ABSTRACT

The concept of time-constrained SQL queries was introduced to address the problem of long-running SQL queries. A key approach adopted for supporting time-constrained SQL queries is to use sampling to reduce the amount of data that needs to be processed, thereby allowing completion of the query in the specified time constraint. However, sampling does make the query results approximate and hence requires the system to estimate the values of the expressions (especially aggregates) occurring in the select list. Thus, coming up with estimates for aggregates is crucial for time-constrained approximate SQL queries to be useful, which is the focus of this paper. Specifically, we address the problem of estimating commonly occurring aggregates (namely, SUM, COUNT, AVG, MEDIAN, MIN, and MAX) in time-constrained approximate queries. We give both point and interval estimates for SUM, COUNT, AVG, and MEDIAN using Bernoulli sampling for various type of queries, including join processing with cross product sampling. For MIN (MAX), we give the confidence level that the proportion 100γ% of the population will exceed the MIN (or be less than the MAX) obtained from the sampled data.

1. INTRODUCTION

The growing nature of databases, compounded with the ability to formulate arbitrarily complex SQL queries, has led to the problem of long-running, complex SQL queries.

A solution being explored is to support time-constrained SQL queries [2], [3] that would complete in a specified time constraint either by computing the first few rows (top-K rows) or approximate results through sampling. Of the two approaches, the latter approach, namely approximate query processing, is very promising in that the query processing time could be reduced significantly by controlling the sample size. A practical application of time-constrained approximate query processing is queries involving aggregate functions.

Such queries are popular in applications such as OLAP and they tend to be long running as they compute aggregate values over large datasets. However, supporting time-constrained approximate SQL queries require work in two areas for them to become a practical and useful solution.

First, the user-specified time-constraint needs to be implicitly transformed to SAMPLE clauses on individual tables. This was addressed in [3], which presented estimation of sample sizes for queries involving various relational operations.

Second, the problem of estimating aggregates needs to be considered. Thus, in this paper, we focus on estimating aggregates in time-constrained approximate SQL queries. Since the aggregates in time-constrained approximate queries are computed only once, the result could vary significantly based on the chosen sample size, which warrants that additional measures are provided characterizing the goodness of the results. We consider commonly occurring aggregates, namely, SUM, COUNT, AVG, MEDIAN, MIN, and MAX. The measures (apart from the point estimate) that are useful are confidence intervals for aggregates SUM, COUNT, AVG, and MEDIAN, and confidence levels for aggregates returning extreme values (such as MIN and MAX) as tolerance limits. The aggregate estimation techniques are presented for join queries that employ cross-product sampling [1].

The rest of this paper is organized as follows: Section 2 describes the Bernoulli sampling scheme. Section 3 discusses the estimation for SUM, COUNT, and AVG. Sections 4 and 5 cover the estimation for MEDIAN, QUANTILE, MIN and MAX. The results in Sections 3, 4, and 5 are presented by assuming row sampling but could be extended to block sampling as well, as discussed in Section 6. Section 7 concludes the paper.

2. BERNOULLI SAMPLING

Oracle Database supports the Bernoulli (coin-flip) sampling scheme, where the sample percentage (f) indicates the probability of each row, or each cluster of rows in the case of block sampling, being independently selected as part of the sample. Because the database does not retrieve the exact sample size of the rows (blocks) of table, Bernoulli sampling is a variable size sampling scheme. The mean and variance of the random sample size n are given by $E(n) = fn$ and $V(n) = f(1-f)N$, where N is the population size, or the number of rows (blocks) in the case of row sampling (block sampling). In this paper we assume that the value of N is known from Oracle Database’s object-level statistics, which includes the number of blocks and the number of rows in a table.

3. SUM, COUNT, AND AVG

We start with the formulas for the estimated COUNT, SUM, and AVG and their variance in join operations. We assume that there are k tables in the join operations and each table’s sample percentage (fj, j = 1, ..., k) is calculated by an algorithm described in [3]. We also assume that the j-th sample $S_j$ has $N_j$ rows chosen from $N_j$ rows of the j-th table $R_j$, where the value of $N_j$ is known from table statistics. These assumptions also apply to the Sections 4 and 5. Note that under the Bernoulli (coin-flip) sampling, $n_j$ is a random variable with mean $E(n_j) = f_jN_j$. 

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the ACM. To copy otherwise, or to republish, to post on servers or to redistribute to lists, requires a fee and/or special permissions from the publisher, ACM.
3.1 SUM without Selection

In this section, we discuss the estimated \( \text{SUM}(\text{expr}) \) without selection, and its variance. We use the following notation to specify \( \text{SUM}: Y = \sum_{i_1}^{R_1} y_{i_1} \cdots \sum_{i_k}^{R_k} y_{i_k} \), where \( \sum_{i_1}^{R_1} \) is an abbreviated notation for \( \sum_{i_1 = 1}^{R_1} \cdots \sum_{i_k = 1}^{R_k} \), and \( y_{i_1} \cdots y_{i_k} \) is the value obtained from the unit or expression after joining the \( k \) tables, i.e., after joining the \( j_1 \)-th row in \( R_1 \), \ldots, and the \( j_k \)-th row in \( R_k \), the value for the resulting unit or expression is denoted by \( y_{i_1} \cdots y_{i_k} \). Two estimators of \( Y \) are given by: \( \hat{Y}_a = \frac{1}{s_1} \sum_{i_1}^{R_1} s_1 y_{i_1} \cdots \sum_{i_k}^{R_k} y_{i_k} \) and \( \hat{Y}_s = \frac{N_1 \cdots N_k}{n_1 \cdots n_k} \sum_{i_1}^{R_1} s_1 y_{i_1} \cdots \sum_{i_k}^{R_k} y_{i_k} \), where \( \hat{Y}_a \) can be shown as a special case of \( \pi \) estimators or Horvitz-Thompson estimators, which is unbiased; and \( \hat{Y}_s \) is an approximately unbiased estimator, which has a smaller variance than \( \hat{Y}_a \). Note that \( \sum_{i_1}^{R_1} \) in \( \hat{Y}_s \) and \( \sum_{i_k}^{R_k} \) in \( \hat{Y}_a \) may be different values because the former is from the resulting unit by joining the \( i_1 \)-th row in \( S_1 \), \ldots, and the \( i_k \)-th row in \( S_k \) whereas the latter is from the resulting unit by joining the \( i_1 \)-th row in \( R_1 \), \ldots, and the \( i_k \)-th row in \( R_k \). The symbol \( \hat{Y} \) denotes an estimate of a population characteristic, which is made from a sample.

**Theorem 1:** \( \hat{Y}_s \) is an unbiased estimator of \( Y \). \( \hat{Y}_s \) is an approximately unbiased estimator of \( Y \).

To prove that \( \hat{Y}_s \) is an unbiased estimator, let \( a_{i_1} \cdots a_{i_k} \) be a random variable that takes the value 1 if the \( i_1 \)-th row of the first table \( R_1 \) is selected in the sample, and the value 0 otherwise. So are \( a_{i_1} \), \( a_{i_2} \), \ldots, and \( a_{i_k} \). It is obvious that \( E(a_{i_1} \cdots a_{i_k}) = E(a_{i_1}) \cdots E(a_{i_k}) = f_{i_1} \cdots f_{i_k} \) because of the independent sampling over different tables. Therefore:

\[
E(\hat{Y}_s) = \frac{1}{f_{i_1} \cdots f_{i_k}} E(\sum_{y_{i_1} \cdots y_{i_k}}) = \frac{1}{f_{i_1} \cdots f_{i_k}} E(\sum_{a_{i_1} \cdots a_{i_k}} y_{i_1} \cdots y_{i_k}) = Y.
\]

That \( \hat{Y}_s \) is approximately unbiased can be proved by using the first-order Taylor approximation:

\[
\hat{Y}_s \approx Y + \frac{1}{f_{i_1} \cdots f_{i_k}} \sum_{y_{i_1} \cdots y_{i_k}} (y_{i_1} \cdots y_{i_k} - Y).
\]

**Theorem 2:** The variance of \( \hat{Y}_s \) is

\[
V(\hat{Y}_s) = \sum_{G \in P(\{1, \ldots, k\})} \prod_{j \in G} \frac{1-f_j}{f_j} \sum_{\{i_1, \cdots, i_k\} \in G} \left( \sum_{y_{i_1} \cdots y_{i_k}} \right)^2
\]

where \( P(\{1, \ldots, k\}) \) is the power set of \( \{1, \ldots, k\} \), or the set of \( k \) tables, \( \sum_{\{i_1, \cdots, i_k\} \in G} \) is an abbreviated notation for \( \sum_{i_1 \in R_1, \cdots, i_k \in R_k} \), when \( G = \{g_1, \ldots, g_x\} \) and \( |G| = x \leq k \), and \( G^c = P(\{1, \ldots, k\}) \setminus G \), or the complement of \( G \). The variance of \( \hat{Y}_s \) is approximately

\[
V(\hat{Y}_s) \approx \sum_{G \in P(\{1, \ldots, k\})} \prod_{j \in G} \frac{1-f_j}{f_j} \sum_{\{i_1, \cdots, i_k\} \in G} \left( \sum_{y_{i_1} \cdots y_{i_k}} \right)^2.
\]

3.2 AVG without Selection

We use \( \overline{Y} \) to denote AVG: \( \overline{Y} = \frac{1}{N_1 \cdots N_k} \sum_{i_1}^{R_1} \cdots \sum_{i_k}^{R_k} y_{i_1} \cdots y_{i_k} \), and \( \hat{Y} \) to denote the estimator of AVG: \( \hat{Y} = \frac{Y}{N_1 \cdots N_k} \), which is approximately unbiased. Note \( \hat{Y} \) is also written as \( \hat{Y} \) because \( \hat{Y} = Y / n_1 \cdots n_k = \sum_{i_1}^{R_1} \cdots \sum_{i_k}^{R_k} y_{i_1} \cdots y_{i_k} / n_1 \cdots n_k = Y / N_1 \cdots N_k \). And the variance of \( \hat{Y} \) or \( \overline{Y} \) and its estimator are given by

\[
V(\hat{Y}) = V(\overline{Y}) = V(\overline{Y}) / N_1 \cdots N_k
\]

and

\[
\hat{V}(\hat{Y}) = \hat{V}(\overline{Y}) = \hat{V}(\overline{Y}) / N_1 \cdots N_k.
\]

3.3 SUM and COUNT with Selection

When there are predicates, we can take \( y_{i_1} \cdots y_{i_k} = 0 \) if the resulting unit is not selected, \( y_{i_1} \cdots y_{i_k} = 1 \) for COUNT and \( y_{i_1} \cdots y_{i_k} \) for SUM if the resulting unit is selected. Thus, the results in Section 3.1 still hold.
3.4 AVG with Selection

In the selection case, AVG(expr_a) can be written as SUM(expr_a)/COUNT(expr_a). When tables are sampled, we can use estimatedSUM(expr_a)/estimatedCOUNT(expr_a) as an estimator of AVG(expr_a). It is called a ratio estimator, which is shown to be an approximately unbiased estimator. In this section, we briefly describe how to calculate the variance of the new estimator.

Let \( \hat{Y} \) and \( \hat{\hat{Y}} \) denote the SUM and its estimator respectively, \( X \) and \( \hat{X} \) denote the COUNT and its estimator respectively, \( R \) and \( \hat{R} \) denote the AVG and its estimator respectively.

**Theorem 4:** The variance of \( \hat{R} \) is approximately

\[
V(\hat{R}) = V(\hat{\hat{Y}}) + \hat{R}V(\hat{X}) - 2R\text{Cov}(\hat{Y}, \hat{X}) = \frac{1}{\hat{X}^2} \sum_{x \in \text{PAT}} \left[ \prod_{j \in x} \frac{1-f_j}{f_j} \sum_{y_j \in \text{PAT}[x]} \left( \sum_{y_{i_1} \in \text{PAT}[x]} - Rx_{i_1} \right) \right]^2
\]

and an estimator for \( V(\hat{R}) \) is given by:

\[
\hat{V}(\hat{R}) = V(\hat{\hat{Y}}) + \hat{R}V(\hat{X}) - 2R\text{Cov}(\hat{Y}, \hat{X}) = \frac{N_C}{\hat{X}^2} \sum_{x \in \text{PAT}} (1-f_j) \sum_{y_j \in \text{PAT}[x]} \left( \sum_{y_{i_1} \in \text{PAT}[x]} - \hat{R}x_{i_1} \right) \right]^2
\]

where \( \hat{Cov}(\hat{Y}, \hat{X}) \) is an estimator of \( \text{Cov}(\hat{Y}, \hat{X}) \).

The proof is omitted due to space limitations.

The ratio estimator technique can be directly applied to the \( \text{SUM(expr_a)}/\text{SUM(expr_b)} \) case. For any complex expression involving aggregates that can be written as an expression of \( \text{SUM} \) and \( \text{COUNT} \), its approximate variance can be obtained, using the first-order approximation of the Taylor series of these expressions.

4. MEDIAN

In [4], Manku et al. studied a sampling-based MEDIAN algorithm. However, their sampling operation occurs only in the final stage. Unlike their approach, we push the sampling operations as early as possible to achieve the time constraint, but run the exact MEDIAN algorithm over the approximated result from sampling operations. We separate our discussion into the cases of without selection, and with selection.

4.1 MEDIAN without Selection

The single table case is omitted since it is similar to the case studied in Section 5 of [4]. So we start with the cross-product case under the Bernoulli sampling scheme. Besides the assumptions made in Section 3, such as \( k \) samples \( (S_j, j = 1, \ldots, k) \) are obtained from \( k \) tables \( (R_j, j = 1, \ldots, k) \), we assume that \( M \) is the MEDIAN of the \( N_j \ldots N_k \) elements. Let \( x_{i_1 \ldots i_k} = 1 \) if \( y_{i_1 \ldots i_k} < M \), \( x_{i_1 \ldots i_k} = 0 \) otherwise.

\[
\sum_{i_1 \ldots i_k} = M, \quad 0 \text{ otherwise; and } \quad \hat{x}_{i_1 \ldots i_k} = M, \quad 0 \text{ otherwise; and } \quad \hat{x}_{i_1 \ldots i_k} = \hat{M} \text{, } 0 \text{ otherwise; and } \quad \hat{\hat{x}}_{i_1 \ldots i_k} = \hat{\hat{M}} \text{, } 0 \text{ otherwise; and } \quad \hat{\hat{x}}_{i_1 \ldots i_k} = \hat{\hat{\hat{M}}}.
\]

We also assume a continuous distribution model, i.e. \( y_{(w)} = m - \mu(y_{(w)}) + \sigma(y_{(w)})z_{(w)} \) if \( m \) is not an integer. Thus \( M = y_{(\mu + \sigma z)}(N_{(\mu + \sigma z)}) \) is the MEDIAN over the order statistics. To estimate the value of \( M \) over a cross-product sample of \( n_1 \ldots n_k \) elements, we can estimate \( \hat{x}_{i_1 \ldots i_k} = \hat{\hat{x}}_{i_1 \ldots i_k} = \hat{\hat{\hat{x}}}_{i_1 \ldots i_k} = \hat{\hat{\hat{\hat{x}}}}_{i_1 \ldots i_k} \), and use the following equal.

4.2 MEDIAN with Selection

When there is a predicate, only a fraction (say \( w \) elements) of the \( n_1 \ldots n_k \) elements (i.e. the sample) is returned. We can get the estimated \( M \) over a cross-product sample of \( w \) elements. An approximate confidence interval is calculated as follows:

Let \( x_{i_1 \ldots i_k} = 1 \) if \( y_{i_1 \ldots i_k} < M \) and \( y_{i_1 \ldots i_k} \) is selected, \( 0.5 \) if \( y_{i_1 \ldots i_k} = M \), and \( 0 \) otherwise, and
\( \tau_{\text{median}} = \frac{1}{n_1 \cdots n_k} \sum_{i=1}^{n_k} y_{i1} \cdots y_{i1} \). Note that \( \tau_{\text{median}} \) is equal to 0.5w/(n_1 \cdots n_k), different from the value of 0.5 in Section 4.1. Therefore, we can have the confidence interval \([y_{ij}, y_{uw}]\) at an approximate 100(1-\( \alpha \))% confidence level, where \( l \) and \( u \) are defined as:

\[
\begin{align*}
  l &= 1 + \left( 0.5 - Z_{\alpha/2} \sqrt{V(\tau_{\text{median}})} / w \right) (w-1) \\
  u &= 1 + \left( 0.5 + Z_{\alpha/2} \sqrt{V(\tau_{\text{median}})} / w \right) (w-1).
\end{align*}
\]

Note that under the Bernoulli (coin-flip) sampling scheme, \( w/n_1 \cdots n_k \) is an approximately unbiased estimator of the population proportion \( w/N_1 \cdots N_k \), where \( w \) elements will be selected from the population of \( N_1 \cdots N_k \) elements. But \( w \) is unknown to our system, because the whole population of \( N_1 \cdots N_k \) elements is never processed. In contrast, \( w \) is known in the case of sampling in the final stage [4], because its purpose is not to reduce the processing time of join and selection operations. Therefore under our sampling scheme, we have to calculate the estimated variance \( V(\tau_{\text{median}}) \) in the context of \( n_1 \cdots n_k \) elements.

### 4.3 Extension to QUANTILE

The techniques in Sections 4.1, and 4.2 can also be applied to the QUANTILE aggregate. For example, the \( \phi \) QUANTILE can return the element in position \( 1+\phi(N-1) \) in the sorted sequence of \( N \) elements. MEDIAN is the 50% QUANTILE. Let \( Q \) be the required \( \phi \) QUANTILE. So we can simply use formulas described in Sections 4.1, and 4.2, and replace 0.5 and \( \sqrt{V(\tau_{\text{median}})} \) with \( \phi \) and \( \sqrt{V(\tau_{\phi})} \) respectively to obtain approximate confidence intervals for QUANTILE.

### 5. MIN AND MAX

In our time-constrained approximate queries, we return the MIN and MAX over the sample as the estimated MIN and MAX over the population. To estimate the goodness of the estimated MIN or MAX that is returned, we compute the confidence level that the proportion 100\( \gamma \)% of the population will exceed the MIN (or be less than the MAX) in the sample. This measure is related to the one-sided tolerance limit, which is given by the MIN (or MAX) in a sample of size \( n \), where \( n \) is determined so that one can assert with 100(1-\( \alpha \))% confidence that at least the proportion 100\( \gamma \)% of the population will exceed the MIN (or be less than the MAX) in the sample.

To compute the confidence level that the proportion \( \gamma \)% of the population will exceed the MIN (or be less than the MAX), we compare the lower bound of \( \phi=5\% \) QUANTILE with the MIN, or compare the upper bound of \( \phi=95\% \) QUANTILE with the MAX. Assume a positive \( Z_{\alpha} \) satisfies \( \Phi(Z_{\alpha}) = 1 - \alpha \), where we directly use \( \alpha \) because we only use one-sided limits to compute \( \Pr(y_{ij} \geq \text{MIN}) \) for \( \phi = 5\% \) and \( \Pr(y_{ij} \leq \text{MAX}) \) for \( \phi = 95\% \).

For example, in the case without selection, \( Z_{\alpha} \) can be computed as follows:

\[
\begin{align*}
  l &= 1 + \left( 0.05 - Z_{\alpha} \sqrt{V(\tau_{\text{median}}) / w} \right) (w-1) \\
  u &= 1 + \left( 0.95 + Z_{\alpha} \sqrt{V(\tau_{\text{median}}) / w} \right) (w-1).
\end{align*}
\]

Note that normally \( Z_{\alpha} > 0 \). Once we obtain \( Z_{\alpha} \), we can obtain the confidence level: 100(1-\( \alpha \)% = 1 - \( \alpha \) = \( \Phi(Z_{\alpha}) \).

### 6. BLOCK SAMPLING

The results in Sections 3, 4 and 5 assume row sampling, but can be extended to block sampling. We briefly discuss one estimator used for the extension.

Let \( y_{i1} \cdots i_k = \sum_{j_1 \cdots j_k} y_{i1} \cdots j_k \), where \( y_{i1} \cdots j_k \) is the value obtained from the unit after joining the \( j_1 \)-th row in the \( i_1 \)-th block \( SB_{i1} \) of the sample \( S_l \) that has \( m_l \) blocks chosen from \( M_l \) blocks of the first table \( R_l \) under Bernoulli sampling, \( \ldots \), and the \( j_k \)-th row in the \( i_k \)-th block \( SB_{i1} \) of the sample \( S_l \) that has \( m_l \) blocks chosen from \( M_l \) blocks of the \( k \)-th table \( R_k \) under Bernoulli sampling. Then take \( \tilde{y}_k = M_1 \cdots M_k \sum_{i_1 \cdots i_k} y_{i1} \cdots j_k \) as an approximately unbiased estimator of \( Y_k \), or SUM, which is similar to \( \tilde{y} \) described in Section 3.1.

### 7. CONCLUSION

In this paper, the most common aggregates in SQL including SUM, COUNT, AVG, MEDIAN, MIN, and MAX are studied in time-constrained approximate queries. We not only present the point estimates for these aggregates, but also present the interval estimates for these aggregates, (more specifically, the confidence intervals for SUM, COUNT, AVG, and MEDIAN, and confidence level that MIN or MAX is taken as a tolerance limit.) These results are the foundation of estimation in time-constrained approximate queries.

### 8. ACKNOWLEDGMENT

We thank Jay Banerjee and Sue Mavris for their encouragement and support.

### 9. REFERENCES


