

**ORACLE BRAIN DUMP**

<i>ORACLE 9i PL/SQL</i>	1
<i>CRYSTAL REPORTS 8.5</i>	2
<i>TUNING</i>	3
<i>Database</i>	3
<i>Queries</i>	4
<i>PRO*C</i>	4
<i>ORACLE CALL INTERFACE</i>	5
<i>FORMS</i>	6
<i>ORACLE 8/8i</i>	7
<i>PL/SQL</i>	8
<i>UNIX</i>	9
<i>ORACLE 8i</i>	10
<i>JAVA</i>	11
<i>DATA WAREHOUSE</i>	12
<i>META DATA/INFORMATION DIRECTORIES</i>	13
<i>BUSINESS INTELLIGENCE TOOLS</i>	14
<i>DATABASE DESIGN</i>	15
<i>TUNING COOKBOOK</i>	16

**ORACLE 9i PL/SQL**

SEARCHED CASE

## Philip Wik

A searched CASE allows you to do more than match on a single expression. It allows you to match on expressions:

```
SELECT course_name
      CASE WHEN period >= 1
            AND period <= 3 THEN 'Morning'
      CASE WHEN period >= 4
            AND period <= 6 THEN 'Afternoon'
      ELSE 'After School'
      END
FROM course;
```

### TABLE JOINS

Joins can now be written in the FROM clause, making it easier to follow the logic of a query. Examples:

	Inner Join	Outer Join
Oracle 8i	SELECT a.col,b.col FROM table a table b WHERE a.col = b.col;	SELECT a.col,b.col FROM table a table b WHERE a.col = b.col(+);
Oracle 9i	SELECT a.col,b.col FROM table a INNER JOIN table b ON a.col = b.col;	LEFT OUTER JOIN RIGHT OUTER JOIN FULL OUTER JOIN SELECT a.col,b.col FROM table a LEFT OUTER JOIN table b ON a.col = b.col; SELECT a.col, b.col FROM table a NATURAL FULL OUTER JOIN table b; SELECT a.col, b.col FROM table a FULL OUTER JOIN table b USING (col);

### MULTITABLE INSERTS

They allow you to insert the results of a SELECT query into several different tables at once.

```
INSERT ALL
  WHEN (period=6) THEN
    INTO course_6 (course_name, period)
    Values (course_name,period)
  WHEN (period=3) THEN
    INTO course_3 (course_name, period)
    Values (course_name,period)
  ELSE
    course_other (course_name, period)
    Values (course_name,period)
ELECT course_name, period
FROM course;
```

### MERGE Statement

```
MERGE INTO table
USING data_source
ON (condition)
WHEN MATCHED THEN update_clause
WHEN NOT MATCHED THEN insert_clause;
```

## CRYSTAL REPORTS 8.5

Philip Wik

Visual AccountMate SQL using Seagate Crystal Reports Professional

Report drill-downs

Distributed via HTML and Excel exports

Modified SQL queries through Active Server Pages

**TUNING**

*Database*

Philip Wik

- Use existing tools (LECCO Technologies/SQL Expert, SQL Impact/Quest, BMC, Platinum, Compuware, Oracle Enterprise Manager)
- Identify heavy hitters with V\$SQLAREA /DISK\_READS.
- Use the EXPLAIN Plan (utlxplan), TRACE, and TKPROF
- Tune the INIT.ORA parameters (V\$PARAMETER): DB\_BLOCK\_BUFFERS, SHARED\_POOL\_SIZE, SORT\_AREA\_SIZE.
- Optimize with hints ALL\_ROWS, FIRST\_ROWS
- Partition
- Limit users
- Tune memory—shared pool, sorts, locks, rollback segments, redo logs

#### *Queries*

- Ask the users what the problem queries are.
- Use explicit cursors
- Encapsulate code in stored procedures in packages.
- Bind rather than concatenate dynamic SQL
- Index all joining columns. Force or suppress indexes as necessary.
- Pin packages into memory
- Don't call SELECT FROM DUAL more than once, i.e. USER, SYSDATE
- Avoid OR and NOT in a queries
- Use nested subqueries rather than joins
- Replace NUMERIC with PLS\_INTEGER

**PRO\*C**

Embedded SQL in C

Philip Wik

Makefile compilations, i.e. make -f

Constructed dynamic SQL using passed parameters and cursors for reporting

**ORACLE CALL INTERFACE**

Version 7 and version 8

Philip Wik

N-tier authentication

Provided access to Oracle database

## **FORMS**

4.5 and 6I in an Oracle 9I environment

Developed tables, blocks, LOVs, PL/SQL stored procedures, alerts, canvases

Philip Wik

Performance tuning, including array processing, erasing global variables, and basing data blocks on stored procedures

Developed pre/post DML triggers

## **ORACLE 8/8I**

Logical database design and normalization

- Entity-Relationship modeling

- Denormalizing

- RAID

Oracle architecture

- Data dictionary

  - Internal RDBMS (X\$) Tables

  - Dynamic performance (Y\$) Tables

- Instance

  - SGA

  - Trace files

- Database

Partitioning

- Indexes

Managing database storage

Backup and recovery

Oracle interfaces and utilities

- SQL \*Plus

Philip Wik

- AUTOTRACE
- COPY
- OEM
  - Managing instances, schemas, security, storage, sessions
- Import/Export
- Supplied packages (DBMS\_OUTPUT, DBMS\_APPLICATION\_INFO)
- SQL \*Loader
  - Conventional path
  - Direct path
- Designer
- Oracle Networking
  - Net8
- Oracle Application Server
  - HTTP listener layer

## **PL/SQL**

Types

- Scalar
- %TYPE

Control structures

- CASE
- Loops

Records

- %ROWTYPE

DML

- Bulk binds
- RETURNING clause

Pseudo columns

- CURRVAL and NEXTVAL
- ROWID
- ROWNUM

Privileges

- GRANT
- REVOKE

Philip Wik

Transaction control

- COMMIT
- ROLLBACK

Functions

- SUBSTR
- INSTR
- NLS
- TO\_

Cursors

- Explicit
- Implicit
- NO\_DATA\_FOUND versus %NOTFOUND
- SELECT FOR UPDATE

Error handling

- EXCEPTION\_INIT

Collections

- Varrays
- Nested tables

Procedures

Functions

Packages

Pinning

Database triggers

- DML triggers
- Instead-Of triggers

Mutating tables

Native dynamic SQL

UTL\_FILE

DBMS\_SQL

## UNIX

```
#!/bin/ksh
#
# Define the segment to be loaded.
#
SEGMENT=$1
TTABLE=$2
PCTANLZ=$3

SQL_USER=BBBBBB_ARCHIVE; export SQL_USER
SQL_PASSWD=BBBBBB_ARCHIVE; export SQL_PASSWD
#
# Define the file paths.
#
CTLDIR=/uuuuucripts/WHSD/guests/BBBBBB/source/ctl
DATADIR=/uuuuuadata/WHSD/guests/BBBBBB/dock
LOGDIR=/uuuuuadata/WHSD/guests/BBBBBB/logs
BADDIR=/uuuuuadata/WHSD/guests/BBBBBB/bad
SECDIR=/uuuuucripts/WHSD/guests/BBBBBB/source/scripts
TMPDIR=/tmp

DAY=`date +%Y.%m.%d.%H.%M`

. /uuuuucripts/WHSD/admin/setenv/setOracle.sh
```

## Philip Wik

```
chmod 777 ${DATADIR}/BBBBBB2.TST.A701RP85.${SEGMENT}.ASC
sqlload ${SQL_USER}/${SQL_PASSWD} control=${CTLDIR}/eur_trunc_load_ged_${SEGMENT}.ctl \
  data=${DATADIR}/BBBBBB2.TST.A701RP85.${SEGMENT}.ASC \
  log=${LOGDIR}/adwdev_BBBBBB_${SEGMENT}.log_${DAY} \
  bad=${BADDIR}/adwdev_BBBBBB_${SEGMENT}.bad_${DAY}
#
# Check for a successful run: records loaded = records in file.
#
export TOT=`wc ${DATADIR}/BBBBBB2.TST.A701RP85.${SEGMENT}.ASC | awk '{print $1}'`
echo "Records In      =" $TOT
foo=`grep -i "successfully loaded" ${LOGDIR}/adwdev_BBBBBB_${SEGMENT}.log_${DAY} | awk
'{print $1}'`
echo "Records Loaded =" $foo
#
if [ "$foo" -eq "$TOT" ] ; then
  echo "Successfully loaded ${SEGMENT}\n"
  sqlplus -s ${SQL_USER}/${SQL_PASSWD} <<EOF
ANALYZE TABLE ${TTABLE} ESTIMATE STATISTICS SAMPLE ${PCTANLZ} PERCENT
/
EOF
  echo "Statistics analyzed for table ${TTABLE}, percent=${PCTANLZ}\n"
else
  exec 3<${SECDIR}/BBBBBB_failed.dat
  while read -u3 email
  do
    echo "SUBJECT: TMPUUUUU - Failed to load BBBBBB ${SEGMENT} "
    >${TMPDIR}/error_page.tmp
    mail $email <${TMPDIR}/error_page.tmp
  done
  echo "Failed to load BBBBBB ${SEGMENT}\n"
fi
exit
```

---

## Oracle 8l

JVM

Materialized view

Transportable tablespace

Aggregate operators CUBE and ROLLUP

Function-based indexes

Embedded dynamic SQL

PL/SQL bulk binds

Schema manager

Net8 network communication protocol

Instance – set of processes

System global area

Background processes—DBW0, LGWR, SMON, PMON, RECO,

Ps -ef | EURM

Outer join--- return results even when the second table doesn't have any records corresponding to the first

Inner join—corresponding records in both tables

Subqueries -- queries within queries

Noncorrelated issue independent of the parent query in example students books loaded

Correlated execute once for each row returned by parent query which uses values from the row.

## Philip Wik

Inline view select from ( select ) where

Decode inline if

INTERSECT  
MINUS

Updates based on subqueries

Optimization OPTIMIZER\_MODE

CBO  
RBO

EXPLAIN plan

TKPROF trace

CPU, disk reads, library cache'

Datatypes

Loops

Exceptions

Stored procedures CREATE OR REPLACE PROCEDURE AS IN,OUT, IN OUT

Massive denormalization STAR

Terabytes of disk storage

Full table scan

Lecco tech

Declare  
Begin  
Exception  
End

Package—specification & body

TOAD  
PL/SQL Developer  
SQL/PLUS

## JAVA

HTTP server is iPlanet Web Server 4.0 on UNIX HP with servlets and JSP enabled.

Client-side JavaScripting Netscape and IE browser compatible

Philip Wik

SSJS

javac

Interface – abstract superclass

Collections

Multithreading

Reflection

Events

JDK applets/appletviewer

CGI/Perl

Enterprise Java Beans

Security is LDAP/Perl/COM wrapper

UNIX/Windows NT

AWT

Class bytecodes

JAR – Java archive

JIT

Overloaded constructors/instance methods/objects

Class variables and methods

Primitive variables arrays/object reference arrays

Extended class- inheritance

Java.sun.com

Development tools Visual Age and WebSphere

## **DATA WAREHOUSE**

Constructed EIS executive information systems

DSS Decision Support System

OLAP (analytical) vs. OLTP (transaction)

Used Impromptu from Cognos – extracts data from Oracle

Philip Wik

PowerPlay Transformer converts data into multidimensional hypercubes

Business Objects make selections from browse windows to generate SQL code behind the scenes

API is ODBC

Used Prism Technology for data scrubbing

Star Schema—normalized fact tables and dimension tables

SQL queries can use pre-defined join paths between fact and dimension tables

## **META DATA/INFORMATION DIRECTORIES**

Information about the contents of data warehouses

XML and PDFs SOAP nbodes

Mapping documents

Rules that define data clean up and enhancement

Data that describes other data

Maintained in Oracle Data Dictionary

Fact/dimension relationships and rules

Philip Wik

## **BUSINESS INTELLIGENCE TOOLS**

Low level aggregates access via MicroStrategy, Q+E/VB and SAS

## **DATABASE DESIGN**

Create a business/enterprise model and definition

Dimensional business model segregates business entities into facts and dimensions

Diagrams, facts, dimensions, hierarchies, relationships, candidate keys

Use cases—interactions of business entities and business functions

Decision tables

Create a data entity model

DFD data flow diagram

Create a database design ERD → Instance chart

Create table definitions

Create a relational data model

Normalize

1<sup>st</sup> attribute are single values

2<sup>nd</sup> Unique id

3<sup>rd</sup> Each table has a primary key

Denormalize

## TUNING COOKBOOK

- Use existing tools (LECCO Technologies/SQL Expert, SQL Impact/Quest)
- Use explicit cursors
- Identify heavy hitters with V\$SQLAREA /DISK\_READS.
- Encapsulate code in stored procedures in packages.
- Bind rather than concatenate dynamic SQL
- Index all joining columns. Force or suppress indexes as necessary.
- Pin packages into memory
- Don't call SELECT FROM DUAL more than once, i.e. USER, SYSDATE
- Use the EXPLAIN Plan (utlxplan), TRACE, and TKPROF
- Tune the INIT.ORA parameters (V\$PARAMETER):  
DB\_BLOCK\_BUFFERS, SHARED\_POOL\_SIZE,  
SORT\_AREA\_SIZE.
- Optimize with hints ALL\_ROWS, FIRST\_ROWS
- Avoid OR and NOT in a queries
- Use nested subqueries rather than joins
- Partition
- Limit users
- Replace NUMERIC with PLS\_INTEGER
- Ask the users what the problem queries are.