CORDS: Automatic Generation of Correlation Statistics in DB2

Ihab F. Ilyas[1] Volker Markl[2] Peter J. Haas[2] Paul G. Brown[2] Ashraf Aboulnaga[2]

[1] University of Waterloo
200 University Avenue
Waterloo, Ontario N2L 3G1, Canada
ilyas@uwaterloo.ca

[2] IBM Almaden Research Center
650 Harry Road, K55/B1
San Jose, CA, 95139
marklv,phaas,pbrown1,aashraf@us.ibm.com

Abstract

When query optimizers erroneously assume that database columns are statistically independent, they can underestimate the selectivities of conjunctive predicates by orders of magnitude. Such underestimation often leads to drastically suboptimal query execution plans. We demonstrate CORDS, an efficient and scalable tool for automatic discovery of correlations and soft functional dependencies between column pairs. We apply CORDS to real, synthetic, and TPC-H benchmark data, and show that CORDS discovers correlations in an efficient and scalable manner. The output of CORDS can be visualized graphically, making CORDS a useful mining and analysis tool for database administrators. CORDS ranks the discovered correlated column pairs and recommends to the optimizer a set of statistics to collect for the "most important" of the pairs. Use of these statistics speeds up processing times by orders of magnitude for a wide range of queries.

1 Introduction

CORDS is a data-driven tool that automatically discovers correlations and soft functional dependencies (FDs) between pairs of columns and, based on these relationships, recommends a set of statistics for the query optimizer to maintain. By *correlations*, we mean general statistical dependencies, not merely approximate linear relationships. By a *soft* FD between columns C_1 and C_2, we mean a generalization of the classical notion of a *hard* FD in which the value of C_1 completely determines the value of C_2. In a soft FD (denoted by $C_1 \Rightarrow C_2$), the value of C_1 determines the value of C_2 not with certainty, but merely with high probability. CORDS focuses on column pairs because this greatly simplifies the algorithms, and experiments have shown that the marginal benefit of capturing n-way dependencies for $n > 2$ is relatively small. The column pairs examined by CORDS can be in the same table or different tables. We briefly outline the CORDS system below; details may be found in [1]. In developing CORDS, we have found that algorithmic simplicity and judicious use of sampling can lead to efficient and highly scalable self-management algorithms that are suitable for immediate incorporation into commercial database systems.

Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004

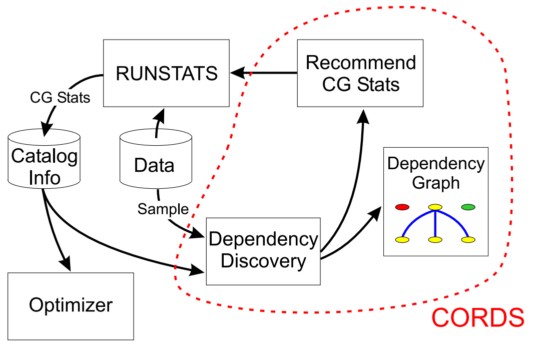


Figure 1: Role of CORDS in query optimization.

2 CORDS Overview

Figure 1 illustrates the role of CORDS in DB2 query optimization. CORDS uses system-catalog statistics that are collected by the DB2 RUNSTATS utility together with samples of the base data to discover dependencies between column pairs. The "most important" column pairs are identified and RUNSTATS is instructed to collect *column group* (CG) statistics on each recommended column pair; these statistics are then stored in the system catalog and available to the query optimizer.

CORDS first searches for column pairs that are likely to be related in an interesting and useful way by systematically enumerating candidate pairs and simultaneously pruning unpromising candidates using a flexible set of heuristics.[1] Pruning rules can include constraints on allowable data types, constraints on statistical properties of the columns, constraints based on schema information, constraints based on workload information, and so forth. CORDS also uses catalog statistics to identify certain "trivial" instances of correlation and, simultaneously, to eliminate certain columns from further consideration. For each surviving candidate, CORDS collects a sample of value-pairs from the columns. CORDS then analyzes the number of distinct values in the sampled columns to test for a soft FD. If no FD is found, CORDS then applies a robust chi-squared analysis to test for statistical dependence. As shown in [1], the required sample size for the chi-squared analysis is essentially independent of the database size, which is why CORDS is scalable to very large databases.

In more detail, CORDS discovers the following properties and relationships:

- *Trivial Cases*: Prior to candidate generation, CORDS examines the system catalog to identify two trivial cases: "soft" keys and "trivial" columns. A *soft* key is "almost" a key w.r.t the number of distinct values. A soft key is trivially statistically correlated with every other column in the table because a value from that column, with high probability, determines the row, and hence the value in any other column. A *trivial* column is a single-valued or NULL column. The value in any row of such a column is trivially "determined" by the values in any other column, leading to spurious correlations.

- *Soft FDs*: The *strength* of a soft FD $C_1 \Rightarrow C_2$ is computed as $|C_1|/|C_1, C_2|$, where $|C_1|$ is the number of distinct values in column C_1 and $|C_1, C_2|$ is the number of distinct value-pairs for columns C_1 and C_2. This strength is always less than or equal to 1, with a strength of 1 indicating a hard functional dependency. CORDS estimates $|C_1|$ and $|C_1, C_2|$, and hence the strength, from the sample and declares the existence of a soft FD if the estimated strength is greater than a prespecified value.

- *Correlations*: Column pairs in this category are those pairs for which CORDS does not detect a soft FD but does detect statistical dependence using the sample-based chi-squared analysis.

The output of CORDS may optionally be displayed as a dependency graph, facilitating database design and data mining. In any case, CORDS recommends to the optimizer sets of CG statistics to maintain. Such recommendations are necessary because, as we show in our demonstration, real-world databases typically contain a very large number of correlations and soft FDs. The overhead of collecting and maintaining statistics for every correlated column pair would outweigh any benefits resulting from a better choice of query plans. The CORDS recommendation algorithm ranks the discovered soft FDs based on their strengths and the discovered correlations on their "benefit" to the query optimizer. A useful measure of benefit is the magnitude of the adjustment factor used by the optimizer to correct selectivity estimates that are based on faulty independence assumptions when CG statistics are available. The larger the adjustment factor, the more serious the estimation error that is corrected. Our demonstration focuses on a simple CG statistic, namely $|C_1, C_2|$ as defined above, and on simple conjunctive predicates of the form "$C_1 = x$ AND $C_2 = y$." A naive estimate of the selectivity for such a predicate, assuming uniform data frequencies and independence between the columns, is $1/|C_1| \cdot 1/|C_2|$, and the adjustment factor that corrects for a faulty independence assumption is $|C_1|\,|C_2|/|C_1, C_2|$. Although this adjustment factor does not correct for a faulty uniformity assumption, we have found in practice that correcting for a faulty independence assumption usually eliminates most of the error.

3 Demonstration

Using synthetic, real-world, and benchmark databases, we demonstrate the application of CORDS and show how CORDS can improve query performance by orders of magnitude. We run CORDS on top of DB2. The scenario of the demonstration for a given database is as follows:

1. We run CORDS using a set of parameters that determine, e.g., the scope of correlation discovery (intra-table or inter-table) and the candidate pruning heuristics.

2. CORDS outputs the discovered correlations and soft FDs in the form of dependency graphs, such as the dependency graphs shown in Figure 2 for the tables in the TPC-H database. Nodes (ovals in Figure 2) correspond to columns and arcs correspond to correlations or soft FDs.[2] The thick-

[1]Technically, a candidate consists of a pair of columns (C_1, C_2) along with a *pairing rule* that specifies which particular C_1 values get paired with which particular C_2 values to form the set of potentially correlated value-pairs. When the columns lie in the same table and each C_1 value is paired with the C_2 value in the same row, the pairing rule is trivial; when the columns lie in different tables, the pairing rule corresponds to a join predicate between the tables. Self-joins are also permitted.

[2]The name CORDS was partially inspired by the visual "cords" that connect correlated columns.

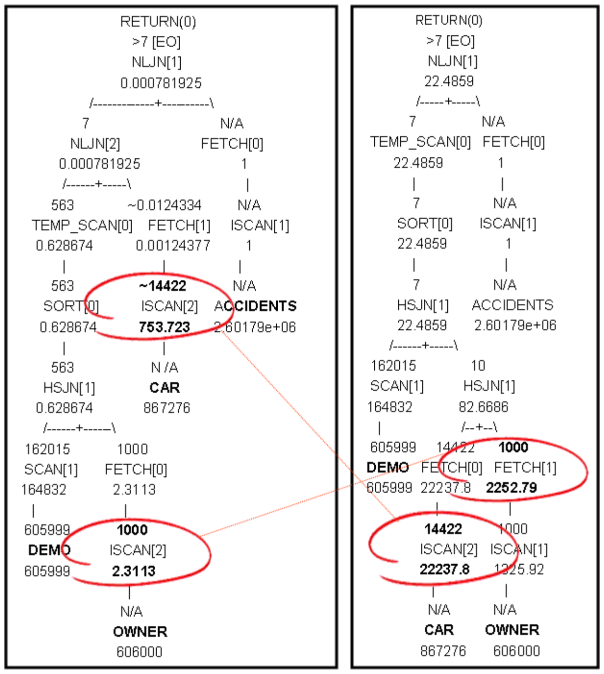
ness or color of the arcs can be used to show the strength of the relationships. For example, the thickness of an arc that represents a soft FD can be an increasing function of the estimated strength. Soft keys and trivial columns are indicated by specified node colors.

3. We apply the CORDS statistics-recommendation algorithm that ranks the discovered correlations and soft FDs based on their benefit to the optimizer. For example, Table 1 gives the ranking of the discovered correlations in a synthetic database of car-accident data. Although CORDS discovers a correlation between the two columns `ACCIDENTS.Driver` and `ACCIDENTS.Damage` (last row in the table), the corresponding adjustment factor equals 1. CORDS does not recommend collection of statistics for this pair of columns, since the presence of such statistics would not change any optimizer cardinality estimates. On the other hand, collecting statistics on the first pair of columns in Table 1 adjusts optimizer estimates by a factor of roughly 3070.

4. We collect CG statistics based on the CORDS recommendations and we contrast the optimizer estimates for several queries both with and without these statistics. We show the actual query execution plan in each case, augmented by the actual and estimated cardinality for each operator. Figure 3 displays such a pair of execution plans for the following query:

```
SELECT o.Name, a.Driver
FROM OWNER o, CAR c, DEMOGRAPHICS d, ACCIDENTS a
WHERE
  c.OwnerID = o.ID AND o.ID = d.OwnerID AND
  c.ID = a.ID AND c.Make = 'Mazda' AND
  c.Model = '323' AND o.Country3 = 'EG' AND
  o.City = 'Cairo' AND d.Age < 30;
```

In the figure, actual cardinalities (i.e., number of rows processed) are displayed above each query operator and estimated cardinalities are displayed below. E.g., in the original query plan without CG statistics (left side), the upper circled index scan processes 14,222 rows, the number of rows that satisfy the predicate `c.make = 'Mazda' AND c.model = '323'`. The optimizer underestimates this number by almost a factor of 20 as 753.723. Using the CG statistics recommended by CORDS, the optimizer improves this estimate to roughly within a factor of 2, as can be seen in the modified plan on the right. Similarly, the optimizer originally underestimates the number of rows satisfying the predicate `o.country3 = 'EG' AND o.city = 'Cairo'` by a factor of almost 500 (see the lower circled index scan on the left). Using CG statistics, the optimizer reduces the error to a factor of about 2 in the modified query plan. The modified plan that is based on the new statistics is orders of magnitude faster than the original.



```
                RETURN(0)
                 >7 [EO]
                 NLJN[1]
               0.000781925
          /------------+---------\
       7                        N/A
    NLJN[2]                   FETCH[0]
  0.000781925                    1
   /------+-----\                |
   563          ~0.0124334      N/A
TEMP_SCAN[0]    FETCH[1]      ISCAN[1]
  0.628674     0.00124377        1
     |              |            |
    563          ~14422         N/A
   SORT[0]       ISCAN[2]    ACCIDENTS
  0.628674       753.723    2.60179e+06
     |              |
    563            N/A
   HSJN[1]         CAR
  0.628674       867276
 /------+-----\
162015      1000
SCAN[1]    FETCH[0]
164832      2.3113
  |           |
605999      1000
DEMO       ISCAN[2]
605999      2.3113
              |
             N/A
            OWNER
           606000
```

```
          RETURN(0)
           >7 [EO]
           NLJN[1]
           22.4859
        /-----+-----\
       7            N/A
TEMP_SCAN[0]    FETCH[0]
   22.4859          1
      |             |
      7            N/A
   SORT[0]       ISCAN[1]
   22.4859          1
      |             |
      7            N/A
   HSJN[1]      ACCIDENTS
   22.4859     2.60179e+06
  /------+-----\
162015         10
SCAN[1]      HSJN[1]
164832       82.6686
  |          /--+--\
605999   14422      1000
DEMO    FETCH[0]   FETCH[1]
605999  22237.8    2252.79
           |          |
         14422       1000
        ISCAN[2]   ISCAN[1]
        22237.8    1325.92
           |          |
          N/A        N/A
          CAR       OWNER
        867276     606000
```

Figure 3: Query execution plans before and after collecting statistics on correlated columns.

The databases that we use are as follows.

- *Synthetic Data*: The *Accidents* database is synthetically generated and represents car-accident records. When generating the database, we created a predetermined set of correlations by manipulating the data distributions. The database also contains a number of natural hard and soft FDs, e.g., `Model` $\Rightarrow$ `Make` in the `CAR` table.

- *Benchmarking Data*: The *TPC-H* benchmark dataset was not explicitly designed to have specified correlations between different columns. CORDS can discover hidden correlations that may not be known to the database designer or administrator.

- *Real-world Data*: We apply CORDS to several real-world databases. These include a subset of the *Census* database and the *Auto* database, which contains motor vehicle information and comprises over 20 tables and hundreds of columns.

References

[1] I. F. Ilyas, V. Markl, P. J. Haas, P. G. Brown, and A. Aboulnaga. CORDS: Automatic discovery of correlations and soft functional dependencies. In *Proc. 2004 ACM SIGMOD*, 2004.

First Column	Second Column	Adj. Factor
DEMOGRAPHICS.Salary	DEMOGRAPHICS.Assets	3069.69
CAR.Color	CAR.OwnerID	311.25
DEMOGRAPHICS.Assets	DEMOGRAPHICS.Age	77.71
ACCIDENTS.CarID	ACCIDENTS.Year	30.06
CAR.Model	CAR.OwnerID	27.14
CAR.Year	CAR.Color	21.80
CAR.Make	CAR.OwnerID	16.00
CAR.Model	CAR.Color	14.91
CAR.Make	CAR.Color	9.15
CAR.Model	CAR.Year	2.31
CAR.Year	CAR.Make	1.88
ACCIDENTS.Driver	ACCIDENTS.With	1.13
ACCIDENTS.Driver	ACCIDENTS.SeatBeltOn	1.00
ACCIDENTS.Damage	ACCIDENTS.With	1.00
ACCIDENTS.Driver	ACCIDENTS.Damage	1.00

Table 1: Discovered correlations ranked by adjustment factor

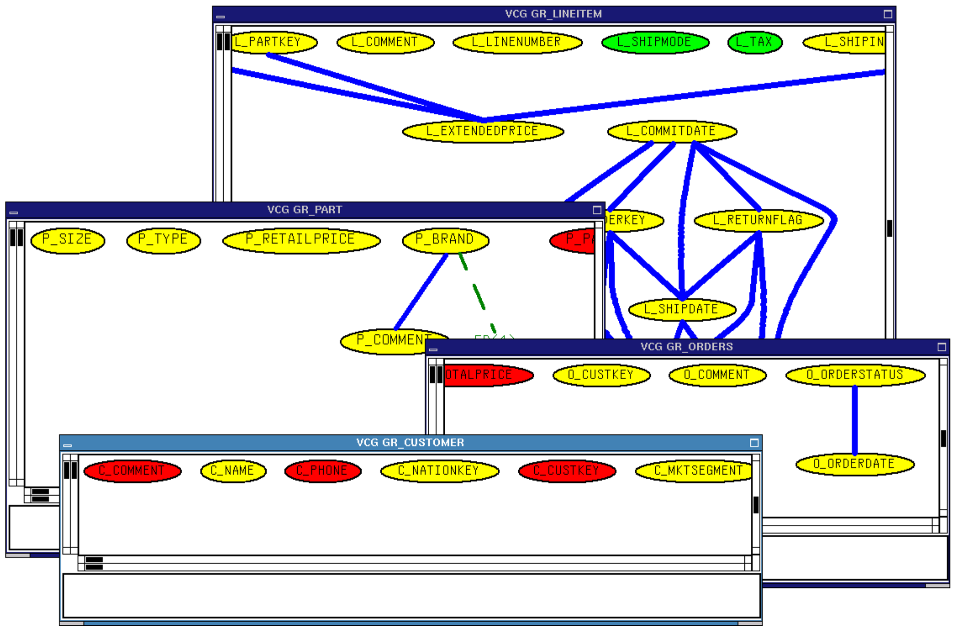


Figure 2: Snapshot of the automatically generated dependency graphs for TPC-H (sample size 10,000 records).