Design and Implementation of Surface Irrigation Soil Loss (SISL) and Soil Condition Index (SCI), (SISL-SCI) Database

by

Alexander K. Nkansah

A Project Report

submitted to

Dept. of Computer Science
Oregon State University
Corvallis, OR, USA

in partial fulfillment of the requirements for the degree of

Master of Computer Science

Completed April 12, 2001

Commencement June 2001

An Abstract of the Report of

Alexander K. Nkansah for the degree of Masters in Computer Science presented on April 12, 2001.

Title: Design and Implementation of Surface Irrigation Soil Loss (SISL) and Soil Condition Index (SCI), (SISL-SCI) Database

Abstract approved: ______________________________

(Toshimi Minoura)

The *Soil Conditioning Index* (SCI) aids the conservationist in designing crop rotations and residue management practices when low organic matter, poor soil tilt, and crusting problems are identified. The Natural Resources and Conservation Services (NRCS) field staff and others use the SCI as a qualitative tool during conservation planning. In this project we implemented a *relational database* for the SCI application and integrated it with the *Soil Loss by Surface Irrigation* (SISL) application by creating one normalized relational database schema. These two agricultural applications possess some commonalities that make them fit well together in one database system. Calculations for all the three subfactors of SCI are performed via stored procedures. These are called from within the SCI triggers for automatic execution. We expect that SISL-SCI database will become a basis for integrating several other agricultural applications, e.g., *Revised Universal Soil Loss Equation (*RUSLE) and *Wind Erosion Estimates model* (RWEQ*).*

Acknowledgements

One person who deserves tonnes of appreciation is Professor Toshimi Minoura of the Computer Science department. As a major advising professor, he has been a real mentor throughout the cause of this work. I am also grateful to Professors Bella Bose and Paul Cull not only for serving on my committee but also for their contribution to the outcome of this final report.

I would like to express my genuine gratitude to all my numerous friends and office mates whose support has been tremendous through the period of this project. I also want to extend my gratitude to the woman who has been the brain behind my success, my mother, Janet Ohene Darko. To my dear wife Beatrice Duah, I say thanks for your understanding and standing by me in the difficult moments.

The last but not he least I am indebted to the Almighty God without whose mighty provisions this whole mission will not have seen the light of day. To him alone be all glory, honor and power forever more.

Table of Contents

Surface Irrigation Soil Loss Soil (SISL) and Soil Conditioning Index (SCI)

Section 1

Introduction

Irrigation Induced erosion caused by irrigation water running over the soil surface has long been recognized as a serious problem on surface irrigated croplands. The studies conducted at the Agricultural Research Services (ARS) over a ten-year period revealed a general trend. Analysis of the data obtained from the research led to the discovery of the prediction model known as the "Surface Irrigation Soil Loss (SISL) model".
This Model can be stated as:

$$\mathbf{SISL = BSL * KA * PC * CP}$$

SISL = surface irrigation soil loss from a field in tons per acre

BSL = base soil loss rate average from ARS soil loss measurements

KA = soil erodibility adjustment for the soil in relation to the soil on which base erosion data is obtained

PC = prior crop impacts on reducing soil erosion

CP = conservation practice impacts on reducing soil erosion.

In this report we will only demonstrate how the SISL can be integrated with the SCI without giving too much details of the SISL model.

The level of organic matter in the soil is a primary indicator of the condition of the soil because it affects such soil characteristics as the cation exchange rate, aggregate stability, water holding capacity, and the level of biological activities. The *Soil Conditioning Index (*SCI*)* is a tool that is used to predict the consequences of a cropping system and a tillage practice on the status of soil organic matter. The SCI system is widely used by Natural Resources and Conservation Services (NRCS) field staff and other conservationists.

Many factors cause degradation of the soil condition. Wind and water erosion removes fine soil particles, organic matter, and nutrients, reducing the ability of the soil to hold water. Excessive tillage accelerates erosion and organic matter decay, and causes compaction of soil. A crop rotation which produces a low amount of residue or which involves extensive residue removal causes an inadequate amount of organic material returned to the soil. The SCI thus indicates the effects of the cropping sequence, soil-disturbing operations, and other management practices on soil organic matter trends. The SCI is designed to aid the conservationist in designing crop rotations and residue management, especially when problems of low organic matter, poor soil tilt, and crusting problems are identified during planning.

The major factors affecting the soil conditioning of a given field can be grouped into three main components: *Organic Material (*OM*), Field Operations (*FO*)* and soil *ERosion (*ER*)*. The actual percentage composition of these three factors depends on the field and its location. Negative SCI means soil organic matter is predicted to be decreasing, and corrective measures should be planned. A large positive SCI indicates an improvement in soil organic matter trends. SCI Values close to zero indicates stability in the soil condition.

Currently, the only means for computing the SCI value for a crop rotation is through the use of a spreadsheet worksheet. The data tables used in the computation of the SCI value are not normalized. This means the data is susceptible to duplications and inconsistencies. For example, the current tables show the city code (19001) for two different cities. There are no proper indexes for search. It is also tedious to compute the SCI value for a field. The problem become exacerbated when the SCI value for multiple fields are required.

The logical design of a database, including the tables and the relationships between them, is the core of an optimized relational database. A good logical database design can lay the foundation for optimal database and application performance. A poor logical database design can impair the performance of the entire system. In this project, we use relational database methodology to provide an efficient SCI system. Our goal was to *normalize* the SISL and SCI data representations, thereby eliminating data redundancies, inconsistencies, and also maintain the integrity of SISL-SCI data. We used stored procedures and triggers in calculating the SCI value and its subfactors.

Section 2

The *Soil Conditioning Index System*

This section discusses the components of the SCI model and introduces the generalized form of the SCI equation. We focus on the current SCI system and a description of how the *SCI* value is computed. The section closes with an illustration of a typical SCI problem. We walk the reader through the solution and then further evaluate an alternative system in comparison to the stated problem. We demonstrate the calculations using the existing tables and as is in the excel application.

2.1 Soil Conditioning Index

The SCI reflects the combined effect of three components on soil organic matter trends. These are the organic material subfactor, field operations subfactor, and erosion subfactor. The generalized form of the *SCI* Model is:

$$SCI = f\,(OM, FO, ER).$$

Organic Material (OM): This component accounts for the effect of organic material returned to the soil. Organic material from plant sources may be either

(a) grown and retained on the site, or

(b) imported to the site

This subfactor does not include compost or liquid manure.

Field Operations (FO): This component accounts for the effect of field operations that stimulate organic matter breakdown. Tillage, planting, fertilizer application, spraying and harvesting more often than not crush and shatter plant residues and aerate or compact the soil. These effects increase the rate of residue decomposition and affect the placement of organic material in the soil profile.

Erosion (ER): This component accounts for the effect of removal of surface soil material by the sheet, rill, and/or wind erosion processes which are predicted by RUSLE and WEQ. Erosion contributes to loss of organic matter and decline in long-term productivity. It does not account for the effect of concentrated flow erosion such as ephemeral or classic gullies.

2.2 Present System

Presently, a computerized worksheet has been developed which operates in Microsoft Excel 5.0 or higher. Using the Excel worksheet and the list boxes built in, the conservationist can quickly and easily enter any information describing the scenario in the appropriate cells of the worksheet. The worksheet uses the variable lookup function in Excel to retrieve text and data in order to minimize operator data entry and to speed up calculations. The tables are stored in stored in other worksheets in the same spreadsheet. The worksheet is shown is Figure 1a and Figure 1b. [9]

Soil Conditioning Index Worksheet

Version 13 JULY 18, 2000

Producer:		Location:	Alabama Reservation	Staff:	
Tract:		Field:			

A. Site Information

Location code:	1001	City:	BIRMINGHAM, AL	Rainfall Factor R:	150 (RUSLE)
(From City Tab)					
Soil:		Soil Erodibility Factor K (adjusted Kf) :	0.27 (RUSLE)		
		Slope, Length & Percent (LS factor) :	1.05 (RUSLE)		
Maintenance Amount:		"Residue Equivalent Values" (REV) :	5943 lbs./ac		

CR GP	Rev Conv Fact
A	1.19
B	1.01
C	1.00
D	0.98
E	0.98
F	0.97
G	0.97
H	0.96

B. Management Information

Number of Yrs in Rotation:	2	Crop Rotation:	Corn and Soyabean
Tillage System:		Crop Mgt Factor C:	0.188 (RUSLE)
Support Conservation Practices:		Support Practice Factor P:	1 (RUSLE)

C. Organic Material (OM)

(From Crop Tab)

Crop #	Crop	Yield	Harv Unit/Ac	Wt Harv Unit (lbs)	Res: Yield Ratio	Res Prod	Root Mass Adjust	Biomass Prod	Biomass Removed or Added	Total Biomass	Crop Group	REV Conv	REV lbs/ac
139	corn; 75bu 30" 90 MD	75	bushels	56	1	4200	1.151	4835		4835	C	1.00	4835
607	soybean; 19" 35bu so	35	bushels	60	1.5	3150	1.114	3510		3510	F	0.97	3402

TOTAL REV:	8237
NO. YEARS IN ROT.:	2
AVE. ANNUAL (RP):	4118
MAINT. AMNT. (MA):	5943
SUBFACTOR (OM):	**-0.31**

Figure 1a. Soil Conditioning Index Worksheet

D. Field Operations (FO)

(From Operation Tab)

Soil Disturbance Rating (SDR)

Op #	Operation Name	# of Trips	SDR	Op #	Operation Name	# of Trips	SDR
2	Anhydous applicator w/ Knife and w/coulter	1	8			1	
13	Chisel; straight points (18" spacing)	1	19			1	
22	Combo Up-rooter/bedder (cotton)	1	24			1	
64	Fertilizer Inject w/ spike wheel injector	1	4			1	
128	Skew treader, backward	1	8			1	
103	Planter/Drill 10-20" Fluted coulter >2"wdth	1	13			1	
96	Planter > 20" Runner shoe	1	1			1	
142	Weed with Machete (PB)	1	2			1	
81	Harvest, w/o soil disturbance	1	5			1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	
		1				1	

Total SDR for Rotation: 84 **Average Annual SDR:** 42.0

Field Operations Subfactor (FO), Table 3: 0.58

E. Erosion (ER)

	Sheet & Rill:	RUSLE Calc.:	Wind:	Irrigation:
Predicted Erosion:		8.0		

Total Avg Annual Erosion: 8.0 ton/ac

Erosion Subfactor (ER) Table 4: -0.75

F. Soil Conditioning Index

SCI = OM*0.4 + FO*0.40 + ER*0.20

Soil Conditioning Index = **-0.04**

G. Notes:

OM factor negative - Change crop rotation, add cover crops, mulch or manure or remove less residue.
FO factor is positive - disturbance is less than steady state conditions.
The erosion rate is excessive, suggest erosion control conservation practices.
The overall soil condtion index is negative, modify system to improve subfactors.

Figure 1b. Soil Conditioning Index Worksheet (cont'd)

2.2.1 Running SCI Excel Worksheet

Once selections are made from the choice lists, the computer looks up the data, copies it in the worksheet, makes the calculations and returns the index for the given scenario. Scenarios can be quickly modified if the producer wants to evaluate the effects of changes in crops, tillage or erosion rates. Worksheets can be saved for typical systems and quickly modified with more site-specific producer information. The user needs to have a working knowledge of Microsoft Excel but can get up to speed pretty quickly even without that skill. We walk the reader through an example. Information can only be entered in the cells shaded yellow. Those having a dark box around the cell are required for the program to run. In part A, select from list button for a climate location. Excel uses a variable lookup function to find the associated information. This will bring in the data for this location from Table 1 (see appendix B). You can either enter the RUSLE factor values or just enter the estimated soil loss. You must, however, enter the number of years in the crop rotation. Cells for describing the rotation and tillage system and producer information are not "live" and are not used by the spreadsheet for calculations but are useful to document the system being evaluated

Next go to part C, Organic Material, using the drop down lists on each line, select crops and yield levels from table 2. Be sure to account for all crops grown. If additional mulch or manure is applied or residue removed show it in column J as a (+) or (-) entry. If you entered the numbers of years in the rotation earlier, it should calculate the subfactor for OM for this cropping system.

In part D, select the operations performed for one complete cycle of the rotation. If more than one trip or pass of the same machine is performed, you may place the number of passes in the cell immediately to the right of the operation name dropdown list box. The FO subfactor should now be calculated.

In part E, if you completed all of the RUSLE factors on the first page, the soil loss is already calculated for RUSLE. If you didn't, or if other erosion is also present just directly enter soil loss for any of the various types of erosion that are present in your example. Soil loss from various sources is additive for purposes of the SCI.

The overall SCI value should now show in part F. When developing the Erosion subfactor using the RUSLE factor values in parts A and B, one can easily do this by changing the C factor. Field offices should be able to develop several typical scenarios and save them for use as "templates" to further speed the calculations and add to the functionality of the SCI. The databases contained in the model currently support nearly 600 climate locations in the US and most major crops and tillage machines.

2.2.2 The Deficiencies of the Current System

Inherent in the excel application is the *normalization* problem. The data tables in the present system are unusually large because of unnormalized data. This means high possibility of duplicate records in the underlying tables, data inconsistencies and no integrity constraints. The system user cannot run data consistency checks; hence no validation check is performed on new data. Integrity of the *SCI* data becomes suspect, making the calculated results from the system less reliable. Yet another flaw is the inability of the current system to handle a large volume of data. As the SCI data continues to grow, a threshold will be reached where the Excel application will not be the ideal software. Again, a user needs to customize the existing worksheet to meet his individual needs. This means several copies of the worksheet together with the data are always being moved around. This results in unnecessary waste of system resources, since scenarios are never constant.

Maintenance and *performance* can sometimes lead to obsoletion of some database systems. Performance enhancement means re-constructing the application from the scratch and re-organizing the structure of the tables each time the need arises. In this information age, the possibility of data integration cannot be ruled out, however, the current system does not provide possibilities for integrating with other related databases. Improving search for a record is not easily attainable since no proper index is created on the rows of the tables. There is an overhead in search time when scanning large volumes of data, thus making the current SCI user wait longer.

2.2.3 Calculating the Soil Conditioning Index

We now describe how the *SCI* value for a given location is calculated as a weighted average of all three sub-factors. On the average, Organic Materials constitutes 50% of the entire SCI value, Field Operations 20%, and Erosion 30%. These percentages may vary depending on various factors existing at the site. First we determine each of the three subfactors followed by the computation of the SCI.

Determine the Maintenance Amount of Crop Residue:

The *Maintenance Amount* (MA) of a crop residue at a selected location, expressed as Residue Equivalent pounds, is the amount of crop residue needed to maintain a constant level of organic matter in the soil. That is, when this amount of residue is added to the soil each year, the OM subfactor for SCI remains zero.
Residue Equivalent Values (REVs) converts all crop residues to common standard. The REV of any plant material is its mass expressed as the equivalent mass of crop group B residue, based on relative annual decomposition rates. Common crops in crop group B include corn, grain sorghum, and sunflower.

Although, not used in the calculation of the soil conditioning index, the *maintenance amounts* shown in Table 1 of appendix B are the amounts that would apply if there were no disturbance by field operations, and no erosion. These amounts may be useful in comparing biomass needs at a given location to the biomass produced by current and alternative cropping systems.

A) Determine the Organic Material Subfactor:

The steps to determine the OM subfactor are as follows:

1. First, determine the total amount of residue produced (TR) on the site by the crop rotation

$$TR = Predicted\ yield * Residue\ per\ unit$$

2. For each crop in the rotation adjust for any residue removed from or added *RMA,* to the site to obtain the non-normalized total residue produced on the site RP_{nonrev}.

$$RP_{nonrev} = TR + RMA$$

Convert the residue amounts for each crop to Residue Equivalent Value. First, identify the group for the crop in question. Next obtain and multiply the *Residue Conversion value* from Table 1. (appendix B) *by* RP_{nonrev} to obtain the RP_{rev}

$$RP_{rev} = RP_{nonrev} * REV$$

Note: this is done to convert residue amounts for each crop to Residue Equivalent Value (REV). REV conversion factors for seven crop groups at selected locations are given in the table below:

Crop Group	**Representative Crops**
AA	Small grain, except NW Wheat & Grain Region
A	Cotton, burley tobacco, Peanuts.
B	Corn, grain sorghum, sunflower.
C	Small grains in the NW Wheat & Range Region, canola, grasses.
D	Legumes.
E	Soybeans, sugar beets.
F	Vegetables and specialty crops.

Note: Crop group B is the basis for Residue Equivalent Values.

4. Divide the total RP_{rev} for the crop rotation by number of years in the rotation to determine average annual residue produced in REV and then calculate the organic material *OM* subfactor value:

$$OM = (RP - MA) / MA,$$

where *RP* is the average annual residue produced and
MA is the Maintenance amount, both expressed as REV

B) Determine the Field Operations Sub-factor:

The steps to determine the FO sub factor are as follows:

1. List all field operations example (tillage, planting, fertilizing, cultivating, etc.). Find the soil disturbance rating (SDR) for each operation from operations table. Disturbance rating is the value that indicates the extent to which these operations are affecting the nature of soil.
2. Add the soil disturbance rating values and divide the cumulative total by the number of years in the rotation to determine average annual soil disturbance rating.
3. Find the corresponding Field Operations (FO) subfactor value from the FO-subfactor table i.e., Table 4, see appendix B.

C) Determine the Erosion Subfactor:

The steps to determine the ER subfactor are as follows:

1. Determine the predicted average annual erosion using RUSLE and/or WEQ. Both wind and water erosion is estimated if present. For purposes of the index, add the two estimates together to determine the ER subfactor value.
2. Using T value for the soil, convert to multiples of T. This is based on Erosion rate (tons/acre/year)
3. Find the corresponding Erosion (ER) subfactor value in Table 4, see appendix B.

D) Calculate the Soil Conditioning Index (SCI):

The Soil Conditioning Index is the sum of the three subfactor values, weighted for their relative importance. The weighting factors are

Organic Material (OM) -- 50%,
Field Operation (FO) - -20% and
Erosion (ER) - -30%

The SCI model equation we stated earlier can be modified as follows

$$SCI = (OM \times 0.5) + (FO \times 0.2) + (ER \times 0.3)$$

E) *Significance of Calculated SCI*

The SCI obtained in this equation becomes relevant only when it can be interpreted and given meaning. If the calculated SCI value is negative, soil organic matter is predicted to be decreasing, and corrective measures should be planned. If the over all SCI value is negative, look at the various subfactors and see which one(s) are influencing it the most. Modify the scenario by changing crops or field operations to develop alternatives for consideration by the producer. If for example, the tillage system is changed from conventional to mulch-till or no-till, the erosion estimate should be changed accordingly. If the SCI value is zero or positive, soil organic matter is predicted to be stable or increasing. Depending on the SCI values, management may deem it necessary to look at alternative solutions that might help improve the SCI of the soil.

F) *Evaluate One or More Alternative Systems*

To formulate alternative, plan changes in the cropping-management system, which will address negative subfactor values, For example:

(i) If the Organic Material (OM) subfactor is negative, plan for additional high residue crops in the rotation, and/or limit residue removal.

(ii) If the field Operations (FO) subfactor is negative, plan changes in the tillage/planting system to reduce the number and/or severity of field operations.

(ii) If the Erosion (ER) subfactor is negative, consider supporting practices such as terracing, strip cropping, etc. as well as changes in the crop rotation or field operations.

Describe the alternative system (rotation and field operations) and follow the same procedure as "to evaluate the present Cropping-Management System" above. No hard and fast rules exist on how to control these effects. The appropriate remedy for a particular situation is the choice for the management of the field. To better illustrate the process outlined in this section, we follow this discussion with an example.

2.3 Example Problem

The Lincoln, Lancaster County NE has decided to adopt the SCI model to find the soil condition Index of a 6% x 200ft (slope), Sharpsburg Silty clay loam field.

1. Two group B crops, corn and soybeans, were planted in the same rotation. Corn was planted in the first year of the rotation followed by soybeans in the second year.
2. In order to prepare the land for a good crop yield the following operations were performed in the first year and second year respectively.

   Year 1: Anhydrous application, Soil Disturbance Rating (from tables 3, is 8; tandem disk (Primary), SDR = 26; plant corn w/double disk opener, SDR = 5; row cultivate (sweep), SDR = 19; harvest, SDR = 5).

   Year 2: Fall chisel w/straight points, SDR = 19; tandem disk, (finishing) SDR = 18; Plant soybeans w//double disk opener, SDR = 5; row cultivate (sweep), SDR = 19; Harvest, SDR = 5.
3. Given that the effect of soil erosion for the field as determined by the RUSLE factors are: R = 150, K = 0.27 (adjusted), LS = 1.05, C = 0.188, P = 1.0 (straight row)

Given this problem, our task is to calculate the SCI value for the Lincoln field. We can only achieve this goal by calculating and substituting the subfactor values into the SCI model. Finally, the SCI value obtained should be interpreted and used by management of the field in making decision.

The information required for calculating the various subfactors can be extracted from the problem statement.

Site Information:

Location: Lincoln, Lancaster County, NE

Soil: Sharpsburg silty clay loam

Slope: 6% x 200 ft.

Maintenance Amount (MA), the reference condition from table 1 (see appendix B) is 4198 lb/ac, REV

Management Information, Present System:

(A) **Organic Material (OM):**

Rotation and predicted yields: Corn 125 bu/ac, Soybeans 35 bu/ac.

Residue management: all residues returned.

Residue returned (adjusted to REV using conversion factors in table 1):

Corn -- 125 * 56 * 1.0 = 7000 lb / ac, REV

Soybeans -- 35 * 75 * 0.92 = 2415 lb / ac, REV

Total REV / 2 = 9415 / 2 = 4708 lb / ac, average annual REV produced *RP*

Organic Material (OM) Subfactor OM is

$$(RP - MA) / MA$$

$$(4708 - 4198) / 4198 = + 0.12$$

(B) **Field Operations (FO):**

Year 1: Anhydrous application, Soil Disturbance Rating (Table 3 = 8; tandem disk (primary), SDR = 26; plant corn w/double disk opener, SDR = 5; row cultivate (sweep), SDR = 19; harvest, SDR = 5.

Year 2: Fall chisel w/straight points, SDR = 19; tandem disk, (finishing) SDR = 18; Plant soybeans w//double disk opener, SDR = 5; row cultivate (sweep), SDR = 19; Harvest, SDR = 5.

Total Soil Disturbance Rating (SDR) for rotation is

8 + 26 + 5+ 19 + 5 + 19 + 18 + 5 + 19 + 5 = 129.

Average Annual SDR = 129 / 2 = 65.

Field Operations (FO) subfactor (read from table) = + 0.35

(C) **Erosion (ER):**

R = 150

K = 0.27 (adjusted)

LS = 1.05

C = 0.188

P = 1.0 (straight row)

Predicted Erosion (A) = 8 T / A / YR

Soil Loss Tolerance T = 5 T / A / YR

Therefore: 8 / 5 = 1.6 multiples of T

Erosion (ER) subfactor (Table 4) = - 0.8

Erosion subfactor = F (A / T)

A / T = multiples of T

ER sub factor = multiples of T

(D) **Soil Conditioning Index (SCI):**

SCI = (OM * 0.5) + (FO * 0.2) + (ER * 0.3)

= (0.12 * 0.5) + (+ 0.35 * 0.2) + (- 0.8 * 0.3)

= 0.06 + 0.07 – 0.24 = - 0.11

The SCI value is negative, hence soil organic matter is predicted to be decreasing, and corrective measures should be planned. Erosion is the major factor affecting organic matter loss. Some alternatives are: (a) change to a no-till system, which will reduce erosion and minimize soil disturbance, or (b) apply measures such as terracing and contour farming to reduce erosion.

Management Information, Alternative System:

To see the importance of SCI value, assume that Management has agreed in a meeting to find a remedy to the situation in Lincoln by adopting an alternative practice. The new information used in the calculation are the as a result of the decision changes.

(A^1) **Organic material (OM):**

Rotation and predicted yields: Corn 125 bu/ac, Drilled Soybeans 40 bu/ac.

Residue management: all residues returned.

Residue returned (adjusted to REV using conversion factors in Table 1):

Corn 125 * 56 * 1.0 = 7000 lb / ac, REV

Drilled Soybeans 40 * 75 * 0.92 = 2760 lb / ac, REV

Total REV / 2 = 9760 / 2 = 4880 lb / ac, average annual REV produced (RP)

Organic Material (OM) Subfactor (RP – MA) / MA = OM

(4880 – 4198) / 4198 = + 0.16

(B[1]) **Field operations (FO):**

Year 1: Anhydrous application, Soil Disturbance Rating (Table 3) = 8; plant corn W/no-till planter, fluted coulter, SDR = 5; harvest, SDR = 5.

Year 2: plant soybeans w/no-till drill w/single disk openers, SDR = 2; harvest, SDR = 5

The total SDR for the rotation is

$$8 + 5 + 5 + 2 + 5 = 25.$$

The average annual SDR is 25 / 2, which is 12.5

We determine the Field Operations (FO) Subfactor from Table 4 of appendix B as + 0.85

(C[1]) **Erosion (ER)**

R = 150

K = 0.27 Adjusted

LS = 1.05

C = 0.076

P = 1.0 (straight row)

Predicted Erosion A = 3.2 T / A / YR (from RUSLE)

Soil Loss Tolerance T = 5T / A / YR

Therefore: 3.2 / 5 = 0.64 multiples of T

From Table 5 Erosion subfactor (ER) = + 0.20

(D[1]) **Soil Conditioning Index (SCI)**

$$
\begin{aligned}
SCI &= (0.5 * OM) + (0.2 * FO) + (0.3 * ER) \\
&= (0.5 * 0.16) + (0.2 * 0.85) + (0.3 * 0.20) \\
&= 0.08 + 0.17 + 0.06 = + 0.31
\end{aligned}
$$

The *SCI* value is positive, hence soil organic matter is predicted to be increasing, and this alternative is suitable.

Management Benefits of SCI Information

SCI aids management in making decisions on the soil management practice. Although, the alternative approach eliminates the earlier problem, this would have gone undetected except through the use of the SCI model. A negative SCI is an indicator of reduced organic matter trends; hence management must introduce an alternative practice that offers an improvement over the first. Zero or more SCI value is an indicator of stability or increase in organic matter levels.

Section 3

SISL-SCI Relational Database

In this section, we describe the design of the relational database system for SISL-SCI. We first give a summary of the purposes of the tables as obtained from the analyses of each of the two applications. We then proceed with the schema diagram also known as Entity relationship model, depicting the relationships among the tables and how we have achieved the integration the SISL and SCI. The section concludes with detailed descriptions of SCI tables and their columns.

3.1 SISL-SCI Relational Tables

The SISL-SCI tables revealed in our relational analysis are listed below. We have enclosed in parenthesis at the end of each table description the name of the database that uses it. We have carefully chosen the names of our tables to depict the exact meaning of the contents.

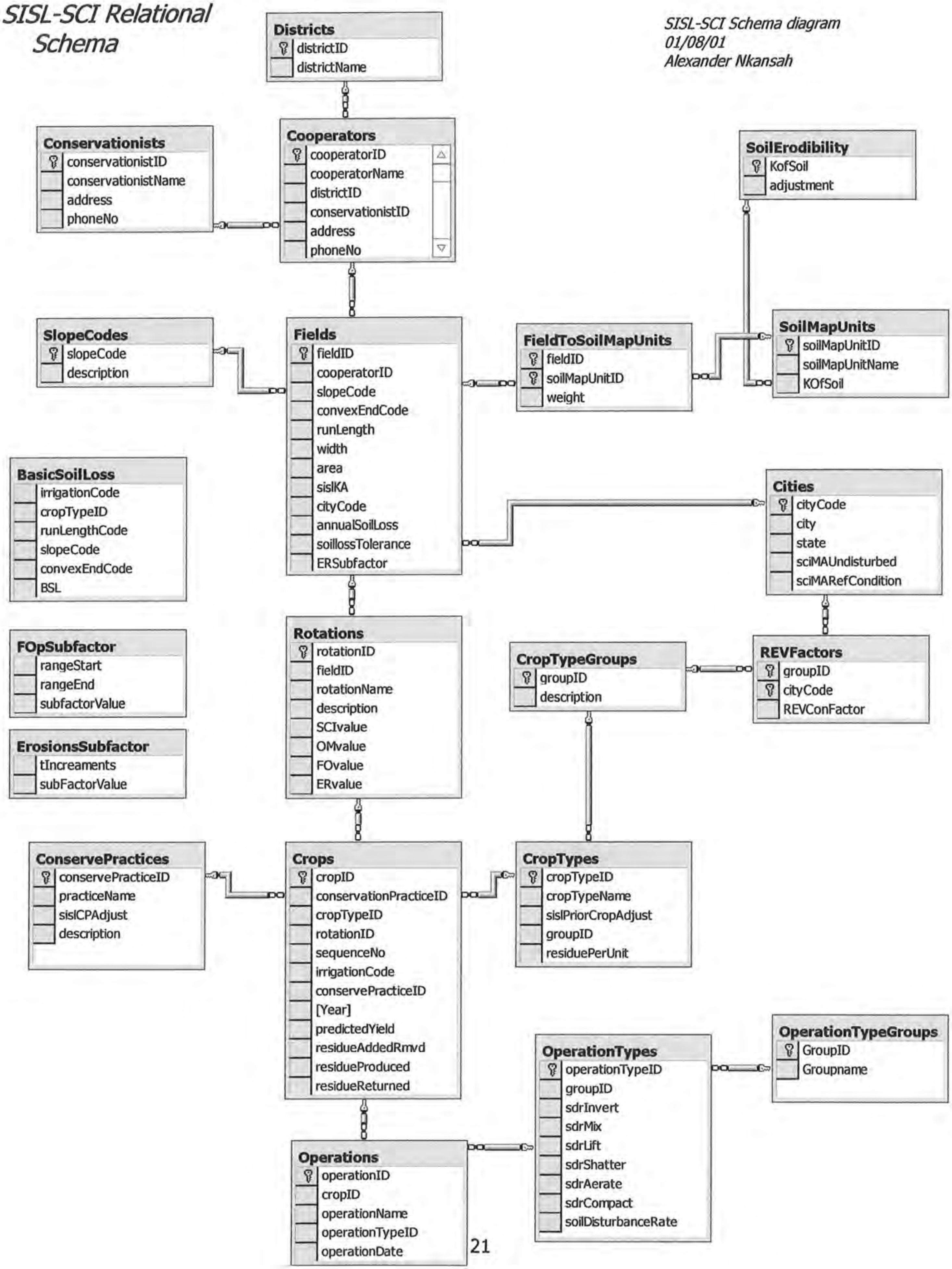
Table Districts	Stores all the information pertaining to a district (SISL).
Table Cooperators	Stores the information on cooperators (farmers) in a given district. One district have many cooperators (SISL).
Table Conservationists	Stores the data on each conservationist in a given district (SISL).
Table ConservePractices	Stores the information on the different conservation practices that have been practiced on a given field (SISL).

Table Fields	Stores the information on a field in a given location. This is the key table that makes the integration of the two databases possible (SISL, SCI)
Table SoilMapUnits	Stores the information of the soil in one geographical area with the same properties, e.g. soil erodibility (SISL).
Table FieldToSoilMapUnits	Stores the information on how much, weight of the field, designated by a fieldID belongs to the soil map unit (SISL).
Table SoilErodibility	Stores the information on the soil erodibility adjustment factors soil with respect to the base erosion data is obtained (SISL).
Table SlopeCodes	Stores the information for the different types of slopes (SCI).
Table BasicSoilLoss	Stores the information for the soil loss due to the nature of the land, the type of irrigation practices and the type of crop (SCI)
Table Cities	Stores the information on cities and their maintenance amount for field locations (SCI).
Table RevFactors	Stores the residue equivalent values for the fields in each city (SCI).

Table Rotations	Stores the rotation information for a given field. i.e. the description of crops that are planted in the rotation period (SCI)
Table Crops	Stores the data on the actual crops that are planted in a given rotation (SCI).
Table CropTypes	Stores the type information of crops, e.g., wheat, soybean and alfalfa (SCI).
Table CropGroupTypes	Stores the grouping information of crop types: A - cotton, burley tobacco and peanuts (SCI) B – corn, grain sorghum, sunflower, etc.
Table Operations	Stores information on the actual operations that are performed on a field during the lifetime of a given crop in a rotation, e.g. an actual tillage (SCI)
Table OperationTypes	Stores the type information of an operation, e.g. e.g., plow, disk plow etc. (SCI)
Table OperationTypeGroups	Stores the grouping information of operations. Different tillage types are classified into groups of primary tillage secondary tillage, etc. (SCI).
Table ErosionSubfactors	Stores the *REV* conversion values for the *ER* sub-factor (SCI).
Table FOSubfactors	Stores the REV conversion values for the *FO* sub-factor. (*SCI*)

SISL-SCI Relational Schema

SISL-SCI Schema diagram
01/08/01
Alexander Nkansah



3.2 SISL-SCI Relational Model

Our relational schema diagram for the SISL-SCI database follows the Microsoft SQL Server 2000 convention. The Relational schema diagram otherwise known as the *Entity Relationship Diagram (ERD)* is a graphical representation of the relationships among the tables in the database. Generally, relationships among tables could have one-to-one, many-to-one or many-to-many cardinality. We have converted all many-to-many cardinalities into two one-to-many cardinalities by introducing an intermediate table. In our notation, the key end of the relationship points to the primary table or the one side of the relationship and the inverted ampersand (∞) end represents the many side of the relationship. The diagram depicts the association among the tables in the integrated SISL-SCI database. For further reading, see Microsoft Books Online and/or Microsoft Developer Network (MSDN) documentations. [6], [7]

We refer to each table as a *relation instance* in the system. The *relation instance* is a table in which each tuple is a row, and all rows have the same number of fields. In this section, we give the details of the *domain constraints* by citing examples of how we have applied it to SISL-SCI relational model. These domain constraints specify an important condition that we want each instance of the relation to follow; hence it defines the data types of the data attributes and other limitations on the values stored in each field. Based on the knowledge of the domain constraints on the data elements we created and modified all the SISL-SCI tables using the subset of the SQL know as *Data Definition Language* (DDL). In creating the DDL's for our tables we have strictly adhered to the SQL-92 language standard [3].

To maintain a consistent and a formidable data as well as to prevent entry of incorrect information, we implemented *Integrity constraints* (IC) on the tables. IC is a condition that is specified on a database and enforced by the Database Management System (DBMS) to restrict the data that can be stored in an instance of the database. In this project, we have implemented *Primary Key Constraints, and* the *foreign key constraints* that lead to enforcing *referential Integrity Constraints.* Other constraints were implemented via the stored procedures and triggers.

A primary key is a statement that a certain minimal subset of the fields in a table is a unique identifier for a tuple, where a tuple is a row of a table. In our relational schema diagram i.e., Figure 2, key fields are indicated by the key symbol. e.g., the key for table Crops is cropID, this means that no two records or tuples in the table Crops will have the same cropID. In other words the cropID uniquely identifies a row of the table Crops. This same scenario applies to each of the SISL-SCI tables. The choice of a key strongly influences the uses of the database and hence we have chosen them with extra care

Foreign key constraints is used to link information stored in one table to information stored in another table. For example to ensure that crops are planted only in bona fide rotations, we defined foreign key rotationID in table Crops. This ensured that the rotationID in table Crops also exist in table rotations.

The last but not the least of the constraints employed in our implementation of the relation schema is the referential integrity constraints. We enforced these constraints by implementing primary keys and foreign keys as well as unique constraints. From the Crops-Rotation illustration above, when a rotation is deleted from the table Rotation, all the crops having rotationID's corresponding to the deleted rotation must also be deleted from table Crops. This kind of referential integrity is normally referred to as Cascade Delete Constraint. This ensures that orphan records are not retained in the Database. We modified all of our DDL to reflect the constraints discussed above and re-ran our DDL on an SQL Server 2000 to produce the Figure 2.

Integration of SISL and SCI

The SCI and SISL are applications that apply to the soil in a given city location. Therefore these two have some data elements relating to the field entity. Again, they both depend on the crop *rotation* as well as the *crops* in the rotation and the *crop type* they belong to. These constitute the four tables Fields, Rotations, Crops and Croptypes identified in SISL-SCI entity relation model. Once we identified these commonalities, we merged all the data elements from SISL and SCI that belonged to these entities into the four tables mentioned above. All the other tables originally from the individual databases are retained in the new SISL-SCI database. The detail is shown in the Figure 2.

3.3 SISL-SCI Table Column Definitions

The data columns of the SCI tables are explained in this subsection. We give a tabular listing of the individual tables and the data types of the column followed by a brief description of the purpose of each column. These tables are the same as the ones described at the beginning of this section.

Table Fields

Column Names	Data Type	Description
fielded	int	Unique Identifier of a field
cooperatorID	int	Unique Identifier of a cooperator
slopeCode	char	unique Identifier of a slope: A = (< 1%), B = (1-1.9%) C = (2-2.9%), D = (> 3%)
convexEndCode	char	type of the convex end: N = none , M = medium, S = severe
runLength	float	run Length of a field
width	char	width of a field
area	float	total area of the field
sislKA	float	SISL value for the field
cityCode	numeric	unique numeric Identifier of field city (location)
annualSoilLoss	float	soil loss calculated from RUSLE: (tones/acre/yr)
soilLossTolerance	int	soil loss tolerance of the field: ability of the soil to withstand adverse conditions:(tones/acre/yr)
ERSubfactor	float	calculated erosion subfactor for SCI

convexEndCode:

Medium -- less than 6" from field level grade to the bottom of a tail water ditch

Severe -- greater than 6" from field level grade to the bottom of a tail water ditch

sislKA:

Soil erodibility adjustment for the soil in relation to the soil on which the base erosion data was obtained.

Table Cities

Column Names	Data Type	Description
cityCode	numeric	Unique numeric Identifier of a city
city	varchar	name of City of field location
state	char	state of the City
sciMAUndisturbed	int	undisturbed maintenance amount (lb/ac)
sciMARefCondition	int	reference condition maintenance amount (lb/ac)

Table REVFactors

Column Names	Data Type	Description
groupID	char	unique Identifier of a field group
cityCode	numeric	unique numeric Identifier of a city
REVConFactor	real	REV Condition Factor

Table Rotations

Column Names	Data Type	Descriptions
rotationID	int	unique Identifier of a rotation
fielded	int	unique Identifier of a field
rotationName	varchar	name of the rotation
description	varchar	description of the rotation
SCIvalue	float	calculated SCI for the rotation
OMvalue	float	calculated OM for the rotation
FOvalue	float	calculated FO for the rotation
ERValue	float	calculated ER for the rotation

Table Crops

Column Names	Data Type	Description
cropID	int	unique Crop Identifier
cropTypeID	int	unique Crop type Identifier
rotationID	int	unique Identifier of a rotation
sequenceNo	int	sequence Number of the crop in rotation
irrigationCode	char	code or irrigation method
conservePracticeID	int	conservation Practice Identification
[Year]	int	year of the crop in rotation
predictedYield	float	predicted yield of crop in rotation (bu/ac)
residueAddedRmvd	float	amount of residue produced by crop (lb/ac)
residueProduced	float	total Residue produced (lb/ac)
residueReturned	float	residue returned to the soil (lb/ac): residue produced added to residue added or removed

Table CropTypes

Column Names	Data Type	Decription
cropTypeID	int	unique Crop type Identifier
cropTypeName	char	name of the Crop type
sislPriorCropAdjust	float	SISL prior crop adjustment
groupID	char	crop type group Identifier
residuePerUnit	float	residue per unit for each crop type

Table CropTypeGroups

Column Names	Data Type	Decription
groupID	char	crop type group Identifier
description	varchar	description of crop group type

Tables Operations

Column Names	Data Type	Decription
operationID	int	unique operation Identifier
cropID	int	unique crop Identifier
operationName	varchar	name of the operation
operationTypeID	int	unique operation type Identifier
operationDate	datetime	date of the operation

Tables OperationTypes

Column Names	Data Type	Description
operationTypeID	int	unique operation type Identifier
groupID	int	unique operation group type Identifier
sdrInvert	int	soil disturbance due to inverting soil
sdrMix	int	soil disturbance due to mixing soil
sdrLift	int	soil disturbance due to lifting soil
sdrShatter	int	soil disturbance due soil shatter
sdrAerate	int	soil disturbance due to soil aeration
sdrCompact	int	soil disturbance due to compacting soil
soilDisturbanceRate	int	total soil disturbance rate

Table OperationTypeGroups

Column Names	Data Type	Description
groupID	int	unique operation group type Identifier
groupName	varchar	the name of the group

Table FOpSubfactor

Column Names	DataType	DESCRIPTION
rangeStart	int	lower bound for FO-subfactor range
rangeEnd	int	upper bound for FO-subfactor range
subfactorValue	float	actual converted FO value

Table ErosionSubfactor

Column Names	Data Type	DESCRIPTION
t-Increments	int	The soil Loss tolerance multiple value for erosion. i.e. the multiples of soil loss tolerance
subFactorValue	float	actual converted erosion subfactor value

Section 4

Implementation of SCI Stored Procedures and Triggers

In this section, we discuss the detailed implementation of the stored procedures used to calculate the three subfactors of SCI. These subfactors, and hence the overall SCI value, depend on the column values of tables Cities, Crops, Fields, Operations, RevFactors, and Croptype. We implemented triggers on these tables, to be automatically invoked when their rows are updated, inserted, or deleted. The triggers perform SCI calculations by making calls to the SCI stored procedures. Here, it must be emphasized that, the driving force of our new implementation is the rotationID of a rotation. This is passed as an argument to two of our stored procedures.

4.1 *Procedure (OM) Subfactor.*

The calculation of the OM subfactor, previously discussed in section 2.2.3 uses the tables Cities, REVFactors, CropTypeGroups, Crops, and Rotations in our implementation. The OM subfactor is computed and stored for each rotation. Hence the main argument passed to the OM procedure is the *rotationID* for a rotation. The columns of the tables used in the stored procedure OMSubfactor are discussed next.

1. The *Maintenance Amount (MA)* of the field on which a crop in a rotation is planted is obtained from table Cities. It is determined for each field in a city.
2. The *Residue Equivalent Value (REV)* of the crop group type i.e. (groupID from table CropTypeGroups) is used to normalize the OM value based on the *REV* conversion scale. The *REV* is determined for each groupID, and each city i.e. cityCode in table REVFactors.
3. Table Crops allows the crops in each rotation to be identified. The *predicted yield* and *residue amounts* produced by each actual crop are also retrieved from table Crops.

4. The key argument for each of the stored procedures is the rotationID, and it is retrieved from the Rotations table for each rotation.

(A1) ***Computing the Organic Material***

Subsection 2.2.3 of this report discussed the computation of the OM subfactor. In this Section however, we give the new method of computing the OM, and how it is implemented in the stored procedure. The working formula for Organic Material (OM) was introduced in the earlier Section, we state it again as a reminder;

$$OM = (RP - MA) / MA$$

and the variables are as before. The Average residue produced (RP) can be expressed as

$$RP = TotREV / n,$$

where n, is the number of years in the Rotation. We derive the total REV from the mathematical equation,

$$TotREV = \sum_{i}^{n}((Y_i * R_i) + R'_i) * REV_i)$$

i ($i = 1, 2, 3 \ldots n$) is an index for a crop in the rotation

Y_i is the predicted yield of crop i in the rotation

R_i is the residue per unit for crops i in the rotation, belonging to a crop group.

R'_i is the residue removed or added for having crop i in the rotation

REV_i is the REV read from the REV factors table for crop i in the rotation belonging to a particular crop group type. The following Select Transact SQL statement performs the summation.

```
SELECT SUM ((predictedYield * CropTypes.residuePerUnit +
residueAddedRmvd) * Rev.RevConFactor)
FROM Crops
INNER JOIN CropTypes ON Crops.cropTypeID = CropTypes.CropTypeID
INNER JOIN CropTypeGroups ON CropTypes.groupID =
CropTypeGroups.groupID
INNER JOIN REVFactors REV ON CropTypeGroups.groupID = Rev.groupID
```

```
WHERE REV.cityCode = @cityCode and Crops.rotationID =@rotationID
```

The number of years in the rotation is computed as the difference between the maximum and the minimum years in the rotation as shown below.

```
SELECT Max (Crops.year) - Min (Crops.year) + 1 from Crops
WHERE Crops.RotationID = @rotationID
```

Substituting these values in formula yields the average residue produced *RP*.
See the actual implementation of the stored procedure OMSubfactor. It is used in triggers when the OM subfactor value is required.

4.2 *Procedure FO Subfactor.*

The formula used for calculating the FO subfactor value was discussed in Section 2.2.3. In our implementation, we used stored procedure FOSubfactor. The SCI tables accessed in this stored procedure are table Rotations, table Crops, table Operations, table OperationTypes, and table FO-Subfactor. The columns involved in the computation are *rotationID* from table Rotations, cropID from table Crops, operationID from table Operations, operationtypeID and *SDR* from table OperationsTypes, and FOSubfactorValue in the range of ranges tart and rangeEnd all from table FOSubfactor. Here again, the FO subfactor value is computed and stored for each rotation therefore, hence the key argument passed is the *rotationID*.

(B1) ***Computing of the FO subfactor.***

In section 2.2.3 B, we illustrated with example how the FO subfactor is computed. Here, we develop the formula which can be translated directly into a transact SQL statement. The FO subfactor is the sub factor value (read from table 4) for the sum of all the soil disturbance ratings (*SDR*) for operations performed for all crops in a rotation. Mathematically, this can be expressed as.

$$total\ SDR = \sum_{i}^{m} \sum_{j}^{n} SDR_{i,j}$$

$SDR_{i,j}$ - The Soil disturbance rating for operations performed on a given Crop.

i, ($i = 1,2,3 \ldots,m$) indicates a crop in the rotation

j, ($j = 1,2,3 \ldots n$) the index of the operation performed for crop i,

m, is the upper bound for total number of crops in the rotation

n, is the upper bound for the operations performed on crop i.

We determine from the table Crops the list of crops belonging to a given rotation followed by the actual operations that are performed for the selected crops in the rotation and subsequently the SDR's for the operations types. We computed the total SDR for all crops in a rotation using the select Transact SQL below.

```
SELECT SUM (OT.soilDisturbanceRate)
FROM OperationTypes   OT
INNER JOIN Operations OP ON OP.operationTypeID = OT.operationTypeID
INNER JOIN Crops ON OP.cropID = Crops.cropID
WHERE Crops.rotationID =  @rotationID
```

We then proceed with the calculation of the average annual SDR as

$$Average\ SDR = total\ SDR / n,$$

n, is total number of years in the rotation.

The subfactorvalue is retrieved from the table FOsubfactors as shown in the following T-SQL statement

```
SELECT subfactorValue FROM FOpSubfactor
WHERE round (@avgSoilDistRate, 0) between rangeStart and rangeEnd
```

The FOSubfactor stored procedure is then called from triggers on the tables when the columns are being modified

4.3 Procedure (ER) Subfactor

Two implementation options exist for FOSubfactor:

a) The first option, calculates the *ER* value without referencing the t-increments table. The *ER* is calculated from an equation that generates table t-increments without actually using the table. The advantage here is that there is no storage wastage in storing the t-increment values in a table and hence no CPU wastage in retrieving stored values. If desired, the value is computed from the equation that generates the table on the fly given the appropriate parameters it needs

b) The second implementation references the t-increments table to access the t-increment values. In this case there is the need to have t-increments table, thereby taking away some extra storage space that would otherwise have been preserved. Our approach is to use the optimum solution whenever more than one options exists hence we adhered to the first approach since it offers an advantage over the latter.

The *ER* Subfactor equation is:

$$\text{ERSubfactor} = 99981 - 1.3106 * \textit{T-increments} + 0.1192 * (\textit{T-increments})^2 - 0.0042196 * (\textit{T-increments})^3$$

This equation holds for T-increments in the range between 0 and 8 inclusive
i.e. (0 <= T-increments >= 8.)

For values of T- Increments greater than 8 the proposed Equation becomes

$$\textit{ERSubfactor} = -0.2 * \textit{T-increments},$$

Unlike the previous two Stored procedures, this one uses the FieldID of the table Field's as the key argument in the procedure call. This is because the FOSubfactor is an attribute of a field. We also stored this value in the table Rotations to be used for comparism if needed. This stored procedure is also called in all the triggers that calculate the SCI values. The actual usage is illustrated in the Triggers.

(D) ***The Soil Condition Index (SCI)***

We stated earlier in the introduction of this report that, *SCI* has the following relation based on the relative importance of the three factors.

SCI = 50% * OM + 20% * FO + 30% * ER.

Now that we have calculated all three subfactors, we substitute the return values of the three stored procedures into equation above and store the *SCI* value for each rotation in the table Rotation.

We also implemented the triggers, which calls these stored procedures for the calculation of *SCI* value. Four Major triggers are defined on four SCI-SISL database tables besides other minor triggers. These are the table *Fields*, table *Cities*, table *REVFactors*, and table *Operations*. Each of the triggers are activated or fired when the some key columns of these tables are updated, inserted or deleted.

4.4 Implementation of SCI Triggers

For the purposes of automatic invocation and execution of the stored procedures for the three SCI subfactors, we implemented triggers on the tables. These triggers are procedures that are fired by the Database Management System (DBMS) in response to specified changes in the database. These columns these tables directly or indirectly affect one of the three factors and hence the overall *SCI* value. We implemented triggers on tables Cities, Crops, Fields, Operations, REVFactors and Croptype. They are fired when the affected columns on the tables are modified. The complexity of the T-SQL depends on how far the target table is away from the table rotation, i.e., the focal point of all calculations. The farther away a table is from the table rotation in the relationship schema, the more complex the joins and vice versa. Here, we discuss the trigger implementation on the table REVFactors. The same technique applies to all the other triggers. We hope the reader will find it useful in understanding the remainder of the triggers. Refer to figure 3 below for the SISL-SCI trigger flow diagram.

4.4.1 Trigger on Table **REVFactors**

The table REVFactors was discussed in Subsection 3.1 under SISL-SCI Tables, and the detailed columns was explained in Subsection 3.2. The table consists of three columns namely; groupID, cityCode, and REVConfactor. All three columns are required in the computation of SCI for a rotation on each field. The REV conversion is required to normalize the calculated OM subfactor value. For each rotation on field REV conversion factor, REVConfactor is determined by the crop group Identification i.e groupID and the City i.e., cityCode. This means that any modification to the city code or REV conversion factor must trigger the calculation of the SCI value. Here, it is assumed that a crop can only belong to one and only one group and will always maintain a constant group Identification. The trigger OMTriggerOnREVFact is shown below. The rest of the triggers and the stored procedures can be found in the appendix A. From the schema diagram, many rotations could be affected by an update of table Cities. Therefore we require the use of a cursor to traverse through each rotation record one at time.

```
IF UPDATE (REVConFactor) OR UPDATE(cityCode)
BEGIN
  DECLARE  DistinctRotID CURSOR  FOR
      SELECT DISTINCT  R.rotationID from Rotations  R
      INNER join Fields F ON R.fieldID = F.fieldID
      INNER join Cities C ON F.citycode  = C.citycode
      INNER JOIN REVFactors  V ON C.citycode = V.citycode
      INNER join Inserted I ON V.citycode =I.citycode

 OPEN  DistinctRotID
 FETCH NEXT FROM DistinctRotID INTO @rotIDs
 WHILE (@@fetch_status  =  0)
 BEGIN
   SELECT @fieldID = F.fieldID
   FROM Fields F
```

```
    INNER JOIN Cities C ON C.cityCode = F.cityCode
    INNER JOIN REVFactors RF ON RF.cityCode  = C.cityCode
    INNER JOIN INSERTED I ON I.cityCode  = RF.cityCode

IF UPDATE (REVConFactor) OR UPDATE(cityCode)
BEGIN
   DECLARE  DistinctRotID CURSOR  FOR
        SELECT DISTINCT  R.rotationID from Rotations  R
        INNER join Fields F ON R.fieldID = F.fieldID
        INNER join Cities C ON F.citycode  = C.citycode
        INNER JOIN REVFactors  V ON C.citycode = V.citycode
        INNER join Inserted I ON V.citycode =I.citycode
  OPEN  DistinctRotID
  FETCH NEXT FROM DistinctRotID INTO @rotIDs
  WHILE (@@fetch_status  =  0)
  BEGIN
     SELECT @fieldID = F.fieldID
     FROM Fields F
     INNER JOIN Cities C ON C.cityCode = F.cityCode
     INNER JOIN REVFactors RF ON RF.cityCode  = C.cityCode
     INNER JOIN INSERTED I ON I.cityCode  = RF.cityCode

--Call the stored proceures to calculate and store the SCI Subfactors
    EXEC OMsubfactors @rotIDs
    EXEC FOSubfactors @rotIDs
    EXEC ErosionSubfactors @fieldID
        SELECT @OMsubfactor = R.OMValue , @FOsubfactor = R.FOValue,
               @ERsubfactor = F.ERsubfactor
     FROM  Rotations R
     INNER JOIN Fields F ON F.fieldID = R.fieldID
     INNER JOIN Cities C ON C.cityCode = F.cityCode
```

```
    INNER JOIN REVFactors RF ON RF.cityCode  = C.cityCode
    INNER JOIN INSERTED I ON I.cityCode = RF.cityCode
    WHERE R.rotationID = @rotIDs
      --Substitute into the SCI equation the subfactor values.
   SELECT @SCIval = 0.5 * @OMsubfactor + 0.2 * @FOsubfactor + 0.3 *
                        @ERsubfactor
   BEGIN TRANSACTION
       UPDATE  Rotations
       SET  SCIvalue = @SCIval, OMvalue = @OMsubfactor, FOvalue =
                        @FOsubfactor,  ERvalue = @ERsubfactor
       WHERE  Rotations.rotationID = @RotIDs
   COMMIT TRANSACTION
   FETCH NEXT FROM DistinctRotID INTO @RotIDs
 END
 CLOSE DistinctRotID
 DEALLOCATE DistinctRotID
END
```

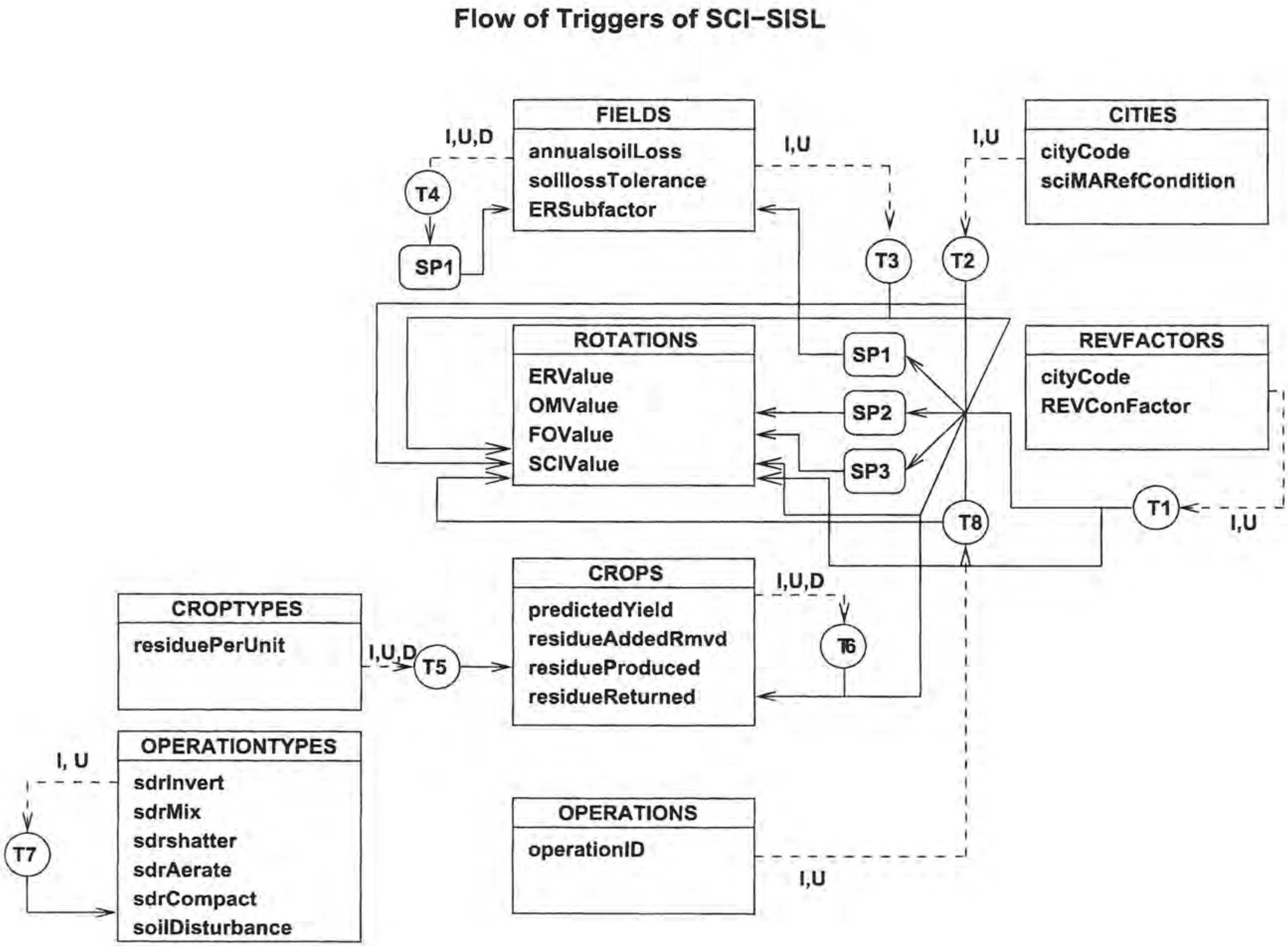


Figure 3 SCI trigger flow diagram

Section 5

Conclusions and Future Work

We designed a relational database for two agricultural applications, namely *Soil Conditioning Index (*SCI*)* and *Surface Irrigation Soil Loss* (SISL*)*. The databases of these two applications were integrated into one SISL-SCI database system, and we implemented the stored procedures and triggers that calculate the *SCI* value for a rotation of each field. In our design of the SISL-SCI database schema, we strictly adhered to normalization principles.

The SCI is used as a tool by Natural Resources and Conservation Services (NRCS), and other agriculturist to predict the consequences of cropping systems and tillage practices on the status of soil organic matter, which is a primary indicator of soil condition. The three main subfactors of the SCI application are the *organic material* subfactor *OM*, the *field operation* subfactor *FO* and the *erosion* subfactor *ER*. In our implementation, *SCI* is calculated by the equation:

$$SCI = 0.5 * OM + 0.2 * FO + 0.3 * ER.$$

A negative *SCI* magnitude predicted from this model indicates decreasing organic matter trends over the years of a crop rotation. Management must remedy such a situation by finding an antidote to the negative SCI scenario. A positive *SCI* means improvement in the soil condition. A small SCI value indicates a relative stability in the soil condition.

We believe that triggers and stored procedures will increase the efficiency of the SCI system vis-à-vis the present system. Furthermore, we expect that the automatic computations of the SCI value and its subfactors will save the interface application programmer an extra effort required to calculate the SCI values.

It is also our hope that SISL-SCI database will become a foundation upon which several other agricultural applications such as *Revise Universal Soil Loss Equation* (RUSLE), *Wind Erosion Estimates model* (RWEQ) will be integrated. A web-based interface will be developed in the near future for our integrated database system.

Bibliography

[1] D.T. Lightle and M.S. Argabright, *A Soil Conditioning Index For Cropland Management Systems, version 1.1, National Soil Survey center, January 97*

[2] Floyd G. Bailey and Dr. David L. Carter, Predicting the effect of induced soil loss on surface cropland, *Technical Notes, USDA-Soil Conservation Service Boise, Idaho, September* 1994

[3] Raghu Ramakrishnan and Johannes Gehrke, *Database Management Systems 2nd Ed., McGraw-Hill Companies Inc, 1998, pp. 58 - 161*

[4] Martin Gruber, *Mastering SQL, New Edition, SYBEX Inc., 1999*

[5] Mark Spenik, MCSD and Orryn Sledge MCSD, et al., *Microsoft SQL Server 7.0 DBA Survival Guide,* Macmillan Computer Publishing, 1999, pp. 493 - 597

[6] Microsoft, *Microsoft SQL Server 2000 Books on Line*

[7] Microsoft Developer Network (MSDN), Online Knowledge Base, *http://**msdn**.microsoft.com/default.asp*

[8] Mohan Choodamani, Lin Li and Toshimi Minoura, *Automatic Generation of Forms and Java Servlets for a Web-Based Database Application, unpublished manuscript, Dept. of Computer Science Oregon State Univ.,2000.*

[9] Ecological Scientist Delivery, http://www.info.usda.gov/nrcs/fowr/ftp/sciver13.xls

Appendix A

SCI Stored Procedures and Triggers.

Three stored procedures were implemented for the three SCI subfactors. The detailed listings of the source code for the procedures are given below. The Organic material subfactor stored procedure comes first followed by the Field Operations and then the Erosion subfactor stored procedure. We also provide the program listing for all the triggers on the tables.

```
--*******************************************************************
--* PROCEDURE NAME :    OMSubfactors                                *
--* PURPOSE     :    Calculates OMSubfactors for SCI and stores     *
--*                  Result in table Rotations                      *
--* Date/Modified  :    11/12/2001 / 02/13/2001                     *
--*******************************************************************

CREATE PROCEDURE OMSubfactors @RotationID INT
AS
  DECLARE @residueProduced float    -- The residue produced for crop
  DECLARE @avgResidueProduced float  -- Averave Residue produced
  DECLARE @MARefCondition INT       -- Maintanence Amount for city
  DECLARE @errorVal INT            -- Error Value
  DECLARE @cityCode INT            -- City Code for the City
  DECLARE @organicMaterial FLOAT    -- Organic Material Calculated
  DECLARE @rotationYears INT            -- Number of years in the rotation
--=====================================================================
  SELECT @ResidueProduced = 0.0  -- Initialise the residue produced to 0

-- Select the Sum of the produce Predicted Yield, Residue and the REV

  BEGIN TRANSACTION

    SELECT @cityCode = C.cityCode FROM Cities C
    INNER JOIN Fields f ON f.cityCode = c.cityCode
    INNER JOIN Rotations R ON R.fieldID = F.fieldID
    WHERE R.rotationID  = @rotationID

    SELECT @residueProduced = (SELECT SUM(((predictedYield *
        CropTypes.residuePerUnit)+ residueAddedrmvd ) * Rev.RevConFactor)
    FROM Crops
    INNER JOIN CropTypes ON  Crops.croptypeID = CropTypes.CropTypeID
    INNER JOIN CropTypeGroups ON CropTypes.groupID =  CropTypeGroups.groupID
    INNER JOIN REVFactors REV ON CropTypeGroups.groupID = Rev.groupID
    WHERE REV.citycode = @cityCode and Crops.rotationID =@rotationID)

      --Check for the presence of Errors and display message

    If @@ERROR > 0
    BEGIN
```

```
      PRINT 'The query cannot be executed at this time '
      SELECT @ErrorVal = @@ERROR
      ROLLBACK TRANSACTION
    END

    --Find the the number of years in the rotation
    SET @rotationYears = (SELECT Max(Crops.year) - Min(Crops.year) + 1 from Crops
    WHERE Crops.RotationID = @RotationID)

    IF (@rotationYears <= 0)
    BEGIN
      SELECT   @rotationYears = 1
    END

    SELECT @avgResidueProduced = @residueProduced /cast(@rotationYears AS FLOAT)
    SELECT @MARefCondition = (SELECT sciMARefCondition FROM Cities
    WHERE cityCode =  @cityCode)

    SELECT @organicMaterial = ((@avgResidueProduced - cast(@MARefCondition AS
FLOAT))
                    /cast(@MARefCondition AS FLOAT))

    UPDATE Rotations
    SET OMvalue = @organicMaterial
    WHERE rotationID = @rotationID

  COMMIT TRANSACTION
GO



--*********************************************************************
--* PROCEDURE NAME :     FOSubfactors                                  *
--* PURPOSE      :    Calculates FO Subfactor for SCI and stores       *
--*                   Result in Rotations                              *
--* Date/Modified  :     11/12/2001 / 02/13/2001                       *
--*********************************************************************


CREATE PROCEDURE FOSubfactors @rotationID INT
AS
  DECLARE @Cursorerror INT
  BEGIN TRAN

--Trap an Errors associated with the select Query
--Please trap the error for the case when the crop does not have
-- any operation
--================================================================
DECLARE @FOsubfactors FLOAT
  DECLARE @soilDistRate FLOAT
  DECLARE @sumSoilDistRate FLOAT
  DECLARE @avgSoilDistRate FLOAT
  DECLARE @cropID INT
  DECLARE @rotationYear INT
--================================================================
```

```
-- Initialise Variables
   SET @soilDistrate = 0
   SET @sumsoilDistRate = 0

   BEGIN
        SET @soilDistRate = (SELECT SUM(OT.soilDisturbanceRate)
     FROM  OperationTypes OT
        INNER JOIN Operations OP ON OP.operationTypeID = OT.operationTypeID
        INNER JOIN Crops ON  OP.cropID =  Crops.cropID
        WHERE  Crops.rotationID =  @rotationID )
        SELECT @sumsoilDistRate =  @soilDistRate

--Checks for Errors upto this point in code. If any give appropriate Error
        IF @@Error <> 0
        BEGIN
           SET @Cursorerror = @@Error
           RAISERROR ('Cannot Proceed With FOSubfactor Calculation :
           Error % d' ,10, 1, @Cursorerror )
           RETURN
       ROLLBACK TRAN
        END
   END

   SET @rotationYear = (SELECT MAX(Crops.year) - MIN(Crops.year) + 1
   FROM Crops
   WHERE Crops.RotationID = @RotationID)

--We have no more than one year of rotation
   IF (@rotationYear <= 0)
   BEGIN
      SELECT @rotationYear = 1
   END

   SELECT @avgSoilDistRate = @sumsoilDistRate / @rotationYear
   SELECT @FOsubfactors = (SELECT subfactorValue FROM FOpSubfactor
   WHERE round(@avgSoilDistRate,0) between rangeStart and  rangeEnd)
        IF @@Error <> 0
     BEGIN
           SET @Cursorerror = @@Error
           RAISERROR ('Subfactor Value cannot be obtained  :
       Error %d' ,10, 1, @Cursorerror )
           RETURN
        ROLLBACK TRAN
        END
   UPDATE Rotations
   SET FOValue = @FOSubFactors
   WHERE rotationID = @rotationID
PRINT @FOSubFactors

--At this point we are successful so commit the transaction
COMMIT TRAN
GO
```

```
--*******************************************************************
--* PROCEDURE NAME :    Erosion Subfactor                            *
--* PURPOSE     :    Calculates ER Subfactor for SCI and stores      *
--*                  Result in Rotations table                       *
--* Date/Modified  :    11/12/2001 / 02/07/2001                      *
--*******************************************************************

CREATE PROCEDURE ErosionSubfactors @fieldID INT
AS

--Variable  Declaration Section of Erosion Subfactor Procedure

   DECLARE @ERsubfactor  FLOAT
   DECLARE @predictedErosion FLOAT
   DECLARE @tincrements FLOAT
   DECLARE @annualSoilLoss FLOAT
   DECLARE @soillossTolerance FLOAT

--Procedure  Erosionsubfactor calculates the ER factor for SCI
--Input Tables - Fields
--Columns     -RUSLE
--Subfactor Value is deduced from formula (.99981-1.3106*T+0.1192*T*T
--                                              -0.0042196*T*T*T)
--for 0<=T<=8. For T>8 Formula is (-2.4-.1*T)
--=================================================================

   SELECT @annualSoilLoss = annualSoilLoss,
        @soillossTolerance = soillossTolerance
   FROM FIELDS
   WHERE fieldID = @fieldID

   SET @predictedErosion = @annualSoilLoss
   SET @tincrements = @predictedErosion / @soillossTolerance

--Verify that we are in the range of the T-Increaments
   IF ((@tincrements >= 0) and (@tincrements <= 8))
   BEGIN
        SELECT @ERsubfactor = 0.99981 - 1.3106 * @tincrements +
         0.1192 * @tincrements * @tincrements - 0042196 * @tincrements
*@tincrements*@tincrements
   END

-- Check for T-Increaments greater than 8 and take alternative action

   IF (@tincrements > 8)
   BEGIN
        select @ERsubfactor = -2.4 - 0.2 * @tincrements
   END

   UPDATE Fields
   SET ERSubfactor = @ERsubfactor
   WHERE fieldID = @fieldID
GO
```

```
--*******************************************************************
--* PROCEDURE NAME :    FOTRIGGER Trigger                            *
--* PURPOSE      :    Calculates Subfactors for SCI and stores      *
--*                   Result in Proper tables                        *
--* Date/Modified  :    11/12/2001 / 02/13/2001                      *
--*******************************************************************

CREATE TRIGGER FOTRIGGER ON dbo.Operations
FOR INSERT, UPDATE
AS

-- variable declaration  section

DECLARE @rotIDs INT            -- Rotation ID for the rotation
DECLARE @omSubfactor FLOAT     -- organic matter Subfactor value
DECLARE  @sciVal FLOAT         -- sci Value
DECLARE @foSubfactor FLOAT     -- field operation Subfactor Value
DECLARE @erSubfactor FLOAT     -- Erosion Subfactor Value
DECLARE @fieldID INT
--=================================================================

SELECT  @rotIDs = (SELECT Crops.rotationID FROM Crops
INNER JOIN INSERTED I ON Crops.CropID = I.CropID)

--Call Stored Procedures to Calculate the SCI Subfactor Values

SELECT @fieldID = F.fieldID
FROM Fields F
INNER JOIN Rotations R ON R.fieldID = F.fieldID
INNER JOIN  Crops C ON C.rotationID = R.rotationID
INNER JOIN Operations O ON O.CropID = C.cropID
INNER JOIN INSERTED I ON I.operationID = O.operationID

--Calls the Stored Procedures to Calculate and store Subfactors
EXEC FOSubfactors @rotIDs
EXEC ErosionSubfactors @fieldID
EXEC OMsubfactors @rotIDs

SELECT @OMsubfactor  = R.OMValue,  @FOsubfactor = R. FOValue,
                  @ERsubfactor = F.ERsubfactor
FROM Fields F
INNER JOIN Rotations R ON  R.fieldID = F.fieldID
INNER JOIN  Crops C ON C.rotationID = R.rotationID
INNER JOIN Operations O ON O.cropID = C.cropID
INNER JOIN INSERTED I ON I.operationID  = O.operationID
WHERE R.rotationID = @rotIDs

--Calculate SCI From the SCI formular.
--===============================================================
SELECT @SCIval = 0.5 * @OMsubfactor + 0.2 * @FOsubfactor + 0.3 * @ERsubfactor
BEGIN TRANSACTION
UPDATE Rotations
SET SCIvalue = @sciVal --OMvalue = @OMsubfactor, FOvalue =
                  --@FOsubfactor, ERvalue = @ERsubfactor
WHERE Rotations.rotationID = @rotIDs
```

```
COMMIT TRANSACTION
--*****************************************************************
--* PROCEDURE NAME :    ResidueProduced Trigger                      *
--* PURPOSE     : Triggers the calculation of SCI when table crops is *
--                modified                                           *
--*               and stores result in right tables                  *
--* Date/Modified  :     04/1/2001                                   *
--*****************************************************************

CREATE TRIGGER ResidueProduced ON [dbo].[Crops]
FOR INSERT, UPDATE, DELETE
AS

DECLARE @rotIDs INT
DECLARE @fieldID INT
DECLARE @OMsubfactor FLOAT
DECLARE  @SCIval FLOAT
DECLARE @FOsubfactor FLOAT
DECLARE @ERsubfactor  FLOAT

IF UPDATE (predictedYield) or UPDATE ( residueAddedRmvd)
BEGIN

        SELECT @rotIDs = Crops.rotationID,  @fieldID  = R.fieldID
        FROM Crops
        INNER JOIN Rotations R ON Crops.rotationID  =  R.rotationID
        INNER JOIN INSERTED I ON Crops.cropID = I.cropID
        WHERE  Crops.cropID = I.cropID

        EXEC FOSubfactors @rotIDs
        EXEC ErosionSubfactors @fieldID
        EXEC OMsubfactors @rotIDs

        SELECT @OMsubfactor  = R.OMValue,  @FOsubfactor = R. FOValue ,
                               @ERsubfactor = F.ERsubfactor
        FROM Fields F
        INNER JOIN  Rotations R ON R.fieldID = F.fieldID
        INNER JOIN Crops ON Crops.rotationID = R.rotationID
        INNER JOIN INSERTED I ON I.cropID  = Crops.cropID
        WHERE R.rotationID = @rotIDs

--Calculate SCI value from the SCI formular.
--=================================================================

        SELECT @SCIVal = 0.5 * @OMsubfactor + 0.2 * @FOsubfactor + 0.3 * @ERsubfactor


        BEGIN TRANSACTION

        UPDATE Rotations
        SET SCIValue = @SCIval  -- OMValue = @OMsubfactor, FOValue = @FOsubfactor,
                ERValue = @ERsubfactor
        WHERE  Rotations.rotationID = @rotIDs

         UPDATE Crops
         SET Crops.residueProduced = (Crops.predictedYield * CropTypes.residuePerUnit),
```

```
        Crops.residueReturned =  (Crops.predictedYield * CropTypes.residueperunit)
                    +Crops.residueAddedRmvd
        FROM Crops
        INNER JOIN CropTypes ON  Crops.croptypeID = CropTypes.CropTypeID
        INNER JOIN INSERTED I ON I.cropTypeID = Crops.CropTypeID

       COMMIT TRANSACTION

END



--*****************************************************************
--* PROCEDURE NAME :     ResidueProducedCT                        *
--* PURPOSE     :     Calculates Residue Produced for crops in    *
--*                the rotations                                  *
--* Date/Modified  :    11/12/2001 / 02/13/2001                   *
--*****************************************************************

CREATE TRIGGER ERTRIGGERONFIELDS ON dbo.Fields
FOR  INSERT,  UPDATE
AS

SET NOCOUNT ON
--=================================================================
--Call FOSubfactor Procedure to calculate FO

DECLARE @rotIDs INT
DECLARE @OMSubfactor FLOAT
DECLARE  @SCIval FLOAT
DECLARE @FOSubfactor FLOAT
DECLARE @ERSubfactor  FLOAT
DECLARE @fieldID INT

IF UPDATE (annualSoilLoss)  OR UPDATE(soilLossTolerance)
BEGIN
  DECLARE DistinctRotID CURSOR FOR
     SELECT R.rotationID from Rotations  R
     INNER join Fields F ON R.fieldID = F.fieldID
     INNER join Inserted I ON F.fieldID = I.fieldID

  OPEN DistinctRotID
  FETCH NEXT FROM DistinctRotID INTO @RotIDs
  WHILE (@@FETCH_STATUS = 0)
  BEGIN
     SELECT @fieldID =  F.fieldID FROM Fields F
     INNER JOIN INSERTED I ON I.fieldID = F.fieldID

   --Execute Stored Procedures to Calculate the SCI factors

     EXEC OMSubfactors @rotIDs
     EXEC FOSubfactors @rotIDs
     EXEC ErosionSubfactors @fieldID
--Calculate SCI From the SCI formular.
--=================================================================
```

```
    SELECT @OMsubfactor = R.OMValue , @FOsubfactor = R.FOValue,
                          @ERsubfactor = F.ERsubfactor
    FROM  Rotations R
    INNER JOIN Fields F ON F.fieldID = R.fieldID
    INNER JOIN INSERTED I ON I.FieldID = F.fieldID
    WHERE R.rotationID = @rotIDs


    SELECT @SCIval = 0.5 * @OMsubfactor + 0.2 * @FOsubfactor +
                0.3 * @ERsubfactor

--Rememeber to replace with function value
     BEGIN TRANSACTION
     UPDATE Rotations
     SET  SCIValue = @SCIval, OMValue = @OMsubfactor,
                FOValue = @FOsubfactor, ERValue = @ERsubfactor
     WHERE Rotations.rotationID = @rotIDs

  --Check for Error and Rollback the transaction if Any
    UPDATE Fields
    SET Fields.ERSubfactor = @ERsubfactor
    FROM INSERTED I
    WHERE I.fieldID = Fields.fieldID
    COMMIT TRANSACTION
    FETCH NEXT FROM DistinctRotID INTO @rotIDs
  END
  CLOSE DistinctRotID
  DEALLOCATE DistinctRotID
END
```

```
--**************************************************************
--* PROCEDURE NAME :    OMTRIGGERONREVFACT                      *
--* PURPOSE     :    Calculates Subfactors for SCI and stores   *
--*                  Result in apropriate tables                *
--* Date/Modified  :    11/12/2001 / 02/13/2001                 *
--**************************************************************

CREATE TRIGGER OMTRIGGERONREVFACT ON dbo.REVFactors
FOR INSERT, UPDATE
AS
--==============================================================
--Call FOSubfactor Procedure to calculate FO
-- Arguements
--Declare @RotID INT

--Please Handle all errors on the next sit down.
--NB Alex
--==============================================================

Declare @rotIDs INT
DECLARE @OMsubfactor FLOAT
DECLARE  @SCIval FLOAT
DECLARE @FOsubfactor FLOAT
DECLARE @ERsubfactor AS FLOAT
DECLARE @fieldID INT

IF UPDATE (REVConFactor) OR UPDATE(cityCode)
BEGIN
  DECLARE  DistinctRotID CURSOR  FOR
        SELECT DISTINCT  R.rotationID from Rotations  R
        INNER join Fields F ON R.fieldID = F.fieldID
        INNER join Cities C ON F.citycode  = C.citycode
        INNER JOIN REVFactors  V ON C.citycode = V.citycode
        INNER join Inserted I ON V.citycode = I.citycode

  OPEN  DistinctRotID
  FETCH NEXT FROM DistinctRotID INTO @rotIDs
  WHILE (@@fetch_status  =  0)
  BEGIN
        set LOCK_TIMEOUT 1000
     SELECT @fieldID = F.fieldID
     FROM Fields F
     INNER JOIN Cities C ON C.cityCode = F.cityCode
        INNER JOIN REVFactors RF ON RF.cityCode  = C.cityCode
        INNER JOIN INSERTED I ON I.cityCode  = RF.cityCode

--We Use Constant for the RUSLE Factor And subtitute later when we
-- know the exact way to do this.

--Call the stored proceures to calculate and store the SCI Subfactors
        EXEC OMsubfactors @rotIDs
        EXEC FOSubfactors @rotIDs
        EXEC ErosionSubfactors @fieldID

        SELECT @OMsubfactor = R.OMValue , @FOsubfactor = R.FOValue,
                  @ERsubfactor = F.ERsubfactor
```

```
        FROM  Rotations R
        INNER JOIN Fields F ON F.fieldID = R.fieldID
        INNER JOIN Cities C ON C.cityCode = F.cityCode
        INNER JOIN REVFactors RF ON RF.cityCode  = C.cityCode
        INNER JOIN INSERTED I ON I.cityCode = RF.cityCode
        WHERE R.rotationID = @rotIDs

--Calculate SCI From the SCI formular.
--================================================================
        SELECT @SCIval = 0.5 * @OMsubfactor + 0.2 * @FOsubfactor + 0.3 *
               @ERsubfactor

         BEGIN TRANSACTION
      UPDATE  Rotations
         SET  SCIvalue = @SCIval, OMvalue = @OMsubfactor, FOvalue = @FOsubfactor,
              ERvalue = @ERsubfactor
      WHERE  Rotations.rotationID = @RotIDs
      FETCH NEXT FROM DistinctRotID INTO @RotIDs
     COMMIT TRANSACTION
  END
  CLOSE DistinctRotID
  DEALLOCATE DistinctRotID
END



--*****************************************************************
--* PROCEDURE NAME :    OMTRIGGER Trigger                            *
--* PURPOSE      :   Calculates Subfactors and SCI and stores       *
--*                     result in the right tables                   *
--* Date/Modified  :    11/12/2001 / 02/13/2001                      *
--*****************************************************************

CREATE TRIGGER OMTRIGGER ON dbo.Cities
FOR INSERT, UPDATE
AS
--Call FOSubfactor Procedure to calculate FO
-- Arguements

Declare @rotIDs INT              -- Unique Rotation ID
DECLARE @OMsubfactor FLOAT         -- OMsubfactor Value
DECLARE  @SCIval FLOAT         -- The Calculated SCI value
DECLARE @FOsubfactor FLOAT         -- FOsubfactor Value
DECLARE @ERsubfactor  FLOAT        -- ERsubfactor Value
DECLARE @fieldID INT               -- field ID of a field
--================================================================

IF UPDATE (sciMARefCondition) OR UPDATE(cityCode)
  BEGIN

  DECLARE DistinctRotID CURSOR FOR
        SELECT R.rotationID FROM Rotations  R
        INNER join Fields F ON R.fieldID = F.fieldID
        INNER join Cities C ON F.cityCode = C.citycode
        INNER join Inserted I ON C.cityCode = I.citycode
```

```
  OPEN DistinctRotID
  FETCH NEXT FROM DistinctRotID INTO @RotIDs
  WHILE (@@FETCH_STATUS <> -1)
  BEGIN
       --Call Stored Procedures to calculate the Subfactors
    EXEC FOSubfactors @rotIDs
    EXEC ErosionSubfactors @fieldID
    EXEC OMsubfactors @rotIDs

    SELECT @OMsubfactor  = R.OMValue,  @FOsubfactor = R. FOValue,
                  @ERsubfactor = F.ERsubfactor
    FROM Rotations R
    INNER JOIN Fields F ON F.fieldID = R.fieldID
    INNER JOIN  Cities C ON C.cityCode = F.cityCode
    INNER JOIN INSERTED I ON I.cityCode = C.citycode
    WHERE R.rotationID = @rotIDs

--Calculate SCI From the SCI formular.
--===================================================================
    SELECT @SCIVal = 0.5 * @OMsubfactor + 0.2 * @FOsubfactor +
                        0.3 * @ERsubfactor

    BEGIN TRANSACTION
      UPDATE Rotations
      SET SCIValue = @SCIval  -- OMValue = @OMsubfactor,
               FOValue = @FOsubfactor, ERValue = @ERsubfactor
      WHERE Rotations.rotationID = @rotIDs
    COMMIT TRANSACTION

    FETCH NEXT FROM DistinctRotID INTO @rotIDs
  END
  CLOSE DistinctRotID
  DEALLOCATE DistinctRotID
END


--*****************************************************************
--* PROCEDURE NAME :    ERsubfactorValue                           *
--* PURPOSE     :     Calculates ERsubfactorValue for SCI and      *
--*                   stores result in table fields                *
--* Date/Modified  :    11/12/2001 / 02/13/2001                    *
--*****************************************************************

CREATE TRIGGER ERsubfactorValue ON [dbo].[Fields]
FOR INSERT, UPDATE, DELETE
AS
DECLARE @fieldID INT
IF UPDATE (annualSoilLoss) OR UPDATE (soilLossTolerance)
BEGIN
  BEGIN  TRANSACTION
  SELECT @fieldID  = (SELECT f.fieldID fro m Fields F
  INNER JOIN INSERTED I ON I.fieldID = F.fieldID)
  EXEC ErosionSubfactors @fieldID
  COMMIT TRANSACTION
END
```

```
--*********************************************************************
--* PROCEDURE NAME :     SCIsoilDisturbanceRate Trigger              *
--* PURPOSE   :          Updates/Sets the Soil Disturbance Rate for   *
--*          particular operation type                                *
--* Date/Modified  :     11/12/2001 / 02/13/2001                      *
--*********************************************************************

CREATE TRIGGER SCIsoilDisturbanceRate ON dbo.OperationTypes
FOR INSERT, UPDATE
AS

BEGIN TRANSACTION

  UPDATE OperationTypes
  SET soilDisturbanceRate = sdrInvert + sdrMix +sdrLift +sdrShatter + sdrAerate
                          + sdrCompact

COMMIT TRANSACTION
```

```
--*****************************************************************
--* PROCEDURE NAME :     ResidueProducedCT                       *
--* PURPOSE     :    Calculates Residue Produced for crops in     *
--*                  the rotations stores Result                  *
--* Date/Modified  :     11/12/2001 / 02/13/2001                  *
--*****************************************************************

CREATE TRIGGER ResidueProducedCT ON dbo.CropTypes
FOR INSERT, UPDATE, DELETE
AS

IF UPDATE (residuePerUnit)
BEGIN
   BEGIN TRANSACTION
   UPDATE Crops
   SET Crops.residueProduced = (Crops.predictedYield *
                                CropTypes.residuePerUnit),

   Crops.residueReturned =  (Crops.predictedYield *
                             CropTypes.residuePerUnit)
                             + Crops.residueAddedRmvd
   FROM Croptypes
   INNER JOIN Crops ON  Crops.croptypeID = CropTypes.CropTypeID
   INNER JOIN INSERTED I ON I.cropTypeID = Crops.CropTypeID
   COMMIT TRANSACTION
END
```

Appendix B

The SCI Tables in the Current System.

The SCI system uses data stored in tables. Since these tables are not normalized it is difficult to search for data. We give portions of these tables to give the overall picture of the entire data in the system.

Table 1. Data for Cities

				Maintenance Amt. Including Roots		REV Conversion Factors: Small Grains except Pacific NW & Manure w/ bedding materials	Cotton, Sugarcane, Tobacco, & Peanuts	Corn, Grain Sorghum, Canola, Safflower & Sunflower	Forage grasses, cover, Manure - open lots & Pacific NW Small Grains	Legumes, Cabbage, & Broccoli	Soybeans, Field Beans, Sugar Beets, Cauliflower,& Strawberries	Vegetables, Specialty Crops & Manure-settling basin	Poultry litter
	CITY CODE	CITY	STATE	Undisturbed	Ref Con	Crop Group A	Crop Group B	Crop Group C	Crop Group D	Crop Group E	Crop Group F	Crop Group G	Crop Group H
1	1001	BIRMINGHAM, AL	AL	2972	5943	1.19	1.01	1.00	0.98	0.98	0.97	0.97	0.96
2	1002	MOBILE, AL	AL	3026	6053	1.14	1.01	1.00	0.99	0.99	0.98	0.98	0.98
3	1003	MONTGOMERY, AL	AL	2480	5960	1.18	1.01	1.00	0.98	0.98	0.97	0.97	0.97
4	2150	BIG DELTA, AK	AK	1828	3652	1.64	1.04	1.00	0.90	0.73	0.79	0.94	0.64
5	2151	BIG DELTA IRR., AK	AK	2012	4024	1.59	1.04	1.00	0.91	0.80	0.81	0.73	0.68
6	2340	FAIRBANKS WSO, AK	AK	1524	3047	1.71	1.05	1.00	0.89	0.62	0.75	0.76	0.57
7	2341	FAIRBANKS IRR, AK	AK	2097	4194	1.57	1.04	1.00	0.92	0.83	0.82	0.68	0.70
8	2430	HOMER WSO, AK	AK	1802	3605	1.65	1.04	1.00	0.90	0.72	0.78	0.77	0.63
9	2490	KENAI, AK	AK	1750	3501	1.66	1.04	1.00	0.90	0.70	0.78	0.72	0.62
10	2670	OLD EDGERTON, AK	AK	1519	3037	1.71	1.05	1.00	0.89	0.62	0.75	0.72	0.57
11	2680	PALMER AAES, AK	AK	1807	3613	1.64	1.04	1.00	0.90	0.72	0.78	0.68	0.63
12	2681	PALMER IRR, AK	AK	2093	4197	1.57	1.04	1.00	0.92	0.82	0.82	0.72	0.70
13	2830	TALKEETNA WSCMO, AK	AK	2048	4097	1.58	1.04	1.00	0.91	0.81	0.81	0.77	0.69
14	3001	FLAGSTAFF, AZ	AZ	2114	4227	1.56	1.04	1.00	0.92	0.90	0.82	0.76	0.71
15	3002	PHOENIX, AZ	AZ	1502	3004	1.72	1.05	1.00	0.89	0.87	0.75	0.77	0.56
16	3003	YUMA, AZ	AZ	679	1359	1.88	1.06	1.00	0.86	0.87	0.68	0.68	0.42
17	3005	WILLCOX, AZ	AZ	2063	4127	1.58	1.04	1.00	0.91	0.90	0.81	0.80	0.69
18	3007	PEARCE, AZ	AZ	2084	4168	1.57	1.04	1.00	0.92	0.90	0.82	0.77	0.70
19	3097	BOWIE IRR, AZ	AZ	2522	5044	1.43	1.03	1.00	0.94	0.95	0.88	0.85	0.82
20	3129	CASA GRANDE IRR, AZ	AZ	2610	5221	1.39	1.02	1.00	0.95	0.98	0.89	0.87	0.85

Table 2. Crop Data

CROP CODE #	CROP NAME	HARVEST UNITS	YIELD	POUNDS PER UNIT	RESIDUE : YIELD RATIO	ABOVE GROUND RESIDUE LBS	SURFACE DECOMP. COEFF.	SUB-SURFACE DECOMP. COEFF.	ROOTS IN TOP 4" (lbs)	ROOT-MASS ADJUST-MENT	Crop Group
1	agroforestry	lbs	20000	1	1	20000	0.015	0.015	2350	1.118	B
2	alf; fall seed	tons	1.5	2000	0.15	450	0.02	0.02	1300	3.889	E
3	alf; spring seed	tons	1.5	2000	0.15	450	0.0200	0.0200	2500	6.556	E
4	alf; summer seed	tons	1.5	2000	0.15	450	0.0200	0.0200	1300	3.889	E
5	alf; y1 reg(spr seed	tons	1.5	2000	0.15	450	0.0200	0.0200	2300	6.111	E
6	alf; y1 reg(sum seed	tons	1.5	2000	0.15	450	0.0200	0.0200	2100	5.667	E
7	alf; y1 sen (oat sil	tons	0.5	2000	1	1000	0.0200	0.0200	2250	3.250	E
8	alf; y1 sen(spr seed	tons	1.5	2000	0.15	450	0.0200	0.0200	2500	6.556	E
9	alf; y1 sen(sum seed	tons	0.1	2000	1	200	0.0200	0.0200	2500	13.500	E
10	alf; y1 senesc (oat)	tons	0.5	2000	1	1000	0.0200	0.0200	2000	3.000	E
11	alf; y2 regrowth	tons	1.5	2000	0.15	450	0.0200	0.0200	3000	7.667	E
12	alf; y2 regrowth 3T	tons	1	2000	0.15	300	0.0200	0.0200	2000	7.667	E
13	alf; y2 senescence	tons	0.15	2000	1	300	0.0200	0.0200	3500	12.667	E
14	alf; y3 regrowth	tons	1.75	2000	0.15	525	0.0200	0.0200	3500	7.667	E
15	alf; y3 regrowth 3T	tons	1	2000	0.15	300	0.0200	0.0200	2300	8.667	E
16	alf; y3 senescence	tons	1.75	2000	0.15	525	0.0200	0.0200	3500	7.667	E
17	alfalfa 2nd year	tons	1.75	2000	0.15	525	0.0200	0.0200	3000	6.714	E
18	alfalfa established	tons	1.75	2000	0.15	525	0.0200	0.0200	3500	7.667	E
19	alfalfa seeding year	tons	1.5	2000	0.15	450	0.0200	0.0200	2500	6.556	E
20	alfalfa summer seed	tons	1.5	2000	0.15	450	0.0200	0.0200	1300	3.889	E
21	alfalfa; established	tons	2	2000	0.15	600	0.02	0.02	3850	7.417	E
22	alfalfa; fall seed	tons	1.5	2000	0.15	450	0.0200	0.0200	650	2.444	E
23	alfalfa; spring seed	tons	1.5	2000	0.15	450	0.0200	0.0200	2600	6.778	E
24	alfalfa; summer seed	tons	1.5	2000	0.15	450	0.0200	0.0200	1300	3.889	E
25	alfalfa-brome 2nd yr	tons	1.75	2000	0.15	525	0.0190	0.0190	4300	9.190	E
26	alfalfa-brome 2y rgs	tons	0.15	2000	1	300	0.0190	0.0190	4300	15.333	E
27	alfalfa-brome est se	tons	0.15	2000	1	300	0.0180	0.0180	4900	17.333	D
28	alfalfa-brome estab.	tons	1.75	2000	0.15	525	0.0180	0.0180	4900	10.333	D

Table 3 Soil Disturbances by Field Operations.

OPERATION NUMBER	FIELD OPERATIONS	SOIL DISTURBING ACTIONS						SOIL DISTURBANCE RATING
		INVERT	MIX	LIFT	SHATTER	AERATE	COMPACT	
1	Aerator, ground driven knife aerator	1	1	2	3	4	1	12
2	Anhydous applicator w/ Knife and w/coulter	1	2	1	2	1	1	8
3	Anhydrous applicator w/Knife (wide)	2	3	2	3	2	1	13
4	Baler, forage harvester	0	0	0	0	0	3	3
5	Bed/Lister/Hill (wide beds)	5	5	5	5	5	4	29
6	Bedder, lister, hipper single row	4	4	3	5	5	2	23
7	Bury drip irrigation line	1	2	1	2	1	1	8
8	Chisel Plow, deep chisel, straight point	3	4	4	4	5	2	22
9	Chisel Plow, deep chisel, twisted point	4	4	5	5	5	2	25
10	Chisel Plow, sweeps	2	3	5	4	4	3	21
11	Chisel/sweep-rod; first oper. after MB plow	2	4	5	4	4	4	23
12	Chisel; straight points (12" spacing)	2	4	4	4	5	2	21
13	Chisel; straight points (18" spacing)	2	3	4	4	4	2	19
14	Chisel; straight points (24" spacing)	2	2	4	4	3	2	17
15	Chisel; straight points 2.0 (rough)	2	3	4	4	4	2	19
16	Chisel; straight points 2.5 (very rough)	2	2	4	4	3	2	17
17	Chisel; twisted points (18" spacing)	3	4	5	5	5	2	24
18	Chisel; twisted points (24" spacing)	3	3	5	5	4	2	22
19	Chisel; twisted points following chop stubble	3	4	5	5	5	2	24
20	Chisel-disk; straight points	3	4	4	4	5	3	23
21	Chisel-disk; twisted points	4	5	5	5	5	3	27
22	Combo Up-rooter/bedder (cotton)	4	4	2	5	5	4	24
23	Corrigation/Furrow maker	3	3	3	4	3	2	18
24	Cultipacker roller	2	3	2	5	3	4	19
25	Cultivator, field, straight point	3	3	3	4	3	2	18
26	Cultivator, ridge till w/ridging attach.	4	4	3	5	5	2	23
27	Cultivator, row w/ Rotary finger wheels	2	1	1	3	3	1	11
28	Cultivator, row w/multiple sweeps	3	2	2	5	5	2	19
29	Cultivator, row w/single sweeps	3	2	2	4	4	1	16
30	Cultivator, row w/Spring tooth shovels	3	2	2	5	5	1	18

Table 4

SDR/FO Subfactor Conversion

Table 4	
SDR	**FO**
0	1.00
1	0.99
2	0.98
3	0.97
4	0.96
5	0.95
6	0.94
7	0.93
8	0.92
9	0.91
10	0.90
11	0.89
12	0.88
13	0.87
14	0.86
15	0.85
16	0.84
17	0.83
18	0.82
19	0.81
20	0.80
21	0.79
22	0.78
23	0.77
24	0.76
25	0.75
26	0.74
27	0.73

Table 5

T/ER Erosion Subfactor Conversion

Table 5	
Rate of Erosion	**ER Subfactor**
0.00	1.00
0.25	0.94
0.50	0.88
0.75	0.81
1.00	0.75
1.25	0.69
1.50	0.63
1.75	0.56
2.00	0.50
2.25	0.44
2.50	0.38
2.75	0.31
3.00	0.25
3.25	0.19
3.50	0.13
3.75	0.06
4.00	0.00
4.25	-0.05
4.50	-0.10
4.75	-0.15
5.00	-0.20
5.25	-0.25
5.50	-0.30
5.75	-0.35
6.00	-0.40
6.25	-0.45
6.50	-0.50