EXPERIMENT 3

AIM:

To perform various OLAP queries such as : slice, pivot, roll up, drill down for your dataset.

THEORY:

Online Analytical Processing (OLAP) is an approach for answering multi-dimensional analytical needs of organisations. It a mechanism to get analytical data (such as aggregations, statistical data, views at different levels, etc.) for a warehouse, perhaps consisting of multiple database systems

Basic data operations and analysis can be performed using a spreadsheet. Here, data values are arranged in a row and column format. This is ideal for two-dimensional data. However, OLAP often contains multidimensional data. Even this data is usually obtained from a different and unrelated source. Special systems can be used to store and analyze multidimensional data in a logical and orderly manner.

Operations performed in OLAP could involve grouping, aggregating and joining data. While suported in Relation al Database Management Systems, they should preferably be implemented in a separate saystem, becuase DBs are optimized for OLTP. With OLAP data is usually pre-calculated and pre-aggregated, making analysis faster.

One way to implement OLAP is to use RDBMS. This is called ROLAP (relational OLAP). Here, GROUP BY, JOIN, and aggregate functions are used to implement various OLAP queries.

Another way is to use Data Cubes. These are special structures that represent data along various axes at different levels of detail. They are useful structures to study OLAP theoretically. However, in practical implementation, the main issue comes in the number of data cubes, which increases super-exponentially.

A hybrid approach utilizes a mixture of both the above systems.

There are five types of operations or queries in OLAP, which are:

- Roll-up
- Drill-down
- Slice
- Dice
- Pivot

ROLL-UP:

Roll-up is an operation to reduce the level of detail. It is also known as consolidation or aggregation. The Roll-up operation can be performed in 2 ways:

- Reducing dimensions (by grouping over all values)
- Climbing up concept hierarchy. This involed internal hierarchies, like variuos levels of time (hour, day, week)

DRILL DOWN

In drill-down, data is analysed at higher level of detail. It is the opposite of the rollup process. It can be done by

- Moving down the concept hierarchy (opposit to roll up)
- Using additional dimensions

SLICE AND DICE

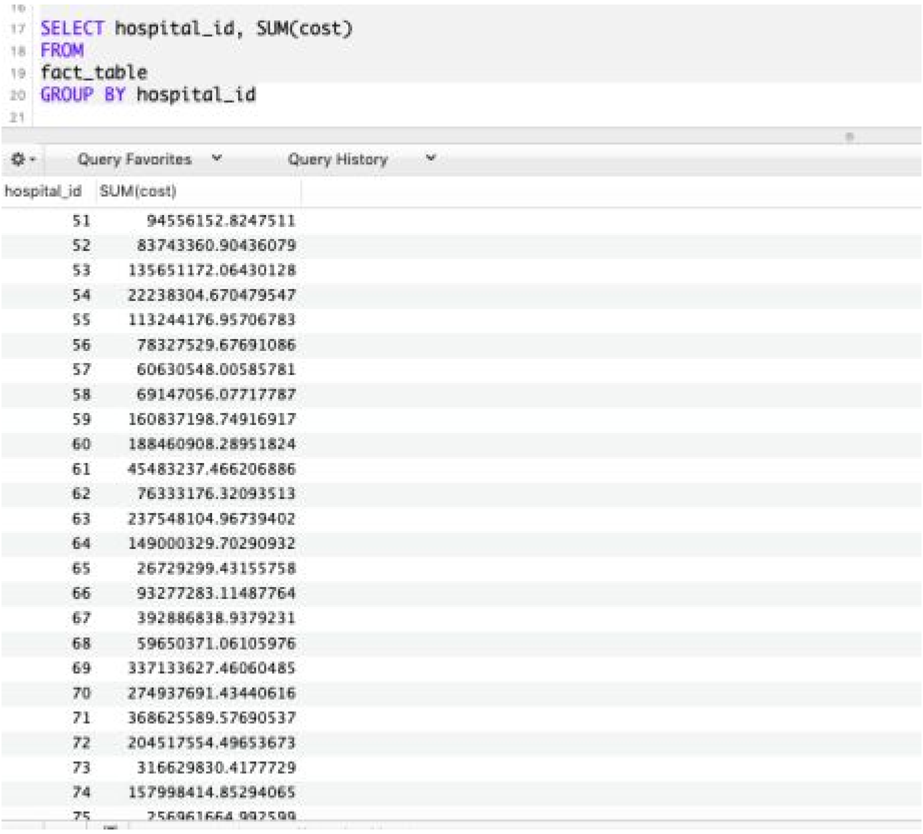
Sub cubes are created by choosing fixed values in a dimension. In slice, only one dimension is affected. In dice, multiple dimensions are affected

PIVOT

In Pivot, you rotate the data axes to provide a substitute presentation of data.

PRACTICAL:

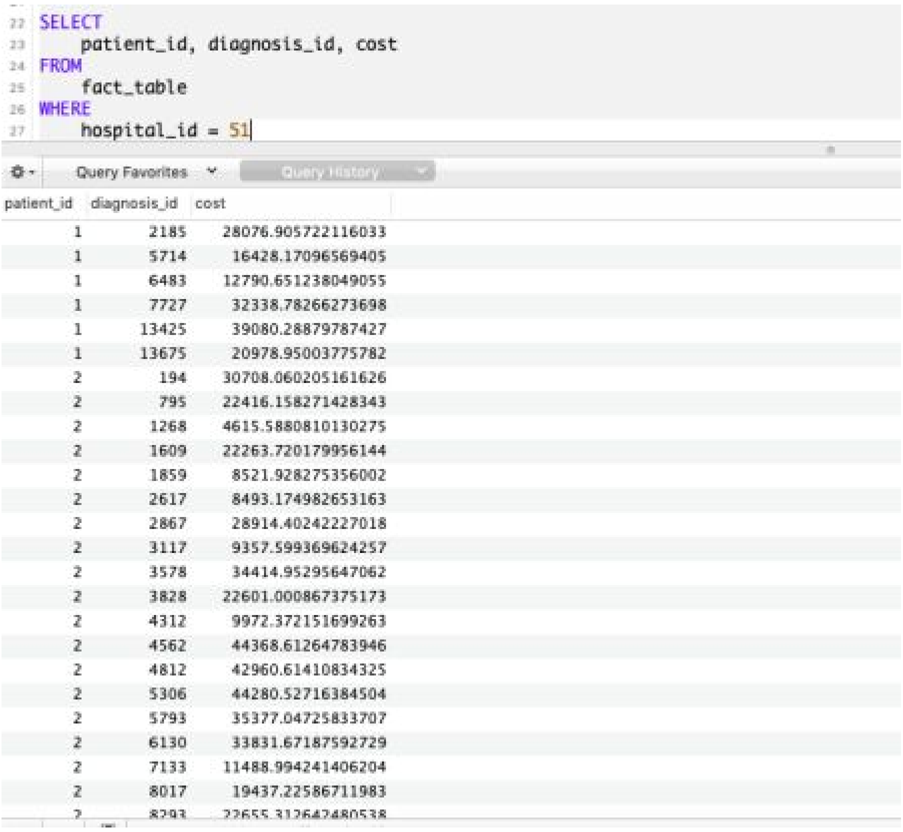
ROLL UP



```sql
SELECT hospital_id, SUM(cost)
FROM
fact_table
GROUP BY hospital_id
```

hospital_id	SUM(cost)
51	94556152.8247511
52	83743360.90436079
53	135651172.06430128
54	22238304.670479547
55	113244176.95706783
56	78327529.67691086
57	60630548.00585781
58	69147056.07717787
59	160837198.74916917
60	188460908.28951824
61	45483237.466206886
62	76333176.32093513
63	237548104.96739402
64	149000329.70290932
65	26729299.43155758
66	93277283.11487764
67	392886838.9379231
68	59650371.06105976
69	337133627.46060485
70	274937691.43440616
71	368625589.57690537
72	204517554.49653673
73	316629830.4177729
74	157998414.85294065
75	256961664.992599

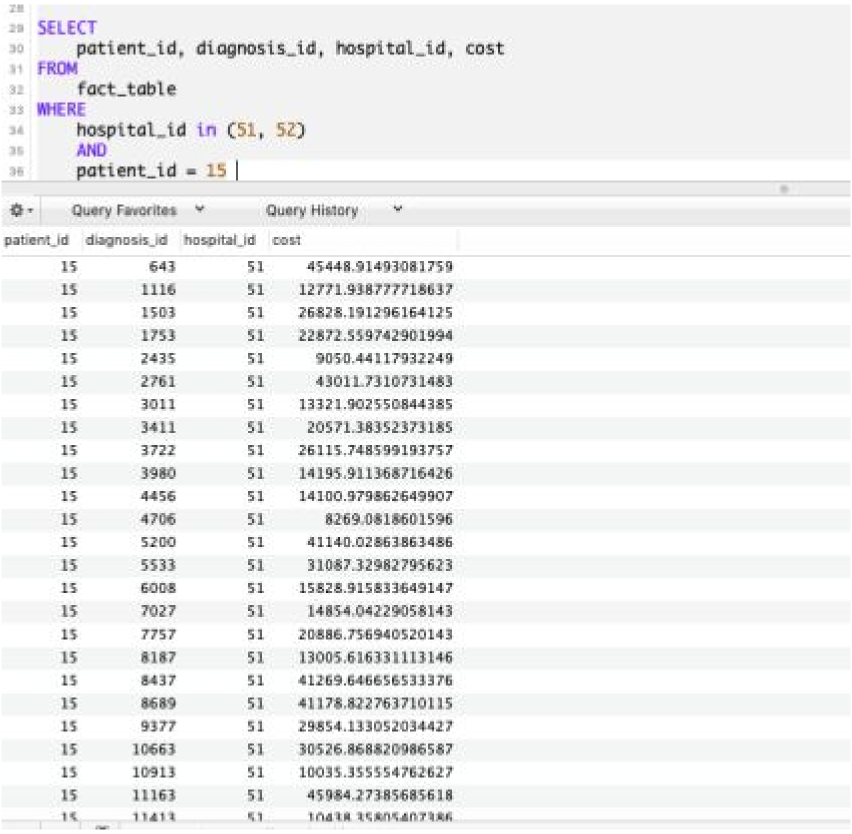
SLICE



```sql
SELECT
    patient_id, diagnosis_id, cost
FROM
    fact_table
WHERE
    hospital_id = 51
```

patient_id	diagnosis_id	cost
1	2185	28076.905722116033
1	5714	16428.17096569405
1	6483	12790.651238049055
1	7727	32338.78266273698
1	13425	39080.28879787427
1	13675	20978.95003775782
2	194	30708.060205161626
2	795	22416.158271428343
2	1268	4615.5880810130275
2	1609	22263.720179956144
2	1859	8521.928275356002
2	2617	8493.174982653163
2	2867	28914.40242227018
2	3117	9357.599369624257
2	3578	34414.95295647062
2	3828	22601.000867375173
2	4312	9972.372151699263
2	4562	44368.61264783946
2	4812	42960.61410834325
2	5306	44280.52716384504
2	5793	35377.04725833707
2	6130	33831.67187592729
2	7133	11488.994241406204
2	8017	19437.22586711983
2	8293	22655.312642480538

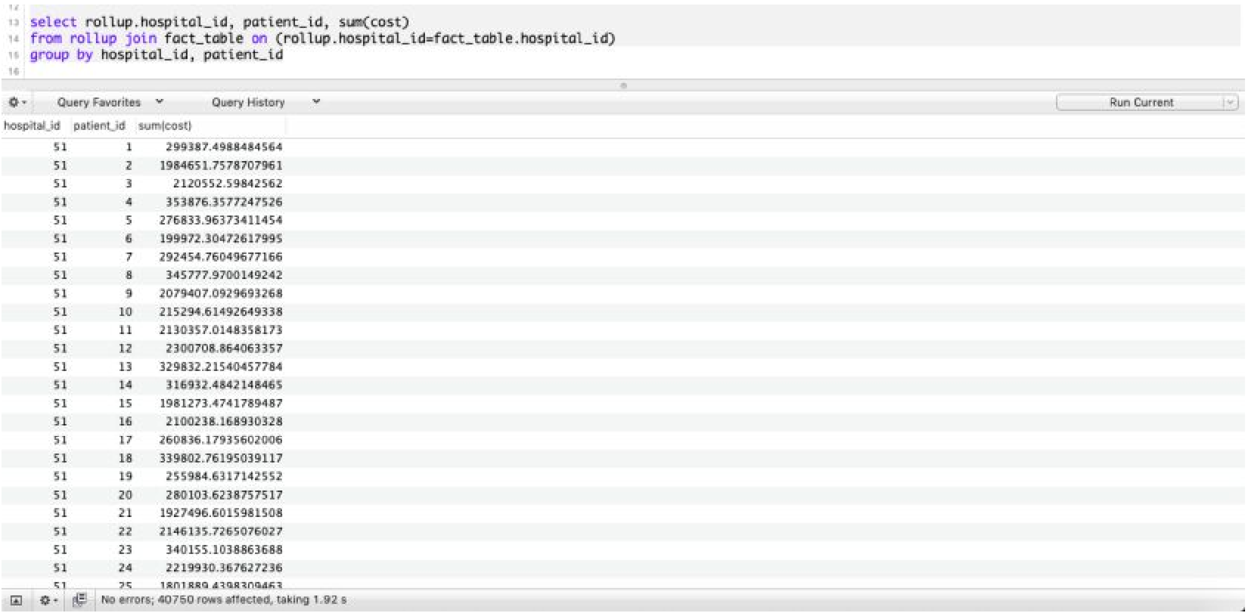
DICE



```sql
SELECT
    patient_id, diagnosis_id, hospital_id, cost
FROM
    fact_table
WHERE
    hospital_id in (51, 52)
    AND
    patient_id = 15
```

patient_id	diagnosis_id	hospital_id	cost
15	643	51	45448.91493081759
15	1116	51	12771.938777718637
15	1503	51	26828.191296164125
15	1753	51	22872.559742901994
15	2435	51	9050.44117932249
15	2761	51	43011.7310731483
15	3011	51	13321.902550844385
15	3411	51	20571.38352373185
15	3722	51	26115.748599193757
15	3980	51	14195.911368716426
15	4456	51	14100.979862649907
15	4706	51	8269.0818601596
15	5200	51	41140.02863863486
15	5533	51	31087.32982795623
15	6008	51	15828.915833649147
15	7027	51	14854.04229058143
15	7757	51	20886.756940520143
15	8187	51	13005.616331113146
15	8437	51	41269.646656533376
15	8689	51	41178.822763710115
15	9377	51	29854.133052034427
15	10663	51	30526.868820986587
15	10913	51	10035.355554762627
15	11163	51	45984.27385685618
15	11413	51	10438.35805407386

DRILL DOWN

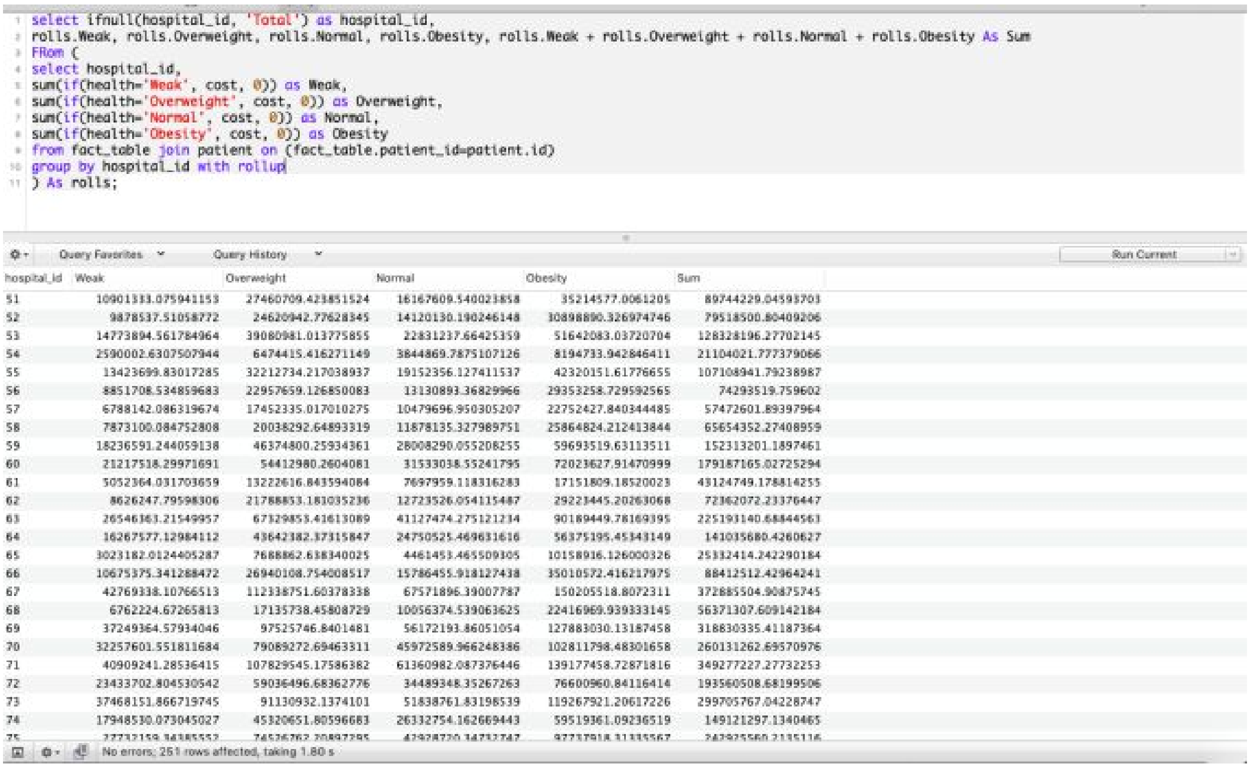


```sql
select rollup.hospital_id, patient_id, sum(cost)
from rollup join fact_table on (rollup.hospital_id=fact_table.hospital_id)
group by hospital_id, patient_id
```

hospital_id	patient_id	sum(cost)
51	1	299387.4988484564
51	2	1984651.7578707961
51	3	2120552.59842562
51	4	353876.3577247526
51	5	276833.96373411454
51	6	199972.30472617995
51	7	292454.76049677166
51	8	345777.9700149242
51	9	2079407.0929693268
51	10	215294.61492649338
51	11	2130357.0148358173
51	12	2300708.864063357
51	13	329832.21540457784
51	14	316932.4842148465
51	15	1981273.4741789487
51	16	2100238.168930328
51	17	260836.17935602006
51	18	339802.76195039117
51	19	255984.6317142552
51	20	280103.6238757517
51	21	1927496.6015981508
51	22	2146135.7265076027
51	23	340155.1038863688
51	24	2219930.367627236
51	25	1801889.4398309463

No errors; 40750 rows affected, taking 1.92 s

PIVOT



```sql
select ifnull(hospital_id, 'Total') as hospital_id,
rolls.Weak, rolls.Overweight, rolls.Normal, rolls.Obesity, rolls.Weak + rolls.Overweight + rolls.Normal + rolls.Obesity As Sum
FRom (
select hospital_id,
sum(if(health='Weak', cost, 0)) as Weak,
sum(if(health='Overweight', cost, 0)) as Overweight,
sum(if(health='Normal', cost, 0)) as Normal,
sum(if(health='Obesity', cost, 0)) as Obesity
from fact_table join patient on (fact_table.patient_id=patient.id)
group by hospital_id with rollup
) As rolls;
```

hospital_id	Weak	Overweight	Normal	Obesity	Sum
51	10901333.075941153	27460709.423851524	16167609.540023858	35214577.0061205	89744229.04593703
52	9878537.51058772	24620942.77628345	14120130.190246148	30898890.326974746	79518500.80409206
53	14773894.561784964	39080981.013775855	22831237.66425359	51642083.03720704	128328196.27702145
54	2590002.6307507944	6474415.416271149	3844869.7875107126	8194733.942846411	21104021.777379066
55	13423699.83017285	32212734.217038937	19152356.127411537	42320151.61776655	107108941.79238987
56	8851708.534859683	22957659.126850083	13130893.36829966	29353258.729592565	74293519.759602
57	6788142.086319674	17452335.017010275	10479696.950305207	22752427.840344485	57472601.89397964
58	7873100.084752808	20038292.64893319	11878135.327989751	25864824.212413844	65654352.27408959
59	18236591.244059138	46374800.25934361	28008290.055208255	59693519.63113511	152313201.1897461
60	21217518.29971691	54412980.2604081	31533038.55241795	72023627.91470999	179187165.02725294
61	5052364.031703659	13222616.843594084	7697959.118316283	17151809.18520023	43124749.178814255
62	8626247.79598306	21788853.181035236	12723526.054115487	29223445.20263068	72362072.23376447
63	26546363.21549957	67329853.41613089	41127474.275121234	90189449.78169395	225193140.68844563
64	16267577.12984112	43642382.37315847	24750525.469631616	56375195.45343149	141035680.4260627
65	3023182.0124405287	7688862.638340025	4461453.465509305	10158916.126000326	25332414.242290184
66	10675375.341288472	26940108.754008517	15786455.918127438	35010572.416217975	88412512.42964241
67	42769338.10766513	112338751.60378338	67571896.39007787	150205518.8072311	372885504.90875745
68	6762224.67265813	17135738.45808729	10056374.539063625	22416969.939333145	56371307.609142184
69	37249364.57934046	97525746.8401481	56172193.86051054	127883030.13187458	318830335.41187364
70	32257601.551811684	79089272.69463311	45972589.966248386	102811798.48301658	260131262.69570976
71	40909241.28536415	107829545.17586382	61360982.087376446	139177458.72871816	349277227.27732253
72	23433702.804530542	59036496.68362776	34489348.35267263	76600960.84116414	193560508.68199506
73	37468151.866719745	91130932.1374101	51838761.83198539	119267921.20617226	299705767.04228747
74	17948530.073045027	45320651.80596683	26332754.162669443	59519361.09236519	149121297.1340465
75	27732159.34385552	74526762.20897295	42928720.34732747	97737918.31335567	242925560.2135116

No errors; 251 rows affected, taking 1.80 s

CONCLUSION:

We successfully implemented the OLAP queries on our dataset.