SAM / MBRS

**CONSERVATION AND SUSTAINABLE USE OF THE MESOAMERICAN BARRIER REEF SYSTEMS PROJECT (MBRS)**

Belize – Guatemala – Honduras - Mexico

# DATABASE DESIGN DOCUMENTATION

Design and Implementation of a

Regional Environmental Information System (REIS)

for the Mesoamerican Barrier Reef Systems Project

Project Coordinating Unit
Coastal Resources Multi-Complex Building
Princess Margaret Drive
P.O. Box 93
Belize City Belize
Tel: (501) 223-3895; 223-4561
Fax: (501) 223-4513
Email: mbrs@btl.net
Website: http://www.mbrs.org.bz

# CONSERVATION AND SUSTAINABLE USE OF THE MESOAMERICAN BARRIER REEF SYSTEM (MBRS)

## DATABASE DESIGN DOCUMENTATION

## DESIGN AND IMPLEMENTATION OF A REGIONAL ENVIRONMENTAL INFORMATION SYSTEM (REIS) FOR THE MESOAMERICAN BARRIER REEF SYSTEMS PROJECT

**Mesoamerican Barrier Reef Systems Project Coordination Unit**
**Coastal Resources Multi-Complex, Fisheries Compound, Princess Margaret Drive**
**P.O. Box 93, Belize City, Belize Central America**
**Tel: 501-22-33895/34561**
**Fax: 501-22-34513**
**E-mail: mbrs@btl.net**

# TABLE OF CONTENTS

# DATABASE DESIGN DOCUMENTATION

## 1. INTRODUCTION

The Mesoamerican Barrier Reef System (MBRS) is the largest barrier reef system in the Caribbean and the second largest reef system in the world. The primary goal of the MBRS project is to enhance protection of these valuable ecosystems. Key to any protection strategy is knowledge of the habitats and the various uses of those habitats. Management decisions need to be based on information of the system. While there may be many efforts to monitor, study, and manage individual portions of the MBRS, a system wide overview of the information available is essential for management of the MBRS as a whole unit. Collecting all of the disparate information and compiling it into one easily accessible database is the goal of the Regional Environmental Information System (REIS). The design of the database is based on the information that is to be collected, or has been collected in the past.

The data are stored in PostgreSQL 7.3.2 on a Dell Server running Red Hat Linux Version 8.2. Access to the data will be through a web interface running on Apache web server and using PHP. This design is to allow easy data entry access and querying to researchers throughout the 4 countries served by MBRS.

The driving philosophy behind the database design was to have an efficient, normalized database that would be easy to maintain and expand, as well as allow easy data entry and access.

# 2. SOFTWARE AND HARDWARE SPECIFICATIONS

## 2.1 Software Selection

Following reviews of the data that would be entered into the database, and the requirements of retrieving the data several criteria were identified that need to be met by the database software. The minimum requirements for the software were:

1. Must support the relational database model, and some version of the SQL language. This is an industry standard, and as a program that will be spanning several countries and many years adhering to this standard will ensure the longevity and portability of the database. In addition most database administrators are familiar with some form of SQL and relational databases, so training of an administrator will be relatively easy.
2. Need to allow multiple users to access tables simultaneously. Since the data is going to be entered by users from 4 countries and numerous agencies, it is likely that multiple people will be entering data into the same table at the same time.
3. Allow running of stored SQL scripts. There are many processes that can be automated with stored scripts, to facility management, updates, editing, and querying of the database. This is especially important if users are accessing the data from the internet. Being able to call and run a stored script is far easier and more efficient than trying to code all the information into a web form.
4. Allow restrictions on the data values entered in columns within a table. Being able to restrict the data to certain ranges or values will reduce the possible errors in data entry.
5. Allow creation of multiple indexes on a table, as well as unique indexes within a table. Also must be able to create one index on multiple columns. Multiple indexes on a table allow faster sorts and queries based on various parameters. Creating a unique index across multiple columns will prevent entering duplicate data.

6. Allow creation of views on the data. This allows a minimum amount of data to be stored and a virtually unlimited number of outputs to be created. Views allow display of calculated values, without having to create additional columns in the data tables and have them populated with the calculated values. Having the additional columns can lead to conflicting data within one record of a data table. In addition, views allow multiple tables to be joined together to provide a customized view of the data in the data table.
7. Allow inner joins, left outer joins, right outer joins, full outer joins, and multiple joins within a query. The joins are different ways of selecting items from one or more tables, in either a query or a view. The inner join selects only the records that exist in both tables and matches them up. The left outer join, selects all of the records from the left table and only the matching records from the right table. The right outer join, selects all of the records from the right table and only the matching records from the left table. The full outer join selects all records from both the left and right table and joins the records that match. The non-matching records are joined with null values.
8. Have some method of replication between two servers. Since the data is going to be housed on two servers some sort of replication is necessary.
9. Allow triggers on the data tables. Triggers will allow predetermined actions to be taken when information is entered, edited, or deleted from a data table. Column data checks are an intrinsic form of triggers.
10. Allow data entry from the internet. Most of the data will be entered into the forms from the internet.
11. Had to run on a Linux System. The project is running a Linux server, therefore the database program must run on Linux.

Based on these requirements the qualified software was examined was Oracle, Informix, Ingres, and PostgreSQL. All of these products met the requirements outlined above. PostgreSQL offered the best price/performance of the qualified software. Based on research of the computer literature, PostgreSQL appeared to serve data over the web as fast or nearly as fast as any of the other

products. Also being an open source program there is no upfront cost to acquiring the software. Based on the cost and performance PostgreSQL was chosen as the software to use for this project.

Following the selection of the database software and operating system, the web server and server side scripting language defaulted to Apache Web server and PHP. This is the best combination that supports Linux and PostgreSQL.

## 2.2 Hardware Requirements

### *2.2.1 General Specifications*

The database server will be used as a web server and database server for a regional project with its central office located in Belize. A tower chassis has been chosen to accommodate an internal LTO tape drive. This server will be connected via the Internet with another server located in Belmopan, Belize. Both servers will have the same hardware and software configuration. The manufacturer of any software included with the hardware must be a reputable and globally recognized manufacturer of that class of software. The Manufacturer of the proposed equipment must be a reputable and globally recognized manufacturer of microcomputer hardware. Absolutely no clones will be considered. The equipment vendor must be an authorized dealer of the proposed equipment and software preferably with an office in each country where equipment is to be located.

### *2.2.2 Detailed Technical Specifications*

The server should meet the following detailed specification. These specifications were determined based on the size of the database, the number of users, and the life expectancy of the project. Based on monetary considerations it is not expected that the server will be replaced for at least 5 years.

| **Item** | **Description** |
|---|---|
| System Processors | Dual Intel® Xeon 2.4GHz with NetBurst Micro-architecture with Hyper-Threading technology |
| Front Side Bus | 400MHz front side bus |
| Cache | 512KB L2 Advanced Transfer Cache |
| Chipset | ServerWorks GC-LE chipset |
| Memory | 2GB 200MHz ECC DDR SDRAM (2 x 1 GB) |
| Memory Expandable to: | Total of 6 DIMM sockets on system board configurable for up to 6 GB |
| Expansion Slots | 3 full length PCI-X slots (1 X 64bit/133MHz, 2 X 64bit/100MHz) |
| RAID Controller<br>(Primary Controller) | Dual channel, integrated RAID Controller<br>With 128MB battery-backed cache<br>2 internal channels<br>Embedded RAID i.e. ROMB (RAID On Motherboard)<br>Capable of handling RAID 1 and RAID 5 |
| Hard Drive Backplane | 5 Bay Hot Plug SCSI Hard Drive Backplane for<br>1 x 5 configuration<br>On-Board RAID 1, RAID 5<br>5 drives connected to on-board RAID |
| Hard Drives | 5 73GB (10,000 rpm) 1 inch Ultra3 (Ultra 160) Hot Plug SCSI |
| Diskette Drive | 1.44MB Diskette Drive |
| Optical Drive | DVD ROM (CD-ROM capable) Drive |
| Monitor | 15in (13.8inch viewable) Monitor |
| Graphics Card | Integrated controller w/8MB of RAM |
| Network Adapter | Intel Pro/100+ Dual Port Server Adapter<br>To allow connection to a 100Mbps port on Internet Switch<br>With failover and load balancing support |
| Keyboard | Standard Windows PS/2 Keyboard<br>With Keyboard Cable |
| Mouse | PS/2 two-button mouse with scroll wheel and<br>With Mouse Cable |

| **Item** | **Description** |
|---|---|
| SCSI Drive Controller<br>(Secondary Controller) | SCSI Drive Controller – compatible with Internal LTO Tape Backup Unit<br>Plus appropriate cable(s) |
| *Tape Backup Unit* | Internal LTO Tape Backup Drive<br>Capacity: 100 GB native, 200 GB compressed<br>Media Type: LTO Ultrium<br>Recording Format: LTO Ultrium Generation 1<br>Average Seek / Access Time: 71s<br>Data Transfer Rate: 15 MBps native, 30 MBps compressed<br>Interface Type: Ultra wide SCSI-2 |
| Tape Backup Software | Veritas Netbackup Datacenter or equivalent<br>Compatible with LTO drive<br>Must be able to run on a RedHat Linux 8.0 operating system<br>Suitable for use in a relational database environment |
| 25-pin parallel port | |
| 9-pin serial ports | |
| Universal Serial Bus ports | |
| Power Supplies | Hot pluggable, redundant 500 watts power supplies (2x500watt) and hot pluggable fans<br>Voltage: 100-240 VAC |
| Chassis | Tower Chassis |
| Operating System | Red Hat Linux 8.0 Professional Installed<br>With up-to-date drivers for all system components including video, SCSI, motherboard, NIC, etc. |
| Operating System Documentation Set | Documentation and Media for Red Hat Linux 8.0 Professional |
| Management Software | Embedded Hardware-based Remote Access (ERA) management features, including built-in port, to enable administrators to access, diagnose and remotely manage the server |
| | Server management tools with the following features:<br>▪ Facilitate system set-up, installation and configuration<br>▪ Complete event management including logging and filtering events<br>▪ Email or paging to keep administrators informed of potential server problems before they become critical<br>▪ Fault monitoring of voltage, fan, and thermal conditions to help ensure notification in case of potential problems<br>▪ Asset management features to enable system administrator to inventory server |

| **Item** | **Description** |
|---|---|
| | configuration, CPU, memory and disk information, helping keep track of systems and keep them up-to-date<br>▪ Built-in remote management<br>▪ Management of drive array under RAID Controller |
| | Pre Executable Environment (PXE) support of embedded NICs |
| | Must have Simple Network Management Protocol (SNMP) agent software available |
| Environmental Parameters | Operating Temperature: 10º C to 35º C (50º F to 95º F)<br>Operating Relative Humidity: 8% to 80% (non-condensing)<br>Storage Relative Humidity: 5% to 95% (non-condensing) |
| Hardware Documentation Set | Users Manual, Installation and Trouble Shooting Guide on CD |
| Warranty and Hardware Support | 3 Year Onsite Parts and Labor Warranty |
| Uninterruptible Power Supply | Stand alone 2200VA/1600W Smart UPS 120 V to provide 30 minutes of runtime at half-load with the following features:<br>▪ Input 120V/ Output 120V,<br>▪ Input frequency 50/60 Hz +/- 3 Hz (auto sensing)<br>▪ DB-9 RS-232 Interface Port w/ Smart UPS signalling RS-232 cable<br>▪ Network-grade line conditioning<br>▪ Management Software on CD<br>▪ User Manual & Installation Guides<br>▪ Overload Indicator and Replace Battery Indicator<br>▪ LED status display with load and battery bar-graphs<br>▪ Optional Emergency Power Off (EPO) Optional<br>▪ Surge energy rating 320 joules<br>▪ Full time multi-pole noise filtering : 0.3% IEEE surge let-through : zero clamping response time : meets UL 1449<br>▪ Maintenance-free battery sealed Lead-Acid battery with suspended electrolyte : leakproof<br>▪ Typical recharge time: 3 hour(s)<br>▪ 2-year repair or replace warranty |

## 3. DATABASE DESIGN

The first step in the database design was to analyze the data that would be collected and determine the expected uses of the data.

For consistency, each data group is defined as a group of related data tables. Data from one or more surveys may be included in a group. Analysis of the datasheets and data collection methods identified several different data groups. The groups are:

Mangrove monitoring

Seagrass monitoring

Coral Reef monitoring

Pollution monitoring

Once the groups where identified, the commonalities between the datasheets within each group where identified. These commonalities would be in one table, to which all of the other tables in the group would be linked. This would provide a connection between the various tables within a group.

To aid in the management of the data and tables, each table would have at least one field that contained a unique identifier for that record, a field to identify who was doing the data entry, and another field to track when the data was entered. In most cases these fields are hidden from the user and are updated automatically by the system. This information is accessible by the administrator for troubleshooting purposes.

Certain conventions were followed in this report. All table names are in **bold**. All column names are in *italic*. The following values are found in the Index Column of the tables in this report:

Index 1 – All of the column names in a table with this value are indexed together to create the primary index on the table.

Index 2 – All of the column names in a table with this value are indexed together to create the secondary index on the table.

Unique Index 1 – All of the column names in a table with this value are indexed together to create a unique primary index on the table. The combination of values in this index cannot be duplicated anywhere in the table.

Unique Index 2 – All of the column names in a table with this value are indexed together to create a unique secondary index on the table. The combination of values in this index cannot be duplicated anywhere in the table.

## 3.1 Common Tables

There is a group of base tables that is used throughout the database. These tables are common to some or most of the groups listed.

### *3.1.1 Site Table*

At the highest level is the site information. The site table was created to store general information for the site. This is one of the smallest tables, yet it provides a spatial reference to all of the data in the database. In the event that a GIS system is used this table can be used as a link between the spatial information and the data. This table is not fully normalized. The fields *location* and *ecosystem* could be kept in a separate table to minimize space used by this table. However, it was decided to include them in this table to facilitate the ease of use. In addition, this table is expected to be relatively small (possibly only a few hundred records), therefore splitting out these fields into a separate table would not result in any significant savings in space. The fields for the **site** table are:

**Table Name - site**

| Column Name | Type* | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *site_id* | varchar(8) | Site ID | | Unique Index | No | This field is a unique alphanumeric identifier for the site. This identifier should be used by all data collectors for the site |
| *location* | varchar(50) | Location | | | No | This field is the name of the SMP Location that is being monitored by this site, for example "Lighthouse Reef". |
| *latitude* | decimal(9,6) | Latitude | 15.0° N to 21.5° N | | No | The latitude of the site in decimal degrees. For accuracy this should be to 5 decimal places (approximately 1 meter.) |
| *longitude* | decimal(9,6) | Longitude | 83.25° W to 89.0° W | | No | The longitude of the site in decimal degrees. For accuracy this should be to 5 decimal places (approximately 1 meter.). |
| *agency* | varchar(80) | Support Agency | | | No | The Laboratory that is conducting the monitoring at this site. |
| *ecosystem* | varchar(25) | Ecosystem | Coral, Mangrove, Seagrass | | No | The ecosystem where the site is located. A list of the ecosystems are given in *Manual of Methods for Synoptic Monitoring*, Page 20. |
| *habitat* | varchar(25) | Habitat | shallow back-reef, shallow fore-reef, deep fore-reef, coastal, fringing | | No | The habitat within the ecosystem that is being monitored. A listing of the habitats are given in *Manual of Methods for Synoptic Monitoring*, Page 20. |
| *country* | varchar(15) | Country | Mexico, Belize, Guatemala, Honduras | | No | The country in which the sampling site is located. |

| Column Name | Type* | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *category* | smallint | Category | 1,2,3 | | No | The monitoring category for the site. Details for the monitoring category are given in *Manual of Methods for Synoptic Monitoring*, Page 6. |
| *description* | text | Site Description | | | Yes | A textual description of the observations made at the site on the initial visit. This can be from 1 paragraph to half a page long. |
| *selection* | varchar(14) | Selection Criteria | unbiased, strategic, representative | | Yes | The method by which the site was chosen.<br>Unbiased – Chosen based on a random sampling strategy.<br>Strategic – Chosen with local knowledge because they are threatened, suspected to be degraded, or in particularly good condition, or because they are currently being monitored through another program.<br>Representative – Chosen with local knowledge to be representative of reefs in that area. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

*A description of all the data types can be found in Appendix A.

### *3.1.2 Survey Table*

One level down from the **site** table is the **survey** table. This table is linked to the **site** table via the *site_id* value. This table is used across all of the groups. This table contains the detailed, sampling specific information such as time, date, sampler, weather conditions, lab providing analysis, etc. All of the specific data collection tables are linked to this table through the **transect** table and the *survey_id* field. *Survey_id* is automatically added by the database when a new survey record is entered. The **survey** table has one entry for each separate dataset (i.e. point intercept benthic survey and benthic coral survey) every time a site is surveyed. This table is connected back to the **site** table by *site_id*. This design allows the site information to be entered only once, regardless of how many samples are collected at a site, on a given day. The **survey** table has one entry for each time a site is sampled. There is a one-to-many relationship between the **site** table and the **survey** table. Below is the layout of the survey table:

**Table Name -survey**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *site_id* | varchar(8) | Site ID | | Index 2 | No | This field relates the survey table back to the **site** table. In the **survey** table this value is automatically entered by the input form. |
| *startdate* | date | Sampling Start Date | After January 1, 2003 and before the entry date. | Index 2 | No | The beginning date for a timed interval sampling, such as leaf litter or seagrass growth. If there is no timed sampling, the start date is the date of the sampling. |
| *enddate* | date | Sampling End Date | After January 1, 2003 and before the entry date. | | Yes | The ending date for a timed interval sampling, such as leaf litter or seagrass growth. If there is no timed sampling, then this field is left blank. |
| *starttime* | time | Time | | | No | The time of day when sampling was started. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *person_col* | integer | Collector or Recorder | Values from pull down list | | No | This field contains an ID number that links to the name of the person collecting or recording the data in the field. This field should be filled out for all transects |
| *person_proc* | integer | Processor | Values from pull down list | | Yes | This field contains an ID number that links to the name of the person processing the data. This field is used only with the seagrass biomass and growth transects. |
| *tide* | varchar(7) | Tide | high, low, falling, rising | | No | Tidal stage at the time of sampling. The values are:<br>high – One hour either side of high water.<br>low – One hour either side of low water.<br>falling – The period between high and low water.<br>rising – The period between low and high water. |
| *sea* | varchar(15) | Sea state | calm, slight, moderate, rough | | No | The sea state at the time of sampling. The values are:<br>calm – Mirror-like to small ripples.<br>slight – Small waves, some whitecaps.<br>moderate – Moderate waves, many whitecaps.<br>rough – Large waves, 2-3 m, whitecaps everywhere, some spray. |
| *wind* | integer | Wind Speed | 1-5 | | Yes | Wind strength category based on wind speed The values are:<br>1 – 0-5 knots<br>2 – 6-10 knots<br>3 – 11-15 knots<br>4 – 16-20 knots<br>5 – 21-25 knots |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *winddir* | integer | Wind Direction | 0-360 | | Yes | The direction the wind is blowing from in degrees. |
| *cloud* | integer | Cloud Cover | 0-8 | | Yes | Cloud cover is quantified in terms of eighths of the sky area covered by clouds. The unit of measure is the okta, with a possible range of 0 (no clouds) to 8 (completely overcast) |
| *secchimark* | decimal(9,3) | Secchi at Marking | | | Yes | Secchi reading at the time the seagrasses are marked for the seagrass growth data. This field is used only in association with the seagrass growth transects. |
| *secchicoll* | decimal(9,3) | Secchi at Collection | | | Yes | Secchi reading at the time the seagrasses are collected for the seagrass growth data. This field is used only in association with the seagrass growth transects. |
| *tempair* | decimal(9,3) | Air Temperature (°C) | | | Yes | The ambient air temperature at the time of the sampling. |
| *currentspd* | decimal(9,3) | Water Current Speed (m/s) | | | Yes | The water current speed in meters/sec at the time of sampling. |
| *currentdir* | decimal(9,3) | Water Current Direction | 0-360 | | Yes | The water current direction at the time of sampling. In degrees from north. |
| *survey_type* | varchar(15) | Survey Type | | | No | This is a character field that identifies what data set this survey record is associated with. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *survey_id* | integer | Survey ID | | Unique Index 1 | No | This is an integer value assigned by the database to uniquely identify this record. The data from all the datasheets links back to this record on the *survey_id* number. This number should not be entered or altered by the user. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

The various sample datasheets are each given their own table, and are related back to the survey table. Multiple sample sheet tables may relate back to one survey table entry. This design minimizes the data entry required for identifying the sampling site and time.

### *3.1.3 Transect*

The **survey** table records all of the information that is collected once per each site visit. In addition to the survey information a table is needed to keep track of the transects at each survey. The information recorded for each transect is usually limited to start times and person. Therefore, a separate table was created for each transect that serves as a link between the detailed data collected and the survey record. An entry is required in this table for every site or transect that is completed. The system assigns each entry in

this table a unique number called *transect_id.* All of the datasheets link to this table on this number. Even if there is only one sampling conducted for a site, and no transect or plots are used an entry is still required in this table, with a transect number of 1.

**Table name – transect**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *survey_id* | integer | Survey ID | | Unique Index 2 | No | This field links these records to the **survey** table. The values are automatically entered by the input form. |
| *transect* | varchar(5) | Transect or Plot | | Unique Index 2 | No | The transect or plot for which the data is being collected. There may be multiple transects within one site on one day. There should be a separate entry for each transect. |
| *tstarttime* | time | Start Time | | | Yes | The time the survey was started for the transect. |
| *person_col* | integer | Collector or Recorder | Values from pull down list | | No | This field contains an ID number that links to the name of the person collecting or recording the data in the field. This field should be filled out for all transects |
| *depthstart* | decimal(9,3) | Water Depth – Start | | | Yes | The water depth in meters at the start of the coral transect. This field is only used for coral transects. |
| *depthend* | decimal(9,3) | Water Depth – End | | | Yes | The water depth in meters at the end of the coral transect. This field is only used for coral transects. |
| *bearing* | integer | Bearing | 0-360 | | Yes | Heading in degrees from north of the transect line from the shoreline. Used for mangrove forest zonation. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 1 | No | This is an integer value assigned by the database to uniquely identify this record. The data from all the datasheets links back to this record on the *transect_id* number. This number should not be entered or altered by the user. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.1.4 Person*

The **person** table is a lookup table that provides an authoritative list of names of the people and their agencies that are registered to collect data for the project. This table is related to the columns *person_col* and *person_proc* in **survey** table or *person_col* in **transect** table on *person_id*. Optionally this table could be expanded to include more information for the individual than just the name and agency.

**Table Name - person**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *person_id* | integer | Person ID | | Unique Index | No | This is an integer value assigned by the database to uniquely identify this record. All other tables that have people's names are linked to this table on this field. |
| *person* | varchar(50) | Name of Individual | | | No | This is the name of the individual. The full name of the individual is entered into this field. |
| *agency* | varchar(120) | Agency | | | Yes | The agency with which the person is affliated. |
| *country* | varchar(10) | Country | Mexico, Belize, Guatemala, Honduras | | Yes | The country the person and agency are associated with. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

## 3.2 Species List

Several tables are required to create an efficient species list. Since this database is being used in several countries and in 2 languages, there could be a problem keeping track of common names and threatened and endangered status of various species. The names and status change from country to country, and may even be different within different regions of a country. Therefore these features have been broken out into separate table to accommodate the potential variety.

### *3.2.1 Species*

The species table is the basis of all the species information. At present it only contains the scientific name (Genus and species), family, order, class and broad taxonomic classification such as bird, fish, invertebrate, etc. It is hoped that it will be able to be expanded to accommodate the full taxonomic information for each species. All of the tables that require species names store only the *species_id* from this table.

**Table Name - species**

| Column Name | Type | Descriptive Name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *species_id* | integer | Species ID | 1-2999 | Unique Index 1 | No | An integer number to link the species name to the various tables. This number is a unique identifier for the species. |
| *gen_spec* | varchar(45) | Scientific Name | | Unique Index 2 | No | The scientific (Genus and species) name of the animal or plant. If species level information is not known then the value in the field would be *Genus_name spp.* |
| *family* | varchar(25) | Family Name | | | Yes | The family the species belongs in. Super- and subfamilies are not included at this time. |
| *ordr* | varchar(25) | Order Name | | | Yes | The order the species belongs in. |

| Column Name | Type | Descriptive Name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *class* | varchar(25) | Class Name | | | Yes | The class the species belongs in. |
| *element* | varchar(10) | Biological Element | FISH, CORAL, TREE, INVERT | Unique Index 2 | No | Broad taxonomic group for the species. This field is used to categories the species in the list for purposes of data entry and data manipulations. Currently the elements are: FISH, CORAL, TREE |
| *form* | varchar(15) | Data Form | | | | Codes for the data form in which the species is used. This is mainly for fish, which are used on several forms. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.2.2 Local Names*

The table **localname** provides the local common name for the species. This table is designed to accommodate various local names based on countries or regions within a country. The information in this table includes the common name, what country that common name is used in, and if appropriate the region of the country. The local name is linked to the species table and the scientific name based on the *species_id*.

**Table Name - localname**

| Column Name | Type | Descriptive Name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *species_id* | integer | Species ID | 1-2999 | Index 2 | No | This is the field that contains the id number for the species and is linked to the **species** table. |
| *name* | varchar(35) | Common Name | | Index 2 | No | Local common name for the species. |
| *country* | char(2) | Country Code | MX,BZ,GT,HN | Index 2 | No | 2 letter country code in which the local name is used. |
| *locality* | varchar(40) | Locality | | | Yes | The locality within a country (if appropriate) for which the common name is valid. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.2.3 Threatened and Endangered*

This table identifies the threatened and endangered status of species. It includes the national ranking for each country as well as the IUCN Red Book listing. The only species included in this list are those that are listed as threatened or endangered in a country, or have and IUCN Red Book listing of critically endangered, endangered, or vulnerable. It is designed to accommodate different listing statuses based on country. This table is linked to **species** on *species_id*.

**Table Name - tande**

| Column Name | Type | Descriptive Name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *species_id* | integer | Species ID | 1-2999 | Index 2 | No | This field contains the id number for the species and is linked to the **species** table. |
| *nation* | char(1) | National Ranking | T,E,R | | Yes | Field to indicate the national endangered or threatened status The values are:<br>T – Threatened<br>E – Endangered<br>R - Rare |
| *iucn* | char(2) | IUCN Red Book Listing | CR,EN,VU | | Yes | Field to indicate the international status, this is based on the IUCN Red List. The values are:<br>CR – critically endangered<br>EN – endangered<br>VU – vulnerable<br>Typically the other categories of the IUCN would not be used. |
| *country* | char(2) | Country Code | MX,BZ,GT,HN | Index 2 | Yes | The country for which the national listing is applicable. |
| *ndate* | integer | National Source Date | | | Yes | An integer number to represent the year and month of the source data for determining the national threatened or endangered status listing. The value is entered as YYYYMM. |

| Column Name | Type | Descriptive Name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *idate* | integer | IUCN Red Book Source Date | | | Yes | An integer number to represent the year and month of the source data for determining the IUCN listing status. The value is entered as YYYYMM. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### 3.3 Mangroves

The following tables are in the mangrove monitoring group: **structure**, **seedling**, **seedlingbio**, **litter**, **zonation**, **interstitialwater**. This is only a logical assemblage of tables and is not physically set as an assemblage in the data structure. The common links for all of these tables is the *transect_id*. Therefore this group can be easily changed, by adding additional tables, or removing tables as needed. In all the tables that require species name, a *species_id* is stored in the data table that is linked to the **species** table identified above. This provides the flexibility of easily accommodating changes to the scientific or common names, if necessary, in the future. It also eliminates the possibility of misspelling a species name.

### *3.3.1 Forest Structure*

This data table is for recording the information for the forest structure. It is based on the forest structure spreadsheet. It is related to the **transect** table by the *transect_id* field. Since there may be numerous plots at one site, there would be one *transect_id* record for each plot. There would then be multiple entries in this table for each transect record. It is also linked to the species table on *species_id*. The table is as shown below:

**Table Name - structure**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *position_x* | decimal(9,3) | Position X | | | No | The x location in meters relative to the corner of the plot. |
| *position_y* | decimal(9,3) | Position Y | | | No | The y location in meters relative to the corner of the plot. |
| *tree* | integer | Tree No. | | Unique Index 2 | No | The tree number in the plot. |
| *species_id* | integer | Species ID | 0-2999 | | No | This is the field that contains the id number for the species and is linked to the **species** table. |
| *cbh* | decimal(9,3) | Circumference (cm) | | | Yes | The circumference at Breast Height. This value is the measure of the circumference of the tree in centimeters. From this the diameter at breast height (DBH) will be calculated. DBH is a common measurement used in forestry. |
| *proproot* | decimal(9,3) | Prop Roots (cm) | | | Yes | The height of the prop root in centimeters. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *trunklength* | decimal(9,3) | Trunk Length (m) | | | Yes | The length of the trunk in meters. This is the distance from the proproot to the first branch. |
| *treeheight* | decimal(9,3) | Tree height (m) | | | Yes | The height of the tree in meters. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | The unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.3.2 Seedling*

The **seedling** table is used for recording the information from the Mangrove structure seedling/sapling data entry form. As with the forest structure, this table is related back to the **transect** table with *transect_id*. It is also linked to the species table on *species_id*. There is one transect record for each plot, and there will be 5 subplots within the plot

**Table Name -seedling**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *subplot* | smallint | Subplot No. | | Unique Index 2 | No | The subplot identifier within the plot. |
| *position_x* | decimal(9,3) | Position X (cm) | | | No | The x location in centimeters relative to the corner of the plot. |
| *position_y* | decimal(9,3) | Position Y (cm) | | | No | The y location in centimeters relative to the corner of the plot. |
| *sapling* | integer | Sapling or Seedling No. | | Unique Index 2 | No | The seedling/sapling number in the subplot. |
| *species_id* | integer | Species ID | 0-2999 | | No | This is the field that contains the id number for the species and is linked to the **species** table. |
| *cbh* | decimal(9,3) | Circumference (cm) | | | Yes | The Circumference at Breast Height (CBH). This value is the measure of the circumference of the tree in centimeters. From this the diameter at breast height (DBH) will be calculated. DBH is a common measurement used in forestry. |
| *height* | decimal(9,3) | Height (cm) | | | No | The height of the seedling/sapling in centimeters from the sediment surface. |
| *live* | Varchar(1) | Live (Y/N) | Y,S,N | | No | Indicates whether the seedling/sapling is alive. Input is Y,S, or N |
| *observation* | text | Observations | | | Yes | General observations about the subplot or seedling. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.3.3 Seedling biomass*

The seedling biomass table **seedlingbio** is based on the Seedling biomass data entry form. As with the forest structure, this table is related back to the **transect** table with *transect_id*. It is also linked to the species table on *species_id*.

**Table Name - seedlingbio**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *species_id* | integer | Species ID | 0-2999 | | No | This is the field that contains the id number for the species and is linked to the **species** table. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *sapling* | varchar(8) | Sapling/ Seedling | sapling, seedling | | Yes | Indicate whether the plant being collected and weighed is a seedling or sapling. This may be used later to determine if there is a difference in the height to weight ratio between seedlings and saplings. |
| *height* | decimal(9,3) | Height (cm) | | | No | The height of the seedling/sapling in centimeters from the sediment surface. |
| *tare* | decimal(9,3) | Tare Wt (g) | | | No | The tare weight in grams. If the scale is already adjusted to compensate for tare weight, enter zero in this field. |
| *total* | decimal(9,3) | Tare Wt + Sample Wt (g) | | | No | The total weight of the sample, including tare, in grams. If the scale is already adjusted to compensate for tare weight, enter the final weight reading here. |
| *record_id* | integer | Record ID | | Unique Index | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.3.4 Interstitial Water*

This table has the interstitial water salinity information for the mangrove areas. It is based on the interstitial water entry form. As with the forest structure, this table is related back to the **transect** table with *transect_id*.

**Table Name - interstitial**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *depth* | integer | Approx. Sample Depth (cm) | | | No | Depth in centimeters at which the sample was collected. |
| *sedexposed* | char(1) | Sediment Surface Exposed | Y,S,N | | No | Indicate whether the sediment surface was above the water level at the time of sample collection. This field takes a Y,S or N. |
| *salinity* | smallint | Salinity (ppt) | | | No | The salinity of the water in parts per thousand. It is expected that salinity would be whole numbers. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.3.5 Leaf Litter*

The litter table is based on the leaf litter data entry form. This is one of the forms that requires a start and end dates. The start and end dates are entered in the **survey** table, and not in the **litter** table. This same table is used for initial biomass, but there would only be an entry for the start date in the **survey** table, and the end date would be blank.

**Table Name - litter**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *trap* | smallinteger | Trap No. | 1-10 | Unique Index 2 | No | The trap id of the trap that was used to collect the leaf litter. This is going to be an integer |
| *species_id* | integer | Species ID | 0-2999 | | Yes | This is the field that contains the id number for the species and is linked to the **species** table. |
| *tare_leaf* | decimal(9,3) | Leaves: Tare (g) | | | Yes | The tare weight for the leaves in grams |
| *total_leaf* | decimal(9,3) | Leaves: Tare + Sample (g) | | | Yes | The total weight for leaves (including tare) in grams. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *tare_bract* | decimal(9,3) | Bract: Tare (g) | | | Yes | The tare weight for bract in grams. |
| *total_bract* | decimal(9,3) | Bract: Tare + Sample (g) | | | Yes | The total weight for bract (including tare) in grams. |
| *tare_flower* | decimal(9,3) | Flower– Tare | | | Yes | The tare weight for flowers in grams. |
| *total_flower* | decimal(9,3) | Flower: Tare + Sample (g) | | | Yes | The total weight for flowers (including tare) in grams. |
| *tare_fruit* | decimal(9,3) | Fruit: Tare (g) | | | Yes | The tare weight for fruit in grams. |
| *total_fruit* | decimal(9,3) | Fruit: Tare + Sample (g) | | | Yes | The total weight for fruit (including tare) in grams. |
| *tare_wood* | decimal(9,3) | Wood: Tare (g) | | | Yes | The tare weight for wood in grams. |
| *total_wood* | decimal(9,3) | Wood: Tare + Sample (g) | | | Yes | The total weight for wood (including tare) in grams. |
| *tare_misc* | decimal(9,3) | Miscellaneous – Tare (g) | | | Yes | The tare weight for the miscellaneous material in grams |
| *total_misc* | decimal(9,3) | Miscellaneous: Tare + Sample (g) | | | Yes | The total weight for miscellaneous material (including tare) in grams. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.3.6 Zonation*

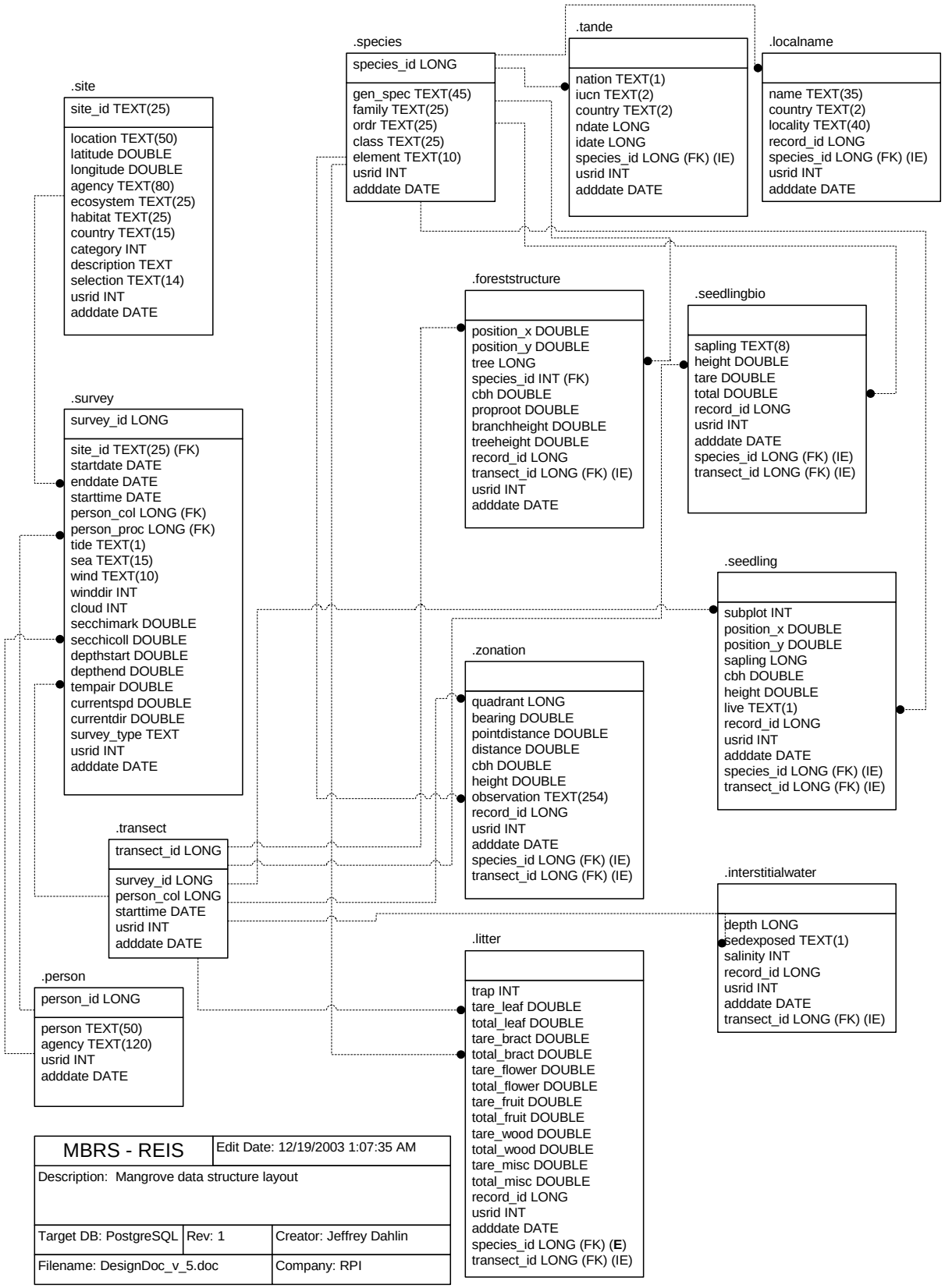
This table contains the data from the Mangrove characterization/zonation data entry form. As with the forest structure, this form is related back to the **transect** form with *transect_id*. It is also linked to the species table on *species_id*.

**Table Name - zonation**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *quadrant* | varchar(3) | Quadrant | I, II, III, IV | Unique Index 2 | No | Quadrant that the tree is located in. |
| *pointdistance* | decimal(9,3) | Point # distance from shore (m) | | Unique Index 2 | No | Distance from the origin of the centerline to the point in meters |
| *species_id* | integer | Species ID | 0-2999 | | No | This is the field that contains the id number for the species and is linked to the **species** table. |
| *distance* | decimal(9,3) | Distance from Center Point (cm) | | | No | The distance in centimeters from the center point of the quadrant. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *cbh* | decimal(9,3) | Circumference at Breast Height (CBH) (cm) | | | No | Circumference at Breast Height. This value is the measure of the circumference of the tree in centimeters. From this the diameter at breast height (DBH) will be calculated. DBH is a common measurement used in forestry. |
| *height* | decimal(9,3) | Total Height (m) | | | No | The height of the tree in meters. |
| *observations* | text | Observations | | | Yes | Free form text field that allows entry of any additional observations made at the location. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

The following page has the entity relationship diagram for mangroves.


.species
species_id LONG
gen_spec TEXT(45)
family TEXT(25)
ordr TEXT(25)
class TEXT(25)
element TEXT(10)
usrid INT
adddate DATE
.tande
nation TEXT(1)
iucn TEXT(2)
country TEXT(2)
ndate LONG
idate LONG
species_id LONG (FK) (IE)
usrid INT
adddate DATE
.localname
name TEXT(35)
country TEXT(2)
locality TEXT(40)
record_id LONG
species_id LONG (FK) (IE)
usrid INT
adddate DATE
.site
site_id TEXT(25)
location TEXT(50)
latitude DOUBLE
longitude DOUBLE
agency TEXT(80)
ecosystem TEXT(25)
habitat TEXT(25)
country TEXT(15)
category INT
description TEXT
selection TEXT(14)
usrid INT
adddate DATE
.foreststructure
position_x DOUBLE
position_y DOUBLE
tree LONG
species_id INT (FK)
cbh DOUBLE
proproot DOUBLE
branchheight DOUBLE
treeheight DOUBLE
record_id LONG
transect_id LONG (FK) (IE)
usrid INT
adddate DATE
.seedlingbio
sapling TEXT(8)
height DOUBLE
tare DOUBLE
total DOUBLE
record_id LONG
usrid INT
adddate DATE
species_id LONG (FK) (IE)
transect_id LONG (FK) (IE)
.survey
survey_id LONG
site_id TEXT(25) (FK)
startdate DATE
enddate DATE
starttime DATE
person_col LONG (FK)
person_proc LONG (FK)
tide TEXT(1)
sea TEXT(15)
wind TEXT(10)
winddir INT
cloud INT
secchimark DOUBLE
secchicoll DOUBLE
depthstart DOUBLE
depthend DOUBLE
tempair DOUBLE
currentspd DOUBLE
currentdir DOUBLE
survey_type TEXT
usrid INT
adddate DATE
.seedling
subplot INT
position_x DOUBLE
position_y DOUBLE
sapling LONG
cbh DOUBLE
height DOUBLE
live TEXT(1)
record_id LONG
usrid INT
adddate DATE
species_id LONG (FK) (IE)
transect_id LONG (FK) (IE)
.zonation
quadrant LONG
bearing DOUBLE
pointdistance DOUBLE
distance DOUBLE
cbh DOUBLE
height DOUBLE
observation TEXT(254)
record_id LONG
usrid INT
adddate DATE
species_id LONG (FK) (IE)
transect_id LONG (FK) (IE)
.transect
transect_id LONG
survey_id LONG
person_col LONG
starttime DATE
usrid INT
adddate DATE
.interstitialwater
depth LONG
sedexposed TEXT(1)
salinity INT
record_id LONG
usrid INT
adddate DATE
transect_id LONG (FK) (IE)
.litter
trap INT
tare_leaf DOUBLE
total_leaf DOUBLE
tare_bract DOUBLE
total_bract DOUBLE
tare_flower DOUBLE
total_flower DOUBLE
tare_fruit DOUBLE
total_fruit DOUBLE
tare_wood DOUBLE
total_wood DOUBLE
tare_misc DOUBLE
total_misc DOUBLE
record_id LONG
usrid INT
adddate DATE
species_id LONG (FK) (E)
transect_id LONG (FK) (IE)
.person
person_id LONG
person TEXT(50)
agency TEXT(120)
usrid INT
adddate DATE
MBRS - REIS
Edit Date: 12/19/2003 1:07:35 AM
Description: Mangrove data structure layout
Target DB: PostgreSQL
Rev: 1
Creator: Jeffrey Dahlin
Filename: DesignDoc_v_5.doc
Company: RPI

## 3.4 Coral Reefs

As with mangroves there are numerous tables that are related to data collected at coral reef sites. The tables for **site** and **survey** are the same as for mangroves. The following tables are in the coral reef monitoring group: **manta**, **adult**, **recruit**, **rover, benthic**, **pointintercept**, and **benthiclut**. This is only a logical assemblage of tables and is not physically set as an assemblage in the data structure. The common links for all of these tables is the *transect_id*. Therefore this group can be easily changed, by adding additional tables, or removing tables as needed. In all the tables that require species name, a *species_id* is stored in the data table that is linked to the **species** table identified above. This provides the flexibility of easily accommodating changes to the scientific or common names, if necessary, in the future. It also eliminates the possibility of misspelling a species name. Below are discussed the tables specific for coral reefs.

### *3.4.1 Manta Tow*

This data table is used for the data collected on the Manta tow spreadsheet. Even though the manta tow is not done on a transect, for compatibility with the other tables in the group an entry has to be made into the **transect** table to be able to enter data into the **manta** table. The *transect* number in the transect table would be 1. Within the **manta** table there cannot be two entries with the same tow number for each transect.

**Table Name - manta**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 2 | No | This field links these records to the transect table. The values are automatically entered by the input form. |
| *tow* | smallint | Tow No. | | Unique Index 2 | No | The number of the tow. This is an integer field. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *corallive* | char(2) | Coral Cover – Live | 0,1,2,3,4,5, -1,-2,-3,-4,-5, +1,+2,+3,+4,+5 | | No | This is a percentage cover of live coral. The following code is used to record the percent coverage:<br>0 – 0<br>1 – 1-10%<br>2 – 11-30%<br>3 – 31-50%<br>4 – 51-75%<br>5 – 76-100%<br>For each category (except 0) a plus(+) or minus(-) is added to denote whether the estimate falls into the upper or lower half of each category. |
| *coraldead* | char(2) | Coral Cover – Dead | 0,1,2,3,4,5, -1,-2,-3,-4,-5, +1,+2,+3,+4,+5 | | No | This is a percentage cover of dead coral. The following code is used to record the percent coverage:<br>0 – 0<br>1 – 1-10%<br>2 – 11-30%<br>3 – 31-50%<br>4 – 51-75%<br>5 – 76-100%<br>For each category (except 0) a plus(+) or minus(-) is added to denote whether the estimate falls into the upper or lower half of each category. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *softcoral* | char(2) | Cover – SC | 0,1,2,3,4,5, -1,-2,-3,-4,-5, +1,+2,+3,+4,+5 | | No | This is a percentage cover of soft coral. The following code is used to record the percent coverage:<br>0 – 0<br>1 – 1-10%<br>2 – 11-30%<br>3 – 31-50%<br>4 – 51-75%<br>5 – 76-100%<br>For each category (except 0) a plus(+) or minus(-) is added to denote whether the estimate falls into the upper or lower half of each category.. |
| *algae* | char(2) | Algae | 0,1,2,3,4,5, -1,-2,-3,-4,-5, +1,+2,+3,+4,+5 | | No | This is a percentage cover of algae The following code is used to record the percent coverage:<br>0 – 0<br>1 – 1-10%<br>2 – 11-30%<br>3 – 31-50%<br>4 – 51-75%<br>5 – 76-100%<br>For each category (except 0) a plus(+) or minus(-) is added to denote whether the estimate falls into the upper or lower half of each category.. |
| *features* | text | Other Features | | | Yes | Free form comments field for other features of the reef. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.4.2 Adult fish*

The **adult** table records the information from the adult fish entry form. On the adult fish data entry form in the SMP Manual there is a row for each species that is to be counted. However, in the table only the species that were sighted and counted on the transect being surveyed are entered. If a species on the data sheet does not have a count associated with it for that survey and transect it is not entered into the table. As with all the other tables this table relates back to the **transect** table on *transect_id*.

**Table Name - adult**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *species_id* | integer | Species ID | 0-2999 | Unique Index 2 | No | This is the field that contains the id number for the species and is linked to the **species** table. |
| *cm0_5* | integer | 0-5 cm | 0-500 | | Yes | Count for the size range from 0 to 5 cm for the species. This is an integer value and may be |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| | | | | | | left blank. Blanks will be considered 0. |
| *cm6_10* | integer | 6-10 cm | 0-500 | | Yes | Count for the size range from 6 to 10 cm for the species. This is an integer value and may be left blank. Blanks will be considered 0. |
| *cm11_20* | integer | 11-20 cm | 0-500 | | Yes | Count for the size range from 11 to 20 cm for the species. This is an integer value and may be left blank. Blanks will be considered 0. |
| *cm21_30* | integer | 21-30 cm | 0-500 | | Yes | Count for the size range from 21 to 30 cm for the species. This is an integer value and may be left blank. Blanks will be considered 0. |
| *cm31_40* | integer | 31–40 cm | 0-500 | | Yes | Count for the size range from 31 to 40 cm for the species. This is an integer value and may be left blank. Blanks will be considered 0. |
| *cm41* | integer | > 40 cm | 0-500 | | Yes | Count for the size range greater than 40 cm for the species. This is an integer value and may be left blank. Blanks will be considered 0. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.4.3 Fish Recruitment*

The fish recruitment table **recruit** is similar to the **adult** table in that only the species that have count information are entered into the database. A separate record in the **transect** table needs to be created for each column on the datasheet. After entering a transect record users only need to enter species and count information into this table. This table is linked to the **transect** table on *transect_id*.

**Table Name - recruit**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *species_id* | integer | Species ID | 0-2999 | Unique Index 2 | No | This is the field that contains the id number for the species and is linked to the **species** table. |
| *fishcount* | integer | Count (Trans # on the datasheet) | 0-500 | | No | The number of fish of the given species below the maxTL found on this transect. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.4.4 Rover Diver*

The rover diver table **rover** is set up similar to the **recruit** table. Only species that are recorded on the survey are entered into the database. This setup minimizes the size of the table and allows for easy expansion of the species list that can be recorded in the table. Even though the rover diver is not based on a transect, for compatibility with the other tables in the group, a transect record still needs to be created before data can be entered into this table. The **rover** table is linked to the **transect** table on *transect_id*.

**Table Name - rover**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *species_id* | integer | Species ID | 0-2999 | Unique Index 2 | No | This is the field that contains the id number for the species and is linked to the species table. |
| *abundance* | char(1) | Abundance Codes | S,F,M,A | | No | A one letter code for the number of individuals seen for the species. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.4.5 Point Intercept*

The **pointintercept** table corresponds to the data form "Point Intercept Transect Data Entry Form" in the SMP Manual. The information in this table identifies the various types of substrate on the transect. The identifier in this table is the column *benthic_id*. This is an ID number that is linked to the table **benthiclut**, which contains the names for the substrate. The information in this table is entered by transect, with the transect number being recorded in the transect table. This table is linked to the **transect** table on *transect_id.*

**Table Name - pointintercept**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *benthic_id* | integer | Benthic Components | | Unique Index 2 | No | This is an ID that links to the benthic component name in the **benthiclut** table. |
| *benthcount* | integer | Occurrence (Trans # on the data sheet) | 0-120 | | No | The number of occurrences for this benthic type along the transect. This is an integer value, theoretically less than or equal to 120. Based on the study design there are only 120 points along each transect where the substrate will be recorded. |
| *comments* | text | Comments | | | Yes | A free form text field for any observations made for the benthic component or transect. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| | | | | | | web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.4.6 Benthiclut*

This table is a lookup table with a list of all of the possible benthic components that would be used in the point intercept data sheet. It is related to the **pointintercept** table on *benthic_id*.

**Table Name - benthiclut**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *benthic_id* | integer | Benthic ID | | Unique Index 1 | No | This field links these records to the **pointintercept** table. |
| *component* | varchar(35) | Benthic Component | | | No | This is the name of the substrate type. |
| *componen_spt* | varchar(35) | Benthic Component | | | No | This is the Spanish name of the substrate type. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |

| | | | | | | |
|---|---|---|---|---|---|---|
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.4.7 Benthic Coral*

This table corresponds to the date form Benthic Data Entry Form. This table records the information for the various corals found along the survey transect. The table design has one row per coral record similar to the data entry form. In the SMP Manual the data sheet has a column for disease in which a code is entered for the disease. The data table uses nine columns for this information. There is a separate column for each type of disease. These are Boolean fields, meaning that all they hold is a true or false value. If the coral has the disease the box is checked and the value in the field is set to true. If the disease is not present the box is left blank and the value in the field is blank.

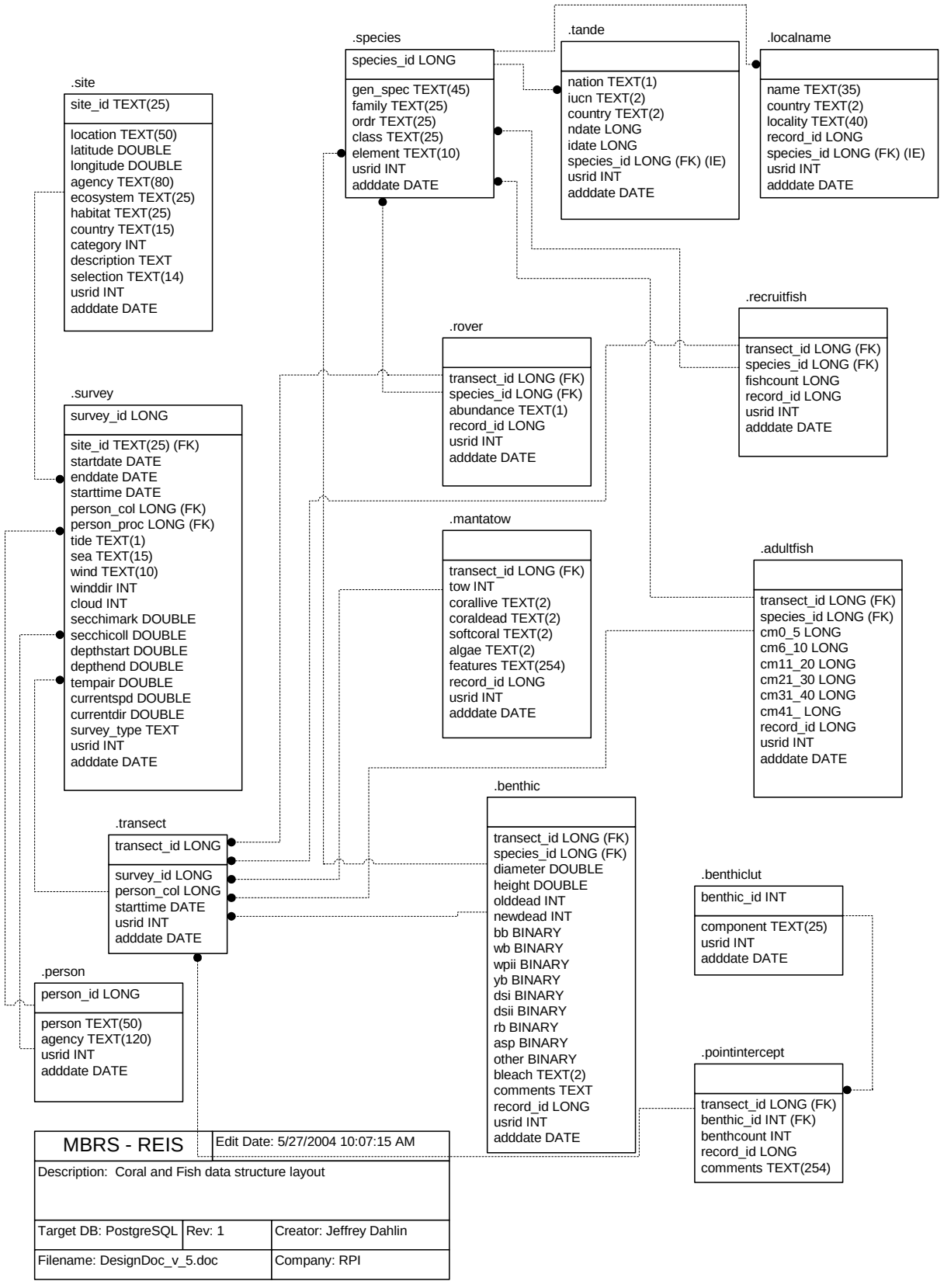
**Table Name - benthic**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *species_id* | integer | Species ID | 0-2999 | Index 2 | No | This is the field that contains the id number for the species and is linked to the **species** table. |
| *diameter* | integer | Max Diameter (cm) | 0-500 | | No | The diameter, in centimeters, of the coral head being measured |
| *height* | integer | Max Height (cm) | 0-500 | | No | The height, in centimeters, of the coral head being measured |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *olddead* | integer | % Dead Old | 0-100 | | No | Percent of the coral that has been long dead. This is an integer from 1 to 100. |
| *newdead* | integer | % Dead Recent | 0-100 | | No | Percent of the coral that has recently died. This is an integer from 1 to 100 |
| *bb* | binary | Black Band Disease | | | Yes | This is a yes/no field. Yes if it has Black Band Disease and No if it does not have the disease. |
| *wb* | binary | White Band Disease | | | Yes | This is a yes/no field. Yes if it has White Band Disease and No if it does not have the disease. |
| *wpii* | binary | White Plague-II | | | Yes | This is a yes/no field. Yes if it has White Plague-II and No if it does not have the disease. |
| *yb* | binary | Yellow-Blotch Disease | | | Yes | This is a yes/no field. Yes if it has Yellow-Blotch Disease and No if it does not have the disease. |
| *dsi* | binary | Dark Spots Disease I | | | Yes | This is a yes/no field. Yes if it has Dark Spots Disease I and No if it does not have the disease. |
| *dsii* | binary | Dark Spots Disease II | | | Yes | This is a yes/no field. Yes if it has Dark Spots Disease II and No if it does not have the disease. |
| *rb* | binary | Red Band Disease | | | Yes | This is a yes/no field. Yes if it has Red Band Disease and No if it does not have the disease. |
| *asp* | binary | Aspergillosis | | | Yes | This is a yes/no field. Yes if it has Aspergillosis and No if it does not have the disease. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *other* | binary | Other | | | Yes | This is a yes/no field. Yes for all other "unconfirmed pathogen produced" diseases and No if it does not have other disease. |
| *bleach* | char(2) | Bleached | N,P,PB,BL | | No | code for level of bleaching<br>N – No Bleaching<br>P – Pale<br>PB – Partly Bleached<br>BL - Bleached |
| *comments* | text | Comments | | | Yes | Free form comments field for other features of the coral. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

The following page has the entity relationship diagram for corals and fish.


.species
species_id LONG
gen_spec TEXT(45)
family TEXT(25)
ordr TEXT(25)
class TEXT(25)
element TEXT(10)
usrid INT
adddate DATE
.tande
nation TEXT(1)
iucn TEXT(2)
country TEXT(2)
ndate LONG
idate LONG
species_id LONG (FK) (IE)
usrid INT
adddate DATE
.localname
name TEXT(35)
country TEXT(2)
locality TEXT(40)
record_id LONG
species_id LONG (FK) (IE)
usrid INT
adddate DATE
.site
site_id TEXT(25)
location TEXT(50)
latitude DOUBLE
longitude DOUBLE
agency TEXT(80)
ecosystem TEXT(25)
habitat TEXT(25)
country TEXT(15)
category INT
description TEXT
selection TEXT(14)
usrid INT
adddate DATE
.recruitfish
transect_id LONG (FK)
species_id LONG (FK)
fishcount LONG
record_id LONG
usrid INT
adddate DATE
.rover
transect_id LONG (FK)
species_id LONG (FK)
abundance TEXT(1)
record_id LONG
usrid INT
adddate DATE
.survey
survey_id LONG
site_id TEXT(25) (FK)
startdate DATE
enddate DATE
starttime DATE
person_col LONG (FK)
person_proc LONG (FK)
tide TEXT(1)
sea TEXT(15)
wind TEXT(10)
winddir INT
cloud INT
secchimark DOUBLE
secchicoll DOUBLE
depthstart DOUBLE
depthend DOUBLE
tempair DOUBLE
currentspd DOUBLE
currentdir DOUBLE
survey_type TEXT
usrid INT
adddate DATE
.mantatow
transect_id LONG (FK)
tow INT
corallive TEXT(2)
coraldead TEXT(2)
softcoral TEXT(2)
algae TEXT(2)
features TEXT(254)
record_id LONG
usrid INT
adddate DATE
.adultfish
transect_id LONG (FK)
species_id LONG (FK)
cm0_5 LONG
cm6_10 LONG
cm11_20 LONG
cm21_30 LONG
cm31_40 LONG
cm41_ LONG
record_id LONG
usrid INT
adddate DATE
.benthic
transect_id LONG (FK)
species_id LONG (FK)
diameter DOUBLE
height DOUBLE
olddead INT
newdead INT
bb BINARY
wb BINARY
wpii BINARY
yb BINARY
dsi BINARY
dsii BINARY
rb BINARY
asp BINARY
other BINARY
bleach TEXT(2)
comments TEXT
record_id LONG
usrid INT
adddate DATE
.transect
transect_id LONG
survey_id LONG
person_col LONG
starttime DATE
usrid INT
adddate DATE
.benthiclut
benthic_id INT
component TEXT(25)
usrid INT
adddate DATE
.person
person_id LONG
person TEXT(50)
agency TEXT(120)
usrid INT
adddate DATE
.pointintercept
transect_id LONG (FK)
benthic_id INT (FK)
benthcount INT
record_id LONG
comments TEXT(254)
MBRS - REIS
Edit Date: 5/27/2004 10:07:15 AM
Description: Coral and Fish data structure layout
Target DB: PostgreSQL
Rev: 1
Creator: Jeffrey Dahlin
Filename: DesignDoc_v_5.doc
Company: RPI

## 3.5 Seagrasses

There are three tables associated with seagrasses, in addition to the **site** and **survey** tables. The seagrass tables were more complex to set up and data entry may be a little more complex. However with this structure, it should facilitate querying, minimize redundancy, and reduce errors.

### *3.5.1 Seagrass Growth*

The seagrass growth table contains the data from the Seagrass Growth Data entry form. For this table the **survey** table should have a start date and end date. As with all other tables this table is linked backed to the **transect** table with the *transect_id*. The quadrat number from the seagrass growth data sheet is recorded in this table and not in the **transect** table, since there is only one set of data entered for each quadrat. An entry in the **transect** table is still required for compatibility with other tables in the group. The transect table links the **sggrowth** table back to the **survey** table. The transect number given in the **transect** table should be 1. None of the calculated values such as areal productivity, turnover, or biomass of the plants are stored in the data table. These are all calculated by the database, as required for view or printing reports, based on the input values.

**Table Name - sggrowth**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *quadrat* | smallint | Quadrat # | 1-6 | Unique Index 2 | No | Numeric value for quadrat within the site. |
| *newtarewt* | decimal (9,3) | Tare Wt. – New Leaves (g) | | | Yes | The tare weight for new leaves in grams. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *newgrosswt* | decimal (9,3) | Gross Wt. – New Leaves (g) | | | Yes | The gross weight for new leaves in grams. From the tare and gross the net weight can be calculated. |
| *oldtarewt* | decimal (9,3) | Tare Wt. – Old Leaves (g) | | | Yes | The tare weight for old leaves in grams. |
| *oldgrosswt* | decimal (9,3) | Gross Wt. – Old Leaves (g) | | | Yes | The gross weight for old leaves in grams. From the tare and gross the net weight can be calculated. |
| *standtarewt* | decimal (9,3) | Tare Wt. – Old Standing Crop (g) | | | Yes | The tare weight for old standing crop in grams. |
| *standgrosswt* | decimal (9,3) | Gross Wt. – Old Standing Crop (g) | | | Yes | The gross weight for old standing crop in grams. From the tare and gross the net weight can be calculated. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.5.2 Seagrass Biomass*

The seagrass biomass table is based on the seagrass biomass entry form. There is one record in the table for each core replicate taken. As with the seagrass growth table, a record needs to be entered into the **transect** table for the seagrass biomass. If two stations are sampled for each site, the entries in the **transect** table would represent the stations. None of the calculated fields, such as Ratio A:B are stored in the table. These are all calculated by the database, as required for viewing or printing reports, based on the input values.

**Table Name - sgbiomass**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |
| *replicate* | integer | Core Replicate | | Unique Index 2 | No | The core replicate number. |
| *diameter* | decimal (9,3) | Core Diameter (cm) | | | No | The diameter of the core in centimeters. |
| *depth* | decimal (9,3) | Core Depth (cm) | | | Yes | Depth of core in centimeters. |
| *livingshoots* | integer | # Living shoots/Core | | | Yes | Number of living shoots in the core. |
| *tgrntare* | decimal (9,3) | Thalassia Green Leaves Tare Wt. (g) | | | Yes | The tare weight for the green leaves in grams. |
| *tgrngross* | decimal (9,3) | Thalassia Green Leaves Gross Wt. (g) | | | Yes | The gross weight for the green leaves in grams. From the tare and gross the net weight can be calculated. |
| *tshttare* | decimal (9,3) | Thalassia | | | Yes | The tare weight for the short shoots in grams. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| | | Short Shoots Tare Wt. (g) | | | | |
| *tshtgross* | decimal (9,3) | Thalassia Short Shoots Gross Wt. (g) | | | Yes | The gross weight for the short shoots in grams. From the tare and gross the net weight can be calculated. |
| *trhztare* | decimal (9,3) | Thalassia Rhizomes Tare Wt. (g) | | | Yes | The tare weight for the rhizomes in grams. |
| *trhzgross* | decimal (9,3) | Thalassia Rhizomes Gross Wt. (g) | | | Yes | The gross weight for the rhizomes in grams. From the tare and gross the net weight can be calculated. |
| *troottare* | decimal (9,3) | Thalassia Roots Tare Wt. | | | Yes | The tare weight for the roots in grams. |
| *trootgross* | decimal (9,3) | Thalassia Roots Gross Wt. (g) | | | Yes | The gross weight for the roots in grams. From the tare and gross the net weight can be calculated. |
| *tdeadtare* | decimal (9,3) | Thalassia Dead Tissue Tare Wt. | | | Yes | The tare weight for the dead tissue in grams. |
| *tdeadgross* | decimal (9,3) | Thalassia Dead Tissue Gross Wt. (g) | | | Yes | The gross weight for the dead tissue in grams. From the tare and gross the net weight can be calculated. |
| *ogrntare* | decimal (9,3) | Other Grass Green Tissue Tare Wt. (g) | | | Yes | The tare weight for green tissue from other grasses in grams. |
| *ogrngross* | decimal (9,3) | Other Grass Green Tissue Gross Wt. (g) | | | Yes | The gross weight for the green tissue from other grasses in grams. From the tare and gross the net weight can be calculated. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *ongrntare* | decimal (9,3) | Other Grass Nongreen Tissue Tare Wt. (g) | | | Yes | The tare weight for the nongreen tissue from other grasses in grams. |
| *ongrngross* | decimal (9,3) | Other Grass Nongreen Tissue Gross Wt. (g) | | | Yes | The gross weight for the nongreen tissue from other grasses in grams. From the tare and gross the net weight can be calculated. |
| *fatare* | decimal (9,3) | Fleshy Algae Tare Wt. (g) | | | Yes | The tare weight for fleshy algae in grams. |
| *fagross* | decimal (9,3) | Fleshy Algae Gross Wt. (g) | | | Yes | The gross weight for the fleshy algae in grams. From the tare and gross the net weight can be calculated. |
| *caabvtare* | decimal (9,3) | Calcareous Algae Above Ground Tare Wt. (g) | | | Yes | The tare weight for above ground calcareous algae in grams. |
| *caabvgross* | decimal (9,3) | Calcareous Algae Above Ground Gross Wt. (g) | | | Yes | The gross weight for the above ground calcareous algae in grams. From the tare and gross the net weight can be calculated. |
| *cablwtart* | decimal (9,3) | Calcareous Algae Below Ground Tare Wt. (g) | | | Yes | The tare weight for the below ground calcareous algae in grams. |
| *cablwgross* | decimal (9,3) | Calcareous Algae Below Ground Gross Wt. (g) | | | Yes | The gross weight for the below ground calcareous algae in grams. From the tare and gross the net weight can be calculated. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

### *3.5.3 Seagrass Leaf Area Index*

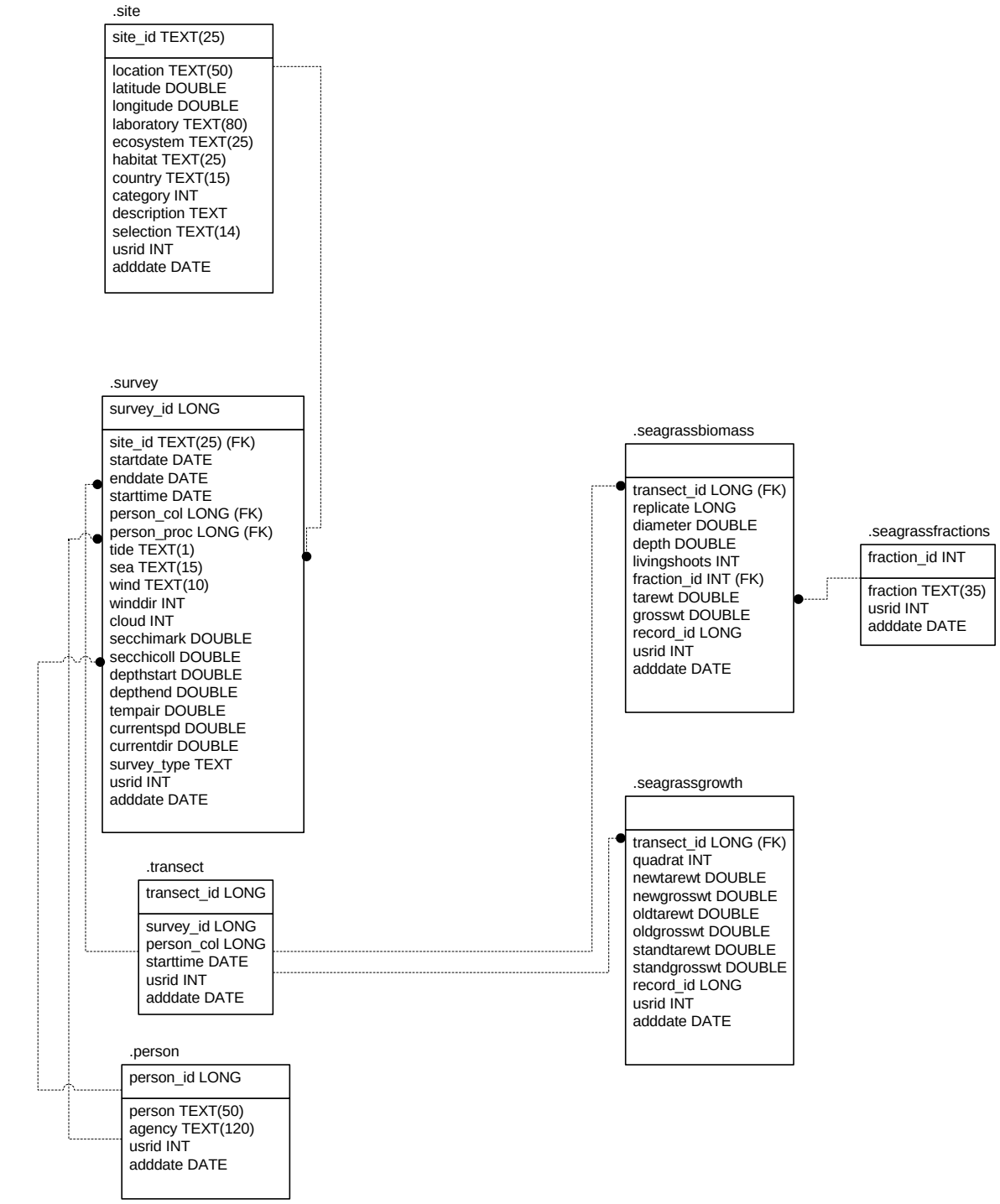
The seagrass leaf area index table (**sglai)** is used to store the information from the leaf area index form. This table is linked back to the **survey** table through the **transect** table. There should be one entry in the **transect** table for each quadrat that is sampled. The **sglai** table has one record for each leaf that is measured. The area for each leaf is not entered. This value is calculated by the database.

**Table Name - sglai**

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *transect_id* | integer | Transect ID | | Unique Index 2 | No | This field links these records to the **transect** table. The values are automatically entered by the input form. |

| Column Name | Type | Descriptive name | Valid Values | Index Column | Allow Nulls | Description |
|---|---|---|---|---|---|---|
| *shoot* | integer | Shoot Number | 1-6 | Unique Index 2 | No | The number of the shoot that is being measured. |
| *leaf* | integer | Leaf Number | 1-6 | Unique Index 2 | No | The number of the leaf on the associated shoot that is being measured. |
| *tip* | char (1) | Round Tip | Y,N,S | | Yes | Indicate whether the tip of the leaf is rounded.<br>Y – Yes<br>N – No<br>S – Si |
| *epis* | decimal (9,3) | Length to epis (cm) | | | Yes | The length in centimeters from the base of the leaf to the first occurrence of epiphytes. If epiphytes cover the entire leaf all the way down to the base this value would be 0. If there are no epiphytes this value would be the same as the leaf length. |
| *length* | decimal (9,3) | Length (cm) | | | Yes | The length of the leaf in centimeters. |
| *width* | decimal (9,3) | Width (cm) | | | Yes | The width of the leaf in centimeters. |
| *record_id* | integer | Record ID | | Unique Index 1 | No | Unique identifier for the record. This is entered automatically by the database and is used for internal operations only. The user does not see this value. |
| *usrid* | integer | User ID | | | No | This field is populated automatically by the web interface with the user id. The id links to the login table so the user name can be displayed. This field is used to keep track of who enters the data. |
| *adddate* | date | Date | | | No | This field is populated automatically by the database program. It is used to identify when the record was added. |

The following page has the entity relationship diagram for seagrasses.




| MBRS - REIS | Edit Date: 12/19/2003 1:12:56 AM | |
|---|---|---|
| Description: Seagrass data structure layout | | |
| Target DB: PostgreSQL | Rev: 1 | Creator: Jeffrey Dahlin |
| Filename: DesignDoc_v_5.doc | | Company: RPI |

## APPENDIX A

## DATA TYPES

| Type Name | Description |
|---|---|
| bigint | signed eight-byte integer (--9223372036854775808 to 9223372036854775807) |
| bigserial | autoincrementing eight-byte integer |
| bit | fixed-length bit string |
| bit varying(*n*) | variable-length bit string |
| boolean | logical Boolean (true/false) |
| bytea | binary data character(*n*) |
| char(*n*) | fixed-length character string |
| date | calendar date (year, month, day) |
| double precision | double precision floating-point number (15 decimal digits) |
| integer | signed four-byte integer (-2147483648 to +2147483647) |
| interval(*p*) | general-use time span |
| decimal [ (*p*, *s*) ] | exact numeric with selectable precision (*p*) and decimal places (*s*). |
| real | single precision floating-point number (6 decimal digits) |
| smallint | signed two-byte integer (-32768 to +32767) |
| serial | autoincrementing four-byte integer |
| text | variable-length character string |
| time | time of day |
| timetz | time of day, including time zone |
| timestamp | date and time |
| timestamptz | date and time, including time zone |
| varchar(*n*) | variable-length character string |