Calfed Smolts Database Dictionary

Tables

-Important tables: These tables will be the main frame of the database, with almost any query needing to go through one or several of these. It is important to familiarize yourself with these tables as much as possible.

1. **Deployments**: The primary goal of this table is to keep track of which VR2 monitor is found where and between what dates. It also makes note of details that are specific to each individual deployment. The "Monitor_Locations" table has more details on the deployment sites themselves. The reason for which "Deployment" details aren't merged with the "Monitor_Locations" table is because some monitors may get moved or replaced at monitor locations, and so we can never assume one location will always have the same VR2.
2. **Detections_With_Loc_Code**: This is the important table that links the data retrieved from the monitors to all of the other data in this database. It is essentially a linked, updating query-table that has all of the fields of "Detections" but with the added bonus of telling us the location code of where the detections were heard. The table takes the data from the "Detections" table, refers to the "Deployments" table to see where the monitor was located at the time of the detection, and then tells us that location. For this step to work perfectly, we will always need to have up-to-date deployments, as well as "Monitor_Locations" data for each deployment.
3. **Monitor_Locations**: This table holds all of the details about all of the monitor locations that ever have existed. Specifically, if you want to find out where to find a monitor, the GPS coordinates can be found here as well as the closest boat ramp for access. This table is linked to the "Deployments" table by the Location name. By referring to the "Deployements" table, one can find out if the location is currently in use.
4. **TagID_Release_Date**: The purpose of this table is to link up all the other tables to the "Release_Details" table by having an entry for every fish ever released and giving us its release date. The release date is what links the fish to a mass release included in "Release_Details.
5. **Tagged_Fish**: This is the main table to which any table dealing with the tagging portion of the project is linked. It has an entry for every fish ever tagged, and includes basic fish details as well as the time and quality of the surgery performed. Beyond that, it is linked to many other tables that will have any other data that isn't individual-specific.

-Important Auxiliary Tables: These tables are not as important and will not be used as regularly as the "Important Tables" but will still be necessary for more specific queries. It would be a good idea to familiarize yourself with these tables too if you plan on doing a lot of work with the database.

6. **Detections**: This table is where all of the VR2 data gets entered. It lists every individual tag detection every heard on a monitor and by telling us the time, tag ID number, monitor ID number and channel the detection was on. However, it does not tell you where the monitor was at the time, so this table is almost useless on its own. One should use Detections_With_Loc_Code for all querying because it is the exact same table as Detections but with the Location also included.

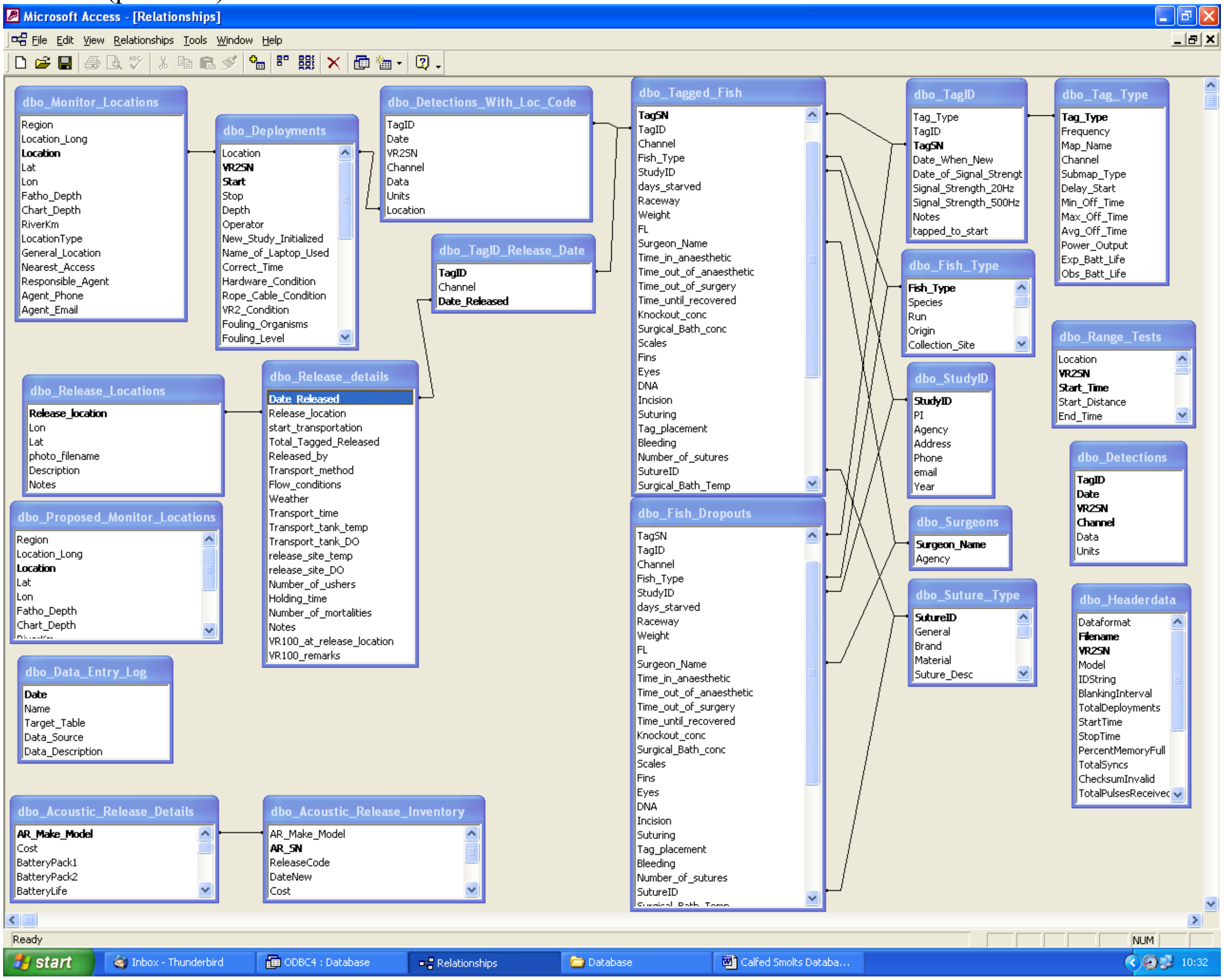
7. **Fish_Type**: The purpose of this table is to simplify the "Tagged_Fish" table. This table is linked to the "Tagged_Fish" table by a fish type code. In this table, the details on the kind of fish are revealed, including species, stock and origin.
8. **Headerdata**: This is the other half of the data retrieved form the VR2s that is automatically loaded into the database. This includes all other data from the monitors not found in the "Detections" table.
9. **Mobile_Detections**: This table includes detections that were found is mobile surveys of the watershed, usually with a VR100. The fields in this table match the fields in the data files that the VR100 outputs. This way, data can be pasted into the table without interpretation. These detections can act as normal monitor detections for query writing as long as each mobile detection includes an accurate latitude and longitude.
10. **Release_Details**: This table contains all the details about each release of tagged fish. It is linked to the rest of the database by the table "TagID_Release_Date" which lists a release date/time for each fish. Since there are many instances of reused release sites in this table, we used release date/time as the unique field.
11. **Release_Locations**: This table includes all the details of where the release location is found. It is linked to the release details by the field "release location". It is important to have good notes on the location of these releases as they act like monitors, since any fish released there can be considered a "detection".
12. **StudyID**: The purpose of this table is to simplify the "Tagged_Fish" table. It includes every group that has tagged fish in the database and lists their agency and contact details. This table is linked to the "Tagged_Fish" table using the "StudyID" field. Thus, every fish in the "Tagged_Fish" table has an associated studyID that refers you to this table so you can see what group was in charge of that fish.
13. **TagID**: This intention of this table is to have one entry for every tag ever used in this project, and includes both its serial number and tag ID. Also, details such as the signal strength tests filenames have been included for quick reference.
14. **Tag_Type**: This table has an entry not only for every different tag type, but also for every way a tag has been programmed for the project. For example, there is an entry for a V7-2L tag with a 5 day delay start, and a separate entry for a V7-2L tag with a 7 day delay start. This table is linked up to the "Tagged_Fish" table by means of the "Tag_Type" field. This means that every fish in the "Tagged_Fish" table has an associated tag type, and by referring back to this table, one can discover more details on the tag that was implanted in that fish.
15. **Temp_Logger_Data**: This table holds all of the temperature readings recorded by the many temperature loggers that are attached to different monitors throughout the watershed. This data is organized a lot like the detection data, with a unique code for each temp logger deployment that links back to the "Temp_Logger_Deployment" table for more information.
16. **Temp_Logger_Deployment**: This table gives extra information on the deployment times and location of the temperature loggers.
17. **Temp_Detectiondata**: This table is under construction, but will like up the temperature data with each specific fish detection, where available.

-Satellite Tables: These tables are, in most part, not even part of the main database frame and do not relate to other tables. They are mainly used to keep track of NMFS equipment and side projects. They will not be needed at all for the majority of the database users. One should take a look at each of these tables once to determine their use in case they are ever needed, but they probably won't be. In fact, for some database users, these tables are not even available.

18. **Acoustic_Release_Details**: This table lists the different acoustic releases that we use and their parameters.
19. **Acoustic_Release_Inventory**: This table lists every individual acoustic release that we have used and has details on each one.
20. **Data_Entry_Log**: If the data managers make any addition or modification to the data or the database layout, the persons name, the change made and the date is noted here.
21. **Fish_Dropouts**: This table is for any fish that was part of a tag experiment and was never released or any fish that was truly tagged but died before release. The fields of this table are identical to the fields of the "Tagged_Fish" table so that one can simply cut out an entry from that table and enter it into "Fish_Dropouts" in the event of mortality.
22. **Lost_Recovered_VR2s**: This table is simply a table to keep track of lost vr2s and if they were ever recovered. This way we can have a list to check when we see any gaps of data at a monitor location. Also, we can try and find trends in the lost vr2s; maybe a certain kind of deployment or depth. Information on the possible recovery of the monitor is including, like who found it, where, when, etc.
23. **Mobile_Surveys**: This table is the equivalent of the purpose of the "deployments" table to the "detections" table, except here it serves to better describe the entries in the "Mobile_Detections" table. It includes the start and stop times of surveys, the general location, operator and VR100 serial number. This table is linked to the "Mobile_Detections" table by a "SurveyID" field.
24. **Proposed_Monitor_Locations**: This table simply holds the same fields as "Monitor_Locations" but includes monitor locations that have yet to be installed.
25. **Range_Tests**: This table includes all of the details of each range test, including start and stop times, locations and conditions.
26. **Surgeons**: This small table simply links a surgeon to the group he or she works for. This table is linked to the "Tagged_Fish" table by means of the "Surgeon Name" (each fish has its tagger's name in its entry.)
27. **Suture_Type**: This table lists the specifications about the suture types used, and where to buy them if more is needed. This table is linked to the "Tagged_Fish" by the "SutureID" field.
28. **Temp_Logger_Inventory**: This table includes all the auxiliary information on each temperature logger unit, including its status (on shelf, in use, lost…)
29. **VR2 Inventory**: This table includes all the VR2s belonging to the consortium grouped by VR2 serial number and includes such details as when they were purchased and using which budget, their cost, and their status (if they are in the field, sitting on a shelf or lost). This table can be useful if one wants to figure out where all the monitors are at one time.

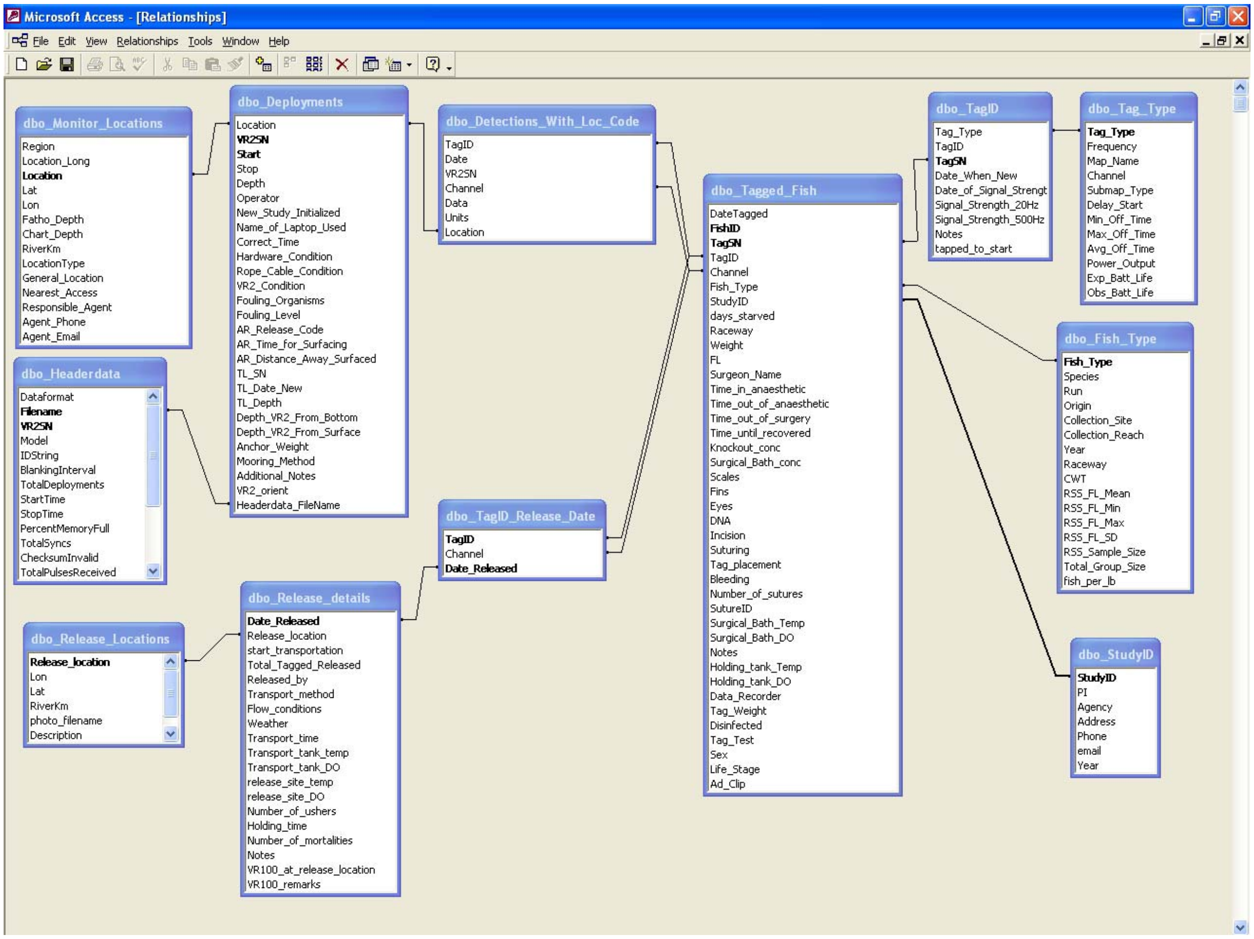
<u>Relationships:</u>

An important feature that one should utilize in this access database is the ability to link tables together. The tables are set up so that one can link many of the tables together, but you will need to actually make the linkages yourself. Linked tables help in many ways, including making query-writing easier and simple table viewing way more informative and efficient. To make the linkages, go to the "relationships" tab on the main access toolbar. One can then add as many of the tables to the relationships window, and then try to link them all up. This will get a little cluttered though, and many linkages that one could make will never be of much help. If one were so inclined, the following picture shows all the possibly useful linkages there are to be made (picture 1).



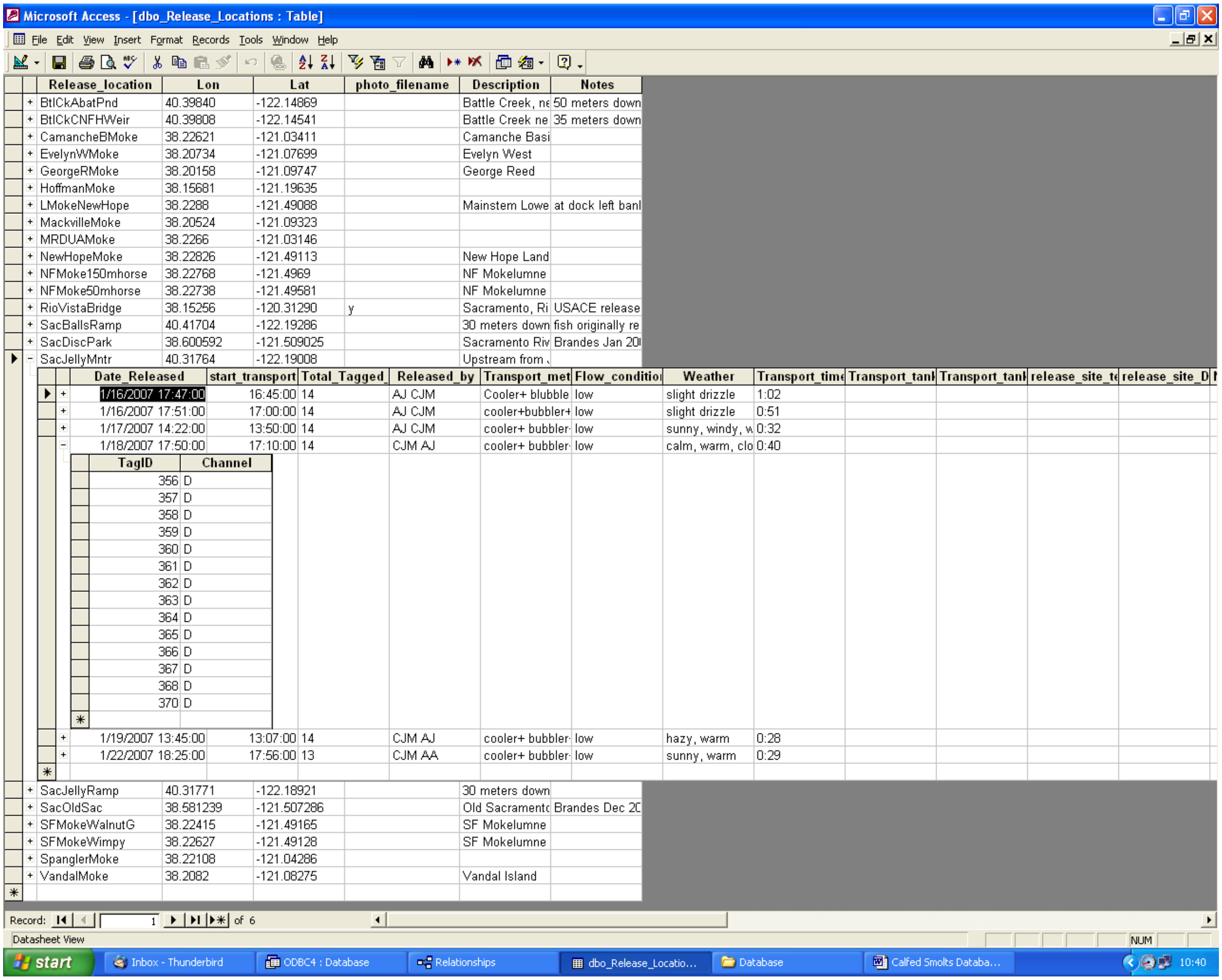
Picture 1: The extent to which relationships could be made in the database

The really important linkages to be made are shown in the second picture (picture 2). The tables that need to be added to the relationships window are as follows: Tagged_Fish, Detections_With_Loc_Code, Deployments, Monitor_Locations, TagID, Tag_Type, Fish_Type, TagID_Release_Date, Release_Details, Release_Locations, and StudyID. The linkages seen in these pictures can be made by clicking on a field name and dragging to the same field name in another table. It will then ask you if you wish to create a linkage. At the bottom of this window, it will describe the linkage as an "indeterminate", "one-to-one" or "one-to-many" linkage. One should always shoot for a one-to-many linkage if possible, because this allows the viewing of tables inside tables (reviewed later). A one-to-many linkage can only be created when one table lists the possible entries for a field, and the other table has a bunch of entries of all the sorts for that field.



Picture 2: The most important relationships to be made for data viewing and querying

If a "one-to-many" connection is made between tables, the "one" table can be used as an interface to access the "many" table. If you go to the "one" table, you will see on the left hand side of each entry a little "+". If you click on this, it will expand and show you all of that entry's linked information to the "many" table. The information will appear in a window inside the "one" table, just below the entry that you expanded. In some situations, you will have several "one-to-many" tables linked sequentially, and from the first "one" table, you could access data from many tables. In the example below (picture 3), you will see that from the "Release_Locations" table I have accessed all of the different releases from the location "SacJellyMntr" by clicking on the "+". By expanding here, I was sent to the "Release_Details" data. By expanding again by clicking on the "+" in the new window, I can now access data from the "TagID_Release_Date" giving me, in this case, the TagID number for every fish released on the 18th of January at that same location, "SacJellyMntr".


Microsoft Access - [dbo_Release_Locations : Table]
File Edit View Insert Format Records Tools Window Help

	Release_location	Lon	Lat	photo_filename	Description	Notes
+	BtlCkAbatPnd	40.39840	-122.14869		Battle Creek, ne	50 meters down
+	BtlCkCNFHWeir	40.39808	-122.14541		Battle Creek ne	35 meters down
+	CamancheBMoke	38.22621	-121.03411		Camanche Basi	
+	EvelynWMoke	38.20734	-121.07699		Evelyn West	
+	GeorgeRMoke	38.20158	-121.09747		George Reed	
+	HoffmanMoke	38.15681	-121.19635			
+	LMokeNewHope	38.2288	-121.49088		Mainstem Lowe	at dock left banl
+	MackvilleMoke	38.20524	-121.09323			
+	MRDUAMoke	38.2266	-121.03146			
+	NewHopeMoke	38.22826	-121.49113		New Hope Land	
+	NFMoke150mhorse	38.22768	-121.4969		NF Mokelumne	
+	NFMoke50mhorse	38.22738	-121.49581		NF Mokelumne	
+	RioVistaBridge	38.15256	-120.31290	y	Sacramento, Ri	USACE release
+	SacBallsRamp	40.41704	-122.19286		30 meters down	fish originally re
+	SacDiscPark	38.600592	-121.509025		Sacramento Riv	Brandes Jan 20
−	SacJellyMntr	40.31764	-122.19008		Upstream from .	

	Date_Released	start_transport	Total_Tagged_	Released_by	Transport_met	Flow_conditio	Weather	Transport_time	Transport_tank	Transport_tank	release_site_t	release_site_D
+	1/16/2007 17:47:00	16:45:00	14	AJ CJM	Cooler+ blubble	low	slight drizzle	1:02				
+	1/16/2007 17:51:00	17:00:00	14	AJ CJM	cooler+bubbler+	low	slight drizzle	0:51				
+	1/17/2007 14:22:00	13:50:00	14	AJ CJM	cooler+ bubbler	low	sunny, windy, w	0:32				
−	1/18/2007 17:50:00	17:10:00	14	CJM AJ	cooler+ bubbler	low	calm, warm, clo	0:40				

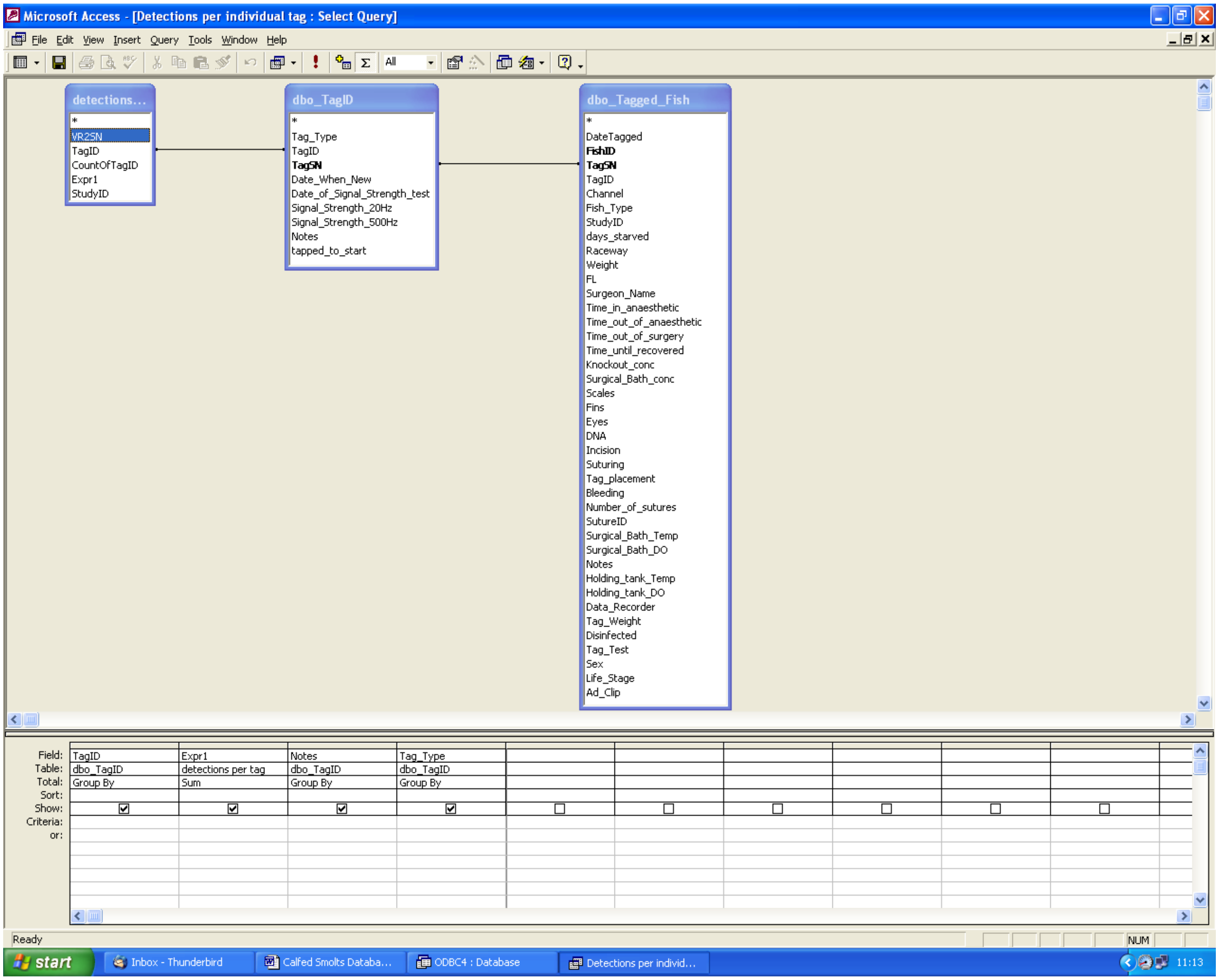
TagID	Channel
356	D
357	D
358	D
359	D
360	D
361	D
362	D
363	D
364	D
365	D
366	D
367	D
368	D
370	D
*	

	Date_Released	start_transport	Total_Tagged_	Released_by	Transport_met	Flow_conditio	Weather	Transport_time	Transport_tank	Transport_tank	release_site_t	release_site_D
+	1/19/2007 13:45:00	13:07:00	14	CJM AJ	cooler+ bubbler	low	hazy, warm	0:28				
+	1/22/2007 18:25:00	17:56:00	13	CJM AA	cooler+ bubbler	low	sunny, warm	0:29				
*												

	Release_location	Lon	Lat	photo_filename	Description	Notes
+	SacJellyRamp	40.31771	-122.18921		30 meters down	
+	SacOldSac	38.581239	-121.507286		Old Sacramentc	Brandes Dec 20
+	SFMokeWalnutG	38.22415	-121.49165		SF Mokelumne	
+	SFMokeWimpy	38.22627	-121.49128		SF Mokelumne	
+	SpanglerMoke	38.22108	-121.04286			
+	VandalMoke	38.2082	-121.08275		Vandal Island	
*						

Picture 3: Using relationships, one can view many different sets of data in one table

Another helpful feature that relationships enable pertains to query making. When you add tables to the query wizard, it will automatically link the tables according to how they are set up in your relationships (Picture 4). These relationships are essential for your query writing as they allow you to compare parameters from different tables.


Microsoft Access - [Detections per individual tag : Select Query]
File Edit View Insert Query Tools Window Help
All
Ready
NUM
start
Inbox - Thunderbird
Calfed Smolts Databa...
ODBC4 : Database
Detections per individ...
11:13

detections...
*
VR2SN
TagID
CountOfTagID
Expr1
StudyID

dbo_TagID
*
Tag_Type
TagID
TagSN
Date_When_New
Date_of_Signal_Strength_test
Signal_Strength_20Hz
Signal_Strength_500Hz
Notes
tapped_to_start

dbo_Tagged_Fish
*
DateTagged
FishID
TagSN
TagID
Channel
Fish_Type
StudyID
days_starved
Raceway
Weight
FL
Surgeon_Name
Time_in_anaesthetic
Time_out_of_anaesthetic
Time_out_of_surgery
Time_until_recovered
Knockout_conc
Surgical_Bath_conc
Scales
Fins
Eyes
DNA
Incision
Suturing
Tag_placement
Bleeding
Number_of_sutures
SutureID
Surgical_Bath_Temp
Surgical_Bath_DO
Notes
Holding_tank_Temp
Holding_tank_DO
Data_Recorder
Tag_Weight
Disinfected
Tag_Test
Sex
Life_Stage
Ad_Clip

Field:	TagID	Expr1	Notes	Tag_Type						
Table:	dbo_TagID	detections per tag	dbo_TagID	dbo_TagID						
Total:	Group By	Sum	Group By	Group By						
Sort:										
Show:	☑	☑	☑	☑	□	□	□	□	□	□
Criteria:										
or:										

Picture 4: When these three tables were added to the query design view, the relationships were automatically added.