1. Hardware

* Make it Dedicated. IQ and RDBMS’s, ETL tools, Cube Builders, etc. should not be on the same box if you can help it !
* Set aside 500mb min (more if you wish to keep many logs, ebfs, etc on disk) for IQ software
* Staging area - if loading from flat files (which is the fastest way to load) make sure they are not on the same disk/s as the DB files.
* Swap – twice physical memory
* Memory – Recommend 1gb for anything more than a development system (even for a proof of concept). Rough rule of thumb is 2 GB per CPU - Get whatever you can afford, and then more!
* CPU – Fast CPU’s make a big difference – Higher clock speeds are important (P4 3ghz+, Itanium) and lots of them ! IQ is more CPU than I/O intensive even for loading ! Ideally 1 CPU per active query.
* OS – If you aren’t committed to one, consider Linux (gives you S/W benefits of Unix, with Hardware benefits of Intel CPU’s).
* 64bit OS & Processor is preferred.
* Multiple controllers and disks recommended. Minimum 3 disks should be used for IQ (1 for OS, SW & logs, 1 for Main store & 1 for temp store).

1. Software

* Install latest patches for OS
* Install ASIQ
* Install latest EBF (base releases of IQ can be a little buggy)

1. Databases

* Sizing – As a start, assume 70% Raw data size for IQ Main Store & 30% for IQ Temp Store (this is a very rough figure based on 75% Gut feel and 25% experience with small to medium DB’s – 10-12-gb, but it’s a start)
* Raw devices are a bit faster (but are a little harder to manage) – Needed for Multiplex, so if you don’t use them you may have to rebuild you DB from scratch down the line (and rebuilding a multi-terabyte database is not trivial).
* IQ Page Size – Min 128k on 64bit platforms (which you should be using if you can !). Probably the best in most cases unless you have tonnes of memory (as the larger the page size the fewer pages fit in memory).
* If possible, create the database as CASE RESPECT. It is more efficient.

1. Configuration

* If it can be set in a config file, it should be set in a config file !
* What to set:

*-n ServerName*

*-iqmc main cache size (start with 40% of whats available for IQ)*

*-iqtc temp cache size (start with 60% of whats available for IQ)*

NB: Above figures assume an even mix of queries. For environments where lots of sort-merge joins or large number of HG index inserts are being done, then a 30/70 split May be more appropriate. Where there is an **extremely** low number of these sorts of operations a 50/50 split may be more appropriate.

*-cl Minimum cache size for catalog store (32mb recommended)*

*-gp catalog store page size (should be same as when db create, usually 4096)*

*-gm number of concurrent users (default 10)*

*-gc checkpoint interval (default 60, recommended 6000)*

*-gr max recovery time (default 2, recommended 6000)*

*-ti default client timeout (default 240, recommended 4440 which is ~72 hours)*

*-tl default network timeout (default 120, recommended 300+)*

-x tcpip{port=xxxx} (if you don’t do this, other protocols are started such as ipx which you don’t need)

* Some DB options you may want to set:

*Set option Force\_No\_Scroll\_Cursors='On';*

*Set option.Query\_Temp\_Space\_Limit = 0;*

*set option public.Minimize\_storage='On'; (see indexing section)*

*sp\_iqcheckoptions – shows values changed from the default.*

*SET OPTION PUBLIC.LOAD\_MEMORY\_MB = x; (where x is the max memory to use for loading. Default is 0, or unlimited.) NB: Remember this memory is separate to what is specified in iqmc and iqtc.*

1. Indexing

* Always create indexes before loading data – it is much quicker than creating the indexes after loading.
* Fast projection (FP) indexes are created on all columns. You can use IQ Unique to specify the cardinalty (lots of work) or… Turnng the option Minimize\_storage on will give you maximum compression on disk and best query performance. Warning – This will slow down the loading of very wide tables (1000+ columns) so turn if off to load these.
* Low fast (LF) indexes should be used where cardinality is <1500. Its used by Sarg searches, Min, Max & Group by operations.
* High Group (HG) indexes should be used where cardinality is 1500+ and on all key/join columns. Can be multi column index
* High Non Group (HNG) indexes are used for range and aggregation queries (except on dates).
* Compare (CMP) indexes are used for comparisions between two columns in the same table (columns must be same datatype inc. precision & scale).
* Word (WD) indexes index textual data separated by spaces, punctuation, etc. very useful for like and contains queries. NB: Restricted to indexing ‘words’ up to 3 bytes.
* Date, Time and DataTime (DTTM) indexes are used for range and datepart queries on date and time data
* Join indexes can be created where two tables are frequently joined together (only create them if you need them, IQ is still v. fast without them). They need a significant amount of disk space and need refreshing after data loads, so may not be worth the trouble.
* Be careful about the use of Unique, primary key, etc. keywords when creating tables. They will create indexes, which you may or may not want. Having said that, PRIMARY KEY indexes do the same as Unique HG, but are more useful to the optimizer and FOREIGN KEY indexes to the same as non-unique HG but are also more useful to the optimizer.
* Try to create indexes so that inserts will be placed at the end of the index.
* Indexes can be create in parallel ie:

Begin Parallel IQ

Create LF index …;

Create HG index …;

...

End Parallel IQ;

It is recommended to create 1 index for each CPU.

* As you may have seen on the training course we can follow the rules show in the indexing chart (see next page).

1. Unions & Views

* Views on physically partitioned tables (ie union views) can perform significantly better than on a single large base table. Loading can also be quicker as you can load into a new table and add it to the view.

1. Loading Data

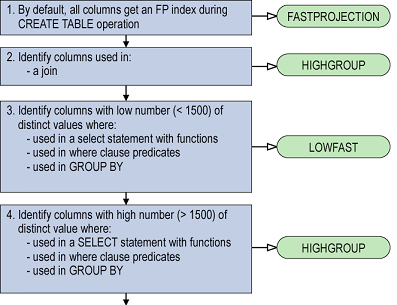
* Ways to load data (From fastest to slowest):
  + Load table command (parallel)
  + Load table command (sequential)
  + Insert from Location
  + Insert Values
* Where using load table specify multiple files with one load table command rather than multiple load table commands for each file.
* To run load table in parallel, you need to specify the ROW DELIMITED BY and DELIMITED BY options.
* If possible use binary mode of load table (can be 3-10 times faster).
* Some settings that may improve load performance:  
  - DISABLE\_RI\_CHECK (RI checks can cause up 5% of load time)  
  - MAX\_IQ\_THREADS\_PER\_CONNECTION  
  - SORT\_PHASE1\_HELPERS  
  - WASH\_AREA\_BUFFERS\_PERCENT  
  - SWEEPER\_THREADS\_PERCENT  
  These settings will require testing to find the right combination, if in doubt, stick to the defaults.

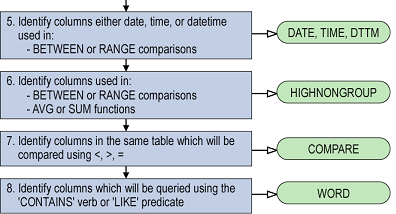
1. Extracting Data

* Ways to extract data (from fastest to slowest):
  + Using Temporary options
  + Proxy tables (pointing to ASIQ from a remote server – Make sure you tune the packet size)
  + Redirection (ie: select \* from employee   
    ># empfile.txt)

1. General

* Avoid using the default port (2638) as this will be used by all ASA based applications (ASA, ASIQ, RS12.6, IO4.0, etc) as a broadcast listener.
* If returning large amounts of data to the client (as is often the case with BI tools), set a larger network packet size using –p in the config file.
* Char is more efficient than varchar – avoid varchar if not needed.
* Avoid storing what you don’t need ie: don’t use datetime if you only need date or integer if you only need smallint.
* Always specify either NULL or NOT NULL. It helps the optimizer make better decisions.
* Make sure columns to be joined use exactly the same datatype to avoid implicit conversion.
* Avoid Cursors - IQ is not designed for Single row processing.
* Set option Public.OS\_File\_Cache\_Buffering = ‘Off’; Stops double buffering of IQ pages.
* Do not do single row inserts if possible - Expect no more than 5,000 to 20,000 operations per hour.
* Pipes can be faster than files – Load table can read from a pipe, so if you can avoid writing to disk this way it can be significantly quicker.
* ANSI joins are faster (and less ambiguous) than TSQL joins.
* Use multiplex – It is v. easy to set up and manage and will allow you to scale to thousands of users.
* What sort of network are your servers running on? Install Gigabit Ethernet for all server to server communication (especially important for insert from location). Consider multiple interfaces, one for clients and one for servers





3. Getting information using stored procedures

Sybase IQ offers several stored procedures that display information about your

database:

? sp\_iqconnection displays statistics about user connections and versions

? sp\_iqcontext displays information about what statements are executing

? sp\_iqcheckdb checks the validity of your current database

? sp\_iqdbstatistics reports results of the most recent sp\_iqcheckdb

? sp\_iqdbsize gives the size of the current database

? sp\_iqspaceinfo displays space usage by each object in the database

? sp\_iqstatus displays miscellaneous status information about the database.

? sp\_iqtablesize gives the size of the table you specify.

? sp\_iqgroupsize lists the members of the specified group.