# Data Warehouse Methodologies

## About the Transactional Database

The transactional database models a veterinary’s office. It includes tables for collecting data on the types and occurrences of procedures that a vet or vet’s office staff may perform on pets and animals. Such information includes the types of procedures, the number of times the procedures occurred, how often a patient may have been seen by a vet, and the products and additional services sold to owners of the pets. The database is in 3rd Normal Form, and is designed to be efficient for insertion of data.

In this application we have worked to build a Java-based GUI application to take the transactional data and copy such data into a data warehouse. Three different data marts have been designed using the “Star Schema” methodology. Each schema exists independent of

## Simple Methodology

The simple methodology is designed around describing a single fact. The methodology includes a simple star schema, with a single FACT table. The fact table in this schema looks at the overall earnings of the vet’s office. This single fact table provides a look at how much a particular vet’s branch or office has earned over a period of time.

The simple methodology includes dimensional tables that are complete copies of the necessary transactional tables. In the process of creating these tables, the application simply examines the existing transactional table schemas and then replicates the schemas in SQL DDL/DML statements, and then executes those statements against the same database where the transactional tables exist. The dimensional tables are prefixed with “DIM\_”, such that a transactional table named “BRANCH” will now have a corresponding dimensional table named “DIM\_BRANCH.”

A single fact table is then constructed from the primary keys from the selected dimensional tables and then non-key attributes that represent the measures are established in the GUI. The name of the fact table is automatically prefixed with the characters “FACT\_” such that if the desired table name is “EARNING” then the full name of the FACT table would be “FACT\_EARNING.”

As part of the semi-automated nature of this application, data from the transactional tables are copied into the corresponding dimensional tables automatically, but a more complex SQL insert statement must populate the FACT table. This more complex insert statement must be written and executed manually.

## Average

The average methodology is similar to the simple methodology in that the names of the generated dimensional tables are prefixed with “DIM\_” and the resulting fact table is prefixed with the letters “FACT\_”. Beyond that, the methodology is more complex in that it illustrates how the user of the application can generate custom dimensional tables using a powerful table builder screen in the application (Screenshot #2). Similar to the simple methodology, a single fact table is generated from the custom dimensional tables.

Population of the dimensional tables is automatic, but just as in the simple methodology, population of the fact table requires manual script writing and execution.

## Complex

The complex methodology illustrates how to build a