Estimating Aggregates in Time-Constrained Approximate Queries in Oracle

Ying Hu
ying.hu@oracle.com

Seema Sundara
seema.sundara@oracle.com

Jagannathan Srinivasan
jagannathan.srinivasan@oracle.com

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.

EDBT'09, March 24-26, 2009, Saint Petersburg, Russia.

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 (f_j, $j = 1, \ldots, k$) 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 $E(n_j) = f_j N_j$.

3.1 SUM without Selection

In this section, we discuss the estimated `SUM(expr)` without selection, and its variance. We use the following notation to specify `SUM`: $Y = \sum_{i_1,\dots,i_k}^{R_1,\dots,R_k} y_{i_1,\dots,i_k}$, where $\sum_{i_1,\dots,i_k}^{R_1,\dots,R_k}$ is an abbreviated notation for $\sum_{i_1\in R_1}\sum_{i_2\in R_2}\dots\sum_{i_k\in R_k}$, and $y_{i_1,\dots,i_k}$ is the value obtained from the unit or expression after joining the k tables, i.e. after joining the i_1-th row in R_1, … , and the i_k-th row in R_k, the value for the resulting unit or expression is denoted by $y_{i_1,\dots,i_k}$. Two estimators of Y are given by: $\hat{Y}_\pi = \frac{1}{f_1\cdots f_k}\sum_{i_1,\dots,i_k}^{S_1,\dots,S_k} y_{i_1,\dots,i_k}$ and $\hat{Y} = \frac{N_1\dots N_k}{n_1\dots n_k}\sum_{i_1,\dots,i_k}^{S_1,\dots,S_k} y_{i_1,\dots,i_k}$ where $\hat{Y}_\pi$ can be shown as a special case of π estimators or Horvitz-Thompson estimators, which is unbiased; and $\hat{Y}$ is an approximately unbiased estimator, which has a smaller variance than $\hat{Y}_\pi$. Note that $y_{i_1,\dots,i_k}$ in $\sum_{i_1,\dots,i_k}^{S_1,\dots,S_k} y_{i_1,\dots,i_k}$ and $y_{i_1,\dots,i_k}$ in $\sum_{i_1,\dots,i_k}^{R_1,\dots,R_k} y_{i_1,\dots,i_k}$ may be different values because the former is from the resulting unit by joining the i_1-th row in S_1, … , 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, … , and the i_k-th row in R_k. The symbol ^ denotes an estimate of a population characteristic, which is made from a sample.

Theorem 1: $\hat{Y}_\pi$ is an unbiased estimator of Y. $\hat{Y}$ is an approximately unbiased estimator of Y.

To prove that $\hat{Y}_\pi$ is an unbiased estimator, let a_{i_1} 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_2}, … , and a_{i_k}. It is obvious that $E(a_{i_1}\dots a_{i_k}) = E(a_{i_1})\dots E(a_{i_k}) = f_1\dots f_k$ because of the independent sampling over different tables. Therefore:

$$E(\hat{Y}_\pi) = \frac{1}{f_1\cdots f_k}E\left(\sum_{i_1,\dots,i_k}^{S_1,\dots,S_k} y_{i_1,\dots,i_k}\right) = \frac{1}{f_1\cdots f_k}E\left(\sum_{i_1,\dots,i_k}^{R_1,\dots,R_k} a_{i_1}\dots a_{i_k}\, y_{i_1,\dots,i_k}\right) = Y.$$

That $\hat{Y}$ is approximately unbiased can be proved by using the first-order Taylor approximation: $\hat{Y} \doteq Y + \frac{1}{f_1\cdots f_k}\sum_{i_1,\dots,i_k}^{S_1,\dots,S_k}\left(y_{i_1,\dots,i_k} - \frac{Y}{N_1\dots N_k}\right)$.

Theorem 2: The variance of $\hat{Y}_\pi$ is

$$V(\hat{Y}_\pi) = \sum_{G\in P(\{1,\dots,k\})}\left[\prod_{g\in G}\frac{1-f_g}{f_g}\sum_{\{i_g\in R_g|g\in G\}}\left(\sum_{\{i_h\in R_h|h\in G^C\}} y_{i_1,\dots,i_k}\right)^2\right]$$

where $P(\{1,\dots,k\})$ is the power set of $\{1, ..., k\}$, or the set of k tables, $\sum_{\{i_g\in R_g|g\in G\}}$ is an abbreviated notation for $\sum_{i_{g_1},\dots,i_{g_x}}^{R_{g_1},\dots,R_{g_x}}$ when $G=\{g_1, \dots, g_x\}$ and $|G| = x \le k$, and $G^C = P(\{1,\dots,k\}) \setminus G$, or the complement of G. The variance of $\hat{Y}$ is approximately

$$V(\hat{Y}) \doteq \sum_{G\in P(\{1,\dots,k\})}\left[\prod_{g\in G}\frac{1-f_g}{f_g}\sum_{\{i_g\in R_g|g\in G\}}\left(\sum_{\{i_h\in R_h|h\in G^C\}}\left(y_{i_1,\dots,i_k} - \frac{Y}{N_1\dots N_k}\right)\right)^2\right]$$

$$= V(\hat{Y}_\pi) - \sum_{G\in P(\{1,\dots,k\})}\left[\prod_{g\in G}\frac{1-f_g}{f_g N_g}Y^2\right].$$

The proof is omitted due to space limitations.

Theorem 3: An unbiased estimator of $V(\hat{Y}_\pi)$ is given by

$$\hat{V}(\hat{Y}_\pi) = \frac{1}{f_1^2\cdots f_k^2}\sum_{G\in P(\{1,\dots,k\})}(-1)^{|G|-1}\left[\prod_{g\in G}(1-f_g)\sum_{\{i_g\in S_g|g\in G\}}\left(\sum_{\{i_h\in S_h|h\in G^C\}} y_{i_1,\dots,i_k}\right)^2\right].$$

An estimator of $V(\hat{Y})$ is given by

$$\hat{V}(\hat{Y}) = \frac{N_1^2\dots N_k^2}{n_1^2\dots n_k^2}\sum_{G\in P(\{1,\dots,k\})}(-1)^{|G|-1}\left[\prod_{g\in G}(1-f_g)\sum_{\{i_g\in S_g|g\in G\}}\left(\sum_{\{i_h\in S_h|h\in G^C\}}\left(y_{i_1,\dots,i_k} - \frac{\hat{Y}}{N_1\dots N_k}\right)\right)^2\right].$$

The proof is omitted due to space limitations.

Note that the condition of achieving the minimal $V(\hat{Y}_\pi)$ or $V(\hat{Y})$ may be different from the condition of achieving the maximal $f_1 * \dots * f_k$ or maximal $n_1 * \dots * n_k$, which is described in [3]. In practice, because many factors in these equations are unknown prior to query, or the knowledge of the variances is absent without trials, we believe that the objective of achieving the maximal $f_1 * \dots * f_k$ or maximal $n_1 * \dots * n_k$ is justified. As $V(\hat{Y})$ is normally smaller than $V(\hat{Y}_\pi)$, we will focus on $\hat{Y}$, $V(\hat{Y})$, and $\hat{V}(\hat{Y})$ in the rest of this paper.

According to finite-population Central Limit Theorem, $(\hat{Y}-Y)/\sqrt{V(\hat{Y})}$ or $(\hat{Y}-Y)/\sqrt{\hat{V}(\hat{Y})}$ tends to normality as n_1, … , and n_k increase. Let $Z_{\alpha/2}$ satisfy $\Phi(Z_{\alpha/2}) = 1-\alpha/2$, where Φ is the cumulative distribution function of $N(0,1)$. So the 100(1-α)% confidence interval for Y is often computed as $[\hat{Y} - Z_{\alpha/2}\sqrt{\hat{V}(\hat{Y})},\ \hat{Y} + Z_{\alpha/2}\sqrt{\hat{V}(\hat{Y})}]$. The normal approximation is used for not only `SUM`, but also `COUNT` and `AVG`, when n_1, … , and n_k are large.

3.2 AVG without Selection

We use $\bar{Y}$ to denote `AVG`: $\bar{Y} = \frac{1}{N_1\dots N_k}\sum_{i_1,\dots,i_k}^{R_1,\dots,R_k} y_{i_1,\dots,i_k}$, and $\hat{\bar{Y}}$ to denote the estimator of `AVG`: $\hat{\bar{Y}} = \hat{Y}/N_1\dots N_k$, which is approximately unbiased. Note $\hat{\bar{Y}}$ is also written as $\bar{y}$ because $\hat{\bar{Y}} = \hat{Y}/N_1\dots N_k = \sum_{i_1,\dots,i_k}^{S_1,\dots,S_k} y_{i_1,\dots,i_k}/n_1\dots n_k = \bar{y}$. And the variance of $\hat{\bar{Y}}$ or $\bar{y}$ and its estimator are given by $V(\hat{\bar{Y}}) = V(\bar{y}) = V(\hat{Y})/N_1^2\dots N_k^2$ and $\hat{V}(\hat{\bar{Y}}) = \hat{V}(\bar{y}) = \hat{V}(\hat{Y})/N_1^2\dots N_k^2$.

3.3 SUM and COUNT with Selection

When there are predicates, we can take $y'_{i_1,\dots,i_k} = 0$ if the resulting unit is not selected, $y'_{i_1,\dots,i_k} = 1$ for COUNT and $y'_{i_1,\dots,i_k} = y_{i_1,\dots,i_k}$ 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 Y and $\hat{Y}$ 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

$$V(\hat{R}) \doteq \frac{V(\hat{Y}) + R^2 V(\hat{X}) - 2RCov(\hat{Y},\hat{X})}{X^2} =$$

$$\frac{1}{X^2} \sum_{G \in P(\{1,\ldots,k\})} \left[\prod_{g \in G} \frac{1-f_g}{f_g} \sum_{\{i_g \in R_g | g \in G\}} \left(\sum_{\{i_h \in R_h | h \in G^C\}} (y_{i_1,\ldots,i_k} - Rx_{i_1,\ldots,i_k}) \right)^2 \right]$$

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

$$\hat{V}(\hat{R}) = \frac{\hat{V}(\hat{Y}) + \hat{R}^2 \hat{V}(\hat{X}) - 2\hat{R}\hat{C}ov(\hat{Y},\hat{X})}{\hat{X}^2} =$$

$$\frac{N_1^2 \ldots N_k^2}{\hat{X}^2 n_1^2 \ldots n_k^2} \sum_{G \in P(\{1,\ldots,k\})} (-1)^{|G|-1} \left[\prod_{g \in G} (1-f_g) \sum_{\{i_g \in S_g | g \in G\}} \left(\sum_{\{i_h \in S_h | h \in G^C\}} (y_{i_1,\ldots,i_k} - \hat{R}x_{i_1,\ldots,i_k}) \right)^2 \right]$$

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

The proof is omitted due to space limitations.

The ratio estimator technique can be directly applied to the `SUM(expr_a)/SUM(expr_b)` case. For any complex expression involving aggregates that can be written as an expression of `SUM` and `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, …, k) are obtained from k tables (R_j, j=1, …, k), we assume that M is the `MEDIAN` of the $N_1 \ldots N_k$ elements. Let $x_{i_1,\ldots,i_k} = 1$ if $y_{i_1,\ldots,i_k} < M$, $x_{i_1,\ldots,i_k} = 0.5$[1] if $y_{i_1,\ldots,i_k} = M$, 0 otherwise; and $\overline{X}_{median} = \frac{1}{N_1 \ldots N_k} \sum_{i_1,\ldots,i_k}^{R_1,\ldots,R_k} x_{i_1,\ldots,i_k} = 0.5$.

We also assume a continuous distribution model, i.e. $y_{(m)} = (\lceil m \rceil - m) y_{(\lfloor m \rfloor)} + (m - \lfloor m \rfloor) y_{(\lceil m \rceil)}$ if m is not an integer. Thus $M = y_{(1+\overline{X}_{median}(N_1 \ldots N_k - 1))}$ is the MEDIAN over the order statistics: $y_{(1)} \le y_{(2)} \le \ldots \le y_{(N_1 \ldots N_k)}$. To estimate the value of M over a cross-product sample of $n_1 \ldots n_k$ elements, we can estimate $\hat{\overline{X}}_{median} = \bar{x}_{median} = \frac{1}{n_1 \ldots n_k} \sum_{i_1,\ldots,i_k}^{S_1,\ldots,S_k} x_{i_1,\ldots,i_k}$, and use the following equal events $\{\bar{x}_{median} = j / n_1 \ldots n_k\} = \{\hat{M} = y_{(1+\bar{x}_{median}(n_1 \ldots n_k - 1))} = y_{(1+j-j/n_1 \ldots n_k)}\}$ to derive $\hat{M}$. However, since M is unknown, we simply cannot decide which $x_{i_1,\ldots,i_k}$ is 1, 0.5, or 0. In practice, we simply take $\bar{x}'_{median} = E(\bar{x}_{median}) = \overline{X}_{median} = 0.5$ to get the estimated $M : \hat{M} = y_{(0.5 n_1 \ldots n_k + 0.5)}$ over the order statistics: $y_{(1)} \le y_{(2)} \le \ldots \le y_{(n_1 \ldots n_k)}$. Furthermore, let $x'_{i_1,\ldots,i_k}$ be 1 if $y_{i_1,\ldots,i_k} < \hat{M}$, 0.5 if $y_{i_1,\ldots,i_k} = \hat{M}$, or 0 otherwise; and $\bar{x}'_{median} = \frac{1}{n_1 \ldots n_k} \sum_{i_1,\ldots,i_k}^{S_1,\ldots,S_k} x'_{i_1,\ldots,i_k}$. The confidence interval for M at a confidence level (normally 95%) can be defined as $[y_{(l)},\ y_{(u)}]$, where $y_{(l)}$ and $y_{(u)}$ are estimated from the sample of the $n_1 \ldots n_k$ elements. To obtain the values of l and u, we need to compute the variance of $\bar{x}_{median}$: $V(\bar{x}_{median}) = E(\bar{x}_{median} - \overline{X}_{median})^2$. Since $\overline{X}_{median}$ is known (0.5), we need to switch $\overline{X}_{median}$ and $\bar{x}_{median}$, so that with 100(1-α)% confidence level, $\bar{x}_{median}$ lies in $[\overline{X}_{median} - Z_{\alpha/2}\sqrt{V(\bar{x}_{median})},\ \overline{X}_{median} + Z_{\alpha/2}\sqrt{V(\bar{x}_{median})}]$. Therefore,

$$[l,\ u] = [1 + (\overline{X}_{median} - Z_{\alpha/2}\sqrt{V(\bar{x}_{median})})(n_1 \ldots n_k - 1),\ 1 + (\overline{X}_{median} + Z_{\alpha/2}\sqrt{V(\bar{x}_{median})})(n_1 \ldots n_k - 1)]$$

where $V(\bar{x}_{median})$ has to be estimated by $\hat{V}(\bar{x}_{median})$. However since we don't know the exact value of M, $\hat{V}(\bar{x}_{median})$ is simply replaced with $\hat{V}(\bar{x}'_{median})$. Thus $\hat{V}(\bar{x}'_{median})$ is an approximate estimator of $V(\bar{x}_{median})$.

Note that one major difference between cross-product sampling and sampling in the final stage [4] is that the variance in the former case has to be computed by using the techniques described in Section 3, because many factors in the variance are unknown prior to query, or the knowledge of the variance is absent without trials. In contrast, the variance in the case of sampling in the final stage is relatively simple. For example, under the simple random sampling without replacement, the variance of $\bar{x}_{median}$ is simply given by $0.25(N-n)/((N-1)n)$.

4.2 MEDIAN with Selection

When there are predicates, 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 $M : \hat{M} = y_{(1+0.5(w-1))}$ over the w elements. An approximate confidence interval is calculated as follows:

Let $x'_{i_1,\ldots,i_k}$ be 1 if $y_{i_1,\ldots,i_k} < \hat{M}$ and $y_{i_1,\ldots,i_k}$ is selected, 0.5 if $y_{i_1,\ldots,i_k} = \hat{M}$ and $y_{i_1,\ldots,i_k}$ is selected, or 0 otherwise, and

[1] For the simplicity of our presentation, no duplicates at M are assumed. This assumption also applies to $\hat{M}$ and `QUANTILE`. When there are duplicates, the actual value in this equation is computed by $(N_1 \ldots N_k / 2 - COUNT(y_{i_1,\ldots,i_k} < M)) / COUNT(y_{i_1,\ldots,i_k} = M)$.

$\bar{x}'_{median} = \frac{1}{n_1 \ldots n_k} \sum_{i_1,\ldots,i_k}^{S_1,\ldots,S_k} x'_{i_1,\ldots,i_k}$. Note that $\bar{x}'_{median}$ is equal to $0.5w/(n_1 \ldots n_k)$, different from the value of 0.5 in Section 4.1. Therefore, we can have the confidence interval $[y_{(l)},\ y_{(u)}]$ at an approximate 100(1-α)% confidence level, where l and u are defined as:

$$\begin{cases} l = 1 + \left(0.5 - Z_{\alpha/2}\sqrt{\hat{V}(\bar{x}'_{median})}\, n_1 \ldots n_k / w\right)(w-1) \\ u = 1 + \left(0.5 + Z_{\alpha/2}\sqrt{\hat{V}(\bar{x}'_{median})}\, n_1 \ldots n_k / w\right)(w-1) \end{cases}.$$

Note that under the Bernoulli (coin-flip) sampling scheme, $w/n_1 \ldots n_k$ is an approximately unbiased estimator of the population proportion $W/N_1 \ldots N_k$, where W elements will be selected from the population of $N_1 \ldots N_k$ elements. But W is unknown to our system, because the whole population of $N_1 \ldots N_k$ 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 $\hat{V}(\bar{x}'_{median})$ in the context of $n_1 \ldots n_k$ 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 φ `QUANTILE` can return the element in position 1+φ(N-1) in the sorted sequence of N elements. `MEDIAN` is the 50% `QUANTILE`. Let Q be the required φ `QUANTILE`. So we can simply use formulas described in Sections 4.1, and 4.2, and replace 0.5 and $\hat{V}(\bar{x}'_{median})$ with φ and $\hat{V}(\bar{x}'_{\varphi})$ 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 100γ% 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-α)% confidence that at least the proportion 100γ% of the population will exceed the `MIN` (or be less than `MAX`) in the sample.

To compute the confidence level that the proportion γ=95% of the population will exceed the `MIN` (or be less than the `MAX`), we compare the lower bound of φ=5% `QUANTILE` with the `MIN`, (or compare the upper bound of φ=95% `QUANTILE` with the `MAX`).

Assume a positive Z_α satisfies $\Phi(Z_\alpha) = 1 - \alpha$, where we directly use α because we only use one-sided limits to compute $\Pr(y_{(l)} \geq MIN)$ for $\varphi = 5\%$, and $\Pr(y_{(u)} \leq MAX)$ for $\varphi = 95\%$. For example, in the case without selection, Z_α can be computed as follows:

$$\begin{cases} l = 1 + \left(0.05 - Z_\alpha\sqrt{\hat{V}(\bar{x}'_{0.05})}\right)(n_1 \ldots n_k - 1) = 1 \\ u = 1 + \left(0.95 + Z_\alpha\sqrt{\hat{V}(\bar{x}'_{0.95})}\right)(n_1 \ldots n_k - 1) = n_1 \ldots n_k \end{cases} \Rightarrow$$

$$\begin{cases} Z_\alpha = 0.05 / \sqrt{\hat{V}(\bar{x}'_{0.05})} & for \quad MIN \\ Z_\alpha = 0.05 / \sqrt{\hat{V}(\bar{x}'_{0.95})} & for \quad MAX \end{cases}.$$

Note that normally $Z_\alpha > 0$. Once we obtain Z_α, 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 $y_{i_1,\ldots,i_k} = \sum_{j_1,\ldots,j_k}^{SB_{i_1},\ldots,SB_{i_k}} y_{i_1 j_1,\ldots,i_k j_k}$, where $y_{i_1 j_1,\ldots,i_k j_k}$ is the value obtained from the unit after joining the j_1-th row in the i_1-th block SB_{i_1} of the sample S_1 that has m_1 blocks chosen from M_1 blocks of the first table R_1 under Bernoulli sampling, … , and the j_k-th row in the i_k-th block SB_{i_k} of the sample S_k that has m_k blocks chosen from M_k blocks of the k-th table R_k under Bernoulli sampling. Then take $\hat{Y}_B = M_1 \ldots M_k \sum_{i_1,\ldots,i_k}^{S_1,\ldots,S_k} y_{i_1,\ldots,i_k} / m_1 \ldots m_k$ as an approximately unbiased estimator of Y, or SUM, which is similar to $\hat{Y}$ 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

[1] P. J. Haas, J. F. Naughton, S. Seshadri, A. N. Swami, "Selectivity and Cost Estimation for Joins Based on Random Sampling," *J. Comput. Syst. Sci.* 52(3), pp. 550-569, 1996.

[2] W.-C. Hou, G. Özsoyoglu, B. K. Taneja, "Processing Aggregate Relational Queries with Hard Time Constraints," *SIGMOD* 1989, pp. 68-77.

[3] Y. Hu, S. Sundara, J. Srinivasan, "Supporting Time-Constrained SQL Queries in Oracle," *VLDB* 2007, pp. 1207-1218.

[4] G. S. Manku, S. Rajagopalan, B. G. Lindsay, "Approximate Medians and other Quantiles in One Pass and with Limited Memory," *SIGMOD* 1998, pp. 426-435.