Chapter 4

DESIGN METHODOLOGY FOR VAS QUERY PROCESSING IN MOBILE DATABASE ENVIRONMENT

This chapter discusses how queries are processed. It proposes categorical hierarchy for storing VAS at local servers. It discusses using personalization for prediction of VAS for users.

4.1 Mobile Computing Environment revisited

Mobile computing provides database applications with useful aspects of wireless technology, and a subset of mobile computing that focuses on query to a central database server is referred as mobile databases (12; 30). The inherent characteristics of the mobile environment have a direct impact on data management and present a number of research challenges. Some queries in mobile databases have unique characteristics such as low bandwidth, limited connectivity, usage cost and data availability. These characteristics mainly arise from the aspect of mobility of the user. It increases complexity of query processing in mobile environment.

In general, each mobile user communicates with a Mobile Base Station(MS) in
In order to carry out any activities such as transaction and information retrieval (63), MS has a wireless interface to establish communication with mobile device users and it serves a large number of mobile users in a specific region called cell. In a mobile computing environment, each MS is connected to a fixed network as illustrated in Figure 4.1.

![Mobile Computing Environment Diagram](image)

> Figure 4.1: Mobile Computing Environment

Mobile clients can move between cells while being active and the intercell movement is known as a handoff process. Each client in a cell can connect to the fixed network via wireless radio, wireless Local Area Network (LAN), wireless cellular or satellite. Each of the wireless networks provides a different bandwidth capacity. Driven by need of information at fingertips many queries are pulled (explicitly requested) by the user from the server.

### 4.2 Query Processing in Mobile Database Environment

Query processing in mobile databases may require transmission of data between servers in the network. The data regarding the queries may be distributed in different servers at different locations. But the system must preserve location trans-
Replication and duplication of the data at many servers improves database performance at end-user worksites (64).

Replication technique involves using specialized software that looks for changes in the database. Once the changes have been identified, the replication process makes all the instances of databases look the same (65). The replication process can be very complex and time-consuming depending on the size and number of tuples or records. This process can also require a lot of time and computer resources (39). Duplication basically identifies one database as a master and then duplicates that database. The duplication process is normally done at a set time after hours. This is to ensure that each distributed location has the same data. In the duplication process, changes to the master database only are allowed to maintain consistency.

**Query Processing Environment**

The query processing in mobile environment is depicted in figure 4.2.

![Query Processing System](image)

**Figure 4.2: Query Processing system**

The query processing system for mobile databases is divided into three parts (66), namely

48
Mobile client system relates to how client manipulates and maintains the data in its cache efficiently and effectively. Wireless network system communicates data using broadcasting systems or PUSH based technique. This technique caters to a large number of users in the system. Server system relates to design techniques for the server to accommodate multiple requests so that the request can be processed as efficiently as possible. The query processing for mobile databases is very much centered on the issues of caching, broadcasting and scheduling. The components of the system are:

- **Mobile Client System:** Each mobile client is composed of three modules as shown in 4.2. A Resource Manager which manages the client CPU for handling the query results, a Client Manager which processes the query requests and passes them to the server, models the disconnection operation, and receives and processes the data records or tuples transmitted from the server and a Query Generator which generates the query requests. Client queries are submitted from an mobile unit to the server to be processed and a message (messages) containing the tuples that form the answer to the query is (are) transmitted back to the mobile unit. The messages containing the tuples are processed by the mobile unit and the tuples are displayed on the screen of the mobile unit accordingly.

Mobile client system defines a number of strategies to maintain cached data items in clients local storage. Basically, wireless communication channel in general suffers due to narrow bandwidth while it is also inherently asymmetric communication, that is; the downstream communication bandwidth from a server to the mobile client is much larger than the upstream communication bandwidth from clients back to server. Due to the above reason, caching of frequently accessed data items in a clients local storage becomes important for enhancing the performance and data availability of data access queries. Another advantage of caching is the ability to handle fault tolerance. This is related to the characteristics of mobile computing in which each mobile client connects or disconnects from the network frequently. In some situation, the base station may not be accessible due to problem like...
signal distortion. However, a query can still be partially processed from caches and at least some of the query results can be returned to the user. (43)

- **Wireless network System:** The Wireless Network System component assumes that all messages are of equal priority that will be served on a First-Come First-Served (FCFS) basis with a service rate of Network Bandwidth. When the Wireless Network Manager finds out (i.e., while sending a message to a mobile unit) that a mobile unit is disconnected, it informs the Server Manager about the disconnection so that the transmission of the tuples to the mobile unit can be paused until the mobile unit reconnects to the network. In this scheme, the server also periodically broadcasts the frequently accessed data items to clients through one or more broadcast channels, but the clients may not cache the data item of interest. This situation might occur when the client does not have enough memory or space to cache the required data.

- **Server System:** In case of pull-based broadcast scheduling where mobile clients send queries to the server through wireless network system, server processes the query and sends the result back to the client. The server system has three subsystems: Resource Manager which manages the server CPU time for query and update processing, an Update Generator which generates update requests, and a Server Manager which coordinates the query requests from mobile units and update requests from the Update Generator.

When a Client Query (CQ) is issued by a mobile unit, it is processed by the Server Manager and the set of tuples satisfying the query are determined. The Server Manager also decides when and which tuples should be transmitted (8). The strategy concerns with broadcast and disk scheduling. Broadcast scheduling is used to cater to many users at a time. It determines how queries are to be served efficiently in the server considering a number of factors such as the length of the query, the wait time and the popularity of the items. A database server is able to enhance the data retrieval performance by incorporating its own main memory and cache to store those database items frequently accessed by most mobile clients. A query can be processed either in the disk server or cache server. If the
relevant data items have been retrieved earlier then the query is processed in the cache server. After processing a query, the results are transmitted to the transmitter queue, which subsequently sends the data items through the wireless channel.

In query processing, the database users generally specify what data is required rather than specifying the procedure to retrieve the required data. Thus, an important aspect of query processing is query optimization. Now in query optimization, the optimizer uses heuristics to optimize execution of the queries. Fragmentation of database and strategies used for indexing affect the performance of distributed queries. The related parts of databases known as relations may be fragmented and/or replicated and considering many sites to access, query response time may become very high (108).

A query is expressed by using a high-level language such as SQL (Structured Query Language) in relational data model. The main function of a relational query processor is to transform a high-level query into an equivalent lower-level query (relational algebra) and the transformation must achieve both correctness and efficiency. Execution strategy for the given query is implemented by lower-level query. Since data is geographically distributed in mobile database system, the processing of a distributed query is composed of the following three phases:

1. local processing phase
2. reduction phase
3. final processing phase

The local processing phase basically involves local processing such as selections and projections. Selection is a database operation which selects a set of records based on certain conditions specified by the user.

Example of selection operation in SQL is

Select * from table customer where address='Mumbai'.

This SQL statement will retrieve all records stored in customer table (file) where customer address is Mumbai.

Projection operation on a database will retrieve certain attributes from a record or records on the conditions specified by the user. It is a subset of set of attributes of file(table) in database.
Example of project operation in SQL is

```
SELECT empno, ename, dno, job from Emp WHERE job='CLERK';
```

The reduction phase uses a sequence of reducers (i.e., semi joins and joins) to reduce the size of relations. A semi-join between two tables returns rows (records) from the first table where one or more matches are found in the second table (67; 68). The difference between a semi-join and a conventional join is that rows in the first table will be returned at most once. Even if the second table contains two matches for a row in the first table, only one copy of the row will be returned. Semi-joins are written using the EXISTS or IN constructs in SQL. Suppose there exists the DEPT and EMP tables in the database and a list of departments with at least one employee is required. This query can be written with a conventional join:

```
SELECT D.deptno, D.dname FROM dept D, emp E WHERE E.deptno = D.deptno ORDER BY D.deptno.
```

The final processing phase sends all resulting relations to the assembly site where the final result of the query is constructed.

A straightforward approach of processing a distributed mobile query would involve sending all relations directly to the assembly site, where all joins are performed. The allocation of the data influences the performance of the distributed systems given by the processing time and overall costs required for applications running in the network. In distributed mobile query processing, partitioning a relation into fragments, union of the fragments to form a whole relation and transferring a relation fragment from one database to another database are common operations.

### 4.3 Cost Model of Query Processing:

Mobile databases incur additional of processing expenses, because of the hardware and software to handle the distribution. These expenses present as the cost of data transfer over the network. Data transferred could be intermediate files resulting from local sites, or final results need to be sent back to the original site that issued the query. Therefore, database designers are concerned about query optimization, which target minimizing the cost of transferring data across the net-
work. Criteria for measuring the cost of a query evaluation strategy for centralized DBMSs number of disk accesses (# blocks read / written) and for distributed mobile databases is the additional cost of data transmission over the network. Potential gain in performance can be achieved from having several sites processing parts of the query in parallel. Join queries in distributed database are shipped whole or fetched as needed. Main considerations of query processing in mobile databases are:

- Communication cost: Cost of establishing communication between network.
- If there is several copies of a relation, decision regarding which copy to use.
- Amount of data being shipped.
- Relative processing speed at each site.
- Site selection: Closer the site to the originator of query; lower is the transmission time.

Cost of processing a query includes the communication cost, I/O cost and CPU cost. The total communication cost is given by

\[ TC = C_0 + x \cdot C_1 \]  \hspace{1cm} (4.1)

TC is total communication cost. 
\( C_0 \): is the cost of initial set up between servers. 
\( C_1 \): transmission cost of unit 
\( x \): amount of data transmitted (in bits)

Response Time is computed as an elapsed time from the starting to completion of query.

### 4.4 Proposed Query Processing Technique

Many times it is observed that certain data are more frequently queried. Stocks, weather information, location based queries such as nearest restaurants, shops etc are often in demand. With huge leap in information processing technologies such as GPS
and Maps; it is possible to pinpoint users current location and send relative information. Caching improves availability of frequently accessed items at mobile unit can reduce the time taken to access items from server. This is beneficial for limited bandwidth and intermittent connections to the network. But cached items may not always reflect current data. During weak connections the cached items can be used and during strong connectivity current values can be requested. Replication of entire database is not feasible at the mobile unit end due to limited memory. Hence to improve data availability summarized VAS information can be stored at Mobile Unit.

VAS data is categorized in a hierarchical manner to model location-dependent data. Concept hierarchies are used generate categories of VAS. This hierarchy defines relationship that generalizes lower level data to high layer information. For instance, to generate domain knowledge of cities that belongs to a state. Domain experts may be consulted to ensure the validity and completeness of the hierarchy. Figure 4.3 depicts location-based concept hierarchy. The proposed hierarchy is designed as distributed directories to provide assistance to find values in the database according to the reference location of a query. (70; 71)

![Figure 4.3: Hierarchy of Location attribute](image-url)
Concept hierarchies are constructed during data definition of database. The attributes of all relations and their functional dependencies are determined to ensure correctness and completeness. The hierarchies of domain values do not change even when new relations are appended to database. Only updation of domain affects the concept hierarchies. This updated hierarchy needs to be transmitted to Mobile Stations and Mobile Unit. This saves much of wireless transmission cost.

Classifications of Databases and Query Responses

The database can be classified according to:

Classification by Correctness

Databases are classified based on the correctness of information they contain with respect to the main database (72). Consider a database S that stores some part (or summary) of a larger database T and assume that both S and T are relational databases. Consider database T as the main database and S as its approximation (also known as summary or condensed database). Some formal definitions are as follows: (72)

**Definition:** S is complete with respect to T if S contains T. That is, all information stored in S includes all the information that is contained in T.

**Definition:** S is sound with respect to T if S is contained in T. That is, the stored information in S includes only true information that stored in T.

**Definition:** S is imprecise with respect to T when it contains an approximation with respect to the values held in T.

For example, S may contain a set of possible values, with the real value (held in T) being one of the elements of this set. Imprecise information is not erroneous and does not necessarily compromise the integrity of an information system.

In terms of the responses to queries:

**Definition:** S is complete with respect to T if a query on S will return at least the data that the same query would return on T.

**Definition:** S is sound with respect to T if a query on S will return no additional data than would the same query on T.
Definition: S is imprecise with respect to T if a query on S contains values that are either numerically approximate or conceptually broader.

Definition: A summary or condensed database S is a special form of information repository in which data are stored in a condensed or summarised form. Summary databases are generally incomplete but sound and may be imprecise.

### 4.4.1 Categorized Query Processing (CQP)

The query processor can adopt three approaches:

1) Query processor at Mobile Unit should be able to distinguish those queries that are more appropriate to categorized databases or direct the queries to server accordingly. This approach requires additional techniques for determining the query type.

2) The query executes simultaneously on both (i.e. summarized data or the main database). The first response is returned to the user and second is discarded. This mechanism may result in different data set. It is useful when results are required in a time bound manner.

3) Query processor can fragment the query and run result in parallel on both the databases.

**Query processing Steps**

1. When a query arrives at mobile unit, the query processor first decides whether to answer the query using the main database in server or using categorized database at mobile. For example VAS such as latest stock information retrieve queries from server whereas Temperature today in Mumbai can be executed using summarized database at mobile unit.

2. Categorized hierarchies are available to query processor at mobile unit or can be obtained by sending a request for particular type of hierarchy associated with the particular data depending on query.

3. Rewrite the SQL query using concept hierarchy.
Example: Consider a relational table MUSIC which stores data regarding musicians and their respective genres with specializations.

![Figure 4.4: MUSIC-data relation](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Genre</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meera</td>
<td>Western</td>
<td>hip-hop</td>
</tr>
<tr>
<td>Anita</td>
<td>Indian</td>
<td>soft rock</td>
</tr>
<tr>
<td>Madhura</td>
<td>Indian</td>
<td>classical</td>
</tr>
<tr>
<td>Amit</td>
<td>western</td>
<td>jazz</td>
</tr>
<tr>
<td>Amogh</td>
<td>Instrumental</td>
<td>carnatic</td>
</tr>
<tr>
<td>Milind</td>
<td>Instrumental</td>
<td>classical</td>
</tr>
</tbody>
</table>

![Figure 4.5: Hierarchy in MUSIC](image)

Figure 4.5 shows the categorized hierarchy of Music relation in database.

figure4.6 shows the condensed or summarized table. Here the "type" attribute from relation music is summarized as shown earlier in figure 4.4.

![Figure 4.6: Condensed relation MUSIC](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Genre</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meera</td>
<td>Western</td>
</tr>
<tr>
<td>Anita</td>
<td>Indian</td>
</tr>
<tr>
<td>Madhura</td>
<td>Indian</td>
</tr>
<tr>
<td>Amit</td>
<td>western</td>
</tr>
<tr>
<td>Amogh</td>
<td>Instrumental</td>
</tr>
<tr>
<td>Milind</td>
<td>Instrumental</td>
</tr>
</tbody>
</table>

Example: Suppose a user poses a query to Find the names of the musicians.
associated with hip hop”. This in SQL will be:

```
SELECT NAME From MUSIC-DATA WHERE genre = Western;
```

This query can be answered using the summarized MUSIC-data relation if it exists in Mobile Unit. The result of the rephrased query will return not only the names of musicians associated with hip hop but all the musicians associated with western genre. Such a result is may be acceptable and the query process is knowingly over-complete, imprecise but a complete approximation of the results returned by the original query using Musician data. The concept hierarchies involving the attributes of are to be used while rephrasing the query.

**Updating Summary and Main Databases**

Approaches that can be adopted when updating the main database at Mobile Station and condensed database at mobile unit are:

1. Update categorized database at mobile unit and main databases at MS at sametime. This takes longer time and may not be feasible in case of mobile environment. Here queries can return results that are sound but may not be complete.

2. Update categorized database at mobile unit and main databases at MS at specific connection times (strongly connected network). Categorized databases are updated and regenerated.

The categorized query processing (CQP) recognizes the meaningful words in a query using knowledge by semantic analysis (words such as near, closest, latest, cheap). Using Location hierarchy and categorizing related data can achieve faster query processing.

Location Heirarchy contains 3 categorized files:

- **Point Of Interest data file (POI):** It represents main buildings, roads and prominent landmarks over the region with specific position information.

- **City name data file:** It maintains the name of all cities and towns in a country. This file is used as knowledge when the query processing system extracts
useful words such as city name and town name from a given query.

- Domain specific data: This file stores data that is not location specific example stocks, weather, and cricket score.

In order to classify POI data, LoDT (List of Destination Type) data is used as a classification standard. This file is industry standard and updates itself frequently depending on service provider. LoDT file includes representative set list of interest-point, location and purpose of the POI. For example, there is "restaurant" element in the LoDT file which is representative set of numerous restaurants which includes the word 'restaurant' in the name of interest-point or purpose of the POI.

The categorization module is shown in figure 4.7

![Figure 4.7: Categorization module](image)
4.4.2 Cost Analysis

The cost-benefit analysis of query-processing model of condensed databases created using concept hierarchy is discussed here. The model has additional associated cost of summarizing a database, which includes the cost of constructing concept hierarchies and query rewriting cost.

**Size Reduction Ratio and Storage Cost:**

The storage cost associated with storing a relation, summarized relation and the associated hierarchies.

Let 'R' be the original data relational table with 'n' number of attributes and 'm' number of tuples.

Let 'S' be the condensed table

- p : number of attributes
- q : number of tuples where p < n and q < m.

Total size reduction ratio (SR) is the ratio of the categorized table verses the actual size of the table. Let r be the number of tuples reduced and k be the number of attributes reduced while summarizing.

\[
SR = \frac{(m-r)(n-k)}{mn} = \frac{pq}{nm}. \quad (4.2)
\]

An SR value of 0.7 indicates 30% saving in comparison to original table. This ratio will always be less than or equal to 1.

**Storage cost:** It is the cost of storing a relation in memory. It will be equal to the number of memory blocks needed. Let

- u: number of bytes stored per memory block.
- b: the average number of bytes needed to store an attribute value.
- \( c_{\text{blocks}} \) be the cost of storing a memory block.
Let a condensed table result in a reduction of \( t_{dup} \) number of tuples.

The number of memory blocks (\( M \)) needed to store a condensed categorized table will be:

\[
M = p \times (q - t_{dup}) \times b/u. \tag{4.3}
\]

The storage cost for a categorized table \( SCOST \) will be:

\[
SCOST = (p \times (q - t_{dup}) \times b/u) \times c_{blocks} \tag{4.4}
\]

Reducing a relation is associated with additional cost of storing concept hierarchies \( SCOST_{ch} \) can be determined as follows:

Let

- \( CH_n \) : number of concept hierarchies per table.
- \( b \) : average size in bytes to store concept hierarchy
- \( B_{con} \) : Number of Concept hierarchies
- \( u \) : number of bytes stored per memory block.

\[
SCOST_{ch} = (CH_n \times b \times B_{con}/u) \times c_{blocks} \tag{4.5}
\]

Storage cost will be less if the number of concept hierarchy constructed will be less.

The storage cost for the original table \( SCOST_{org} \) will be =

\[
SCOST_{org} = (n \times m \times b/u) \times c_{blocks} \tag{4.6}
\]

Net benefit is

\[
SCOST_{org} - (SCOST + SCOST_{ch}) = (n \times m \times b/u) \times c_{blocks}
\]

\[
((p \times (q - t_{dup}) \times b/u) \times c_{blocks} \\
+ (CH_n \times b \times B_{con}/u) \times c_{blocks})
\]

\[
= (b \times c_{blocks})/u(n \times m(p \times (q - t_{dup}) + CH_n \times B_{con})) \tag{4.7}
\]

If \( n \) (number of attributes) and \( m \) (number of tuples) are large then the prob-
ability of \( p \) (number of attributes in condensed table) and \( q \) (number of tuples in condensed table) being smaller and \( t_{dup} \) being larger will be more. Thus there are chances of categorized table being smaller which gives more cost benefit. The size and number of concept hierarchies associated with a relation depends on the domains associated with its attributes and how closely are they related. There has to be a tradeoff between storing a complete table and storing a condensed table along with the concepts hierarchy involved.

The concept hierarchies do not change with increase in number of rows. Therefore these hierarchies can be stored at the mobile unit which are relevant to the user.

**Transmission cost:**
Cost of transmitting a relation (R) to user from server is as follows:

\[
m: \text{number of tuples} \quad n: \text{number of attributes} \quad b: \text{fixed number of bytes}.
\]

Total number of bytes in a relation

\[
T_b = m \times n \times b.
\] (4.8)

Let \( b_{dist} \) be the number of uniques values in a relation and also in a concept. Then the total number of distinct values (DV) in a relation will be

\[
DV = m \times n \times b_{dist}
\] (4.9)

Let Smax be the maximum size of a message. The total number of messages (TM) will be:

\[
TM = m \times n \times b_{dist} / Smax
\] (4.10)

Let \( c \) be the cost of transmitting each distinct byte values from server to user and \( C_{mc} \) be the message cost per message.

The total cost of transmitting a relation (without categorizing) \( TC_{NC} \) is

\[
TC_{NC} = m \times n \times b_{dist} \times c + (m \times n \times b_{dist} / Smax) \times C_{mc}
\] (4.11)
Suppose after the categorization the relation has $p$ number of tuples and $q$ number of attributes. The total transmission cost of a summarized table will be

$$TC_c = p * q * b_{dist} * c + (p * q * b_{dist}/S_{max}) * C_{mc} \quad (4.12)$$

Total saving in transmission cost will be:

$$TCa - TCb = \{m * n * b_{dist} * c + (m * n * b_{dist}/S_{max}) * C_{mc} - (p * q * b_{dist}/L_{max}) * C_{mc}\}$$

$$= (m * n - p * q) * b_{dist} * c + (m * np * q) * (b_{dist}/S_{max}) * C_{mc}$$

$$= (m*n - p*q) * b_{dist} (c + C_{mc}/S_{max}) \quad (4.13)$$

The total saving in transmission cost will be more if the number of bytes transferred from server to user are less. The number of bytes will be less if $m$ and $n$ are large and $p$ and $q$ being small. The transmission cost may also include the cost of transmitting concept hierarchies associated with the condensed table.

The transmission cost for the transmission of concepts associated with a categorized table will be as follows:

The total cost of transmitting $CH_n$ number of concept hierarchies is:

$$CH_n = (CH_n * b_{dist} * B_{con}) * c + (CH_n * b_{dist} * B_{con}/S_{max}) * C_{mc} \quad (4.14)$$

This cost will be minimum provided the size of the domain for each attribute is small as it will restrict the number of concepts associate to it.

**Query Processing Cost**

63
Consider query involving only one table and one project operation. Let the query access \( z \) number of attribute values while answering the query using categorized table and \( y \) number of attribute values when using original table. After categorization \( h < H \). Let \( c \) be the cost associated with each access of an attribute value.

We have query processing cost ratio (QCR) given by

\[
QCR = \frac{z \cdot c}{y \cdot c} = \frac{z}{y}.
\]  

(4.15)

when \( z = y \) answering a query using categorized table is more costly then answering a query using the original table. Thus, answering a query using categorized table is more cost effective only if \( y \) is large. Query processing cost will includes data transmission as well as message transmission cost if the query has to be evaluated at server. Thus, it is cost effective to store the categorized table at mobile unit provided memory is available.

Cost limitations to categorizing using hierarchies:

Apart from storage cost, transmission cost and query processing cost, there are some additional costs associated with categorizing using concept heirarchy in a relation are as follows:

1. It involves the cost of projecting out some attributes.
2. It also includes cost for eliminating possible duplicates
3. Cost of rewriting a query.
4. Cost of generating concept hierarchies.

### 4.5 Experimental Setup and Testing Environment

The simulation was done using MySQL on windows 2000 platform. The pattern matching algorithm used for extraction is using Boyer-Moore Pattern Matching Algorithm and K-Means Clustering algorithm. Boyer Moore pattern match-
ing algorithm has been the typical benchmark in the field of string searching techniques (73). It is an especially effective pattern matching and string searching technique. The execution time of the Boyer Moore algorithm lesser than numerous other algorithms of the same genre, as it doesn’t have to survey each character of the string to be looked, however rather skips over some of them. The algorithm quickens as the key being looked for extends. The way that with every unsuccessful endeavour to match the string with the content, it utilizes the data picked up from that endeavour to skip out whatever number positions as could be allowed where the string doesn’t match, making it very efficient pattern matching algorithm. Boyer-Moore algorithm holds a window containing pattern over text much as the navel string matching algorithm does (90). Queries fired by 15 users were selected at random. System simulated 250 queries. Categories considered are movies, weather, restaurants, shops and stocks. The number of key descriptors considered were 6. Number of locations considered for simulation was 10. The query set was run on the system where the categories were not formed. The database performed sequential search in this case. The same set of queries were fired now on categorized database and the response time was measured as shown in figure 4.8.

![Figure 4.8: Response times using categorized and non categorized techniques](image)

Figure 4.8: Response times using categorized and non categorized techniques
4.6 Personalization and Prediction of User Queries for VAS

This section focuses on technique for pull based queries that is queries explicitly put forth by the users. User profiles are used to determine the probable data that the user may require. Many previous works used these techniques by using various data mining algorithms (12; 30; 74; 75; 34). In this work the novelty is incorporation of feedback mechanism to improvise predictability.

Personalization is a mechanism to allow a user to adapt, or produce, a service to fit users personal needs, and that after such personalization, all subsequent services rendered by this service towards user is changed accordingly (27; 76; 77; 78).

User profile is one of the important factors for successful content personalization. Nonetheless, a solution is required to adequately meet the user’s needs when the required information has not been categorized appropriately according to their profile and contextual information. One possible solution is building the user profile. However, to construct user profile without prior knowledge about the user is difficult. Data mining algorithms such as clustering algorithm can often be applied to lighten the complexity by classifying the users into homogenous classes that exhibit similar behavior. The clusters of users posing similar queries is identified and classified according to various categories.

In order to predict the type of user, the users preferences and ranking of services need to be analyzed (76). Data mining approaches have been used to analyze the collected data on users behavior and usage pattern in order to determine specific user groups and deliver the recommended items according to users needs. A personalization scheme using user profiles for predicting the user requirements is developed. The issues regarding user profiles are:

- **User Modeling:** Mobile User Behavior Modeling is based on task-oriented model using ontology.

- **User Profile with Demographic Factors to Perform Personalization Task:**
  Personalization should include important component like user profile in order to make assumption that user can belong to a specific group. Age, gender, ethnicity and socio-economic status showed that there are differences in the behavior
of using the wireless device. Users may rate different items differently based on their preferences and different influential factors. Moreover, the demographic factors such as gender, age, income and the types of mobile devices can also influence the ranking of services. The importance and up-to-date information including context information such as time to acquire the information also affected the browsing behavior of each content item on mobile device.

This system used relative ranking and Naive Bayes theorem to predict the users requirement. The Naive Bayes Theorem indicates how the probability that a theory is true is affected by a new piece of evidence. It has been used in a wide variety of contexts, ranging from marine biology to the development of "Bayesian" spam blockers for email systems. The theorem is given as

$$p(c_j|d) = \frac{p(d|c_j)p(c_j)}{p(d)}$$

where

- $p(c_j|d) = \text{probability of instance } d \text{ being in class } c_j$
- $p(d|c_j) = \text{probability of generating instance } d \text{ given class } c_j$
- $p(c_j) = \text{probability of occurrence of class } c_j$
- $p(d) = \text{probability of instance } d \text{ occurring}$

A detailed centralized source database holds the data on the subscribed users and their respective profiles. This database is maintained at the server end of the service provider. The user profiles are subjected to modules of prioritization, followed by summarization, thereby creating a summarized or condensed database. This database containing the most likely useful information of a particular user will be sent when user is either weakly connected or will be possibly disconnected. This summarized database can be located at the Mobile Unit or Mobile Service Station (MSS) thereby enhancing the speed of requests or alerts. A Relative Ranking Algorithm is designed to predict and prioritize the VAS most relevant to the subscriber. The algorithm learns by assessing the users location, logs, profile and feedback over a period of time, thereby trying to take the profile to be near accurate. An application at the users side allows the users to launch queries for MVAS, which are logged as a part of the users profile. The design is shown in figure 4.10.
The model aims at personalizing preferential MVAS for each consumer subscribed to this service. The model uses a rank algorithm to rank various services with respect to the consumer. The initial training dataset is based on a survey conducted. The survey was local to the Mumbai city, India. The participants involved belonged to a wide range of age groups and socio-economic background. Using this survey influential parameters for the generation of user profile were determined. These parameters were ranked according to their importance (27; 76). This dataset forms the basis for classification and association rules of the model. Accuracy of the model was 87 percent and 78 percent when gender and age was used as parameter for prediction (27).

But the most important limitation of personalization is accurately adapting and predicting user needs [42]. Given the plethora of services nowadays prediction has become challenging. Users are interested in newer things and therefore personalization and prediction will always never be accurate. Also the magnitude of data processing and preprocessing is enormous since it has to cater to individual users. It is therefore suitable to personalize profiles for long term loyal users. Also with advent of apps (downloadable VAS), Pull based queries have reduced response
times. This is because the apps update themselves as background processes.

4.7 Limitations of Personalization and Prediction

The Personalization technique requires user profiling which is a difficult task. It raises issues of privacy and accuracy of data. Moreover, the user profiles have to be referred to for user requests. This adds to the cost of storage and updates too. The user is interested in new services. The system sends data which the user may not even require depending on the profile generated.

Summary

This chapter discussed the query processing system in mobile databases. It described various query types in the mobile environment. It proposes a level and categorized technique for processing queries regarding VAS. The response time depends on the semantic extraction of the query. We have seen during the tests that generally three (3 gram) keywords are sufficient to get the results in acceptable time frames. In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus. n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" (or, less commonly, a "digram"); size 3 is a "trigram" (80). Larger sizes are sometimes referred to by the value of n, e.g. "four-gram", "five-gram". More number of keywords can result in overfitting and may not even return approximate results. In over fitting the system matches all words exactly and hence may not return any result.

Personalization proves to be inadequate in terms of accuracy and storage of user profiles and learning of each user.

The next chapter proposes a scheme to organize the data related to MVAS in such a way that it is optimized to cater overall global user requirements instead of personal requirements.