APPENDIX D
MICROSOFT ACCESS PROGRAM
FOR ASSESSING IMPROVED PROPERTY IN LEE COUNTY, FLORIDA

Introduction

The data and algorithms used to calculate a risk-based special assessment for local emergency management costs associated with hurricanes in Lee County, Florida, are contained in two linked Microsoft Access files: db1.mdb and db2.mdb. The data, obtained from the Lee County Property Appraiser's Office, are contained in db1.mdb. The queries and tables used to calculate the special assessment for each property parcel under the assumptions used in the project, are contained in db1.mdb. The files must be installed in a directory on the c: drive with the name “JMacDonald” for the links to operate.

The following are detailed instructions for some of the more enigmatic portions of the special assessment algorithm. Though some of these tasks may repeat themselves in the algorithm, one set of explanations is presented as an example. In addition to syntax techniques, information also is provided about the logic behind the protocol.

Variables

The following variables are used in the algorithm's tables and queries. The definitions here match those given under the "DESCRIPTION" sections in appropriate tables. In some rare cases, a variable is only included in a query, where there is no "DESCRIPTION" section. The next section identifies the tables and queries where each variable is located.

Each of the following variables either comes directly from the Lee County Property Appraiser's Office database or is derived from that database. The distinction is made in the first table, ParcelData, where all variables come directly from Lee County; the table, StormCat, where variables come from the Florida Planning and Development Laboratory at Florida State University, and EvacZone, where variables come from the Southwest Florida Emergency Management Office. Each subsequent table and query which contains new variables derived from those provided follows ParcelData2 in the description of tables and queries after the list of variables. Each table or query entry includes a list of variables in that particular table and query, and a description of how new variables are derived. The definitions listed in this section are intended as a reference so that when the reader comes across a variable in a table or query entry, he or she can check its definition to fully understand its role in the Access database.

ADJUST DAMAGE SUM:	Tests whether DAMAGE SUM equals or exceeds value of ASSESSED. If true, then ADJUST DAMAGE is set equal to ASSESSED; if false, ADJUST VALUE is set equal to DAMAGE SUM

ANNUAL DAMAGE:	The product of STRIKE ANNUAL PROB and ADJUST DAMAGE yields the damage expected during a typical year for the parcel in question (currency integer)
ANTICI PROTECT INDEX:	The ratio of a parcel's PE to the sum of all parcel PE values in Lee County (5-decimal non-integer)
APPLY DAMAGE CURVE:	Describes lookup source based on MAX WAVE HGT (text)
ASSESSED:	The assessor's appraisal of the improvement (whole-number currency)
ASSESSRATIO:	The ratio of a parcel's total assessed value to the sum of all assessed values in Lee County (8-decimal non-integer)
BFE:	Elevation of flood level referenced to NGVD (sea-level) in feet (whole number)
CAT 1 SURGE LEVEL:	Surge level expected over parcel in Category 1 storm referenced to NGVD (sea-level), in feet (2-decimal non-integer)
CAT 2 SURGE LEVEL:	Surge level expected over parcel in Category 2 storm referenced to NGVD (sea-level), in feet (2-decimal non-integer)
CAT 3 SURGE LEVEL:	Surge level expected over parcel in Category 3 storm referenced to NGVD (sea-level), in feet (2-decimal non-integer)
CAT 45 SURGE LEVEL:	Surge level expected over parcel in Category 4 or 5 storm referenced to NGVD (sea-level), in feet (2-decimal non-integer)
CATEGORY:	The category of a storm, according to Saffir-Simpson scale (intensity) (integer)
COAST DISTANCE NAUT:	Distance from parcel reference point to coast in nautical miles (2-4 decimal non-integer, where # decimal places dependent on need)
COAST DISTANCE STAT:	Distance from parcel reference point to coast in statute miles (2-decimal non-integer)

COMPOSITE RISK INDEX:	The average value of DAMAGE RISK INDEX, PUBLIC FACILITY RISK INDEX, and ANTICI PROTECT INDEX (5-8 decimal non-integer, where # decimal places dependent on need)
CURRENT COST:	Estimate of a property's current tax, based on the inclusion of the "All Hazards Tax" currently imposed on unincorporated parcels (2-decimal, non-integer currency)
DAMAGE RISK INDEX:	The ratio of a parcel's TOTAL ANNUAL DAMAGE to the sum of all parcel TOTAL ANNUAL DAMAGE values in Lee County (5-decimal non-integer)
DAMAGE SUM:	The sum of WAVEWATER DAMAGE and WIND DAMAGE (currency integer)
EV:	Mean elevation of improvement referenced to NGVD (sea-level) in feet (whole number)
EVAC ZONE:	Evacuation zone in which parcel is located (text)
EXPECTED DAMAGE:	Damage expected from the wave/surge scenario and the property's assessed value (wave/surge scenario replaced with "maximum peak gust scenario" for use in the wind algorithm) (whole-number currency)
FULLASSESSED	Sum of ASSESSED and LANDASSESSED (whole number currency)
HIGH ANTICI COST SHARE:	High estimate of a property owner's financial obligation for anticipatory protective measures cost (whole-number currency)
HIGH ANTICIPATE COST:	High estimate of county anticipatory protective measures cost (Cat B) (whole-number currency)
HIGH CONTINUE COST:	High estimate of county annual continuing costs (CC) (whole-number currency)
HIGH CONTINUE COST SHARE:	High estimate of a property owner's financial obligation for annual continuing cost (whole-number currency)

HIGH CURRENT COST:	High estimate of a property's value-based tax (whole-number currency)
HIGH DAMAGE COST:	High estimate of county public infrastructure damage cost (Cat C-G) (whole-number currency)
HIGH DEBRIS COST SHARE:	High estimate of a property owner's financial obligation for debris removal cost (whole-number currency)
HIGH DEBRIS REMOVAL COST:	High estimate of county debris removal cost (Cat A) (whole-number currency)
HIGH PUBLIC COST SHARE:	High estimate of a property owner's financial obligation for public damage cost (whole-number currency)
HIGH TAX ESTIMATE:	High estimate of a property owner's risk-based tax (2-decimal, non-integer currency)
INTER PEAK GUST:	The integer obtained when PEAK GUST is divided by 5 (integer)
LANDASSESSED:	The assessor's appraisal of the land (whole-number currency)
LOW ANTICI COST SHARE:	Low estimate of a property owner's financial obligation for anticipatory protective measures cost (whole-number currency)
LOW ANTICIPATE COST:	Low estimate of county anticipatory protective measures cost (Cat B) (whole-number currency)
LOW CONTINUE COST:	Low estimate of county annual continuing costs (CC) (whole-number currency)
LOW CONTINUE COST SHARE:	Low estimate of a property owner's financial obligation for annual continuing cost (whole-number currency)
LOW CURRENT COST:	Low estimate of a property's value-based tax (whole-number currency)
LOW DAMAGE COST:	Low estimate of county public infrastructure damage cost (Cat C-G) (whole-number currency)

LOW DEBRIS COST SHARE:	Low estimate of a property owner's financial obligation for debris removal cost (whole-number currency)
LOW DEBRIS REMOVAL COST:	Low estimate of county debris removal cost (Cat A) (whole-number currency)
LOW PUBLIC COST SHARE:	Low estimate of a property owner's financial obligation for public damage cost (whole-number currency)
LOW TAX ESTIMATE:	Low estimate of a property owner's value-based tax (2-decimal, non-integer currency)
MAX WAVE HGT:	Maximum height of wave dependent on still-water surge depth, based on 1990 study by the Army Corps of Engineers (2-digit non-integer)
MSSW MIDDLE:	The "middle" value of the MSSW range of values associated with the relevant storm category (whole number)
MSSW RANGE:	The range of maximum sustained wind speeds associated with the relevant storm category (text)
NAUT1:	Dummy variable used to test whether a parcel is gulf ward of the synthetic coastline, by its township and range values in the parcel identification number. If the values match the test values in this variable's list, COAST DISTANCE NAUT=0, otherwise COAST DISTANCE NAUT= COAST DISTANCE STAT/6076.1 (numeric)
NAUT2:	Dummy variable used to test whether a parcel is gulf ward of the synthetic coastline, by its township and range values in the parcel identification number. If the values match the test values in this variable's list, COAST DISTANCE NAUT=0, otherwise COAST DISTANCE NAUT=NAUT1 (numeric)

NAUT3:	Dummy variable used to test whether a parcel is gulf ward of the synthetic coastline by its township and range values in the parcel identification number. If the values match the test values in this variable's list,COAST DISTANCE NAUT=0, otherwise COAST DISTANCE NAUT=NAUT2 (numeric)
NAUT4:	Dummy variable used to test whether a parcel is gulf ward of the synthetic coastline by its township and range values in the parcel identification number. If the values match the test values in this variable's list, COAST DISTANCE NAUT=0, otherwise COAST DISTANCE NAUT=NAUT3 (numeric)
NAUT5:	Dummy variable used to test whether a parcel is gulf ward of the synthetic coastline by its township and range values in the parcel identification number. If the values match the test values in this variable's list, COAST DISTANCE NAUT=0, otherwise COAST DISTANCE NAUT=NAUT4 (numeric)
NUMBER FLOORS:	Number of levels in improved structure (text)
PARCEL ID:	Coded reference unique to all parcels in Lee County (text)
PE:	The probability of evacuation in that particular category plus probabilities from all lower categories (2-decimal non-integer)
PEAK GUST:	Simple algorithm to calculate maximum wind gust based on maximum sustained wind speed at site (2-decimal non-integer)
PERCENT DAMAGE:	Percent damage based on the damage lookup table values and wave-plus-water depth through WHITEWATER ("wave-plus-water depth through WHITEWATER" replaced with "maximum wind gust through ROUND PEAK GUST" for use in the wind algorithm) (percent)
PUBLIC FACIL USE:	The product of SQFT and ASSESSED (whole number)

PUBLIC FACILITY RISK INDEX:	The ratio of a parcel's PUBLIC FACIL USE to the sum of all parcel PUBLIC FACIL USE values in Lee County (5-decimal non-integer)
ROUND PEAK GUST:	The integer product of INTER PEAK GUST and 5 (integer)
SEV:	Elevation of first floor of structure referenced to NGVD (sea-level) (2-decimal non-integer)
SH BENCHMARK:	Benchmark reference code for surge levels of various storm categories (integer)
SITE WIND:	Completes the WINDY test. Tests whether COAST DISTANCE NAUT is 0, defers to MSSW MIDDLE if true and WINDY if false (2-decimal non-integer)
SQFT:	Size of interior floor area in square feet
STRIKE ANNUAL PROB:	The probability a storm of a particular category will affect the parcel in question (percent)
STRUCTURE TYPE:	Type of improvement (text)
TOTAL ANNUAL DMG:	The sum of ANNUAL DAMAGE values from all tables in the AnnualTotal series (2-decimal currency)
TOTAL FULLASSESSED	The sum of FULLASSESSED values for all property parcels (2-decimal currency)
TOTAL WAVEWATER LEV:	Elevation of the water surface (referenced to NGVD, sea-level up to surge-level) (2-digit non-integer)
WATER DEPTH:	Water depth at parcel site in a particular category storm, equals the difference between site elevation and particular category surge level (2-digit non-integer)
WAVE HGT ABOVE SURGE:	Determined by a multiplier of 0.656 on WATER DEPTH (Corps) (2-digit non-integer)
WAVEWATER DAMAGE:	The value of EXPECTED DAMAGE taken from the SubQuerySW table series (currency integer)

WAVEWATER DEPTH:	Depth measured from the first floor of the structure on-site to the water surface (2-digit non-integer)
WHITEWATER:	Has a value of "99" if WAVEWATER DEPTH is negative. Otherwise, it assumes the value of WAVEWATER DEPTH (whole number)
WIND DAMAGE:	The value of EXPECTED DAMAGE taken from the SubQWindTable table series (currency integer)
WINDY:	Tests whether COAST DISTANCE NAUT is between 0 and 10 inclusive, defers to algorithm for parcels 0 to 10 nautical miles from coast if true, and to algorithm for parcels greater than 10 miles from the coast if false (2-decimal non-integer)
YEAR BUILT:	Year construction began (4-digit integer)

Tables and Queries

This section identifies both tables (T) and queries (Q) in the algorithm. Each table or query name is followed by a description of the table or query, a list of variables included, a list of queries where the table or query may be found, the required relationships between tables and queries if it is a query, and the primary key variable(s) if it is a table. The opening sections describe how input data tables and queries are constructed.

Construction of Input Data Tables

All input data are included in db2.mdb in manually-created tables. These property data must be updated regularly to reflect changes in the use of property and the structures present on individual property parcels. Any variable whose value must be calculated is considered output data; it is reported in output tables produced by queries - see next section.

Setup

To create an input data table such as those included in db2.mdb, click on the "Table" tab at the top of the database window, then click "New." In the new table window, click design view. This is where field names and field properties are created for the table.

Once inside the design view, tables are fairly self-explanatory. Clicking in any space allows the user to type there. The choice of Data Type for all variables should be either Text or Number. The Description of the field is the definition to help the user understand its purpose more accurately. Format is designed so the user interprets the

data correctly. For example, monetary values must be specified as Single or Double numbers under the Field Size option, Currency under the format option, and 2 decimal places under that option. There is room for personal preference, however.

Primary Keys

One of the most important features of the table design view is the Primary Key function. Once a field is designated a primary key, it can no longer contain duplicate or null (blank) values in the table. The primary key is the linking variable through which all connections to outside tables and queries are made. One cannot create the single, double, or triple joins required in this database without primary keys. Every table should have one so that it can be connected with fundamental parcel data or other important references. To designate a field as a primary key, simply highlight the field of interest, then select the "key" icon in the toolbar at the top of the screen. To designate more than one primary key in a table, highlight more than one field and press the same "key" icon just once. It may be easier to fill tables with data before this process, though, since no duplicate or null values are permitted for the primary key field.

Construction of Queries

Setup

The first step in designing a query is to consider the query's purpose. Identify what the query to accomplish, then determine what data sources and expressions will help get the job done. To create a query, click on the "Query" tab at the top of the database window, then click "New." In the new query window, click design view. This is where field names and field properties are created for the query.

New Fields

To enter fields in a query, first click on the "Show Table/Query" button located at the top of the screen in the toolbar. The button is identified by a "+" sign in the upper left corner and a table/query icon in the lower right corner. Then select tables and/or queries which contain fields to be included in the query, through the "Add" button. When finished, close the "Show Table/Query" window. To add fields from the selected tables and/or queries (now displayed in the upper half of the design view window), simply click on the field name under the table/query of choice and drag it to an empty field space in the bottom half of the design view window.

There are many spaces for information under each field. The name of the table from which the field name was dragged appears below the "Field: row space in the "Table: row space. The "Sort: row space allows the user to arrange the results of the query in order according to the menu obtained in the space. The "Show: row space allows the user to use fields in the query without displaying them in datasheet view. The "Criteria:row space allows one to filter data in that particular field. For example, one of the fields in a particular query is "Category." In that query, data are only needed for Category 1 storms on the Saffir-Simpson scale. The number "1" is typed in the "Criteria"

space to tell Access to only list records of parcel information which pertain to that particular storm scenario.

Make-Table Queries

Queries are the means by which output values are calculated in the algorithm. Each query is set up uniquely for the particular task it must perform. Queries use input from input data tables and from other queries.

To "pool" information together before it is sorted, filtered, or manipulated, it is necessary to join tables and queries with one another. This is accomplished simply by dragging the primary key of a table to the query with which it is to be connected (the destination query's field does not have to be specified as a primary key). Now information from both sources is linked together, and may be treated as a single data source.

In this Access data base, all query output is written to tables rather than being directed as input to other queries. This is done because the amount of RAM required to run a query increases dramatically if the query requires input data from other queries. This problem occurs because each import query must recalculate the required output. A chain of queries can easily deplete the RAM necessary to run the application. Because of the complexity of this application, all output from each query is sent to a table so that subsequent queries can import data without further calculation.

Use of the Expression Builder

Some fields are not taken from existing data sources; they must be generated. Expression Builder is the mode to create new variables from input data. To open Expression Builder, first enter a field name in the "Field: space. Then, right click on the "Field: space and select "Build" from the menu. This opens the Expression Builder with which an expression can be created to define the new field.

The Expression Builder is useful for simple arithmetic expressions, logic statements, and even complex calculations. To import existing fields into the expression, simply click on the appropriate folder in the menu in the bottom left screen, and select the field from the subsequent options. Expressions are easy to assemble as fields (variables), functions, and operators are added according to need. However, the Expression Builder is limited in scope, so some innovation may be necessary to accomplish specific objectives. Below are several "examples" of situations in which the Expression Builder is useful, and how to avoid potential pitfalls.

(a) Expressions with fields from the existing query

To use a field in the existing query, click on that query's folder in the lower left screen of Expression Builder. The field will not show, however, if the query has not been saved since the field's creation. Simply save the query and return to Expression Builder. Note, though, that the query cannot be saved if one leaves Expression Builder with a field name in the "Field: space but with

no expression. In this situation, one must first delete the name, then save the query, re-enter the field name, and build the expression.

(b) Logical expressions

The function for "If" statements is found in the Functions folder, under "Built-In Functions. "IIf" is the function required. If one clicks on this function, a model of the function is displayed at the bottom of the Expression Builder window. The "expr" is the condition to test, the "truepart" is the value of the field if "expr" is true, and the "falsepart" is the value of the field if "expr" is false. Enter the expression to be tested, followed by the appropriate values for the field if the statement is true or false.

(c) Value format

Some fields calculated in Expression Builder must be expressed as integers, others as non-integers. According to need, there are functions in Expression Builder to accomplish these objectives. These functions are found in the same place as other functions (see previous paragraph). They are the first function in the equation and are outside the set of parentheses which encompasses the rest of the equation. CInt and CSng are the most common in this algorithm. They convert values to integers and single non-integers respectively.

Data Base Details

Required Relationships

A required relationship is a link between identical variables in two different queries or tables. If there is a table involved, at least one of the linked variables must be the primary key for its respective table to make the relationship. A link is in place when a solid black line connects the two variables in the upper half of the Design Query window. In the following descriptions, the two queries and/or tables that comprise a required relationship are joined by a link. The variable in parentheses is the primary key which makes this link possible.

Tables in Data Base db2.mdb

ParcelData2(T)

Description: This table includes the basic information on each of 146,672 property parcels in the county. All the variables contain input data.

Variables: PARCEL ID, STRUCTURE TYPE, NUMBER FLOORS, YEAR BUILT, EV, SQFT, ASSESSED, SH BENCHMARK, BFE, COAST DISTANCE STAT, COAST DISTANCE NAUT, EVAC ZONE

Primary Key(s): PARCEL ID

Included in the Following Queries: All - this table is the primary reference for this entire Access database.

StormCat(T)

Description: This is a lookup table which describes the appropriate wind parameters and strike and evacuation probabilities for each Saffir-Simpson hurricane category.

Variables: CATEGORY, STRIKE ANNUAL PROB, MSSW RANGE, MSSW MIDDLE

Primary Key(s): CATEGORY

Included in the Following Queries: QWind1, QWind2, QWind3, QWind45, QannualTotal1, QAnnualTotal2, QAnnualTotal3, QAnnualTotal45

EvacZone(T)

Description: This table provides the cumulative evacuation probabilities for each hurricane category.

Variables: EVAC ZONE, PE

Primary Key(s): EVAC ZONE

Included in the Following Queries: APMRI, APMRI2*Included in the Following Queries:* APMRI, APMRI2

SurgeDam(T)

Description: A lookup table for storm surge damage. If the maximum wave height is less than 3 feet, then this table is used to determine PERCENT DAMAGE. Each PERCENT DAMAGE value corresponds to a unique combination of STRUCTURE TYPE, NUMBER FLOORS, and WHITEWATER values.

Variables: STRUCTURE TYPE, NUMBER FLOORS, WHITEWATER, PERCENT DAMAGE

Primary Key(s): STRUCTURE TYPE, NUMBER FLOORS, WHITEWATER

Included in the Following Queries: SubQSurgeWave1, SubQSurgeWave2, SubQSurgeWave3, SubQSurgeWave45

WaveDam(T)

Description: A lookup table for storm surge and wave damage. If the maximum wave height is greater than or equal to 3 feet, then this table is used to determine PERCENT DAMAGE. Each PERCENT DAMAGE value corresponds to a unique combination of STRUCTURE TYPE, NUMBER FLOORS, and WHITEWATER values.

Variables: STRUCTURE TYPE, NUMBER FLOORS, WHITEWATER, PERCENT DAMAGE

Primary Key(s): STRUCTURE TYPE, NUMBER FLOORS, WHITEWATER

Included in the Following Queries: SubQSurgeWave1, SubQSurgeWave2, SubQSurgeWave3, SubQSurgeWave45

WindDam(T)

Description: A lookup table for wind damage. Each PERCENT DAMAGE value corresponds to a unique combination of STRUCTURE TYPE and ROUND PEAK GUST values.

Variables: STRUCTURE TYPE, ROUND PEAK GUST, PERCENT DAMAGE

Primary Key(s): STRUCTURE TYPE, ROUND PEAK GUST

Included in the Following Queries: SubQWind1, SubQWind2, SubQWind3, SubQWind4 Tables in Data Base db1.mdbTables in Data Base db1.mdb

SurgeBench1(T)

Description: This table lists each surge height benchmark reference and the surge level expected on that parcel for a Category 1 storm.

Variables: SH BENCHMARK, CAT 1 SURGE LEVEL

Primary Key Link(s): SH BENCHMARK

Included in the Following Queries: QSurgeWave1

SurgeBench2(T)

Description: This table lists each surge height benchmark reference and the surge level expected on that parcel for a Category 2 storm.

Variables: SH BENCHMARK, CAT 2 SURGE LEVEL

Primary Key Link(s): SH BENCHMARK

Included in the Following Queries: QSurgeWave2

SurgeBench3(T)

Description: This table lists each surge height benchmark reference and the surge level expected on that parcel for a Category 3 storm.

Variables: SH BENCHMARK, CAT 3 SURGE LEVEL

Primary Key Link(s): SH BENCHMARK
Included in the Following Queries: QSurgeWave3

SurgeBench45(T)

Description: This table lists each surge height benchmark and the surge level expected on that parcel for a Category 4 or 5 storm.

Variables: SH BENCHMARK, CAT 45 SURGE LEVEL

Primary Key Link(s): SH BENCHMARK

Included in the Following Queries: QSurgeWave45

QSurgeWave1Table(T)

Description: This tables lists all variables used in QSurgeWave1(Q). The MAX WAVE HGT and WAVEWATER values from this table series are part of the SubQSurgeWave query series.

Variables: PARCEL ID, STRUCTURE TYPE, SH BENCHMARK, NUMBER FLOORS, CAT 1 SURGE LEVEL, SEV, WATER DEPTH, MAX WAVE HGT, APPLY DAMAGE CURVE, WAVE HGT ABOVE SURGE, TOTAL WAVEWATER LEV, WAVEWATER DEPTH, WHITEWATER

Primary Key Link(s): PARCEL ID

Included in the Following Queries: SubQSurgeWave1

QSurgeWave2Table(T)

Description: This tables lists all variables used in QSurgeWave2(Q). The MAX WAVE HGT and WAVEWATER values from this table series are part of the SubQSurgeWave query series.

Variables: PARCEL ID, STRUCTURE TYPE, SH BENCHMARK, NUMBER FLOORS, CAT 2 SURGE LEVEL, SEV, WATER DEPTH, MAX WAVE HGT, APPLY DAMAGE CURVE, WAVE HGT ABOVE SURGE, TOTAL WAVEWATER LEV, WAVEWATER DEPTH, WHITEWATER

Primary Key Link(s): PARCEL ID

Included in the Following Queries: SubQSurgeWave2

QSurgeWave3Table(T)

Description: This tables lists all variables used in QSurgeWave3(Q). The MAX WAVE HGT and WAVEWATER values from this table series are part of the SubQSurgeWave query series.

Variables: PARCEL ID, STRUCTURE TYPE, SH BENCHMARK, NUMBER FLOORS, CAT 3 SURGE LEVEL, SEV, WATER DEPTH, MAX WAVE HGT, APPLY DAMAGE CURVE, WAVE HGT ABOVE SURGE, TOTAL WAVEWATER LEV, WAVEWATER DEPTH, WHITEWATER

Primary Key Link(s): PARCEL ID

Included in the Following Queries: SubQSurgeWave3

QSurgeWave45Table(T)

Description: This tables lists all variables used in QSurgeWave45(Q). The MAX WAVE HGT and WAVEWATER values from this table series are part of the SubQSurgeWave query series.

Variables: PARCEL ID, STRUCTURE TYPE, SH BENCHMARK, NUMBER FLOORS, CAT 45 SURGE LEVEL, SEV, WATER DEPTH, MAX WAVE HGT, APPLY DAMAGE CURVE, WAVE HGT ABOVE SURGE, TOTAL WAVEWATER LEV, WAVEWATER DEPTH, WHITEWATER

Primary Key Link(s): PARCEL ID

Included in the Following Queries: SubQSurgeWave45

SubQuerySW1(T)

Description: This table displays the output from the subquery, SubQSurgeWave1.

Variables: See query SubQSurgeWave1

Primary Key Link(s): PARCEL ID

Included in the Following Queries: QAnnualTotal1

SubQuerySW2(T)

Description: This table displays the output from the subquery, SubQSurgeWave2.

Variables: See query SubQSurgeWave2

Primary Key Link(s): PARCEL ID

Included in the Following Queries: QAnnualTotal2

SubQuerySW3(T)

Description: This table displays the output from the subquery, SubQSurgeWave3.

Variables: See query SubQSurgeWave3

Primary Key Link(s): PARCEL ID

Included in the Following Queries: QAnnualTotal3

SubQuerySW45(T)

Description: This table displays the output from the subquery, SubQSurgeWave45.

Variables: See query SubQSurgeWave45

Primary Key Link(s): PARCEL ID

Included in the Following Queries: QAnnualTotal45

QWind1Table(T)

Description: The purpose of this table is to store the integer value of INTER PEAK GUST from the query QWind1. This is necessary since the "CInt" function in the Expression Builder does not erase the information to the right of the decimal point. Rather, it affects the display of the number only. Therefore, INTER PEAK GUST can only be used to calculate an integer value for ROUND PEAK GUST if it is imported as an integer from a table. This is done in the query InterQWind1.

Variables: See QWind1(Q)

Primary Key(s): PARCEL ID

Included in the Following Queries: InterQWind1

QWind2Table(T)

Description: The purpose of this table is to store the integer value of INTER PEAK GUST from the query QWind2.

Variables: See QWind2(Q)

Primary Key(s): PARCEL ID

Included in the Following Queries: InterQWind2

QWind3Table(T)

Description: The purpose of this table is to store the integer value of INTER PEAK GUST from the query QWind3.

Variables: See QWind3(Q)

Primary Key(s): PARCEL ID

Included in the Following Queries: InterQWind3

QWind45Table(T)

Description: The purpose of this table is to store the integer value of INTER PEAK GUST from the query QWind45.

Variables: The same as for QWind45(Q)

Primary Key(s): PARCEL ID

Included in the Following Queries: InterQWind45

InterQWind1Table(T)

Description: This table serves as a stop after ROUND PEAK GUST is calculated in QSurgeWave1(Q).

Variables: See InterQWind1(Q)

Primary Key(s): PARCEL ID

Included in the Following Queries: SubQWind1(Q)

InterQWind2Table(T)

Discussion: This table serves as a stop after ROUND PEAK GUST is calculated in QSurgeWave2(Q).

Variables: See InterQWind2(Q)

Primary Key(s): PARCEL ID

Included in the Following Queries: SubQWind2(Q)

InterQWind3Table(T)

Discussion: This table serves as a stop after ROUND PEAK GUST is calculated in QSurgeWave3(Q).

Variables: See InterQWind3(Q)

Primary Key(s): PARCEL ID

Included in the Following Queries: SubQWind3(Q)

InterQWind45Table(T)

Discussion: This table serves as a stop after ROUND PEAK GUST is calculated in QSurgeWave45(Q).

Variables: See InterQWind45(Q)

Primary Key(s): PARCEL ID

Included in the Following Queries: SubQWind45(Q)

SubQWind1Table(T)

Description: This table receives output from the subquery SubQWind1 and is used in subsequent queries for the algorithm.

Variables: See SubQWind1(Q)

Required Relationships: SubQWind1Table--ParcelData2 (PARCEL ID)

Included in the Following Queries: QAnnualTotal1

SubQWind2Table(T)

Description: This table receives output from the subquery SubQWind2 and is used in subsequent queries for further steps in the algorithm.

Variables: See SubQWind2(Q)

Required Relationships: SubQWind2Table--ParcelData2 (PARCEL ID)

Included in the Following Queries: QAnnualTotal2

SubQWind3Table(T)

Description: This table receives output from the subquery SubQWind3 and is used in subsequent queries for further steps in the algorithm.

Variables: See SubQWind3(Q)

Required Relationships: SubQWind3Table--ParcelData2 (PARCEL ID)

Included in the Following Queries: QAnnualTotal3

SubQWind45Table(T)

Description: This table receives output from the subquery SubQWind45 and is used in subsequent queries for further steps in the algorithm.

Variables: See SubQWind45(Q)

Required Relationships: SubQWind3Table--ParcelData2 (PARCEL ID)

Included in the Following Queries: QAnnualTotal45

AnnualTotal1(T)

Description: This table retrieves all the variables from QAnnualTable1 as yet another "stop" in the algorithm. The information from this table is passed on to the cost index queries.

Variables: See QAnnualTotal1(Q).

Primary Key(s): PARCEL ID

Included in the Following Queries: QAnnualTotal

AnnualTotal2(T)

Description: This table retrieves all the variables from QAnnualTotal2(Q) as yet another "stop" in the algorithm. The information from this table is passed on to the cost index queries.

Variables: See QAnnualTotal2(Q).

Primary Key(s): PARCEL ID

Included in the Following Queries: QAnnualTotal2

AnnualTotal3(T)

Description: This table retrieves all the variables from QAnnualTotal3(Q) as yet another "stop" in the algorithm. The information from this table is passed on to the cost index queries.

Variables: See QAnnualTotal3(Q).

Primary Key(s): PARCEL ID

Included in the Following Queries: QAnnualTotal3

AnnualTotal45(T)

Description: This table retrieves all the variables from QAnnualTotal45(Q) as yet another "stop" in the algorithm. The information from this table is passed on to the cost index queries.

Variables: See QAnnualTotal45(Q).

Primary Key(s): PARCEL ID

Included in the Following Queries: QAnnualTotal45

QDRI2(T)

Description: This table is another "stop" in the algorithm to preserve memory and to allow the user to check for errors.

Variables: See DRI2(Q)

Primary Key(s): PARCEL ID

Included in the Following Queries: CCI, TaxAllo

QAPMRI2(T)

Description: This table is another "stop" in the algorithm to preserve memory and to allow the user to check for errors.

Variables: See APMRI2 (Q)

Primary Key(s): PARCEL ID

Included in the Following Queries: CCI, TaxAllo

QPFRI2(T)

Description: This table is another "stop" in the algorithm to store results from the query PFRI2.

Variables: See query PFRI2

Primary Key(s): PARCEL ID

Included in the Following Queries: CCI, TaxAllo

QCCI(T)

Description: This table receives the output from the query CCI. It serves as a "stop" in the algorithm to check COMPOSITE RISK INDEX.

Variables: All variables listed under query CCI

Primary Key(s): PARCEL ID

Included in the Following Queries: TaxAllo

TaxAllocate(T)

Description: This table holds the information for high and low estimates of emergency management service costs in Lee County. This data is used in the query TaxAllo to calculate each parcel's risk-based assessment.

Variables: LOW CONTINUE COST, HIGH CONTINUE COST, LOW ANTICIPATE COST, HIGH ANTICIPATE COST, LOW DEBRIS REMOVAL COST, HIGH DEBRIS REMOVAL COST, LOW DAMAGE COST, HIGH DAMAGE COST, TOTAL FULLASSESSED

Primary Key(s): PARCEL ID

Included in the Following Queries: TaxAllo

FinalTax(T)

Description: This is the output table for the TaxAllo query. It reports each parcel's risk-based assessment for the low and high estimates of county emergency management costs.

Variables: LOW CONTINUE COST SHARE, HIGH CONTINUE COST SHARE, LOW ANTICI COST SHARE, HIGH ANTICI COST SHARE, LOW DEBRIS COST SHARE, HIGH DEBRIS COST SHARE, LOW PUBLIC COST SHARE, HIGH PUBLIC COST SHARE, LOW TAX ESTIMATE, HIGH TAX ESTIMATE

Primary Key(s): PARCEL ID

Included in the Following Queries: None

Queries in Data Base db1.mdb

QParcelData(Q)

Description: This query translates each property's distance from the coast in statute miles into nautical miles, with a built in check to determine whether the parcel falls on or "gulf ward" of the coastline established to separate the barrier islands and beachfront property from the rest of Lee County parcels.

Variables: PARCEL ID, STRUCTURE TYPE, NUMBER FLOORS, YEAR BUILT, EV, SQFT, ASSESSED, SH BENCHMARK, BFE, NAUT1, NAUT2, NAUT3, NAUT4, NAUT5, COAST DISTANCE NAUT, EVAC ZONE

Primary Key(s): PARCEL ID

Included in the Following Queries: None

QSurgeWave1(Q)

Description: This query uses data from the ParcelData2 and SurgeBench1 tables to calculate the depth of water and waves combined. The output of this query is then sent to subquery SubQSurgeWave1 to calculate PERCENT DAMAGE and EXPECTED DAMAGE.

Variables: PARCEL ID, STRUCTURE TYPE, SH BENCHMARK, NUMBER FLOORS, CAT 1 SURGE LEVEL, SEV, WATER DEPTH, MAX WAVE HGT, APPLY DAMAGE CURVE, WAVE HGT ABOVE SURGE, TOTAL WAVEWATER LEV, WAVEWATER DEPTH, WHITEWATER

Required Relationships: SurgeBench1--ParcelData (SH BENCHMARK)

Included in the Following Queries: None

QSurgeWave2(Q)

Description: This query uses data from the ParcelData2 and SurgeBench2 tables to calculate the depth of water and waves combined. The output of this query is then sent to subquery SubQSurgeWave2 to calculate PERCENT DAMAGE and EXPECTED DAMAGE.

Variables: PARCEL ID, STRUCTURE TYPE, SH BENCHMARK, NUMBER FLOORS, CAT 2 SURGE LEVEL, SEV, WATER DEPTH, MAX WAVE HGT, APPLY DAMAGE CURVE, WAVE HGT ABOVE SURGE, TOTAL WAVEWATER LEV, WAVEWATER DEPTH, WHITEWATER

Required Relationships: SurgeBench2--ParcelData (SH BENCHMARK)

Included in the Following Queries: None

QSurgeWave3(Q)

Description: This query uses data from the ParcelData2 and SurgeBench3 tables to calculate the depth of water and waves combined. The output of this query is then sent to subquery SubQSurgeWave3 to calculate PERCENT DAMAGE and EXPECTED DAMAGE.

Variables: PARCEL ID, STRUCTURE TYPE, SH BENCHMARK, NUMBER FLOORS, CAT 3 SURGE LEVEL, SEV, WATER DEPTH, MAX WAVE HGT, APPLY DAMAGE CURVE, WAVE HGT ABOVE SURGE, TOTAL WAVEWATER LEV, WAVEWATER DEPTH, WHITEWATER

Required Relationships: SurgeBench3--ParcelData (SH BENCHMARK)

Included in the Following Queries: None

QSurgeWave45(Q)

Description: This query uses data from the ParcelData2 and SurgeWave45 tables to calculate the depth of water and waves combined. The output of this query is then sent to subquery SubQSurgeWave45 to calculate PERCENT DAMAGE and EXPECTED DAMAGE.

Variables: PARCEL ID, STRUCTURE TYPE, SH BENCHMARK, NUMBER FLOORS, CAT 45 SURGE LEVEL, SEV, WATER DEPTH, MAX WAVE HGT, APPLY DAMAGE CURVE, WAVE HGT ABOVE SURGE, TOTAL WAVEWATER LEV, WAVEWATER

DEPTH, WHITEWATER

Required Relationships: SurgeBench45--ParcelData (SH BENCHMARK)

Included in the Following Queries: None

SubQSurgeWave1(Q)

Description: This query uses the results of the parent query (QSurgeWave1) and QSurgeWave1Table table series to "lookup" the PERCENT DAMAGE values from either the SurgeDam or WaveDam lookup tables. This variable is then used to calculate the value of EXPECTED DAMAGE. The lookup is accomplished because the combination of STRUCTURE TYPE, NUMBER FLOORS, and WHITEWATER form a unique set of information which corresponds to exactly one value for PERCENT DAMAGE. The results of this query are then output to table SubQuerySW1.

Variables: PARCEL ID, MAX WAVE HGT, WHITEWATER, PERCENT DAMAGE, EXPECTED DAMAGE

Required Relationships:	SurgeDam--ParcelData	(STRUCTURE TYPE)
	SurgeDam--QSurgeWave1	(STRUCTURE TYPE)
		(NUMBER FLOORS)
		(WHITEWATER)
	WaveDam--ParcelData	(STRUCTURE TYPE)
	WaveDam--QSurgeWave1	(STRUCTURE TYPE)
		(NUMBER FLOORS)
		(WHITEWATER)

Included in the Following Queries: None

Unusual Feature(s): Multiple joins are used here to "lookup" the appropriate PERCENT DAMAGE value from the SurgeDam and WaveDam lookup tables. In this case, a triple join is required, since the unique PERCENT DAMAGE values correspond not to a single field value, but to a unique combination of STRUCTURE TYPE, NUMBER FLOORS, and WHITEWATER values. The triple join guarantees the query will treat each group of values from these three fields as a single entity to match a single PERCENT DAMAGE value.

SubQSurgeWave2(Q)

Description: This query uses the results of the parent query (QSurgeWave2) and QSurgeWave2Table table series to "lookup" the PERCENT DAMAGE values from either the SurgeDam or WaveDam lookup tables. This variable is then used to calculate the value of EXPECTED DAMAGE. The lookup is accomplished because the combination of STRUCTURE TYPE, NUMBER FLOORS, and WHITEWATER form a unique set of information which corresponds to exactly one value for PERCENT

DAMAGE. The results of this query are then output to table SubQuerySW2.

Variables: PARCEL ID, MAX WAVE HGT, WHITEWATER, PERCENT DAMAGE, EXPECTED DAMAGE

Required Relationships:	SurgeDam--ParcelData	(STRUCTURE TYPE)
	SurgeDam--QSurgeWave2	(STRUCTURE TYPE)
		(NUMBER FLOORS)
		(WHITEWATER)
	WaveDam--ParcelData	(STRUCTURE TYPE)
	WaveDam--QSurgeWave2	(STRUCTURE TYPE)
		(NUMBER FLOORS)
		(WHITEWATER)

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave1(Q)

SubQSurgeWave3(Q)

Description: This query uses the results of the parent query (QSurgeWave3) and QSurgeWave3Table table series to "lookup" the PERCENT DAMAGE values from either the SurgeDam or WaveDam lookup tables. This variables is then used to calculate the value of EXPECTED DAMAGE. The lookup is accomplished because the combination of STRUCTURE TYPE, NUMBER FLOORS, and WHITEWATER form a unique set of information which corresponds to exactly one value for PERCENT DAMAGE. The results of this query are then output to table SubQuerySW3.

Variables: PARCEL ID, MAX WAVE HGT, WHITEWATER, PERCENT DAMAGE, EXPECTED DAMAGE

Required Relationships:	SurgeDam--ParcelData	(STRUCTURE TYPE)
	SurgeDam--QSurgeWave3	(STRUCTURE TYPE)
		(NUMBER FLOORS)
		(WHITEWATER)
	WaveDam--ParcelData	(STRUCTURE TYPE)
	WaveDam--QSurgeWave3	(STRUCTURE TYPE)
		(NUMBER FLOORS)
		(WHITEWATER)

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave1(Q)

SubQSurgeWave45(Q)

Description: This query uses the results of the parent query (QSurgeWave45) and QSurgeWave3Table table series to "lookup" the PERCENT DAMAGE values from either the SurgeDam or WaveDam lookup tables. This variable is then used to calculate the value of EXPECTED DAMAGE. The lookup is accomplished because the combination of STRUCTURE TYPE, NUMBER FLOORS, and WHITEWATER form a unique set of information which corresponds to exactly one value for PERCENT DAMAGE. The results of this query are then output to table, SubQuerySW45.

Variables: PARCEL ID, MAX WAVE HGT, WHITEWATER, PERCENT DAMAGE, EXPECTED DAMAGE

Required Relationships:

SurgeDam--ParcelData	(STRUCTURE TYPE)
SurgeDam--QSurgeWave45	(STRUCTURE TYPE) (NUMBER FLOORS) (WHITEWATER)
WaveDam--ParcelData	(STRUCTURE TYPE)
WaveDam--QSurgeWave45	(STRUCTURE TYPE) (NUMBER FLOORS) (WHITEWATER)

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave1(Q)

QWind1(Q)

Description: The purpose of this query is to calculate the speed of the maximum wind gust at each particular site for a Category 1 storm. There are 3 different expressions to use to determine site wind for a parcel. The decision is based on the distance to the coast from a parcel reference point in nautical miles. If equal to 0, use Option 1. If greater than 0 but not greater than 10, use Option 2. If greater than 10, use Option 3. It seems straightforward enough, but is a bit sticky in Expression Builder.

Variables: PARCEL ID, STRUCTURE TYPE, CATEGORY, MSSW MIDDLE, COAST DISTANCE NAUT, WINDY, SITE WIND, PEAK GUST, INTER PEAK GUST

Required Relationships: None

Included in the Following Queries: None

Unusual Feature(s): To solve the coastal distance problem, first create a dummy variable WINDY in the query and then right click on the "Field: space to build the expression. Click on the "IIf" function found in the Functions folder under Built-In

Functions. Set up the parentheses according to the model IIF expression at the bottom of the window. Then set the expression as "COAST DISTANCE NAUT > 0 AND COAST DISTANCE NAUT =< 10". If this statement is true, then WINDY equals the value of the algorithm specified for parcels greater than 0 but not greater than 10 nautical miles from the coast. If the statement is false, WINDY equals the value of the algorithm specified for parcels greater than 10 nautical miles from the coast.

WINDY alone cannot handle cases where parcels are more than 10 nautical miles from the coast (these cases would be assigned the algorithm for parcels at the coast). Therefore, save the query as is and then create a new field called SITE WIND. Build an IIf expression for SITE WIND in the following manner: set the expression as "COAST DISTANCE NAUT = 0." If this statement is true, then "SITE WIND" equals the value of the algorithm specified for parcels at the coast, or MSSW MIDDLE. If the statement is false, then SITE WIND equals WINDY, which has been defined to differentiate between the other two scenarios. Therefore, the query will correctly assign the appropriate algorithm, regardless of the value for COAST DISTANCE NAUT.

QWind2(Q)

Description: The purpose of this query is to calculate the speed of the maximum wind gust at each particular site for a Category 2 storm.

Variables: PARCEL ID, STRUCTURE TYPE, CATEGORY, MSSW MIDDLE, COAST DISTANCE, WINDY, SITE WIND, PEAK GUST, INTER PEAK GUST

Required Relationships: None

Included in the Following Queries: None

Unusual Feature(s): See QWind1(Q).

QWind3(Q)

Description: The purpose of this query is to calculate the speed of the maximum wind gust at each particular site for a Category 3 storm.
Variables: PARCEL ID, STRUCTURE TYPE, CATEGORY, MSSW MIDDLE, COAST DISTANCE, WINDY, SITE WIND, PEAK GUST, INTER PEAK GUST

Required Relationships: None

Included in the Following Queries: None

Unusual Feature(s): See QWind1(Q).

QWind45(Q)

Description: The purpose of this query is to calculate the speed of the maximum wind gust at each particular site for a Category 4 or Category 5 storm.

Variables: PARCEL ID, STRUCTURE TYPE, CATEGORY, MSSW MIDDLE, COAST DISTANCE, WINDY, SITE WIND, PEAK GUST, INTER PEAK GUST

Required Relationships: None

Included in the Following Queries: None

Unusual Feature(s): See QWind1(Q).

InterQWind1(Q)

Description: This interim query retrieves the integer value of INTER PEAK GUST from QWind1Table(T) and computes ROUND PEAK GUST, the product of INTER PEAK GUST and 5. This action effectively completes the conversion of PEAK GUST to its nearest multiple of 5, a value required for SubQWind1(Q).

Variables: PARCEL ID, STRUCTURE TYPE, CATEGORY, INTER PEAK GUST, ROUND PEAK GUST

Required Relationships: None

Included in the Following Queries: None

Unusual Feature(s): See QWind1(Q).

InterQWind2(Q)

Description: This interim query retrieves the integer value of INTER PEAK GUST from QWind2Table(T) and computes ROUND PEAK GUST, the product of INTER PEAK GUST and 5. This action effectively completes the conversion of PEAK GUST to its nearest multiple of 5, a value required for SubQWind2(Q).
Variables: PARCEL ID, STRUCTURE TYPE, CATEGORY, INTER PEAK GUST, ROUND PEAK GUST

Required Relationships: None

Included in the Following Queries: None

Unusual Feature(s): See QWind2(Q).

InterQWind3(Q)

Description: This interim query retrieves the integer value of INTER PEAK GUST from QWind3Table(T) and computes ROUND PEAK GUST, the product of INTER PEAK GUST and 5. This action effectively completes the conversion of PEAK GUST to its nearest multiple of 5, a value required for SubQWind3(Q).

Variables: PARCEL ID, STRUCTURE TYPE, CATEGORY, INTER PEAK GUST, ROUND PEAK GUST

Required Relationships: None

Included in the Following Queries: None

Unusual Feature(s): See QWind3(Q).

InterQWind45(Q)

Description: This interim query retrieves the integer value of INTER PEAK GUST from QWind45Table(T) and computes ROUND PEAK GUST, the product of INTER PEAK GUST and 5. This action effectively completes the conversion of PEAK GUST to its nearest multiple of 5, a value required for SubQWind45(Q).

Variables: PARCEL ID, STRUCTURE TYPE, CATEGORY, INTER PEAK GUST, ROUND PEAK GUST

Required Relationships: None

Included in the Following Queries: None

Unusual Feature(s): See QWind45(Q).

SubQWind1(Q)

Description: This subquery selects the appropriate PERCENT DAMAGE value from the lookup WindDam(T) through the unique combination of STRUCTURE TYPE and ROUND PEAK GUST. The product of PERCENT DAMAGE and ASSESSED yields EXPECTED DAMAGE.

Variables: All variables from InterQWind1Table(T), PERCENT DAMAGE, EXPECTED DAMAGE

Required Relationships:

InterQWind1Table--ParcelData2	(PARCEL ID)
InterQWind1Table--WindDam	(STRUCTURE TYPE, ROUND PEAK GUST)
WindDam--ParcelData2	(STRUCTURE TYPE)

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave1(Q)

SubQWind2(Q)

Description: This subquery selects the appropriate PERCENT DAMAGE value from the lookup WindDam(T) through the unique combination of STRUCTURE TYPE and ROUND PEAK GUST. The product of PERCENT DAMAGE and ASSESSED yields EXPECTED DAMAGE.

Variables: All variables from InterQWind2Table(T), PERCENT DAMAGE, EXPECTED DAMAGE

Required Relationships:		
	InterQWind2Table--ParcelData2	(PARCEL ID)
	InterQWind2Table--WindDam	(STRUCTURE TYPE, ROUND PEAK GUST)
	WindDam--ParcelData2	(STRUCTURE TYPE)

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave1(Q)

SubQWind3(Q)

Description: This subquery selects the appropriate PERCENT DAMAGE value from the lookup WindDam(T) through the unique combination of STRUCTURE TYPE and ROUND PEAK GUST. The product of PERCENT DAMAGE and ASSESSED yields EXPECTED DAMAGE.

Variables: All variables from InterQWind3Table(T), PERCENT DAMAGE, EXPECTED DAMAGE

Required Relationships:		
	InterQWind3Table--ParcelData2	(PARCEL ID)
	InterQWind3Table--WindDam	(STRUCTURE TYPE, ROUND PEAK GUST)
	WindDam--ParcelData2	(STRUCTURE TYPE)

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave1(Q)

SubQWind45(Q)

Description: This subquery selects the appropriate PERCENT DAMAGE value from the lookup WindDam(T) through the unique combination of STRUCTURE TYPE and

ROUND PEAK GUST. The product of PERCENT DAMAGE and ASSESSED yields EXPECTED DAMAGE.

Variables: All variables from InterQWind45Table(T), PERCENT DAMAGE, EXPECTED DAMAGE

Required Relationships:	InterQWind45Table--ParcelData2	(PARCEL ID)
	InterQWind45Table--WindDam	(STRUCTURE TYPE, ROUND PEAK GUST)
	WindDam--ParcelData2	(STRUCTURE TYPE)

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave1(Q)

QAnnualTotal1(Q)

Description: This query combines the effects of wind and water with the annual probability of a Category 1 hurricane strike to yield a composite annual expected damage.

Variables: PARCEL ID, WAVEWATER DAMAGE, WIND DAMAGE, DAMAGE SUM, ADJUST DAMAGE SUM, CATEGORY, STRIKE ANNUAL PROB, ANNUAL DAMAGE

Required Relationships:	ParcelData2--SubQWind1Table	(PARCEL ID)
	ParcelData2--SubQuerySW1	(PARCEL ID)
	StormCat--SubQWind1Table	(CATEGORY)

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave1(Q)

QAnnualTotal2(Q)

Description: This query combines the effects of wind and water with the annual probability of a Category 2 hurricane strike to yield a composite annual expected damage.

Variables: PARCEL ID, WAVEWATER DAMAGE, WIND DAMAGE, DAMAGE SUM, ADJUST DAMAGE SUM, CATEGORY, STRIKE ANNUAL PROB, ANNUAL DAMAGE

Required Relationships:	ParcelData2--SubQWind2Table	(PARCEL ID)
	ParcelData2--SubQuerySW2	(PARCEL ID)
	StormCat--SubQWind2Table	(CATEGORY)

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave2(Q)

QAnnualTotal3(Q)

Description: This query combines the effects of wind and water with the annual probability of a Category 3 hurricane strike to yield a composite annual expected damage.

Variables: PARCEL ID, WAVEWATER DAMAGE, WIND DAMAGE, DAMAGE SUM, ADJUST DAMAGE SUM, CATEGORY, STRIKE ANNUAL PROB, ANNUAL DAMAGE

Required Relationships:	ParcelData2--SubQWind3Table	(PARCEL ID)
	ParcelData2--SubQuerySW3	(PARCEL ID)
	StormCat--SubQWind3Table	(CATEGORY)

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave3(Q)

QAnnualTotal45(Q)

Description: This query combines the effects of wind and water with the annual probability of a Category 4 or 5 hurricane strike to yield a composite annual expected damage.

Variables: PARCEL ID, WAVEWATER DAMAGE, WIND DAMAGE, DAMAGE SUM, ADJUST DAMAGE SUM, CATEGORY, STRIKE ANNUAL PROB, ANNUAL DAMAGE

Required Relationships:	ParcelData2--SubQWind45Table	(PARCEL ID)
	ParcelData2--SubQuerySW45	(PARCEL ID)
	StormCat--SubQWind45Table	(CATEGORY)

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave45(Q)

DRI(Q)

Description: This query sums the total expected damage from all potential storm scenarios for each parcel, then sums the total potential damage of all county parcels to yield the total Lee County damage. This value serves as the denominator in subsequent formulas used to calculate each parcel's share of the Lee County risk. The values of ANNUAL DMG are read from each AnnualTotal table and summed to yield the value of TOTAL ANNUAL DMG. Then the query counts the PARCEL ID entries and sums the TOTAL ANNUAL DMG values for all parcels. The PARCEL ID count is just a dummy to complement the summation of TOTAL ANNUAL DMG values. Without the count, the

query could not compute the TOTAL ANNUAL DMG summation. It is the first of two queries and one table required to calculate the DAMAGE RISK INDEX value. Such a lengthy process is necessary because the sum of all ANNUAL DAMAGE values must be stored by the query DRI, output to the query DRI2 as a single value, then output to the table QDRI2 (another "stop"). The query is unable to sum all TOTAL ANNUAL DMG values and use this new value in a separate expression simultaneously.

Variables: PARCEL ID, TOTAL ANNUAL DMG

Required Relationships:	ParcelData2--AnnualTotal1	(PARCEL ID)
	ParcelData2--AnnualTotal2	"
	ParcelData2--AnnualTotal3	"
	ParcelData2--AnnualTotal45	"

Included in the Following Queries: DRI2(Q)

Unusual Feature(s): See SubQSurgeWave45(Q)

DRI2(Q)

Description: This query calculates the TOTAL ANNUAL DMG of each parcel and uses that value and the sum of all TOTAL ANNUAL DMG values from the query DRI to calculate the DAMAGE RISK INDEX. This data is then output to the table QDRI2

Variables: PARCEL ID, TOTAL ANNUAL DMG, DAMAGE RISK INDEX

Required Relationships:	ParcelData2--AnnualTotal1	(PARCEL ID)
	ParcelData2--AnnualTotal2	"
	ParcelData2--AnnualTotal3	"
	ParcelData2--AnnualTotal45	"

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave45(Q)

APMRI(Q)

Description: This totals query reads in the PARCEL ID and PE values as a preliminary step, counts the PARCEL ID entries, and sums the PE values. The PARCEL ID count is just a dummy to complement the summation of PE values. Without the count, the query could not compute the PE summation. It is the first of two queries and one table required to calculate the ANTICI PROTECT INDEX value. Such a lengthy process is necessary because the sum of all PE values must be stored by the query APMRI, output to the query APMRI2 as a single value, then output to the table QAPMRI2 (another "stop"). The query is unable to sum all PE values and use this new value in a separate expression simultaneously.

Variables: PARCEL ID, PE

Required Relationships: ParcelData2--EvacZone (EVAC ZONE)

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave45(Q)

APMRI2(Q)

Description: This query uses the sum of all PE values from the query APMRI to calculate ANTICI PROTECT INDEX. Output from this query goes to the table QAPMRI2.

Variables: PARCEL ID, PE, ANTICI PROTECT INDEX

Required Relationships: ParcelData2--EvacZone (EVAC ZONE)

Unusual Feature(s): See SubQSurgeWave45(Q)

PFRI(Q)

Description: This query computes the product of SQFT and ASSESSED values as the PUBLIC FACIL USE per parcel. Similar to the other index queries, this is a totals query and is the first of three steps to yield the PUBLIC FACILITY RISK INDEX value in the table QPFRI2. This totals query performs a "dummy" count of PARCEL ID, SQFT, and ASSESSED values. Then it calculates the sum of PUBLIC FACIL USE values for all parcels in Lee County. This data is output to the query PFRI2.

Variables: PARCEL ID, SQFT, ASSESSED, PUBLIC FACIL USE

Required Relationships: None

Included in the Following Queries: PFRI2

Unusual Feature(s): See SubQSurgeWave45(Q)

PFRI2(Q)

Description: This query uses the sum of all Lee County PUBLIC FACIL USE values to calculate the PUBLIC FACILITY RISK INDEX. The data is then output to the table QPFRI2.

Variables: PARCEL ID, SQFT, ASSESSED, PUBLIC FACIL USE, PUBLIC FACILITY RISK INDEX

Required Relationships: None

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave45(Q)

CCI(Q)

Description: This query calculates the value of COMPOSITE RISK INDEX. The data are then output to the table QCCI.

Variables: PARCEL ID, DAMAGE RISK INDEX, PUBLIC FACILITY RISK INDEX, ANTICI PROTECT INDEX, COMPOSITE RISK INDEX

Required Relationships:

ParcelData2--QAPMRI2	(PARCEL ID)
ParcelData2--QDRI2	“
ParcelData2--QPFRI2	“

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave45(Q)

TaxAllo(Q)

Description: The final query of the algorithm applies the appropriate risk index to each cost category and sums the weighted costs to calculate the low and high estimates of tax owed by each property owner to meet the total cost set by Lee County.

Variables: PARCEL ID, LOW CONTINUE COST SHARE, LOW ANTICI COST SHARE, LOW DEBRIS COST SHARE, LOW PUBLIC COST SHARE, LOW TAX ESTIMATE, HIGH CONTINUE COST SHARE, HIGH ANTICI COST SHARE, HIGH DEBRIS COST SHARE, HIGH PUBLIC COST SHARE, HIGH TAX ESTIMATE

Required Relationships:

ParcelData2--QAPMRI2	(PARCEL ID)
ParcelData2--QDRI2	“
ParcelData2--QPFRI2	“
ParcelData2--QCCI	“

Included in the Following Queries: None

Unusual Feature(s): See SubQSurgeWave45(Q)