# SKYLINE QUERY APPROACHES IN STATIC AND DYNAMIC INCOMPLETE DATABASES

BY

YONIS GULZAR

A thesis submitted in fulfillment of the requirement for the degree of Doctor of Philosophy in Computer Science

Kulliyyah of Information and Communication Technology
International Islamic University Malaysia

SEPTEMBER 2018

# ABSTRACT

Skyline queries attempts to return the superior data items from a database which are not being dominated by any other data items. In some real-world databases where data might not be complete, i.e. data items often have missing values in one or more dimensions, applying skyline algorithms designed for complete data is inappropriate due to the fact that missing values leads to losing *transitivity property* of skyline method which would raise the issue of *cyclic dominance*. Nevertheless, several research works have been conducted focusing on the issue of processing skyline queries in static incomplete database, whereby data rarely changes. However, an efficient approach is yet to be proposed aiming at reducing the number of pairwise comparisons to identify skylines in *static* incomplete database. Moreover, the problem might be more worsen in cloud environment whereby database relations are spread over many datacentres and remote access is needed to identify the skylines. Collecting data from these remote datacentres without prior filtration is undesirable and results in transferring unnecessary large amount of data. In addition, in some real-life database systems, database might be under frequent update operations such as insert. This insert operation keeps the content of the database to be *dynamic* in which the contents always updated with new data items. This insert operation will definitely result in invalidating the skyline results and, therefore, re-evaluating skylines on the updated database must be performed to identify the new skyline answer. Re-applying skyline method on the entire updated database is impractical and leads to prohibitive cost due to the exhaustive pairwise comparisons.

This thesis proposes an efficient skyline approach which is able to derive the skylines in a database with incomplete data. The proposed approach exploits the idea of sorting and clustering data as well as employs the concept of generating *domination power* (*dp*) to eliminate the dominated data items which in turn reduces the number of pairwise comparisons to identify the skylines. Two optimization techniques have been used in the proposed approach to eliminate many dominated data items before applying skyline technique. The approach is extended to process skyline queries in cloud incomplete databases in which database relations are distributed among many remote datacentres. The approach attempts to identify the skylines with the purpose of reducing the pairwise comparisons, processing time and amount of data transfer. In addition to that, in this thesis, an approach is proposed to process skyline queries in incomplete database with update operation (insert). The approach tries to derive the new skylines after inserting data items into the database with the goal to reduce the search space by avoiding re-evaluating skylines on the entire database. The idea of the proposed approach relies on performing a progressive scan on a subset of a database to re-identify the skylines based on the new contents. Many experiments on synthetic and real datasets have been accomplished. The results showed that our proposed approach for processing skyline queries in incomplete database has reduced the number of pairwise comparisons and the processing time compared to the previous approaches. Besides, for cloud databases, our approach for processing skyline queries achieved a significant reduction in the processing time and network cost compared to the previous approaches. Lastly, the results for processing skyline queries on incomplete database with insert operation have shown that our approach outperforms the previous approaches in terms of number of pairwise comparisons and processing time.

# خلاصة البحث

يسعى استعلام ال SKYLINES جاهدا لاسترجاع السيطرة على عناصر البيانات من قاعدة بيانات جديدة غير متحكم بها من اي عناصر بيانات اخرى. في كثير من تطبيقات قواعد البيانات قد تكون البيانات غير مكتملة ، اي بمعنى عناصر البيانات غالبا ما تفقد بعض القيم في عمود واحد أو اكثر. تطبيق خوارزميات SKYLINE المصممة للبيانات الغير ناقصة غير ملائمة نتيجة لافتقاده القيم، وبالتالي تفقده خاصية الانتقال(*transitivity property*) الخاصة بطريقة SKYLINE والتي لربما تنشئ مشكلة السيطرة الدائرية (*cyclic dominance*). ركزت العديد من البحوث على مشكلة عملية استعلام SKYLINE في قاعدة بيانات ثابتة و غير مكتملة والتي بها البيانات نادرا ما تتغير. ومع ذلك تم اقتراح دراسة فعالة تهدف الى تقليص عدد المقارنات الزوجية لايجاد SKYLINES في قاعدة بيانات ثابتة وغير مكتملة. علاوة على هذا المشكلة ربما تبدو المشكلة أسوء في البيئة السحابية ( CLOUD) حيث علاقات البيانات تكون منتشرة عبر العديد من مراكز البيانات و يكون الوصول عن بعد مطلوب لغرض إيجاد SKYLINES. تجميع البيانات من مراكز البيانات عن بعد بدون اي تصفية مبدئية يعتبر شي غير مرغوب به وينتج عنه جمع مقدار ضخم من البيانات الغير ضرورية. بالاضافة الى هذا، فان قواعد البيانات في بعض انظمة قواعد البيانات الحقيقية تكون خاضعة لعمليات تحديث متكررة مثل عمليات أضافة بيانات جديدة. عملية إدخال بيانات جديدة تؤدي إلى تغير نتائج عمليات الأستعلام لل SYKLINES و بالتالي يجب أن يتم إعادة تقييم نتائج هذه الأستعلامات لل SKYLINES من جديد. إعادة تطبيق طريقة SKYLINE على كل المحتوى في قاعدة البيانات الجديدة بشكل كامل يعتبر حل غير عملي ويؤدي الى تكلفة باهضة نتيجة للمقارنات الزوجية الغير ضرورية و المستنزفة. هذه الاطروحة تقترح اسلوب فعال للتعامل مع أستعلام الSKYLINES والذي يكون قادر على إيجاد ال SKYLINES في قاعدة البيانات الجديدة ذات بيانات غير مكتملة. الاسلوب المقترح يستكشف فكرة تنظيم وتجميع البيانات بالاضافة الى انه يخدم مفهوم توليد القوة المسيطرة (*dp*) *domination power* للحد من عناصر البيانات المسيطرة والذي يؤدي الى تقليل عدد المقارنات الزوجية في ايجاد الSKYLINES. تم استعمال تقنيتين مثاليتين في الدراسة المقترحة للتقليل من عدد عناصر البيانات المسيطرة قبل تطبيق تقنية SKYLINE. امتدت الدراسة لتشمل العمل باستعلام SKYLINE في قواعد البيانات السحابية غير المكتملة والتي تشتمل على علاقات قاعدة بيانات منتشرة عبر مراكز المعلومات عن بعد. وتسعى هذه الدراسة لايجاد SKYLINES يكون هدفها تقليص المقارنات بين القيم والتحكم بوقت ومقدار نقل

المعلومات. بالاضاقة الى هذا الدراسة المقترحة في هذه الاطروحة تقترح تطبيق استعلام SKYLINE على قاعدة بيانات غير مكتملة مع عمليات الأضافة لبيانات جديدة. كما تسعى هذه الدراسة لاشتقاق SKYLINES جديدة بعد ادخال عناصر جديدة في قاعدة البيانات يكون هدفها الحد من مساحة البحث عن طريق تجنب اعادة تقييم SKYLINE في قاعدة البيانات ككل. فكرة هذه الدراسة المقترحة تعتمد على تطبيق مسح فعال على مجموعة قاعدة البيانات لإعادة ايجاد SKYLINES بناء على المحتويات الجديدة. تم اجراء العديد من التجارب على قواعد بيانات حقيقة ومركبة، وقد اظهرت نتائج هذه الدراسة المقترحة لمعالجة استعلام SKYLINE على قاعدة بيانات غير مكتملة تقلص واضح في عدد المقارنات الزوجية وفي الوقت المستهلك للتشغيل مقارنة بدراسات سابقة. الى جانب هذا بالنسبة لقواعد البيانات السحابية فان دراستنا التي تعتمد على معالجة استعلام SKYLINE قد حققت تقليص ملحوظ في الوقت المستهلك للتشغيل وفي تكلفة الشبكة مقارنة بدراسات سابقة. ختاما اظهرت نتائج معالجة استعلام SKYLINE على قاعدة بيانات غير مكتملة مع عملية الادخال بان دراستنا قد فاقت نظيراتها السابقة من حيث عدد المقارانات الزوجية والوقت المستهلك.

# APPROVAL PAGE

The thesis of Yonis Gulzar has been approved by the following:

_____
Ali A. Alwan
Supervisor

_____
Norsaremah Salleh
Co-Supervisor

_____
Amelia Ritahani Bt. Ismail
Internal Examiner

_____
Rohaya Latip
External Examiner

_____
Feras Hanandeh
External Examiner

_____
Sohirin Mohammad Solihin
Chairman

# DECLARATION

I hereby declare that this thesis is the result of my own investigations, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at IIUM or other institutions.

Yonis Gulzar

Signature ..... Date .....

## INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

## DECLARATION OF COPYRIGHT AND AFFIRMATION OF FAIR USE OF UNPUBLISHED RESEARCH

## SKYLINE QUERY APPROACHES IN STATIC AND DYNAMIC INCOMPLETE DATABASES

I declare that the copyright holders of this thesis are jointly owned by the Student and IIUM.





By signing this form, I acknowledged that I have read and understood the IIUM Intellectual Property Right and Commercialization policy.

Affirmed by Yonis Gulzar

……..…….. ……..
Signature Date

*To my Dearest and First Teachers; My Father and Mother*
*To my lovely sisters*
*To my beloved friends and colleagues*

# ACKNOWLEDGMENTS

First of all, I am gratified with the core of my heart to Almighty Allah who made it possible to complete this thesis.

I must acknowledge my work to my dear father Gulzar Ahmad Dar, mother Shameema Bano and my sisters Kulsuma Gulzar, Sakeena Gulzar and Fatima Gulzar. Without their support, concern, and love, it was impossible for me to complete my Ph.D. studies. I especially thank my father who encouraged me to pursue my Ph.D.

I am also grateful to my supportive supervisor Assist. Prof. Dr. Ali A. Alwan and co-supervisor Assoc. Prof. Norsaremah Salleh who have continuously encouraged me throughout my research. I am especially thankful to my main supervisor, Dr. Ali who guided me with great patience and persistently motivating me during my research. He made me learn many things ever since I enrolled in the Ph.D. program at IIUM despite my humble beginnings in research work. Thank you very much, Dr. Ali, for being my supervisor and mentor.

I am highly indebted to my dear friends namely Mohammad Shuaib Mir, Ms. Norizan Tan and the members of my research unit 10 at IIUM, Br. Badamasi Imam, Dr. Arjumand, Dr. Norzariyah, and Sr. Marwa for their presence in sharing and exchanging thought-provoking ideas for the solution for my research problems. Br. Imam and Ms. Norizan Tan thank you for your support during my thesis submission as it would be far more difficult for me to do it without your support.

Thank you so much.

## TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER ONE

# INTRODUCTION

## 1.1 OVERVIEW

Query processing is one of the vital operations in the database system. When a user submits a query to a database, a set of data items is retrieved from the database, which satisfies the query constraints. In this era, there are numerous numbers of interactive applications storing a tremendous amount of data. To obtain results from these kinds of applications efficient processing of queries is needed. So that users obtain results in brief timeframe and at the same time suit their necessities.

Lately, there has been much focus on the design and development of database management systems that incorporate and provide more flexible query operators that return data items, which are not dominated by any other data item in all attributes (dimensions). For instance, if there are two data items $dt_1$ and $dt_2$ and a query prefers $dt_1$ than $dt_2$. It can be only possible if and only if $dt_1$ is better than $dt_2$ in all dimensions and is not worse than $dt_2$ in at least one dimension. Such type of query is called preference query. The preference queries are significant and mostly used in various application domains, like multi-criteria decision-making applications (Chan et al., 2006a, 2006b; M. Kontaki et al., 2008; Mohamed A. Soliman et al., 2010; Yiu & Mamoulis, 2007), where numerous criteria are involved with the query statement to choose the most appropriate answer that fits the user needs. Decision support system and recommender system is another database application where preference queries are applied. In these systems, various interests are combined in order to help users by recommending a strategic decision. Hotel recommender (Godfrey et al., 2005; Wong et al., 2008) and restaurant finder (Kukhun et al., 2008; Mokbel & Levandoski, 2009) are typical

examples that show the significance of preference queries. Furthermore, E-commerce environment is also a significant area that involves preference queries (Fotiadou & Pitoura, 2008). For instance, helping shopper to make a tradeoff between the price, quality, and efficiency of the goods to be bought. Last but not least, preference queries are also used in the peer-to-peer network (Fotiadou & Pitoura, 2008).

Due to the significance of preference queries that has been clearly shown in many database applications, different types of preference evaluations techniques have been proposed but not limited to, top-*k* (Chaudhuri & Gravano, 1999; Ilyas et al., 2008), Skyline (A. A. Alwan et al., 2016; Bartolini et al., 2006; Borzsony et al., 2001; Jan Chomicki et al., 2005; Godfrey et al., 2005; Khalefa et al., 2008; Kossmann et al., 2002; Jongwuk Lee, Im, et al., 2016; Pei et al., 2005; Mohamed A. Soliman et al., 2010; Wong et al., 2008; Y. Yuan et al., 2005), k-dominance (Chan et al., 2006b), top-*k* dominating top(Miao, Gao, Zheng, et al., 2016; Yiu & Mamoulis, 2007), and *k*-frequency (Chan et al., 2006a). It has been very obvious that skyline queries are very beneficial and most widely used in the contemporary database applications.

Two significant concerns have been raised by the researchers since the introduction of skyline queries into the database community. The first concern is how to derive the skylines with the intention of shrinking the searching space. While the second concern is to avoid scanning the whole database and limit the search to those data items with the high potential to be in the skyline results. The searching space is determined by the number of pairwise comparisons that need to be performed between the data items in identifying skylines. That means a higher number of pairwise comparisons results in larger searching space and vice versa. Skyline queries endeavors to identify the unrivaled data items (tuples) that are not dominated by any other data item in the database. In the multi-dimensional database, a set of data items *S* called

skylines prunes other data items that not better than data items in *S* in any dimension. For instance, assume $r_1$ and $r_2$ are two different data items, $r_1$ can be only part of the skyline set, S if and only if $r_1$ is better than $r_2$ at least in one dimension and $r_1$ is not worse than $r_2$ in any dimension.

## 1.2 PROBLEM STATEMENT

Many solutions rely on the concept of skyline technique have been suggested aiming at deriving skylines, this include Divide-and-Conquer (D&C), Block Nested-Loop (BNL)(Borzsony et al., 2001), Bitmap and Index (Tan et al., 2001), Sort Filter Skyline (Jan Chomicki et al., 2005), Nearest Neighbor (NN) (Kossmann et al., 2002), Linear Elimination and Sort Skyline (LESS) (Godfrey et al., 2005), Branch and Bound Skyline (BBS) (Papadias et al., 2003), and Sort and Limit Skyline algorithm (SaLSa) (Bartolini et al., 2006).

From the literature, it is very obvious to conclude that most of these solutions assumed that data item values are present and dimensions (attributes) are always populated with some values (Bartolini et al., 2006; Borzsony et al., 2001; Jan Chomicki et al., 2005; Godfrey et al., 2005; Z. Huang & Wang, 2006; Kossmann et al., 2002; J. Lee et al., 2009; Papadias et al., 2003; Pei et al., 2005; Tan et al., 2001; Yiu & Mamoulis, 2007, 2009; Y. Yuan et al., 2005). Nevertheless, it is argued that this assumption is not necessary to be always true, particularly for a database with a high number of dimensions and a tremendous amount of data. In many of the contemporary databases application such as crowd-sourcing, temporal, spatial, probabilistic, uncertain and big data databases, it is most likely that some values are missing due to many causes. Moreover, the incompleteness of data has a direct negative impact on processing skyline queries and in many cases, results in high overhead, due to exhaustive pairwise

comparisons between the data items. Most importantly, the incompleteness of data leads to the loss of the *transitivity property* of skyline technique, which is held on all existing skyline techniques applied on complete data. This further leads to *cyclic dominance* between the data items as some data items are incomparable with each other and thus no data item is considered as a skyline (A. A. Alwan et al., 2016; Bharuka & Kumar, 2013a; Khalefa et al., 2008; Jongwuk Lee, Im, et al., 2016; Mohamed A. Soliman et al., 2010).

For example, consider the following scenario where a tourist is seeking a restaurant in a city that is the nearest to his current location, cheaper in price and has the best rating. A restaurant $r_i$ is represented in three dimensions ($d_i$, $p_i$, $r_i$) where $d_i$, $p_i$, and $r_i$, represent the distance, price, and the restaurant rating respectively. Assume the restaurant database consists of 3 data items with missing values, $r_1$(3, *, 4), $r_2$(4, 3, *), and $r_3$(*, 4, 5). The symbol (*) is used to represent the missing values in the records. Based on the common dimensions with non-missing values, $r_1$ dominates $r_2$ as $r_1$ is better than $r_2$ in the first dimension (distance) (greater is better), while $r_2$ dominates $r_3$ as $r_2$ is better than $r_3$ in the second dimension (price). Based on the transitivity property when $r_1$ dominates $r_2$, and $r_2$ dominates $r_3$, therefore $r_1$ must dominate $r_3$, which does not happen in our example. However, $r_3$ dominates $r_1$ in the third dimension (better rating) which means that the dominance relation is cyclic. From this example, all these three restaurants are being dominated, and thus, no skylines will be revealed.

(Khalefa et al., 2008) is the first study that highlights the issue of finding skylines in incomplete database proposing an approach named *Iskyline*. The idea of *Iskyline* relies on dividing the initial incomplete database into different nodes where each node has different *bitmap representation*. The proposed solution uses two optimization techniques namely *virtual points* and *shadow skylines* to reduce the number of data

items (points) in local skylines and candidate skylines list. Nevertheless, *Iskyline* algorithm is time consuming while taking a lot of pairwise comparisons between data items to retrieve final skylines. Furthermore, (Bharuka & Kumar, 2013a) also, have highlighted the issue of finding skyline queries over incomplete data. They proposed an approach named Sort-based Incomplete Data Skyline (SIDS) that sorts data according to descending order for each dimension to determine the processing order of data item and processes every data item in a round-robin fashion. Then, the data item having the best value in non-missing dimension is processed next. However, SIDS algorithm is taking higher number of pairwise comparisons to derive the skylines from the incomplete database due to the following reasons. First, all data items present in the dataset are compared with one another without any advance filtration to prune the dominated data items before applying skyline process. Second, the pairwise comparisons conducted between data items are sequential which makes the execution time to be proportional to dataset size.

The most recent work on skyline queries in an incomplete database is contributed by (A. A. Alwan et al., 2016). They proposed an approach called Incoskyline for handling skyline queries in incomplete data. Incoskyline uses the same idea proposed by (Khalefa et al., 2008) to divide the data items into clusters according to their *bitmap representation*. However, it uses two optimization techniques *data grouping* and *deriving k-dom skylines* to simplify the skyline process and reduce the number of pairwise comparisons performed between data items during skyline operation. However, in the extreme case where database has a high ratio of missing values the approach makes the pairwise comparison more complicated. Thus, an approach is needed to avoid the unnecessary pairwise comparison among data items when retrieving skylines.

Most of the previous approaches (A. A. Alwan et al., 2016; Bharuka & Kumar, 2013a; Khalefa et al., 2008) are tailored for a centralized database and accessed only one table to identify the skylines. However, in many contemporary database applications, this might not be the case, particularly for a database with incomplete data and many database relations are spread over various remote locations such as cloud environment. Applying skyline approaches designed for centralized database directly on cloud databases is undesirable due to the prohibitive cost of transferring the amount of data from one datacenter to another during skyline process. To the best of our knowledge, the latest work that raised the issue of processing the skyline queries in incomplete distributed databases is contributed by (Ali A. Alwan et al., 2017). However, this work is limited to the case of database tables, which are vertically partitioned where the attributes (dimensions) of the table are located on different sites and remote access needs to be performed during the skyline process. Besides, the architecture of cloud is quite different from the distributed environment (Abourezq & Idrissi, 2014; Rehioui et al., 2016). Hence, an approach is needed to process skyline queries in cloud incomplete database aiming at reducing number of data centers involved, amount of data transferred, the number of pairwise comparisons and reducing processing time.

Another issue that needs to be tackled is determining skylines in a dynamic incomplete database where data regularly change through insert operation. The work in (A. A. Alwan et al., 2016; Bharuka & Kumar, 2013a; Khalefa et al., 2008) addressed only the issues related to incomplete data in a static database where data do not change often. The newly inserted data might result in invalidating the skyline result and might leads to re-apply skyline technique on the entire database to identify the new skylines. The process of re-computing skyline on the entire database is prohibitive and undesirable due to the high overhead and the exhaustive pairwise comparisons. Thus, a

method is needed to process skyline queries in dynamic incomplete database by avoiding re-applying skyline technique directly on the database.

## 1.3 OBJECTIVE OF THE RESEARCH

The aim of this research work is to achieve the following objectives:

I. To propose an efficient approach that identifies the skylines in incomplete database with the intention of reducing the number of pairwise comparisons and processing time.

II. To identify the problem of processing skyline queries in cloud incomplete database where data is distributed horizontally. To this end, an efficient approach to derive skylines in cloud incomplete databases will be proposed with the aim of reducing the domination tests, the processing time and the amount of data transfer to derive the skylines.

III. To design and develop an approach that processes skyline queries in incomplete database with insert operation aiming at avoiding recomputing skylines over the entire database to update the skyline results.

## 1.4 RESEARCH SCOPE

The scope of this research work is outlined in the following points:

- This research work used the relational data model as it is the most dominant model among the conventional models (A. A. Alwan et al., 2016; Bharuka & Kumar, 2013a; Khalefa et al., 2008; K. C. Lee et al., 2010; Levandoski et al., 2010; Lin et al., 2007; Sun et al., 2008; Wolf et al., 2009; Y. Yuan et al., 2005).

- The type of queries which is considered in this research is the preference queries and limited to skyline queries as it is the most frequently applied technique in the multi-criteria decision-making applications (A. A. Alwan et al., 2016; Bartolini et al., 2006; Bharuka & Kumar, 2013a; Borzsony et al., 2001; Jan Chomicki et al., 2005; Godfrey et al., 2005; Khalefa et al., 2008; Kossmann et al., 2002; J. Lee et al., 2009; Pei et al., 2005; Tan et al., 2001; Yiu & Mamoulis, 2007, 2009; Y. Yuan et al., 2005).
- This research work concentrates on two types of database, namely: synthetic and real database. The synthetic database includes independent, correlated, anti-correlated while the real databases include NBA, Movie-Lens and COIL Insurance Company. These are the most popular types of databases that have been used in this area (A. A. Alwan et al., 2016; Bharuka & Kumar, 2013a; Chan et al., 2006a, 2006b; Khalefa et al., 2008; K. C. Lee et al., 2010; Yiu & Mamoulis, 2007).
- This research work assumes that the missing values might occur in one or more dimensions (A. A. Alwan et al., 2016; Bharuka & Kumar, 2013a; Khalefa et al., 2008).
- This research work assumes that the database is dynamic where data frequently changes through insert operation.