Selecting an optimal set of secondary indices

Theo Härder, Technische Hochschule Darmstadt

1. Introduction

An important design problem concerning the effectiveness of data base systems is the selection of access paths (secondary indices) supporting the fast access to sets of records qualified by transactions. However, the exploitation of access paths is useful only when they save processing time compared with the serial scan of the total data base. Consequently, an access path must essentially reduce the response time of a query, because it involves additional expenses with respect to storage and maintenance.

In a data base system the global optimization of the selection problem regarding all transaction types and their frequencies captured for all heterogeneous applications is necessary for economic reason. The indexing problem can be stated in the following way:
Invert a subset of attributes such that the total expected costs resulting from all transactions are minimized or, conversely, the gain is maximized with respect to the serial scan of all queries. The index optimization can be subject to the additional constraint that in case of storage restrictions only a limited number of secondary indices is available.

A number of studies have appeared in reference literature considering this problem. Lum has taken an empirical approach (Lu71), while others (Ki74, St74, Sk74) use stochastic models concerning probabilities for retrieval and update to obtain analytic solutions. In our refined model there are some additional means to describe transaction types and different kinds of processing. Essentially, the following characteristics and parameters allow a detailed description when considering design questions:

- query, update, insert and delete operations,
- statistical queries without access to the records,
- access time as a function of a transaction's hit ratio,
- random processing according to random and random-sorted lists,
- sequential processing in case of high hit ratios,
- storage costs for access paths,
- storage characteristics of real devices.

2. Model assumptions

Let $A_1, A_2, \ldots, A_n$ be nonempty sets, not necessarily distinct. A subset R of the Cartesian product $A_1 \times A_2 \times \ldots \times A_n$ is called a relation, and A_i is designated a domain or attribute of R. Let r be an element of R, then r is a n-tuple $(r_1, r_2, \ldots, r_n)$ where r_i belongs to A_i. The attribute or minimal group of attributes which guarantees the uniqueness of tuples is called primary key. The physical occurrences of a relation and of a tuple are denoted respectively as file and record.

We assume that the N_{REC} records of the file are randomly stored in secondary memory and are not clustered according to any criteria. The records are uniformly distributed over neighbouring cylinders of a disk, and access to them is performed with equal probability for all transactions. The assumption of clustered records (Ro74) is not useful when dealing with arbitrary transactions.

The selection problem of access paths is discussed concerning one relation. In case of independence between relations the results can be directly extended to a multi-relational data base.

We distinguish two classes of access paths. Obtaining access via primary key one record is found at the most. In comparison to this a secondary key qualifies n records. The corresponding access path is called secondary index. A primary key access path is necessary for an efficient support of batch processing. The following three operation primitives are important for its implementation:

- random access to any record,
- sequential processing of all records in a particular order,
- efficient maintenance.

To support the processing of ad hoc queries secondary indices are suitable. An index on the i-th attribute of R is a mapping from elements in A_i (attribute values) into those tuples in R for which that element in the i-th attribute occurs, i.e., a mapping $I_i : A_i \rightarrow 2^R$ (Ki74). We assume the existence of atoms for the domains (Wo71) so that each attribute has a finite set of elements. The attribute A_i has j_i attribute values with the frequencies $N_1, N_2, \ldots, N_{j_i}$, where

$$\sum_{k=1}^{j_i} N_k = N_{REC} \quad \text{is valid.}$$

To describe the distribution we use the resolution factor (Wa73) $Rf_i = N_{REC}/j_i$ as the expected value, which is justified under the consideration of various transaction types.

A two level hierarchical organization for the index consisting of the catalogue for the attribute values and the corresponding target lists is considered. The catalogue entries (attribute values) contain the actual frequencies N_{ik} for statistical information purposes, besides the address of the target list. The catalogue of A_i is organized as a B-tree implemented in physical blocks of length B in secondary storage. In order to find the target list of a particular value of A_i a B-tree of height h must be traversed. Assuming the length of an attribut value entry to be D ($D \approx B/100$) the number of necessary accesses to locate the target list is

$$h_i = \begin{cases} 1 \text{ for } & j_i \leq B/D \approx 10^2 \\ 2 \text{ for } B/D & < j_i \leq (B/D)^2 \approx 10^4 \\ 3 \text{ for } (B/D)^2 & < j_i \leq (B/D)^3 \approx 10^6 \end{cases}$$

The target list itself can contain primary keys, tids or record addresses in case of a vector implementation or logical or physical marks when packed boolean arrays are taken. The storage requirement for one entry or mark be S_M. Then a target list is consequently stored in

$$N_{Bi} = \left\lceil \frac{N_{REC} \cdot S_M}{j_i \cdot B} \right\rceil$$

physical blocks.

In order to express the storage costs C_{ST} of an index we introduce a cost factor c which is related to storage unit and time. The dimension of c should be chosen such that storage costs and access costs are comparable. Neglecting the storage space for catalogue information the costs for the inversion of one attribute are

$$C_{ST} = c \cdot S_M \cdot N_{REC},$$

since N_{REC} target entries exist for each attribute in normalized relations.

In our model four different types of transactions are considered. First of all we restrict ourselves to read and update operations which extend only to one attribute. Hence a "decoupled" transaction model follows in which all attributes can be analysed separately.

In a retrieval transaction Q_{Ri} all records with a given attribute value for the i-th attribute are selected. All qualified records must be looked up and placed to the user's disposal. The probability of this transaction type be p_i. In many applications only statistical information is retrieved which refers to the frequency of certain elements of a data base. The search for the actual records is not necessary if frequency information is contained in the catalogue or the target list of an index. Assuming the probability s_i of that query type, the probability r_i of a retrieval transaction for the i-th attribute follows

$$p_i + s_i = r_i.$$

The total retrieval probability in a normalized relation with n attributes as candidates for inversion is given by

$$\sum_{i=1}^{n} (p_i + s_i) = \sum_{i=1}^{n} r_i = R$$

Three different types of maintenance operations are distinguished:

- update of an existing record,
- deletion of an existing record,
- insertion of a new record.

In all three transaction types it is assumed that the operation is restricted to one record which is identified by primary key. The update of a record requires the search via the access path of the primary key and the rewrite of the updated record. Insertions and deletions are performed via primary key. With respect to our model only the overhead caused by a secondary index is relevant.

In an update transaction Q_{Ui} the value of the i-th attribute of a record is changed. In case of inversion the maintenance of the index requires reading and rewriting of two target lists and catalogue entries. Assuming an update probability u_i for the i-th attribute the total probability of an update transaction is given by

$$\sum_{i=1}^{n} u_i = U$$

In an insertion or deletion transaction Q_I resp. Q_D the storage address of the record is found via the access path of the primary key and the corresponding operation is performed. For all inverted attributes the corresponding target lists and catalogue entries must be updated. Given the probabilities I and D for these transactions, then

$$R + U + I + D = 1.$$

To estimate the costs of index operations the following general assumptions are made. The catalogue with all attribute value entries and all target lists are stored on disk. In any case a head movement with an average seek time t_{sav} must be performed in order to find the catalogue. Another one is needed to locate at the beginning of the target list. The transfer of data from disk is done block after block where $\bar{t}_o$ is the average time for rotational delay and transfer of a block or page.

For a transaction of type Q_{Ri} the cost of transferring the total target list is

$$C_{pi} = 2 \cdot t_{sav} + (h_i + N_{Bi}) \cdot \bar{t}_o .$$

In case of a statistical query only the corresponding entry of the attribute value must be searched. Thus,

$$C_{si} = t_{sav} + h_i \cdot \bar{t}_o .$$

For the maintenance operations (Q_U, Q_I, Q_D) it is supposed that the update in the target lists can be done locally so that in each case only one block has to be transferred to main storage and written back to the original bucket of the disk. In addition the changed attribute value entry containing the actual sum (N_{ik}) has to be rewritten. The index costs for insertion and deletion of a record are always equal. The expense per inverted attribte A_i is given by

$$C_{Mi} = 2 \cdot t_{sav} + (h_i + 3) \cdot \bar{t}_o .$$

Since two catalogue entries and target lists are always involved in an update operation the overhead $C_{Ui} = 2 \cdot C_{Mi}$ arises with a transaction of type Q_{Ui}.

If no index exists for an attribute the total file has to be read and examined sequentially in case of qualification. Since the order of accessing records does not matter a physical sequential processing will be chosen for economic reasons. In our model the pages are transferred continuously from the external storage medium at maximal channel speed. By application of a synchronizing buffer technique the internal computing times for the selection of the qualified records can be overlapped. It is assumed that speed and availability of the central processor do not affect the sequential processing. This serial scan may possibly be delegated to a peripheral specialized computer (Ka75).

Assuming a cost constant K_{seq} for one record the total cost of scanning the whole file is given by

$$C_{seq} = K_{seq} \cdot N_{REC} .$$

The cost C_A for the random processing of a transaction depends mainly on the number of records (N_{qual}) accessed on disk ($C_A = N_{qual} \cdot \bar{t}$). Therefore we introduce the hit ratio of a transaction by

$$HR = \frac{N_{qual}}{N_{REC}} \cdot 100 \ \% .$$

In our simple transaction model N_{qual} corresponds to the resolution factor Rf. The expected hit ratio of a transaction Q_{Ri} is

$$HR_i = \frac{100}{j_i} \cdot \% \ .$$

The total cost for the random accesses of a transaction increases with the hit ratio, while it remains constant in the course of sequential processing. Hence, random processing is only preferable for $C_A \leqslant C_{seq}$. The hit ratio must be less than the critical boundary $HR_o = \frac{100}{j_o}$ % with $j_o = \frac{\bar{t}}{K_{seq}}$ otherwise sequential processing is more advantageous, even if an access path exists.

Random access is considered in two different cases:

1) The seeks are not coordinated. Any cylinder is entered with equal probability in random order. The average access motion is a function of the number of occupied cylinders N_{CYL}. An average access time $\bar{t}_r$ results independent of the hit ratio. The cost of this processing according to random lists is given by

$$C_{Ai} = \frac{N_{REC}}{j_i} \cdot \bar{t}_r$$

2) Since all record addresses are known to the data base system at the beginning of a random transaction processing, the access pattern to the external storage may be determined in such a manner that the access motions are performed only in one direction und thus become minimal. The average seek is a function of N_{CYL} and HR. The average access time $\bar{t}_s$ to a record diminishes with increasing hit ratio. The cost of this processing according to random-sorted lists is given by

$$C_{Ai} = \frac{N_{REC}}{j_i} \cdot \bar{t}_s(j_i) \ .$$

3. Selection of attributes for simple transaction types

Since the attributes appear to be single and independent in our transaction model they can be considered separately. The resulting costs are shown in the following table in which sequential processing is compared to random processing supported by an index. The common overhead in maintenance operations is omitted.

costs		no secondary index for A_i	secondary index for A_i
Retrieval:	C_{Ri}	$r_i \cdot C_{seq}$	$p_i \cdot C_{Ai} + s_i \cdot C_{si} + p_i \cdot C_{pi}$
Update:	C_{Ui}	0	$u_i \cdot C_{Ui}$
Insertion:	C_{Ii}	0	$I \cdot C_{Mi}$
Deletion:	C_{Di}	0	$D \cdot C_{Mi}$
Storage:	C_{STi}	0	C_{ST}

The basic demand for the inversion decision is the reduction of costs to realize a gain G compared with the physical sequential processing. Thus,

$$G_i = r_i \cdot C_{seq} - p_i \cdot C_{Ai} - C_{ST} - C_{Index} > 0 \qquad (3.1)$$

with $$C_{Index} = p_i \cdot C_{pi} + s_i \cdot C_{si} + u_i \cdot C_{Ui} + (I+D) \cdot C_{Mi}$$

In order to find the lower boundary of attribute values j_i for the inversion of A_i we replace C_{Index} with 0. From equation (3.1) follows

$$j_i > j_{\ell i} = \frac{p_i \cdot \bar{t}}{r_i \cdot K_{seq} - c \cdot S_M} \qquad \text{with } \bar{t} = \bar{t}_r \text{ resp. } \bar{t}_s \ (j_{\ell i}) \qquad (3.2)$$

Hence, the lower boundary of j is determined by

- the probability of retrieval operations,
- the average random access time,
- the speed of sequential processing,
- the storage costs.

This characteristic can be considered independent of N_{REC}, if the weak influence of N_{REC} on $\bar{t}_r$ is neglected. A critical parameter for the lower boundary of j is the length of a record R_ℓ , because the speed of sequential processing is linearly dependent on this factor, while the average random access time for $R_\ell < B$ remains approximately constant. It is shown in (Hä75) that the lower boundary of j is a useful parameter, which is easy to determine and of sufficient precision for practical applications. Note that j_o derived from relation (3.2) with $c = 0$ and $r_i = p_i$ has considerable values. Assuming the characteristics of a disk IBM 2314, $N_B = 3$ (pages per track) and load factor $\beta = 1$, some values of j_o are given in the following table.

	$\bar{t} = \bar{t}_s(j_o)$	$\bar{t} = \bar{t}_r$
$R_\ell = 80$	$j_o = 57$	$j_o = 270$
$R_\ell = 400$	$j_o = 12$	$j_o = 54$

The selection of indices without storage restrictions is obtained by inversion of $n' < n$ attributes with $j > j_\ell$ oder $G > 0$. When merely $m < n'$ attributes can be inverted, those with the highest gain contributions should be adopted. A_i is preferred to A_k, when

$$\Delta_{ik} = G_i - G_k > 0 \tag{3.3}$$

holds. Relation (3.3) is discussed for the case $r_i = p_i$. Because of $C_{Mi} \approx C_{Mk}$ it can be simplified to

$$r_i \cdot (C_{seq} - C_{Ai} - C_{pi}) - 2 \cdot u_i \cdot C_{Mi} > r_k \cdot (C_{seq} - C_{Ak} - C_{pk}) - 2 \cdot u_k \cdot C_{Mk} \tag{3.4}$$

In a given application with fixed characteristics for storage structure, file allocation and type of random processing, the three parameters $\{j,r,u\}$ of each attribute must be considered when selecting the optimal set. The comparison of pairs of triples can be achieved graphically by curve bundles in the r-j-plane. It can be shown that the update probability u is of minor importance in this respect. For sufficient large files ($N_{REC} > 10^4$) N_{REC} has no influence on the outcome of the selection, which is checked by further evaluation of relation (3.4), that is, the decision for an appropriate set of indices remains valid in case of growth or shrinkage of the file too. It should be underlined that the parameters j and r exert the dominant influence on the solution of the index problem.

4. Selection of attributes for complex transaction types

In our extended model, complex operations are admitted for retrieval and update. Disjunctions and conjunctions of search criteria are allowed in a query. For an attribute A_i it is conceivable to specify $1 \leq q_i \leq j_i$ attribute values which are connected by $\vee$ (OR). Note also that when $q_i = j_i$, the i-th attribute in no way restricts qualifying tuples and can be dropped consequently from the query. q_i is the complexity of the attribute condition (Ca73). The disjunctions of all attributes with $q_i \neq j_i$ are associated mutually by the boolean operator $\wedge$(AND). The number of the $q_i \neq j_i$ conditions denotes the complexity of the tuple condition (Ca73). Formally the transaction type Q_R is described by

$$Q_R = (q_1, q_2, \ldots, q_n) \text{ with } 1 \leq q_i \leq j_i, \quad i = 1, 2, \ldots, n$$

In case of queries related to different relations or which enforce recursions for their evaluation, Q_R can be viewed as a query primitive, which should be embedded in an appropriate non-procedural language (As74).

The access to the target lists of the specified attribute values is performed according to the model of section 2. The single lists are merged in main storage according to the applied boolean operators. It is assumed that the resulting costs of the final target list can be neglected. If all attributes specified in Q_R are inverted, the target list T(L) contains exactly the entries for the qualified tuples. In other cases only a list S(L) specifying a superset of the searched tuples can be derived. Based on the cardinality of T(L) or S(L) the decision is made whether random or sequential processing is preferable. The examination and selection of records achieved in case of a list S(L) in core is neglected in our cost model.

In case of statistical queries the answer is completed as soon as a target list T(L) is constructed. If only a list S(L) is derived all records according to this list have to be read and examined.

In case of complex transactions the expense for an attribute cannot be separated from the total costs, which depend on the set of existing indices SI respectively. With regard to section 2 the costs for index operations are obtained likewise:

Retrieval:
$$C_{pk} = \sum_{i \in SI \wedge (q_{ik} < j_i)} (2 \cdot t_{sav} + (h_i + N_{Bi}) \cdot \bar{t}_o) \cdot q_{ik}$$

Update:
$$C_{Uz} = \sum_{i \in SI \wedge (q_i = 1)} 2 \cdot C_{Mi}$$

Insertion and Deletion:
$$C_I = C_D = \sum_{i \in SI} C_{Mi}$$

For statistical and normal retrieval transactions the access costs to the records must be distinguished with regard to the list types T(L) or S(L). If the necessary access rate exceeds

$$AR \geqslant HR_o = \frac{100}{j_o} \;\%,$$

physical sequential processing is performed, because the time factor is more favourable than in random processing. Otherwise, random processing is provided for the subsetting of records. When a complete list T(L) is established for a statistical query, the access to the records is not required. Hence, the following expression for access costs holds for normal retrieval operations

$$C_{A_{pk}} = \begin{cases} \dfrac{\bar{t}}{\prod\limits_{i \varepsilon SI} \left(\frac{j_i}{q_{ik}}\right)} \cdot N_{REC} & \text{for } AR_k < \dfrac{100}{j_o} \% \\ C_{seq} & \text{for } AR_k \geqslant \dfrac{100}{j_o} \% \end{cases}$$

and for statistical retrieval operations

$$C_{A_{s\ell}} = \begin{cases} 0 & \text{for } AR_\ell = HR \\ \dfrac{\bar{t}}{\prod\limits_{i \varepsilon SI} \left(\frac{j_i}{q_{i\ell}}\right)} \cdot N_{REC} & \text{for } AR_\ell < \dfrac{100}{j_o} \% \\ C_{seq} & \text{for } AR \geqslant \dfrac{100}{j_o} \% \end{cases}$$

In normalized relations the target lists with regard to one attribute are distinct. Therefore the disjunction of those lists corresponds to an addition. The expected hit ratio of A_i is given by

$$HR_i = \frac{q_i}{j_i} \cdot 100 \% \,.$$

For the conjunction of attributes stochastic independence is assumed between their values so that the hit ratio of Q_R can be described by

$$HR = \frac{100}{\prod\limits_{i=1}^{n} \left(\frac{j_i}{q_i}\right)} \%$$

For the computation of costs an access rate AR must be applied, which is determined for a given set SI of secondary indices by

$$AR = \frac{100}{\prod\limits_{i \varepsilon SI} \left(\frac{j_i}{q_i}\right)} \% \quad \text{with } AR \geqslant HR$$

The transactions of type Q_R are separated in r_s statistical and r_p normal retrieval types with the probabilities s_ℓ resp. p_k. Hence,

$$\sum_{\ell=1}^{r_s} s_\ell + \sum_{k=1}^{r_p} p_k = R.$$

An update transaction Q_U is performed on one record, which is accessed via primary key PK. Several changes of attribute values are allowed. Formally the transaction type Q_U is described by

$$Q_U = PK : (a_1, a_2, \ldots, a_n) \quad \text{with } a_i \in \{0,1\}$$

$a_i = 1$ denotes the update of A_i. For each changed attribute value the operations described in section 2 are accomplished, if an index exists. The total probability of an update follows from the probabilities u_z of the single transaction types by

$$\sum_{z=1}^{w} u_z = U .$$

Analogous to equation (3.1) the costs resulting from all transactions in case of existing indices are gathered and compared to the results obtained if no access paths are available to support direct access. The following global gain function is received, which comprises all transactions types, their probabilities and the storage costs for indices.

$$G = R\cdot C_{seq} - \sum_{k=1}^{r_p} p_k\cdot(C_{A_{pk}}+C_{pk}) - \sum_{\ell=1}^{r_s} s_\ell\cdot(C_{A_{s\ell}}+C_{p\ell}) - \sum_{z=1}^{w} u_z\cdot C_{Uz} - (I+D)\cdot C_D - \sum_{i\in SI} C_{ST} \quad (4.1)$$

To solve the selection problem the gain G is to be maximized, that is, the set SI of secondary indices must be found such that the total gain resulting from all applications is maximal. Generally, the gain is to be computed for all possible subsets SI from the n candidates among the attributes. Only the examination of the powerset with 2^n sets guarantees an optimal solution, because no index sets can be excluded a priori. Besides the number of attribute values j_i of an attribute A_i, its links to other attributes in the single transactions of type Q_R determine, above all, its membership to the optimal solution. Global retrieval probabilities are no adequate means to describe a transaction load (Sk74, St74). It is shown in (Hä75) that generally no proper solutions can be obtained under model simplifications of this kind.

In special cases the number of the index candidates for the optimal solution can be reduced a priori to improve the computation overhead. We discuss applications without statistical queries ($s_\ell = s = 0$). When the hit ratio of a transaction k exceeds the critical limit HR_o described by

$$HR_k \geq HR_o \quad \text{or} \quad \prod_{i=1}^{n} \left(\frac{j_i}{q_{ik}}\right) \leq j_o \tag{4.2}$$

then the transaction is handled in any case in a sequential manner. If condition (4.2) holds for all transactions in which A_i happens to qualify ($q_{ik} < j_i$), than A_i is no candidate for the optimal index set IS, because no efficiency is derived comparing it with sequential processing.

$$\forall\, Q_{Rk} \;:\; (A_i \mid (q_{ik} < j_i) \wedge (\prod_{\ell=1}^{n} \left(\frac{j_\ell}{q_{\ell k}}\right) \leq j_o)) \Longrightarrow A_i \notin IS$$

This condition is independent of application parameters such as transaction probabilities.

In a second example the costs for index operations and storage space are assumed to be negligible ($C_{Index} = 0$, $c = 0$). This assumption allows a good approximate solution in many applications. Furthermore, the following relation holds

$$\frac{j_i}{q_{ik}} > j_o \qquad \text{for } j_i \neq q_{ik},$$

that is, all attributes improve the total gain in case of inversion independent on each other.

Under these assumptions equation (4.1) yields

$$\frac{G}{N_{REC}} = R \cdot K_{seq} - \sum_{k=1}^{r_p} p_k \cdot \frac{\bar{t}}{\prod_{i \in SI} \left(\frac{j_i}{q_{ik}}\right)}$$

In order to maximize the gain the cost of random processing is to be minimized. Hence,

$$\sum_{k=1}^{r_p} p_k \cdot \frac{\bar{t}}{\prod_{i \in SI} \left(\frac{j_i}{q_{ik}}\right)} \stackrel{!}{=} \text{Min}$$

The minimum is attained, if the single terms of the sum are minimized or if the expressions

$$\prod_{i \in SI} \left(\frac{j_i}{q_{ik}}\right), \quad k = 1, 2, \ldots, r_p$$

are maximized. Since $j_i \geqslant q_{ik}$, the single products are maximal, if the index set SI is maximal. Hence, all n attributes are to be inverted in this special case without storage restrictions. If only a limited number m of indices is allowed, the selection is reduced to a combinatorial problem, where the number of combinations of n attributes taken m at a time without repetitions must be examined. This is because the total gain increases constantly in this special case with the cardinality of optimal index sets. The number of index sets to be computed is given in this limited selection by

$$Z = \binom{n}{m} = \frac{n!}{m!\,(n-m)!}$$

as opposed to 2^n in the normal case.

5. Dynamic evaluation of secondary indices

Even in the case of an optimal index set, existing access paths are not used in any transaction to support retrieval, because sequential processing is preferable for high access rates. Therefore the data base system needs appropriate rules for rapid decision making. Here we discuss a procedure for predicting access rates with different degrees of precision. In queries which are formulated in a non-procedural language the qualifying predicates can be divided so that they can be represented by a binary predicate tree (As74). Three cases are distinguishable for their evaluation. A predicate is represented by a list T(L) or S(L), otherwise there is no way of locating the qualified tuples so that N_{REC} tuples must be considered. The evaluation of the predicate tree starts at the leaves, which consist of the single attribute values, and is continued recursively up to the root.

According to (As74) the possible links of these cases by AND- and OR-nodes are given in the following table. The recursive application of these connections on all non-leaf nodes permits the computation of the expected access rate. For this reason the list lengths related to N_{REC} are invented as hit probabilities p_i under the assumption of stochastic independence. On the level of the leaves these probabilities can be directly expressed by the parameters of the value distributions j_i. For disjunctions of values concerning one attribute in a normalized relation, $p_1 \cdot p_2 = 0$ yields.

		list of left subtree		
	AND	T(L2)	S(L2)	N_{REC}
list of	T(L1)	T(L1 ∧ L2)	S(L1 ∧ L2)	S(L1)
right	S(L1)	S(L1 ∧ L2)	S(L1 ∧ L2)	S(L1)
subtree	N_{REC}	S(L2)	S(L2)	N_{REC}

		hit probability of left subtree		
	AND	$T(p_2)$	$S(p_2)$	1
hit probability	$T(p_1)$	$T(p_1 \cdot p_2)$	$S(p_1 \cdot p_2)$	$S(p_1)$
of right	$S(p_1)$	$S(p_1 \cdot p_2)$	$S(p_1 \cdot p_2)$	$S(p_1)$
subtree	1	$S(p_2)$	$S(p_2)$	1

		list of left subtree		
	OR	T(L2)	S(L2)	N_{REC}
list of	T(L1)	T(L1 ∨ L2)	S(L1 ∨ L2)	N_{REC}
right	S(L1)	S(L1 ∨ L2)	S(L1 ∨ L2)	N_{REC}
subtree	N_{REC}	N_{REC}	N_{REC}	N_{REC}

		hit probability of left subtree		
	OR	$T(p_2)$	$S(p_2)$	1
hit probability	$T(p_1)$	$T(p_1+p_2-p_1 \cdot p_2)$	$S(p_1+p_2-p_1 \cdot p_2)$	1
of right	$S(p_1)$	$S(p_1+p_2-p_1 \cdot p_2)$	$S(p_1+p_2-p_1 \cdot p_2)$	1
subtree	1	1	1	1

The expected hit probability for the root is needed to determine the kind of further processing. The outlined procedure renders the expected value of the access rate of a transaction type. In order to improve the accuracy of the forecast the actual list lengths N_{ik} stored in the catalogue can be taken to process a particular query. However, additional overhead is imposed to acquire this information. The most expensive way to support the decision is finally the actual merging of target lists, which provides the actual access rate.

6. Conclusion

The problem of finding an optimal set of secondary indices for a given transaction load is considered. In our model the description of queries, updates, insertions and deletions, which are characterized by their type and probability, is provided. For simple transaction types, which refer only to one single attribute, a "decoupled model" is established. Each attribute can be considered separately so that simple analytic solutions are obtained. With respect to inversion a lower boundary depending on detailed storage and processing characteristics exists for the number of attribute values per attribute.

For complex transaction types their exact attribute combinations and their probabilities are taken into account. To find an optimal index set, in the general case the powerset of all attributes must be examined. In special cases one can get simple solutions.

A prediction procedure for the estimation of the access time behavior during the dynamic evaluation of secondary indices is given.

References

(As74) Astrahan, M.M., Chamberlin, D.D.: Implementation of a Structured English Query Language, IBM Research Report, RJ 1464, San José, Oct. 28, 1974.

(Ca73) Cardenas, A.F.: Evaluation and Selection of File Organization - a Model and System, in: CACM, Vol. 16, No. 9, Sept. 1975, pp. 540-548.

(Co70) Codd, E.F.: A Relational Model of Data for Large Shared Data Banks, in: CACM, Vol. 13, No. 6, June 1970, pp. 377-387.

(Hä75) Härder, T.: Auswahl optimaler Indexmengen, Research Report DV 75-2, FG DVS, TH Darmstadt, Sept. 1975.

(Ka75) Karlowsky, I., Leilich, H.-D., Stiege, G.: Ein Suchrechnerkonzept für Datenbankanwendungen, in: Elektronische Rechenanlagen 17 (1975), No. 3, pp. 108-118.

(Ki74) King, W.F.: On the Selection of Indices for a File, IBM Research Report, RJ 1341, San José, 1974.

(Lu71) Lum, V.A., Ling, H.: An Optimization Problem on the Selection of Secondary Keys, in: Proc. of ACM National Conference, 1971, pp. 349-356.

(Ro74) Rothnie, J.B., Lozano, T.: Attribute Based File Organization in a Paged Memory Environment, in: CACM, Vol. 17, No. 2, Feb. 1974, pp. 63-69.

(Sk74) Schkolnick, M.: Optimizing Partial Inversions for Files, IBM Research Report, RJ 1477, San José, Nov. 21, 1974.

(St74) Stonebraker, M.: The Choice of Partial Inversions and Combined Indices, appears in Journal of Computer and Information Sciences.

(Wa73) Wang, C.P.: Parametrization of Information System Application, IBM Research Report, RJ 1199, San José, April 11, 1973.

(We73) Wedekind, H.: Systemanalyse, Carl Hanser Verlag, München 1973.

(Wo71) Wong, E., Chiang, T.C.: Canonical Structure in Attribute Based File Organization, in: CACM, Vol. 14, No. 9, Sept. 1971, pp. 593-597.