6

Inference Controls

When information derived from confidential data must be declassified for wider distribution, simple flow controls as described in the previous chapter are inadequate. This is true of statistical databases, which can contain sensitive information about individuals or companies. The objective is to provide access to statistics about groups of individuals, while restricting access to information about any particular individual. Census bureaus, for example, are responsible for collecting information about all citizens and reporting this information in a way that does not jeopardize individual privacy.

The problem is that statistics contain vestiges of the original information. By correlating different statistics, a clever user may be able to deduce confidential information about some individual. For example, by comparing the total salaries of two groups differing only by a single record, the user can deduce the salary of the individual whose record is in one group but not in the other. The objective of inference controls is to ensure that the statistics released by the database do not lead to the disclosure of confidential data.

Although many databases are used for statistics only (e.g., census data), general-purpose database systems may provide both statistical and nonstatistical access. In a hospital database, for example, doctors may be given direct access to patients' medical records, while researchers are only permitted access to statistical summaries of the records. Although we are primarily interested in protection mechanisms for general-purpose systems, we shall also describe mechanisms for statistics-only databases.

6.1 STATISTICAL DATABASE MODEL

We shall describe a statistical database in terms of an abstract model.† Although the model does not accurately describe either the logical or physical organization of most database systems, its simplicity allows us to focus on the disclosure problem and compare different controls.

6.1.1 Information State

The **information state** of a statistical database system has two components: the data stored in the database and external knowledge. The database contains information about the attributes of N individuals (organizations, companies, etc.). There are M **attributes** (also called **variables**), where each attribute A_j ($1 \leq j \leq M$) has $|A_j|$ possible values. An example of an attribute is *Sex*, whose two possible values are *Male* and *Female*. We let x_{ij} denote the value of attribute j for individual i. When the subscript j is not important to the discussion, we shall write simply x_i to denote the value of an attribute A for individual i.

It is convenient to view a statistical database as a collection of N records, where each record contains M fields, and x_{ij} is stored in record i, field j (see Figure 6.1). Note that this is equivalent to a relation (table) in a relational database [Codd70,Codd79], where the records are M-tuples of the relation. If the information stored in the database is scattered throughout several relations, then the relation depicted in Figure 6.1 corresponds to the "natural join" of these relations. We shall assume that each field is defined for all individuals, and that each individual has a single record.

Example:
Table 6.1 shows a (sub) database containing $N = 13$ confidential student records for a hypothetical university having 50 departments. Each record has $M = 5$ fields (excluding the identifier), whose possible values are shown in Table 6.2. The attribute *SAT* specifies a student's average on the *SAT* (Scholastic Aptitude Test) and *GP* specifies a student's current grade-point. Unless otherwise stated, all examples refer to this database. ■

External knowledge refers to the information users have about the database. There are two broad classes of external knowledge: working knowledge and supplementary knowledge. **Working knowledge** is knowledge about the attributes represented in the database (e.g., the information in Table 6.2) and the types of

†Jan Schlörer, Elisabeth Wehrle, and myself [Denn82] have extended the model described here, showing how a statistical database can be viewed as a lattice of tables, and how different controls can be interpreted in terms of the lattice structure. Because this work was done after the book had gone into production, it was not possible to integrate it into this chapter.

FIGURE 6.1 Abstract view of a statistical database.

Record	A_1	...	A_j	...	A_M
1	x_{11}	...	x_{1j}	...	x_{1M}
	.		.		.
	.		.		.
	.		.		.
i	x_{i1}	...	x_{ij}	...	x_{iM}
	.		.		.
	.		.		.
	.		.		.
N	x_{N1}	...	x_{Nj}	...	x_{NM}

TABLE 6.1 Statistical database with $N = 13$ students.

Name	Sex	Major	Class	SAT	GP
Allen	*Female*	*CS*	1980	600	3.4
Baker	*Female*	*EE*	1980	520	2.5
Cook	*Male*	*EE*	1978	630	3.5
Davis	*Female*	*CS*	1978	800	4.0
Evans	*Male*	*Bio*	1979	500	2.2
Frank	*Male*	*EE*	1981	580	3.0
Good	*Male*	*CS*	1978	700	3.8
Hall	*Female*	*Psy*	1979	580	2.8
Iles	*Male*	*CS*	1981	600	3.2
Jones	*Female*	*Bio*	1979	750	3.8
Kline	*Female*	*Psy*	1981	500	2.5
Lane	*Male*	*EE*	1978	600	3.0
Moore	*Male*	*CS*	1979	650	3.5

TABLE 6.2 Attribute values for Table 6.1.

Attribute A_j	Values	$\lvert A_j \rvert$
Sex	*Male*, *Female*	2
Major	*Bio*, *CS*, *EE*, *Psy*, ...	50
Class	1978, 1979, 1980, 1981	4
SAT	310, 320, 330, ..., 790, 800	50
GP	0.0, 0.1, 0.2, ..., 3.9, 4.0	41

statistics available. **Supplementary knowledge** is information that is not normally released by the database. This information may be confidential (e.g., a particular student's *GP* or *SAT* score) or nonconfidential (e.g., the student's sex).

6.1.2 Types of Statistics

Statistics are computed for subgroups of records having common attributes. A subgroup is specified by a **characteristic formula** C, which, informally, is any logical formula over the values of attributes using the operators "or" (+), "and" (•), and "not" (~), where the operators are written in order of increasing priority. An example of a formula is

$$(Sex = Male) \bullet ((Major = CS) + (Major = EE)) \ ,$$

which specifies all male students majoring in either CS or EE. We shall omit attribute names where they are clear from context, e.g., "$Male \bullet (CS + EE)$". We shall also use relational operators in the specification of characteristics, since these are simply abbreviations for the "or" of several values; for example, "$GP > 3.7$" is equivalent to "$(GP = 3.8) + (GP = 3.9) + (GP = 4.0)$".

The set of records whose values match a characteristic formula C is called the **query set** of C. For example, the query set of C = "$Female \bullet CS$" is {1, 4}, which consists of the records for Allen and Davis. We shall write "C" to denote both a formula and its query set, and $|C|$ to denote the number of records in C (i.e., the size of C). We denote by All a formula whose query set is the entire database; thus $C \subseteq All$ for any formula C, where "$\subseteq$" denotes query set inclusion.

Given $|A_j|$ values for each of the M attributes A_j ($j = 1, \ldots, M$), there are

$$E = \prod_{j=1}^{M} |A_j|$$

possible distinguishable records described by formulas of the form

$$(A_1 = a_1) \bullet \ldots \bullet (A_M = a_M) \ ,$$

where a_j is some value of attribute A_j. The query set corresponding to a formula of this form is called an **elementary set** because it cannot be further decomposed. The records in an elementary set (if any) are indistinguishable. Thus there are E elementary sets in the database, some of which may be empty. We let g denote the maximum size of all elementary sets; thus g is the maximum number of individuals having identical records, that is, the size of the largest indecomposable query set. If the number of records N satisfies $N \leq E$, then every individual may be identified by a unique elementary set, giving $g = 1$.

Example:
If we allow queries over all five attributes in Table 6.1, E = (2) (50) (4) (50) (41) = 820,000; $g = 1$ because each record is uniquely identifiable. If queries are restricted to the attributes Sex, $Major$, and $Class$, then $E = 400$; $g = 2$ because two students have the common characteristic "$Male \bullet EE \bullet 1978$". ■

Statistics are calculated over the values associated with a query set C. The simplest statistics are **counts** (**frequencies**) and **sums**:

$$\textbf{count}(C) = |C|$$
$$\textbf{sum}(C, A_j) = \sum_{i \epsilon C} x_{ij} \ .$$

Example:
count(*Female* • *CS*) = 2, and **sum**(*Female* • *CS*, *SAT*) = 1400. ■

Note that sums apply only to numeric data (e.g., *GP*, and *SAT*). The responses from counts and sums are used to calculate **relative frequencies** and **averages (means)**:

$$\textbf{rfreq}(C) = \frac{\textbf{count}(C)}{N} = \frac{|C|}{N}$$
$$\textbf{avg}(C, A_j) = \frac{\textbf{sum}(C, A_j)}{|C|}$$

Example:
$$\textbf{avg}(Female \bullet CS, SAT) = \frac{1400}{2} = 700. \ \blacksquare$$

More general types of statistics can be expressed as **finite moments** of the form:

$$q(C, e_1, \ldots, e_M) = \sum_{i \epsilon C} x_{i1}^{e1} x_{i2}^{e2} \ldots x_{iM}^{eM} \ , \tag{6.1}$$

where the exponents $e_1, \ldots, e_M$ are nonnegative integers. Note that counts and sums can be expressed as moments:

$$\textbf{count}(C) = q(C, 0, \ldots, 0)$$
$$\textbf{sum}(C, A_j) = q(C, 0, \ldots, 0, 1, 0, \ldots, 0) \ ,$$

where the jth exponent for the **sum** is 1, and all other exponents are 0. Means, variances, covariances, and correlation coefficients can also be computed. For example, the **mean** and **variance** of attribute A_1 are given by:

$$\overline{A}_1 = \textbf{avg}(C, A_1) = \frac{q(C, 1, 0, \ldots, 0)}{|C|}$$

$$\sigma_1^2 = \textbf{var}(C, A_1) = \frac{q(C, 2, 0, \ldots, 0)}{|C| - 1} - (\overline{A}_1)^2 \ .$$

The **covariance** of attributes A_1 and A_2 is given by

$$\sigma_{12}^2 = \textbf{covar}(C, A_1, A_2) = \frac{q(C, 1, 1, 0, \ldots, 0)}{|C| - 1} - \overline{A}_1 \overline{A}_2$$

and the **correlation coefficient** of A_1 and A_2 is

$$\rho_{12} = \textbf{corcoef}(C, A_1, A_2) = \frac{\sigma_{12}^2}{\sigma_1 \sigma_2} \ .$$

By $q(C)$ we shall mean any statistic, or **query** for a statistic, of the form (6.1).

Another type of statistic selects some value (smallest, largest, median, etc.) from the query set. We shall write

median(C, A_j)

to denote the **median** or $\lceil |C|/2 \rceil$ largest value of attribute A_j in the query set C, where "$\lceil\ \rceil$" denotes the ceiling (round up to nearest integer). Note that when the query-set size is even, the median is the smaller of the two middle values, and not their average.

Example:
The set of *GP*s for all female students is {2.5, 2.5, 2.8, 3.4, 3.8, 4.0}; thus **median**(*Female*, *GP*) = 2.8. ■

Statistics derived from the values of m distinct attributes are called ***m*-order statistics**. The attributes can be specified by terms in the characteristic formula C, or by nonzero exponents e_j in Formula (6.1). There is a single 0-order statistic, namely **count**(*All*). Examples of 1-order statistics are **count**(*Male*) and **sum**(*All*, *GP*). Examples of 2-order statistics are **count**(*Male* • *CS*) and **sum**(*Male*, *GP*). Note that **count**(*EE* + *CS*) is a 1-order statistic because *CS* and *EE* are values of the same attribute.

6.1.3 Disclosure of Sensitive Statistics

A statistic is **sensitive** if it discloses too much confidential information about some individual (organization, company, etc.). A statistic computed from confidential information in a query set of size 1 is always sensitive.

Example:
The statistic

sum(*EE* • *Female*, *GP*) = 2.5

is sensitive, because it gives the exact grade-point of Baker, the only female student in *EE*. ■

A statistic computed from confidential information in a query set of size 2 may also be classified as sensitive, because a user with supplementary knowledge about one of the values may be able to deduce the other from the statistic. The exact criterion for sensitivity is determined by the policies of the system. One criterion used by the U.S. Census Bureau for sums of economic data is the ***n*-respondent, *k*%-dominance** rule, which defines a **sensitive** statistic to be one where n or fewer values contribute more than k% of the total [Cox80].

Example:
A statistic giving the sum of the exact earnings of IBM and all early music

stores in Indiana would be sensitive under a 1-respondent, 99%-dominance criterion; its release would disclose a considerable amount of information about IBM's earnings (namely the high-order digits), though it would disclose little about the early music stores (which would be hidden in the low-order digits). ■

Clearly, all sensitive statistics must be restricted (i.e., not permitted). In addition, it may be necessary to restrict certain nonsensitive statistics if they could lead to disclosure of sensitive ones.

Example:
Suppose that the only statistics classified as sensitive in the sample database are those computed from query sets of size 1. Then neither **sum**(*EE, GP*) nor **sum**(*EE* • *Male, GP*) is sensitive. At least one of these statistics must be restricted, however, because if they are both released, Baker's grade-point is disclosed:

$$\begin{aligned}&\textbf{sum}(EE \bullet \textit{Female}, GP)\\ &\quad = \textbf{sum}(EE, GP) - \textbf{sum}(EE \bullet \textit{Male}, GP)\\ &\quad = \quad 12.0 \quad - \quad 9.5\\ &\quad = \quad 2.5\ . \quad \blacksquare\end{aligned}$$

Let R be a set of statistics released to a particular user, and let K denote the user's supplementary knowledge. **Statistical disclosure** [Haq75] occurs whenever the user can deduce from R and K something about a restricted statistic q; in terms of classical information theory (see Section 1.4),

$$H_{K,R}(q) < H_K(q)\ ,$$

where $H_K(q)$ is the equivocation (conditional entropy) of q given K, and $H_{K,R}(q)$ is the equivocation of q given K and R. Statistical disclosure of a sensitive statistic is sometimes called **residual** [Fell72] or **personal disclosure** (**compromise**) [Haq75]. If a disclosure occurs without supplementary knowledge, it is called **resultant disclosure**; if supplementary knowledge is necessary, it is called **external disclosure** [Haq75]. Supplementary knowledge is always required for personal disclosure to match the value disclosed with a particular individual.

Example:
Disclosure of **sum**(*EE* • *Female*, *GP*) can lead to personal disclosure of Baker's *GP* only if some user knows that Baker is a female student majoring in *EE*. The user must also know that Baker is the only female student in *EE*; this could be deduced from the statistic:

$$\textbf{count}(EE \bullet \textit{Female}) = 1$$

or, if this statistic is restricted (because it isolates a single individual), from

$$\textbf{count}(EE) - \textbf{count}(EE \bullet \textit{Male}) = 4 - 3 = 1\ . \quad \blacksquare$$

The **amount of information** (in bits) that a set of statistics R discloses about a statistic q is measured by the reduction of entropy: $H_K(q) - H_{K,R}(q)$.

A disclosure may be either exact or approximate. **Exact disclosure** occurs when q is determined exactly; thus, $H_{K,R}(q) = 0$. For example, the preceding disclosure of Baker's grade-point is exact.

Approximate disclosure occurs when q is not determined exactly. Dalenius describes three types of approximate disclosure [Dale77]. First, a disclosure may reveal **bounds** L and U such that $L \leq q \leq U$.

Example:
If it is known only that $\mathbf{count}(EE \bullet Female) \geq 1$, then release of the statistic $R = \mathbf{count}(EE) = 4$ implies

$$1 \leq \mathbf{count}(EE \bullet Female) \leq 4 ,$$

thereby reducing the uncertainty about $\mathbf{count}(EE \bullet Female)$ to

$$H_{K,R}(\mathbf{(count}(EE \bullet \mathbf{Female))} = 2$$

bits of information (assuming all counts in the range [1, 4] are equally likely) . ■

Example:
Release of the statistics

$$\begin{aligned} &\mathbf{count}(EE) && = 4 \\ &\mathbf{count}\big(EE \bullet (GP \geq 3.0)\big) && = 3 \\ &\mathbf{count}(EE \bullet Male) && = 3 \\ &\mathbf{count}\big(EE \bullet Male \bullet (GP \geq 3.0)\big) && = 3 \end{aligned}$$

reveals that

$$0 \leq \mathbf{sum}(EE \bullet Female, GP) < 3.0 \ . \quad ■$$

Second, a disclosure may be **negative** in the sense of revealing that $q \neq y$, for some value y. For example, a user may learn that $\mathbf{sum}(EE \bullet Female, GP) \neq 3.5$.

Third, a disclosure may be **probabilistic** in the sense of disclosing information that is true only with some probability. An example is **interval estimation**, where it is learned that q falls in some interval $[L, U]$ with probability p; that is,

$$Pr[q \in [L, U]] = p \ .$$

The interval $[L, U]$ is called the **confidence interval** for q, and the probability p the **confidence level**.

Example:
Suppose an estimate $\hat{q}$ of q is a random variable drawn from a distribution approximately normal with standard deviation $\sigma_{\hat{q}}$. We then have:

$$\begin{aligned} &Pr[q \in [\hat{q} \pm 1.645\sigma_{\hat{q}}]] \simeq .90 \\ &Pr[q \in [\hat{q} \pm 1.960\sigma_{\hat{q}}]] \simeq .95 \end{aligned}$$

$$Pr[q \in [\hat{q} \pm 2.575\sigma_{\hat{q}}]] \simeq .99 .$$

The interval $[\hat{q} \pm 1.645\sigma_{\hat{q}}]$, for example, is called the 90% confidence interval of q because it is 90% certain that q lies in this interval. ■

6.1.4 Perfect Secrecy and Protection

A statistical database provides **perfect secrecy** if and only if no sensitive statistic is disclosed. In practice, no statistical database can provide perfect secrecy, because any released statistic contains some information about the data used to compute it. Even the statistic **sum**(*All*, *GP*) contains some information about each student's grade-point, though it is difficult to extract a particular student's grade-point from it without additional statistics or supplementary information.

Our definition of perfect secrecy in statistical database systems is similar to the definition of perfect secrecy in cryptographic systems (see Section 1.4.2). But whereas it is a reasonable objective for cryptography, it is unreasonable for statistical databases—perfect secrecy would require that no information be released.

We are more interested in the difficulty of obtaining close approximations of confidential values. Given a sensitive statistic q, we say that q is **protected** from disclosure if and only if it is not possible to obtain an estimate $\hat{q}$ with confidence interval $[L_{\hat{q}}, U_{\hat{q}}]$ such that

$$Pr[q \in [L_{\hat{q}}, U_{\hat{q}}]] \geq p \quad \text{where} \quad (U_{\hat{q}} - L_{\hat{q}}) \leq k \tag{6.2}$$

for probability p and interval length k (p and k can depend on q); otherwise, it is **compromisable**. Clearly, any statistic is compromisable for a sufficiently large interval or sufficiently small probability; therefore, we are only concerned about disclosure for relatively small k, and p near 1.0. Disclosure (compromise) occurs when an estimate $\hat{q}$ satisfying Eq. (6.2) can be obtained from a released set of statistics R. Note that this covers all forms of approximate disclosure except for negative disclosure. It also covers exact disclosure; here $p = 1.0$ and $k = 0$.

Example:
Let q = **sum**($EE \bullet Female$, GP) = 2.5, p = .95, and k = 1.0. Then an estimate $\hat{q}$ = 2.0 such that

$$Pr[q \in [2.0 \pm 0.5]] \simeq .95$$

discloses q. Note that if a released set of statistics shows that q must lie in the interval [2.0, 3.0], then q is disclosed because [2.0, 3.0] is a 100% confidence interval for every estimate in the interval. ■

6.1.5 Complexity of Disclosure

If a statistic q is not protected, we would like to know the difficulty of obtaining an estimate $\hat{q}$ satisfying Eq. (6.2). This will be measured by the number $N_{\hat{q}}$ of released statistics that a user with supplementary knowledge K needs to obtain $\hat{q}$.

Note that $N_{\hat{q}}$ is similar to the unicity distance of a cipher (see Section 1.4.3); it is the number of statistics needed to reduce the uncertainty about q to an unacceptable level.

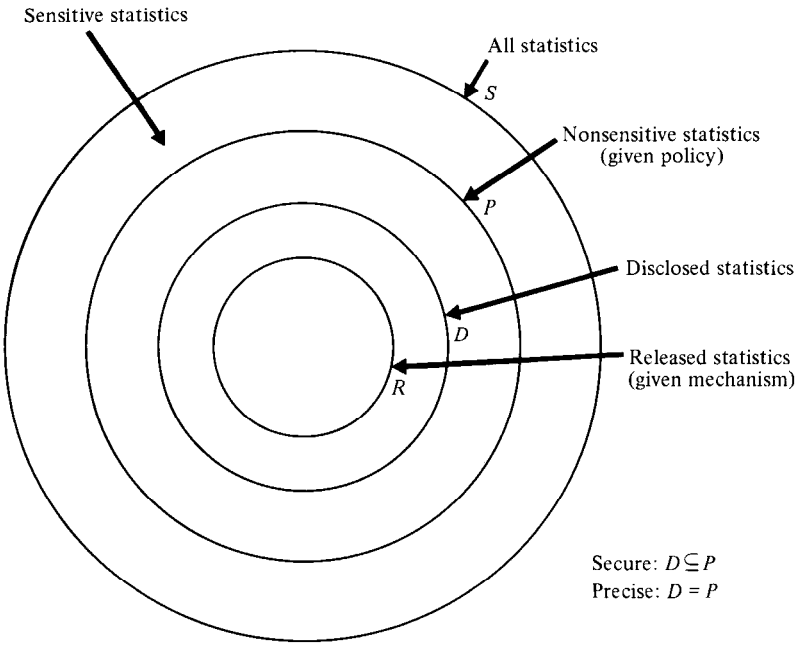
Frank [Fran77] has investigated another way of applying information theory to disclosure. He defines the **disclosure set** D to be the set of individuals whose attributes are known. Personal disclosure occurs when the size of this set increases by at least 1. The uncertainty $H_K(D)$ of D is a function of the frequency distribution of the variables in the database and a user's supplementary knowledge. This uncertainty decreases by $H_K(D) - H_{K,R}(D)$ when a set R of frequency distributions is released.

6.2 INFERENCE CONTROL MECHANISMS

6.2.1 Security and Precision

An inference control mechanism must protect all sensitive statistics. Let S be the set of all statistics, P the subset of S classified as nonsensitive, and R the subset of S released. Let D be the set of statistics disclosed by R (including the statistics in R). The statistical database is **secure** if $D \subseteq P$; that is, no sensitive statistic is disclosed by R.

FIGURE 6.2 Security and precision.



We would like the released set R to be complete in the sense that all nonsensitive statistics are in R or are computable from R (i.e., disclosed by R). A system in which $D = P$ is said to be **precise**. Whereas secrecy is required for privacy, precision is required for freedom of information. Figure 6.2 illustrates the requirements for security and precision; note the similarity of this figure with Figures 4.4 and 5.4.

The problem is that it can be extremely difficult to determine whether releasing a statistic will lead to disclosure of a sensitive statistic (violating security), or prevent the release of a complete set of statistics (violating precision). Most statistics lead to disclosure only when they are correlated with other statistics.

Example:
Although neither of the statistics **sum**(*EE*, *GP*) and **sum**(*EE* • *Male*, *GP*) is sensitive, if one is released, the other must be restricted to protect Baker's grade-point. Furthermore, it must be impossible to compute the restricted statistic from the set of released statistics. ■

This example shows that it is not generally possible to release a complete set of statistics; thus, any inference control mechanism must be imprecise. If we settle for releasing a maximal set of statistics, we find that the problem of determining a maximal set of statistics is **NP**-complete [Chin80].

Whether a statistic can lead to disclosure depends on a user's supplementary knowledge. Because it is not usually feasible to account for a particular user's supplementary knowledge, many mechanisms are based on a worst-case assumption about supplementary knowledge. A mechanism for the student record database, for example, might assume that a user knows the *Sex*, *Major*, and *Class* of every student, and the *GP* and *SAT* of some of the students.

To avoid restricting too many statistics, many statistical databases add "noise" to the data or to released statistics. The objective is to add enough noise that most nonsensitive statistics can be released without endangering sensitive ones—but not so much that the released statistics become meaningless.

6.2.2 Methods of Release

Many of the mechanisms depend on the method in which statistics are released. Census bureaus and other agencies that conduct population surveys have traditionally released statistics in two formats: macrostatistics and microstatistics.

Macrostatistics. These are collections of related statistics, usually presented in the form of 2-dimensional tables containing counts and sums.

Example:
Tables 6.3 and 6.4 show counts and total *SAT* scores for the student record database. The entries inside the tables give statistics for query sets defined by all possible values of *Sex* and *Class*. For example, the entry in row 1,

TABLE 6.3 Student counts by *Sex* and *Class*.

Sex	Class 1978	1979	1980	1981	Sum	
Female	1	2	2	1	6	
Male	3	2	0	2	7	
Sum	4	4	2	3	13	Total

TABLE 6.4 Total *SAT* scores by *Sex* and *Class*.

Sex	Class 1978	1979	1980	1981	Sum	
Female	800	1330	1120	500	3750	
Male	1930	1150	0	1180	4260	
Sum	2730	2480	1120	1680	8010	Total

column 3 gives the 2-order statistic **count**(*Female* • 1980) in Table 6.3, and the 3-order statistic **sum**(*Female* • 1980, *SAT*) in Table 6.4. The row sums give statistics for the query sets defined by *Sex*, and the column sums for the query sets defined by *Class*. For example, the sum for column 3 gives the 1-order statistic **count**(1980) in Table 6.3, and the 2-order statistic **sum** (1980, *SAT*) in Table 6.4. Finally, the total gives the 0-order statistic **count** (*All*) in Table 6.3 and the 1-order statistic **sum** (*All*, *SAT*) in Table 6.4. ■

Macrostatistics have the disadvantage of providing only a limited subset of all statistics. For example, it is not possible to compute correlations of *SAT* scores and grade-points from the data in Tables 6.3 and 6.4, or to compute higher-order statistics [e.g., **sum**(*Female* • *CS* • 1980, *SAT*)].

Because the set of released statistics is greatly restricted, macrostatistics provide a higher level of security than many other forms of release. Even so, it may be necessary to suppress certain cells from the tables or to add noise to the statistics.

Example:

Because Davis is the only female student in the class of 1978, the total *SAT* score shown in row 1, column 1 of Table 6.4 should be suppressed; otherwise, any user knowing that she is represented in the database can deduce her *SAT* score (the same holds for column 4, which represents Kline's *SAT* score). We shall return to this example and study the principles of cell suppression in Section 6.4.1. ■

Microstatistics. These consist of individual data records having the format shown in Figure 6.1. The data is typically distributed on tape, and statistical

evaluation programs are used to compute desired statistics. These programs have facilities for assembling (in main memory or on disk) query sets from the records on the tape, and for computing statistics over the assembled records. New query sets can be formed by taking a subset of the assembled records, or by assembling a new set from the tape.

Because no assumptions can be made about the programs that process the tapes, protection mechanisms must be applied at the time the tapes are created. Census bureaus control disclosure by

1. removing names and other identifying information from the records,
2. adding noise to the data (e.g., by rounding—see also discussion of privacy transformations in Section 3.5.2),
3. suppressing highly sensitive data,
4. removing records with extreme values,
5. placing restrictions on the size of the population for which microdata can be released, and
6. providing relatively small samples of the complete data.

The 1960 U.S. Census, for example, was distributed on tape as a random sample of 1 record out of 1000 with names, addresses, and exact geographical locations removed [Hans71]. A snooper would have at best a 1/1000 chance of associating a given sample record with the right individual.

Macrostatistics and microstatistics have been used for the one-time publication of data collected from surveys. Because they can be time-consuming and costly to produce, they are not well suited for the release of statistics in on-line database systems that are frequently modified.

Query-Processing Systems. The development of on-line query-processing systems has made it possible to calculate statistics at the time they are requested; released statistics, therefore, reflect the current state of the system. These systems have powerful **query languages**, which make it easy to access arbitrary subsets of data for both statistical and nonstatistical purposes. The data is logically and physically organized for fast retrieval, so that query sets can be constructed much more rapidly than from sequential files of records stored on tape or disk.

Because all accesses to the data are restricted to the query-processing programs, mechanisms that enforce access, flow, or inference controls can be placed in these programs. The final decision whether to release a statistic or to grant direct access to data can be made at the time the query is made.

Many of the methods used to protect macrostatistics and microstatistics are not applicable to these systems. Techniques that add noise to the stored data generally cannot be used, because accuracy of the data may be essential for nonstatistical purposes. Techniques such as cell suppression that involve costly and time-consuming computations cannot be applied on a per-query basis. Sampling techniques that use relatively small subsets of the database may not give sufficiently accurate statistics in small to medium size systems.

Because we are primarily interested in controls for query-processing systems,

most of the techniques discussed in this chapter are for these systems. A comprehensive discussion of the techniques used by government agencies to protect macrostatistics and microstatistics is given in [U.S.78].

We shall first study methods of attacking statistical databases, and then study techniques that reduce the threat of such attacks.

6.3 METHODS OF ATTACK

Before we can evaluate the effectiveness of existing and proposed inference controls, we must understand the threat. In this section, we shall examine several kinds of disclosure techniques. All the methods involve using released statistics and supplementary knowledge to solve a system of equations for some unknown.

6.3.1 Small and Large Query Set Attacks

Hoffman and Miller [Hoff70] showed that it is easy to compromise a database that releases statistics about small query sets. Suppose that a user knows an individual I who is represented in the database and who satisfies the characteristic formula C. If the user queries the database for the statistic **count**(C) and the system responds "1", then the user has identified I in the database and can learn whether I has an additional characteristic D by asking for the statistic **count**($C \bullet D$), where:

$$\textbf{count}(C \bullet D) = \begin{cases} 1 \text{ implies } I \text{ has } D \\ 0 \text{ implies } I \text{ does not have } D \end{cases}.$$

Similarly, the user can learn the value of attribute A for I by asking for the statistic **sum**(C, A).

Example:
Suppose a user knows Evans is represented in the student record database, and that Evans is a male biology student in the class of 1979. The statistic

$$\textbf{count}(Male \bullet Bio \bullet 1979) = 1$$

reveals Evans is the only such student. The statistic

$$\textbf{count}(Male \bullet Bio \bullet 1979 \bullet (Sat \geq 600)) = 0$$

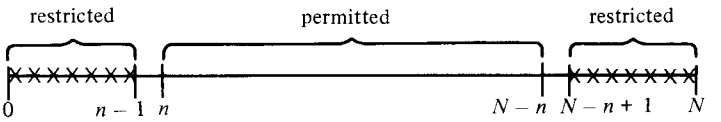
reveals that his SAT score is under 600, and the statistic

$$\textbf{sum}(Male \bullet Bio \bullet 1979, SAT) = 500$$

reveals his exact SAT score. ■

This type of attack may work even when an individual cannot be uniquely identified. Suppose that an individual I is known to satisfy C and **count**(C) > 1. If

FIGURE 6.3 Query-set size control.



count($C \bullet D$) = **count**(C), then I must also satisfy D; however, if **count**($C \bullet D$) < **count**(C), then nothing can be concluded about whether I satisfies D. (See also Haq [Haq74,Haq75].)

To protect against this kind of attack, statistics based on small query sets must be restricted. Because these statistics are normally classified as sensitive, their restriction would be automatic. If the query language permits complementation, large query sets must also be restricted (even though they are not sensitive); otherwise, users could pose their queries relative to the complement $\sim C$ of the desired characteristic C. Suppose as before that C uniquely identifies an individual I in the database; thus **count**($\sim C$) = $N - 1$, where N = **count**(All) is the size of the database. A user can determine whether C satisfies an additional characteristic D by posing the query **count**($\sim(C \bullet D)$), because

$$\textbf{count}(\sim(C \bullet D)) = \begin{cases} N & \text{implies } I \text{ does not have } D \\ N-1 & \text{implies } I \text{ has } D \end{cases} .$$

The user can learn the value of an attribute A for I from

$$\textbf{sum}(C, A) = \textbf{sum}(All, A) - \textbf{sum}(\sim C, A) .$$

In general, for any query $q(C)$ of the form (6.1),

$$q(C) = q(All) - q(\sim C) .$$

Query-Set-Size Control. These results show that any statistical database needs at least a mechanism that restricts query sets having fewer than n or more than $N - n$ records, for some positive integer n:

Query-Set-Size Control:
A statistic $q(C)$ is permitted only if

$$n \leq |C| \leq N - n ,$$

where $n \geq 0$ is a parameter of the database. ■

(See Figure 6.3.) Note that $n \leq N/2$ if any statistics at all are to be released. Note also that this restricts the statistics $q(All)$. In practice these statistics can be released; if they are restricted, they can be computed from $q(All) = q(C) + q(\sim C)$ for any C such that $n \leq |C| \leq N - n$.

6.3.2 Tracker Attacks

The query-set-size control provides a simple mechanism for preventing many trivial compromises. Unfortunately, the control is easily subverted. Schlörer [Schl75] showed that compromises may be possible even for n near $N/2$ by a simple snooping tool called the "tracker". The basic idea is to pad small query sets with enough extra records to put them in the allowable range, and then subtract out the effect of the padding. The following describes different types of trackers.

Individual Trackers. Schlörer considered statistics for counts that are released only for query-set sizes in the range $[n, N - n]$, where $n > 1$. Suppose that a user knows an individual I who is uniquely characterized by a formula C, and that the user seeks to learn whether I also has the characteristic D. Because $\mathbf{count}(C \bullet D) \leq \mathbf{count}(C) = 1 < n$, the previous method cannot be used to determine whether I also has the characteristic D. If C can be divided in two parts, the user may be able to calculate $\mathbf{count}(C \bullet D)$, however, from two answerable queries involving the parts.

Suppose that the formula C can be decomposed into the product $C = C1 \bullet C2$, such that $\mathbf{count}(C1 \bullet \sim C2)$ and $\mathbf{count}(C1)$ are both permitted:

$$n \leq \mathbf{count}(C1 \bullet \sim C2) \leq \mathbf{count}(C1) \leq N - n \ .$$

The pair of formulas $\{C1, C1 \bullet \sim C2\}$ is called the **individual tracker** (of I) because it helps the user to "track down" additional characteristics of I. The method of compromise is summarized as follows:

Individual Tracker Compromise:
Let $C = C1 \bullet C2$ be a formula uniquely identifying individual I, and let $T = C1 \bullet \sim C2$ [see Figure 6.4(a)]. Using the permitted statistics $\mathbf{count}(T)$ and $\mathbf{count}(T + C1 \bullet D)$, compute:

$$\mathbf{count}(C \bullet D) = \mathbf{count}(T + C1 \bullet D) - \mathbf{count}(T) \qquad \textbf{(6.3)}$$

[see Figure 6.4(b)]. If $\mathbf{count}(C \bullet D) = 0$, I does not have characteristic D. If $\mathbf{count}(C \bullet D) = \mathbf{count}(C)$, I has characteristic D. If $\mathbf{count}(C) = 1$, Palme [Palm74] showed that the value of an attribute A of I can be computed from

$$\mathbf{sum}(C, A) = \mathbf{sum}(C1, A) - \mathbf{sum}(T, A) \ .$$

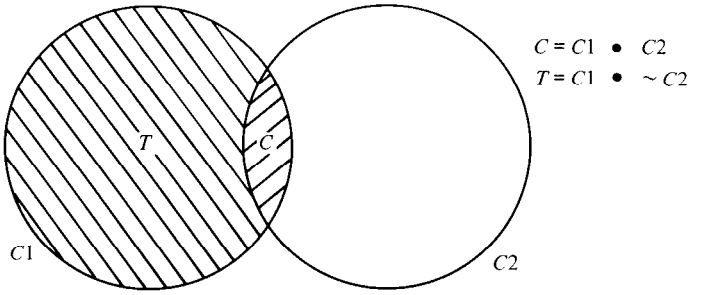
In general, if $q(C)$ is any restricted query for a finite moment of the form (6.1), then $q(C)$ can be computed from

$$q(C) = q(C1) - q(T) \ . \qquad \textbf{(6.4)}$$

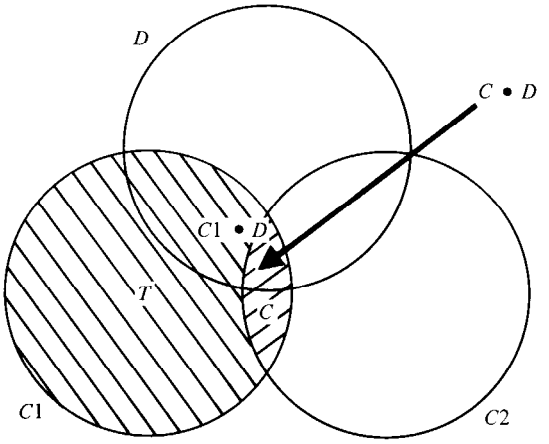
[See Figure 6.4(a).] If it is not known whether the formula C uniquely identifies I, Eq. (6.4) can be used to determine whether $\mathbf{count}(C) = 1$:

$$\mathbf{count}(C) = \mathbf{count}(C1) - \mathbf{count}(T) \ . \ \blacksquare$$

FIGURE 6.4 Individual tracker compromise.



a) $q(C) = q(C1) - q(T)$



b) $q(C \bullet D) = q(T + C1 \bullet D) - q(T)$

Example:
Evans is identified by the formula

$$C = Male \bullet Bio \bullet 1979 \ .$$

Let $n = 3$, $C1 = Male$, and $C2 = Bio \bullet 1979$. Then

$$\begin{aligned} T &= C1 \bullet \sim C2 \\ &= Male \bullet \sim (Bio \bullet 1979) \ . \end{aligned}$$

We can determine whether Evans is uniquely identified by C and whether his SAT score is at least 600 by applying Eqs. (6.4) and (6.3), where $D = (SAT \geq 600)$:

$$\begin{aligned} &\mathbf{count}(Male \bullet Bio \bullet 1979) \\ &\quad = \mathbf{count}(Male) - \mathbf{count}(Male \bullet \sim (Bio \bullet 1979)) \end{aligned}$$

$$
\begin{aligned}
&= 7 - 6 \\
&= 1 \\
&\mathbf{count}(Male \bullet Bio \bullet 1979 \bullet (SAT \geq 600)) \\
&\quad = \mathbf{count}(Male \bullet \sim (Bio \bullet 1979) + Male \bullet (SAT \geq 600)) \\
&\qquad - \mathbf{count}(Male \bullet \sim (Bio \bullet 1979)) \\
&\quad = 6 - 6 \\
&\quad = 0 .
\end{aligned}
$$

His GP can be determined by applying Eq. (6.4):

$$
\begin{aligned}
&\mathbf{sum}(Male \bullet Bio \bullet 1979, GP) \\
&\quad = \mathbf{sum}(Male, GP) - \mathbf{sum}(Male \bullet \sim (Bio \bullet 1979), GP) \\
&\quad = 22.2 - 20.0 \\
&\quad = 2.2 . \blacksquare
\end{aligned}
$$

This type of compromise is not prevented by lack of a decomposition of C giving query sets $C1$ and T in the range $[n, N - n]$. Schlörer pointed out that restricted sets $C1$ and T can often be replaced with permitted sets $C1 + CM$ and $T + CM$, where $\mathbf{count}(C1 \bullet CM) = 0$. The formula CM, called the "mask", serves only to pad the small query sets with enough (irrelevant) records to put them in the permitted range.

General Trackers. The individual tracker is based on the concept of using categories known to describe a certain individual to determine other information about that individual. A new individual tracker must be found for each person. Schwartz [Schw77] and Denning, Denning, and Schwartz [Denn79] showed that this restriction could be removed with "general" and "double" trackers. A single general or double tracker can be used to compute the answer to every restricted statistic in the database. No prior knowledge about anyone in the database is required (though some supplementary knowledge is still required for personal disclosure).

A **general tracker** is any characteristic formula T such that

$$2n \leq |T| \leq N - 2n .$$

Notice that queries $q(T)$ are always answerable, because $|T|$ is well within the range $[n, N - n]$ (see Figure 6.5).

Obviously, n must not exceed $N/4$ if a general tracker is to exist at all. Schlörer [Schl80] showed that if $g \leq N - 4n$, where g is the size of the largest elementary set (see Section 6.1.2), then the database must contain at least one general tracker.

FIGURE 6.5 General tracker.

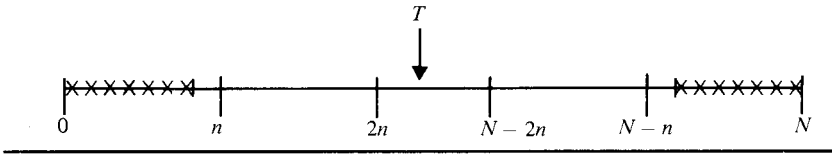
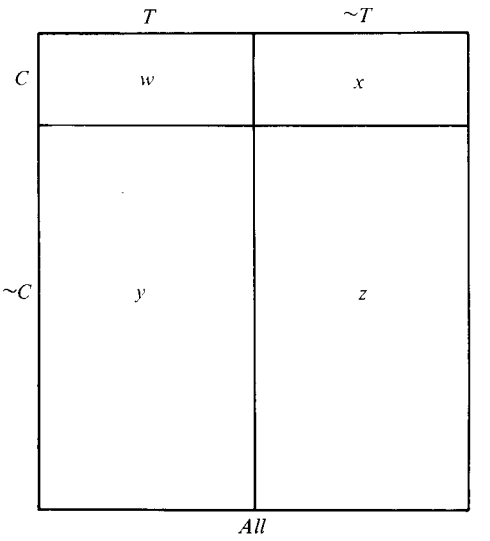


FIGURE 6.6 General tracker compromise.



$$q(All) = q(T) + q(\sim T) = w + x + y + z$$

$$q(C) = q(C + T) + q(C + \sim T) - q(All)$$

$$= (w + x + y) + (w + x + z) - (w + x + y + z)$$

$$= w + x$$

General Tracker Compromise:
Let T be a general tracker and let $q(C)$ be a restricted query for any finite moment of the form (6.1). First calculate

$$q(All) = q(T) + q(\sim T) \ .$$

If $|C| < n$, $q(C)$ can be computed from

$$q(C) = q(C + T) + q(C + \sim T) - q(All) \qquad \textbf{(6.5)}$$

(see Figure 6.6), and if $|C| > N - n$, $q(C)$ can be computed from

$$q(C) = 2q(All) - q(\sim C + T) - q(\sim C + \sim T) \ . \qquad \textbf{(6.6)}$$

If the user does not know whether the query set is too small or too large, Formula (6.5) can be tried first; if the queries on the right-hand side are permitted, the user can proceed; otherwise, Formula (6.6) can be used. Thus, $q(C)$ can be computed with at most five queries. ■

Example:
Let $n = 3$ in the student record database. To be answerable, a query set's size must fall in the range [3, 10], but a general tracker's query-set size must fall

in the subrange [6, 7]. Because $g = 1$ and $N - 4n = 13 - 12 = 1$, the database must contain at least one general tracker. The database actually contains several trackers; we shall use the tracker $T = Male$, where $|T| = 7$.

Suppose it is known that Jones satisfies the characteristic $C = Female \bullet Bio$, but it is not known whether C uniquely identifies her. The restricted statistic **count**($Female \bullet Bio$) can be computed from formula (6.5):

$$\begin{aligned}
&\textbf{count}(All) = \textbf{count}(Male) + \textbf{count}(\sim Male)\\
&\qquad = \quad 7 \qquad + \quad 6\\
&\qquad = \quad 13\\
&\textbf{count}(Female \bullet Bio)\\
&\quad = \quad \textbf{count}(Female \bullet Bio + Male)\\
&\qquad + \textbf{count}(Female \bullet Bio + \sim Male) - \textbf{count}(All)\\
&\quad = \qquad 8 \quad + \quad 6 \quad - \quad 13\\
&\quad = \qquad 1\ .
\end{aligned}$$

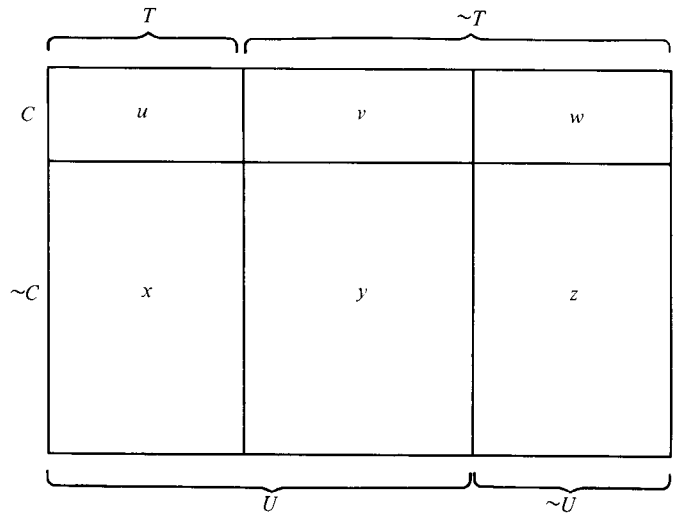
Because Jones is uniquely identified by C, her GP can be deduced by computing the restricted statistic **sum**($Female \bullet Bio$, GP):

$$\begin{aligned}
&\textbf{sum}(All, GP) = \textbf{sum}(Male, GP) + \textbf{sum}(\sim Male, GP)\\
&\qquad = \quad 22.2 \qquad + \quad 19.0\\
&\qquad = \quad 41.2\\
&\textbf{sum}(Female \bullet Bio, GP)\\
&\quad = \quad \textbf{sum}(Female \bullet Bio + Male, GP)\\
&\qquad + \textbf{sum}(Female \bullet Bio + \sim Male, GP) - \textbf{sum}(All, GP)\\
&\quad = \qquad 26.0 \quad + \quad 19.0 \quad - \quad 41.2\\
&\quad = \qquad 3.8\ . \ \blacksquare
\end{aligned}$$

Once a tracker is found, any restricted statistic can be computed with just a few queries. We might hope that finding trackers would be difficult, but unfortunately this is not the case. Denning and Schlörer [Denn80a] constructed an algorithm that finds a tracker within $O(\log_2 E)$ queries, where E is the number of elementary sets. The algorithm begins by taking a formula $C1$ such that $|C1| < 2n$, and a formula $C2 = All$. Then $C1$ is padded and $C2$ reduced until either $|C1|$ or $|C2|$ falls inside the range $[2n, N - 2n]$. The padding and reducing is done using a binary search strategy, whence convergence is in logarithmic time. The results of an experiment performed on a medical database showed that a tracker could often be found with just one or two queries. Schlörer [Schl80] also showed that large proportions of the possible queries in most databases are general trackers; thus, a general tracker is apt to be discovered quickly simply by guessing.

Double Trackers. We might hope to secure the database by restricting the range of allowable query sets even further. Because general trackers may exist for n near $N/4$, we must make $n > N/4$. Before we consider the security of such a strategy, let us examine its implications: $n = N/4$ already restricts half of all possible query-set sizes; even this is probably too large for most statistical applications. Any larger value of n is likely to seriously impair the utility of the database.

FIGURE 6.7 Double tracker compromise.



$$q(C) = q(U) + q(C+T) - q(T) - q(\sim(C \bullet T) \bullet U)$$
$$= (u+v+x+y) + (u+v+w+x) - (u+x) - (v+x+y)$$
$$= u+v+w$$

Nevertheless, we found that trackers can circumvent much larger values of n. If $n \leq N/3$, compromise may be possible using a double tracker. A **double tracker** is a pair of characteristic formulas (T, U) for which

$T \subseteq U$,
$n \leq |T| \leq N - 2n$, and
$2n \leq |U| \leq N - n$.

Double Tracker Compromise:
Let $q(C)$ be a query for a restricted statistic, and let (T, U) be a double tracker. If $|C| < n$, $q(C)$ can be computed from

$$q(C) = q(U) + q(C+T) - q(T) - q(\sim(C \bullet T) \bullet U) \quad \textbf{(6.7)}$$

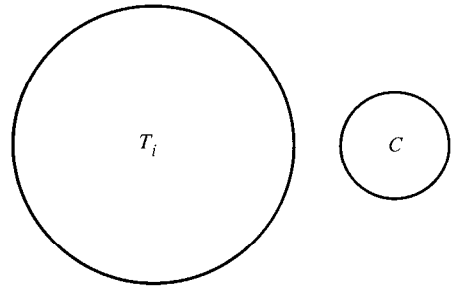
(see Figure 6.7), and if $|C| > N - n$, $q(C)$ can be computed from

$$q(C) = q(\sim U) - q(\sim C + T) + q(T) + q(\sim(\sim C \bullet T) \bullet U) \ . \quad \textbf{(6.8)}$$

Thus, $q(C)$ can be computed with at most seven queries. ■

Union Trackers. Schlörer generalized the concept of trackers to "union" trackers. Such trackers may exist even when n is near $N/2$, that is, when the only released statistics are those involving approximately half the population.

FIGURE 6.8 Union tracker compromise.



A **union tracker** is a set of $u \geq 2$ disjoint formulas $\{T_1, \ldots, T_u\}$ such that

$$n \leq |T_i| \leq N - n - g$$

for $i = 1, \ldots, u$, where g is the size of the largest elementary set. The formulas T_i can be used to compute any restricted statistic $q(C)$ when $n \leq N/2 - g$.

Union Tracker Compromise:
Let $q(C)$ be a restricted statistic, and let $\{T_1, \ldots, T_u\}$ be a union tracker. Split C into elementary sets $S_1, \ldots, S_t$ such that $C = S_1 + \ldots + S_t$. For each S_j, find a T_i such that $S_j \not\subseteq T_i$ (whence $S_j \cap T_i = \phi$), and compute

$$q(S_j) = q(T_i + S_j) - q(T_i) \ .$$

Because $|S_j| \leq g$, $q(T_i + S_j)$ is permitted for each j. Finally, compute

$$q(C) = q(S_1) + \cdots + q(S_t) \ .$$

Note that if C is a formula uniquely identifying some individual in the database, the preceding simplifies to

$$q(C) = q(T_i + C) - q(T_i) \ , \tag{6.9}$$

where $C \cap T_i = \phi$ (see Figure 6.8) . ■

In general, the method is more difficult to apply than general or double trackers, especially for n near $N/2 - g$. Still, it demonstrates that controls restricting only the sizes of query sets are doomed to failure.

6.3.3 Linear System Attacks

Let $q_1, \ldots, q_m$ be a set of released statistics of the form $q_i = \mathbf{sum}(C_i, A)$ $(1 \leq i \leq m)$. A **linear-system attack** involves solving a system of equations

$$H X = Q$$

for some x_j, where $X = (x_1, \ldots, x_N)$ and $Q = (q_1, \ldots, q_m)$ are column vectors, and $h_{ij} = 1$ if $j \in C_i$ and $h_{ij} = 0$ otherwise $(1 \leq i \leq m, 1 \leq j \leq N)$.

Example:
The queries $q_1 = \mathbf{sum}(Female, GP) = 19.0$ and $q_2 = \mathbf{sum}(Female + Male \bullet CS \bullet 1979, GP) = 22.5$ correspond to the linear system:

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \end{pmatrix} = \begin{pmatrix} 19.0 \\ 22.5 \end{pmatrix} .$$

Moore's GP can be compromised by solving for $x_{13} = \mathbf{sum}(Male \bullet CS \bullet 1979, GP) = q_2 - q_1 = 22.5 - 19.0 = 3.5$. ■

All the tracker attacks studied in the previous section are examples of linear-system attacks.

Key-Specified Queries. Dobkin, Jones, and Lipton [Dobk79] studied **key-specified queries** of the form $\mathbf{sum}(C, A)$, where C is specified by a set of k **keys** $\{i_1, \ldots, i_k\}$ identifying the records in the query set C. The value k is fixed for all queries, ruling out tracker attacks, which use different size query sets.

When disclosure is possible, the number of queries needed to achieve exact disclosure is no worse than linear in k.

Example:
Suppose $k = 3$ and consider the following queries:

$$\mathbf{sum}(\{1, 2, 3\}, A) = x_1 + x_2 + x_3 = q_1$$
$$\mathbf{sum}(\{1, 2, 4\}, A) = x_1 + x_2 + x_4 = q_2$$
$$\mathbf{sum}(\{1, 3, 4\}, A) = x_1 + x_3 + x_4 = q_3$$
$$\mathbf{sum}(\{2, 3, 4\}, A) = x_2 + x_3 + x_4 = q_4 .$$

These queries can be expressed as the following linear system:

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} .$$

The value x_1 can be compromised by computing

$$x_1 = \frac{1}{3}(q_1 + q_2 + q_3 - 2q_4) \ . \ \blacksquare$$

Schwartz, Denning, and Denning [Schw79,Schw77] extended these results to **weighted sums** of the form

$$\textbf{wsum}(\{i_1, \ldots, i_k\}, A) = \sum_{j=1}^{k} w_j x_{i_j} \ .$$

If the weights w_j are unknown, k of the x_i can be compromised within $k(k + 1)$ queries with supplementary knowledge of at least one x_j, and all x_i can be compromised within $N + k^2 - 1$ queries. Compromise is impossible, however, without supplementary knowledge, assuming none of the data values are zero [Schw79,Denn80b,Liu80].

Query-Set-Overlap Control. Dobkin, Jones, and Lipton were primarily concerned with the complexity of linear systems when queries are not allowed to overlap by more than a few records. They observed that many compromises, including the one preceding, use query sets that have a large number of overlapping records. Such compromises could be prevented with a mechanism that restricts query sets having more than r records in common:

> ***Query-Set-Overlap Control:***
> A statistic $q(C)$ is permitted only if $|C \bullet D| \leq r$ for all $q(D)$ that have been released, where $r > 0$ is a parameter of the system. ■

Now, implementing such a control is probably infeasible: before releasing a statistic, the database would have to compare the latest query set with every previous one.

The control would also seriously impair the usefulness of the database (i.e., it would be very imprecise), because statistics could not be released for both a set and its subsets (e.g., all males and all male biology majors). It would rule out publishing row and column sums in 2-dimensional tables of counts or aggregates.

What is interesting and somewhat surprising is that the control does not prevent many attacks. Let $m(N, k, r)$ be the minimum number of key-specified queries for groups of size k needed to compromise a value x_i in a database of N elements having an overlap restriction with parameter r. Dobkin, Jones, and Lipton showed that without supplementary knowledge, compromise is impossible when

$$N < \frac{k^2 - 1}{2r} + \frac{k + 1}{2} \ .$$

For $r = 1$ (i.e., any pair of query sets can have at most one record in common), compromise is, therefore, impossible when $N < k(k + 1)/2$. They showed that compromise is possible for $r = 1$ when $N \geq k^2 - k + 1$; the number of queries needed is bounded by:

$$k < m(N, k, 1) \leq 2k - 1 \ .$$

The possibility of compromising when

$$\frac{k(k+1)}{2} \leq N < k^2 - k + 1$$

was left open.

Example:
The following illustrates how x_7 can be compromised with five queries when $k = 3$ and $r = 1$:

$$\begin{aligned}
q_1 &= x_1 + x_2 + x_3 \\
q_2 &= \qquad\quad x_4 + x_5 + x_6 \\
q_3 &= x_1 \qquad + x_4 \qquad\quad + x_7 \\
q_4 &= \quad x_2 \qquad\quad + x_5 \quad + x_7 \\
q_5 &= \qquad x_3 \qquad\quad + x_6 + x_7 \ .
\end{aligned}$$

Then $x_7 = (q_3 + q_4 + q_5 - q_1 - q_2)/3$. ■

In general, x_{k^2-k+1} can be determined from the $2k - 1$ queries:

$$q_i = \sum_{j=1}^{k} x_{k(i-1)+j} \qquad i = 1, \ldots, k-1$$

$$q_{k+i-1} = \sum_{j=1}^{k-1} x_{k(j-1)+i} + x_{k^2-k+1} \qquad i = 1, \ldots, k \ .$$

Then

$$x_{k^2-k+1} = \frac{\sum_{i=1}^{k} q_{k+i-1} - \sum_{i=1}^{k-1} q_i}{k} \ .$$

Davida, Linton, Szelag, and Wells [Davi78] and Kam and Ullman [Kam77] have also shown that a minimum overlap control often can be subverted using key-specified queries.

Characteristic-Specified Queries. In on-line statistical database systems, users should not be allowed to specify groups of records by listing individual identifiers. Nevertheless, the preceding results give insight into the vulnerabilities of databases when groups of records are specified by characteristic formulas. On the one hand, we observe that if each record (individual) i in the database is uniquely identified by the characteristic C_i, then any key list $\{i_1, \ldots, i_k\}$ can be expressed as the characteristic formula $C_{i_1} + \ldots + C_{i_k}$. Therefore, if a database can be compromised with key-specified queries, it can be compromised with characteristic-specified queries. Caution must be taken in applying this result, however. Even though key-lists can be formulated as characteristics, it is not possible to do so without

knowledge of the characteristics identifying the individuals named. Without this supplementary knowledge, it is not possible to precisely control the composition of query sets as for key-specified queries. Thus, achieving compromise with characteristics may be considerably more difficult than with keys. In practice, a database may be safe from this type of attack, especially if the attack must be done manually without the aid of a computer to formulate and pose queries.

On the other hand, the key model assumes all query sets are the same size k, ruling out simple attacks based on trackers. Because these attacks depend on highly overlapping query sets, we could eliminate tracker attacks with a query-set-overlap control. The preceding results are significant in that they show there are other methods of attack that do not use overlapping query sets. Because an overlap control would also seriously impair the usefulness of the database and be expensive to implement, we must search for other kinds of controls.

Fellegi [Fell72] and Chin and Ozsoyoglu [Chin80] show it is possible to determine whether the response to a query, when correlated with the responses to earlier queries, could result in exact disclosure. Chin and Ozsoyoglu do this by recording a complete history (audit trail) of all queries about some confidential attribute in a binary matrix H having N columns and at most N linearly independent rows. Each column represents one individual, and the rows represent a basis for the set of queries deducible from the previously answered queries (i.e., the set D in Figure 6.2); thus, each query that has been answered or could be deduced is expressed as a linear combination of the rows of H. When a new query is asked, the matrix is updated so that if the query would compromise the jth individual, updating the matrix introduces a row with all zeros except for a 1 in column j; thus, the potential compromise is easily detected. The matrix can be updated in $O(N^2)$ time, so the method is tractable for small databases.

6.3.4 Median Attacks

Another type of attack uses queries that select some value from the query set. In this section we consider queries for medians.

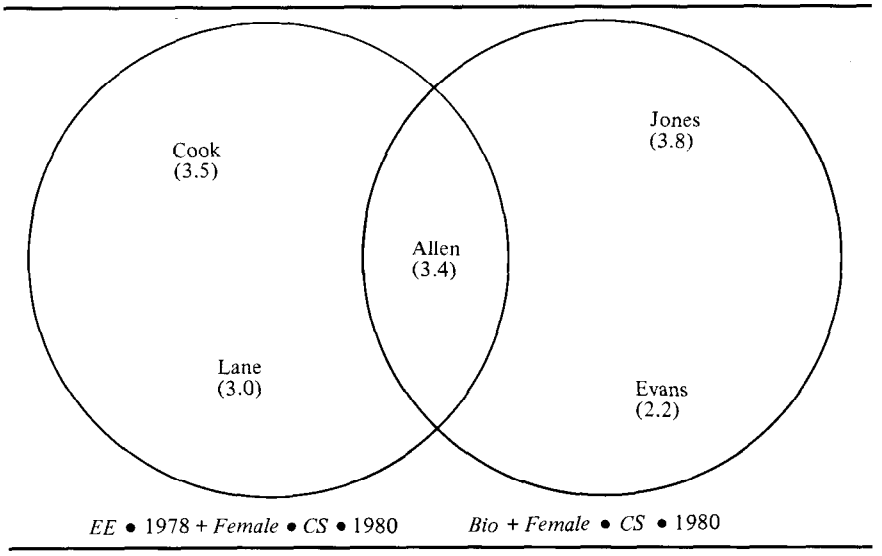
Example:
Suppose a user knows that Allen is the only student satisfying the formula (*Female* • *CS* • 1980). Consider these statistics:

median(*EE* • 1978 + *Female* • *CS* • 1980, *GP*) = 3.4
median(*BIO* + *Female* • *CS* • 1980, *GP*) = 3.4

Because both query sets have the same median and each student in these sets has a different *GP*, the *GP* of 3.4 must correspond to a student in both query sets. Because Allen is the only student in both sets, this must be Allen's grade-point (see Figure 6.9). ■

This example further demonstrates the futility of a query-set-overlap control—indeed, the attack exploits the fact that Allen is the only student in both sets.

FIGURE 6.9 Compromise with medians.



In general, let i be a record and C and D query sets such that:

1. $C \bullet D = \{i\}$,
2. **median**(C, A) = **median**(D, A), and
3. $x_j \neq x_{j'}$, for all $j, j' \in C \cup D, j \neq j'$.

Then x_i = **median**(C, A).

To employ this attack, it is necessary to find two query sets that have the same median and a single common record. DeMillo, Dobkin, and Lipton [DeMi77] define $m(k)$ to be the number of queries of the form **median**$(\{i_1, \ldots, i_k\}, A)$ needed to find query sets satisfying Properties (1)–(3) under an overlap restriction of $r = 1$, assuming no two values are the same. They show that compromise of some x_i is always possible within

$$m(k) = \begin{cases} k^2 + 1 & \text{if } k \text{ is a prime power} \\ 4(k^2 + 1) & \text{otherwise} \end{cases}$$

queries if $N \geq m(k) - 1$.

A curious aspect of this result is that it applies to any type of selection query—including those that lie! As long as the database always returns some value in the query set, compromise is possible with any two queries satisfying only Properties (1)–(3).

Example:

Consider these queries:

largest$(EE \bullet 1978 + Female \bullet CS \bullet 1980, GP)$ = 3.4 (lying)

$$\textbf{smallest}(Bio + Female \bullet CS \bullet 1980, GP) \qquad = 3.4 \text{ (lying)} \ .$$

By the same reasoning as before, the response 3.4 must be Allen's *GP*. ■

Compromise is even easier when there is no overlap control. Reiss [Reis78] shows that some element can always be found within $O(\log^2 k)$ queries, and that a specific element can usually be found within $O(\log k)$ queries with supplementary knowledge and within $O(k)$ queries without supplementary knowledge if its value is not too extreme. DeMillo and Dobkin [DeMi78] show that at least $O(\log k)$ queries are required, so

$$O(\log k) \leq m(k) \leq O(\log^2 k) \ .$$

6.3.5 Insertion and Deletion Attacks

Dynamic databases that allow insertions and deletions of records are vulnerable to additional attacks. Hoffman [Hoff77] observed that a query-set-size restriction of n can be subverted if records can be added to the database. If $|C| < n$, then dummy records satisfying C are added to the database; if $|C| > N - n$, then dummy records satisfying $\sim C$ are added. Of course, this type of attack presupposes that the user has permission to add new records to the database; a user with statistical access only cannot use this technique.

A second type of attack involves compromising newly inserted records. Let i be a new record satisfying a formula C, and consider the following sequence of operations:

$q_1 = q(C)$
insert (i)
$q_2 = q(C)$.

Then $q(i) = q_2 - q_1$. Chin and Ozsoyoglu [Chin79] show this threat can be eliminated by processing insertions and deletions in pairs.

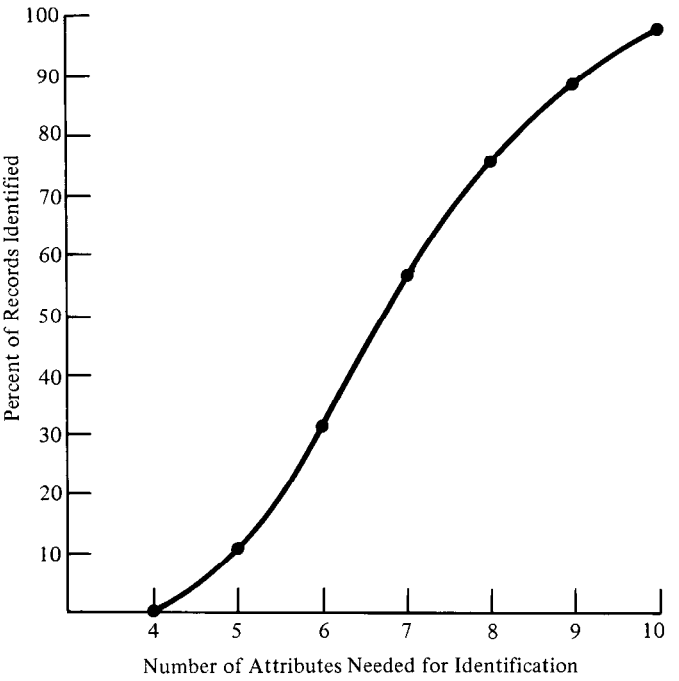
A third type of attack involves compromising an existing record i in a query set C by observing the changes to a statistic $q(C)$ when (pairs of) records are added to or deleted from C. If $|C|$ is odd, then i may be determined exactly (see exercises at end of chapter). Chin and Ozsoyoglu show that this threat can be eliminated by requiring that all query sets contain an even number of records (dummy records may be added to achieve this).

These attacks may not pose a serious threat if users with statistics-only access cannot insert and delete records, or otherwise control the changes being made to the database. Thus, many systems may not need controls that counter these attacks.

6.4 MECHANISMS THAT RESTRICT STATISTICS

We have studied two controls that restrict statistics that might lead to compromise: a query-set-size control and a query-set-overlap control. A size control, while

FIGURE 6.10 Identification of records in sample data.



extremely valuable and simple to implement, is insufficient. An overlap control is generally impractical, imprecise, and insufficient.

Another possibility is a **maximum-order control**, which restricts any statistic that employs too many attribute values. This would prevent compromises that require a large number of attributes to identify a particular individual. In a sample of 100 records drawn from a medical database containing over 30,000 records, Schlörer found that none of the records could be uniquely identified with fewer than 4 attributes, only 1 record could be identified with 4 attributes, about half of the records with 7 or fewer attributes, and nearly all with 10 attributes [Schl75] (see Figure 6.10). Thus, restricting queries to 3-order statistics might prevent most compromises in this database. Unfortunately, this can be overly restrictive because many of the higher-order statistics may be safe.

In the remainder of this section we examine three other possible controls aimed at restricting statistics that could lead to disclosure. The first is used by census bureaus to suppress statistics from tables of macrostatistics. While extremely effective, it is time-consuming and may not be suitable for on-line, dynamic databases. The second aims to decide at the time a query is posed whether release of the statistic could lead to compromise. The third partitions a database so that statistics computed over any partition are safe; queries about subsets of a partition are not permitted.

In [Denn82], we report on another restriction technique, called the $S_m/N -$

criterion.† The control, proposed by Schlörer [Schl76], restricts query sets over attributes that decompose the database into too many sets relative to the size N of the database (whence some of the sets are likely to be small). Formally, let C be a query set over attributes $A_1, \ldots, A_m$, and let $S_m = \prod_{i=1}^{m} |A_i|$ be the number of elementary sets over $A_1, \ldots, A_m$. A statistic $q(C)$ is restricted if

$$S_m/N > t$$

for some threshold t (e.g., $t = .1$). The control is extremely efficient and less restrictive than a maximum-order control. Although it does not guarantee security, it can be combined with a simple perturbation technique to provide a high level of security at low cost.

6.4.1 Cell Suppression

Cell suppression is one technique used by census bureaus to protect data published in the form of 2-dimensional tables of macrostatistics. It involves suppressing from the tables all sensitive statistics together with a sufficient number of nonsensitive ones, called **complementary suppressions**, to ensure that sensitive statistics cannot be derived from the published data. The sensitivity criterion for counts is typically a minimum query-set size. The sensitivity criterion for sums might be an n-respondent, k%-dominance rule, where a sensitive statistic is one in which n or fewer values contribute more than k% of the total.

Example:
Let us now return to Table 6.4 in Section 6.2.2. Suppose we have a (n, k) sensitivity rule, where $n = 1$ and $k = 90$. Clearly the entries in row 1,

TABLE 6.5 Total *SAT* scores by *Sex* and *Class*.

	Class					
Sex	1978	1979	1980	1981	Sum	
Female	—	1330	1120	—	3750	
Male	1930	1150	0	1180	4260	
Sum	2730	2480	1120	1680	8010	Total

TABLE 6.6 Total *SAT* scores by *Sex* and *Class*.

	Class					
Sex	1978	1979	1980	1981	Sum	
Female	—	1330	1120	—	3750	
Male	—	1150	0	—	4260	
Sum	2730	2480	1120	1680	8010	Total

†Because the study was performed after the book had gone into production, it is not possible to give details here.

columns 1 and 4 must be suppressed; in both cases, one student contributed 100% of the total *SAT* score. Table 6.5 shows these changes. Note that none of the other statistics in the table is sensitive.

Now, suppressing only these entries is insufficient, because they can be computed by subtracting the entries in the corresponding columns of row 2 from the column sums. Therefore, it is necessary to suppress either the entries in row 2 or the column sums; Table 6.6 shows the result of the former approach. The table is now safe from exact disclosure (see exercises at end of chapter). ■

It is easy to determine whether a statistic is sensitive by itself. Consider the statistic q = **sum**(C, A), and let d = **sum**(C, A, n) denote the sum of the n largest (dominant) values used to compute q. Thus, if $|C| = m$ and

$$q = x_1 + \cdots + x_n + \cdots + x_m \ ,$$

where

$$x_1 \geq \cdots \geq x_n \geq \cdots \geq x_m \ ,$$

then

$$d = x_1 + \cdots + x_n \ .$$

The statistic q is sensitive if $d > (k/100)q$; that is, if $q < q^+$, where $q^+ = (100/k)d$ (see Figure 6.11). Note that it is actually d that requires protection.

It is considerably more difficult to determine whether a nonsensitive statistic can be used to derive—exactly or approximately—a sensitive statistic. Following Cox [Cox76,Cox78], we first discuss acceptable bounds on estimates of sensitive statistics.

Let $\hat{q}$ be an estimate of a sensitive statistic q. Now if $\hat{q} \geq q^+$, then $\hat{q}$ does not reveal any information about q that would not have been released if q were not sensitive [i.e., if $q \geq (100/k)d$ were true]. Thus, a lower bound on an **acceptable upper estimate** of q is given by

$$q^+ = \left(\frac{100}{k}\right)d \ . \tag{6.10}$$

To determine an acceptable lower estimate, we assume that n, k, and m are known, where $m = |C|$ (in practice, these values are not usually disclosed). Observe that $d \geq (n/m)q$, for any statistic q, sensitive or not. Suppose q is not sensitive; that is, d lies in the interval $[(n/m)q, (k/100)q]$. If q is right at the sensitivity threshold, that is, $q = (100/k)d$, this interval is $[(n/m)\ (100/k)d, d]$ (see Figure 6.12). Thus, $q_d^- = (n/m)\ (100/k)d$ is an acceptable lower estimate of d

FIGURE 6.11 Restricted statistic q.

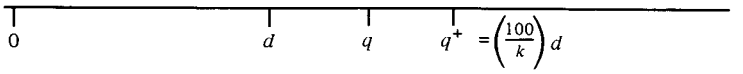
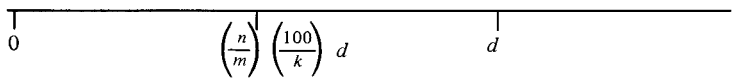


FIGURE 6.12 Interval for d at sensitivity threshold.



if q is sensitive, because q_d^- gives no more information about d than would be obtained if q were at the sensitivity threshold. Now, a lower estimate $L_{\hat{d}}$ of d can be obtained from a lower estimate $L_{\hat{q}}$ of q by

$$L_{\hat{d}} = \left(\frac{k}{100}\right) L_{\hat{q}} \ .$$

Therefore, an upper bound on an **acceptable lower estimate** of q is given by

$$q^- = \begin{cases} \left(\frac{n}{m}\right)\left(\frac{100}{k}\right)^2 d & \text{if } m > n \\ 0 & \text{if } m \leq n \ . \end{cases} \tag{6.11}$$

Example:
Let $k = 90$, $n = 2$, $m = 10$, $q = 950$, and $d = 900$. Then q is sensitive, because $950 < (100/90)900$. An acceptable upper estimate of q is $q^+ = (100/90)900 = 1000$, because q would then have been released. Now, if q were at the sensitivity threshold $(100/90)900 = 1000$, we could conclude d falls in the interval $[(2/10)\ (100/90)900, 900] = [200, 900]$. An acceptable lower estimate of q is thus $q^- = (2/10)\ (100/90)^2 900 = 222.2$, which gives a lower bound of 200 for d. ■

Cox explains it may be desirable to lower q^- so that q^+ and q^- are, respectively, subadditive and superadditive [Sand77]. The bound q^+ is **subadditive**, because the relation

$$q^+(C + D) \leq q^+(C) + q^+(D)$$

holds, where $q^+(C)$ denotes the acceptable upper estimate for the cell corresponding to $q(C)$. The bound q^- is not **superadditive**, however, because the relation

$$q^-(C) + q^-(D) \leq q^-(C + D)$$

does not hold (see exercises at end of chapter). Subadditivity and superadditivity reflect the principle that aggregation decreases the sensitivity of data; thus, we would expect the acceptable bounds to become tighter as data is aggregated.

Given q^- and q^+, an interval estimate $I = [L_{\hat{q}}, U_{\hat{q}}]$ of q is acceptable if I falls below q^-, above q^+, or strictly contains the interval $[q^-, q^+]$. (In the latter case, I is too large to be of much value.) Interval estimates are obtained for each sensitive statistic using linear algebra, and statistics that derive unacceptable estimates are suppressed from the tables.

TABLE 6.7 Table with suppressed cells.

x_{11}	6	x_{13}	25
8	x_{22}	x_{23}	30
x_{31}	x_{32}	3	20
20	30	25	75

Example:
Consider Table 6.7, where the variables x_{ij} denote sensitive entries that have been suppressed. The six unknowns are related by the following equations:

$$x_{11} + x_{13} = 25 - 6 = 19 \quad (1)$$
$$x_{22} + x_{23} = 30 - 8 = 22 \quad (2)$$
$$x_{31} + x_{32} = 20 - 3 = 17 \quad (3)$$
$$x_{11} + x_{31} = 20 - 8 = 12 \quad (4)$$
$$x_{22} + x_{32} = 30 - 6 = 24 \quad (5)$$
$$x_{13} + x_{23} = 25 - 3 = 22 \; . \quad (6)$$

Equations (1)–(6) imply

$$0 \le x_{11} \le 12 \quad \text{(by 1 and 4)} \quad (7)$$
$$0 \le x_{13} \le 19 \quad \text{(by 1 and 6)} \quad (8)$$
$$0 \le x_{22} \le 22 \quad \text{(by 2 and 5)} \quad (9)$$
$$0 \le x_{23} \le 22 \quad \text{(by 2 and 6)} \quad (10)$$
$$0 \le x_{31} \le 12 \quad \text{(by 3 and 4)} \quad (11)$$
$$0 \le x_{32} \le 17 \quad \text{(by 3 and 5)} \; . \quad (12)$$

These interval estimates can then be used to derive tighter bounds for x_{13}, x_{22}, x_{23}, and x_{32}. By Eq. (3),

$$x_{32} = 17 - x_{31} \; .$$

Because x_{31} is at most 12 [by Eq. (11)], we have

$$5 \le x_{32} \le 17 \; . \quad (13)$$

The bounds on x_{32}, in turn, affect the interval estimate for x_{22}. By Eq. (5),

$$x_{22} = 24 - x_{32} \; ;$$

thus Eq. (13) implies

$$7 \le x_{22} \le 19 \; . \quad (14)$$

By Eq. (2),

$$x_{23} = 22 - x_{22} \; ,$$

so Eq. (14) implies

$$3 \le x_{23} \le 15 \; . \quad (15)$$

Similarly, by Eq. (6),

TABLE 6.8 Interval estimates for Table 6.7.

0–12	6	7–19	25
8	7–19	3–15	30
0–12	5–17	3	20
20	30	25	75

$$x_{13} = 22 - x_{23} ,$$

so Eq. (15) implies

$$7 \leq x_{13} \leq 19 . \tag{16}$$

Equations (7), (11), and (13)–(16) give the best possible bounds on the unknowns (see Table 6.8). If any of the interval estimates is unacceptable, additional cells must be suppressed. ■

Cox [Cox78] gives a linear analysis algorithm that determines interval estimates for suppressed internal cells in a 2-dimensional table. The algorithm is a modification of the simplex algorithm of linear programming (e.g., see [Dant63]). If the analysis uncovers unacceptable interval estimates, additional cells are suppressed (the complementary suppressions), and the analysis repeated until all estimates are acceptable. The method does not always determine the minimum set of complementary suppressions; this is an open problem. Sande [Sand77] has developed a similar interval analysis procedure.

Cell suppression is limited by the computational complexity of the analysis procedure. Whereas it has been successfully applied to 2- and 3-dimensional tables, whether it could be adapted to query-processing systems for general-purpose databases is an open problem. The set of all possible statistics for a database with 10 fields of data in each record corresponds to a 10-dimensional table. Applying cell suppression to a table of this size may not be tractable.

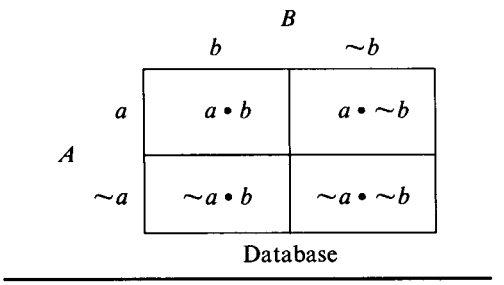
6.4.2 Implied Queries

Let us consider the possibility of dynamically applying a limited form of cell suppression in general-purpose databases when sensitivity is defined by a minimum query-set size. A statistic $q(C)$ is thus sensitive when $|C| < n$. For simplicity, we shall restrict attention to exact disclosure.

Let $q(C)$ be a sensitive statistic. Because $q(C)$ can be trivially computed from the relation $q(C) = q(All) - q(\sim C)$, it is necessary to suppress $q(\sim C)$ to protect $q(C)$ [we shall assume $q(All)$ is not restricted]. Because $|C| < n$ if and only if $|\sim C| > N - n$, we can prevent such trivial compromises by suppressing any statistic whose query-set size falls outside the range $[n, N - n]$; this is the same query-set-size restriction introduced in Section 6.3.1.

We observed earlier that sensitive statistics can often be derived from nonsensitive ones by means of trackers and linear-system attacks. Friedman and Hoff-

FIGURE 6.13 Partitioning over two attributes.



man [Frie80] show how some of these attacks can be prevented by suppressing nonsensitive statistics whose "implied query sets" fall outside the range $[n, N - n]$.

Let $q(C)$ be a 2-order statistic of the form $C = a \bullet b$ or $C = a + b$, where a and b are values of attributes A and B respectively. The following relations hold (see Figure 6.13):

$$q(a \bullet \sim b) = q(a) - q(a \bullet b) \quad (1)$$
$$q(\sim a \bullet b) = q(b) - q(a \bullet b) \quad (2)$$
$$q(a \bullet b) = q(a) + q(b) - q(a + b) \quad (3)$$
$$q(\sim a \bullet \sim b) = q(All) - q(a + b) = q(All) - q(a) - q(b) + q(a \bullet b) \; . \quad (4)$$

Given $q(a \bullet b)$, Eqs. (1) and (2) can be used to derive $q(a \bullet \sim b)$ and $q(\sim a \bullet b)$, either of which could be sensitive even if $q(a \bullet b)$ is not. The formulas $(a \bullet \sim b)$ and $(\sim a \bullet b)$ are called the **implied query sets** of $(a \bullet b)$, and the statistic $q(a \bullet b)$ is restricted if either its query set or its implied query sets falls outside the range $[n, N - n]$. If $q(All)$ is permitted (even though $|All| > N - n$), the formula $q(\sim a \bullet \sim b)$ can be derived by Eq. (4), so that $(\sim a \bullet \sim b)$ is also implied by $(a \bullet b)$. Although $q(a + b)$ can also be derived from $q(a \bullet b)$ by Eq. (3), $q(a + b)$ cannot lead to disclosure if $q(\sim a \bullet \sim b)$ does not; therefore, it need not be explicitly checked.

Given $q(a + b)$, we can similarly derive $q(a \bullet b)$ by Eq. (3), and thereby also derive $q(a \bullet \sim b)$, $q(\sim a \bullet b)$, and $q(\sim a \bullet \sim b)$. Therefore, the statistic $q(a + b)$ is restricted if its query set or any of its other implied query sets $(a \bullet b)$,† $(a \bullet \sim b)$, $(\sim a \bullet b)$, and $(\sim a \bullet \sim b)$ fall outside the range $[n, N - n]$. Because $|\sim a \bullet \sim b| = N - |a + b|$, it is not necessary to explicitly check the size of both $(\sim a \bullet \sim b)$ and $(a + b)$.

To summarize: a query $q(a \bullet b)$ or $q(a + b)$ is permitted if and only if the sizes of the following query sets fall in the range $[n, N - n]$:

$$a \bullet b, \; a \bullet \sim b, \; \sim a \bullet b, \; \sim a \bullet \sim b \; .$$

† Friedman and Hoffman did not include $(a \bullet b)$ in their list of implied query sets for $(a + b)$; we have included it because $q(a \bullet b)$ can be sensitive even if $q(a + b)$ and its other implied queries are not.

By symmetry, the queries $q(\sim a \bullet b)$, $q(a \bullet \sim b)$, $q(\sim a \bullet \sim b)$, $q(\sim a + b)$, $q(a + \sim b)$, and $q(\sim a + \sim b)$ have the same four implied query sets. The four implied query sets partition the database as shown in Figure 6.13. The partitioning is such that given a statistic over any one of these areas (or over three of the four areas), then the same statistic can be computed over all the areas using only lower-order statistics [namely $q(a)$, $q(b)$, and $q(All)$].

Because the database is partitioned by the four query sets, it is not possible for one of the sets to be larger than $N - n$ unless some other set is smaller than n; therefore, it is not necessary to check upper bounds. It is necessary to check all four lower bounds, however, because any one of the four query sets could be sensitive even if the other three are not. We also observe that if the four 2-order query sets are not sensitive, then the 1-order statistics $q(a)$ and $q(b)$ cannot be sensitive. The converse, however, is not true.

Example:

We shall show how two attacks aimed at learning Baker's *GP* can be thwarted by checking implied query sets. Because Baker is the only *Female* student in *EE*, her *GP* could be derived using the following:

$$\textbf{sum}(Female \bullet EE, GP) = \textbf{sum}(Female, GP) + \textbf{sum}(EE, GP) - \textbf{sum}(Female + EE, GP) .$$

Because the query set $(Female + EE)$ has the implied query set $(Female \bullet EE)$ where $|Female \bullet EE| = 1$, the statistic $\textbf{sum}(Female + EE, GP)$ would be suppressed, thwarting the attack.

Similarly, Baker's *GP* could be derived from:

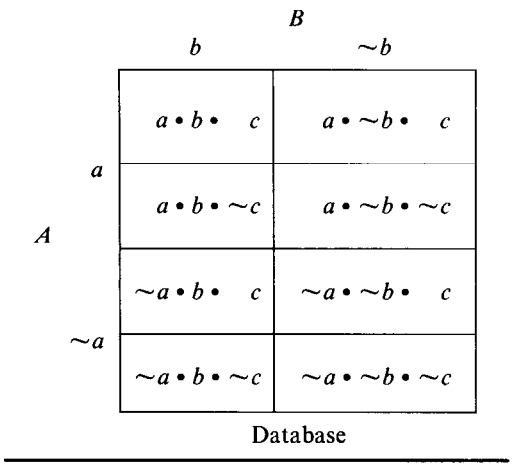
$$\textbf{sum}(Female \bullet EE, GP) = \textbf{sum}(Female, GP) - \textbf{sum}(Female \bullet \sim EE, GP) .$$

Because the query set $(Female \bullet \sim EE)$ has the implied query set $(Female \bullet \sim\sim EE) = (Female \bullet EE)$, the statistic $\textbf{sum}(Female \bullet \sim EE, GP)$ would similarly be suppressed, thwarting the attack. ■

The situation is considerably more complicated when the characteristic formula C of a query $q(C)$ is composed of more than two terms. We show in [Denn81] that given any m-order statistic q of the form $q(\alpha_1 \bullet \cdots \bullet \alpha_m)$ or $q(\alpha_1 + \cdots + \alpha_m)$, where $\alpha_i = a_i$ or $\sim a_i$ and a_i is the value of attribute A_i $(1 \leq i \leq m)$, the following 2^m statistics can be computed from q and lower-order statistics:

$$\begin{array}{l} q(\ a_1 \bullet\ a_2 \bullet \cdots \bullet\ a_m) \\ q(\ a_1 \bullet\ a_2 \bullet \cdots \bullet \sim a_m) \\ \qquad\vdots \\ q(\ a_1 \bullet \sim a_2 \bullet \cdots \bullet \sim a_m) \\ q(\sim a_1 \bullet\ a_2 \bullet \cdots \bullet\ a_m) \\ q(\sim a_1 \bullet\ a_2 \bullet \cdots \bullet \sim a_m) \end{array}$$

FIGURE 6.14 Partitioning over three attributes.



.
.
.

$q(\sim a_1 \bullet \sim a_2 \bullet \cdots \bullet \sim a_m)$.

We are thus led to the following control:

Implied-Queries Control:
An m-order statistic over attribute values $a_1, \ldots, a_m$ is permitted if and only if all 2^m implied query sets listed above have at least n records. ■

Figure 6.14 shows the eight implied query sets for the case $m = 3$. The formulas relating a statistic q computed over one of the query sets to the remaining query sets are:

$$
\begin{aligned}
q(\ a \bullet\ \ b \bullet \sim c) &= q(a \bullet\ \ b) - q(a \bullet b \bullet c) \\
q(\ a \bullet \sim b \bullet\ \ c) &= q(a \bullet\ \ c) - q(a \bullet b \bullet c) \\
q(\sim a \bullet\ \ b \bullet\ \ c) &= q(b \bullet\ \ c) - q(a \bullet b \bullet c) \\
q(\ a \bullet \sim b \bullet \sim c) &= q(a \bullet \sim b) - q(a \bullet \sim b \bullet c) \\
&= q(a) - q(a \bullet b) - q(a \bullet c) + q(a \bullet b \bullet c) \\
q(\sim a \bullet \sim b \bullet\ \ c) &= q(c) - q(a \bullet c) - q(b \bullet c) + q(a \bullet b \bullet c) \\
q(\sim a \bullet\ \ b \bullet \sim c) &= q(b) - q(a \bullet b) - q(b \bullet c) + q(a \bullet b \bullet c) \\
q(\sim a \bullet \sim b \bullet \sim c) &= q(All) - q(a) - q(b) - q(c) \\
&\quad + q(a \bullet b) + q(a \bullet c) + q(b \bullet c) - q(a \bullet b \bullet c) \ .
\end{aligned}
$$

Because of the exponential growth of the number of implied queries, we conclude that an implied-queries control may become impractical for high-order

TABLE 6.9 Table with sensitive statistic $q(a_1 \bullet b_1)$.

		B				
		b_1	b_2	...	b_t	
	a_1	—	$q(a_1 \bullet b_2)$	...	$q(a_1 \bullet b_t)$	$q(a_1)$
	a_2	$q(a_2 \bullet b_1)$	$q(a_2 \bullet b_2)$	...	$q(a_2 \bullet b_t)$	$q(a_2)$
A	.	.	.	.	.	.
	.	.	.	.	.	.
	.	.	.	.	.	.
	a_s	$q(a_s \bullet b_1)$	$q(a_s \bullet b_2)$	...	$q(a_s \bullet b_t)$	$q(a_s)$
		$q(b_1)$	$q(b_2)$	...	$q(b_t)$	$q(All)$

statistics. A 10-order statistic, for example, has 1024 implied queries.

Even if we examine all implied query sets, the control would not prevent deduction of sensitive statistics. To see why, suppose that attribute A has values a_1, ..., a_s, and that attribute B has values b_1, ..., b_t. Then the records of the database are partitioned into st groups, as shown in Table 6.9.

Suppose that the statistic $q(a_1 \bullet b_1)$ is sensitive, but that none of the remaining cells is sensitive. Then any attempt to deduce $q(a_1 \bullet b_1)$ from a statistic whose implied query sets include $(a_1 \bullet b_1)$ will be thwarted; for example, the following attack could not be used:

$$q(a_1 \bullet b_1) = q(b_1) - q(\sim a_1 \bullet b_1) .$$

Suppressing statistics that directly imply $q(a_1 \bullet b_1)$ does not, however, preclude deduction of $q(a_1 \bullet b_1)$ from queries about disjoint subsets of a_1 or b_1. For example,

$$q(a_1 \bullet b_1) = q(b_1) - [q(a_2 \bullet b_1) + \cdots + q(a_s \bullet b_1)].$$

This example suggests that for a given m-order statistic over attributes A_1, ..., A_m, it would be necessary to check all elementary sets defined by these attributes. If each attribute A_i has $|A_i|$ possible values, this would involve checking

$$\prod_{i=1}^{m} |A_i|$$

query sets.

It is more efficient to keep a history of previously asked queries for the purpose of determining whether each new query causes compromise (see Section 6.3.3) than it is to determine whether a query could potentially cause compromise. Moreover, the implied-queries approach is likely to be much less precise, because the additional queries needed to cause compromise may never be asked.

6.4.3 Partitioning

Yu and Chin [Yu77] and Chin and Ozsoyoglu [Chin79] have studied the feasibility of **partitioning** a dynamic database at the physical level into disjoint groups such that:

TABLE 6.10 Partitioned database.

Sex	Class			
	1978	1979	1980	1981
Female	4	2	2	0
Male		2	0	2

1. Each group G has $g = |G|$ records, where $g = 0$ or $g \geq n$, and g is even.
2. Records are added to or deleted from G in pairs.
3. Query sets must include entire groups. If the query set for a statistic includes one or more records from each of m groups $G_1, \ldots, G_m$, then $q(G_1 + \cdots + G_m)$ is released.

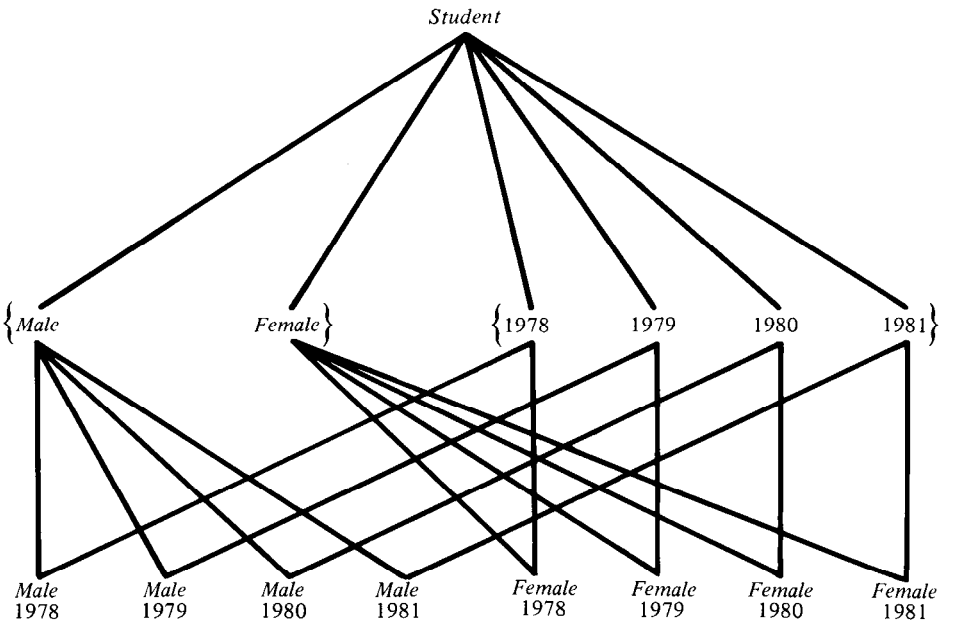
The first two conditions prevent attacks based on small query sets and insertion or deletions of records (see Section 6.3.5). The third condition prevents exact disclosure from attacks based on isolating a particular individual—for example, by using a tracker or a linear system of equations. Clever query sequences can at best disclose information about an entire group.

Example:
Table 6.10 gives counts for a possible partitioning of the student record database when $k = 1$. Because the database has an odd number of records, the record for Kline has been omitted. Because the query set "*Female* • 1978" contains a single record and the set "*Male* • 1978" contains an odd number of records (3) these sets have been merged (Olsson calls this "rolling up" [Olss75]). A query for a statistic **count**(*Male* • 1978) would thus return **count**(1978) = 4 (rather than 3), and a query **count**(*EE*) would return **count**(1978 + *Female* • 1980 + *Male* • 1981) = 8 (rather than 4) because there are *EE* majors in all three groups. ■

Partitioning by 1-, 2-, or 3-order statistics is equivalent to releasing tables of macrostatistics as described in the previous section. Therefore, to control approximate disclosures (as by the n-respondent, k%-dominance rule) cell-suppression techniques must be applied; this may not be practical for dynamic databases. Using broad categories defined by 2- or 3-order statistics may also limit the usefulness of the database. Yet if we partition by higher-order statistics, cell suppression may be too costly.

Chin and Ozsoyoglu [Chin81] also consider the design of a complete database system that supports partitioning at the logical level. They describe their approach in terms of the Data Abstraction Model of Smith and Smith [Smit77]. The model partitions the individuals represented in the database into populations having common characteristics; populations can be decomposed into subpopulations, and populations that cannot be further decomposed are "atomic." The complete set of populations forms a hierarchy such that each nonatomic population is composed of disjoint atomic populations. Disjoint populations having a common parent may be grouped into "clusters".

FIGURE 6.15 Data abstraction model of student record database.



Example:
Figure 6.15 illustrates a partitioning of the student record database by *Sex* and *Class* similar to the partitioning in Table 6.10. The populations at the leaves of the hierarchy are atomic, and the populations at the middle level form two clusters: *Sex* = {*Male*, *Female*} and *Class* = {1978, 1979, 1980, 1981}. ■

A **Population Definition Construct** (PDC) defines each population, the operations that can be performed over the population, and the security constraints of the population. The permitted statistics $q(P)$ for a population P must satisfy the constraints.

1. $q(P)$ is permitted if and only if $q(P')$ is permitted for every population P' in a cluster with P.
2. $q(P)$ is permitted if $q(S)$ is permitted for any subpopulation S of P.

If $P_1, \ldots, P_m$ are atomic populations in the same cluster, condition (1) says that if any P_i must be suppressed, then all P_i must be suppressed. This may be much more restrictive than necessary.

A **User Knowledge Construct** (UKC) defines groups of users and their supplementary knowledge about the database, the operations permitted to members of the group, and the security constraints of the group. Security information in both

the PDCs and a UKC is used to determine whether a statistic should be released to a particular user.

Feige and Watts [Feig70] describe a variant of partitioning called **microaggregation:** individuals are grouped to create many synthetic "average individuals"; statistics are computed for these synthetic individuals rather than the real ones.

Partitioning may limit the free flow of statistical information if groups are excessively large or ill-conceived, or if only a limited set of statistics can be computed for each group. But if a rich set of statistical functions is available, large groups may not severely impact the practicality of some databases.

Dalenius and Denning [Dale79] considered the possibility of a single-partition database; that is, the only available statistics are those computed over the entire database. All released statistics would be finite moments of the form:

$$q(All, e_1, \ldots, e_M) = \sum_{i=1}^{N} x_{i1}^{e_1} x_{i2}^{e_2} \ldots x_{iM}^{e_M},$$

where

$$e_1 + e_2 + \cdots + e_M \leq e$$

for some given e. Because all statistics are computed over the entire database, disclosure is extremely difficult if not impossible. At the same time, it is possible for the statistician to compute correlations of attributes.

We considered the feasibility of releasing all moments for a given e as an alternative to releasing macrostatistics or microstatistics. This approach would provide a richer set of statistics than are provided by macrostatistics, but a higher level of protection than provided by microstatistics. The total number of moments increases rapidly with e and M, however, so it may not be feasible to compute and release more than a relatively small subset of all possible moments. The total number of moments over M variables is given by

$$m_e(M) = \binom{M+e}{e}.$$

Example:
If $M = 40$, then $m_2(40) = 861$ and $m_3(40) = 12{,}341$. ■

6.5 MECHANISMS THAT ADD NOISE

Restricting statistics that might lead to disclosure can be costly and imprecise, especially if we take into account users' supplementary knowledge. Consequently, there is considerable interest in simple mechanisms that control disclosure by adding noise to the statistics. These mechanisms are generally more efficient to apply, and allow the release of more nonsensitive statistics.

6.5.1 Response Perturbation (Rounding)

Response perturbation refers to any scheme which perturbs a statistic $q = q(C)$ by some function $r(q)$ before it is released. The perturbation usually involves some form of rounding—that is, q is rounded up or down to the nearest multiple of some base b.

There are two kinds of rounding: systematic rounding and random rounding. **Systematic rounding** always rounds q either up or down according to the following rule. Let $b' = \lfloor (b + 1)/2 \rfloor$ and $d = q \bmod b$. Then

$$r(q) = \begin{cases} q & \text{if } d = 0 \\ q - d & \text{if } d < b' \quad \text{(round down)} \\ q + (b - d) & \text{if } d \geq b' \quad \text{(round up)} \end{cases} .$$

Given a rounded value $r(q)$, a user can deduce that q lies in the interval $[r(q) - b' + 1, r(q) + b' - 1]$. For example, if $b = 5$, then $r(q) = 25$ implies $q \in [23, 27]$.

Under certain conditions, it is possible to recover exact statistics from their rounded values. Let $C_1, \ldots, C_m$ be disjoint query sets, and let $C_{m+1} = C_1 \cup \ldots \cup C_m$; thus, $q_{m+1} = q_1 + \ldots + q_m$, where $q_i = \mathbf{sum}(C_i, A)$ for some attribute A $(1 \leq i \leq m + 1)$. Let $[L_i, U_i]$ be the interval estimates for each $r(q_i)$, and let $L = L_1 + \ldots + L_m$ and $U = U_1 + \ldots + U_m$. Achugbue and Chin [Achu79] show that it is possible to deduce the exact values of the q_i from the rounded values $r(q_i)$ when either:

(i) $U = L_{m+1}$, in which case

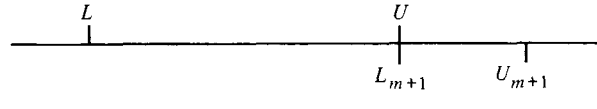
$$q_i = U_i \quad (1 \leq i \leq m)$$
$$q_{m+1} = L_{m+1}, \text{ or}$$

(ii) $L = U_{m+1}$, in which case

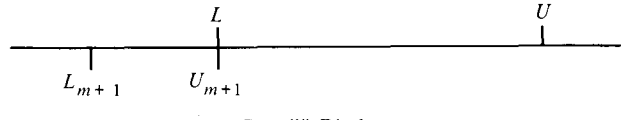
$$q_i = L_i \quad (1 \leq i \leq m)$$
$$q_{m+1} = U_{m+1} .$$

(See Figure 6.16.)

FIGURE 6.16 Disclosure from rounded values.



Case (i) Disclosure



Case (ii) Disclosure

TABLE 6.11 Disclosure under systematic rounding.

	$r(q_i)$	L_i	U_i
q_1	15	13	17
q_2	10	8	12
q_3	15	13	17
q_4	20	18	22
		52	68
q_{m+1}	70	68	72

Case (i) is illustrated in Table 6.11. Because $U = 68 = L_{m+1}$, each q_i $(1 \leq i \leq m)$ must achieve its maximum; otherwise q_{m+1} would be smaller, making $r(q_{m+1})$ and L_{m+1} smaller.

If neither Case (i) nor Case (ii) applies, then exact disclosure is not possible from $r(q_1), \ldots, r(q_m)$. Nevertheless, if the overlap between $[L, U]$ and $[L_{m+1}, U_{m+1}]$, is not too large, it may be possible to reduce the interval estimates $[L_i, U_i]$. If other (rounded) statistics are available, the intervals may be further reduced. (Schlörer [Schl77] also investigated the vulnerabilities of systematic rounding to tracker attacks.)

Random rounding rounds a statistic q up or down according to the following rule:

$$r(q) = \begin{cases} q & \text{if } d = 0 \\ q - d & \text{with probability } 1 - p \quad \text{(round down)} \\ q + (b - d) & \text{with probability } \quad p \quad \text{(round up)} \end{cases}$$

When $p = d/b$, random rounding has the advantage of being unbiased.

Random rounding is subject to the same methods of error removal as systematic rounding.

Example:

Table 6.12 shows how exact values can be recovered from a 2-dimensional table of rounded sums, where the rounding base b is 5. Here the total must be at least 76, which is achievable only when the entries inside the table achieve their maximum possible values. ■

Random rounding is also vulnerable to another kind of attack in query-processing systems. If a query q is asked many times, its true value can be deduced by averaging the rounded values. (See also [Fell74,Narg72,Haq77,Palm74].)

Both systematic and random rounding have the drawback that the sum of the rounded statistics for disjoint query sets can differ from the rounded statistic for the union of the sets. As illustrated by Tables 6.11 and 6.12, this can often be exploited to obtain better estimates of the rounded values. **Controlled rounding** overcomes this deficiency by requiring the sum of rounded statistics to equal their rounded sum [Caus79,Cox81]; that is, if $C_1, \ldots, C_m$ are disjoint query sets and C_{m+1} is the union $C_{m+1} = C_1 \cup \ldots \cup C_m$, then

TABLE 6.12 Disclosure under random rounding (adapted from [Schl77]).

10	20	35
10	20	40
25	50	80

(a) Table of Rounded Values.

6–14	16–24	31–39
6–14	16–24	36–44
21–29	46–54	76–84

(b) Table of Interval Estimates.

14	24	38
14	24	38
28	48	76

(c) Table of Exact Values.

$$r(q_1) + \cdots + r(q_m) = r(q_{m+1}) \ .$$

Cox [Cox81] describes a method of achieving controlled rounding in 1- or 2-dimensional tables of macrostatistics. For a given integer $p \geq 1$, the method finds an **optimal controlled rounding** that minimizes the sum of the pth powers of the absolute values of the differences between the true statistics and their rounded values; that is, that minimizes the objective function

$$z_p = \sum_q | q - r(q) |^p$$

The problem of finding an optimal controlled rounding can be expressed as a capacity-constrained transportation problem [Dant63], and thereby solved using standard algorithms. The technique is particularly well-suited for protecting tables of relatively small frequency counts.

Other kinds of response perturbations are possible. For example, Schwartz [Schw77] studies functions $r(q)$ that return a pseudo-random value uniformly distributed over the interval $[q - d, q + d]$ for relatively small values of d. To prevent error removal by averaging, the same query must always return the same response.

6.5.2 Random-Sample Queries

We mentioned earlier that census agencies often protect microstatistics by releasing relatively small samples of the total number of records. In this section we shall describe the results of an ongoing research project aimed at applying sampling to query-processing systems.

Most inference controls are subverted by a single basic principle of compromise: because the user can control the composition of each query set, he can isolate a single record or value by intersecting query sets. Denning [Denn80c] introduced a new class of queries called **Random-Sample Queries** (RSQs) that deny the intruder precise control over the queried records. RSQs introduce enough uncertainty that users cannot isolate a confidential record but can get accurate statistics for groups of records.

This query-based control differs from the sampling controls employed in population surveys in two respects. First, it uses relatively large samples, say on the order of 80–90% of the total number of records in the database; thus, the released statistics are fairly accurate. Second, it uses a different sample to compute each statistic. It is this aspect of the strategy that radically departs from the traditional use of sampling and allows the use of large samples. But whereas this is economical to implement in query-processing systems, it would be expensive to use with microstatistics or macrostatistics. The control is defined as follows:

Random-Sample-Query Control:
Given a query $q(C)$, as the query processor examines each record i in C, it applies a selection function $f(C, i)$ that determines whether i is used to compute the statistic. The set of selected records forms a sampled query set

$$C^* = \{i \in C \mid f(C, i) = 1\} ,$$

from which the query processor returns $q^* = q(C^*)$. A parameter p specifies the sampling probability that a record is selected. ■

The uncertainty introduced by this control is the same as the uncertainty in sampling the entire database, with a probability p of selecting a particular record for a sample. The expected size of a random sample over the entire database of size N is pN.

The control is easy to implement when $p = 1 - 1/2^k$. Let $r(i)$ be a function that maps the ith record into a random sequence of $m \geq k$ bits. Let $s(C)$ be a function that maps formula C into a random sequence of length m over the alphabet {0, 1, *}; this string includes exactly k bits and $m - k$ asterisks (asterisks denote "don't care"). The ith record is excluded from the sampled query set whenever $r(i)$ matches $s(C)$ [a "match" exists whenever each nonasterisk character of $s(C)$ is the same as the corresponding symbol of $r(i)$]. The selection function $f(C, i)$ is thus given by

$$f(C, i) = \begin{cases} 1 & \text{if } r(i) \text{ does not match } s(C) \\ 0 & \text{if } r(i) \text{ matches } s(C) \end{cases} .$$

This method applies for $p > 1/2$ (e.g., $p = .5, .75, .875$, and $.9375$). For $p < 1/2$, use $p = 1/2^k$; the ith record is included in the sample if and only if $r(i)$ matches $s(C)$.

Example:
Let $p = 7/8$, $m = 8$, and $s(C) =$ "*10*1***". If $r(i) =$ "11011000" for some

i, that record would match $s(C)$ and be excluded from C^*. If r generates unique random bit sequences, then the expected size of C^* is 7/8 that of C. ■

Encryption algorithms are excellent candidates for the functions r and s, as they yield seemingly random bit sequences. If the database is encrypted anyway, the function r could simply select m bits from some invariant part of the record (e.g., the identifier field); this would avoid the computation of $r(i)$ during query processing. With a good encryption algorithm, two formulas C and D having almost identical query sets will map to quite different $s(C)$ and $s(D)$, thereby ensuring that C^* and D^* differ by as much as they would with pure random sampling.

Under RSQs, it is more natural to return relative frequencies and averages directly, since the statistics are not based on the entire database, and the users may not know what percentage of the records are included in the random sample. Recall that the true relative frequencies and averages are given by:

$$\mathbf{rfreq}(C) = \frac{|C|}{N}$$

$$\mathbf{avg}(C, A) = \frac{1}{|C|} \sum_{i \epsilon C} x_i \ .$$

The sampled relative frequencies and averages are:

$$\mathbf{rfreq}^*(C) = \frac{|C^*|}{pN}$$

$$\mathbf{avg}^*(C, A) = \frac{1}{|C^*|} \sum_{i \epsilon C^*} x_i \ .$$

Note the expected value of $|C^*|$ is $p|C|$; thus, the expected value of the sampled frequency is $|C|/N$, the true relative frequency.

The values of p and N may be published so users can judge the significance of the estimates returned. A user who knows p and N can then compute approximations for both the sampled and unsampled counts and sums.

A minimum query-set-size restriction is still needed with RSQs if the sampling probability is large. Otherwise, all records in a small query set would be included in a sample with high probability, making compromise possible.

Compromise is controlled by introducing small sampling errors into the statistics. For frequencies, the relative error between the sampled frequency and the true frequency is given by

$$f_C = \frac{\mathbf{rfreq}^*(C) - \mathbf{rfreq}(C)}{\mathbf{rfreq}(C)} \ .$$

The expected relative error is zero; thus, the sampled relative frequency is an unbiased estimator of the true relative frequency. The root-mean-squared relative error is

$$\hat{R}(f_C) = \sqrt{\frac{1-p}{|C|p}} \ .$$

Thus, for fixed p, the expected error decreases as the square root of the query-set size.

Figure 6.17 shows a graph of the error $\hat{R}(f_C)$ as a function of $|C|$ for several values of p. For $p > .5, |C| > 100$ gives less than a 10% error. For $p = .9375, |C| > 667$ gives less than a 1% error. For extremely small query sets, however, the relative errors may be unacceptably high. Absolute errors for counts are greater than those for relative frequencies by a factor of N; however, their relative errors are comparable. The same holds for sums as compared with averages.

The relative error between a sampled average and the true average is given by:

$$a_C = \frac{\mathbf{avg}^*(C, A) - \mathbf{avg}(C, A)}{\mathbf{avg}(C, A)} \ .$$

The sampled average is not unbiased, but its bias is negligible. The root-mean-square error depends on the distribution of the data values in the query set. For sufficiently large $|C|$, it is approximately

$$\hat{R}(a_C) \simeq \mathbf{cfv}(C, A)\hat{R}(f_C) \ ,$$

where $\mathbf{cfv}(C, A) = \mathbf{var}(C, A)^{1/2}/\mathbf{avg}(C, A)$ is the coefficient of variation for the distribution of data values of A in C. If the data values are uniformly distributed over a moderately large interval $[1, d]$ (e.g., $d > 10$), the root-mean-square error becomes

$$\hat{R}(a_C) \simeq 0.6\ \hat{R}(f_C) \ ,$$

showing that the relative errors in averages behave the same as in frequencies but are 40% smaller. These results were confirmed experimentally on a simulated database.

RSQs control compromise by reducing a questioner's ability to interrogate the desired query sets precisely. We have studied the extent to which the control may be circumvented by small query sets, general trackers, and error removal by averaging. Compromise may be possible with small query sets unless p is small or a minimum query-set-size restriction is imposed. Trackers, on the other hand, are no longer a useful tool for compromise.

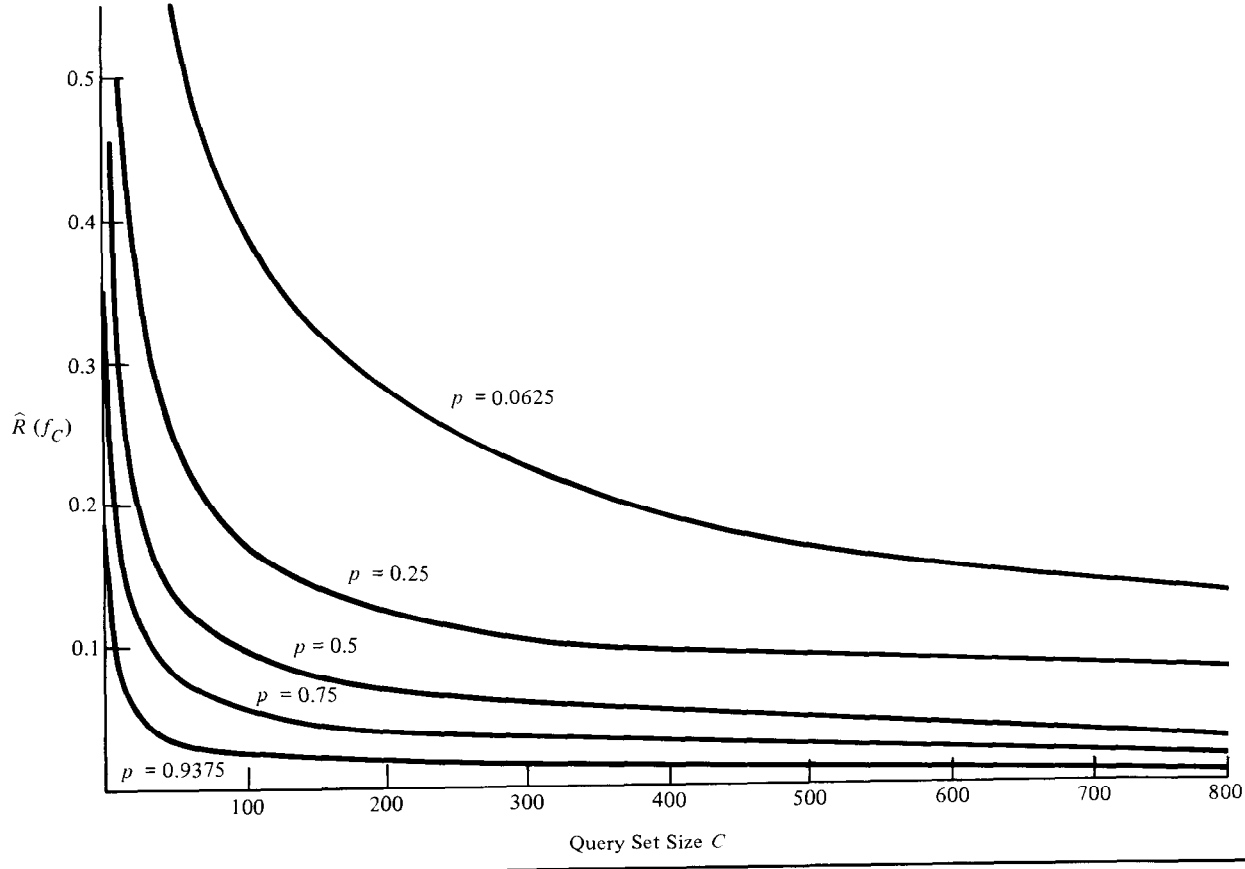
RSQs appear to be most vulnerable to attacks based on error removal by averaging. Because the same query always returns the same response, it is necessary to pose different but "equivalent" queries to remove sampling errors. One method involves averaging the responses of equivalent queries that use different formulas to specify the same query set.

Example:

The statistic $q(Male \bullet 1978)$ could be estimated from the sampled statistics:

$$q^*(Male \bullet 1978)$$

FIGURE 6.17 Expected root mean squared relative error in frequency.



$q^*(\sim Female \bullet 1978)$
$q^*(Male \bullet \sim(1979 + 1980 + 1981))$
$q^*(Male \bullet (Bio \bullet 1978) + Male \bullet (\sim Bio \bullet 1978))$
.
.
. ■

Schlörer observed that this problem does not arise if $s(C)$ is a function of the query set C rather than the characteristic formula so that $s(C) = s(D)$ whenever formulas C and D are reducible to each other. Still, this would not prevent a second method of averaging that uses disjoint subsets of query sets.

Example:
The statistic $q(Male \bullet 1978)$ could be estimated from:

$q^*(Male \bullet 1978 \bullet Bio) + q^*(Male \bullet 1978 \bullet \sim Bio)$
$q^*(Male \bullet 1978 \bullet CS) + q^*(Male \bullet 1978 \bullet \sim CS)$
$q^*(Male \bullet 1978 \bullet EE) + q^*(Male \bullet 1978 \bullet \sim EE)$
$q^*(Male \bullet 1978 \bullet Psy) + q^*(Male \bullet 1978 \bullet \sim Psy)$
.
.
. ■

Let $q_1^*, \ldots, q_m^*$ be the responses from m independent queries that estimate q = **rfreq**(C), and let

$$\hat{q} = \frac{1}{m} \sum_{i=1}^{m} q_i^*$$

be an estimate of q. The mean and variance of $\hat{q}$ are:

$$E(\hat{q}) = \frac{|C|}{N}$$

$$\text{Var}(\hat{q}) = \frac{|C|(1-p)}{m N^2 p} .$$

For large m ($m \geq 30$ should be sufficient when the distribution of possible responses for each q_i^* is symmetric), the distribution of $\hat{q}$ is approximately normal. Letting $\sigma_{\hat{q}} = [\text{Var}(\hat{q})]^{1/2}$, the confidence intervals for the true frequency q given $\hat{q}$ are:

$$Pr[q \in [\hat{q} \pm 1.645\sigma_{\hat{q}}]] \simeq .90$$
$$Pr[q \in [\hat{q} \pm 1.960\sigma_{\hat{q}}]] \simeq .95$$
$$Pr[q \in [\hat{q} \pm 2.575\sigma_{\hat{q}}]] \simeq .99 .$$

If we assume that a 95% confidence interval is required for disclosure, the length of this interval is given by

$$k = 3.92\sigma_{\hat{q}} = \frac{3.92}{N}\sqrt{\frac{(1-p)\,|C|}{p\,m}} \quad .$$

Now, $k \leq 1/N$ is required to estimate q to within one record (such accuracy is required, for example, to estimate relative frequencies for small query sets using trackers). The number of queries required to achieve this accuracy is

$$m \geq (3.92)^2 \left(\frac{1-p}{p}\right) |C| > 15\left(\frac{1-p}{p}\right) |C| \quad .$$

For fixed p, the function grows linearly in $|C|$. For $p = .5$, over 450 queries are required to estimate frequencies for query sets of size 30; over 1500 queries are required to estimate frequencies for query sets of size 100. For $p = .9375$, 100 queries are required to estimate frequencies for query sets of size 100.

For averages taken over variables uniformly distributed over a range $[1, s]$,

$$m > 128\left(\frac{1-p}{p}\right) |C|$$

queries are required to obtain an estimate sufficiently accurate to enable personal disclosure by a simple tracker attack (more complex linear-system attacks would require even more queries). This is about an order of magnitude greater than for frequencies; whereas the relative errors in averages (for uniform distributions) are lower than in frequencies, more queries are required to obtain estimates accurate enough to compromise with averages than with frequencies. S. Kurzban observed, however, that compromise may be easier for skewed distributions, which would also have higher relative errors.

For large query sets, the number of queries required to obtain reliable estimates of confidential data under RSQs is large enough to protect against manual attacks. Nevertheless, a computer might be able to subvert the control by systematically generating the necessary queries. Threat monitoring (i.e., keeping a log or audit trail) is probably necessary to detect this type of systematic attack [Hoff70].

6.5.3 Data Perturbation

Noise can also be added to the data values directly—either by permanently modifying the data stored in the database, or by temporarily perturbing the data when it is used in the calculation of some statistic. The first approach is useful for protecting data published in the form of microstatistics, but it cannot be used in general-purpose databases where the accuracy of the data is essential for nonstatistical purposes. This section describes a temporary perturbation scheme for general-purpose systems. The next section describes data modification for published microstatistics.

Data perturbation involves perturbing each data value x_i used to compute a statistic $q(C)$ by some function $f(x_i)$, and then using $x_i' = f(x_i)$ in place of x_i in the computation. Beck [Beck80] showed how data perturbation could be integrated

into a query-processing system for **count**, **sum**, and selection queries. We shall describe his approach for protecting data released as sums.

Consider the query

$$S = \mathbf{sum}(C, A) = \sum_{i \epsilon C} x_i \ .$$

Rather than releasing $\mathbf{sum}(C, A)$, the system computes and releases

$$S' = \mathbf{sum}'(C, A) = \sum_{i \epsilon C} x_i' \ , \tag{6.12}$$

where

$$x_i' = x_i + zl_i(x_i - \bar{x}_C) + z2_i \ ,$$

and $\bar{x}_C = \mathbf{avg}(C, A) = \mathbf{sum}(C, A)/|C|$ is the mean value taken over the query set C, and zl_i and $z2_i$ are independent random variables, generated for each query, with expected value and variance:

$$E(zl_i) = 0 \quad \mathrm{Var}(zl_i) = 2a^2$$
$$E(z2_i) = 0 \quad \mathrm{Var}(z2_i) = \frac{2a^2}{|C|}(\bar{x}_C - \bar{x})^2 \ ,$$

where $\bar{x} = \mathbf{avg}(All, A)$ is the mean taken over the entire database, and the parameter a is constant for all queries. The expected value of S' is $E(S') = S$; thus, the perturbed statistic is an unbiased estimator of the true statistic.

The variance of S' is

$$\mathrm{Var}(S') = 2a^2\sigma_C^2|C| + 2a^2(\bar{x}_C - \bar{x})^2 \ , \tag{6.13}$$

where

$$\sigma_C^2 = \frac{1}{|C|} \sum_{i \epsilon C} (x_i - \bar{x}_C)^2$$

is the sample variance over C. It is bounded below by

$$\mathrm{Var}(S') \geq a^2(x_i - \bar{x})^2 \tag{6.14}$$

for each x_i, which implies

$$\sigma_{S'} > a|x_i - \bar{x}| \ ,$$

where $\sigma_{S'}$ is the standard deviation of the estimate S'.

Beck defines a value x_i to be safe if it is not possible to obtain an estimate $\hat{x}_i$ of x_i such that

$$\sigma_{\hat{x}_i} < c|x_i - \bar{x}| \ ,$$

where c is a parameter of the system. The preceding result shows that it is not possible to compromise a value x_i using a single query if $a \geq c$.

Beck also shows that it takes a least $n = (a/c)^2$ queries to compromise a

database using any kind of linear-system attack, including those based on error removal by averaging. Combining these two results, we see that compromise can be prevented by picking $a \gg c$.

Unfortunately, this has the undesirable side effect of introducing extremely large errors into the released statistics. Beck solves this problem by introducing a scheme for changing the $z1_i$ and $z2_i$ so that it is possible to achieve an exponential increase in the number of queries needed to compromise, with less than a linear increase in the standard deviation of the responses.

Rather than picking a completely new $z1_i$ and $z2_i$ for each record and each query, $z1_i$ and $z2_i$ are computed from the sum of m independent random variables:

$$z1_i = \sum_{j=1}^{m} z1_{ij} \ , \quad z2_i = \sum_{j=1}^{m} z2_{ij} \ ,$$

where:

$$E(z1_{ij}) = 0 \quad \mathrm{Var}(z1_{ij}) = 2a^2$$

$$E(z2_{ij}) = 0 \quad \mathrm{Var}(z2_{ij}) = \frac{2a^2}{|C|}(\bar{x}_C - \bar{x})^2$$

$(1 \leq j \leq m)$. Then

$$E(z1_i) = 0 \quad \mathrm{Var}(z1_i) = 2ma^2$$

$$E(z2_i) = 0 \quad \mathrm{Var}(z2_i) = \frac{2ma^2}{|C|}(\bar{x}_C - \bar{x})^2 \ .$$

This means that the response S' will have variance

$$\mathrm{Var}(S') = 2ma^2\sigma_C^2|C| + 2ma^2(\bar{x}_C - \bar{x})^2 \ ,$$

so that the standard deviation will be bounded by

$$\sigma_{S'} > am^{1/2}|x_i - \bar{x}| \ .$$

Decomposing the $z1_i$ and $z2_i$ into m components therefore increases the standard deviation in the error by $m^{1/2}$.

Each $z1_i$ and $z2_i$ is changed for each query by changing at least one of the $z1_{ij}$ and $z2_{ij}$ $(1 \leq j \leq m)$. Beck shows that by changing only $z1_{i1}$ and $z2_{i1}$ after each query, compromise may be possible with $n = (a/c)^2$ queries as before. If in addition we change $z1_{i2}$ and $z2_{i2}$ after every $d = \lfloor (a/c)^2 \rfloor$ queries, then $n = d^2$ queries are required to compromise. If we continue in this way to change $z1_{ij}$ and $z2_{ij}$ after every d^{j-1} queries $(1 \leq j \leq m)$, then $n = d^m = \lfloor (a/c)^2 \rfloor^m$ queries are needed to compromise, whereas the standard deviation in the errors is proportional to only $am^{1/2}$. Beck shows that picking $a = 3^{1/2}c$ (i.e., $d = 3$) minimizes $am^{1/2}$ while holding $\lfloor (a/c)^2 \rfloor^m$ constant.

As m increases, the standard deviation of responses grows as $m^{1/2}$, while the difficulty of compromisng grows as 3^m. Using a simulated database, Beck confirmed his hypothesis that it was possible to protect against billions of queries and still provide reasonably accurate statistics.

Beck outlines a simple implementation that allows the released value S' to be

computed with just a single pass over the data values. Let $r(i)$ be a function that maps the ith record into a pseudorandom bit pattern, as in the random-sample-queries control; similarly, let $r(S)$ be a pseudorandom bit pattern of the query S. We also associate $m - 1$ bit patterns $b_2, \ldots, b_m$ with the database, where each b_j is changed after every d^{j-1} queries. Then $z1_{il}$ is generated using $r(i) \oplus r(S)$ as a seed to a random number generator (where $\oplus$ denotes exclusive-or); and for $j = 2, \ldots, m$, $z1_{ij}$ is generated using $r(i) \oplus b_j$ as the seed. Similarly, $z2^*_{ij}$ is generated, using a different random number generator, where $z2^*_{ij}$ is the same as $z2_{ij}$, except that $\text{Var}(z2^*_{ij}) = 1$ (thus it can be generated without knowing $\bar{x}_C$); therefore,

$$z2_{ij} = \left(\frac{2a^2(\bar{x}_C - \bar{x})^2}{|C|}\right)^{1/2} z2^*_{ij} \ .$$

To compute the released statistic in a single pass over the query set, we observe that

$$S' = \sum_{i \epsilon C} x'_i = \sum_{i \epsilon C} [x_i + z1_i(x_i - \bar{x}_C) + z2_i]$$

$$= S + S1 - \bar{x}_C Z1 + \left(\frac{2a^2(\bar{x}_C - \bar{x})^2}{|C|}\right)^{1/2} Z2^* \ ,$$

where:

$$S1 = \sum_{i \epsilon C} z1_i x_i$$

$$Z1 = \sum_{i \epsilon C} z1_i$$

$$Z2^* = \sum_{i \epsilon C} z2^*_i \ .$$

6.5.4 Data Swapping

Schlörer [Schl77] suggested a data transformation scheme based on interchanging values in the records. The objective is to interchange (swap) enough values that nothing can be deduced from disclosure of individual records, but at the same time to preserve the accuracy of at least low-order statistics. The approach has subsequently been studied by Schlörer [Schl81] and by Dalenius and Reiss [Dale78], who introduced the term "data swapping".

Schlörer defines a database D to be **d-transformable** if there exists at least one other database D' such that

1. D and D' have the same k-order frequency counts for $k = 0, 1, \ldots, d$, and
2. D and D' have no records in common.

Example:

Table 6.13 shows a 2-transformable database D of student records containing the three fields *Sex*, *Major*, and *GP*. This database is 2-transformable

TABLE 6.13 A 2-transformable database.

Record	D			D'		
	Sex	Major	GP	Sex	Major	GP
1	*Female*	*Bio*	4.0	*Male*	*Bio*	4.0
2	*Female*	*CS*	3.0	*Male*	*CS*	3.0
3	*Female*	*EE*	3.0	*Male*	*EE*	3.0
4	*Female*	*Psy*	4.0	*Male*	*Psy*	4.0
5	*Male*	*Bio*	3.0	*Female*	*Bio*	3.0
6	*Male*	*CS*	4.0	*Female*	*CS*	4.0
7	*Male*	*EE*	4.0	*Female*	*EE*	4.0
8	*Male*	*Psy*	3.0	*Female*	*Psy*	3.0

because the database D' has the same 0-, 1-, and 2-order statistics. For example, **count**(*Female* • *CS*) = 1 in both D and D'. Note, however, that 3-order statistics are not preserved. For example,

$$\textbf{count}(Female \bullet CS \bullet 3.0) = \begin{cases} 1 & \text{in } D \\ 0 & \text{in } D' \end{cases}. \quad \blacksquare$$

Because all 1-order counts must be preserved, D' must contain exactly the same set of values, and in the same quantities, as D. Thus, D' can be obtained from D by swapping the values among the records. If swapping is done on a single attribute A (as in Table 6.13), it suffices to check counts that involve the values of A to determine whether all low-order statistics are preserved.

Schlörer studied the conditions under which a database D is d-transformable; he proved that

1. D must have $M \geq d + 1$ attributes.
2. D must contain at least $N \geq (m/2)2^d$ records, where m is the maximum number of values $|A_j|$ for any attribute A_j $(1 \leq j \leq M)$.

He also showed that D must have a recursive structure. Consider each subdatabase D_1 of D consisting of all records having the same value for some attribute A (in D_1 the attribute A is omitted). If D is d-transformable, then D_1 must be $(d - 1)$-transformable over the remaining attributes.

Example:
Figure 6.18 illustrates the recursive structure of the database D of Table 6.13, where A = *Sex*. Note that the subdatabase D_1' is a 1-transformation of the subdatabase D_1, and vice-versa. ■

Data swapping could be used in two ways. One way would be to take a given database D, find a d-transformation on D for some suitable choice of d, and then release the transformed database D'. This method could be used with statistics-only databases and the publication of microstatistics; it could not be used with

FIGURE 6.18. Recursive structure of database.

D

	D_1	
Female	*Bio*	4.0
Female	*CS*	3.0
Female	*EE*	3.0
Female	*Psy*	4.0
Male	*Bio*	3.0
Male	*CS*	4.0
Male	*EE*	4.0
Male	*Psy*	3.0

D_1'

general-purpose databases, where accuracy of the data is needed for nonstatistical purposes. Because k-order statistics for $k > d$ are not necessarily preserved, these statistics would not be computed from the released data.

If swapping is done over all confidential variables, the released data is protected from disclosure. Reiss [Reis79] has shown, however, that the problem of finding a general data swap is **NP**-complete. Thus, the method appears to be impractical.

To overcome these limitations, Reiss [Reis80] has studied the possibility of applying **approximate data swapping** to the release of microstatistics. Here a portion of the original database is replaced with a randomly generated database having approximately the same k-order statistics ($k = 0, \ldots, d$) as the original database. The released database is generated one record at a time, where the values chosen for each record are randomly drawn from a distribution defined by the k-order statistics of the original data. Reiss shows that it is possible to provide fairly accurate statistics while ensuring confidentiality.

Any scheme that modifies the data records cannot be used in general-purpose systems. There is, however, another way of applying data swapping that would be applicable to these systems. If we could simply show that a database D is d-transformable, we could safely release any k-order statistic ($k \leq d$), because such a statistic could have been derived from a different set of records. For example, there is no way of determining (without supplementary knowledge) a student's *GP* from the 0-, 1-, and 2-order statistics of the database shown in Table 6.13, even though each student is uniquely identifiable by *Sex* and *Major*. The problem with this approach is that there is no known efficient algorithm for testing a database for d-transformability. Even if an efficient algorithm could be found, security cannot be guaranteed if the users have supplementary knowledge about the database (see exercises at end of chapter).

Data released in the form of microstatistics may be perturbed in other ways—for example, by rounding or by swapping a random subset of values with-

out regard to preserving more than 0 and 1-order statistics. Some of these techniques are discussed by Dalenius [Dale76] and by Campbell, Boruch, Schwartz, and Steinberg [Camp77].

6.5.5 Randomized Response (Inquiry)

Because many individuals fear invasion of their privacy, they do not respond truthfully to sensitive survey questions. For example, if an individual is asked "Have you ever taken drugs for depression?", the individual may lie and respond *No*, thereby biasing the results of the survey. Warner [Warn65] introduced a "randomized response" technique to deal with this problem. The technique is applied at the time the data is gathered—that is, at the time of inquiry.

The basic idea is that the individual is asked to draw a question at random from a set of questions, where some of the questions are sensitive and some are not. Then the individual is asked to respond to that question, but to not reveal the question answered.

Bourke and Dalenius [Bour75] and Dalenius [Dale76] discuss several strategies for doing this. Warner's original scheme is illustrated by the following sample questions:

1. Were you born in August?
2. Have you ever taken drugs for depression?

The objective of the survey is to determine the percentage of the population who have taken drugs for depression. The respondent picks one question at random (e.g., by tossing a coin), and then answers *Yes* or *No*. Assuming that the percentage of the population born in August is known, the percentage who have taken drugs for depression can be deduced from the number of *Yes* answers (see exercises at end of chapter).

Although it is not possible to determine whether an individual has taken drugs for depression from a *Yes* answer, a *Yes* answer might seem potentially more revealing than a *No* answer. This lack of symmetry may, therefore, bias the results, though the bias will be less than if the respondent is given no choice at all and asked Question 2 directly.

Bourke suggested a symmetric scheme to remove this bias. Here the respondent is asked to draw a card at random from a deck and respond with the number on the card that describes him. His approach is illustrated next:

Card 1: (1) I have taken drugs for depression.
(2) I have not taken drugs for depression.

Card 2: (1) I was born in August.
(2) I was not born in August.

Card 3: (1) I have not taken drugs for depression.
(2) I have taken drugs for depression.

Because a response of "1" (or "2") is linked to both having taken drugs and not having taken drugs, an individual might feel less threatened about responding truthfully to the survey.

Let p_i be the proportion of card (i) in the deck $(i = 1, 2, 3)$, let b be the probability that an individual is born in August. Suppose that N individuals are surveyed, and that N_1 of them respond "1". To determine the probability d that an individual has taken drugs for depression, we observe that the expected number of individuals responding "1" is given by

$$E(1) = [p_1 d + p_2 b + p_3(1 - d)]N .$$

If $p_1 \neq p_3$, we can solve for d, getting

$$d = \frac{\frac{E(1)}{N} - p_2 b - p_3}{p_1 - p_3} .$$

Because N_1 is an estimate of $E(1)$, we can estimate d with

$$\hat{d} = \frac{\frac{N_1}{N} - p_2 b - p_3}{p_1 - p_3} .$$

Example:
Let $N = 1000$, $p_1 = .4$, $p_2 = .3$, $p_3 = .3$, $b = .1$, and $N_1 = 350$. Then $d = .2$. ■

The randomized response technique can be viewed as an example of data perturbation, where each individual perturbs the data by a random selection. For obvious reasons, it is not applicable to general-purpose systems where accuracy of the data is essential (e.g., a hospital database or the student record database).

6.6 SUMMARY

Although it is not surprising that simple inference controls can be subverted, it is surprising that often only a few queries are required to do so. A query-set-size control, while necessary, is insufficient because it can be subverted with trackers. A query-set-overlap control is infeasible to implement, possible to subvert, and too restrictive to be of practical interest.

Controls that provide a high level of security by restricting statistics are not always practical for general-purpose database systems. Cell suppression may be too time-consuming to apply to on-line dynamic databases. Data swapping may be limited to the publication of microstatistics. Partitioning at the logical level is an attractive approach, but it could limit the free flow of statistical information if the partitions do not match the query sets needed by researchers, or if the set of available statistics are not sufficiently rich. If partitioning is used, it must be included in the initial design of the database; it cannot be added to an arbitrary

database structure. The S_m / N–criterion is an alternative to partitioning that may be less restrictive and easier to integrate into an existing database system.

Controls that add noise to the statistics are an interesting alternative because they are efficient and allow the release of more nonsensitive statistics. They are also simple to implement and could be added to almost any existing database system. Two controls that look promising here are random-sample queries and data perturbation. These controls could augment simple restriction techniques such as a query-set-size control and the S_m / N–criterion.

EXERCISES

6.1 Show how Good's *GP* can be compromised under a query-set-size restriction of $n = 3$ using

a) An individual tracker attack [Eq. (6.4)], where $C1 = CS$ and $C2 = Male \bullet 1978$.
b) A general tracker attack [Eq. (6.5)], where $T = Male$.
c) A double tracker attack [Eq (6.7)], where $T = CS$ and $U = CS + EE$.
d) A union tracker attack [Eq. (6.9)], where $T_1 = Female$ and $T_2 = Male$.

6.2 Show Moore's *GP* can be compromised by a linear system attack under an overlap constraint of $r = 1$ using key-specified queries for sums, where the size of the key list is $k = 4$ (see Section 6.3.3).

6.3 Let $q_1, \ldots, q_m$ be key-specified queries of the form **median**$(\{i_1, \ldots, i_k\}, A)$, such that no two queries have more than one record in common and such that each query set consists of some subset of k records in the set $\{1, \ldots, m - 1\}$ (assume m is large enough that this is possible). Assuming all x_j are unique $(1 \leq j \leq m - 1)$, show that some x_j can be determined from $q_1, \ldots, q_m$.

6.4 Let C be a formula with initial query set $\{1, 2, 3\}$. Suppose that records satisfying C can be added to or deleted from the database in pairs, and that the query **sum**(C, A) can be posed between such additions or deletions. Assuming a query-set-size restriction of $n = 3$, construct a minimal sequence of insertions, deletions, and queries that disclose x_1 when none of the x_i are known. Generalize your result to $C = \{1, 2, \ldots, s\}$ for any odd s and $n = s$.

6.5 Show the suppressed entries in Table 6.6 cannot be exactly determined from the remaining entries without supplementary knowledge by showing the four unknowns are related by only three linearly independent equations. Determine the best possible interval estimate for each unknown cell.

6.6 Prove the cell suppression bound $q^+(C)$ defined by Eq. (6.10) is subadditive by showing that

$$q^+(C + D) \leq q^+(C) + q^+(D) .$$

6.7 Prove the cell suppression bound $q^-(C)$ defined by Eq. (6.11) is not superadditive by showing that the following may hold:

$$q^-(C) + q^-(D) > q^-(C + D) .$$

6.8 Let $q = 100 + 5 + 5 + 5 + 5 = 120$. Show q is sensitive under a 1-respondent, 75%-dominance sensitivity criterion. Determine a lower bound on an acceptable upper estimate q^+ of q[Eq. (6.10)], and an upper bound on an acceptable lower estimate q^- of q[Eq. (6.11)].

6.9 Given the following table of rounded values, determine the exact values assuming that systematic rounding base 5 was used.

10	10	25
10	10	25
25	25	50

6.10 Karpinski showed it may be possible to subvert a systematic rounding control when records can be added to the database [Karp70]. As an example, suppose that **count**(C) = 49 and systematic rounding base 5 is used; thus, the system returns the value 50 in response to a query for **count**(C). Show how a user can determine the true value 49 by adding records that satisfy C to the database, and posing the query **count**(C) after each addition. Can this attack be successfully applied with random rounding?

6.11 Consider Beck's method of data perturbation. Show that S' as defined by Eq. (6.12) is an unbiased estimator of the true sum S. Derive Eqs. (6.13) and (6.14) for Var (S').

6.12 Consider the 2-transformable database D in Table 6.13 of Section 6.5.4. Suppose that a user knows Jane is represented in the database and that Jane is a *CS* major. Explain why it is not possible to deduce Jane's *GP* from 0-, 1-, and 2-order statistics without supplementary knowledge. Show that it is possible, however, to deduce Jane's *GP* if the user knows that her *GP* is not 4.0.

6.13 Consider the method of randomized response, and suppose that $N = 1000$ individuals participate in a population survey. Each individual is asked to toss a coin to determine which of the following questions to answer:

(1) Have you ever been to New York?
(2) Have you ever taken drugs for depression?

Suppose that 300 people respond *Yes*, and that it is known that roughly 40% of the population surveyed has visited New York. What is the expected percentage of the population surveyed who have taken drugs for depression? Now, suppose that it is learned that Smith answered *Yes* to the survey, but it is not known whether Smith has been to New York. What is the probability Smith answered Question (2)? What is the probability Smith has taken drugs for depression?

REFERENCES

Achu79. Achugbue, J. O. and Chin, F. Y., "The Effectiveness of Output Modification by Rounding for Protection of Statistical Databases," *INFOR* Vol. 17(3) pp. 209–218 (Mar. 1979).

Beck80. Beck, L. L., "A Security Mechanism for Statistical Databases," *ACM Trans. on Database Syst.* Vol. 5(3) pp. 316–338 (Sept. 1980).

Bour75. Bourke, P. D. and Dalenius, T., "Some New Ideas in the Realm of Randomized Inquiries," Confidentiality in Surveys, Report No. 5, Dept. of Statistics, Univ. of Stockholm, Stockholm, Sweden (Sept. 1975).

Camp77. Campbell, D. T., Boruch, R. F., Schwartz, R. D., and Steinberg, J., "Confidentiality-Preserving Modes of Access to Files and to Interfile Exchange for Useful Statistical Analysis," *Eval. Q.* Vol. 1(2) pp. 269–299 (May 1977).

Caus79. Causey, B., "Approaches to Statistical Disclosure," in *Proc. Amer. Stat. Assoc., Soc. Stat. Sec.* Washington, D.C. (1979).

Chin79. Chin., F. Y. and Ozsoyoglu, G., "Security in Partitioned Dynamic Statistical Databases," pp. 594–601 in *Proc. IEEE COMPSAC Conf.* (1979).

Chin80. Chin, F. Y. and Ozsoyoglu, G., "Auditing and Inference Control in Statistical Databases," Univ. of Calif., San Diego, Calif. (Dec. 1980).

Chin81. Chin, F. Y. and Ozsoyoglu, G., "Statistical Database Design," *ACM Trans. on Database Syst.* Vol. 6(1) pp. 113–139 (Mar. 1981).

Codd70. Codd, E. F., "A Relational Model for Large Shared Data Banks," *Comm. ACM* Vol. 13(6) pp. 377–387 (1970).

Codd79. Codd, E. F., "Extending the Database Relational Model to Capture More Meaning," *ACM Trans. on Database Syst.* Vol. 4(4) pp. 397–434 (Dec. 1979).

Cox76. Cox, L. H., "Statistical Disclosure in Publication Hierarchies," presented at the Amer. Stat. Assoc. Meeting, Stat. Comp. Sec. (1976).

Cox78. Cox, L. H., "Suppression Methodology and Statistical Disclosure Control," Confidentiality in Surveys, Report No. 26, Dept. of Statistics, Univ. of Stockholm, Stockholm, Sweden (Jan. 1978).

Cox80. Cox, L. H., "Suppression Methodology and Statistical Disclosure Control," *J. Amer. Stat. Assoc.* Vol. 75(370) pp. 377–385 (June 1980).

Cox81. Cox, L. H. and Ernst, L. R., "Controlled Rounding," U.S. Bureau of the Census, Washington, D.C. (Jan. 1981).

Dale76. Dalenius, T., "Confidentiality in Surveys," *J. Statistical Research* Vol. 10(1) pp. 15–41 (Jan. 1976).

Dale77. Dalenius, T., "Towards a Methodology for Statistical Disclosure Control," *Statistisk tidskrift* Vol. 15 pp. 429–444 (1977).

Dale78. Dalenius, T. and Reiss, S. P., "Data-Swapping—A Technique for Disclosure Control," Confidentiality in Surveys, Report No. 31, Dept. of Statistics, Univ. of Stockholm, Stockholm, Sweden (May 1978).

Dale79. Dalenius, T. and Denning, D., "A Hybrid Scheme for Statistical Release," Computer Sciences Dept., Purdue Univ., W. Lafayette, Ind. (Oct. 1979).

Dant63. Dantzig, G., *Linear Programming and Extensions,* Princeton Univ. Press, Princeton, N.J. (1963).

Davi78. Davida, G. I., Linton, D. J., Szelag, C. R., and Wells, D. L., "Data Base Security," *IEEE Trans. on Software Eng.* Vol. SE-4(6) pp. 531–533 (Nov. 1978).

DeMi77. DeMillo, R. A., Dobkin, D. P., and Lipton, R. J., "Even Databases That Lie Can Be Compromised," *IEEE Trans. on Software Eng.* Vol. SE-4(1) pp. 73–75 (Jan. 1977).

DeMi78. DeMillo, R. A. and Dobkin, D. P., "Combinatorial Inference," pp. 27–35 in *Foundations of Secure Computation,* Academic Press, New York (1978).

Denn79. Denning, D. E., Denning, P. J., and Schwartz, M. D., "The Tracker: A Threat to Statistical Database Security," *ACM Trans. on Database Syst.* Vol. 4(1) pp. 76–96 (March 1979).

Denn80a. Denning, D. E. and Schlörer, J., "A Fast Procedure for Finding a Tracker in a Statistical Database," *ACM Trans. on Database Syst.* Vol. 5(1) pp. 88–102 (Mar. 1980).

Denn80b. Denning, D. E., "Corrigenda: Linear Queries in Statistical Databases," *ACM Trans. on Database Syst.* Vol. 5(3) p. 383 (Sept. 1980).

Denn80c. Denning, D. E., "Secure Statistical Databases Under Random Sample Queries," *ACM Trans. on Database Syst.* Vol. 5(3) pp. 291–315 (Sept. 1980).

Denn81. Denning, D. E., "Restricting Queries That Might Lead to Compromise," in *Proc. 1981 Symp. on Security and Privacy,* IEEE Computer Society (Apr. 1981).

Denn82. Denning, D. E., Schlörer, J., and Wehrle, E., "Memoryless Inference Controls for Statistical Databases," manuscript in preparation (1982).

Dobk79. Dobkin, D., Jones, A. K., and Lipton, R. J., "Secure Databases: Protection Against User Inference," *ACM Trans. on Database Syst.* Vol. 4(1) pp. 97–106 (Mar. 1979).

Feig70. Feige, E. L. and Watts, H. W., "Protection of Privacy Through Microaggregation," in *Databases, Computers, and the Social Sciences,* ed. R. L. Bisco, Wiley-Interscience, New York (1970).

Fell72. Fellegi, I. P., "On the Question of Statistical Confidentiality," *J. Amer. Stat. Assoc.* Vol. 67(337) pp. 7–18 (Mar. 1972).

Fell74. Fellegi, I. P. and Phillips, J. L., "Statistical Confidentiality: Some Theory and Applications to Data Dissemination," *Annals Econ. Soc'l Measurement* Vol. 3(2) pp. 399–409 (Apr. 1974).

Fran77. Frank, O., "An Application of Information Theory to the Problem of Statistical Disclosure," Confidentiality in Surveys, Report No. 20, Dept. of Statistics, Univ. of Stockholm, Stockholm, Sweden (Feb. 1977).

Frie80. Friedman, A. D. and Hoffman, L. J., "Towards a Fail-Safe Approach to Secure Databases," pp. 18–21 in *Proc. 1980 Symp. on Security and Privacy,* IEEE Computer Society (Apr. 1980).

Hans71. Hansen, M. H., "Insuring Confidentiality of Individual Records in Data Storage and Retrieval for Statistical Purposes," *Proc. Fall Jt. Computer Conf.,* Vol. 39, pp. 579–585 AFIPS Press, Montvale, N.J. (1971).

Haq74. Haq, M. I., "Security in a Statistical Data Base," *Proc. Amer. Soc. Info. Sci.* Vol. 11 pp. 33–39 (1974).

Haq75. Haq, M. I., "Insuring Individual's Privacy from Statistical Data Base Users," pp. 941–946 in *Proc. NCC,* Vol. 44, AFIPS Press, Montvale, N.J. (1975).

Haq77. Haq, M. I., "On Safeguarding Statistical Disclosure by Giving Approximate Answers to Queries," *Int. Computing Symp.,* North-Holland, New York (1977).

Hoff70. Hoffman, L. J. and Miller, W. F., "Getting a Personal Dossier from a Statistical Data Bank," *Datamation* Vol. 16(5) pp. 74–75 (May 1970).

Hoff77. Hoffman, L. J., *Modern Methods for Computer Security and Privacy,* Prentice-Hall, Englewood Cliffs, N.J. (1977).

Kam77. Kam, J. B. and Ullman, J. D., "A Model of Statistical Databases and their Security," *ACM Trans. on Database Syst.* Vol. 2(1) pp. 1–10 (Mar. 1977).

Karp70. Karpinski, R. H., "Reply to Hoffman and Shaw," *Datamation* Vol. 16(10) p. 11 (Oct. 1970).

Liu80. Liu, L., "On Linear Queries in Statistical Databases," The MITRE Corp., Bedford, Mass. (1980).

Narg72. Nargundkar, M. S. and Saveland, W., "Random Rounding to Prevent Statistical Disclosure," *Proc. Amer. Stat. Assoc., Soc. Stat. Sec.,* pp. 382–385 (1972).

Olss75. Olsson, L., "Protection of Output and Stored Data in Statistical Databases," ADB-Information, 4, Statistika Centralbyrän, Stockholm, Sweden (1975).

Palm74. Palme, J., "Software Security," *Datamation* Vol. 20(1) pp. 51–55 (Jan. 1974).

Reis78. Reiss, S. B., "Medians and Database Security," pp. 57–92 in *Foundations of Secure Computation,* ed. R. A. DeMillo et al., Academic Press, New York (1978).

Reis79. Reiss, S. B., "The Practicality of Data Swapping," Technical Report No. CS-48, Dept. of Computer Science, Brown Univ., Providence, R.I. (1979).

Reis80. Reiss, S. B., "Practical Data-Swapping: The First Steps," pp. 38–45 in *Proc. 1980 Symp. on Security and Privacy,* IEEE Computer Society (Apr. 1980).

Sand77. Sande, G., "Towards Automated Disclosure Analysis for Establishment Based Statistics," Statistics Canada (1977).

Schl75. Schlörer, J., "Identification and Retrieval of Personal Records from a Statistical Data Bank," *Methods Inf. Med.* Vol. 14(1) pp. 7–13 (Jan. 1975).

Schl76. Schlörer, J., "Confidentiality of Statistical Records: A Threat Monitoring Scheme for On-Line Dialogue," *Meth. Inf. Med.,* Vol. 15(1), pp. 36 42 (1976).

Schl77. Schlörer, J., "Confidentiality and Security in Statistical Data Banks," pp. 101–123 in *Data Documentation: Some Principles and Applications in Science and Industry; Proc. Workshop on Data Documentation,* ed. W. Guas and R. Henzler, Verlag Dokumentation, Munich, Germany (1977).

Schl80. Schlörer, J., "Disclosure from Statistical Databases: Quantitative Aspects of Trackers," *ACM Trans. on Database Syst.* Vol. 5(4) pp. 467–492 (Dec. 1980).

Schl81. Schlörer, J., "Security of Statistical Databases: Multidimensional Transformation," *ACM Trans. on Database Syst.* Vol. 6(1) pp. 95–112 (Mar. 1981).

Schw77. Schwartz, M. D., "Inference from Statistical Data Bases," Ph.D. Thesis, Computer Sciences Dept., Purdue Univ., W. Lafayette, Ind. (Aug. 1977).

Schw79. Schwartz, M. D., Denning, D. E., and Denning, P. J., "Linear Queries in Statistical Databases," *ACM Trans. on Database Syst.* Vol. 4(1) pp. 476–482 (Mar. 1979).

Smit77. Smith, J. M. and Smith, D. C. P., "Database Abstractions: Aggregation and Generalization," *ACM Trans. on Database Syst.* Vol. 2(2) pp. 105–133 (June 1977).

U.S.78. U.S. Dept. of Commerce, "Report on Statistical Disclosure and Disclosure-Avoidance Techniques," U.S. Government Printing Office, Washington, D.C. (1978).

Warn65. Warner, S. L., "Randomized Response: A Technique for Eliminating Evasive Answer Bias," *J. Amer. Stat. Assoc.* Vol. 60 pp. 63–69 (1965).

Yu77. Yu, C. T. and Chin, F. Y., "A Study on the Protection of Statistical Databases," *Proc. ACM SIGMOD Int. Conf. Management of Data,* pp. 169–181 (1977).