

# Taxis

## Taxis

- Overview
- Application Design
- Creating queries
- Importing data
- Other Topics

## Overview

- Combines traditional relational operations with text queries
- Has locking for concurrency
- No transactions
- Performance oriented

## Application Design

- Identify the critical parts of the application
  - Search performance
  - Update rate
  - Simple code
  - Data maintenance
- Design for those first
- Build from there

## Designing the Database

- Database and Application
- Performance for websites
- Choosing tables and disk layouts
- Efficiency and reducing redundant work
- Writing the Vortex

## Creating Queries

- Query, index, table layout
- Search strategies
- Ordering clauses
- Joins and subqueries
- Choosing indices to create

## Search Strategies

- Keep searches as simple as possible
- Put most restrictive clauses first
- LIKE clauses typically first
- Want to avoid
  - looking at lots of records
  - reordering results
  - functions in WHERE clause

## SQL File Functions

- fromfile
- fromfiletext
- totext
- toind

## SQL String Functions

- abstract
- text2mm
- keywords
- length
- lower
- upper
- initcap
- sandr

## SQL Math Functions

- acos
- asin
- atan
- atan2
- ceil
- cos
- cosh
- exp
- fabs
- floor
- fmod
- log
- log10
- pow
- sin
- sinh
- sqrt
- tan
- tanh

## SQL Date Functions

- dayofmonth
- dayofweek
- dayofyear
- dayname
- monthname
- year
- quarter
- month
- week
- hour
- minute
- second

## SQL Bit Functions

- bitand
- bitor
- bitxor
- bitnot
- bitsize
- bitcount
- bitmin
- bitmax
- bitlist
- bitshiftleft
- bitshiftright
- bitrotateleft
- bitrotateright
- bitset
- bitclear
- bitisset

## SQL IP functions

- inetabbrev
- inetcannon
- inetnetwork
- inethost
- inetbroadcast
- inetnetmask
- inetnetmasklen
- inetcontains
- inetclass
- inet2int
- int2inet

## SQL Geo Functions

- latlon2geocode
- geocode2lat
- geocode2lon
- distgeocode
- distlatlon
- dms2dec
- dec2dms (e.g. 35.25 → 351500 [35°15'])

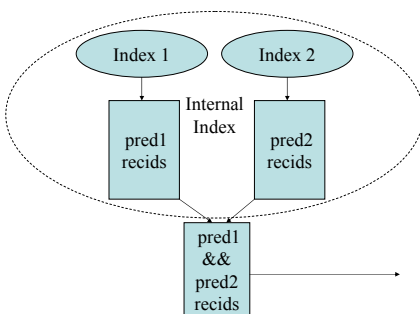
## Other SQL Functions

- exec
- mminfo
- convert
- seq
- random

## mminfo

- mminfo(query, data, nhits, 0, msgs)
- msgs
  - 1 includes 300 messages
  - 2 disables text
  - 4 counts hits, prints message
  - 8 counts hits, just prints count

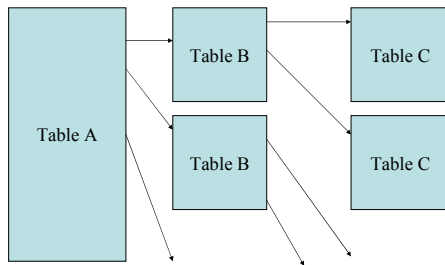
## Query evaluation



## Ordering Clauses

- Most restrictive first
- btreshold
- maxlinearrows
- keep compound index clauses together  
...AND (A = \$qA AND B = \$qB) AND ...

## Joins (FROM A,B,C)



## Joins

- Nested select
  - First unindexed, then most indexed and so on
- Taxis optimizes table order
  - knowledge of data and query
- May be able to do better manually
  - knowledge of data and query

## Subqueries

- Evaluated early if not correlated
- If correlated evaluates for each record
- Some subqueries are similar to joins
- Can also emulate subqueries with Vortex

## Creating Indices

- Goal is to improve performance
  - Taxis chooses index to use automatically
  - Scores if multiple potential indices
- Too many indices is not good
  - compound index
  - virtual fields

## Regular Index (Btree)

- Used for =, >, MATCHES etc.
- Rapid access to records that match a single or range of values
- Can include multiple fields
- UNIQUE flag

## Metamorph Queries

- Set Logic
  - each term is a set of equivalent patterns
  - presence or absence defines a hit
  - intersects @0
  - delimiters are also sets

## Syntax

- White space delimited terms, quote phrases
- Leading character(s) define operation
  - @ - intersects
  - w/ W/ - delimiters
  - +, - - required, exclude
  - / # % - special pattern matchers
  - ~ thesaurus toggle

## Query Protection

- Enabled in Vortex by default
- Protect webserver from pathological queries
- <putmsg> to view messages
- default warn and ignore
- <apicp> to disable
- Post-processing, linear search

## Query processing options

- Suffix processing
  - suffix, minwordlen, defsuffrm
- Thesaurus
  - keepeqvs, eqprefix, ueqprefix
  - editable with backref
- Noise list

## Types of Text Query

- LIKEP
- LIKE
- LIKER
- LIKE3
- LIKEIN
- MATCHES

## LIKEP

- Most common, orders by rank. Most detailed ranking
- Rank factors
  - frequency in document and corpus
  - cluster proximity
  - result term ordering
  - proximity to start of document

## LIKE

- Other commonly used search
- Calculates a quick rank, but doesn't sort
- Record either matches or doesn't
- Does not require post-process if index can resolve query

## Other query types

- LIKER - first stage of LIKEP. Quick rank, no proximity information, longer the query the more useful it is
- LIKE3 - first stage of LIKE, index only. Never post-processes.
- LIKEIN - for profiling
- MATCHES - tradition SQL LIKE

## Text Index Types

- METAMORPH INVERTED INDEX
  - full inversion, most useful, highest overhead
- METAMORPH INDEX
  - lower resolution and overhead
  - can resolve LIKE for simple keyword searches
- METAMORPH COUNTER
  - as METAMORPH, but extra info for LIKEIN

## Index Issues

- Index expressions. Default `[alnum]{2,99}`
- `addexp`, `delexp`
- foreign characters, punctuation, special characters (hyphenphrase)
- Noise list
  - CD “Who are you?” by “The Who”
  - States, IN, ME and OR.

## Indexing

- Regular Index
- Metamorph Index
- Goal is to improve performance
- Don't create too many indexes

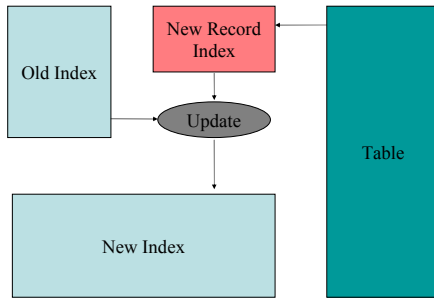
## Metamorph Index

- Three types, regular, inverted and counter
- Virtual fields
- Compound indexes

## Metamorph Index Update

- Single record update is expensive
- Many record update about the same
- Index updates deferred to be batched
- manual vs. automatic updates
- Real-time searches maintained

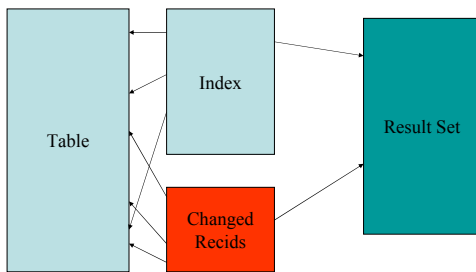
## Metamorph Index Update



## Manual vs. Automatic

- Manual updates good for batches or precise control.
- Bad if forgotten
- Automatic updates good for continuous streams
- Bad if they occur in the middle of a batch

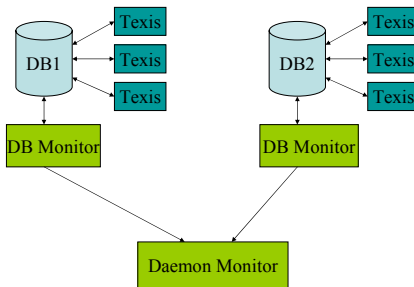
## Real-time searches



## Typical Installation

- Morph3 directory
- Setuid programs to Taxis user
  - most programs symlinked to monitor
- vhttpd needs root
- taxis for scripts, tsql ad-hoc SQL
- timport, creatdb, ltest, chkind

## Typical Processes



## Taxis Parameters

- Search and Optimization
- Metamorph
- Rank Knobs
- Other ranking properties
- Indexing
- Locking
- Miscellaneous

## Search and Optimization

- defaultlike
- matchmode
- predopttype
- ignorecase
- btrecachesize
- btreethreshold
- maxlinearrows
- ramrows
- ramlimit
- bubble
- optimize
- nooptimize
- phrasewordproc {mono,last}
- ignorenewlist

## Metamorph

- minwordlen
- keepnoise
- suffixproc
- prefixproc
- rebuild
- defsuffrm
- hyphenphrase
- wordc
- langc
- inc\_{s|e}dexp
- {s|e}dexp
- req{s|e}delim
- olddelim
- eqprefix
- ueqprefix
- useequiv
- withinmode
- wildoneword
- wildsingle

## Rank Knobs

- likeproximity
- likepleadbias
- likeporder
- likepdofreq
- likeptblfreq

## Other ranking properties

- likeprows
- likeptime
- likepallmatch
- likepindexthresh
- likepobeyintersects

## Indexing

- Indexspace
- indexblock
- indexmem
- addexp
- delexp
- lstexp
- infthresh
- infpercent
- addindextmp
- delindextmp
- lstindextmp
- likerrows
- indexaccess
- indexchunk
- indexmeter

## Locking

- singleuser
- lockmode
- locksleppmethod
- locksleeptime
- fairlock
- lockverbose
- nolocking
- verifysingle

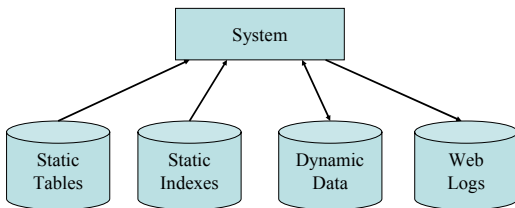
## Miscellaneous

- tablespace
- datefmt
- indirectcompat
- triggermode
- paramchk
- indirectspace
- locale
- optimize
- nooptimize
- message
- nomessage
- timezone

## Disk Layout

- Disk IO can be a big performance killer
- Tables and indices on separate disks
- Static and dynamic data on separate disks
- NO software RAID or NFS
- Use RAID 1, 1/0, potentially 5

## Disk Layout



## Ordering Data

- Most efficient read method is in disk order
- Taxis reads in table order by default
- Exceptions
  - ORDER BY, GROUP BY, DISTINCT
  - LIKEP
  - Single btree WHERE clause (bubble)

## RAM

- RAM can be a big performance improvement
- More of the index/tables you can keep cached in RAM the better

## Large number of records

- Keep queries simple, e.g. LIKEP or =
- Complex queries take more work, especially with a lot of hits
- Ordering should be either rank, or a simple compound index
- likepallmatch
- likeprows

## Many fields

- Use Virtual fields
  - either as part of the query or to restrict result set
  - normally should be first in WHERE
- Compound indexes
  - put LIKE first, remaining terms at end of query
- Ensure clause ordering

## Load sharing

- Distributing the load
  - different jobs on different machines
  - each optimized to that job
  - can handle updates
- Replicating database to spread load
  - same job on many machines
  - inherent backups
  - best for static data

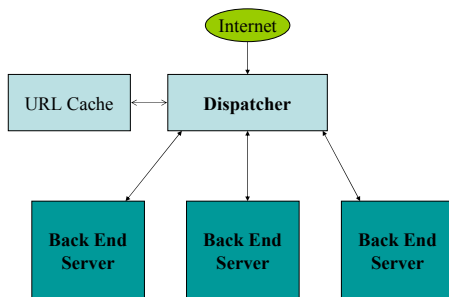
## Minimizing work done

- Preordering data
  - if there is a most common ordering for results
  - group commonly requested documents together
  - can save an order by in some cases
- Denormalizing
  - can save many additional record lookups
  - increases chance of data integrity problems

## Caching Results

- Cache on demand vs. pre-built pages
  - ease to build, disk space, frequency of access
- Size of cache
- Expiring the cache
  - how fast does the data change
  - how soon do you need the space
  - separate process

## Caching Proxy Setup



## Caching Proxy Code

```
<SCRIPT LANGUAGE=vortex>
<A NAME=main>
<$rqp=($SCRIPT_NAME+$PATH_INFO+ '?' +$QUERY_STRING)>
<SQL MAX=1 "select Html from cache where Path=$rqp">
  <send $Html>
  <exit>
</SQL>
<$machines=search1 search2 search3>
<randpick $machines>
<$hrq=( 'http://' + $ret+$rqp)>
<fetch $hrq><$Html=$ret>
<send $Html>
<SQL NOVARS "insert into cache (id, Html, Path)
  values(counter, $Html, $rqp)"></SQL>
</A>
</SCRIPT>
```

## Profiling

- Searching for queries
- Uses
  - Message routing
  - Notification
  - Ad rotation
  - News filtering

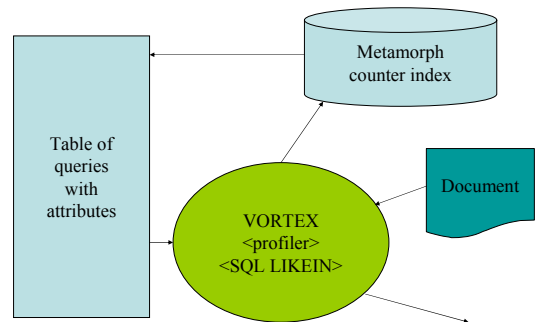
## Metamorph Counter Index

- Same as Metamorph index with additional information for LIKEIN
- Created and updated the same as other Metamorph indices
- Needs special index expressions  
set delexp=0;  
set addexp='-?\alpha{2,99}';

## Simple profiling demo

```
<profiler INIT queries prof query>
<while 1 eq 1>
  <getdocument>
  <$doc=$ret>
  <profiler GET $doc prof>
  <SQL ROW "SELECT id FROM queries
    WHERE query LIKEIN $ret">
    <processresult>
  </SQL>
</while>
```

## Profiling flow



## Timport

- Program vs. Function
- Common formats
  - csv
  - columnar
  - DBF
  - Oracle
  - XML
- REX for any format

## Timport Schemas

- Options followed by fields
- Populate fields or variables
- DBF creates a copy of the table
- field name type tag default-value

## Example XML Schema

```
xml
xmldatasetlevel 1
xmlns http://www.coolreports.com
xmlns:mc http://www.mycompany.com
field name varchar reports/report/name
field description varchar reports/report/description
field tpsid long reports/report/mc:tpsid
field tpsComment varchar reports/report/mc:tpsComment
```

## Example XML Schema

```
database xx
trimspace
xml
field yy.id counter - join
field yy.jkl varchar abc/def/jkl ''
field yy.xyz varchar abc/def/xyz ''
field xx.id counter - join
field xx.def varchar abc/def ''
field xx.defe varchar abc/def@e ''
```

## Example CSV Schema

```
# indicate csv format with a delimiter of |
csv |
# name type tag
field CustID varchar(10) 1
field Company varchar(80) 2
field Address varchar(80) 3
field City varchar(20) 4
field State varchar(10) 5
field Zip varchar(10) 6
field Country varchar(10) 7
field Phone varchar(20) 8
```

## Sample REX Schema

```
recdelim \x0a
trimspace
datefmt dd/mmm/yyyy:HH:MM:SS
recexpr >>^\P=[^ ]+ +-[^ ]* +-[^ ]* +\[=[^\\]+\]=
+=[^ ]+ +[^\s"]+ HTTP/?\d\d\d\d?\.?\d\d\d\d?"+ +[^\s"]+ +[^\s"]+
\x0a]+
# name type tag
default_val
field Client varchar(40) 2
field Ident varchar(40) 5
field User varchar(20) 8
field Date date 11
field Method varchar(10) 15
field Request varchar(100) 17
field Protocol varchar(10) 18-21
field Status integer 24
field Bytes integer 26
```

## Default REX

- >> \$\Rname:=\P[\x20\x09]\*[^\x0d\x0a]+
  - >> anchor
  - \$ Newline
  - \R respect case
  - =, + \* repetition operators
  - [] character classes

## REX Features

- Fast
- Deterministic
- Longest preceding sequence for repetition
- Repetition operators are greedy
- Directional
  - >><tag>=</tag>+</tag>=
- Not line oriented. \$, ^ match newline

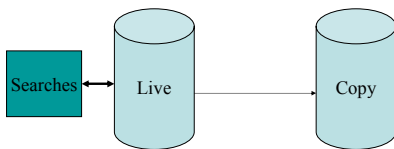
## Other Import Methods

- C API (NCGDIRECT)
  - flexible for complex data sources that timport can't handle
- Perl DBD
  - useful to copy data from another database, or if you already have Perl loader
- ODBC & JDBC
  - networked, poor performance

## Processing Batches

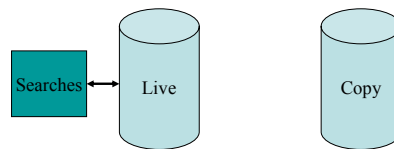
- Generate a lot of updates or inserts quickly
- Potential problems
  - disk utilization
  - database locks
  - out of date Metamorph index
- Copy of database solves problems

## Batch Updates



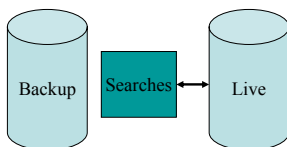
Copy Database

## Batch Updates



Batch Updates - Reindex

## Batch Updates



Make Copy Live

## Continuous Updates

- Chkind to maintain Metamorph indexes
- May want manual updates before peak search load
- New data is often most wanted

## Performance Impact

- CPU load to process inserts
- Disk IO and cache
- Locking
- Metamorph linear search
- Metamorph index updates
- Redoing complex cached queries

## Standard Tables

- SYSTABLES
- SYSINDEX
- SYSTRIGGER
- SYSPERMS
- SYSLIB
- SYSCOLUMNS
- SYSMETAINDEX
- SYSUSER
- SYSSTATISTICS
- SYSDEP

- Read only except SYSMETAINDEX

## SYSINDEX Types

TYPE	Description
F,f	Metamorph Inverted (being created)
M,m	Metamorph (being created)
B	Regular (BTREE)
3,C	Old style Metamorph (being created)
V	Inverted
T	Temporary
D	Deleted

## Data Types

- counter
- varchar
- indirect
- blob
- strlst

## Security

- Taxis level security
- System level security
- Hide scripts
- Encrypt sensitive data
- Proxy
- vhttpd, entry and exit scripts

## Taxis Level Security

- Table level permissions
- GRANT/REVOKE
- User/Password required to log in
- PUBLIC and \_SYSTEM are default

## System Level Security

- Who has access to the files?
- Files are normal OS files
- Encrypt sensitive data to prevent OS level snooping

## Further Security

- Hide Scripts. Vortex does not need the source to run
  - vhttpd by default will not allow script download
- Using a Vortex Proxy provides more control
- Vhttpd entry/exit scripts