ADVANDB MCO1: Query Optimization

Ryan Austin Fernandez

CS-ST Student  
22A, E. Padua Street  
BF Resort Village, Las Pinas City  
(+63)917-496-5071

ryanaustinf@yahoo.com

Marc Dominic San Pedro

Level 63 Wizard  
69A, Third Level  
Minas Tirith, Gondor  
(+63)977-817-5699

fuckme@outlook.com

Jonah E. Syfu

Tagadurog ng Pangarap Sa Weiss  
2419, Something Building  
Taft Avenue, Fuyuki City  
(+63)915-620-3788

Lol\_ewan@fuckdauniverse.shit

# INTRODUCTION

The group has built a simple application that interfaces with the CBMS database system of Palawan to extract certain data including the average number of overseas Filipino workers per nuclear family, the number of children above a given nutritional index in each geographical division, the average age of death per sex in each geographical location, the amount of fish of a particular type caught in each geographical division, crop densities, total aquatic animal catches per aquatic equipment used, and counts of common beneficiaries of particular classifications of Philhealth beneficiaries.

# ORIGINAL QUERIES

**2.1. Average OFW’s Per Nuclear Family**

This query is to determine the average number of OFW’s per nuclear family for each household that holds a nuclear family, divided by geographical location. The exact query is as follows.

SELECT mun, zone, brgy, purok, SUM(nnucfam) AS `Nuclear Families`, SUM(nofw) AS OFWs, SUM(nofw) / SUM(nnucfam) AS `Average OFW's per Nuclear Family`

FROM db\_hpq.hpq\_hh

WHERE nnucfam > 0

GROUP BY mun, zone, brgy, purok

HAVING SUM(nofw) > :count

The expected output is a series of geographical locations and their corresponding nuclear family counts, OFW counts, and the ratio between OFW’s and nuclear families. The query returned five hundred and fifty-seven rows and seven columns for :count = 0. Since the query does not have conditions that are too restricting, any general value given to :count resulted in the query running for an average of 129.9453 seconds.

**2.2. Number of Children Above a Particular Nutritional Index**

This query is to determine the count of children above a certain nutritional index in each geographical division. The query is as follows.

SELECT country\_resid, prov\_resid\_code, mnutind, COUNT(mnutind) nutCount

FROM hpq\_mem

WHERE mnutind <= :minimum\_index

GROUP BY country\_resid, prov\_resid\_code,mnutind

HAVING nutCount > :count

The expected output is a series of geographical locations and their corresponding number of kids above the given index. A <= operator is used since the hpq\_mem table uses 1 for the healthiest index and 4 for the lowest index. The query returned seventy-three rows and four columns for :minimum\_index = 4 and :count = 0. For the worst nutritional index, the query ran for 13.375 seconds. For the midlevel index, the query ran for 14.461 seconds. For the best nutritional index, the query ran for 12.359 seconds.

**2.3. Average Age of Death Divided by Sex and Geographical Location**

This query is to determine the average age of death per sex per geographical location. The exact query is as follows.

SELECT H.mun,H.zone,H.brgy, mdeadsx, AVG(mdeadage) avg\_death\_age

FROM hpq\_hh H, hpq\_death D

WHERE H.id = D.hpq\_hh\_id AND mdeady = :reason

GROUP BY H.mun,H.zone,H.brgy,mdeadsx

HAVING AVG(mdeadage) > :count

The expected output is a series of geographical locations, the two sexes, and their corresponding average death age. The query returned three hundred and eighty-six rows and five columns for :count = 0 and no specified :reason. For the most common cause of death “other”, the query ran for 5.625 seconds. For the median cause of death, “diabetes”, the query ran for 0.469 seconds. For the least common cause of death, “measles”, the query ran for 0.47 seconds.

**2.4. Amount of Fish Per Type Caught**

This query is to determine the total count of fish caught in given geographical locations. The exact query is as follows.

SELECT H.mun,H.zone,H.brgy, COUNT(H.id) fishcount

FROM hpq\_hh H, hpq\_aquani A

WHERE H.id = A.hpq\_hh\_id AND aquanitype = :type

GROUP BY H.mun,H.zone,H.brgy

HAVING COUNT(H.id) > :count

The expected output is a series of geographical locations and their corresponding counts for number of the given type of aquatic animal caught. The query returned three hundred and seventy-seven rows and four columns for :count = 0 and not :type specified. For the most common type of fish, “other”, the query ran for an average of 75.7475 seconds. For the median type of fish, “milkfish”, the query ran for 1.922 seconds. For the least common type of fish, “tilapia”, the query ran for 0.859 seconds.

**2.5. Crop Densities**

This query is to determine the total crop volume, land area, and corresponding crop densities (volume/land area unit) per geographical division The exact query is as follows.

SELECT H.mun,H.zone,H.brgy, SUM(crop\_vol) AS totalcrop, SUM(alp\_area) AS totalArea, SUM(crop\_vol)/SUM(alp\_area) AS cropDensity

FROM hpq\_hh H, hpq\_alp A, hpq\_crop C

WHERE H.id = A.hpq\_hh\_id AND H.id = C.hpq\_hh\_id AND croptype = :croptype

GROUP BY H.mun,H.zone,H.brgy

HAVING cropDensity > 0

The expected output is a series of geographical locations and their corresponding crop volume, land areas, and the ratio between crop volume and land area referred to as crop density. The query returned three hundred and ninety-four rows and six columns for :count = 0 and no specified :croptype. <performance>

**2.6. Amount of Aquatic Animals Caught Per Type of Aquatic Equipment**

This query is to determine the number of aquatic animals caught per type of aquatic equipment used. The exact query is as follows.

SELECT mun, zone, brgy,SUM(aquaequip\_line) AS totalequip, SUM(aquani\_vol) AS totalvol, SUM(aquani\_vol) / SUM(aquaequip\_line) AS CatchPerEquip

FROM hpq\_aquaequip AA, hpq\_aquani AP, hpq\_hh H

WHERE aquaequiptype = :equip AND aquanitype = :animal AND H.id = AA.hpq\_hh\_id AND H.id = AP.hpq\_hh\_id

GROUP BY H.mun,H.zone,H.brgy

HAVING SUM(aquani\_vol) / SUM(aquaequip\_line) > :count

The expected output is a series of geographical locations and their corresponding counts of the specific type of aquatic equipment used, counts of aquatic animals caught, and the ratio of animals caught to number of equipment used. The query returned three hundred and seventy-five rows and six columns for :count = 0 and no specific type of animal or equipment specified. <performance>

**2.7. Number of Common Philhealth Beneficiaries**

This query is to determine the counts per geographical location of citizens who are beneficiaries of Philhealth who are employed, individually paying, sponsored, and lifetime members. The exact query is as follows.

SELECT H.mun,H.zone,H.brgy,COUNT(H.id) benefCount

FROM hpq\_hh H, hpq\_phiheal\_spon\_mem PSM, hpq\_phiheal\_empl\_mem PEM, hpq\_phiheal\_indiv\_mem PIM, hpq\_phiheal\_life\_mem PLM

WHERE H.id = PSM.hpq\_hh\_id AND H.id = PEM.hpq\_hh\_id AND H.id = PIM.hpq\_hh\_id AND H.id = PLM.hpq\_hh\_id AND PSM.phiheal\_spon\_mem\_refno = PEM.phiheal\_empl\_mem\_refno AND PEM.phiheal\_empl\_mem\_refno = PIM.phiheal\_indiv\_mem\_refno AND PIM.phiheal\_indiv\_mem\_refno = PLM.phiheal\_life\_mem\_refno

GROUP BY H.mun,H.zone,H.brgy

HAVING benefCount > :count

The expected output is a series of geographical locations and their corresponding number of common beneficiaries in all the aforementioned Philhealth divisions. The query returned seven rows and four columns for :count = 0. <performance>

# QUERY OPTIMIZATION

## Heuristic Optimization

Four basic steps, according to Silberschatz (2010), were applied to the queries. The first was pushing all select operations lower into the tree. All tables in the FROM clause were then put into their own individual subquery and any WHERE clauses applied to the table were included in the subquery.

The second step was applying the most restrictive select operations first. The tables with WHERE clauses were the first to have their Cartesian products taken. If there was a tie, the priority was determined by the row count of the table. The table with less number of rows were joined first to decrease intermediate table sizes.

The third step was to transform Cartesian products into theta joins. This step was simple as it was just a syntactical change, eliminating the encompassing WHERE clause altogether.

The final step was to perform project operations early. In each of the base subqueries, the SELECT clause was changed from SELECT \* to only select the necessary columns, which were the columns that will be aggregated and the foreign key.

The performance of these five steps resulted in the heuristically optimized query.

**3.2. Indices**

The next approach was to add indices on the commonly referenced columns in the peripheral tables, specifically the columns used in the WHERE clause.

**3.3. Views**

Furthermore, certain operations on the relations were abstracted into views to allow for simpler queries. The base queries used were the heuristically optimized queries.

**3.4. Stored Procedures**

The final step was to encapsulate the entire query into a stored procedure, with the parameters abstracted as formal parameters to the procedure.

# RESULTS AND ANALYSIS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eget mollis ante, ut feugiat felis. Nam molestie ex massa, non maximus dolor tempus a. Pellentesque ultrices tincidunt est accumsan bibendum. Nunc dictum velit sed pharetra malesuada. Mauris lectus ex, dapibus id dui ac, fermentum egestas quam. Sed in posuere ante. Nulla auctor convallis posuere. Nunc in justo id dui bibendum fermentum. Morbi a leo neque. Pellentesque id magna dui.

In vitae lorem sed urna blandit pharetra. Phasellus risus urna, eleifend vel ultricies non, pharetra non felis. Nunc quis justo hendrerit, tincidunt tortor vel, vestibulum leo. Morbi ac ex ex. In hac habitasse platea dictumst. Fusce et erat sagittis, luctus lectus eget, varius orci. Proin consectetur tempor elit vel laoreet. Cras suscipit dolor sit amet justo cursus blandit. Donec euismod nisi ut ultrices imperdiet. Curabitur hendrerit malesuada justo vitae facilisis. Suspendisse interdum gravida ligula, eu pellentesque mi commodo quis. Duis velit tellus, vehicula vel leo et, aliquam faucibus ex. Cras rhoncus tristique placerat. Mauris lacinia mi id pharetra tempus. Etiam tincidunt nec metus ac suscipit. Sed eu faucibus metus, non rhoncus mi.

Vestibulum non rutrum lectus. Quisque sed orci ut arcu tempus ultricies. Donec facilisis leo ligula, et elementum felis auctor eget. In volutpat consectetur magna porttitor lobortis. Cras luctus arcu at ornare condimentum. Quisque lacinia sem quis eleifend dictum. Aliquam vitae arcu vel est bibendum porttitor. Phasellus id magna sed lorem egestas hendrerit. Nullam in ullamcorper lorem, laoreet dapibus justo.

# CONCLUSION

Aliquam auctor leo nisl, et cursus neque luctus nec. Phasellus a velit eu ligula gravida lobortis nec sed enim. Proin quis dui et est condimentum vulputate a finibus arcu. Maecenas tincidunt mi metus, sit amet iaculis lectus suscipit at. Donec hendrerit, tortor at congue scelerisque, massa elit hendrerit mauris, id imperdiet metus enim eleifend sapien. Proin imperdiet efficitur risus eu ultrices. Proin lobortis sit amet nisi ac blandit. Suspendisse lacinia dolor sed nibh convallis ullamcorper. Vestibulum porta et dui in fringilla. Vivamus massa eros, venenatis et mattis ac, pulvinar vel neque. Vestibulum sed facilisis urna, sollicitudin rutrum justo. Praesent ac odio quis purus posuere hendrerit.

Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Ut aliquam justo eget aliquam faucibus. Aliquam semper at urna ac sagittis. Cras id cursus diam, eget malesuada erat. Proin volutpat leo non neque ornare, ut consequat justo eleifend. Suspendisse ut risus suscipit, sollicitudin odio ut, suscipit lorem. Nam vel quam ultrices, cursus justo vel, pharetra purus. Aenean porta ipsum vitae risus tincidunt convallis. In eget aliquam velit. Nulla sit amet nibh et justo vulputate congue in ac elit. Quisque dolor justo, euismod quis metus in, euismod lobortis magna. Vestibulum dictum, sapien quis auctor semper, dui metus posuere elit, sed volutpat odio magna a justo. Ut congue non enim laoreet vestibulum. Vestibulum semper tempor est, nec semper velit euismod eget. Ut pellentesque nibh tortor, non bibendum ante facilisis quis. Vivamus vel quam justo.

# REFERENCES

1. Silberschatz, A., Korth, H. & Sudarshan, S. (2010). Database System Concepts, 6th Edition, McGraw-Hill Book Co. Ding, W. and Marchionini, G. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park.