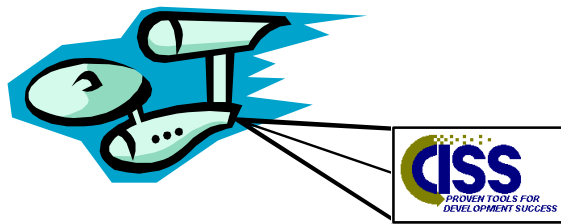


Track 7:Dynamic Searcher

Session 730

Taking your IEF application where
no IEF application has gone before.



Prepared By: Michael Yaffe

Design Objectives

- Platform Independance
- Support a limitless combination of queries
- Provide sub second response time
- Provide online, print or file output
- Reusable components
- Statically linked and bound
- In real-time mode provide the ability to link to the source object(double click event to link to the thing that was found)

Prepared By: Michael Yaffe

Roadblocks to Dynamic Searching



- Multiple physical platforms
- Multiple database environments
- Both Online and Batch Applications
- Bias against complex queries in production
- Shortage of skilled resources
- Absence of a good technical solution

Prepared By: Michael Yaffe

Technical Solution

- Create a Table whose sole purpose is to find things(including support for between and like)
- Ensure that the column layouts are generic
- Ensure that all access is through indexes
- Ensure that the data can easily be re-loaded to maintain currency
- Join the table to itself, but no other????



Prepared By: Michael Yaffe

Join to Itself ????

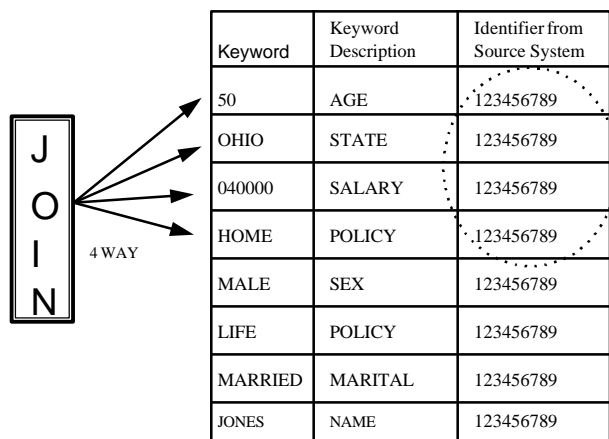
I need to know all people who are over 50 ,earn more than \$40,000, live in Ohio and have a Home Insurance policy.

- In a normal relational system this would require the joining of three or four tables assuming customer demographics and insurance policies are in the same DBMS.
- Adding other criteria would expotentially increase the number of SQL statements required.
- When you join a table to itself you find all keywords that match the above criteria /Age = 50 / State = Ohio / Policy = Home / Salary = \$40,000 /



Prepared By: Michael Yaffe

Join to Itself ????



Prepared By: Michael Yaffe

Table Design

- **Keyword Type** (Identifies the source of the data)
- **Keyword** (text your searching on (50, \$40,000, ohio, Home))
- **What the keyword represents** (Age, Salary, State, Policy type)
- **Feeder system identifier** (Primary key on the feeder system of what's been indexed "Customer #")
- **Description** (Textual description of the indexed entity "Customer Name")
- **Number of SQL statements equals the maximum number of concurrent keywords you need to search on.**
Experience suggests that it is difficult to find anything that will meet eight criteria .

Prepared By: Michael Yaffe

Database Layout

Record	KEYWORD_SEARCH		103			
Field	KEYWORD	Text	20	Not Null		
Field	KEYWORD_TYPE	Text	1	Not Null		
Field	ORIGINATOR_ID	Integer	4	Not Null		
Field	KEYWORD_DESCRIPTOR	Text	8	Not Null		
Field	ORIGINATOR_DESCRIPTOR	Varchar	62	Not Null		
EntryPoint (U) SND 24						
Field	KEYWORD	Text	20	Not Null	Ascend	
Field	ORIGINATOR_ID	Integer	4	Not Null	Ascend	
EntryPoint (U) I1 (Primary) 25						
Field	KEYWORD_TYPE	Text	1	Not Null	Ascend	
Field	ORIGINATOR_ID	Integer	4	Not Null	Ascend	
Field	KEYWORD	Text	20	Not Null	Ascend	

Bottom Line

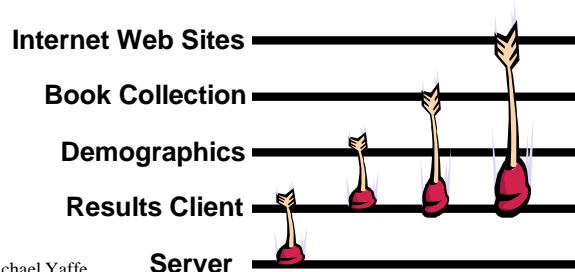
Not all data fits nicely into a normalized relational structure

Try and design a database that describes the products in a L.L. Bean Catalogue. There are too many products that fall into too many categories. eg. Shirt, Short Sleeve, Cotton, Button Down, Striped, \$35, Mens

Prepared By: Michael Yaffe

Program Design

- Based on two re-usable components and as many custom clients as required.
- Server - Contains SQL, formats response to either client window, print or file. Has no knowledge of the Client.
- Results Client - displays results of query
- Query Client - Collects search criteria from the user



Prepared By: Michael Yaffe

Sample Server SQL

```
35 CASE 2
40 READ EACH first keyword_search_facility
40 second keyword_search_facility
40 TARGETING export_list FROM THE BEGINNING UNTIL FULL
40 WHERE DESIRED first keyword_search_facility keyword IS GREATER OR EQUAL TO
40 local_1st_from keyword_search_facility keyword
40 AND DESIRED first keyword_search_facility keyword IS LESS OR EQUAL TO
40 local_1st_plus_zzzz keyword_search_facility keyword
40 AND DESIRED first keyword_search_facility keyword_type IS EQUAL TO
40 local_1st_from keyword_search_facility keyword_type
40 AND DESIRED first keyword_search_facility keyword_descriptor IS LIKE
40 local_1st_from keyword_search_facility keyword_descriptor
40 AND DESIRED second keyword_search_facility keyword IS GREATER OR EQUAL TO
40 local_2nd_from keyword_search_facility keyword
40 AND DESIRED second keyword_search_facility keyword IS LESS OR EQUAL TO
40 local_2nd_plus_zzzz keyword_search_facility keyword
40 AND DESIRED second keyword_search_facility keyword_descriptor IS LIKE
40 local_2nd_from keyword_search_facility keyword_descriptor
40 AND DESIRED second keyword_search_facility keyword_type IS EQUAL TO
40 local_1st_from keyword_search_facility keyword_type
40 AND DESIRED first keyword_search_facility originator_id IS EQUAL TO DESIRED
40 second keyword_search_facility originator_id
41 MOVE first keyword_search_facility TO export keyword_search_facility
42 SET export_records ief_supplied count TO 1 + export_records ief_supplied count
43 SET export ief_supplied select_char TO SPACES
```

Prepared By: Michael Yaffe

Sample Client Code

```
12 EVENT ACTION search
13 SET SUBSCRIPT OF export_search_group TO 0
14 IF import_1st_from keyword_search_facility keyword IS NOT EQUAL TO SPACES
15 SET SUBSCRIPT OF export_search_group TO SUBSCRIPT OF export_search_group + 1
16 MOVE import_1st_from keyword_search_facility TO
16 export_1st_from keyword_search_facility
17 SET export_fromgrp_item keyword_search_facility keyword TO
17 import_1st_from keyword_search_facility keyword
18 SET export_fromgrp_item keyword_search_facility keyword_descriptor TO "%%%%%%%%%"
19 SET export_fromgrp_item keyword_search_facility keyword_type TO "I"
20 SET export_to_grp_item keyword_search_facility keyword TO
20 import_1st_from keyword_search_facility keyword
14
21 IF import_2nd_from keyword_search_facility keyword IS NOT EQUAL TO SPACES
22 SET SUBSCRIPT OF export_search_group TO SUBSCRIPT OF export_search_group + 1
23 MOVE import_2nd_from keyword_search_facility TO
23 export_2nd_from keyword_search_facility
24 SET export_fromgrp_item keyword_search_facility keyword_descriptor TO "%%%%%%%%%"
25 SET export_fromgrp_item keyword_search_facility keyword TO
25 import_2nd_from keyword_search_facility keyword
26 SET export_to_grp_item keyword_search_facility keyword TO
26 import_2nd_from keyword_search_facility keyword
21
```

Prepared By: Michael Yaffe

Pros and Cons

- Cheap to implement
- Flexibility limited to your ability to dream up new uses
- Re-usable components
- Platform independent
- Maximizes IEF's ability to link a variety of clients to a server
- Hours to develop a new client load the database and be up and running



- Duplication of data
- Data may become stale
- Spoil your users



Prepared By: Michael Yaffe

Demo

- *Internet Client*
- *Book Collector Client*
- *Demographic Client*
- *Display Results Client*
- *Search Server*
- 100,000+ rows in Search Table
- Toshiba T4900/P75/810 Hard Drive/40 Meg of memory
- All programs, tables and indices on one drive - a worst case scenario
- IEF 5.3.1, OS2 Warp, DB2/2

Prepared By: Michael Yaffe