Abstract—The explosive growth of information in recent years has brought up a new question/challenge of disseminating information from a huge plethora of data. This has led to an upscale development of interest in Data warehousing. Different indexing and Performance Enhancement techniques have been developed which are being used for fast data retrieval in Data warehouse environment, but it is quite difficult to find an appropriate technique for a specific query type or specific operation. So an investigation is needed to find Enhancement and some optimization techniques to improve the data warehouse performance. The objective of our paper is to find out performance optimization and Enhancement techniques which improve the processing time and faster data retrieval in data warehousing.

Keywords—Data Warehouse, ETL process, Parallel Processing, Views, Partitioning, Clustering, Indexing

I. INTRODUCTION

A data warehouse is generally defined as a collection of subject-oriented, integrated, non-volatile, and time varying data to support decisions makers. A data warehouse is the copy of transaction data especially structured for querying, reporting and analysis purpose. Data warehouse is the source of stable and integrated data designed to support decision makers and business analysts. Data are acquired from various operational data stores across the enterprise. After acquisition of the data, cleansing, transformation and possibly denormalization is been applied for better performance. Analysts use the data warehouse to answer unlimited variety of questions, which may be very difficult to answer in operational database. Data warehouse contains a number of databases regardless of the number. A data warehouse is a centralized repository that containing comprehensive detailed and summary data that provides a complete view of Business needs. The Mission of the Data Warehouse is to provide the business with standardized, consistent, timely and accurate information to improve the effectiveness, efficiency and timeliness of business insights and decisions.

Section I tells us about the Basic introduction Data warehouse. Section II suggests the seven steps to optimize Data warehouse performance and some enhancement techniques. And apply these techniques on Case study of Library management system which is based on decision support system. Paper concludes with conclusion in Section III.

II. LITERATURE SURVEY

To meet user demands for more timely and flexible analysis, companies can use a step-by-step approach to move from maintaining detailed information to using summary-level data. The evolution of data warehousing has followed a pattern of compromising the project to overcome technical barriers at the expense of business value. As data volume grew on analytical systems, queries took more time. Many developers began preplanning for the Design and implementing better Data warehouse. This preplanning included

- Creating summary tables of aggregated data
- Placing indexes in the system to eliminate scanning of large volumes of data
- Denormalizing the model—putting data into one table instead of having tables that need to be joined together
- Storing the data in sorted order, thus avoiding and sort activity.

In this paper we select Library management system as our case study for performing performance improvement of data warehouse. The College library management information system can have produced massive the management on books and borrower, borrowing and returning books and some other data every day, and other department separately has also provided the reference room and the books management information system.

![Figure 1: Data Warehouse Structure Diagram of Library Management System](image-url)

In this paper, a new library decision support system based on the data warehouse was constructed as shown in [Figure 1]. Source systems are various existing application system of
daily transaction process. After being integrated, classified, purified and so on, these data of original transaction process systems and external data source are extracted and loaded into the library data warehouse.

A. SEVEN STEP TO OPTIMIZE DATA WAREHOUSE PERFORMANCE

As Below Figure 2 shows, seven distinct steps [1]. An important point to remember is that if your technology cannot support starting at the lowest level, you should seriously rethink the ability of that technology to support your ongoing business needs. This step should be applied on the basis of business type which is run by an organization.

1. Extract data from Operational Data Source (ODS)

2. Schema of Data warehouse like star, snowflake schema.

Borrow of book star schema Shown in [Figure 3]. Star schema is a dimensional model with a fact table (large) and a set of dimension tables (small). The whole set-up is totally denormalized. However in cases where the dimensions tables are split to many table that is where the schema is slightly inclined towards normalization (reduce redundancy and dependency) there comes the snowflake schema. The snowflake schema does the best performance when queries involve aggregation. Star model increased speed of processing as compared to snowflake schema because no need to use complicated joins and get queries results fast.

Step 1: Use Functionally Neutral Data Model

An important starting point for designing, building and implementing an EDW is the design and construction of an appropriate Data Model for this environment. The Enterprise Data Model for the operational environment requires extensive transformations and further refinements if it is to effectively represent the data requirements of the Data Warehouse. The lowest level consists of using atomic data in a normalized model that is not specific to any function or group. The data includes the lowest level of detail necessary to support the business objective. In this step we choose a model which is best suitable for managing organizational Database.

Data warehouse's multi-dimensional data model structure mainly has star schema, snowflake schema and mixed schema. Specially, a star schema can express the Multidimensional data clearly, easy to understand and expand, is good with the existing relational database management system coordination. A star schema is a specific type of database model used to support analytical processing. In this step mainly include two operations

1. Expand System and Add Indexes

2. Implement Table View

3. Use Functionally Neutral Data Model
CREATE VIEW Dept_Stud_List AS SELECT * FROM Student_DB WHERE Stud_dept='Computer'

We can also use the Materialized view. A Materialized View is effectively a database table that contains the results of a query. Materialized views can be used for many purposes, including:

- Data Warehousing
- Replication
- Validation
- Denormalization

A materialized view can be stored in the same database as its base table(s) or in a different database. Materialized views stored in the same database as their base tables can improve query performance through query rewrites. Query rewrites are particularly useful in a data warehouse environment. A materialized view log is a schema object that records changes to a master table's data so that a materialized view defined on the master table can be refreshed incrementally. A Materialized View is effectively a database table that contains the results of a query. The power of materialized views comes from the fact that once created.

CREATE MATERIALIZED VIEW View_borrow_fact AS SELECT * FROM Borrow_fact_table;

Data in materialized views must be refreshed to keep it synchronized with its base table. Refreshing can either be done manually, as below, or automatically by Oracle in some cases.

Execute dbms_mview. REFRESH (View_borrow_fact);

Step 3: Add Indexes

To improve response time, data warehouse administrator usually uses Indexing techniques. The index should be able to operate with other indexes to filter out the records before accessing original data. There are many advantages of indexing such as:

- Faster key-based access to table data
- Reduced storage requirements
- Efficient retrieval

In this paper, we studied different indexing techniques and analysis based on this indexing techniques, such an indexing techniques include [2]

1. Bit- Mapped indexing techniques
2. Cluster indexing Techniques
3. Hash-based index
4. B-Tree Index

In our case study of Library database, we applied Bit-Mapped indexing Techniques because bitmap index is a special type of structure used by database management systems to optimize search and retrieval for low variability data. Mostly Bitmap indexes are widely used in data warehousing where that contain large amounts of data but a low level of concurrent DML transactions. For such applications, bitmap indexing provides:[2]

- Reduced response time for large classes of ad-hoc queries.
- Reduced storage requirements as compared to other indexing techniques.
- Performance should be gains on hardware if relatively small number of CPUs or a small amount of memory presents
- Efficient maintenance during parallel DML.

E.g. CREATE BITMAP INDEX borrow_ids ON Borrow_fact_table (Stud_id, Book_id, acc_id);

Step 4: Expand the System

The physical environment of a data warehouse is critical to its successful implementation. The physical architecture of the warehouse includes all:

- Hardware
- Network configurations
- Security requirements
- End user tools
- Tangible pieces of the data warehouse.

Selecting the Operating system, software, hardware, Computer system and network capacity, DWH tools and Techniques, this supports to your organizational structure. For this purpose we have to consider cost, disk space, Data management and Time lags that support to data warehouse. In our case study is college level so it’s not necessary to expand the system configuration at higher level. So this step is optional

Step 5: Rational Summaries and Denormalization

In this step, you make summary tables not completely tailored to a specific function—for example, at branch level instead of account level—to keep the same data structure but at a higher level of aggregation. This approach is possible in relatively small, less frequently maintained reference tables, but it is less feasible as data volume or volatility increases. We, make summary table not completely too specific function

But in library database system we accept data from different department is nothing but data mart. So this step is not necessary for our case study.

Step 6: Irrational Summaries and Denormalization

If rational denormalization still does not achieve the desired performance, you can take the “irrational” step of building summaries and data models specifically geared to a single function. The tables will have the exact columns you need in the report, but using this approach means that you forgo any pretense of flexible analysis. A data mart is a subset of an organizational data store, usually oriented to a specific purpose or major data subject that may be distributed to
support business needs.

Here we used data mart concept at department wise database. A data mart is a scaled down version of a data warehouse that focuses on a particular subject area. A data mart is a subset of an organizational data store, usually oriented to a specific purpose or major data subject that may be useful to support business needs. Here we create different data mart related to different department. Following are some Reasons for creating a data mart

- Easily access to frequently needed data
- Creates collective view by a group of users
- End-user response time should be improved
- Easily created in less amount of time
- Lower cost than implementing a full Data warehouse
- Specific Data should be restricted to the particular user

**Step 7: Export Data**

If you still don’t get the desired performance, you must make some difficult decisions. You can expand the system to drive performance, export data off the platform to a special system for functional analysis, or revisit the performance requirement the business.

Import/Export tools are used to transfer data from Source database to destination database. Export is used to export data from operational database system and Import is used to load data into the data warehouse after clean and transform. When you export tables from source database export tool will extracts the tables and puts it into the dump file. This dump file is transferred to the target database. At the target database the Import tool will copy the data from dump file to the target database. Sometimes we can also be used ETL tool for Extract, transfer and load operations.

**Command Line Parameters of Export tool:**

C:\> EXP SCOTT/TIGER GRANTS=Y TABLES=
(Student_DB,Book_DB,Borrow_DB,Account_DB)

**Export full database:**

C :> EXP USERID=SCOTT/TIGER FULL= y FILE=myDB.dmp

**B. PERFORMANCE ENHANCEMENT TECHNIQUES**

Apart from the indexing techniques we have discussed in the previous section, a few other methods[4] also improve performance in a data warehouse. For example, physically compacting data when writing to storage enables more data to be loaded into a single block. That also means that more data may be retrieved in one read. Another method for improving performance is the merging of tables. Again, this method enables more data to be retrieved in one read. If you purge unwanted and unnecessary data from the warehouse in a regular manner, you can improve the overall performance.

**a) Data Partitioning**

Partitioning means deliberate splitting of a table and its index data into manageable parts. The DBMS supports and provides the mechanism for partitioning. When you define the table, you can define the partitions as well. Each partition of a table is treated as a separate object. The ability to partition table data across multiple storage objects provides your database administrators with greater scalability, flexibility, control, and performance. Specific benefits and features of table partitioning include:

- Separating data using table partitioning allows you to improve query processing performance by avoiding scans of irrelevant data.
- Using table partitioning with the Database Partitioning Feature (DPF), you can spread ranges of data evenly across database partitions to take advantage of the intra-query parallelism and database partition load balancing features of DPF.
- Administration is more flexible because you can perform administrative tasks on individual data partitions, breaking down time-consuming maintenance operations into a series of smaller operations. For example, you can back up and restore individual data partitions instead of entire tables.
- The load utility inserts data records into the correct data partition without the need to use an external utility to partition the input data before loading.

In this paper we learned Different partition Method [6], as below. You can apply this method as per your requirement.

- Range Partitioning
- Hash Partitioning
- List Partitioning
- Composite Partitioning

**For example- By using List Partitioning**

```
CREATE TABLE Book_DB
(Book_id NUMBER (5),
Book_name VARCHAR2 (30),
Book_author VARCHAR2 (20),
Book_IBN_No VARCHAR2 (20),
Book_price NUMBER (10,2),
Book_Type VARCHAR2 (20))
PARTITION BY LIST (Book_Type)
(PARTITION Book_Computer VALUES 1(7:25),
PARTITION Book_Mech VALUES ('Autocad', 'Machinery',
'Geometry', 'Calculus', 'Probability', 'Statistic', 'Descript Structure'));
```
b) Data Clustering

In the data warehouse, many Queries require sequential access of huge volume of data. The technique of data clustering facilitates such sequential access. Data Clustering can be achieved by physically placing related tables close to each other in storage. When you declare a cluster of table to the DBMS, the tables are placed in neighboring areas on disk. By using data clustering we reduced data accessing time.

CREATE CLUSTER Stud_Dept_cls (Stud_dept VARCHAR2 (10)) SIZE 512 STORAGE (INITIAL 100K NEXT 50K PCTINCREASE 10);

c) Parallel Processing

Consider a query that accesses large quantities of data perform summation based on multiple constraints. To achieve high performance split the processing into component and execute the component in parallel. The simultaneous concurrent executions will produce the result faster. In case of parallel processing, Oracle runs single SQL statements in parallel, in that case multiple processes are work together simultaneously by dividing the work necessary to run a statement among multiple processes. Oracle can run the statement much faster than a single process. This is called parallel execution or parallel processing. But it takes more computing power. Parallel processing of query is shown in below [Figure 4].

Parallel execution performs these operations in parallel using multiple parallel processes. One process, known as the parallel execution coordinator, dispatches the execution of a statement to several parallel execution servers and coordinates the results from all of the server processes to send the results back to the user.

The number of parallel execution servers associated with a single operation is known as the degree of parallelism (DOP). Example:

CREATE TABLE summary (C1, AVG C2, SUM C3) PARALLEL (5) AS SELECT C1, AVG (C2), SUM (C3) FROM DAILY_BOOK_TRAC GROUP BY (C1);

We also create parallelism by using

SELECT /*+ PARALLEL (Student_DB, 5) */ C1, AVG (C2), SUM (C3) FROM Student_DB;

Here we select degree of parallelism (DOP) is 5. As shown in [figure 5].

Figure 5: Parallel execution of query

d) Summary Level

DW needs to contain both detailed and summary data. Sometimes it is necessary to keep daily, weekly, monthly summaries. For this purpose we use roll up and drill down operation.

- Rolling up – it enables the navigation up the given dimension hierarchy to calculate the more and more aggregated value,
- Drilling down – that enables to go deep into a hierarchy of the dimension to carry out the more detailed analysis, e.g. searching more detailed information about sales in each store, everyday for a particular category of products,

e) Initialization Parameter

DBMS installation signals the start of performance improvement. At the start of the installation of the database system, you can carefully plan how to set the initialization parameters. Many times you will realize that performance degradation is to a substantial extent the result of inappropriate parameters. The data warehouse administrator has a special responsibility to choose the right parameters. For example, if you set the maximum number of concurrent users too High, the users will run into bottlenecks. For example-

LICENSE_MAX_USERS = 200

f) Data Array

Many times each occurrence of a sequence of data is in a different physical location Beneficial to collect all occurrences together and store as an array in a single row Makes sense only if there are a stable number of occurrences which are accessed together In a data warehouse, such situations arise...
naturally due to time based orientation can create an array by month

  g) Staging Area

  The Data Warehouse Staging Area is temporary location where data from source systems is copied. A staging area is mainly required in a Data Warehousing Architecture for timing reasons. In short, all required data must be available before data can be integrated into the Data Warehouse.

  Due to varying business cycles, data processing cycles, hardware and network resource limitations and geographical factors, it is not feasible to extract all the data from all Operational databases at exactly the same time. For that purpose of speed matching staging area is kept between operational system and data warehouse. Not all business requires a Data Warehouse Staging Area. For many businesses it is feasible to use ETL to copy data directly from operational databases into the Data Warehouse.

III. CONCLUSION

We have seen Performance Optimization and Enhancement Techniques of Data Warehouse. Data Warehousing is not a new phenomenon. All large organizations already have data warehouses, but they had some difficulty to managing them properly. Data warehouse performance is heavily dependent on proper indexing strategy. B-Tree indexes and bitmapped indexes are suitable. A proper indexing technique is crucial to avoid I/O intensive table scans against large data warehouse tables. It also depends upon the System Configuration and Volume of Data. So proper selecting of right techniques for storing as well as retrieving is necessary for data warehouse.

Other performance improvement schemes that are part of the physical design include the following: data partitioning, data clustering, parallel processing, creation of summaries, adjusting referential integrity checks, proper setting of DBMS Initialization Parameters and Use of Data Array.

REFERENCES

[1] “Seven Step To Optimize Data Warehouse Performance” By Rob Armstrongs
[5] “Ideal Strategy to Improve Data warehouse Performance “by Fahad Sultan, Dr. Abdul Aziz;aculty of information technology University of Central Punjab,Lahore, Pakistan
[8] AN ARCHITECTURE FOR DATA WAREHOUSE SYSTEMS. Jixue Liu Millist Vincent School of Computer Science, the University of South Australia
[9] Application Research of Data Warehousing Technology in Library Decision Support System, Fushan Wang Dept. of Computer Science and Technology Department of Computer Science and Technology, Dezhou University, China