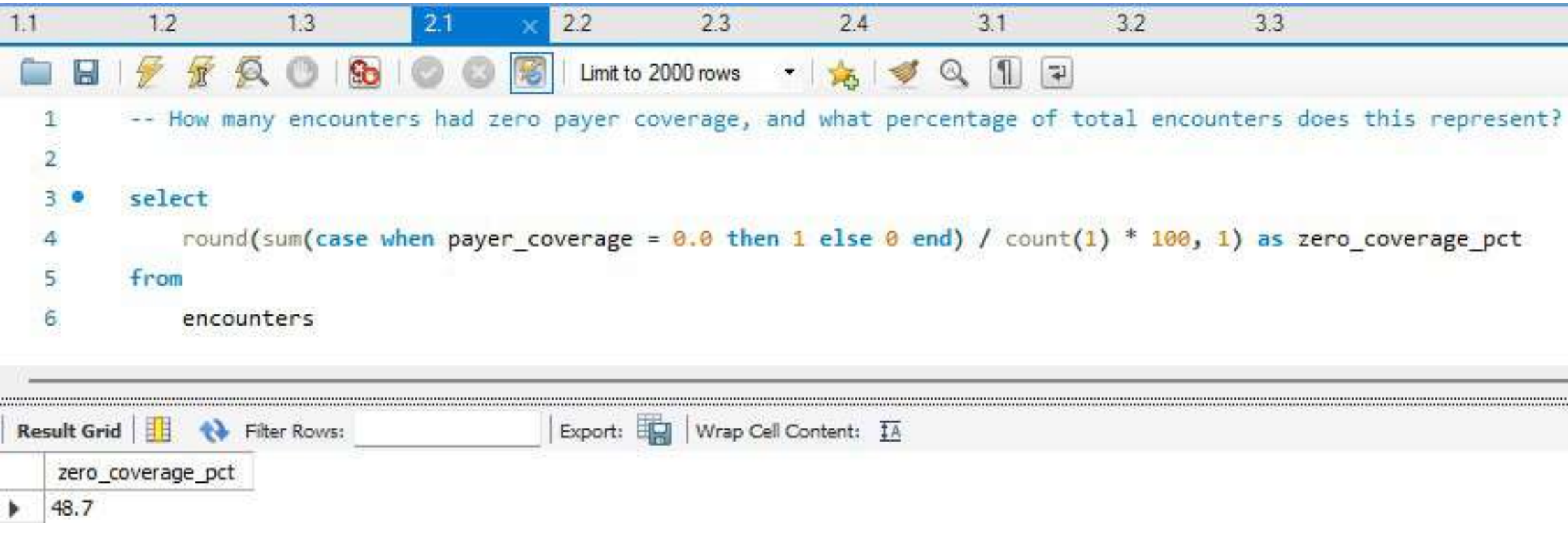
```sql
-- For each year, what percentage of all encounters belonged to each encounter class (ambulatory, outpatient, wellness, urgentcare, emergency, and inpatient)?

select
    year(start) as my_year,
    round((sum(case when encounterclass = "ambulatory" then 1 else 0 end)/count(*)*100), 2) as ambulatory,
    round((sum(case when encounterclass = "outpatient" then 1 else 0 end)/count(*)*100), 2) as outpatient,
    round((sum(case when encounterclass = "wellness" then 1 else 0 end)/count(*)*100), 2) as wellness,
    round((sum(case when encounterclass = "urgentcare" then 1 else 0 end)/count(*)*100), 2) as urgentcare,
    round((sum(case when encounterclass = "emergency" then 1 else 0 end)/count(*)*100), 2) as emergency,
    round((sum(case when encounterclass = "inpatient" then 1 else 0 end)/count(*)*100), 2) as inpatient
from
    encounters
group by
    my_year
order by
    my_year
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

my_year	ambulatory	outpatient	wellness	urgentcare	emergency	inpatient
2011	49.93	24.48	13.02	2.25	4.12	6.21
2012	42.50	21.08	9.12	14.20	8.69	4.42
2013	44.33	19.44	7.45	14.39	9.02	5.37
2014	60.26	17.86	4.86	8.42	5.56	3.04
2015	43.46	20.49	6.93	15.39	9.23	4.50
2016	43.78	19.62	7.43	13.91	10.20	5.06
2017	41.82	20.13	7.16	16.31	9.24	5.34
2018	40.71	20.94	7.59	16.45	10.78	3.53
2019	37.97	20.47	7.54	17.82	10.19	6.01
2020	47.32	19.73	6.31	14.49	9.29	2.86
2021	36.91	40.17	4.39	10.71	6.26	1.56
2022	55.00	16.82	5.45	12.73	8.18	1.82


1.1
1.2
1.3
2.1
2.2
2.3
2.4
3.1
3.2
3.3
Limit to 2000 rows
1 -- What percentage of encounters were over 24 hours versus under 24 hours?
2
3 select
4 round(sum(case when timestampdiff(hour, start, stop) >= 24 then 1 else 0 end) / count(1) * 100, 2) as less_than_24,
5 round(sum(case when timestampdiff(hour, start, stop) < 24 then 1 else 0 end) / count(1) * 100, 2) as more_than_24
6 from
7 encounters
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
less_than_24
more_than_24
4.13
95.87


1.1
1.2
1.3
2.1
2.2
2.3
2.4
3.1
3.2
3.3
Limit to 2000 rows
1 -- How many encounters had zero payer coverage, and what percentage of total encounters does this represent?
2
3 select
4 round(sum(case when payer_coverage = 0.0 then 1 else 0 end) / count(1) * 100, 1) as zero_coverage_pct
5 from
6 encounters
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
zero_coverage_pct
48.7

1.1 1.2 1.3 2.1 2.2 2.3 2.4 3.1 3.2 3.3

Limit to 2000 rows

```sql
-- What are the top 10 most frequent procedures performed and the average base cost for each?

select
    description,
    count(1) as procedure_count,
    round(avg(base_cost), 2) as avg_base_cost
from
    procedures
group by
    description
order by
    procedure_count desc
limit 10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

description	procedure_count	avg_base_cost
Assessment of health and social care needs (procedure)	4596	431.00
Hospice care (regime/therapy)	4098	431.00
Depression screening (procedure)	3614	431.00
Depression screening using Patient Health Questionnaire Two-Item score (procedure)	3614	431.00
Assessment of substance use (procedure)	2906	431.00
Renal dialysis (procedure)	2746	1004.09
Assessment using Morse Fall Scale (procedure)	2422	431.00
Assessment of anxiety (procedure)	2288	431.00
Medication Reconciliation (procedure)	2284	509.12
Screening for drug abuse (procedure)	1484	431.00

1.1 1.2 1.3 2.1 2.2 2.3 2.4 3.1 3.2 3.3

Limit to 2000 rows

```sql
-- What are the top 10 procedures with the highest average base cost and the number of times they were performed?

select
    description,
    round(avg(base_cost), 2) as avg_base_cost,
    count(1) as procedure_count
from
    procedures
group by
    description
order by
    avg_base_cost desc
limit 10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

description	avg_base_cost	procedure_count
Admit to ICU (procedure)	206260.40	5
Coronary artery bypass grafting	47085.89	9
Lumpectomy of breast (procedure)	29353.00	5
Hemodialysis (procedure)	29299.56	27
Insertion of biventricular implantable cardioverter defibrillator	27201.00	4
Electrical cardioversion	25903.11	1383
Partial resection of colon	25229.29	7
Fine needle aspiration biopsy of lung (procedure)	23141.00	1
Percutaneous mechanical thrombectomy of portal vein using fluoroscopic guidance	20228.04	57
Percutaneous coronary intervention	19728.00	9

1.1 1.2 1.3 2.1 2.2 2.3 2.4 3.1 3.2 3.3

Limit to 2000 rows

```sql
-- What is the average total claim cost for encounters, broken down by payer?

select
    p.name,
    round(avg(e.total_claim_cost),2) as avg_total_claim_cost
from
    encounters e
left join
    payers p
    on e.payer = p.id
group by
    p.name
order by
    avg_total_claim_cost desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	avg_total_claim_cost
Medicaid	6205.22
NO_INSURANCE	5593.20
Anthem	4236.81
Humana	3269.30
Blue Cross Blue Shield	3245.58
Cigna Health	2996.95
UnitedHealthcare	2848.34
Aetna	2767.05
Medicare	2167.55
Dual Eligible	1696.19

1.1 1.2 1.3 2.1 2.2 2.3 2.4 3.1 3.2 3.3

Limit to 2000 rows

```sql
-- How many unique patients were admitted each quarter over time?

select
    concat(cast(year(start) as char), "-", cast(quarter(start) as char)) as year_qtr,
    count(distinct(patient)) as unique_patients
from
    encounters
group by
    concat(cast(year(start) as char), "-", cast(quarter(start) as char))
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year_qtr	unique_patients
2011-1	156
2011-2	162
2011-3	155
2011-4	168
2012-1	249
2012-2	261
2012-3	232
2012-4	240
2013-1	243
2013-2	271
2013-3	259
2013-4	242
2014-1	394
2014-2	315
2014-3	272
2014-4	260
2015-1	244
2015-2	259
2015-3	252
2015-4	244
2016-1	237
2016-2	245
2016-3	241
2016-4	258
2017-1	235
2017-2	256
2017-3	234
2017-4	243
2018-1	250
2018-2	256
2018-3	228

3.2* | 3.3

Limit to 2000 rows

```sql
-- 3.2  How many patients were re-admitted within 30 days of a previous encounter?

with cte as (
    select
        patient,
        start,
        stop,
        lead(start) over(partition by patient order by start asc) as next_admission
    from
        encounters
)

select
    count(distinct(patient)) as readmissions
from
    cte
where
    datediff(next_admission, stop) < 30
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

readmissions
771

3.2* 3.3

Limit to 2000 rows

```sql
-- 3.3  Which patients had the most readmissions?

with cte as (
    select
        patient,
        start,
        stop,
        lead(start) over(partition by patient order by start asc) as next_admission
    from
        encounters
)

select
    patient,
    count(*) as readmission_count
from
    cte
where
    datediff(next_admission, stop) < 30
group by
    patient
order by
    readmission_count desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

patient	readmission_count
1712d26d-822d-1e3a-2267-0a9dba31d7c8	1376
3de74169-7f67-9304-91d4-757e0f3a14d2	876
5e055638-0dad-dfd5-005d-1e74b6fd29ac	871
3f523789-55f3-bb31-2757-4803ca6a9c2a	442
5dcb295d-92df-a147-ebcc-aa49b6262830	421
442dc617-c7f2-0513-15cd-c35c4efdba73	384
b4ab9ab3-f52a-751e-a990-3bd5654d5870	376
9b8ae606-5059-b3a6-19c7-c812901898bb	347
b4671e80-7e87-9260-ca31-6a9397d465d1	291
ff331e5c-ab16-e218-f39a-63e11de1ed75	280
381e16cd-86c8-d369-cd76-2dd9e0b5faf5	262
dabb67a9-bc4b-bc40-696e-65546a548d39	228
a31f5105-476f-b129-fb9a-514cf45a34ac	218
b17ffd11-c8c2-28aa-0c26-8ddc6bc5957d	207
df607b64-3d5e-5795-1646-edd2ccf55192	194
de60a9e4-1334-6078-c1f1-e5083bb20361	186
6443e813-1b9b-620c-f7de-af77f9d02733	175
c06513da-7f35-b4eb-5bab-28c87ff97a10	173