

APOLLO 24|7 PROJE

Name: Shubhang Shukla

LinkedIn: <https://www.linkedin.com/in/shubhangshukla/>

Github: <https://github.com/shubhang239>

Tool Used: SQLite

Problem Statement:

Suggest 5 test packages combining different items that will be best suited for our customers. The objective of creating packages is to increase Revenue. List down the steps and logic you followed to arrive at the 5 packages.

Additional task (Optional): List down the criteria/metrics you will use to judge these packages' future performance.

My approach:

I believe there are multiple ways to approach this problem, we can make packages on the basis of and order on the basis of max revenue.

1. Popularity of the tests - (Top 5 Most Ordered Tests)
2. The combination of tests ordered together frequently - (Frequently Bought Tests)
3. Tests based on the Sample Type - (One Sample, All Checkups)
4. On basis of Time Taken - (The 12 Hr package, The 24 Hr package and so on)
5. For Loyal Customers
 - Since not all the columns are important I made a View of the database containing the columns I had selected as relevant for analysis. For this analysis I only selected those rows where orderStatus = Order Completed.
 - To avoid joining the 2 tables repeatedly I performed an Outer Left Join to get the itemCode as well the itemName together in one table only.

Creating the View

New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database

Database Structure
Browse Data
Edit Pragmas
Execute SQL

SQL 1

```

1 create view TableView as
2 select O.displayId, O.city, state, O.totalPrice, O.LoyalUser,O.itemCode, O.sampleTypeName, I.itemName, I.PromisedreportTAThrs
3 from OrderDetails as O
4 left outer join ItemDetails as I
5 on O.itemCode=I.itemCode
6 where O.orderStatus='ORDER_COMPLETED';
7
8
9 select * from TableView

```

	displayId	city	state	totalPrice	LoyalUser	itemCode	sampleTypeName	itemName	PromisedreportTAThrs
1	11643	North Delhi	DELHI	500	FALSE	1273	NULL	COVID 19 RT PCR WITH HOME ...	12
2	11315	Ahmedabad	GUJARAT	550	FALSE	1273	NULL	COVID 19 RT PCR WITH HOME ...	12
3	11533	Kolkata	WEST BENGAL	2415	TRUE	74	SERUM	PHOSPHORUS, INORGANIC - SERUM	12
4	11533	Kolkata	WEST BENGAL	2415	TRUE	74	SERUM	URIC ACID - SERUM	24
5	11533	Kolkata	WEST BENGAL	2415	TRUE	75	SERUM	CALCIUM, SERUM	24
6	11533	Kolkata	WEST BENGAL	2415	TRUE	73	SERUM	CREATININE, SERUM	48
7	11533	Kolkata	WEST BENGAL	2415	TRUE	64	WHOLE BLOOD-EDTA	HEMOGLOBIN	12
8	11533	Kolkata	WEST BENGAL	2415	TRUE	64	WHOLE BLOOD-EDTA	PERIPHERAL SMEAR FOR MALARIAL...	48
9	11533	Kolkata	WEST BENGAL	2415	TRUE	85	SERUM	POTASSIUM - SERUM / PLASMA	12
10	11533	Kolkata	WEST BENGAL	2415	TRUE	85	SERUM	SODIUM - SERUM	48
11	11533	Kolkata	WEST BENGAL	2415	TRUE	296	SERUM	PARATHYROID HORMONE [PTH]	48
12	11533	Kolkata	WEST BENGAL	2415	TRUE	74	SERUM	PHOSPHORUS, INORGANIC - SERUM	12
13	11533	Kolkata	WEST BENGAL	2415	TRUE	74	SERUM	URIC ACID - SERUM	24

Execution finished without errors.
Result: 13670 rows returned in 295ms
At line 18:
select * from TableView

Finding out the most popular tests and revenue generated

```

select itemCode, itemName, count(itemCode) as no_of_test, sum(totalPrice) as Total_Revenue
from TableView
group by itemCode
order by Total_Revenue desc
limit 10

```

	itemCode	itemName	no_of_test	Total_Revenue
1	484	COMPLETE BLOOD COUNT (CBC)	1800	3076293.4
2	1273	COVID 19 RT PCR WITH HOME ...	2782	2008532
3	493	LIPID PROFILE	1172	1972240.0
4	54	GLUCOSE, FASTING	838	834057.4
5	126	AMYLASE - SERUM	356	815147.0
6	497	LIVER FUNCTION TEST (LFT)	291	671698.2
7	204	DEHYDROEPIANDROSTERONE ...	178	649815.0
8	127	HBA1C, GLYCATED HEMOGLOBIN	281	558450.2
9	1782	VITAMIN PACKAGE	170	470369.4
10	63	BLOOD GROUP ABO AND RH FACTOR	216	460547.8

Execution finished without errors.
Result: 10 rows returned in 105ms
At line 11:
select itemCode, itemName, count(itemCode) as no_of_test, sum(totalPrice) as Total_Revenue
from TableView
group by itemCode
order by Total_Revenue desc
limit 10

The top tests such as Covid 19 RT_PCR, CBC, Hemogram, Thyroid Profile and Lipid Profile are driving maximum revenue. Hence we can make a package for the Top 5 most commonly ordered tests.

Finding out the tests which are most commonly ordered together:

```
select displayId, group_concat(itemName, ',') as test_bundle, count(itemCode) as no_of_test, sum(totalPrice) as Total_Amount
from TableView
group by displayId
order by Total_Amount DESC
```

	displayId	test_bundle	no_of_test	Total_Amount
1	11473	LIPID PROFILE,THYROID PROFILE ...	38	315530
2	11566	LIPID PROFILE,THYROID PROFILE ...	46	246709
3	11063	COVID 19 RT PCR WITH HOME ...	24	210485
4	11304	COVID 19 RT PCR WITH HOME ...	47	171051
5	11050	LIPID PROFILE,THYROID PROFILE ...	20	170882
6	11125	POTASSIUM - SERUM / ...	28	169381
7	11668	GLUCOSE, FASTING,GLUCOSE, POST ...	28	168523
8	11382	COVID 19 RT PCR WITH HOME ...	22	166400
9	11596	Total Thyroid Profile with Antithyroid ...	46	159780
10	11663	COVID 19 RT PCR WITH HOME ...	36	143019
11	11430	COVID 19 RT PCR WITH HOME ...	22	136308
12	11534	VITAMIN PACKAGE,COMPLETE BLOO...	21	135241
13	10851	ALANINE AMINOTRANSFERASE (AL...	36	133870
14	11358	COMPLETE BLOOD COUNT ...	37	132752
15	11224	RENAL PROFILE/RENAL FUNCTION ...	38	132066
16	11438	FERRITIN,MALARIAL ANTIGEN (...	29	122522.4
17	11051	HRSAG / AUSTRALIA	34	116160

```
Execution finished without errors.
Result: 1047 rows returned in 180ms
At line 27:
select displayId, group_concat(itemName, ',') as test_bundle, count(itemCode) as no_of_test, sum(totalPrice) as Total_Amount
from TableView
group by displayId
order by Total_Amount DESC
```

So from the results it's clearly evident the maximum amount spent by the top customer and which are the tests they bought together. Hence we can analyze the test bundle and see which are the tests which users commonly buy and make our packages including those tests.

We can group tests by sample type because by collecting just one sample from the user a large number of tests can be performed easily. This is much more convenient for the customers since they don't have to give blood or urine samples every time for every different test. Therefore it reduces the effort and makes it a better experience.

```
select sampleTypeName, group_concat(itemName, ',') as bundle_of_tests, count(itemCode) as no_of_test, sum(DISTINCT totalPrice) as Revenue_by_sample
from TableView
group by sampleTypeName
order by Revenue_by_sample desc
```

	sampleTypeName	bundle_of_tests	no_of_test	Revenue_by_sample
1	SERUM	PHOSPHORUS, INORGANIC - ...	4633	2102947.5
2	NULL	COVID 19 RT PCR WITH HOME ...	4765	1711638.7
3	WHOLE BLOOD-EDTA	HEMOGLOBIN,PERIPHERAL SMEAR ...	2579	1576347.5
4	URINE	COMPLETE URINE ...	490	489207.2
5	NAF PLASMA(F)	GLUCOSE, FASTING,GLUCOSE, POST ...	620	306073.4
6	NAF PLASMA(R)	GLUCOSE, RANDOM,GLUCOSE, ...	129	162551.6
7	NA CITRATE	D-DIMER,D-DIMER,D-DIMER,D-...	48	138081.0
8	WHOLE BLOOD- NA CITRATE	PROTHROMBIN TIME (PT/...	90	117804.9
9	SERUM (F)	IGE (TOTAL),INSULIN (FASTING),IG...	41	90837
10	NAF PLASMA(PP)	GLUCOSE, FASTING,GLUCOSE, POST ...	218	52217.3
11	LITHIUM HEPARIN (2 TUBES)	TB GOLD (QUANTIFERON) (GAMMA ...	14	45996
12	NAF PLASMA(F) AND SERUM(F)	CORTISOL, SERUM (RANDOM),HOM...	20	21733
13	SERUM (PP)	C - PEPTIDE (PP),CERULOPLASMIN - ...	6	17307
14	WHOLE BLOOD BACTEC BOTTLE ...	CULTURE AND SENSITIVITY - ...	5	8861
15	ASCITIC/PERITONEAL FLUID	X-PERT MTB / RIF	1	4986
16	BONE MARROW- NA HEPARIN	CD20 (PAN B CELL MARKER),CD19 ...	2	4300

Execution finished without errors.
Result: 20 rows returned in 123ms
At line 22:
select sampleTypeName, group_concat(itemName,',') as bundle_of_tests, count(itemCode) as no_of_test, sum(DISTINCT totalPrice) as Revenue_by_sample
from TableView
group by sampleTypeName
order by Revenue_by_sample desc

We can make packages on tests based on sample type Serum, Whole Blood ETA, Urine etc as these are quite popular and they generate the maximum revenue as shown above.

On the basis of time taken, here i have selected the top revenue generating tests that belong to the 12 Hr package

```
select distinct itemName, sum(totalPrice) over(partition by itemName) as Revenue_Generated
from TableView
where PromisedreportTAThrs = 12
order by Revenue_Generated DESC
```

	itemName	Revenue_Generated
1	COVID 19 RT PCR WITH HOME ...	2008532
2	COMPLETE BLOOD COUNT (CBC)	1538146.7
3	HEMOGRAM	1538146.7
4	C-REACTIVE PROTEIN CRP ...	407573.5
5	RENAL PROFILE/RENAL FUNCTION ...	391588.5
6	DIABETES AND LIPID PROFILE	312095.1
7	POTASSIUM - SERUM / PLASMA	219598.8
8	CULTURE AND SENSITIVITY URINE	200120
9	PHOSPHORUS, INORGANIC - SERUM	187107.0
10	THYROXINE (T4, TOTAL), THYROID ...	155043.3
11	PROTHROMBIN TIME (PT/INR)	121990.9
12	IGE (TOTAL)	117929
13	STDs PACKAGE	115037
14	HEMOGLOBIN	97528.5
15	COMPREHENSIVE PACKAGE - SENIO...	75879
16	ESTRADIOL (E2)	58289.9
17	FOLIC ACID SERUM	49616

Execution finished without errors.
Result: 97 rows returned in 189ms
At line 33:
select distinct itemName, sum(totalPrice) over(partition by itemName) as Revenue_Generated
from TableView
where PromisedreportTAThrs = 12
order by Revenue_Generated DESC

PERFORMANCE METRICS

The future performance of these packages can be judged by:

1. No of Order Cancellations of the packages created.
2. The number of Successfully Completed Orders and the Amount or Revenue corresponding to them.
3. The revenue can be compared to pre-package revenues for the same States/Cities and Dates to judge how much was the increase.