DISCRETE AND CONTINUOUS
DYNAMICAL SYSTEMS SERIES S
Volume **8**, Number **6**, December **2015**

doi:10.3934/dcdss.2015.8.1341

pp. **1341–1356**

STATISTICAL QUERY-BASED RULE DERIVATION SYSTEM BY BACKWARD ELIMINATION ALGORITHM

Ayla Sayli* and Ayse Oncu Sarihan

Dept. of Mathematical Engineering
Yildiz Technical University
Istanbul, 34210, Turkey

Abstract. Computers play a serious role in human life, especially web-based applications running twenty four hours per day. These applications are based on relational database management system and they receive many queries from the users. These queries are executed in the commercial systems one by one without any consideration of past experiences and data analysis. The execution of queries can be faster if some rules were derived from the past queries. In this paper, we propose a statistical query-based rule derivation system by the backward elimination algorithm, which analysis the data based on the past queries in order to derive new rules, and then it uses these rules for the execution of new queries. The computational results are presented and analysed that the system is very efficient and promising.

1. **Introduction.** The approach of Semantic Query Optimisation (SQO) was researched as semantic reasoning by Hammer and Mcleod [8]. It was then announced by King [13] and the complete system was developed in the late 1980s [7, 3, 24]. The main advantage of SQO is its ability to derive rules and then use them to reduce the execution time of future queries. The use of rules from past experiences makes the optimiser more intelligent. The approach has several components: Query Representation, Query Optimisation and Rule Maintenance, and Automatic Rule Derivation.

The first component of the approach is to represent a query in a query language in order to express it in a way that can be understood by the chosen database environment. Many Relational Database Management Systems use the Structural Query Language and then internally the relational algebra and query trees are mainly used for this purpose [3, 20, 10, 6].

The second component is optimising a given query. In the optimisation process, two situations may occur which are named as Query Answering and Query Refutation. In the first situation, the part of the user query conflicts with any matching rule in the rule set. In the second situation, the answer set can be found from the matching rules, without any need to access the database. These special situations are the most profitable occurrences of the approach, with substantial time savings (%99) reported in [20]. Otherwise the query optimiser constructs alternative queries using the rule set and then selects the best alternative as an optimum query. In

2010 *Mathematics Subject Classification.* Primary: 68P15; Secondary: 68Q55.

Key words and phrases. Rule derivation, backward elimination algorithm, semantic query optimisation, statistics, relational database management systems.

* Corresponding author: Ayla Sayli.

most situations, the execution time of queries could be reduced, especially when the consequent of a rule contains the primary key and indexes [13, 7, 3, 20, 10, 6, 1]. Moreover if n is the number of the matching rules, the number of alternative queries can be found from the equation of $(2^n - 1)$. For example if 10 matching rules are found for the original query, then the total number of alternative queries is 1024. It can be expected that the number of alternative queries increases rapidly. For this reason, two time consuming problems occur: How to construct the alternative queries and How to optimise the alternative queries to select the optimum query Although these problems are not within the scope of this paper directly, if the number of the rules can be limited in the rule derivation process, this would reduce the construction time and query optimisation time indirectly. For this purpose, our system can be used efficiently.

The third component is used to keep the rules up to date for changes in the databases because these rules represent a time-dependent property of a database at a particular database state. When updates are made on the database, these rules may become invalid. Therefore derived rules need to be maintained to be consistent with the current database state [25, 12, 14, 18].

The last component is the automatic rule derivation to learn new rules based on the conditions of the given query and the query answer from the database. There are a number of automatic rule derivation techniques [8, 13, 7, 3, 24, 9, 23, 2, 22]. Despite the advantages of the SQO approach, the automatic rule derivation takes a considerably long time and the time increases according to the database size. Another problem is that the total number of rules in the rule set may increase rapidly. In order to solve these problems, many researchers suggest using statistics to measure the dependencies on the database [8, 13, 7, 3, 24, 20, 9, 23, 2, 22, 4, 15, 11, 5, 16].

Traditional approach of Sematic Query Optimisation (SQO) in Figure 1 starts with the users query, and then the query is rewritten in rule representation [8, 13, 7, 3, 24]. If the represented query contains a SELECT clause in the view definition language, the query optimisation takes the parts of the query in order to execute in the relational database systems. After the query execution, the answer set of this query and the parts of the represented query are used to trigger the automatic rule derivation in order to learn new rules (if possible), and then rules are saved in the rule set. These rules are used for the query transformation and optimisation in upcoming queries. Moreover, when the next query is entered into the system, in query transformation and optimisation, the parts of this query is compared with the rules from the rules set in rule matching step and when the matching is found, then the matched rule(s) can be used to construct the alternative queries in order to select the optimum query, which can be executed faster than the original query given by user. If the clause is one of INSERT, UPDATE and DELETE of data manipulation language. The clause is taken into the process of the rule maintenance manager. It is important to ensure the accuracy of the rule set. Another issue of the rule maintenance manager in Figure 1 is to keep the rule set accurate for new coming rules in case of having the same antecedent attribute and the consequent attribute with different comparison operator and values in order to manipulate their boundaries correctly. For this purpose, our previous papers can be taken into account [21, 19, 17].

In this paper, a Statistical Query-based Rule Derivation System by Backward Elimination Algorithm is proposed. Our system takes the condition(s) of the original

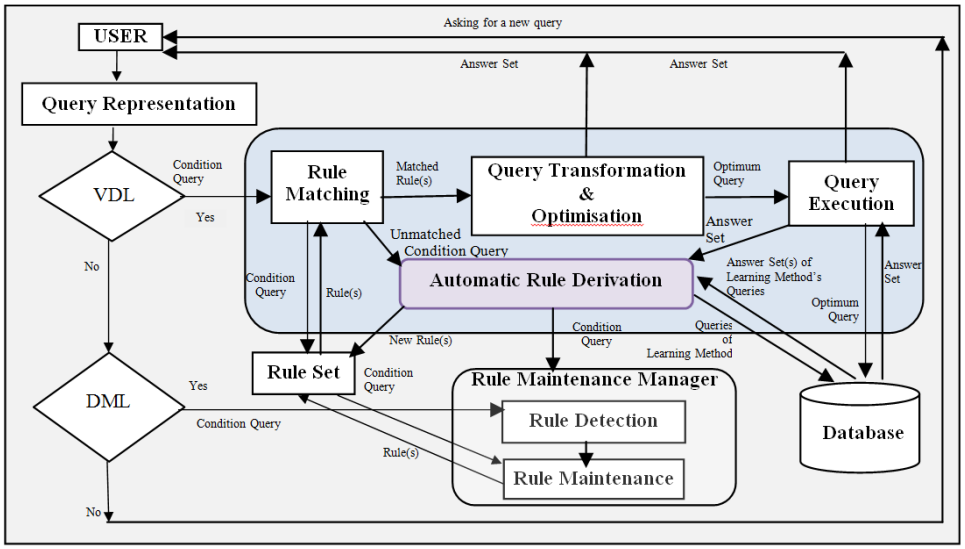


FIGURE 1. Traditional Approach of Semantic Query Optimisation

query and the answer set of a query, then it specifies the attributes of relation in two groups as dependent and independent attributes. After that it examines values of these attributes to see whether these attributes are related or not, by the use of the backward elimination algorithm. If they are not, the first elimination is done. Otherwise the system continues with multiple regression analysis and calculates the standardised regression coefficients. Then it estimates the coefficient ratio, which is the averaged rank of the coefficients of the linear multiple regression analysis. The second elimination is done by the use of this calculated ratio. The last but least important, the system ends with the rule construction with remaining attributes.

Our system is very efficient, fast and dynamic. It can work for any table on any database at any time, without any need for reconstruction. Computational results of the system show that it limits the number of rules easily and can reduce the rule derivation time and the query optimisation time. Main concepts of the SQO approach are explained in Section 2. Our system of the automatic rule derivation is then introduced in Section 3. The computational results are given in Section 4. Finally, conclusions are given in Section 5.

2. Main concepts of semantic query optimisation.

There are two main concepts of SQO named as Simple Rules (described in Subsection 2.1) and Semantically Equivalent Queries (described in Subsection 2.2).

2.1. Simple rules.

A vast majority of researchers on this approach use the simple rule, which has a condition on the left-hand side (called Antecedent) and a condition on the right-hand side (called Consequent). This type of rule cannot contain more than one condition on either side. Other rules can be regarded as complex, which can have more conditions on one side. Although somehow complex rules can be expressed, using them in the query optimisation becomes too difficult; also the rule maintenance becomes problematical. Therefore the rule representation is limited to using simple rules [8, 13, 7, 3, 24, 20, 10, 6, 1, 25, 12, 14, 18, 9, 23, 2, 22]. Assuming

a table is R (x, y, z) where x, y and z, are attributes of R. A rule can be formed as x⊕a → y⊗b where a and b are constant values; ⊕ and ⊗ are one of comparison operators ¡, ¡=, ¿, ¿=, =, !=. When the antecedent is satisfied, then the consequent has to be true. A rule can be defined in the rule set in Table 1.

TABLE 1. Add caption

Rule	*Data Type*	*Description*	*Sample*
AntAtt	varchar	Antecedent Attribute(attribute of the left hand side of the rule)	X
AntOper	varchar	Antecedent Operator (comparison operator of the left hand side of the rule)	⊕
AntValue	float	Antecedent Value	a
AntIndex	bit	If antecedent attribute has index, AntIndex is one, otherwise zero	1 or 0
AntKey	bit	If antecedent attribute has key, AntKey is one, otherwise zero	1 or 0
AntRecord	bigint	Number of records satisfies the antecedent	
ConsAtt	varchar	Consequent Attribute (attribute of the right hand side of the rule)	y
ConsOper	varchar	Consequent Operator (comparison operator of the right hand side of the rule)	⊗
ConsValue	float	Consequent Value	b
ConsIndex	bit	If consequent attribute has index, ConsIndex is one, otherwise zero	1 or 0
ConsKey	bit	If consequent attribute has key, ConsKey is one, otherwise zero	1 or 0
ConsRecord	bigint	Number of records satisfies the consequent	
LT	float	Learning Time	

2.2. **Semantically equivalent queries.** Another important concept of the SQO approach is Semantically Equivalent Queries, which were described as the heart of the SQO approach [13] because the main idea of the SQO approach is to use rules to transform a given query into alternative queries that have the same answer set as the given query, but constructed semantically. For example, assume that the following two rules are learned for the EMPLOYEE table and kept in the rule set:

R_1 : If the employees name is Ayla SAYLI, the salary is higher than 40K:

$$\text{Name} = \text{Ayla SAYLI} \rightarrow \text{Salary} > 40\text{K}$$

R_2 : If the employees name is Ayla SAYLI, the social security number (SSN) is 98765:

$$Name = AylaSAYLI \rightarrow SSN = 98765$$

If we are looking at the detail of the employee whose name is Ayla SAYLI, the original query can be represented in SQL as follows: SELECT * FROM EMPLOYEE WHERE Name = Ayla SAYLI; Using R_1 and R_2 alternative queries can be built as Q1, Q2 and Q3 that are semantically equivalent to the original:

Q_1:SELECT * FROM EMPLOYEE WHERE Name = Ayla SAYLI and Salary > 40K;

Q_2:SELECT * FROM EMPLOYEE WHERE Name = Ayla SAYLI and SSN = 98765;

Q_3:SELECT * FROM EMPLOYEE WHERE Name = Ayla SAYLI and Salary > 40K and SSN = 98765;

These alternatives are then compared with each other in the query optimisation component to determine the most cost-efficient query. Further information for cost estimation can be found in [20].

Two other situations can happen during query processing as it mentioned in Section 1 as Query Answering and Query Refutation. For Query Answering, if we are looking at the social security number of the employee whose name is Ayla SAYLI, the query can be represented as follows:

SELECT SSN FROM EMLOYEE WHERE Name = Ayla SAYLI;

The answer set of this query would be found as 98765 by the use of R2 without accessing the database.

For Query Refutation, if we are looking at the detail of the employee whose social security number is 98765 and name is Ayla SAYLI, then this query can be represented as follows:

SELECT * FROM EMPLOYEE WHERE SSN = 98764 and Name = Ayla SAYLI;

The answer set of the query has to be NULL by the use of R2. Therefore we do not need to search the database as Query Answering.

It is necessary to underline differences between the concept of Semantically Equivalent Queries and that of Logically Equivalent Queries. In some situations, queries can be semantically equivalent but logically not. The first difference is that logical equivalence is based on established logical equations such as De-Morgans Laws, composition, decomposition and transition. This makes the logically equivalent queries give the same answer set for all database states. However, the semantic equivalence is changeable according to database states. The second difference is that semantic equivalence is based on rules which were derived before the construction of alternative queries. These derived rules may be violated during SQO when database updates are made. Therefore, the logically equivalent queries are clearly semantically equivalent but the semantically equivalent queries may not be logically equivalent [8, 13, 7, 3, 24, 20, 10, 6, 1, 25, 12, 14, 18, 9, 23, 2, 22].

3. **Statistical query-based rule derivation system by backward elimination algorithm.** It is mentioned before that statistics is suggested on analysing the data in order to derive the simple rules by many researchers [8, 13, 7, 3, 24, 20, 9, 23, 2, 22, 4, 15, 11, 5, 16]. In the statistical literature, there are different ways to find out the relationship between the dependent and independent attributes which are mainly calculated with correlations. In this paper, we aimed to eliminate the weak independent attribute(s) in order to limit the number of new rule derivation. Therefore we use multi-collinearity from the statistics for elimination which is given in Subsection 3.1. There are three approaches of eliminating attributes, which are "Backward Elimination", "Forward Selection" and "Bidirectional Elimination". Backward elimination starts with all candidate variables, tests the deletion of each variable using a chosen model comparison criterion, deletes the variable (if any) that improves the model the most when deleted, and repeats this process until no further improvement is possible. Forward selection starts with no variables in the model, tests the addition of each variable using a chosen model comparison criterion, adds the variable (if any) that improves the model the most, and repeats this process until none of the remaining attributes improves can be made to the model. Bidirectional elimination, a combination of the first two, tests at each step for variables to be included or excluded. In this study, we use the backward elimination which will be shown in Subsection 3.2. Our system is pictorially shown in Figure 2. It has six levels namely User Level, Query Parsing Level, First Elimination Level, Regression Level, Second Elimination Level and Rule Construction Level. User Level takes the original query of the user and the answer set of the query. Query Parsing Level dismembers the original query as the requested attribute(s) of the query, the resource(s) of the query and the condition(s) of the query to specify the dependent and independent attributes using the answer set. First Elimination Level takes the dependent and independent attributes with their data in the answer set and executes the backward elimination algorithm which is given in Subsection 3.2. By the use of the algorithm, our first elimination is done for independent attributes with the higher VIF factor (this factor is described in Subsection 3.1 briefly). Regression

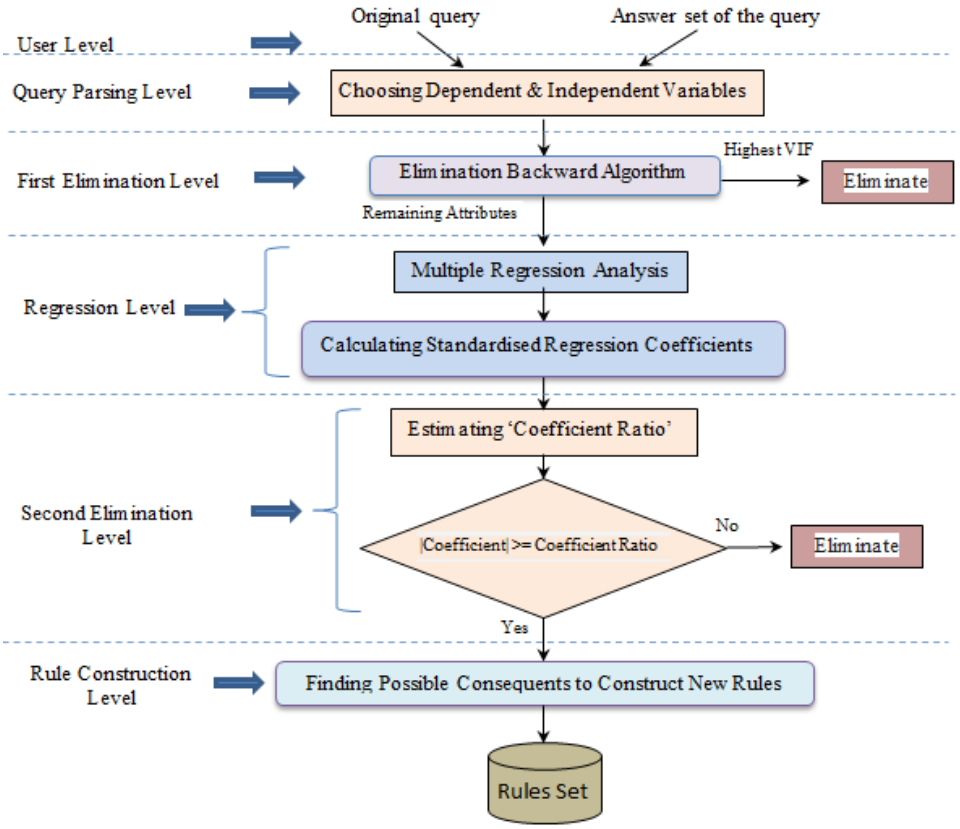


FIGURE 2. Statistical Query-based Rule Derivation System by Backward Elimination Algorithm

Level takes the remaining attributes to calculate their standardised regression coefficients. Second Elimination Level uses these coefficients to compute Coefficient Ratio and then independent attributes are eliminated if they have lower coefficient value than the ratio. The last but least important level is to construct new rules based on the remaining independent attributes which will be the attribute in the consequent of the rule. This level is detailed in subsection 3.3. The system is built using Visual Basic programming language based on SQL-Server database. The data analysis is done dynamically by using the backward elimination to analyse records. Our software system based on the automatic elimination also works with the up-to-date information in any given database which can change at any time without any need of reconstructing the complete system.

3.1. **Query Parsing Level.** This level dismembers the original query which is given in SQL as the requested attribute(s) of the query, the resource(s) of the query and the condition(s) of the query. Then attribute of each condition is chosen as a dependent attribute to be a candidate of the attribute of antecedent of a new rule and the attributes of the result set of the query is chosen as independent attributes in order to check whether any of these independent attributes can be the attribute of the consequent of the new rule after executing all levels.

3.2. **First Elimination Level.** In statistics, the simple way to find the multi-collinearity among dependent attributes is to estimate the Variance Inflation Factors (VIF) [22]. The following equation given below can be used to calculate the VIF and the higher the value, the higher the multi-collinearity. A VIF for a single dependent attribute is obtained using the R-squared value of the regression of that variable against all other dependent attributes:

$$VIF_n = \frac{1}{1 - R_n^2} \tag{1}$$

where VIF for attribute n is the reciprocal of the inverse of R_n^2 from the regression. The VIF is calculated for each dependent attribute and those with high values are removed. The definition of high is somewhat arbitrary but values in the range of 5-10 are commonly used. We opted to use 10 as a threshold in our experiments. However our high is 5 and several others, the results were not changed. This backward elimination algorithm based on VIF factor is given in Figure 3 which can be described in five steps.

Step 1: Take values of the dependent (Y) and independent attributes ($X_1, X_2, \ldots, X_n$) from the database

Step 2: Consider the multi-regression model as in $\hat{Y} = A_0 + A_1X_1 + A_2X_2 + \ldots + A_nX_n$

Step 3: Calculate variance inflation factors (VIF) of each X attribute

Step 4: If there is more than one attribute's VIF is greater than 10, then the attribute with the maximum VIF is excluded from the model and go to Step 3 with the remaining attributes. However if there is only one attribute's VIF is greater than 10, this attribute is excluded from the model and go to step 3.

Step 5: When the model does not have any attribute with the VIF which is greater than 10, then the remaining attributes are used in the final regression model in order to calculate the coefficients of the remaining attributes using the least-squared method.

FIGURE 3. Backward Elimination Algorithm

3.3. **Regression level.** The multiple regression analysis examines values of the attributes to see whether or not any linear function between the dependent and independent attributes can be constructed. In Equation 2, Y is an estimated-dependent attribute;

$$X_1, X_2, X_3, \ldots, X_n$$

are independent attributes;

$$b_0,\ b_1, b_2, \ldots, b_n$$

are coefficient values, which reflect the dependencys influences on the dependent attribute.

$$Y = b_0 + b_1{*}X_1 + b_2{*}X_2 + b_3{*}X_3 + \ldots + b_n{*}X_n$$

(2) Multiple regression analysis represents a logical extension of two-attribute regression analysis. Instead of a single independent attribute, two or more independent attributes are used to estimate the values of a dependent attribute. However, the fundamental concepts in the analysis remain the same. Using the least squares method, the regression equation is derived. The method uses the linear multiple regression equation given in Equation 2. It gets difficult to estimate the attributes when the attributes and functions are increasing. For the general equation, the matrices are used for estimated-attributes. In general, assuming dependent and independent attribute values X and Y, and b unknown matrices are given above:

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad b = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_k \end{bmatrix} \tag{2}$$

Then the b matrix can be calculated as follows:

$$b = (X^T \times X)^{-1} \times (X^T \times Y) \tag{3}$$

The elements of the b matrix are the coefficients of the multiple regressions. After calculating these elements, the linear function can be constructed. In order to make the elimination, the coefficients have to be standardised. Therefore coefficients of multiple determinations are given in Subsection 3.3.1 and then the standardised regression coefficients are described in Subsection 3.3.2 in detail.

3.3.1. *Coefficients of multiple determinations.* A measure of association between a dependent attribute and several independent attributes is the multiple correlation coefficient R, where $0 <= R <= 1$. The multiple coefficient of determination can be calculated from:

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_E}{SS_T} \tag{4}$$

where

$$SS_T = \sum_{i=1}^{n} (y_i - \bar{y})^2, \quad SS_R = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2, \text{ and } SS_E = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

and $SS_T = SS_R + SS_E$. In Equation 4, y_i shows the actual values of the dependent attribute, $\bar{y}$ shows the mean of the actual values of the dependent attributes, $\bar{y}_i$ shows the predicted values of the dependent attribute and n shows the total number of the records.

3.3.2. *Standardised regression coefficients.* Standardised regression coefficients determine the relation between the dependent and independent attributes. Regression coefficients cannot provide an idea about the effect of the independent attribute on the dependent attribute. If we transform the regression coefficients to the standardised regression coefficients, the effect of the independent attribute on the dependent attribute can become clear. When the model is generated for the system which has more than two attributes, Equation 5 is used:

$$S_{kj} = \sum_{i=1}^{n} (x_{ik} - \bar{x_k})(x_{ij} - \bar{x_j}) \tag{5}$$

and $r_{kj} = \frac{S_{kj}}{(S_{kk}S_{jj})^{(1/2)}}$, the correlation matrix is

$$R = \begin{bmatrix} 1 & r_{12} & \dots & r_{1k} \\ r_{12} & 1 & \dots & r_{2k} \\ \vdots & \vdots & \dots & \vdots \\ r_{1k} & r_{2k} & \dots & 1 \end{bmatrix} \tag{6}$$

The matrix which is originated by the correlations between the dependent attribute and independent attribute (k unit) is:

$$g = \begin{bmatrix} r_{1y} \\ r_{2y} \\ \vdots \\ r_{ky} \end{bmatrix} \tag{7}$$

With the help of two matrices, Equations 6 and 7, the standardised regression coefficients are calculated for the k unit independent attribute using Equation 8:

$$\alpha = R^{-1}g \tag{8}$$

After obtaining the standardised regression coefficients, the coefficient ratio can be estimated.

3.4. **Second Elimination Level.** By the use of standardised regression coefficients, the coefficient ratio of independent attributes can be calculated by averaging the standardised coefficients arithmetically as formulated below:

$$Coefficient_Ratio = \frac{(|\alpha_1| + |\alpha_2| + \ldots + |\alpha_n|)}{n} \tag{9}$$

Then if the coefficients of the independent attributes are greater than or equal to the ratio, they are used for new rule construction. Otherwise, independent attributes are eliminated.

3.5. **Rule Construction Level.** After the second elimination level, the system constructs the consequents for un-eliminated independent variables using their minimum and maximum values in the result set of the query. Finally, the consequents are found, and new rules are saved in the rule set.

4. **Computational results.** During the experiments, EPISODE was used as the data set, with 41,046 records. It was created from The Department of Health (England), Hospital Episode Statistics in 1998/2003. Attributes and their data types in EPISODE are given in Table 2. The main reason for using this database was to prevent difficulty in getting any suitable database for the confidentiality of data security. A sample set with 10 records from EPISODE is given in Table 3. In

TABLE 2. 2 Attributes of EPISODE

EPISODE
Attributes:
id#, startyear, finishyear, epcodes, episodes, admissions, male, emergency, waiting_list, mean_waiting, median_waiting, mean_length, median_length, mean_age, age_0_14, age_15_59, age_60_74, age_75, day_case, bed_days
id# is the primary key

Subsection 4.1, an example is used to explain how the system works for a new query.

TABLE 3. Sample Set with 10 Records from EPISODE

ID	*EPISODES*	*ADMISSIONS*	*MALE*	*EMERGENCY*	*WAITING LIST*	*MEAN WAITIG*	*MEDIAN WAITING*	*MEAN LENGTH*	*MEDIAN LENGTH*	*MEAN AGE*	*AGE 0 14*	*AGE 15 59*	*AGE 60 74*	*AGE 75*	*DAY CASE*	*BED DAYS*
1	1403	1187	635	1361	6	5	1	6.7	4	36	414	570	202	217	2	8100.2
2	87	64	42	78	4	11	11	14.3	10	49	11	32	23	20	1	945.7
3	23	18	14	18	3	47	47	12.1	8	32	10	10	2	1	0	226
4	22	20	16	20	2	77.5	77	5.7	4	23	8	11	1	2	0	112.8
5	275	225	125	266	4	11.7	13	7.1	5	41	49	131	39	56	1	1674.3
6	128	114	50	116	2	13	13	6.6	4	26	62	42	14	10	0	706.5
7	2276	1984	1122	1975	262	53	35	5.4	4	43	334	1138	422	381	248	9418.9
8	18	13	8	15	3	18.3	5	13.6	10	42	0	9	0	9	1	179.5
9	1956	1321	662	1793	35	16.8	7	21.8	14	74	13	230	491	1222	6	29697
10	2686	2553	1261	388	2152	51.2	36	10.3	5	53	92	1479	771	344	2236	3242.2

Computational results are given for rule derivation and query optimisation in order to compare the number of derived rules and their effects on query optimisation, by the use of our proposed system to Traditional Rule Derivation by Siegels System. In Subsection 4.2, the rule derivation results are pointed out for the number of derived rules and the derivation times of these rules. In Subsection 4.3, execution times of the optimisation using the rules are graphically illustrated.

4.1. **Example.** A query as SELECT * FROM EPISODE WHERE Episodes=1384 is executed on EPISODE; the rules derived by Siegels system are shown in Table 4. For the condition of the query, the dependent attribute of Episodes, the remaining attributes after running our system are ID, AGE_0_14, AGE_15_59, AGE_60_74 and AGE_75. According to these remaining attributes and the condition of the given query, the derived rules can be seen in Table 5. It can be seen from Table 4 and 5 that our system can limit the number of the rule set from 32 to 10 rules, which is 68.75% decreasing for the query of this example. Further experiments were made and results are obtained to have an overall look on the rule derivation and its outcomes.

4.2. **Rule derivation. Experiment 1**. During the derivation, 55635 different queries are executed. According to the rule derivation method by Siegel s system and our system, two comparisons can be seen in Figure 4 for the number of rules and in Figure 5 for the learning time. After the executions of these queries, in total, the number of rules by Siegels system was 1233224 and our system has derived 365634 which is 70.35% decreasing on average. Similar conclusion can be made for learning time as 71.58%. Therefore deriving all possible rules takes time and the question is Are they all useful. The experiments in Subsection 4.3 are done for this purpose. During our experiments, it is also observed that the rule derivation process is depended on the number of the different values of the attribute contained in the query and the number of the records satisfied by each value. For example, bed_days has the maximum different values and also the number of records satisfied

TABLE 4. Rules Derived by Siegels System

ID	*Ant Att*	*Ant Oper*	*Ant Value*	*Ant Index*	*Ant Key*	*AntRrecord*	*Cons Att*	*Cons Oper*	*Cons Value*	*Cons Index*	*Cons Key*	*Cons Record*	*LT*
1	EPISODES	=	1384	0	0	4	ID	<=	27107	0	1	21109	118
2	EPISODES	=	1384	0	0	4	ID	>=	9027	0	1	26894	135
3	EPISODES	=	1384	0	0	4	ADMISSIONS	<=	1369	0	0	28558	48
4	EPISODES	=	1384	0	0	4	ADMISSIONS	>=	1226	0	0	5718	66
5	EPISODES	=	1384	0	0	4	MALE	<=	863	0	0	29710	38
6	EPISODES	=	1384	0	0	4	MALE	>=	52	0	0	16630	66
7	EPISODES	=	1384	0	0	4	EMERGENCY	<=	1077	0	0	31235	32
8	EPISODES	=	1384	0	0	4	EMERGENCY	>=	49	0	0	13710	50
9	EPISODES	=	1384	0	0	4	WAITING_LIST	<=	734	0	0	30085	30
10	EPISODES	=	1384	0	0	4	WAITING_LIST	>=	115	0	0	10420	49
11	EPISODES	=	1384	0	0	4	MEAN_WAITING	<=	40	0	0	16371	36
12	EPISODES	=	1384	0	0	4	MEAN_WAITING	>=	14	0	0	26411	56
13	EPISODES	=	1384	0	0	4	MEDIAN_WAITING	<=	21	0	0	16164	30
14	EPISODES	=	1384	0	0	4	MEDIAN_WAITING	>=	6	0	0	27422	50
15	EPISODES	=	1384	0	0	4	MEAN_LENGTH	<=	40551	0	0	22871	31
16	EPISODES	=	1384	0	0	4	MEAN_LENGTH	>=	40758	0	0	20180	49
17	EPISODES	=	1384	0	0	4	MEDIAN_LENGTH	<=	4	0	0	23421	30
18	EPISODES	=	1384	0	0	4	MEDIAN_LENGTH	>=	2	0	0	23630	49
19	EPISODES	=	1384	0	0	4	MEAN_AGE	<=	59	0	0	27079	30
20	EPISODES	=	1384	0	0	4	MEAN_AGE	>=	9	0	0	31159	50
21	EPISODES	=	1384	0	0	4	AGE_0_14	<=	1157	0	0	33024	31
22	EPISODES	=	1384	0	0	4	AGE_0_14	>=	0	0	0	33873	50
23	EPISODES	=	1384	0	0	4	AGE_15_59	<=	941	0	0	29624	32
24	EPISODES	=	1384	0	0	4	AGE_15_59	>=	150	0	0	11600	49
25	EPISODES	=	1384	0	0	4	AGE_60_74	<=	481	1	0	30576	31
26	EPISODES	=	1384	0	0	4	AGE_60_74	>=	61	1	0	10041	53
27	EPISODES	=	1384	0	0	4	AGE_75	<=	175	0	0	28816	31
28	EPISODES	=	1384	0	0	4	AGE_75	>=	15	0	0	13975	49
29	EPISODES	=	1384	0	0	4	DAY_CASE	<=	872	0	0	31210	32
30	EPISODES	=	1384	0	0	4	DAY_CASE	>=	74	0	0	9498	49
31	EPISODES	=	1384	0	0	4	BED_DAYS	<=	5248.6	0	0	28832	37
32	EPISODES	=	1384	0	0	4	BED_DAYS	>=	2968.5	0	0	7155	58

TABLE 5. Rules Derived by Our System

ID	*Ant Att*	*Ant Op*	*Ant Value*	*Ant Index*	*Ant Key*	*Ant Rn*	*Cons Att*	*Cons Op*	*Cons Value*	*Cons Index*	*Cons Key*	*Cons Rn*	*Rlt*
1	EPISODES	=	1384	0	0	4	ID	<=	27107	0	1	21109	118
2	EPISODES	=	1384	0	0	4	ID	>=	9027	0	1	26894	135
21	EPISODES	=	1384	0	0	4	AGE_0_14	<=	1157	0	0	33024	31
22	EPISODES	=	1384	0	0	4	AGE_0_14	>=	0	0	0	33873	50
23	EPISODES	=	1384	0	0	4	AGE_15_59	<=	941	0	0	29624	32
24	EPISODES	=	1384	0	0	4	AGE_15_59	>=	150	0	0	11600	49
25	EPISODES	=	1384	0	0	4	AGE_60_74	<=	481	1	0	30576	31
26	EPISODES	=	1384	0	0	4	AGE_60_74	>=	61	1	0	10041	53
27	EPISODES	=	1384	0	0	4	AGE_75	<=	175	0	0	28816	31
28	EPISODES	=	1384	0	0	4	AGE_75	>=	15	0	0	13975	49

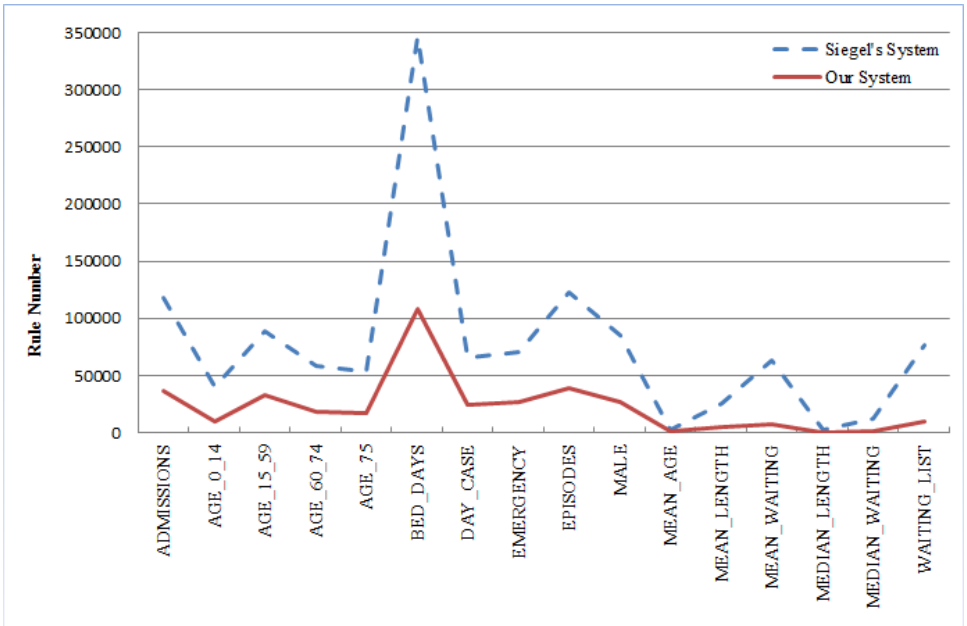


FIGURE 4. Comparison of Rule Numbers Between Siegel's System and Our System

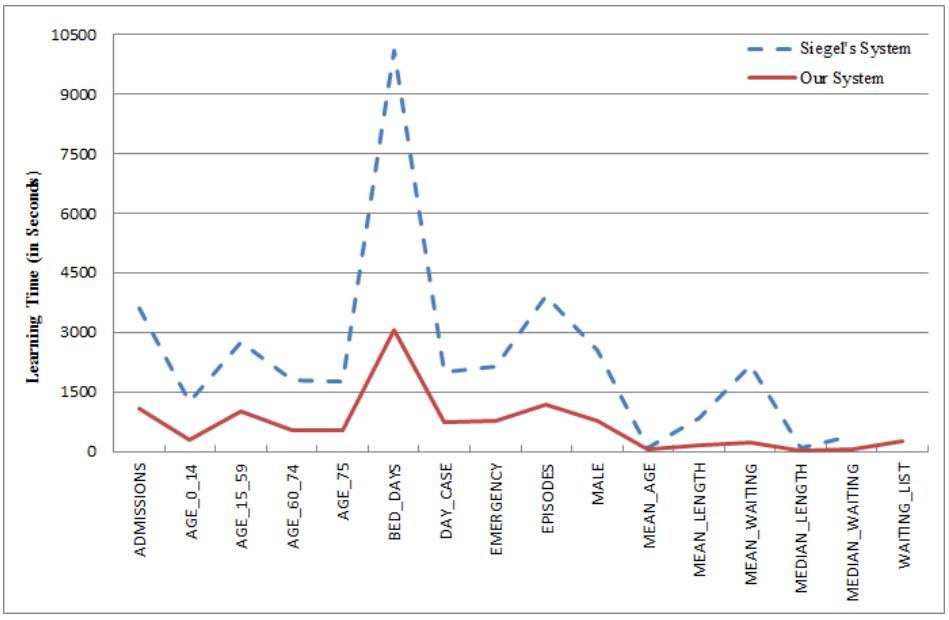


FIGURE 5. Comparison of Rule Learning Time Between Siegel's System and Our System

by each value, therefore the number of rules and learning time also are highest for the queries containing this attribute.

Experiment 2. This experiment is done to find out other effects of the learning process such as the content of the query which can be given in four different cases described below:
Case I. Consequent attribute is the primary key and its condition contains equality comparison
Case II. Consequent attribute is the primary key and its condition contains range comparison
Case III. Consequent attribute is no primary key and its condition contains equality comparison
Case IV. Consequent attribute is no primary key and its condition contains range comparison

For these cases, following results in Table 6 are estimated. It can be seen that

TABLE 6. Learning Cases

	Number of Rules Learned By Our System	**Total Learning Time Of Our System (in seconds)**
Case I	33,999	600.910
Case II	43,272	1,179.228
Case III	513,097	12,473.109
Case IV	642,856	23,495.301

Case I takes the minimum learning time and Case IV takes the maximum learning times. Therefore the learning process is also affected by the content of the query.

4.3. **Query optimisation. Experiment 3**. 878 different queries are randomly chosen to calculate their total execution time in three different execution modes in order to find out the significances of the rules on query execution. They are No rule used during query optimisation, 1233224 rules by Siegels system are used, and 365634 rules by our system are used, which are shown in the following Table 7. Our system is 69.28% faster than the original query execution and 57.00% faster

TABLE 7. Total Time for query optimisation (in millisecond)

Original Query Execution (No rule)	**Optimum Query Execution By Siegels System (1233224 learned rules)**	**Optimum Query Execution By Our System (365634 learned rules)**
160.956	115.016	49.446

than the optimum query execution by Siegels system. These results show that the original query execution can be supported by rules in order to have a better system. Moreover it is possible to say that the usefulness of the rules is different because the number of rules did not reduce the query execution time, contrarily increased it during the long optimisation process. Our learning system has derived 857590 less rules and gave the better query optimisation.

Experiment 4. For this experiment, 50 different queries are randomly chosen to find out the computational results of the query optimisation in order to give more detail graphically for each individual query execution, with and without the use of

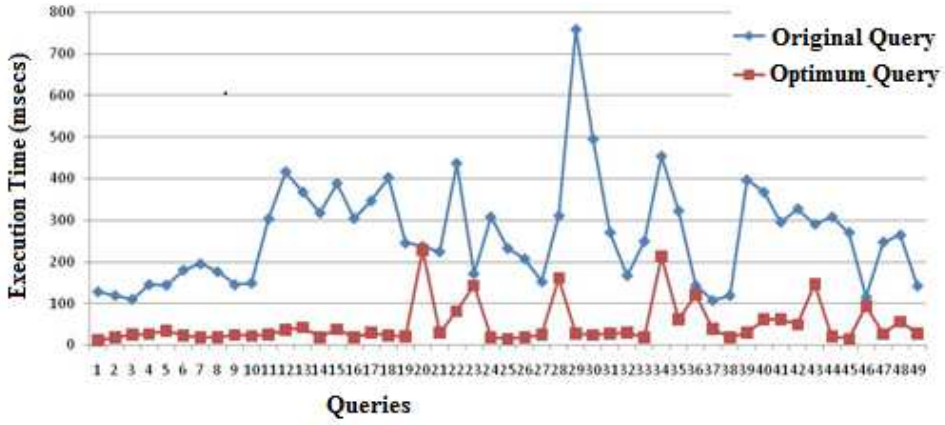


FIGURE 6. Comparison of Query Execution Time

rules. The results are shown in Figure 6. From Figure 6, the most of the optimum queries by our system could be executed in less time than original queries. Query 20, Query 36 and Query 46 gave the similar results because the contents of these queries were remained to be the same on the query optimisation.

Experiment 5. This experiment is designed to find out other effects of the optimisation process related to the content of the query (described as four cases in Experiment 2) in order to point out the optimisation results of these cases. In the experiment, the number of queries between 200 and 300 for each case are executed and the average execution times for each case are given in Table 8 below. The

TABLE 8. Learning Cases

	Optimum Query by Siegel's System in miliseconds	**Optimum Query by Our System in miliseconds**
Case I	100.14	36.07
Case II	234.95	124.52
Case III	347.79	269.54
Case IV	364.69	278.06

results of this experiment in Table 8 are very similar to Experiment 2; Case I takes the minimum query execution time and Case IV takes the maximum. Therefore the query execution is also related to the contents of queries and their matched rules on optimisation. It is very important to derive the most useful rules instead of learning all.

5. **Conclusion.** From the experiments, main conclusions can be listed as follows:

- Learning all possible rules does not give better query optimisation.
- Deriving rules based on data analysis can help us to select useful rules.
- Query optimisation depends on the content of the original query and the content of the rule which is used for this query.
- The last but the most importantly the use of rules can make the query execution faster than the original query execution.

Although several researchers are working on data analysis for rule derivation, there are still existing problems on the components of SQO such as new learning methods, rule evaluation, rule maintenance, query optimisation. For example, we next wish to extend our system implementation using Forward Selection and /or Bidirectional Elimination and do further experiments. Finally we hope that this paper can be the inspiring point of new future works for the rule derivation and query optimisation.

REFERENCES

[1] J. Bakus and M. S. Kamel, Higher order feature selection for text classification, *Knowledge and Information Systems*, **9** (2006), 468–491.

[2] S. Ceri, P. Fraternali, S. Paraboschi and L. Tanca, Automatic generation of production rules for integrity maintenance, *ACM Transactions on Database Systems*, **19** (1994), 367–422.

[3] U. S. Chakravarthy, J. Grant and J. Minker, Logic-based approach to semantic query optimization, *ACM Transactions on Database Systems)*, **15** (1990), 162–207.

[4] K. C. Chan and A. K. Wong, A statistical technique for extracting classificatory knowledge from databases, *Discovery in Databases*, (1991), 107–124.

[5] F. De Marchi, S. Lopes and J.-M. Petit, Efficient algorithms for mining inclusion dependencies, in *Advances in Database Technology-EDBT 2002*, 2287, Springer, 2002, 464–476.

[6] D. Genest and M. Chein, A content-search information retrieval process based on conceptual graphs, *Knowledge and Information Systems*, **8** (2005), 292–309.

[7] G. Graefe and D. J. DeWitt, *The EXODUS Optimizer Generator*, vol. 16, ACM, 1987.

[8] M. M. Hammer and D. J. McLeod, Semantic integrity in a relational data base system, in *Proceedings of the 1st International Conference on Very Large Data Bases*, 1975, 25–47.

[9] C.-N. Hsu and C. A. Knoblock, *Learning Database Abstractions for Query Reformulation*, University of Southern California, Information Sciences Institute, 1993.

[10] I. K. Ibrahim, V. Dignum, W. Winiwarter and E. Weippl, Logic based approach to semantic query transformation for knowledge management applications, *Proceeding of the International Conference on Knowledge Management*, Berlin, Springer, 2002.

[11] I. Imam, R. Michalski and L. Kerschberg, Discovering attribute dependence in databases by integrating symbolic learning and statistical analysis techniques, in *Proceeding of the AAAI-93 Workshop on Knowledge Discovery in Databases, Washington DC*, 1993.

[12] N. S. Ishakbeyoğlu and Z. M. Özsoyoğlu, On the maintenance of implication integrity constraints, in *Database and Expert Systems Applications*, Springer, 1993, 221–232.

[13] J. J. King, Quist: A system for semantic query optimization in relational databases, in *Proceedings of the seventh international conference on Very Large Data Bases-Volume 7*, VLDB Endowment, 1981, 510–517.

[14] S.-G. Lee, L. J. Henschen, J. Chun and T. Lee, Identifying relevant constraints for semantic query optimization, *Information and Software Technology*, **42** (2000), 899–914.

[15] G. Piatetsky-Shapiro and C. Matheus, Measuring data dependencies in large databases, in *Knowledge Discovery in Databases Workshop*, 1993, 162–173.

[16] V. Pudi and J. R. Haritsa, Generalized closed itemsets for association rule mining, in *Data Engineering, 2003. Proceedings. 19th International Conference on*, IEEE, 2003, 714–716.

[17] A. Sayli and A. Elibol, A rule rectification algorithm on semantic query optimisation, in *17th International Conference on Systems Research, Informatics and Cybernetics Focus on Advances in Decision Technology and Intelligent Information Systems*, **6** (2005), 12–16.

[18] A. Sayli and A. Elibol, Rule evaluation algorithm for semantic query optimisation, *Applied Mathematics and Information Sciences*, **7** (2013), 1773–1781.

[19] A. Sayli and B. Gokce, The use of SQO rules in DMLs, in *Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium on*, vol. 1, IEEE, 2004, 146–151.

[20] A. Sayli and B. Lowden, A fast transformation method to semantic query optimisation, in *International Database Engineering and Applications Symposium*, IEEE, 1997, 319–326.

[21] A. Sayli and B. Lowden, Ensuring rule consistency in the presence of db updates, in *Proc. XII International Symposium on Computer and Information Sciences, Turkey*, 1997.

[22] A. Sayli and O. Uysal, A dynamic self-learning method for semantic query optimisation, *International Journal of Technology, Policy and Management*, **8** (2008), 126–147.

[23] S. Shekhar, B. Hamidzadeh, A. Kohli and M. Coyle, Learning transformation rules for semantic query optimization: A data-driven approach, *IEEE Transactions on Knowledge & Data Engineering*, **5** (2002), 950–964.
[24] M. Siegel, E. Sciore and S. Salveter, A method for automatic rule derivation to support semantic query optimization, *ACM Transactions on Database Systems*, **17** (1992), 563–600.
[25] C. T. Yu and W. Sun, Automatic knowledge acquisition and maintenance for semantic query optimization, *IEEE Transactions on Knowledge and Data Engineering*, **1** (1989), 362–375.

Received July 2015; revised September 2015.

E-mail address: sayli@yildiz.edu.tr
E-mail address: oncuuysal@yahoo.com