

DESIGN AND IMPLEMENTATION OF AN ANESTHESIA DATA WAREHOUSE USING OPEN SOURCE TOOLS AND SOFTWARE

Matthew A. Levin, MD; Patrick J. McCormick, MD; Marina Krol, PhD

Department of Anesthesiology, Mount Sinai School of Medicine, New York, New York

Introduction: Large scale data mining and analysis is becoming critical to achieving the next level of safety and quality in anesthesia. For the past four years we have been developing our own single-institution data warehouse, with an emphasis on the use of open source tools and software.

Methods: The current platform is a 3.8Ghz dual processor machine with 6 GB RAM and 1.5 TB disk space, running Ubuntu 10.04 (Linux) and MySQL 5.1. The core of the schema is 18 tables containing case data extracted from our AIMS, normalized loosely by data type. Surrounding this are several dozen lookup/reference tables. Extensive use of views provides an abstraction layer should the structure of the base tables change.

On top of this we have built a suite of tools in Perl, using a variant of the “extract transform load” data warehousing paradigm that we call “load filter transform”. There is one program that is responsible for loading data, and another which allows users to query the warehouse. Object oriented design and modularity provides flexibility and extensibility (Table 1).

Results: The warehouse currently contains more than 150GB of data from over 300,000 anesthetics, and growing. The largest table, containing physiologic data, has over 3 billion rows. Intelligent use of partitioning on this table allows for fast inserts and queries. Typical processing time for a complex research question involving calculations on physiologic data is 2.5 seconds per case. Next steps include performance optimization and migration to a larger server supported by hospital IT.

Table 1

Program	Purpose
Case.pm	object that represents a case
crload.pl, CRParser.pm	extract data from AIMS and load into warehouse
extract-physio.pl	main program for querying the warehouse
Filter.pm	logic to filter and clean cases prior to processing
IncludeExclude.pm	transforms inclusion/exclusion criteria into SQL
PhysioCalc.pm	routines to transform physiologic data
Math.pm, Util.pm, Config.pm	common, reusable code