Background

- Normalisation
  - Data to many tables
  - Less entry
  - More complex data retrieval
  - Referential integrity

In practice few applications have data fully normalised data behind them.

Terminology f4b/MultiStat/MultiStock

- A User View
  - an SQL Statement that returns data that looks like a single table.
  - We term the returned data a Recordset

- Activities/Data Sets/Workflows/Workbooks
  - We term a set of User Views an Activity

SQL

- Originally just Data Manipulation
  - SELECT
  - INSERT
  - UPDATE
  - DELETE
- Later Data Definition
  - CREATE TABLE, INDEX, VIEW…
  - ALTER TABLE
- Problem was although SQL is an ANSII standard all the implementations are different (mySQL,SEQUEL,ORACLE, PROGRESS)

f4b SQL

- Using ODBC f4b deals with the different dialects of SQL.
- In general f4b native SQL is SEQUEL

Query Design Tools and f4b

- We have used the query designer of Microsoft Access for years.
- Other Microsoft products (SEQUEL, MSQUERY…) presented similar designers.
- We found none that we liked to use against a PROGRESS database.
- So we built one into f4b, but if you like you can use an external designer and paste the SQL into f4b.
- The aim of f4b is to make accessing and using all that SQL consistent and simple

Creating an Activity from a Rollup View

1) Open existing tables and User Views of the data you require and save them as a new Activity.
2) Filter and sort the fields to show exactly the data required (Equals a Specification)
3) Create ‘SYSTEM’ views from your User Views.
4) Join your system views to create a new User View (Rollup)
5) Create a ‘SYSTEM’ view of your Rollup.
6) Cache the resultant recordset from Rollup.
7) Link the tabs to provide drill down capabilities.
8) Analyse your Rollup View to provide aggregate analysis.
9) Publish your Views and Activities.