

WEB LAB - Subset Extraction

Fall 2005

Authors:
Megha Siddavanahalli
Swati Singhal

Table of Contents:

Sl. No.	Topic	Page No.
1	Abstract	2
2	Introduction	2
3	Background	2
4	Scope and Constraints	3
5	Basic Subset Extraction	5
6	Design Features	5
7	Design of Subset Extraction Application	6
8	Results	9
9	Future Work	10
10	References	10
11	Acknowledgements	10
12	Appendix 1 - Discussion with Prof. Jayavel Shanmugasundaram about usage of views	11

Abstract:

Information retrieval and storage is critical and one of our major concerns. The goal of subset extraction is to facilitate access to the terabytes of data in the archive. Is it practical to scan through the entire data each time user sends in a query? What are the performance constraints? What are the storage constraints? A substantial amount of properties of subset extraction vary from user to user and it is not possible to come up with a static set of requirements. Hence we need to design a system which can adapt to different requirements of the user and provide optimal performance. The following report discusses a practical design and implementation of a subset extraction system.

Introduction:

A subset is defined as the part of the information in the database which satisfies the user query. Subset extraction is the process which provides the functionality of querying the database and creates a subset which satisfies the query. The size of the subset may vary for different queries. A user may generate a query which covers the entire database or a query which is a small section of the database. The size may vary from terabytes to even kilobytes of information.

Our problem is to design a system which can grow and shrink depending on the size of the subset required. Storage space is critical when we are talking about large data sets and care must be taken to have a system which utilizes the space in an optimal manner. Note that we do not want to replicate the data we have in the database every time a query is executed; hence we require a system that is scalable. Several models were considered.

Background:

The underlying database used for the subset extraction for the Web Laboratory project is described below -
The underlying database stores the DAT files and is shown in the following section. The database is made unique to every crawl; every crawl has its own database. The crawls are given a unique crawl id.

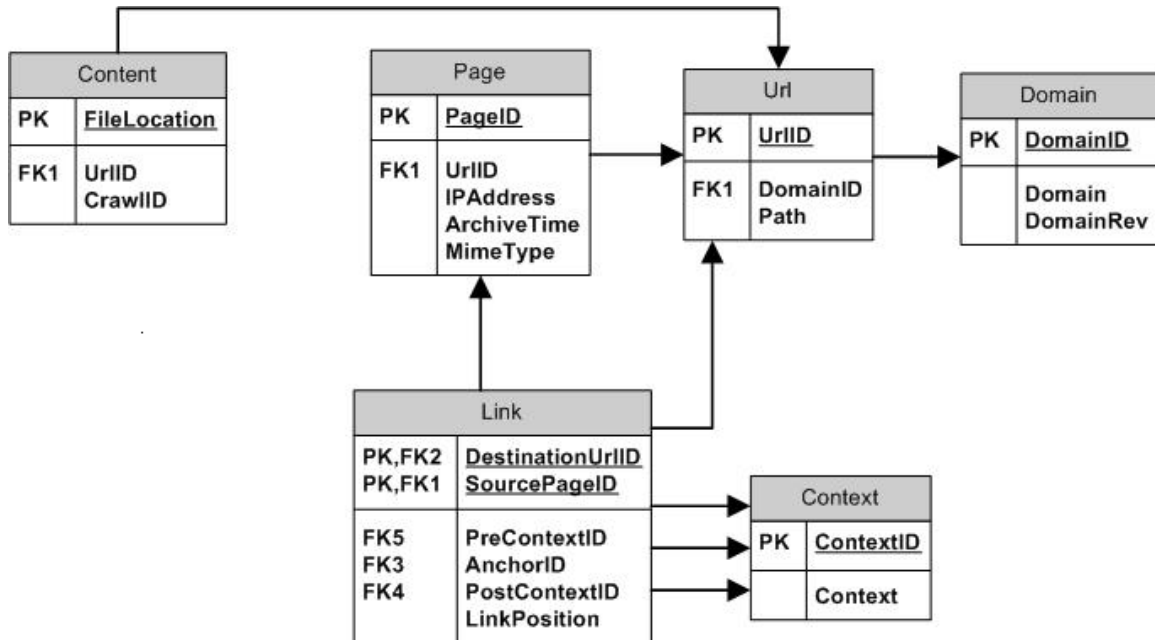
The archive database:

The archive database is used to store the internet archive data in the form of a database for easy querying.

The database has a 2 tier schema:

Web Laboratory Subset Extraction Fall 2005

Each of the crawls has its own set of tables. Diagrammatic representation of the tables is as follows:



Each crawl is identified by the CrawlID and CrawlDate

Scope and Constraints:

For the subset extraction process information retrieval and storage are critical. The solution to data storage is using Relational Database Management Systems (RDBMS).

The factors that influence the subset extraction and storage of the data extracted after retrieval are:

1. The subset schema should be scalable depending on the user queries. The user may query on general and broad topics which may produce a large set of documents.

The Web Laboratory project database size ranges in terabytes and there is constant incoming data from the internet archive. To create a subset of this database requires a system that can support thousands of researchers and provide storage for the subsets. A subset is nothing but a part of the database relation or a combination of relations. It is a replica of information already existing in the database. This replication leads to data redundancy. Data redundancy leads to problems of data inconsistency since any change in the database will have to be reflected in the replicas otherwise

Web Laboratory Subset Extraction Fall 2005

data goes in inconsistent state. Since the data stores is critical and of high importance we need to come up with a model which is scalable and takes care of the problems of data redundancy and inconsistency.

2. Providing users options to store the retrieved data subset on the system. User has the option to use the data retrieved by a particular query for further research and need not run the same query again to retrieve the data.

A user may run a query, which may require hours to run and create a subset since the underlying database is very big in size. It is only reasonable to store the subset for the future use by the user and drop it when user no longer requires it. This saves the processor time and user time and makes the system more efficient.

Also a user may find it tedious to run the same query again every time. The system must be built in such a way so as to take care of the needs of the end users and make the system easy and convenient to use.

3. An efficient mechanism is required for loading and retrieving data records to and from the database in utilizing the data for research purposes.
4. Another challenge in subset extraction is to store the data which spans over different crawls. The pages in the database are stored for different crawls and the same URL maybe stored in several crawl but each copy of the URL will have a different time associated with it.

User queries may be complex and may require information that requires the join of two or more tables, scanning different tables in a single crawl or scanning different crawls and underlying tables for each crawl. These are complex operations and an efficient query language would be required to create a subset which spans over different tables.

5. Some of the key features that a subset should possess are scalability, efficiency and minimum storage requirements. The subset should provide all the functionality that would make the system more user friendly.

Basic Subset Extraction:

- This application is used to allow users to access the database. The users may just specify parameters to search, for example - url, domain, date
- These inputs will then be converted into an SQL query and a subset of the database is created using this query.
- A unique subset id is assigned to each such subset generated and returned to the user. This subset id can uniquely identify the SQL query results to the users.
- The subset is stored for future queries. The user can use this subset id in further research and need not run the query again.

Design features:

- The subset is stored in form of regular views with each subset being assigned a unique subset id.
- The views are virtual tables. They are references to the tables in the database and hence the system is scalable since there is no overhead with data storage.
- Any changes made to the database are reflected in the views and hence this eliminates the problem of data inconsistency.
- The subset information table keeps the record of all subsets generated along with the query used to generate them. The user can specify the subset id and the view will be given to the user.

Query the archive - Subset extraction

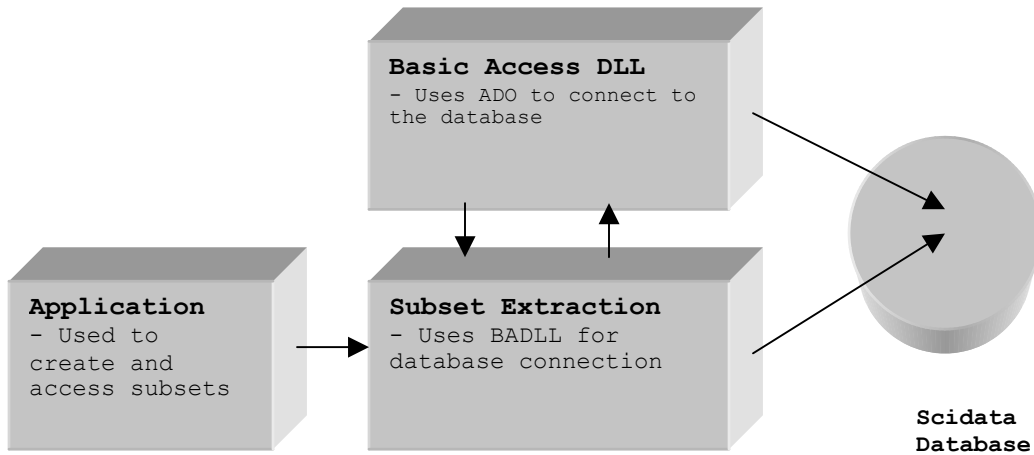
When the user runs queries on the archive the result will be a series of pages that were stored in the archive.

Properties of these pages

- They can span over more than one crawl
- In a single crawl multiple pages with the same URL can be present
- They have the following properties - Page ID, URL ID, IP Address, Arc Time, MIME type

Design of the Subset Extraction Application:

Architecture Diagram:



Application:

Applications are used to allow users to access the database. For the specific task of subset extraction, these applications should

- Allow user to specify parameters
- Convert these parameters into an SQL query
- Return a subset ID to the user that can uniquely identify the SQL query results to the users

MSSQL Database:

The subset extraction logic adds the following table to the database that allows it to manage the subsets

TABLE SUBSET_INFORMATION

Name	Data Type	Description
Subset_ID	Integer	Simple integral count incremented for each new subset
View_Table_Name	Varchar(20)	Name of the view table that corresponds to the subset.
SQL_Query	Varchar(2000)	The user query that was used to generate this subset

Web Laboratory Subset Extraction Fall 2005

Subset ID: The primary key of this table will be the subset ID that is used to uniquely identify them and this will be the parameter that is passed back to the user

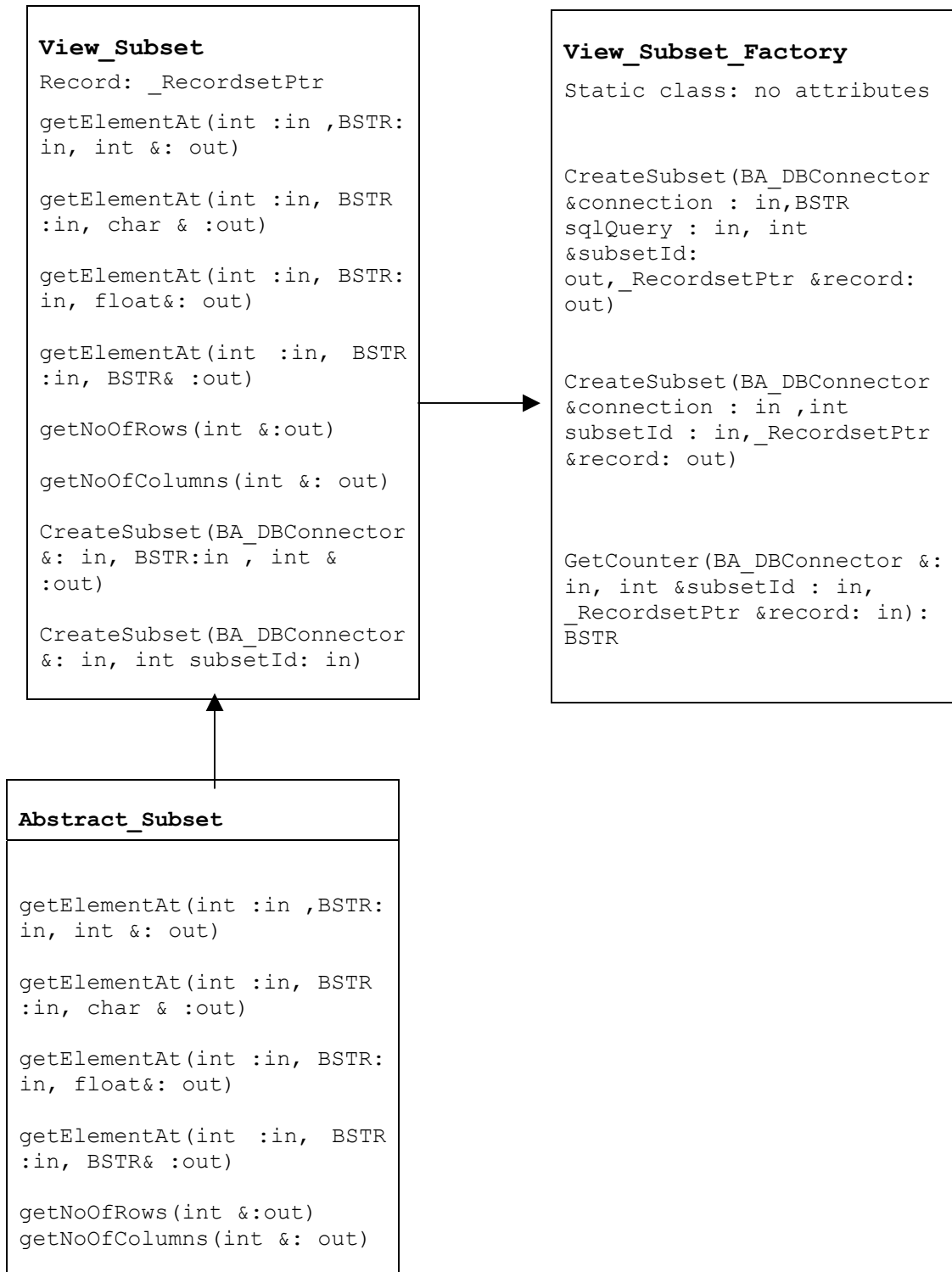
View_Table_Name: If the subset is just a database view, then it will be associated with a view table that is created when the subset is generated. The View_Table_Name will be used to iterate over the data in a subset that is a simple view

SQL_Query: The query that was used to generate the view is also stored so that the view can be recreated when the database crashes or if a view is dropped unintentionally

DLL

The Subset Extraction functionality has been converted to a dll that can be invoked on SciData1 to extract subsets. The DLL has the exact same functionality described in the functions in the design document and the class diagram below.

Class Diagram



Web Laboratory Subset Extraction Fall 2005

For details on the implementation refer the design document for Subset Extraction API on the gforge site.

Web Service

The functions are also exported as web services using ATL programming.

To cross platform and programming language, we wrap Subset Extraction DLL API and export its functions by Subset_WS to give users more alternatives and flexibilities to utilize our service.

Subset_WS is a set of API functions implemented by ATL Server Web Service in Microsoft .Net Framework. Users can invoke Subset_WS API functions in their native programming languages to do basic manipulations through our web service.

For details please refer the design document for Subset Extraction Web Service on the gforge site.

Results:

- This program is able to generate a view subset for the query given by the user
- The user can just provide the subset id and reuse an existing subset by acquiring a pointer to the view.
- It can give the user view details such as number of rows and number of columns
- It can iterate through the subset and access any element at any given position of the subset.
- The subset information is stored for future reference.
- Views are virtual and maintain a reference to the database tables.
- Any changes to the database tables is also reflected in the view hence there is no issue of data inconsistency.
- This application is scalable since there is no storage overhead associated with the storing of subsets.
- The functionality is provided in the form of a DLL for users to use
- The functionality is also exported as a web service that users can remotely use

Future Work:

The future work in this area involves scaling the subset extraction to handle different kinds of subsets for example file subsets. Currently views serve the purpose in an efficient manner but the user may want to save the subset generated on the client machine. We would require the subset to be in form of files. The subset extraction application provides

the basic functionality to the user, in future we would require to increase the functionalities provided to the user to make the interface easy to use and convenient. Provide the functionality to generate web graphs and full text indexing to facilitate search using query words.

References:

- 1] Web Laboratory Project - <https://gforge.cis.cornell.edu/projects/wri/>
- 2] Raghu Ramakrishnan, University of Wisconsin and Johannes Gehrke, Cornell University "Database Management Systems" Third Edition
- 3] Inside Microsoft SQL Server 2000, Delaney, et. al., Microsoft Press, ISBN 0-7356-0998-5
- 4] Mastering SQL Server 2000, Gunderloy, et. al., Sybex Press, ISBN 0-7821-2627-8
- 5] Jim Gray of MS Research and Gerd Heber of the Cornell Fracture Group: <ftp://ftp.research.microsoft.com/pub/tr/TR-2005-49.pdf>

Acknowledgements:

This work is a part of the Web Laboratory, which is a joint project of Cornell University and the Internet Archive. Other members of the team are: William Arms, Blazej Kot, Mindaou and Shantanu. Professor Jayavel Shanmugasundaram in guiding us. This work is funded in part by National Science Foundation grants 0403340, 0127308, and 0537606.

Appendix 1:

Meeting with Professor Jayavel Shanmugasundaram regarding subset extraction:

1. Is using materialized or index views a solution for subset extraction?

No. Because what if a user's query consists of most of the database then the entire database will have to replicated. This would not be a good solution. Indexed views in MSSQL are like materialized views in Oracle. They are not virtual but make a copy of the data required.

2. Would default views be a good solution for storing subsets?

Default views in the current case would be the ideal way of storing the subsets. Default views can be any number since they do not use any space. They just reference the tables in the database.

3. Are the default views reliable?

They are like the database tables and get erased only when they are dropped. Hence they are stored for as long as the user wishes the view to exist.

Updating the views: There is no problem of data inconsistency since the views get updated as soon as the tables they refer to get updated.

Scalability is not an issue in case of default views since they do not contain real data. They are virtual and do not occupy memory space.

4. A user may want to query on the subset after creating the views. Is there a way we could associate certain views with the users who created them?

The view created by a user could have the name starting with the user id. There should be a way to give each user his own namespace and all the subsequent queries could be first searched in the views that already exist.