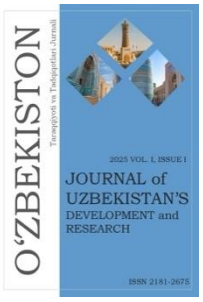


Journal of Uzbekistan's Development and Research (JUDR)

Journal home page: https://ijournal.uz/index.php/judr

THE ROLE OF THE BLOCKING METHOD IN LINKING RECORDS

Ishniyazov Odil[1]

Babajanov Mumin[2]

Alimov Xayriddin[3]

[1,3] *Tashkent University of Information Technologies named after Muhammad al-Khwarizmi*

[2] *International Islamic Academy of Uzbekistan*

KEYWORDS

Blocking method, Search optimization, Data retrieval, Querying, Databases, Block organization, Search queries, Information retrieval, Processor power,Disk writing time, Large databases, Information extraction, Data organization, Efficiency.

ABSTRACT

The blocking method is an efficient search optimization technique used to expedite the process of querying and retrieving data from databases or tables. By organizing the data into blocks, each representing a subset of the database, this method facilitates pre-query filtering of relevant records. Users formulate search queries based on entered information, and the system navigates through blocks to quickly identify and extract pertinent data.

DOI: **10.5281/zenodo.15609061**

The blocking method is a shortened search technique used to identify rows or entries in tables or databases. In this method, the data is divided into blocks, and each block may have specific characteristics based on the rows and columns in tables or the features of a database.

The main purpose of the blocking method is to expedite the process of searching for entries in a database and optimize the search. This method is particularly beneficial for efficiently managing large databases and speeding up the retrieval of information.

The blocking method can be performed in the following steps:

1. Division into blocks: entries in a database are divided into blocks according to established criteria. For example, records in a database may be by name, age, address, position, or other characteristics.

2. Query creation: the user or application automatically creates a query to complete the blocking. Criteria for block identification (e.g. name, age, address) are entered in the survey.

3. Block identification: blocks are identified using a query, which consists of rows and

[1] Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

[2] International Islamic Academy of Uzbekistan

[3] Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

columns in tables (tables) or a database.

4. Identifying linking entries by performing a search: entries within blocks, those corresponding to query criteria are identified. Linking records within blocks provide quick access to search acceleration and data in the database.

5. Return results: when connecting records are found inside blocks, they are returned to the user or application. These records are used according to the result, according to sorting, filtering or other purposes.

The blocking method is a method that is very useful in optimizing the process of creating search queries and obtaining answers. This method involves filtering the database or tables of linking records before searching. This speeds up data detection and tracking processes as well as allowing very efficient use of resources (e.g. processor power, disk write time).

The blocking method speeds up the search by dividing the entries in the database into blocks according to different characteristics. This method is very used when developing optimized queries of databases and finding data quickly.

In general, it is possible to associate with the following types of properties:

1. ID: usually a unique ID will be available for each entry. Based on this ID, numerical values are given. Blocking by ID can help you quickly identify a record.

2. Text or Name: records can be names or texts. Blocking by features of this type can help speed up text-based search for records.

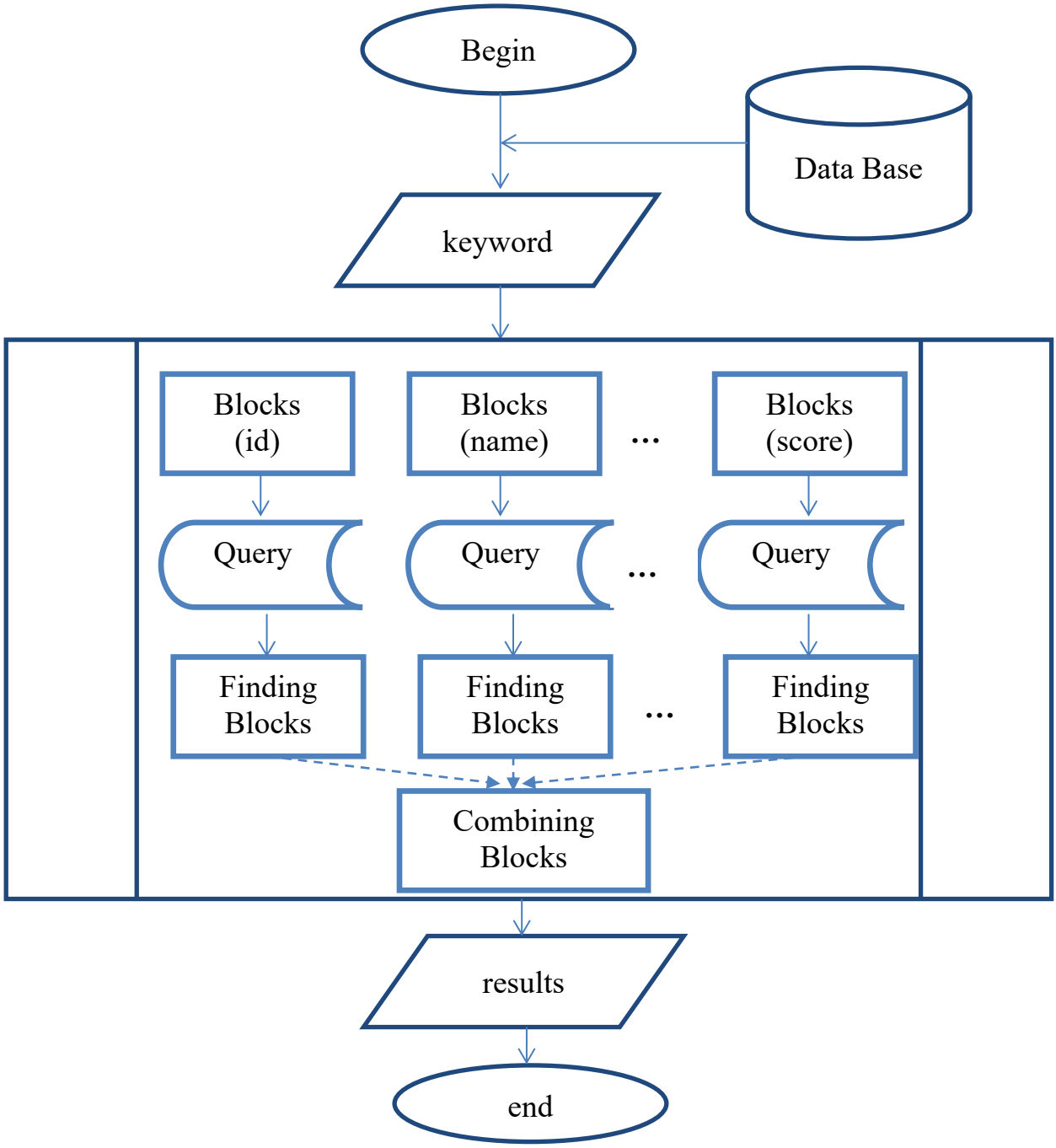
3. Number or Score: if records can get average grades (score), then blocking by numbers will be very useful.

4. Date: can be the date of creation of records. Blocking by these types of features can help you find records that match a specific date.

5. Geographical information: blocking records by Address, City or state, blocking by other geographical features can be used.

6. Categories: records can be based on some category. Category-by-Category blocking can help identify entries belonging to a specified category.

However, depending on the effectiveness and application of the blocking, you should explain what features should largely depend on the data in your database. These characteristics will be related to the structure and purpose of the database.



1-block-scheme. Algorithm of blocking method.

This block-scheme of the algorithm is implemented step by step as follows:

1. Start: the beginning of the algorithm.

2. Data reading: reading data from a database or table.

3. Dividing into blocks: dividing the read data into blocks according to different characteristics (ID, Name, Score, etc).

4. Query creation: create a query to perform a search for a user or application. This includes the necessary information (e.g. ID=2 or Name="Ali")to identify the blocks being linked.

5. Finding blocks: finding blocks that are being linked using a query.

6. Combining blocks: combining found connecting blocks. In this case, it is possible to attach the necessary blocks based on the search terms entered by the user from the query.

7. Output of the result: the output of data in the combined blocks or their use for the purposes derived from them.

8. Algorithm completion: completion of algorithm work.

List of references:

1. N. Adly. Efficient record linkage using a double embedding scheme. In DMIN, pages 274–281, 2009.
2. F. Afrati, A. D. Sarma, D. Menestrina, A. Parameswaran, and J. Ullman. Fuzzy joins using mapreduce. In ICDE, pages 498–509, 2012.
3. A. N. Aizawa and K. Oyama. A fast linkage detection scheme for multi-source information integration. In WIRI, pages 30–39, 2005.
4. A. Allam, S. Skiadopoulos, and P. Kalnis. Improved suffix blocking for record linkage and entity resolution. DKE, 117:98–113, 2018.
5. Y. Altowim, D. V. Kalashnikov, and S. Mehrotra. Progressive approach to relational entity resolution. PVLDB, 7(11):999–1010, 2014.
6. Y. Altowim and S. Mehrotra. Parallel progressive approach to entity resolution using mapreduce. In ICDE, pages 909–920, 2017.
7. A. Arasu, V. Ganti, and R. Kaushik. Efficient exact set-similarity joins. In VLDB, pages 918–929, 2006.