FileMaker Architecture: Transactions, say *hello* to Tiers

Chris Irvine
Senior Technology Consultant
Threeprong.com LLC

About Me

- Oregon Native
- Many years as an IT Manager, Platforms, Languages
- Threeprong.com

Architecture Primmer

- Separation of Concerns (SoC)
- Tightly Coupled vs Loosely Coupled System
- Defensive Programming
- Capitalize on Strengths
- Separation Model ≠ Multitier Architecture

Single File Solution

FileMaker
In Practice

.fmp12

User Interface
Reports
Automation
Persistent Database
Account Management
API Integration
Event Triggers

Separation Model

FileMaker
In Practice

Solution File

.fmp12

User Interface
Reports
Automation
API Integration
Event Triggers

Data File

.fmp12

Persistent Database
Account Management

Separation Model

FileMaker
Practice

	Development	Production
Solution File	.fmp12 User Interface Reports Automation API Integration Event Triggers	.fmp12 User Interface Reports Automation API Integration Event Triggers
Data File	.fmp12 Persistent Database Account Management	.fmp12 Persistent Database Account Management

Three-tier Architecture

FileMaker
In Practice

Presentation Tier

.fmp12

User Interface
Presented Reports
Interface Automation
Event Triggers

Logic Tier

.fmp12

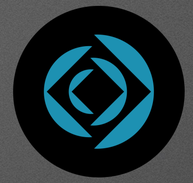
Business Automation
Delivered Reports
API Integration

Data Tier

.fmp12

Persistent Database
Account Management

Three-tier Architecture



FileMaker
In Practice

Presentation Tier	User Interface Presented Reports Interface Automation Event Triggers .fmp12	Donner.fmp12
Logic Tier	Business Automation Delivered Reports API Integration .fmp12	Donner Wingman.fmp12 (hidden)
Data Tier	Persistent Database Account Management .fmp12	Donner Table Storage.fmp12 (hidden)

Three-tier Architecture

FileMaker
In Practice

Presentation Tier

.fmp12

User Interface
Presented Reports
Interface Automation
Event Triggers

Logic Tier

.fmp12

Business Automation
API Integration
Delivered Reports

Data Tier

.fmp12

Persistent Database
Account Management

.fmp12

Three-tier Architecture

FileMaker
In Practice

Presentation Tier

.fmp12
User Interface
Presented Reports
Interface Automation
Event Triggers

.fmp12
Mobile or Web UI

Logic Tier

.fmp12
Business Automation
API Integration
Delivered Reports

Data Tier

.fmp12
Persistent Database
Account Management

.fmp12

Three-tier Architecture

FileMaker
In Practice

Presentation Tier

.fmp12
User Interface
Presented Reports
Interface Automation
Event Triggers

.fmp12
Mobile or Web UI

Logic Tier

.fmp12
Business Automation
API Integration
Delivered Reports

Data Tier

.fmp12
Partitioned Database
~~Account Management~~

.fmp12
Partitioned Database
~~Account Management~~

Three-tier Architecture

FileMaker
In Practice

Presentation Tier

.fmp12
User Interface
Presented Reports
Interface Automation
Event Triggers

.fmp12
Mobile or Web UI

Logic Tier

.fmp12
Business Automation
API Integration
Delivered Reports

Exposed API

Data Tier

.fmp12
Partitioned Database
~~Account Management~~

.fmp12
Partitioned Database
~~Account Management~~

Three-tier Architecture

FileMaker
In Practice

Presentation Tier

.fmp12
User Interface
Presented Reports
Interface Automation
Event Triggers

.fmp12
Mobile or Web UI

Logic Tier

.fmp12
Business Automation
API Integration
Delivered Reports

Server
Schedule

Data Tier

.fmp12
Partitioned Database
~~Account Management~~

.fmp12
Partitioned Database
~~Account Management~~

Three-tier Architecture

FileMaker
In Practice

Presentation Tier

.fmp12
User Interface
Presented Reports
Interface Automation
Event Triggers

.fmp12
Mobile or Web UI

Logic Tier

.fmp12
Business Automation
API Integration
Delivered Reports

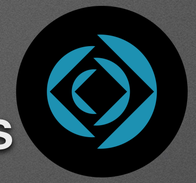
Data Tier

.fmp12
Partitioned Database
~~Account Management~~

.fmp12
Partitioned Database
~~Account Management~~

Self-contained Module

Three-tier Architecture



FileMaker
In Practice

Presentation Tier

.fmp12

User Interface
Presented Reports
Interface Automation
Event Triggers

Logic Tier

.fmp12

Business Automation
Delivered Reports
API Integration

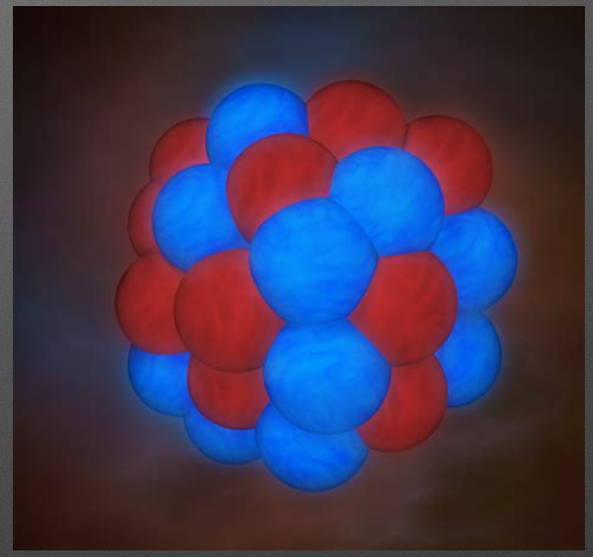
Data Tier

.fmp12

Persistent Database
Account Management

- Separation of Concerns
  - Simplified Relationship Graphs
- Testing & Debugging
- PSoS and Server Schedules
  - No Trigger Complications
  - Better Performance
- Data Durability & Recovery

Atomic Transaction



- Smallest possible part
  - *“indivisible & irreducible”*
- Go/No-Go; All or Nothing
- In progress transactions can not be observed by another client

Real World Demos

- Sample File
  - Direct User Transaction Behavior
  - User w/Script Managed Transaction
  - Decoupled Transaction Script (logic layer, no UI)

- Demo

Some Technical Tidbits

- FileMaker's internal Record ID
  - Allocated on record creation, not recovered on revert
- "Magic Key" behavior
  - When parent record has blank foreign key, child creation backfills key
- Primary Keys can be Numeric Serials, UUIDs, or Numeric UUIDs
  - UUIDs are less entangled with table definition, flexible assignment

Some Technical Tidbits

- Practical Limits
  - 20-50 open records on client seem fine; it depends
  - hundreds or thousands of open records if using PSoS
- Preflight > “Critical Section” > Postflight; Minimize Lock Duration

Some Technical Tidbits

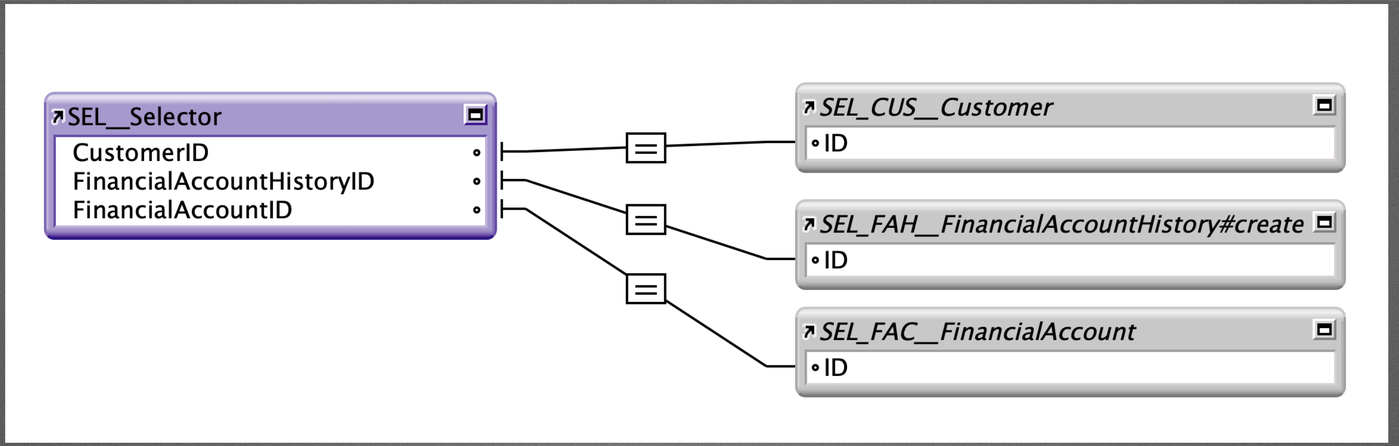
- Selector Table - one-record table of temporary fields for accessing various records via relationship graph
  - Table is an implementation trick & does not belong in the data layer (virtual list similar)
  - Existing Records: Select, edit, orphan, re-select, edit, commit
  - Record Creation: Create, populate, orphan, commit (re-select doesn’t work)

Some Technical Tidbits

- Selector Table; one-record table of temporary fields for accessing various records via relationship graph
  - Don’t use selector technique to manage record deletion. May need a temporary record or a portal element.
  - Missing record in selector table will block record creation.
  - Watch out for uses that lock your selector record.

Some Technical Tidbits

- Watch out for uses that lock your selector record.



- Watch out for uses that lock your selector record.

SELECTOR LOCKING BEHAVIOR	ONLY GLOBALS IN TABLE	GLOBALS AND STORED FIELDS (UNUSED)	GLOBALS AND STORED FIELDS USED AS FOREIGN KEY
USER INTERACTIVE	ANCHOR & BUOY	ANCHOR & BUOY	ANCHOR & BUOY
SCRIPT STEP: SET FIELD	BUOY	BUOY	ANCHOR & BUOY

“Blah blah blah… I’ve been doing this 20 years and I have never needed this kind of thing in real customer systems.”

-Anonymous Complainer

- Not Hard
- Sanitary
- Error Trapping Needed Regardless
- Future Proof