ORACLE®

YESSQL!

Thomas Kyte
http://asktom.oracle.com/

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

The beginning…

June 1979 – Version 2

- First Commercial SQL Database
- Fairly Impressive First SQL
  - Joins, Subqueries
  - Unique extensions (at the time) – Outer Joins, Connect By
- Simple Server, no transactions, reliability was a “future”
- Portable

March 1983 – Even more portability, Version 3

- All Code now written in a new language – ‘C’

- Architectural Changes
  - Transactions
  - Non-blocking reads (but no read consistency)
  - AI/BI journaling

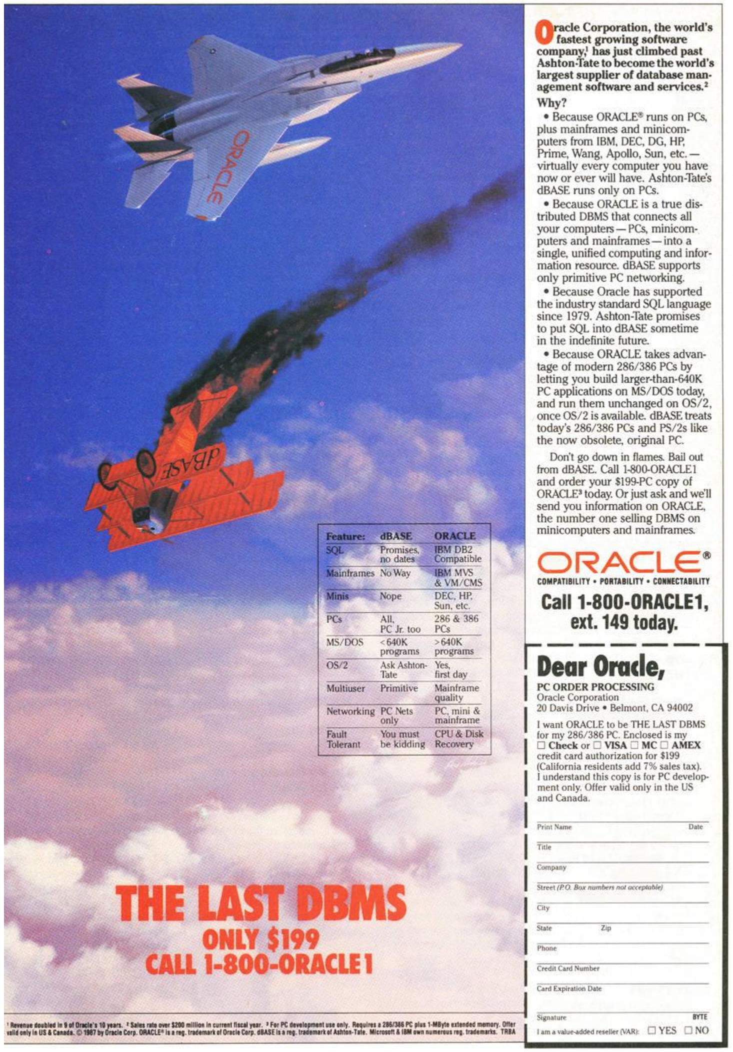
- Oracle gets its name…

October 1984 – Reliability, Oracle Version 4

- Larger Installed Base, More Reliable, Even More Portable
- Architectural Changes
  - Read Consistency
- Support For everything from Mainframe to Desktop
  - Ran on MS-DOS in less than 640k

April 1985 – Cooperative Server, Version 5

- First Client/Server SQL Database
- Distributed & Clustered
  - SQL*Net
  - Distributed Query
  - SQL*Connect for heterogeneous environments
  - Parallel Server
- Portability
  - Able to go beyond 640k limit on PC's
  - Macintosh support
- SQL_TRACE! select trace('sql',1),1 from dual;



THE LAST DBMS

ONLY $199

CALL 1-800-ORACLE1

Feature:	dBASE	ORACLE
SQL	Promises, no dates	IBM DB2 Compatible
Mainframes	No Way	IBM MVS & VM/CMS
Minis	Nope	DEC, HP, Sun, etc.
PCs	All, PC Jr. too	286 & 386 PCs
MS/DOS	<640K programs	>640K programs
OS/2	Ask Ashton-Tate	Yes, first day
Multiuser	Primitive	Mainframe quality
Networking	PC Nets only	PC, mini & mainframe
Fault Tolerant	You must be kidding	CPU & Disk Recovery

Oracle Corporation, the world's fastest growing software company,[1] has just climbed past Ashton-Tate to become the world's largest supplier of database management software and services.[2]

Why?

- Because ORACLE® runs on PCs, plus mainframes and minicomputers from IBM, DEC, DG, HP, Prime, Wang, Apollo, Sun, etc. — virtually every computer you have now or ever will have. Ashton-Tate's dBASE runs only on PCs.
- Because ORACLE is a true distributed DBMS that connects all your computers — PCs, minicomputers and mainframes — into a single, unified computing and information resource. dBASE supports only primitive PC networking.
- Because Oracle has supported the industry standard SQL language since 1979. Ashton-Tate promises to put SQL into dBASE sometime in the indefinite future.
- Because ORACLE takes advantage of modern 286/386 PCs by letting you build larger-than-640K PC applications on MS/DOS today, and run them unchanged on OS/2, once OS/2 is available. dBASE treats today's 286/386 PCs and PS/2s like the now obsolete, original PC.

Don't go down in flames. Bail out from dBASE. Call 1-800-ORACLE1 and order your $199-PC copy of ORACLE[3] today. Or just ask and we'll send you information on ORACLE, the number one selling DBMS on minicomputers and mainframes.

ORACLE®

COMPATIBILITY • PORTABILITY • CONNECTABILITY

Call 1-800-ORACLE1, ext. 149 today.

Dear Oracle,

PC ORDER PROCESSING
Oracle Corporation
20 Davis Drive • Belmont, CA 94002

I want ORACLE to be THE LAST DBMS for my 286/386 PC. Enclosed is my □ **Check** or □ **VISA** □ **MC** □ **AMEX** credit card authorization for $199 (California residents add 7% sales tax). I understand this copy is for PC development only. Offer valid only in the US and Canada.

Print Name Date

Title

Company

Street *(P.O. Box numbers not acceptable)*

City

State Zip

Phone

Credit Card Number

Card Expiration Date

Signature BYTE

I am a value-added reseller (VAR): □ YES □ NO

[1] Revenue doubled in 9 of Oracle's 10 years. [2] Sales rate over $200 million in current fiscal year. [3] For PC development use only. Requires a 286/386 PC plus 1-MByte extended memory. Offer valid only in US & Canada. © 1987 by Oracle Corp. ORACLE® is a reg. trademark of Oracle Corp. dBASE is a reg. trademark of Ashton-Tate. Microsoft & IBM own numerous reg. trademarks. TRBA

July 1988 – New Architecture & TPO, version 6

- Server Rewritten
  - Buffer cache added
  - *Row Level Locking*
  - Online backup/recovery
  - SQL*Dba
- SMP support
- *PL/SQL executes in the server*
- *Referential Integrity (ok, the syntax anyway!)*

June 1992 – Cooperative Server, Oracle Version 7

- Shared SQL
- CBO
- First ANSI/ISO SQL (including declarative RI that actually did something)
- Roles
- Stored Procedures
- Triggers
- Dbms_alert, Dbms_Pipes
- Automatic 2PC
- Snapshots
- XA
- Transparent and Procedural Gateways

7.1, 7.2, 7.3…

- SQL92 Certification
- Symmetric Data Replication
- Parallel operations
- Server Manager
- Read only Tablespaces

- Resizable, autoextend datafiles
- Shrink Rollback Segments
- Unrecoverable DDL
- Check summing
- Sorting improvements
- Ref Cursors

- Dbms_application_info, Dbms_job

- Partitioned Views
- Bitmapped Indexes
- Index rebuilds
- DBV
- Unlimited extents
- Histograms
- Hash joins
- Context
- SDO

8.0, 8*i*

- Objects in the database
- External Procedures
- AQ
- 1,000 columns/table
- Images, Video, Context, Time Series, Spatial, Locator
- RMAN
- IOT’s
- Reverse Key Indexes
- New Rowid Format

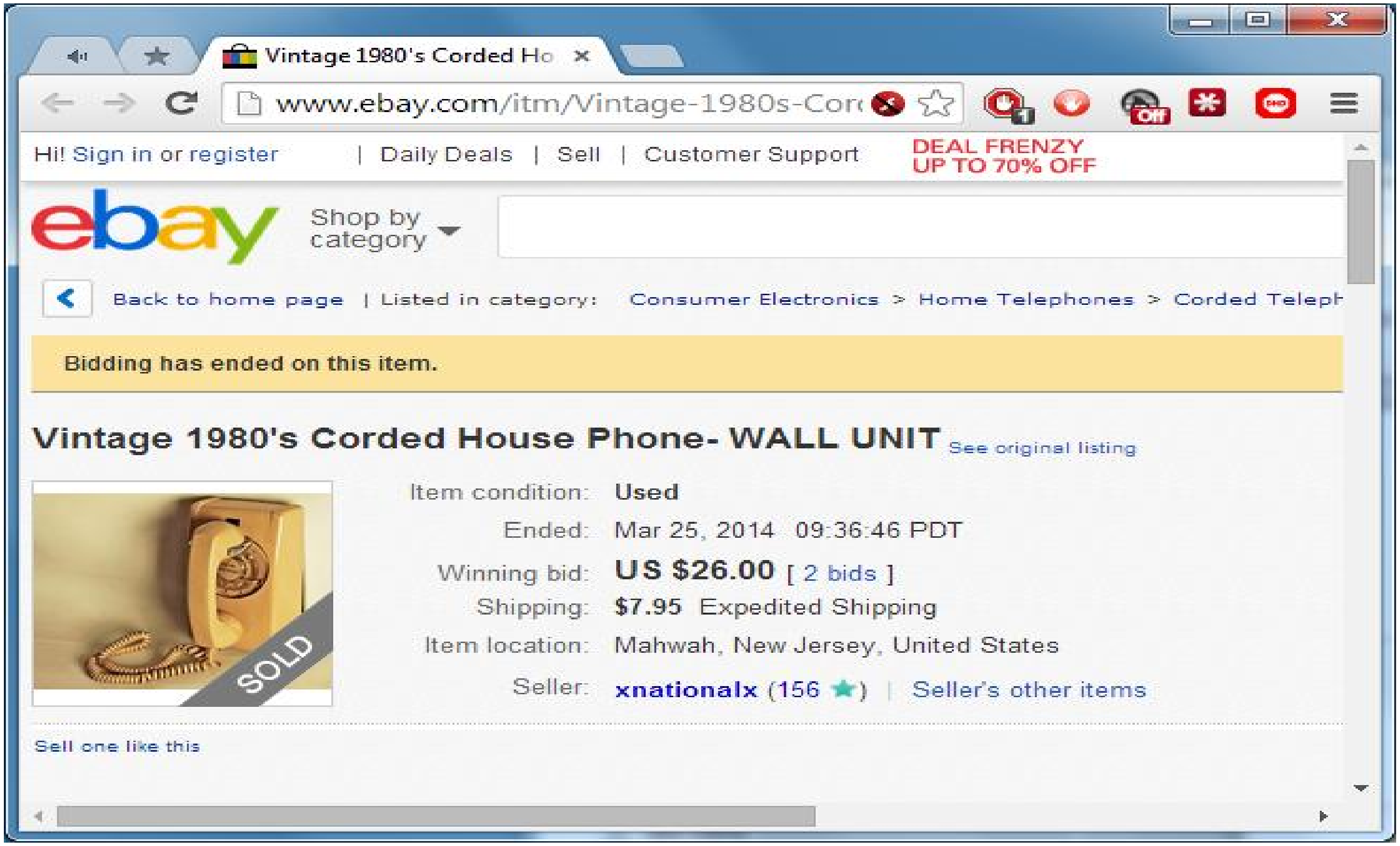
- Java option
- Transportable Tablespaces
- Function Based Indexes
- Query Plan Stability
- Drop Column
- VPD
- DBMS_STATS
- Locally Managed Tablespaces
- Online Index operations
- Alter table move
- Resource Manager
- Data Guard

9*i*, 10*g*, 11*g*, 12*c*…

- A lot about the DBA
  - Auto this and that, undo management, memory management…
  - Real Application Clusters
  - Stored Parameter Files
  - Compression
  - ASM
  - AWR
  - ADDM
  - Advisors
  - In memory
  - And so on…

If you can remember all of these things…

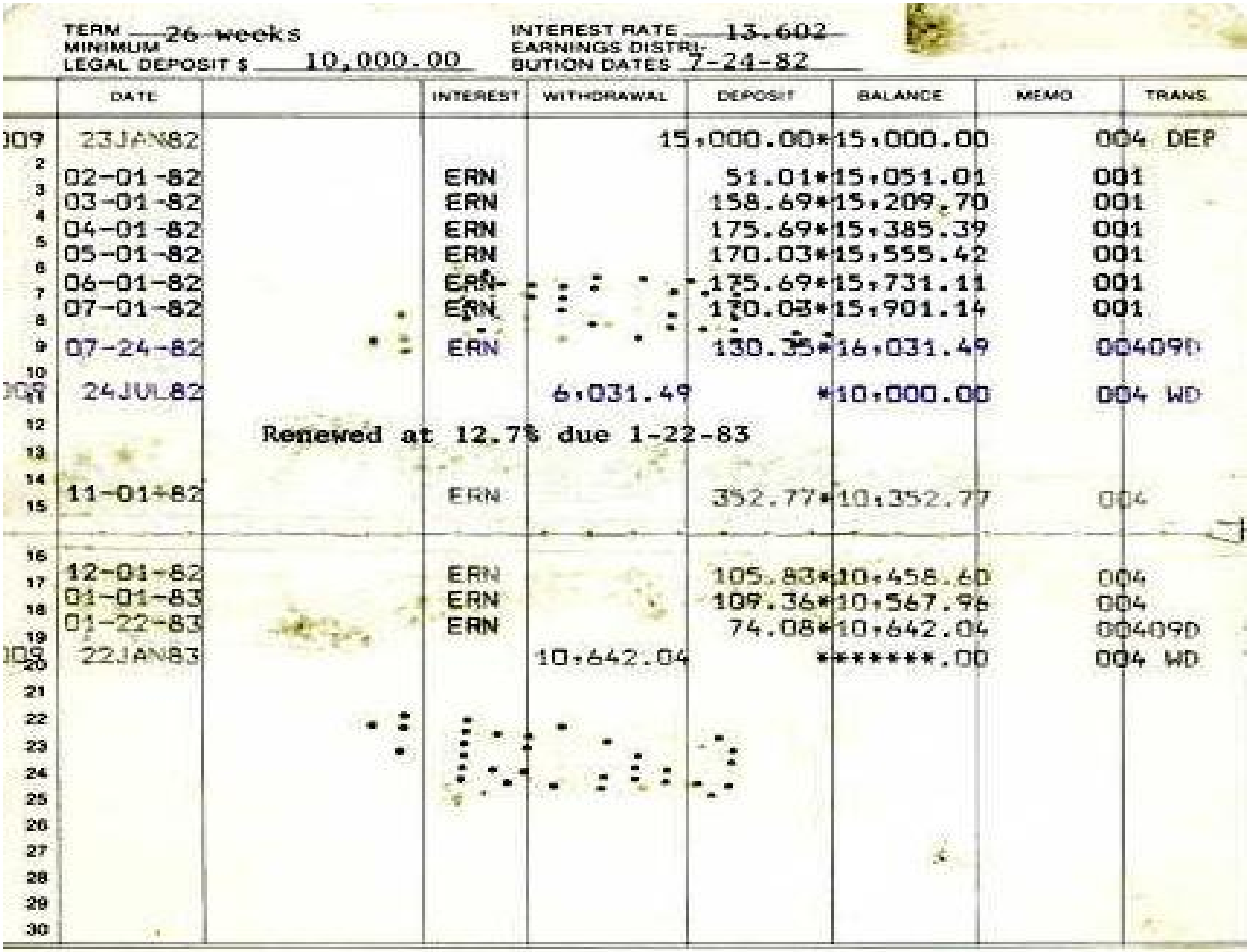
You were around in the beginning, when SQL was just being invented



Bidding has ended on this item.

Vintage 1980's Corded House Phone- WALL UNIT See original listing

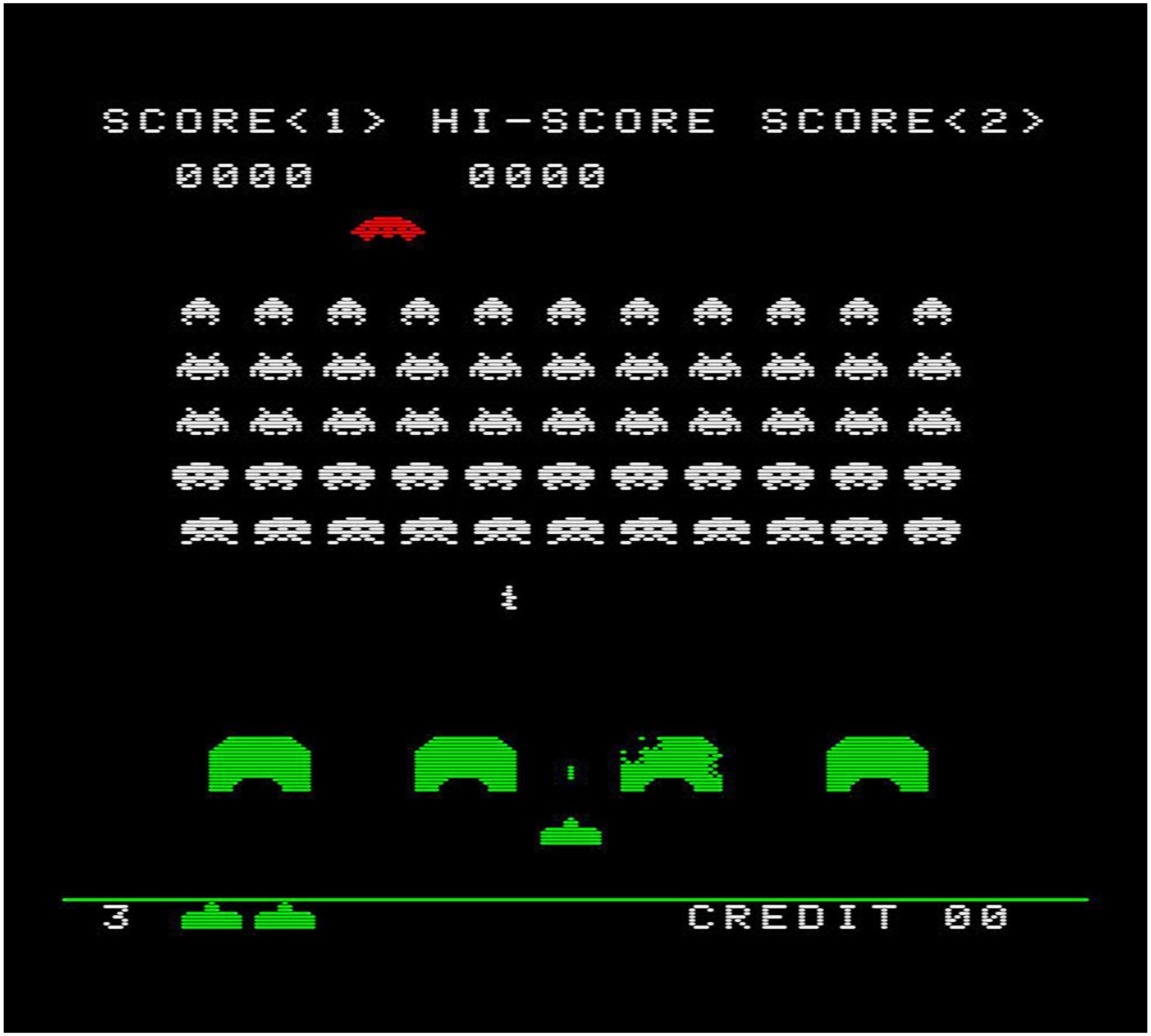
Item condition:	Used
Ended:	Mar 25, 2014 09:36:46 PDT
Winning bid:	US $26.00 [2 bids]
Shipping:	$7.95 Expedited Shipping
Item location:	Mahwah, New Jersey, United States
Seller:	xnationalx (156 ★) | Seller's other items



TERM 26-weeks
MINIMUM
LEGAL DEPOSIT $ 10,000.00

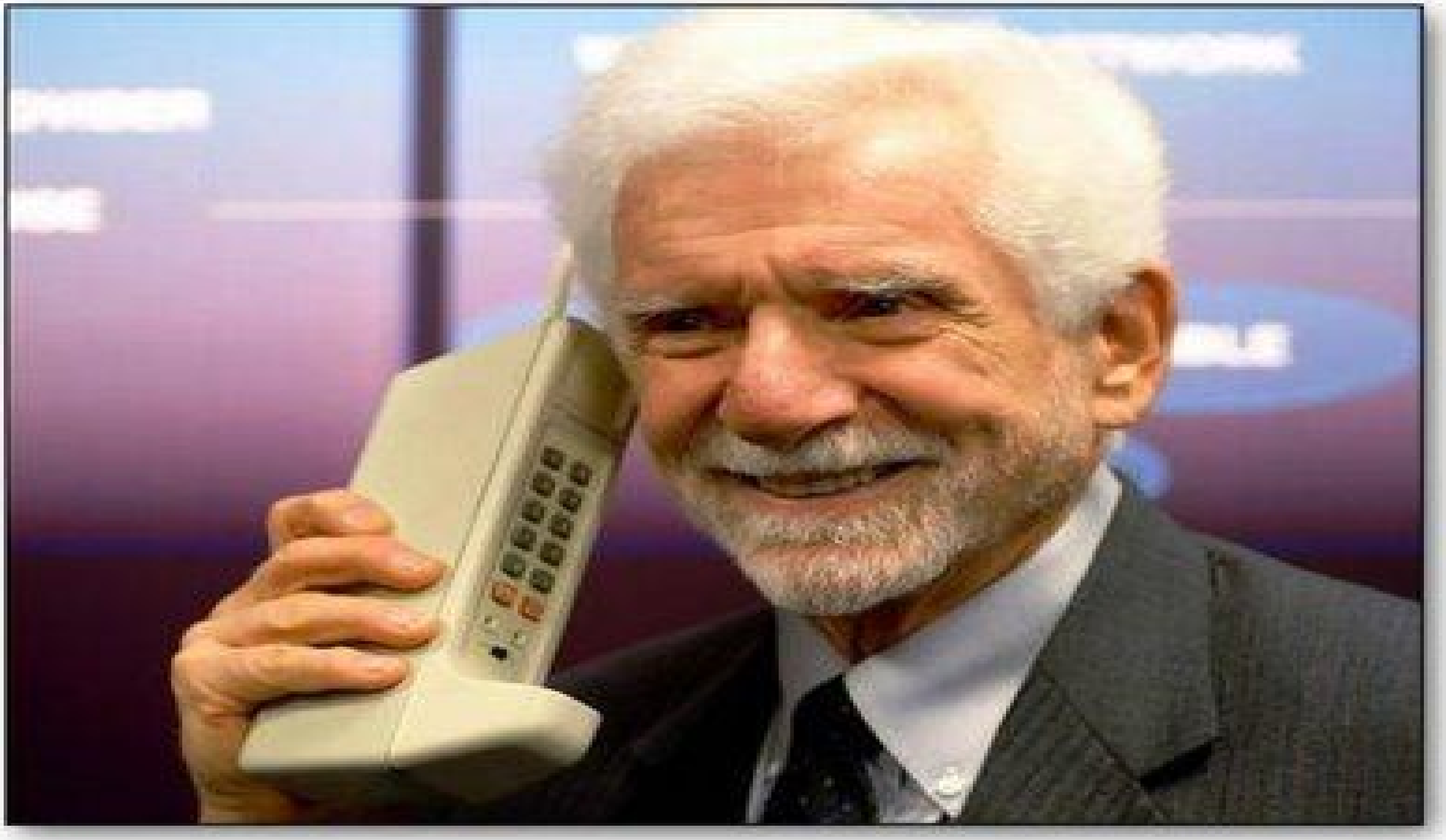
INTEREST RATE 13.602
EARNINGS DISTRI-
BUTION DATES 7-24-82

	DATE		INTEREST	WITHDRAWAL	DEPOSIT	BALANCE	MEMO	TRANS.
009	23JAN82				15,000.00*	15,000.00		004 DEP
	02-01-82		ERN		51.01*	15,051.01		001
	03-01-82		ERN		158.69*	15,209.70		001
	04-01-82		ERN		175.69*	15,385.39		001
	05-01-82		ERN		170.03*	15,555.42		001
	06-01-82		ERN		175.69*	15,731.11		001
	07-01-82		ERN		170.03*	15,901.14		001
	07-24-82		ERN		130.35*	16,031.49		00409D
008	24JUL82			6,031.49		*10,000.00		004 WD
		Renewed at 12.7% due 1-22-83						
	11-01-82		ERN		352.77*	10,352.77		004
	12-01-82		ERN		105.83*	10,458.60		004
	01-01-83		ERN		109.36*	10,567.96		004
	01-22-83		ERN		74.08*	10,642.04		00409D
009	22JAN83			10,642.04		*******.00		004 WD


SCORE<1> HI-SCORE SCORE<2>
0000 0000
3
CREDIT 00

13 U 2
12
11
10
9
8
7
VHF
UHF
BRIGHT
CONTRAST
OFF-VOL.
GoldStar

2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37
JERROLD

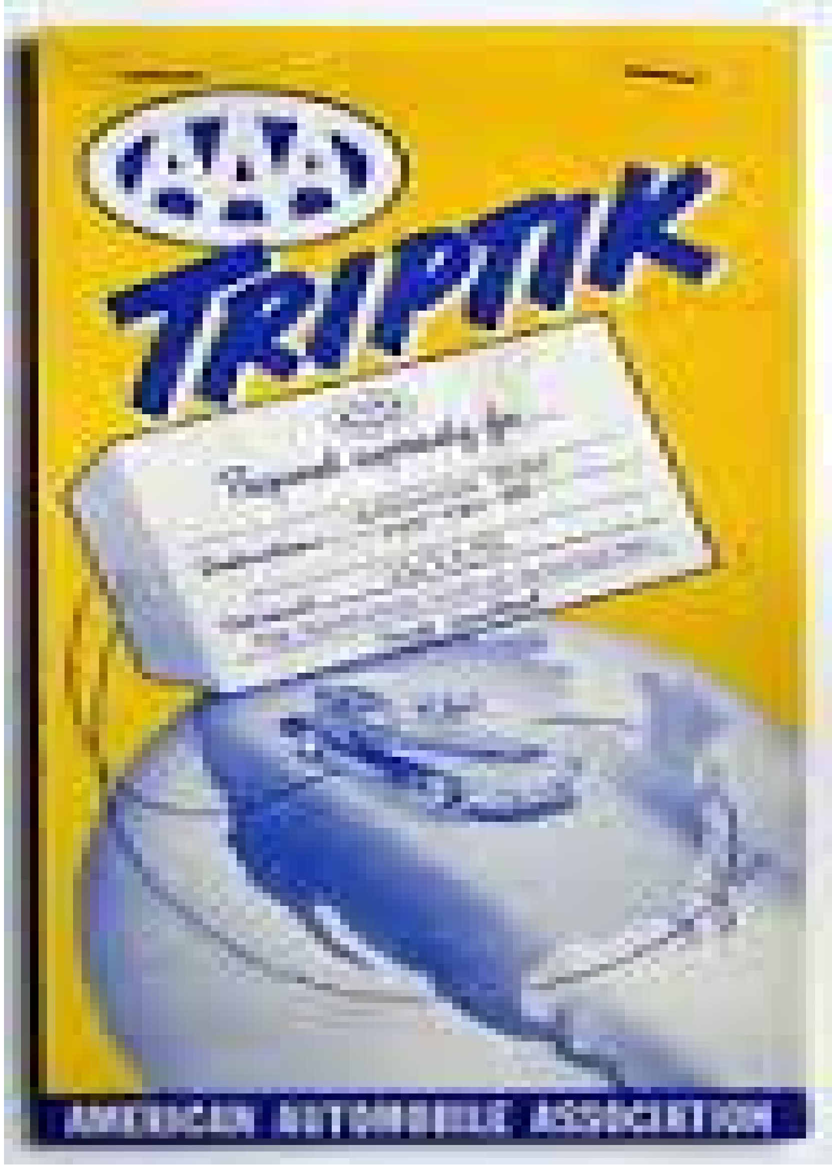


[illegible]
[illegible]
APO [illegible] % P.M.
San Francisco California

Free Air Mail

VIA AIR MAIL

Mr. Elmer R. [illegible]
#2 Market Street
Harrisburg, Pennsylvania, U.S.A.

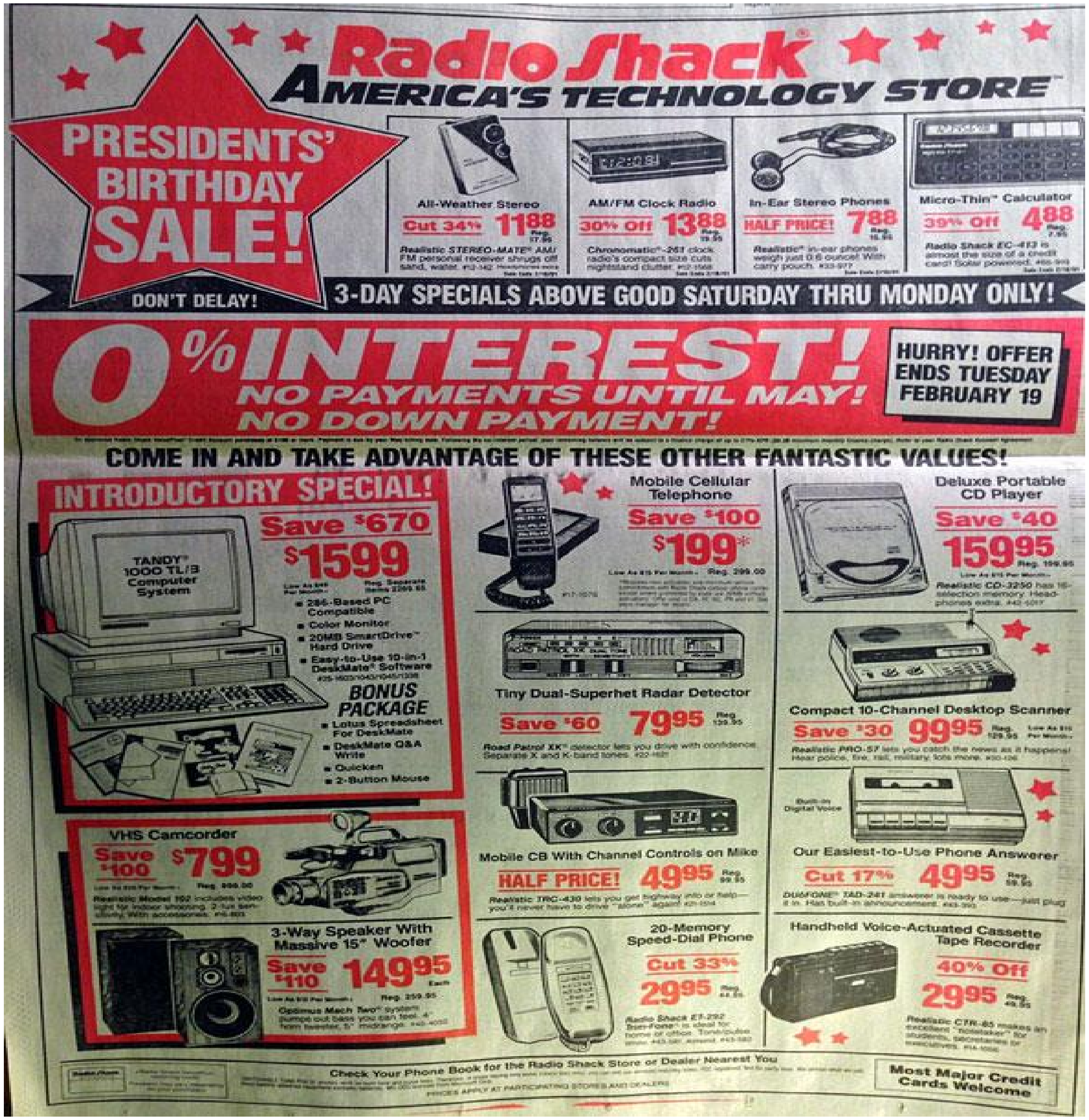

A.A.A.
TRIPTIK
Prepared expressly for
AMERICAN AUTOMOBILE ASSOCIATION

3,054.82 then

$5,100 today

3,054.82 then

$5,100 today



Radio Shack®

AMERICA'S TECHNOLOGY STORE™

PRESIDENTS' BIRTHDAY SALE!

DON'T DELAY!

All-Weather Stereo
Cut 34% 11.88 Reg. 17.95
Realistic STEREO-MATE™ AM/FM personal receiver shrugs off sand, water. #12-142 Headphones extra
Sale Ends 2/18/91

AM/FM Clock Radio
30% Off 13.88 Reg. 19.95
Chronomatic®-263 clock radio's compact size cuts nightstand clutter. #12-1568
Sale Ends 2/18/91

In-Ear Stereo Phones
HALF PRICE! 7.88 Reg. 15.95
Realistic® in-ear phones weigh just 0.6 ounce! With carry pouch. #33-977
Sale Ends 2/18/91

Micro-Thin™ Calculator
39% Off 4.88 Reg. 7.95
Radio Shack EC-413 is almost the size of a credit card! Solar powered. #65-913
Sale Ends 2/18/91

3-DAY SPECIALS ABOVE GOOD SATURDAY THRU MONDAY ONLY!

0% INTEREST!
NO PAYMENTS UNTIL MAY!
NO DOWN PAYMENT!

HURRY! OFFER ENDS TUESDAY FEBRUARY 19

[illegible]

COME IN AND TAKE ADVANTAGE OF THESE OTHER FANTASTIC VALUES!

INTRODUCTORY SPECIAL!

TANDY® 1000 TL/3 Computer System

Save $670
$1599
Low As $49 Per Month
Reg. Separate Items 2269.85

- 286-Based PC Compatible
- Color Monitor
- 20MB SmartDrive™ Hard Drive
- Easy-to-Use 10-in-1 DeskMate® Software

#25-1603/1040/1045/1308

BONUS PACKAGE
- Lotus Spreadsheet For DeskMate
- DeskMate Q&A Write
- Quicken
- 2-Button Mouse

VHS Camcorder
Save $100 $799
Low As $26 Per Month
Reg. 899.00
Realistic Model 102 includes video light for indoor shooting. 2-lux sensitivity. With accessories. #16-803

3-Way Speaker With Massive 15" Woofer
Save $110 149.95 Each
Low As $10 Per Month
Reg. 259.95
Optimus Mach Two™ system pumps out bass you can feel. 4" horn tweeter, 5" midrange. #40-4032

Mobile Cellular Telephone
Save $100
$199*
Low As $15 Per Month
Reg. 299.00
[illegible]
#17-1076

Tiny Dual-Superhet Radar Detector
Save $60 79.95 Reg. 139.95
Road Patrol XK™ detector lets you drive with confidence. Separate X and K-band tones. #22-1621

Mobile CB With Channel Controls on Mike
HALF PRICE! 49.95 Reg. 99.95
Realistic TRC-430 lets you get highway info or help—you'll never have to drive "alone" again! #21-1504

20-Memory Speed-Dial Phone
Cut 33% 29.95 Reg. 44.95
Radio Shack ET-292 Tam-Fone® is ideal for home or office. Tone/pulse. #43-581, Almond #43-582

Deluxe Portable CD Player
Save $40
159.95 Reg. 199.95
Low As $15 Per Month
Realistic CD-3250 has 16-selection memory. Headphones extra. #42-5017

Compact 10-Channel Desktop Scanner
Save $30 99.95 Reg. 129.95
Low As $15 Per Month
Realistic PRO-57 lets you catch the news as it happens! Hear police, fire, rail, military, lots more. #20-126

Built-in Digital Voice

Our Easiest-to-Use Phone Answerer
Cut 17% 49.95 Reg. 59.95
DUOFONE® TAD-241 answerer is ready to use—just plug it in. Has built-in announcement. #43-390

Handheld Voice-Actuated Cassette Tape Recorder
40% Off
29.95 Reg. 49.95
Realistic CTR-65 makes an excellent "notetaker" for students, secretaries or executives. #14-1050

Check Your Phone Book for the Radio Shack Store or Dealer Nearest You

[illegible]

PRICES APPLY AT PARTICIPATING STORES AND DEALERS

Most Major Credit Cards Welcome

Press SPACE BAR to continue

Save 200^{00} on the 2200 "Everything System"

- Realistic STA-2200 AM/FM Stereo Receiver
- Two Mach One Floor Speakers
- LAB-400 Turntable with Base, Dust Cover, [illegible] Magnetic Cartridge
- SCT-[illegible] 3-Head Stereo Cassette Deck with Dolby NR System
- [illegible] Stereo Headphones

Reg. Separate Items [illegible] **1699^00**

Save 140^00

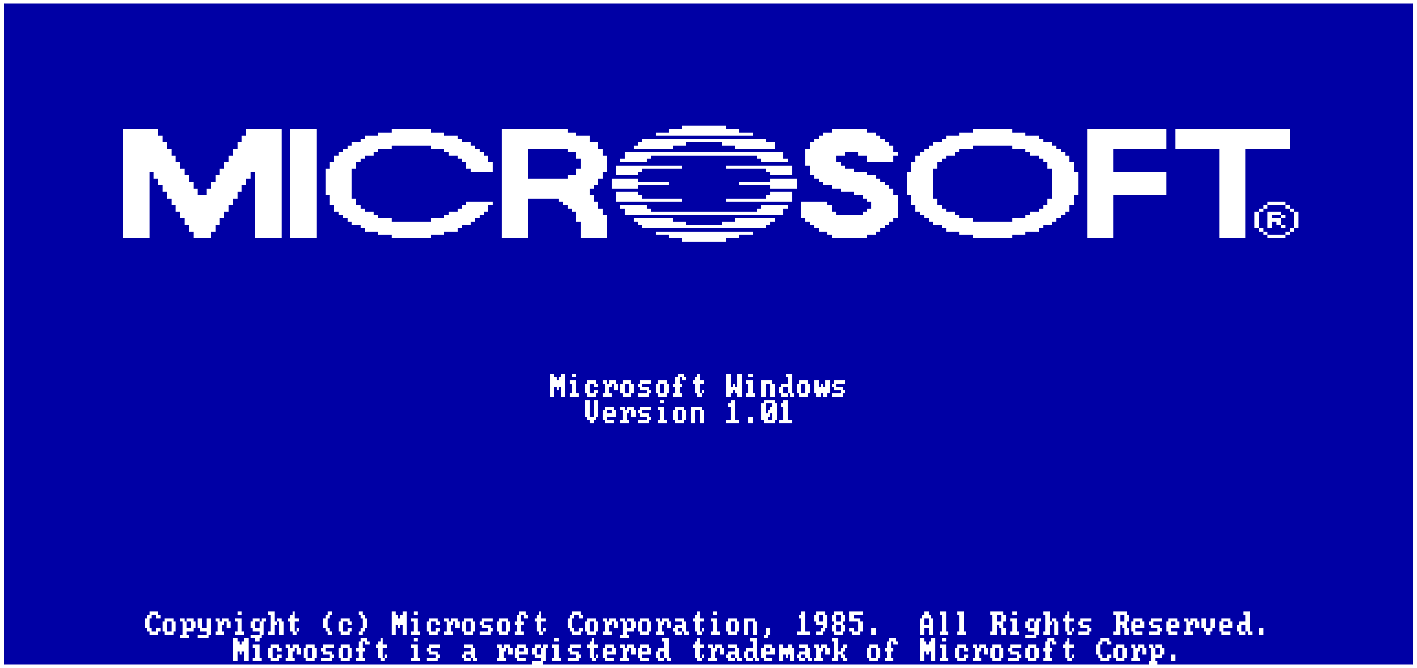
Reg. Separate Items [illegible] **899^00**

- Realistic STA-2200 AM/FM Stereo Receiver
- Two Optimus-[illegible] Floor Speakers
- LAB-[illegible] Turntable with Base, Dust Cover, [illegible] Magnetic Cartridge

Save 150^00

Reg. Separate Items [illegible] **949^00**

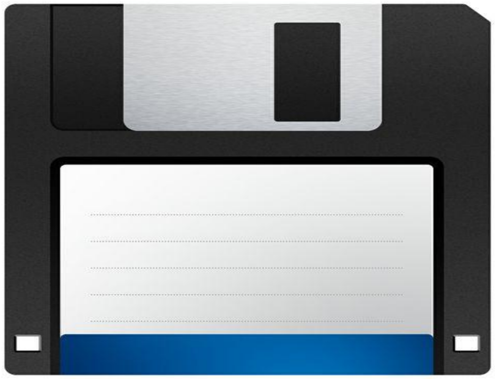
- Realistic STA-2200 AM/FM Stereo Receiver
- Two Optimus-[illegible] Floor Speakers
- LAB-[illegible] Turntable with Base, Dust Cover, [illegible] Magnetic Cartridge



Microsoft Windows
Version 1.01

Copyright (c) Microsoft Corporation, 1985. All Rights Reserved.
Microsoft is a registered trademark of Microsoft Corp.

360k – 1.2mb

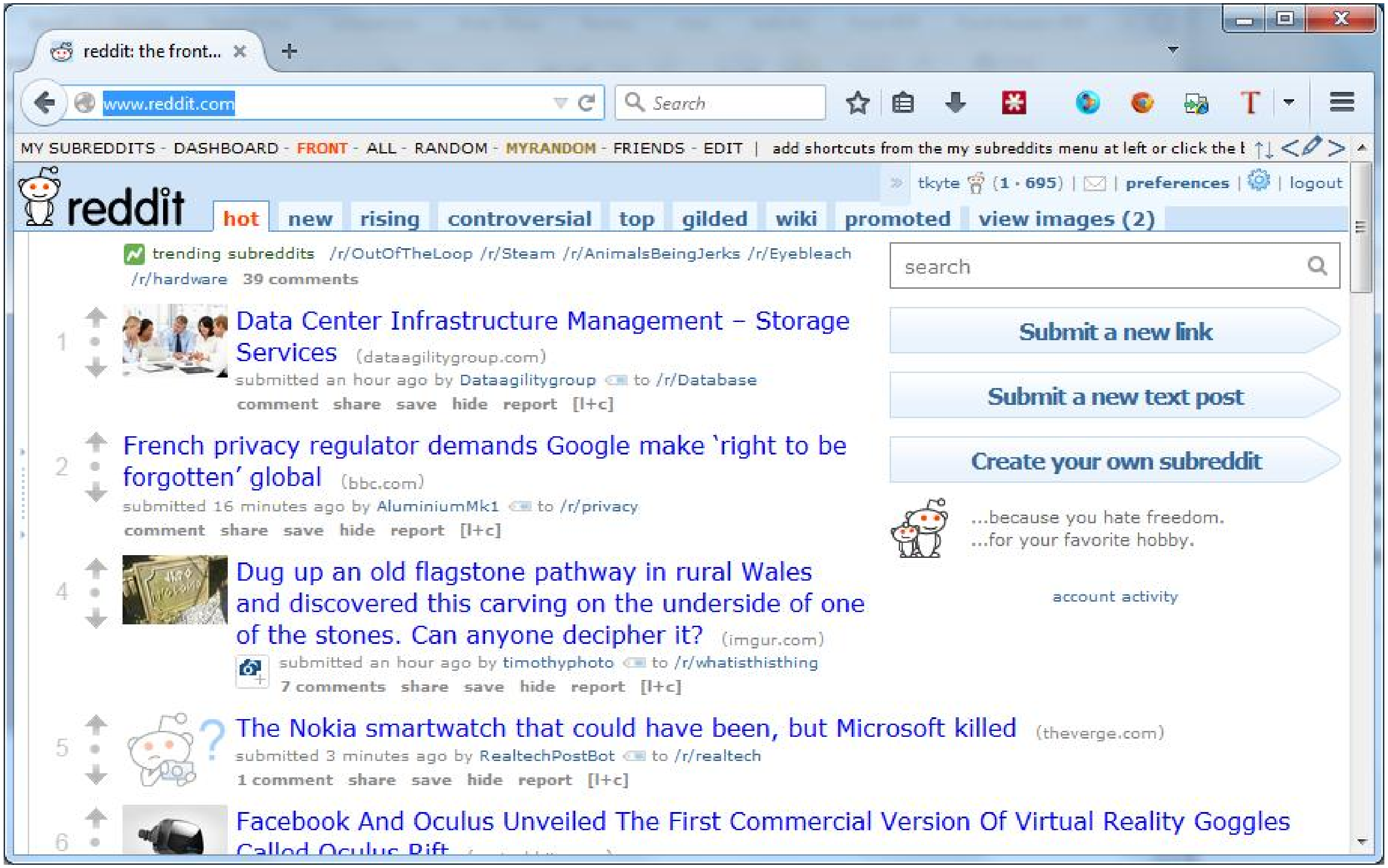


1.44mb

IBM

Easy to use keyboard. Up to 25 character correction memory. 13 characters per second printing speed.

SQL can be important for that interview


reddit: the front...
www.reddit.com
Search
MY SUBREDDITS - DASHBOARD - FRONT - ALL - RANDOM - MYRANDOM - FRIENDS - EDIT | add shortcuts from the my subreddits menu at left or click the t
reddit
hot
new
rising
controversial
top
gilded
wiki
promoted
view images (2)
tkyte (1 · 695) | preferences | logout
search
Submit a new link
Submit a new text post
Create your own subreddit
account activity

trending subreddits /r/OutOfTheLoop /r/Steam /r/AnimalsBeingJerks /r/Eyebleach /r/hardware 39 comments

1 Data Center Infrastructure Management – Storage Services (dataagilitygroup.com)
submitted an hour ago by Dataagilitygroup to /r/Database
comment share save hide report [l+c]

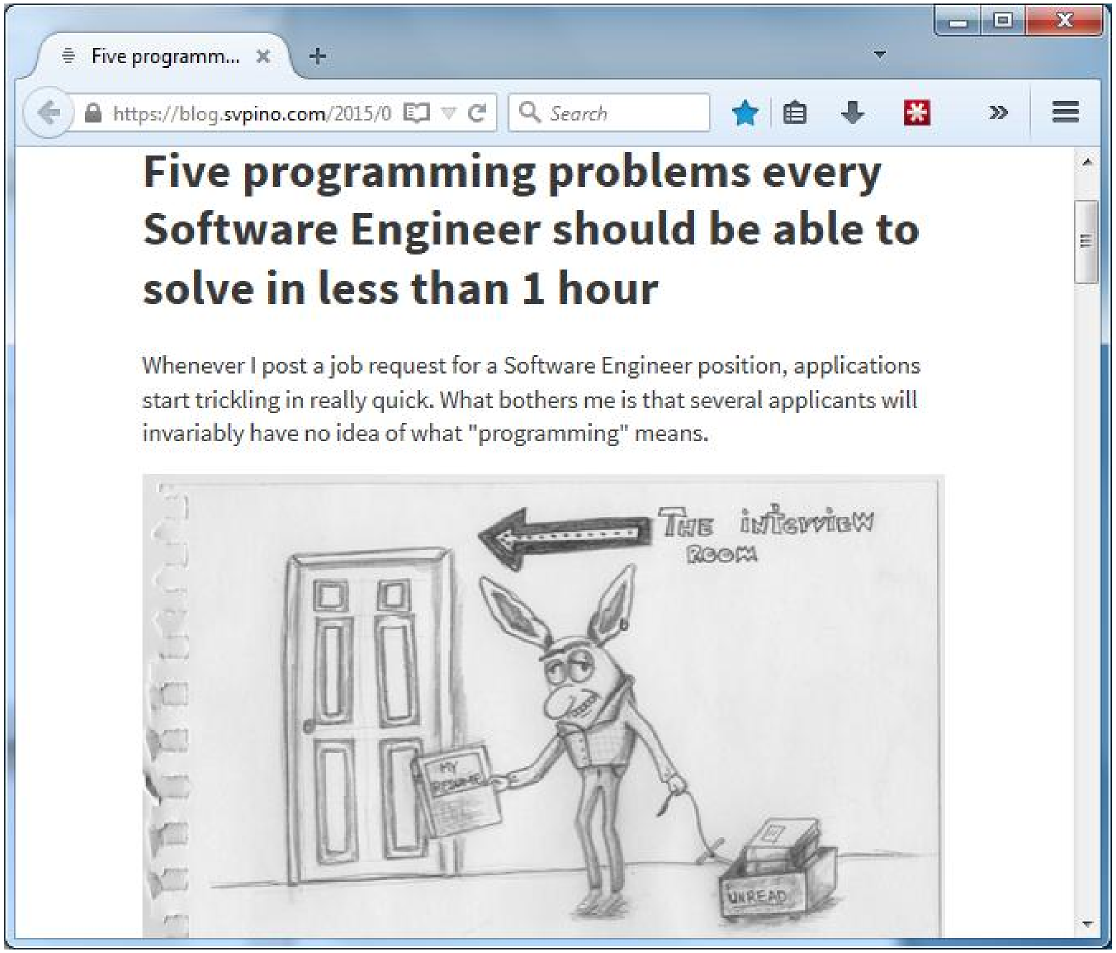
2 French privacy regulator demands Google make 'right to be forgotten' global (bbc.com)
submitted 16 minutes ago by AluminiumMk1 to /r/privacy
comment share save hide report [l+c]

4 Dug up an old flagstone pathway in rural Wales and discovered this carving on the underside of one of the stones. Can anyone decipher it? (imgur.com)
submitted an hour ago by timothyphoto to /r/whatisthisthing
7 comments share save hide report [l+c]

5 The Nokia smartwatch that could have been, but Microsoft killed (theverge.com)
submitted 3 minutes ago by RealtechPostBot to /r/realtech
1 comment share save hide report [l+c]

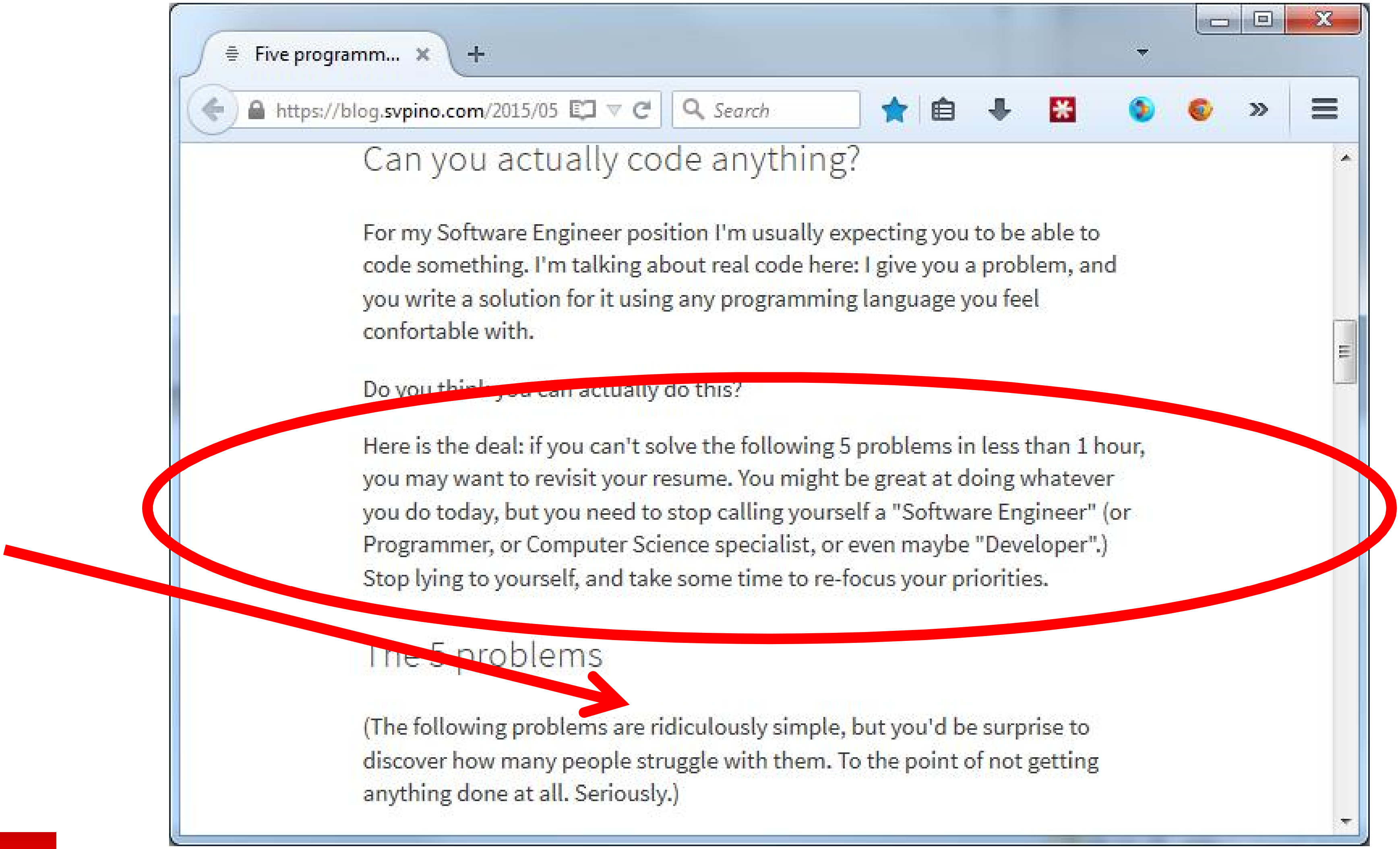
6 Facebook And Oculus Unveiled The First Commercial Version Of Virtual Reality Goggles Called Oculus Rift

...because you hate freedom.
...for your favorite hobby.



Five programming problems every Software Engineer should be able to solve in less than 1 hour

Whenever I post a job request for a Software Engineer position, applications start trickling in really quick. What bothers me is that several applicants will invariably have no idea of what "programming" means.



Can you actually code anything?

For my Software Engineer position I'm usually expecting you to be able to code something. I'm talking about real code here: I give you a problem, and you write a solution for it using any programming language you feel confortable with.

Do you think you can actually do this?

Here is the deal: if you can't solve the following 5 problems in less than 1 hour, you may want to revisit your resume. You might be great at doing whatever you do today, but you need to stop calling yourself a "Software Engineer" (or Programmer, or Computer Science specialist, or even maybe "Developer".) Stop lying to yourself, and take some time to re-focus your priorities.

The 5 problems

(The following problems are ridiculously simple, but you'd be surprise to discover how many people struggle with them. To the point of not getting anything done at all. Seriously.)

Problem 1

Write three functions that compute the sum of the numbers in a given list using a for-loop, a while-loop, and recursion.

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select column_value val
  5    from table( sys.odciNumberList(1,2,3,3) )
  6  )
  7  select sum(val)
  8    from list
  9  /

  SUM(VAL)
----------
         9
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rownum-1 r, column_value val
  5    from table( sys.odciNumberList(1,2,3,3) )
  6  )
  7  select summed
  8    from list
  9   model return updated rows
 10      dimension by (r)
 11       measures (val, 0 as summed )
 12          rules iterate (4294967295)
 13                until val[ITERATION_NUMBER] is null
 14                ( summed[0] = summed[0]+nvl(val[ITERATION_NUMBER],0))
 15  /

    SUMMED
----------
         9
```

```
ops$tkyte%ORA11GR2> with
  2   list as
  3   (
  4   select rownum-1 r, column_value val
  5     from table( sys.odciNumberList(1,2,3,3) )
  6   )
  7   select summed
  8     from list
  9    model return updated rows
 10         dimension by (r)
 11          measures (val, 0 as summed )
 12             rules iterate (4294967295)
 13                   until val[ITERATION_NUMBER] is null
 14                   ( summed[0] = summed[0]+nvl(val[ITERATION_NUMBER],0))
 15   /

    SUMMED
----------
         9
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rownum-1 r, column_value val
  5    from table( sys.odciNumberList(1,2,3,3) )
  6  )
  7  select summed
  8    from list
  9   model return updated rows
 10      dimension by (r)
 11       measures (val, 0 as summed )
 12          rules iterate (4294967295)
 13                until val[ITERATION_NUMBER] is null
 14                ( summed[0] = summed[0]+nvl(val[ITERATION_NUMBER],0))
 15  /

    SUMMED
----------
         9
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rownum-1 r, column_value val
  5    from table( sys.odciNumberList(1,2,3,3) )
  6  )
  7  select summed
  8    from list
  9   model return updated rows
 10       dimension by (r)
 11        measures (val, 0 as summed )
 12            rules iterate (4294967295)
 13                  until val[ITERATION_NUMBER] is null
 14                  ( summed[0] = summed[0]+nvl(val[ITERATION_NUMBER],0))
 15  /

    SUMMED
----------
         9
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rownum-1 r, column_value val
  5    from table( sys.odciNumberList(1,2,3,3) )
  6  )
  7  select summed
  8    from list
  9   model return updated rows
 10       dimension by (r)
 11        measures (val, 0 as summed )
 12           rules iterate (4294967295)
 13                 until val[ITERATION_NUMBER] is null
 14                 ( summed[0] = summed[0]+nvl(val[ITERATION_NUMBER],0))
 15  /

    SUMMED
----------
         9
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rownum-1 r, column_value val
  5    from table( sys.odciNumberList(1,2,3,3) )
  6  ),
  7  summation( r, val, summed ) as
  8  (
  9  select r, val, val summed
 10    from list
 11   where r=0
 12  union all
 13  select list.r, list.val, list.val+summation.summed
 14    from list,summation
 15   where list.r=summation.r+1
 16  )
 17  select summed
 18    from summation
 19   where r = (select count(*)-1 from list)
 20  /

    SUMMED
----------
         9
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rownum-1 r, column_value val
  5    from table( sys.odciNumberList(1,2,3,3) )
  6  ),
  7  summation( r, val, summed ) as
  8  (
  9  select r, val, val summed
 10    from list
 11   where r=0
 12  union all
 13  select list.r, list.val, list.val+summation.summed
 14    from list,summation
 15   where list.r=summation.r+1
 16  )
 17  select summed
 18    from summation
 19   where r = (select count(*)-1 from list)
 20  /

    SUMMED
----------
         9
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rownum-1 r, column_value val
  5    from table( sys.odciNumberList(1,2,3,3) )
  6  ),
  7  summation( r, val, summed ) as
  8  (
  9  select r, val, val summed
 10    from list
 11   where r=0
 12  union all
 13  select list.r, list.val, list.val+summation.summed
 14    from list,summation
 15   where list.r=summation.r+1
 16  )
 17  select summed
 18    from summation
 19   where r = (select count(*)-1 from list)
 20  /

    SUMMED
----------
         9
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rownum-1 r, column_value val
  5    from table( sys.odciNumberList(1,2,3,3) )
  6  ),
  7  summation( r, val, summed ) as
  8  (
  9  select r, val, val summed
 10    from list
 11   where r=0
 12  union all
 13  select list.r, list.val, list.val+summation.summed
 14    from list,summation
 15   where list.r=summation.r+1
 16  )
 17  select summed
 18    from summation
 19   where r = (select count(*)-1 from list)
 20  /

    SUMMED
----------
         9
```

Problem 2

Write a function that combines two lists by alternatingly taking elements. For example: given the two lists [1,2,3] and [a,b,c], the function should return [1, a, 2, b, 3, c].

```
ops$tkyte%ORA11GR2> with
  2  list1 as
  3  (
  4  select rownum*2-1 r, column_value val
  5    from table( sys.odciVarchar2List('1','2','3') )
  6  ),
  7  list2 as
  8  (
  9  select rownum*2 r, column_value val
 10    from table( sys.odciVarchar2List('a','b','c') )
 11  ),
 12  all_of_them as
 13  (
 14  select r, val
 15    from list1
 16   union all
 17  select r, val
 18    from list2
 19  )
 20  select cast( collect( val order by r ) as sys.odciVarchar2List ) new_list
 21    from all_of_them
 22  /

NEW_LIST
--------------------------------------------------
ODCIVARCHAR2LIST('1', 'a', '2', 'b', '3', 'c')
```

```
ops$tkyte%ORA11GR2> with
  2  list1 as
  3  (
  4  select rownum*2-1 r, column_value val     -- generates R=(1,3,5,7,9,….)
  5    from table( sys.odciVarchar2List('1','2','3') )
  6  ),
  7  list2 as
  8  (
  9  select rownum*2 r, column_value val
 10    from table( sys.odciVarchar2List('a','b','c') )
 11  ),
 12  all_of_them as
 13  (
 14  select r, val
 15    from list1
 16   union all
 17  select r, val
 18    from list2
 19  )
 20  select cast( collect( val order by r ) as sys.odciVarchar2List ) new_list
 21    from all_of_them
 22  /

NEW_LIST
--------------------------------------------------
ODCIVARCHAR2LIST('1', 'a', '2', 'b', '3', 'c')
```

```
ops$tkyte%ORA11GR2> with
  2  list1 as
  3  (
  4  select rownum*2-1 r, column_value val
  5    from table( sys.odciVarchar2List('1','2','3') )
  6  ),
  7  list2 as
  8  (
  9  select rownum*2 r, column_value val     -- generates R=(2,4,,6,8,…)
 10    from table( sys.odciVarchar2List('a','b','c') )
 11  ),
 12  all_of_them as
 13  (
 14  select r, val
 15    from list1
 16   union all
 17  select r, val
 18    from list2
 19  )
 20  select cast( collect( val order by r ) as sys.odciVarchar2List ) new_list
 21    from all_of_them
 22  /

NEW_LIST
--------------------------------------------------
ODCIVARCHAR2LIST('1', 'a', '2', 'b', '3', 'c')
```

```
ops$tkyte%ORA11GR2> with
  2  list1 as
  3  (
  4  select rownum*2-1 r, column_value val
  5    from table( sys.odciVarchar2List('1','2','3') )
  6  ),
  7  list2 as
  8  (
  9  select rownum*2 r, column_value val
 10    from table( sys.odciVarchar2List('a','b','c') )
 11  ),
 12  all_of_them as
 13  (
 14  select r, val
 15    from list1
 16   union all
 17  select r, val
 18    from list2
 19  )
 20  select cast( collect( val order by r ) as sys.odciVarchar2List ) new_list
 21    from all_of_them
 22  /

NEW_LIST
--------------------------------------------------
ODCIVARCHAR2LIST('1', 'a', '2', 'b', '3', 'c')
```

```
ops$tkyte%ORA11GR2> with
  2  list1 as
  3  (
  4  select rownum*2-1 r, column_value val
  5    from table( sys.odciVarchar2List('1','2','3') )
  6  ),
  7  list2 as
  8  (
  9  select rownum*2 r, column_value val
 10    from table( sys.odciVarchar2List('a','b','c') )
 11  ),
 12  all_of_them as
 13  (
 14  select r, val
 15    from list1
 16   union all
 17  select r, val
 18    from list2
 19  )
 20  select cast( collect( val order by r ) as sys.odciVarchar2List ) new_list
 21    from all_of_them
 22  /

NEW_LIST
--------------------------------------------------
ODCIVARCHAR2LIST('1', 'a', '2', 'b', '3', 'c')
```

Problem 3

Write a function that computes the list of the first 100 Fibonacci numbers. By definition, the first two numbers in the Fibonacci sequence are 0 and 1, and each subsequent number is the sum of the previous two. As an example, here are the first 10 Fibonacci numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, and 34.

```
ops$tkyte%ORA11GR2> select to_char( fib, '999,999,999,999,999,999,999,999' )
  2     from dual
  3    model return updated rows
  4       dimension by (0 as d)
  5        measures (0 as fib )
  6           rules iterate (100)
  7                  ( fib[iteration_number] =
                          case when iteration_number < 2 then iteration_number
  8                            else fib[cv()-2]+fib[cv()-1]
  9                         end )
 10   /

TO_CHAR(FIB,'999,999,999,999,999
--------------------------------
                               0
                               1
                               1
                               2
                               3
...
      83,621,143,489,848,422,977
     135,301,852,344,706,746,049
     218,922,995,834,555,169,026

100 rows selected.
```

(Two Hundred Eighteen quintillion … bigger than a "long long" C type or java long type which is only 9 quintillion. Don't even think about what PHP might do with this!)

```
ops$tkyte%ORA11GR2> select to_char( fib, '999,999,999,999,999,999,999,999' )
  2     from dual
  3   model return updated rows
  4       dimension by (0 as d)
  5        measures (0 as fib )
  6           rules iterate (100)
  7                 ( fib[iteration_number] =
                        case when iteration_number < 2 then iteration_number
  8                          else fib[cv()-2]+fib[cv()-1]
  9                      end )
 10  /

TO_CHAR(FIB,'999,999,999,999,999
--------------------------------
                               0
                               1
                               1
                               2
                               3
...
      83,621,143,489,848,422,977
     135,301,852,344,706,746,049
     218,922,995,834,555,169,026

100 rows selected.
```

```
ops$tkyte%ORA11GR2> select to_char( fib, '999,999,999,999,999,999,999,999' )
  2      from dual
  3     model return updated rows
  4        dimension by (0 as d)
  5         measures (0 as fib )
  6            rules iterate (100)
  7                  ( fib[iteration_number] =
                         case when iteration_number < 2 then iteration_number
  8                           else fib[cv()-2]+fib[cv()-1]
  9                          end )
 10   /

TO_CHAR(FIB,'999,999,999,999,999
--------------------------------
                               0
                               1
                               1
                               2
                               3
...
      83,621,143,489,848,422,977
     135,301,852,344,706,746,049
     218,922,995,834,555,169,026

100 rows selected.
```

```
ops$tkyte%ORA11GR2> select to_char( fib, '999,999,999,999,999,999,999,999' )
  2     from dual
  3    model return updated rows
  4       dimension by (0 as d)
  5        measures (0 as fib )
  6           rules iterate (100)
  7                ( fib[iteration_number] =
                        case when iteration_number < 2 then iteration_number
  8                          else fib[cv()-2]+fib[cv()-1]
  9                         end )
 10   /

TO_CHAR(FIB,'999,999,999,999,999
--------------------------------
                               0
                               1
                               1
                               2
                               3
...
      83,621,143,489,848,422,977
     135,301,852,344,706,746,049
     218,922,995,834,555,169,026

100 rows selected.
```

```
ops$tkyte%ORA11GR2> select to_char( fib, '999,999,999,999,999,999,999,999' )
  2     from dual
  3    model return updated rows
  4       dimension by (0 as d)
  5        measures (0 as fib )
  6           rules iterate (100)
  7                   ( fib[iteration_number] =
                           case when iteration_number < 2 then iteration_number
  8                             else fib[cv()-2]+fib[cv()-1]
  9                         end )
 10  /

TO_CHAR(FIB,'999,999,999,999,999
--------------------------------
                               0
                               1
                               1
                               2
                               3
...
      83,621,143,489,848,422,977
     135,301,852,344,706,746,049
     218,922,995,834,555,169,026

100 rows selected.
```

```
ops$tkyte%ORA11GR2> select to_char( fib, '999,999,999,999,999,999,999,999' )
  2     from dual
  3    model return updated rows
  4       dimension by (0 as d)
  5        measures (0 as fib )
  6           rules iterate (100)
  7                 ( fib[iteration_number] =
                         case when iteration_number < 2 then iteration_number
  8                           else fib[cv()-2]+fib[cv()-1]
  9                          end )
 10   /

TO_CHAR(FIB,'999,999,999,999,999
--------------------------------
                               0
                               1
                               1
                               2
                               3
...
      83,621,143,489,848,422,977
     135,301,852,344,706,746,049
     218,922,995,834,555,169,026

100 rows selected.
```

```
ops$tkyte%ORA11GR2> select to_char( fib, '999,999,999,999,999,999,999,999' )
  2     from dual
  3    model return updated rows
  4       dimension by (0 as d)
  5        measures (0 as fib )
  6           rules iterate (100)
  7                 ( fib[iteration_number] =
                      case when iteration_number < 2 then iteration_number
  8                        else fib[cv()-2]+fib[cv()-1]
  9                       end )
 10   /

TO_CHAR(FIB,'999,999,999,999,999
--------------------------------
                               0
                               1
                               1
                               2
                               3
...
      83,621,143,489,848,422,977
     135,301,852,344,706,746,049
     218,922,995,834,555,169,026

100 rows selected.
```

Problem 4

Write a function that given a list of non negative integers, arranges them such that they form the largest possible number. For example, given [50, 2, 1, 9], the largest formed number is 95021.

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rpad('x',rownum, 'x') || column_value val
  5    from table( sys.odciNumberList(50,2,9,1) )
  6  ),
  7  unique_combos as
  8  (
  9   select sys_connect_by_path( val, ',' ) scbp
 10     from list
 11    where level = (select count(*) from list)
 12  connect by nocycle
                level <= (select count(*) from list)
         and val <> prior val
 13  )
 14  select max( to_number( replace( replace( scbp, 'x', '' ),
                                                        ',', '' ) ) ) qed
 15    from unique_combos
 16  /
       QED
----------
     95021
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rpad('x',rownum, 'x') || column_value val
  5    from table( sys.odciNumberList(50,2,9,1) )
  6  ),
  7  unique_combos as
  8  (
  9   select sys_connect_by_path( val, ',' ) scbp
 10     from list
 11    where level = (select count(*) from list)
 12  connect by nocycle
                level <= (select count(*) from list)
            and val <> prior val
 13  )
 14  select max( to_number( replace( replace( scbp, 'x', '' ),
                                                      ',', '' ) ) ) qed
 15    from unique_combos
 16  /
       QED
----------
     95021
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rpad('x',rownum, 'x') || column_value val
  5    from table( sys.odciNumberList(50,2,9,1) )
  6  ),
  7  unique_combos as
  8  (
  9   select sys_connect_by_path( val, ',' ) scbp
 10     from list
 11    where level = (select count(*) from list)
 12  connect by nocycle
                level <= (select count(*) from list)
            and val <> prior val
 13  )
 14  select max( to_number( replace( replace( scbp, 'x', '' ),
                                                          ',', '' ) ) ) qed
 15    from unique_combos
 16  /
       QED
----------
     95021
```

```
SCBP
--------------------------------------------------
,x50,xx2,xxx9,xxxx1
,x50,xx2,xxxx1,xxx9
,x50,xxx9,xx2,xxxx1
,x50,xxx9,xxxx1,xx2
,x50,xxxx1,xx2,xxx9
,x50,xxxx1,xxx9,xx2
,xx2,x50,xxx9,xxxx1
,xx2,x50,xxxx1,xxx9
,xx2,xxx9,x50,xxxx1
,xx2,xxx9,xxxx1,x50
,xx2,xxxx1,x50,xxx9
,xx2,xxxx1,xxx9,x50
,xxx9,x50,xx2,xxxx1
,xxx9,x50,xxxx1,xx2
,xxx9,xx2,x50,xxxx1
,xxx9,xx2,xxxx1,x50
,xxx9,xxxx1,x50,xx2
,xxx9,xxxx1,xx2,x50
,xxxx1,x50,xx2,xxx9
,xxxx1,x50,xxx9,xx2
,xxxx1,xx2,x50,xxx9
,xxxx1,xx2,xxx9,x50
,xxxx1,xxx9,x50,xx2
,xxxx1,xxx9,xx2,x50
24 rows selected.
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rpad('x',rownum, 'x') || column_value val
  5    from table( sys.odciNumberList(50,2,9,1) )
  6  ),
  7  unique_combos as
  8  (
  9   select sys_connect_by_path( val, ',' ) scbp
 10     from list
 11    where level = (select count(*) from list)
 12  connect by nocycle
             level <= (select count(*) from list)
         and val <> prior val
 13  )
 14  select max( to_number( replace( replace( scbp, 'x', '' ),
                                                  ',', '' ) ) ) qed
 15    from unique_combos
 16  /
       QED
----------
     95021
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rpad('x',rownum, 'x') || column_value val
  5    from table( sys.odciNumberList(50,2,9,1) )
  6  ),
  7  unique_combos as
  8  (
  9   select sys_connect_by_path( val, ',' ) scbp
 10     from list
 11    where level = (select count(*) from list)
 12  connect by nocycle
               level <= (select count(*) from list)
           and val <> prior val
 13  )
 14  select max( to_number( replace( replace( scbp, 'x', '' ),
                                                 ',', '' ) ) ) qed
 15    from unique_combos
 16  /
       QED
----------
     95021
```

```
ops$tkyte%ORA11GR2> with
  2  list as
  3  (
  4  select rpad('x',rownum, 'x') || column_value val
  5    from table( sys.odciNumberList(50,2,9,1) )
  6  ),
  7  unique_combos as
  8  (
  9   select sys_connect_by_path( val, ',' ) scbp
 10     from list
 11    where level = (select count(*) from list)
 12  connect by nocycle
                level <= (select count(*) from list)
            and val <> prior val
 13  )
 14  select max( to_number( replace( replace( scbp, 'x', '' ),
                                                    ',', '' ) ) ) qed
 15    from unique_combos
 16  /
       QED
----------
     95021
```

Problem 5

Write a program that outputs all possibilities to put + or - or nothing between the numbers 1, 2, ..., 9 (in this order) such that the result is always 100. For example: 1 + 2 + 34 – 5 + 67 – 8 + 9 = 100.

```
ops$tkyte%ORA11GR2> with
ops as
(
select column_value v
  from table( sys.odciVarchar2List( '+', '-' ) )
),
ops_strings as
(
select o1.v||o2.v||o3.v||o4.v||o5.v||o6.v||o7.v||o8.v||o9.v os
  from ops o1, ops o2, ops o3, ops o4, ops o5, ops o6, ops o7, ops o8, ops o9
 where o1.v = '+'
),
digits as
(
select column_value d
  from table( sys.odciVarchar2List('1','2','3','4','5','6','7','8','9') )
),
nums as
(
select d, replace(sys_connect_by_path(d,'|'),'|','') n
  from digits
connect by prior d = d-1
),
nums2 as
(
select *
  from (
select rownum grp, sys_connect_by_path( n, ' ' )||' ' n2
  from nums
connect by substr(prior n,length(prior n)) = substr(n,1,1)-1
       )
 where n2 like ' 1%' and n2 like '%9 '
),
nums3 as
(
select distinct grp, substr( os, 1, length(n2)-length(replace(n2,' ',''))-1 ) os, n2
  from nums2, ops_strings
),
grps as
(
select grp, os, to_number( substr( column_value, 1, 1 ) ) r, to_number( substr( column_value, 2 ) ) x
  from nums3, table( cast( multiset( select
                                       to_char(rownum,'fm0') ||
                                       trim( substr (n2,
                                             instr (n2, ' ', 1, level  ) + 1,
                                             instr (n2, ' ', 1, level+1)
                                                - instr (n2, ' ', 1, level) -1 ) )
                                     from dual
                                  connect by level <=
                                     length(n2)-length( replace( n2, ' ', '' ) ) - 1)
                         as sys.odciVarchar2List ) )
)
select listagg( substr( os, r, 1 ) || x ) within group ( order by r ) la
  from grps
 group by grp, os
having sum( x * decode( substr( os, r, 1 ), '+', +1, -1 ) ) = 100
/
```

```
ops$tkyte%ORA11GR2> with
  2   ops as
  3   (
  4   select column_value v
  5     from table( sys.odciVarchar2List( '+', '-' ) )
  6   ),
  7   ops_strings as
  8   (
  9   select o1.v||o2.v||o3.v||o4.v||o5.v||o6.v||o7.v||o8.v||o9.v os
 10     from ops o1, ops o2, ops o3, ops o4, ops o5, ops o6, ops o7, ops o
 11    where o1.v = '+'
 12   ), …

OS
-------------------------
+++++++++
+-+++++++
++-++++++
...
+++------
+-+------
++-------
+--------
256 rows selected.
```

```
ops$tkyte%ORA11GR2> with
…
 13  digits as
 14  (
 15  select column_value d
 16    from table(sys.odciVarchar2List('1','2','3','4','5','6',…'))
 17  ),
 18  nums as
 19  (
 20  select d, replace(sys_connect_by_path(d,'|'),'|','') n
 21    from digits
 22  connect by prior d = d-1
 23  ),
…
D N              7 1234567        7 234567         8 345678
- ----------     8 12345678       8 2345678        9 3456789
1 1              9 123456789      9 23456789       4 4
2 12             2 2              3 3              …
3 123            3 23             4 34
4 1234           4 234            5 345
5 12345          5 2345           6 3456
6 123456         6 23456          7 34567
```

```
ops$tkyte%ORA11GR2> with
…
 24  nums2 as
 25  (
 26  select *
 27    from (
 28  select rownum grp, sys_connect_by_path( n, ' ' )||' ' n2
 29    from nums
 30  connect by substr(prior n,length(prior n)) = substr(n,1,1)-1
 31         )
 32   where n2 like ' 1%' and n2 like '%9 '
 33  )
…
 GRP N2
---------- -------------------------
         9  1 2 3 4 5 6 7 8 9
        10  1 2 3 4 5 6 7 89
        12  1 2 3 4 5 6 78 9
        ...
       508  1234567 89
       510  12345678 9
       511  123456789

256 rows selected.
```

```
ops$tkyte%ORA11GR2> with
…
 34   nums3 as
 35   (
 36   select distinct grp, substr( os, 1, length(n2)-
                                length(replace(n2,' ',''))-1 ) os, n2
 37     from nums2, ops_strings
 38   ),
 …

 GRP OS                          N2
---------- ------------------------- -------------------------
         9 +++++++--                  1 2 3 4 5 6 7 8 9
         9 ++++-++++                  1 2 3 4 5 6 7 8 9
        12 +++++--+                   1 2 3 4 5 6 78 9
. . .
       423 +---+                      123 45 6 7 89
       458 +---                       1234 5 6 789

6561 rows selected.
```

```
ops$tkyte%ORA11GR2> with
…
 39  grps as
 40  (
 41  select grp, os, to_number( substr( column_value, 1, 1 ) ) r,
            to_number( substr( column_value, 2 ) ) x
 42    from nums3,
         table( cast( multiset( select
 43                  to_char(rownum,'fm0') ||
 44                  trim( substr (n2,
 45                          instr (n2, ' ', 1, level  ) + 1,
 46                          instr (n2, ' ', 1, level+1)
 47                            - instr (n2, ' ', 1, level) -1 ) )
 48                   from dual
 49                connect by level <=
 50                   length(n2)-length( replace( n2, ' ', '' ) ) - 1)
 51                   as sys.odciVarchar2List ) )
 52  )…
```

```
       GRP OS         N2                 R          X
---------- ---------- ----------------- -- ----------
         9 +++++++--  1 2 3 4 5 6 7 8 9  1          1
         9 +++++++--  1 2 3 4 5 6 7 8 9  2          2
         9 +++++++--  1 2 3 4 5 6 7 8 9  3          3
         9 +++++++--  1 2 3 4 5 6 7 8 9  4          4
         9 +++++++--  1 2 3 4 5 6 7 8 9  5          5
         9 +++++++--  1 2 3 4 5 6 7 8 9  6          6
         9 +++++++--  1 2 3 4 5 6 7 8 9  7          7
         9 +++++++--  1 2 3 4 5 6 7 8 9  8          8
         9 +++++++--  1 2 3 4 5 6 7 8 9  9          9
...
        55 ++--+++    1 2 3 456 7 8 9    1          1
        55 ++--+++    1 2 3 456 7 8 9    2          2
        55 ++--+++    1 2 3 456 7 8 9    3          3
        55 ++--+++    1 2 3 456 7 8 9    4        456
        55 ++--+++    1 2 3 456 7 8 9    5          7
        55 ++--+++    1 2 3 456 7 8 9    6          8
        55 ++--+++    1 2 3 456 7 8 9    7          9
...
```

```
ops$tkyte%ORA11GR2> with
 53  select listagg( substr( os, r, 1 ) || x )
                                    within group ( order by r ) la
 54    from grps
 55   group by grp, os
 56  having sum( x * decode( substr( os, r, 1 ), '+', +1, -1 ) ) = 100
 57  /

LA
------------------------------------
+1+2+3-4+5+6+78+9
+1+2+34-5+67-8+9
+1+23-4+5+6+78-9
+1+23-4+56+7+8+9
+12+3+4+5-6-7+89
+12-3-4+5-6+7+89
+12+3-4+5+67+8+9
+123-4-5-6-7+8-9
+123+4-5+67-89
+123+45-67+8-9
+123-45-67+89
11 rows selected.
```

Here is *another* set of interview questions…

What is the “Best Way”

It Depends!!!

What is the best way to name all of the numbers from 1-100 which have the letter ‘A’ in their spellings?

What is the “Best Way”

```
ops$tkyte%ORA11GR2> select *
  2    from ( select to_char( to_date( rownum, 'j' ), 'jsp' ) num
  3                from dual
  4             connect by level <= 100  )
  5  /

NUM
------------------------------------------------------------------------------
one
two
three ...
```

What is the “Best Way”

```
ops$tkyte%ORA11GR2> select *
  2    from ( select to_char( to_date( rownum, 'j' ), 'jsp' ) num
  3                 from dual
  4              connect by level <= 100  )
  5   where num like '%a%'
  6  /
no rows selected
```

What is the “Best Way”

Now what are all of the characters in A-Z that do not appear in the spellings of the first 100 numbers?

```
ops$tkyte%ORA11GR2> select num, column_value ch
  2    from ( select to_char( to_date( rownum, 'j' ), 'jsp' ) num
  3                from dual
  4             connect by level <= 100  ),
  5           table( cast( multiset( select substr( num, level, 1 )
ch
  8                                     from dual
  9                                  connect by level <= length(num)
 10                          ) as sys.odciVarchar2List ) )
 13   where column_value between 'a' and 'z'
 14  /

NUM                  CH
-------------------- --
one                  o
one                  n
one                  e
two                  t
two                  w
…
```

```
ops$tkyte%ORA11GR2> select distinct column_value ch
  2    from ( select to_char( to_date( rownum, 'j' ), 'jsp' ) num
  3              from dual
  4           connect by level <= 100  ),
  5          table( cast( multiset( select substr( num, level, 1 ) ch
  8                                   from dual
  9                                connect by level <= length(num)
 10                      ) as sys.odciVarchar2List ) )
 13   where column_value between 'a' and 'z'
 14  /
CH
--
e
h
i
o
s
d
r …
```

```
ops$tkyte%ORA11GR2> select listagg(ch,'') within group (order by ch)
  2    from (
  3  select distinct column_value ch
  4    from ( select to_char( to_date( rownum, 'j' ), 'jsp' ) num
  5                  from dual
  6              connect by level <= 100  ),
  7             table(cast(multiset( select substr( num, level, 1 ) ch
 10                                     from dual
 11                                  connect by level <= length(num)
 12                           ) as sys.odciVarchar2List ))
 15   where column_value between 'a' and 'z' )
 17  /

LISTAGG(CH,'')WITHINGROUP(ORDERBYCH)
----------------------------------------------
defghilnorstuvwxy
```

```
ops$tkyte%ORA11GR2> select replace(
  2           translate( 'abcdefghijklmnopqrstuvwxyz',
  3                       listagg(ch,'') within group (order by ch),
  4                       '@' ), '@', '')
  6    from (
  7  select distinct column_value ch
  8    from ( select to_char( to_date( rownum, 'j' ), 'jsp' ) num
  9             from dual
 10          connect by level <= 100  ),
 11         table(cast(multiset( select substr( num, level, 1 ) ch
 14                                from dual
 15                             connect by level <= length(num)
 16                    ) as sys.odciVarchar2List ) )
 19   where column_value between 'a' and 'z' )
 21  /
--------------------------
abcjkmpqz
```

Here is *my* set of interview questions…

Using SCOTT.EMP show

- The number of records in this table
- The number of records in this table by the month of hiredate (and why is this question ambiguous?)
- Display the month with the most records (the most popular month). As a bonus to this question - please explain why the question itself is not very well formed.

Number of records…

```
ops$tkyte%ORA11GR2> select count(*) from scott.emp;

  COUNT(*)
----------
        14
```

| |

Number of records by month…

```
SQL > select to_char(hiredate,'mm'), count(*)
  2      from scott.emp
  3     group by to_char(hiredate,'mm')
  4*    order by to_char(hiredate,'mm');
```

```
TO   COUNT(*)
-- ----------
01          2
02          2
04          1
05          1
06          1
09          2
11          1
12          4

8 rows selected.     ???????????????? Is that right, aren’t there 12 months in a year??
```

Number of records by month…

```
ops$tkyte%ORA11GR2> with months as
  2   (select to_char(level,'fm00') mm
  3      from dual
  4   connect by level <= 12)
  5   select months.mm, count(e.empno) cnt
  6     from scott.emp e right outer join months on
  7           (to_char(e.hiredate,'mm')=months.mm)
  8    group by months.mm
  9    order by months.mm;
```

Number of records by month…

```
MM         CNT
--- ----------
01           1
02           2
03           0
04           2
05           2
06           1
07           0
08           0
09           2
10           0
11           1
12           3

12 rows selected.
```

Most Frequently Occurring Month

- Assumes there is a *single* most frequently occurring month!
- They all might be.

Most Frequently Occurring Month

```
ops$tkyte%ORA11GR2> select to_char(hiredate,'mm'), count(*)
  2    from scott.emp
  3   group by to_char(hiredate,'mm')
  4  having count(*) = (select max(count(*))
  5                       from scott.emp
  6                      group by to_char(hiredate,'mm'))
  7  /

TO   COUNT(*)
-- ----------
12          4
```

Most Frequently Occurring Month

```
ops$tkyte%ORA11GR2> select *
  2    from (
  3  select to_char(hiredate,'mm') mm, count(*),
  4         rank() over (order by count(*) DESC) rnk
  5    from scott.emp
  6   group by to_char(hiredate,'mm')
  7         )
  8   where rnk = 1
  9   order by mm
 10  /

MM   COUNT(*)        RNK
-- ---------- ----------
12          4          1
```

Most Frequently Occurring Month

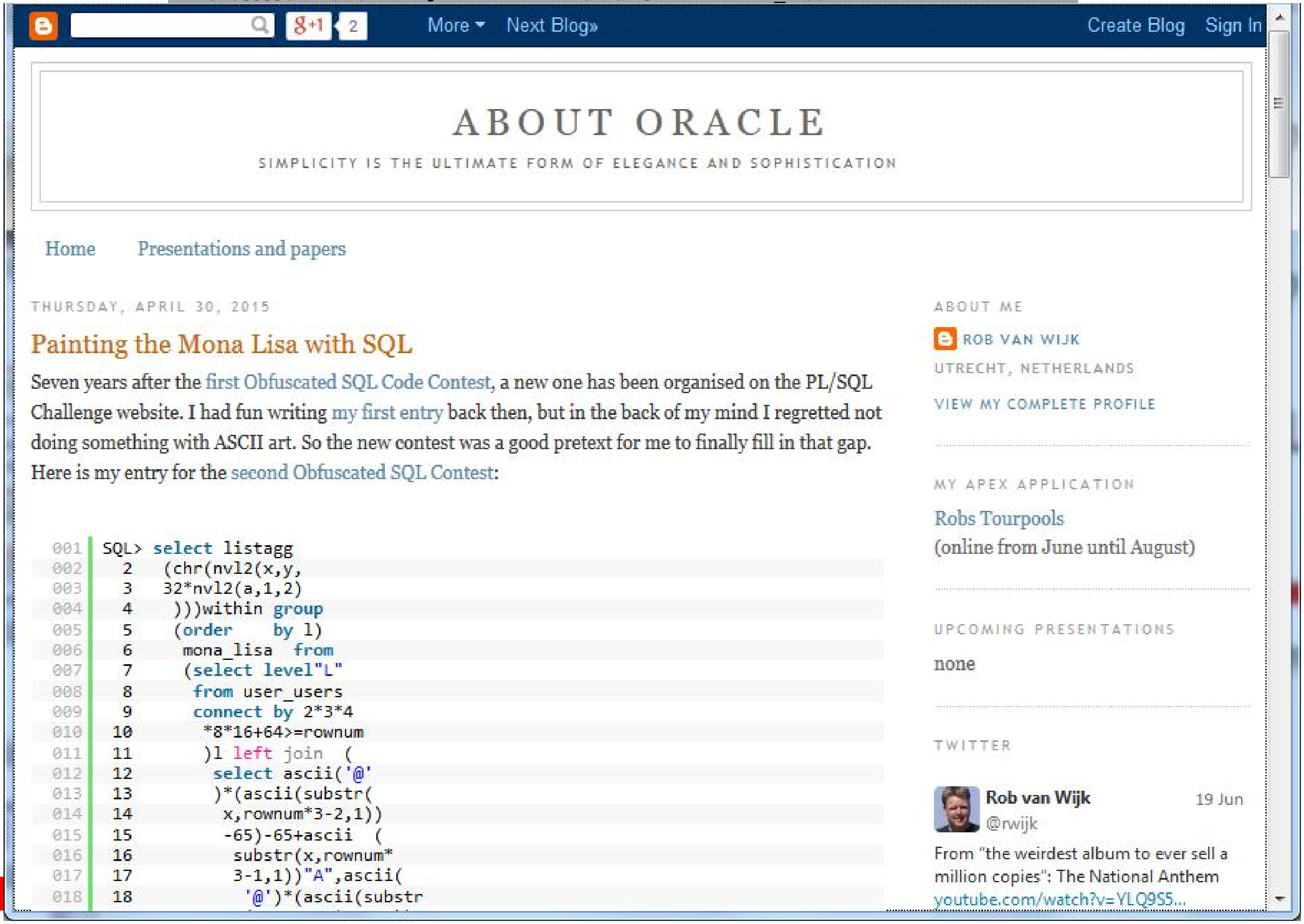
```
ops$tkyte%ORA11GR2> select *
  2    from (
  3  select to_char(hiredate,'mm') mm, count(*) cnt,
  4         max(count(*)) over () mx
  5    from scott.emp
  6   group by to_char(hiredate,'mm')
  7         )
  8   where mx = cnt
  9   order by mm
 10  /

MM        CNT         MX
-- ---------- ----------
12          4          4
```

Most Frequently Occurring Month

```
ops$tkyte%ORA12CR1> select to_char(hiredate,'mm'), count(*) cnt
  2     from scott.emp
  3    group by to_char(hiredate,'mm')
  4    order by cnt DESC
  5    FETCH FIRST 1 ROWS WITH TIES
  6   /

TO         CNT
-- ----------
12           4
```

```
ops$tkyte%ORA11GR2> select listagg
  2     (chr(nvl2(x,y,
  3     32*nvl2(a,1,2)
  4      )))within group
  5      (order    by l)
  6       mona_lisa  from
  7       (select level"L"
  8        from user_users
  9        connect by 2*3*4
 10         *8*16+64>=rownum
 11         )l left join  (
 12          select ascii('@'
 13          )*(ascii(substr(
 14           x,rownum*3-2,1))
 15           -65)-65+ascii  (
 16            substr(x,rownum*
 17            3-1,1))"A",ascii(
 18             '@')*(ascii(substr
 19             (x,rownum*3-2,1))-
 20              65)  -131+ascii(
 21              substr(x,rownum*3
 22              -1,1))+ascii(substr
 23               (x,rownum*3,1))"B"
 24               from(select'HUKI'||
 25               'RLJRPKOULOUMNWN'||
 26                'NVONUPNVQBDQNYR'||
 27                'BDRMZSNOTUGUVEV'||
 28                 'WDW]EXVEXgEYLOY'||
 29                 ']NZLNZ\N[MN[^K\'||
 30                  'MO\]L]NM]^J^OG^'||
 31                  'VD^^I_OF_^H`QF`'||
 32                   '^EdSEeTDfUHgUHh'||
 33                   'WDaSF'x from dual
 34                    where 1=1/*voor@
 35                     patch72*/group by
 36                       cube('m','o',
 37                       'n','a','l',
 38                     'i','s','a') having
 39                  grouping_id('m','o','n','a',
 40             'l','i','s','a')between 11 and 58))x on
 41             (l between a and b) left join (with x(v)
 42              as(select 1 from dual union   all select
 43              x.v+1 from dual join x on(28>=x.v))search
 44               depth first by v set a select(16+v+decode
 45               (sign(round((v-6)/5)),-1,0,0,1,4))*64"X",
 46                  91"Y"from x union all select p,x from (
 47                   select x,y,sum(nvl2(nullif(z,1),z-1,1))
 48                    over (order by i) p from (select i,x,y
 49                     ,nvl(nullif(dense_rank()over(order by
 50                      decode(y,'louvre',x)   nulls first
 51                        ),28),29)z from(select i,x,case
 52                         when x in(34,35,36,37,38,48,
 53                          49,51,54,57,58,59,60,62,63
 54                           ,64,66,67,69,70,72,73,79,
 55                             81,82,83,85)then'louvre'
 56                               end y from(select i,
 57                                 158-ascii(regexp_substr
 58                                   (x,'((.)?)',i,1))x from
 59                                      user_users model return      updated
 60                                        rows dimension by(1 i) measures
 61                                          (rpad('leonardo da vinci',7
 62      *6*5*4*2-1)x)(x[for i from 1 to 1679 increment 1]='5>5*u("CCCCvoo'||
 63      '+su>vqBBoTT7<::m:1SSv;}voqClao;;*o}q(B}?RBu"Aq}ov}quBRBu(";iv}r}'||
 64      'R::Y:SoA};BBB("5Bo;JB(vv;qA}>o(Boovq"*BH(;R}1xQX<R*BB"u;olq((uo5'||
 65      '*BpqqoqrBrB(Br"uo(o1nSO<S}(}qv~uq}pCo;J~oopBqqwqq"qo;v9mZZK+BBBB'||
 66      'q""o(};Bpqrq"B>""BqBo5:kPI<pur>qrqo;q>5}rB}q}q}w}q}:mNCs tt]]]O<'||
 67      'opo>;qopBgqqqwpqq~qor5:ztN}>eBwDfV5pBBwpBp;~wr>r>BwqrqT{i>qdqq~>'||
 68      'q5E]e<h?Bq"qB;q~wp~qpr>opW{Sq~>_q~ugSQY;qqqqqo~q~>q~~qq5{]5Xq}DD'||
 69      'Y<}~B>"q>~>qrwq~WzCrXwpZfSYCoqrqopq>~pqo(z]uUquD:Q:[Bw~?B~q~>~r5'||
 70      'HTz}Vwq}vSiS[Cvo+Cp~~q~5BWzDqXw~}~q5iiDD\uvju>~r~q"<y}BVw~~>~"q*'||
 71      'EW^FPEDqz**y]qrMw~~>uv:^Zuf]yufxBLrqwqoS:\SZ{~~AADy:"q`r55p;rr~q'||
 72      'p5;RDSSnf|Shv|f|~~SPfxv~p5}(WW?nTjS<fDS<|<<S|:|ffm<e:S|q~uofSzS|'||
 73      '<|:N   JjvyT:{g>???+|QyQQf:z:cRQf|B}jf|Syf{:tz<H5z<]SRE  nt88oBP'||
 74      'Qf_:<{rBBS<z:|p *>~}t }>p~~qQ]uyw wpup>~>rwS\]S<|r>fQn~q>rr7(D>>'||
 75      'y]ooq~~pp;BsB>y5A:nQdf|5B|f|SzZ`B}wb}5Dfzf]xf:|<zfQ|9f:iYw}qbwqu'||
 76      'f[Qf5f:Df|Qyqb(~~rerB*Sc]nDl"|<|]Sz<|C`ood;;*<]z]S{:{:zSz<15{f|Q'||
 77      '|fzNqcqaopho5Zf:`]z:yfzQ]yBqnpz?u7l5kqvv:SSx:y:{SnSy<]S{S{u5xw;r'||
 78      'rS<{<SmuFuS<|Q{:zQ<{]|:nfzS|<y]|]]o5qx QyQ;yprB;auZf:{<z:ef:{]xQ'||
 79      '<nvw;B>xNNSvvr5A("qBq(ufFP]:QSm<{QS|]kf{QS|:|]|R}qq;u+?r~rvD+<R<'||
 80      '|B>q~pqPAoSQm:z]k:yS<zS|<zoq~~}ww  H88tPi8~q>~qrBuuSSmQS{:zQf:|<'||
 81      'x<zS]Snlqqyrr?TTo5uoq~"o(uPS|f:efkQ|Szf:|Sz<x:}rzwHHBuBBuBgvqujT'||
 82      '<|f|]zS|:mfhQyf{S|S<z:|(pm>qBBo(uff<|Q|]|S{Qz]{Q|f{]|:|f|Q|D<|:|'||
 83      'Q|Se1mqu}o+*<<|<{]|<xS<m<zSyS{:x:|Q|fxS+zrqo(TZ{f:|SnfeQSz<{Q{*P'||
 84      '|<{:|:|QeRRjTS:|]{:{SyQSyfd<nuPQ|f<{]z]`]{]{f{S|:yQ{S{S|]zf{Qxv*'||
 85      'QQz:{Qfy]y]z]h]yf<{f:^Q|*PD{Sz:y]xfzojP{f|<{:{Q|f|QS|Qf|Sn:hQoS]'||
 86      'fzQ]zQeC"*tD|Pyf{QfD]DD|Qz]|:xSz:{<ETjQn]cZpBwvESSDZ|f:S|i]fjED|'||
 87      ':Qc:zSi:Qf|f:|:kfnppq}ouuDSi]Sju]EvoFEoDz]z:zQxEf:P]{]S{:z<hRrqr'||
 88      'B}AABFviuDog;BB*ioSk<mfQfz]z]{fhSZR*}~}rq"*wvqBB}rBorB}DTWz:|Q:zW'
 89      )))))where y is null)y on(l.l=y.x)group by ceil(l.l/(2*ascii(' ')))
 90  /d
```

```
MONA_LISA
---------------------------------------------
i`it)v|[[[[(//s+)`(-\\/JJgbdd@@@@@@@dmKK(c!(/-[2=/cct/!-v\!_L\)|
```



ABOUT ORACLE

SIMPLICITY IS THE ULTIMATE FORM OF ELEGANCE AND SOPHISTICATION

Home Presentations and papers

THURSDAY, APRIL 30, 2015

Painting the Mona Lisa with SQL

Seven years after the first Obfuscated SQL Code Contest, a new one has been organised on the PL/SQL Challenge website. I had fun writing my first entry back then, but in the back of my mind I regretted not doing something with ASCII art. So the new contest was a good pretext for me to finally fill in that gap. Here is my entry for the second Obfuscated SQL Contest:

```
SQL> select listagg
  2   (chr(nvl2(x,y,
  3   32*nvl2(a,1,2)
  4    )))within group
  5    (order    by l)
  6     mona_lisa  from
  7     (select level"L"
  8      from user_users
  9      connect by 2*3*4
 10       *8*16+64>=rownum
 11       )l left join  (
 12        select ascii('@'
 13        )*(ascii(substr(
 14         x,rownum*3-2,1))
 15         -65)-65+ascii  (
 16          substr(x,rownum*
 17          3-1,1))"A",ascii(
 18           '@')*(ascii(substr
```

ABOUT ME

ROB VAN WIJK

UTRECHT, NETHERLANDS

VIEW MY COMPLETE PROFILE

MY APEX APPLICATION

Robs Tourpools

(online from June until August)

UPCOMING PRESENTATIONS

none

TWITTER

Rob van Wijk 19 Jun
@rwijk

From "the weirdest album to ever sell a million copies": The National Anthem youtube.com/watch?v=YLQ9S5...

```
49 rows selected.
```

NoSQL is here to replace SQL right?

What was SQL invented to replace?

Before SQL there was No SQL, SQL was invented to replace NoSQL

SQL is Critical

"….the complexity of dealing with a non-ACID data store in every part of our business logic would be too great, and there was simply no way our business could function without SQL queries."

Google, VLDB 2013

https://www.linkedin.com/groups/Find-out-why-Google-decided-4434815.S.273792742

"[Facebook] started in the Hadoop world. We are now bringing in relational to enhance that. ... [we] realized that using the wrong technology for certain kinds of problems can be difficult."

Ken Rudin, Facebook, TDWI 2013

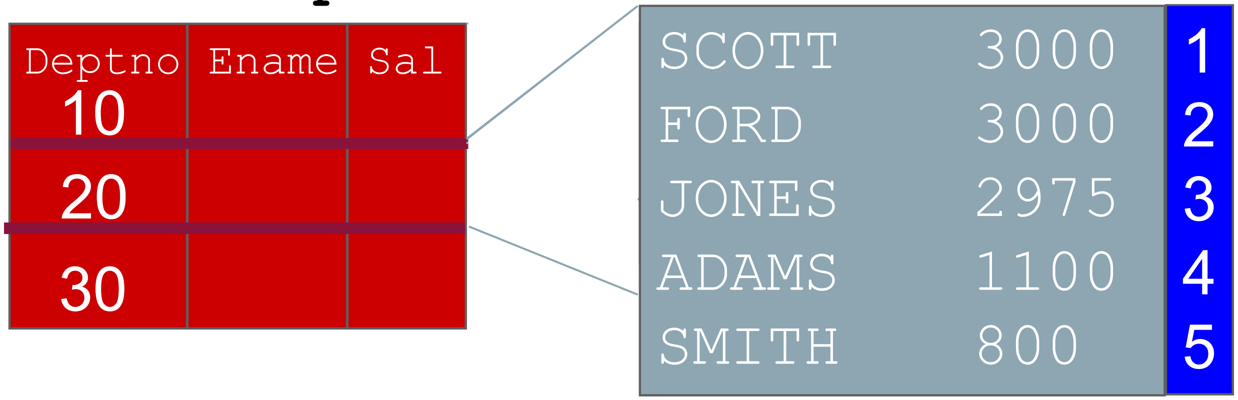
http://tdwi.org/articles/2013/05/06/facebooks-relational-platform.aspx

ORACLE

Analytics

Analytics
Ordered Array Semantics in SQL queries

```
Select deptno,ename,sal
  Row_number() over (partition by deptno
                     Order by sal desc )
   from emp
```



Deptno	Ename	Sal
10		
20		
30		

SCOTT	3000	1
FORD	3000	2
JONES	2975	3
ADAMS	1100	4
SMITH	800	5

Why Analytics

- A running total
- Percentages within a group
- Top-N queries
- Moving Averages
- Ranking Queries
- Medians
- And the list is infinitely long
  - "*Analytics are the coolest thing to happen to SQL since the keyword Select*"

ORACLE

Find the average amount of time between patient visits

```
TKYTE@ORA12C> create table patient_visits
  2  as
  3  select distinct
  4         object_type||
  5            trunc(row_number() over
  6                        (partition by object_type
  7                             order by created) / 250) patient_id,
  8         created+rownum visit_date
  9    from all_objects;

Table created.

TKYTE@ORA12C> alter table patient_visits
  2  add constraint
  3  patient_visits_pk
  4  primary key(patient_id,visit_date);

Table altered.
```

```
TKYTE@ORA12C> select avg(visit_date-last_visit_date)
  2    from (
  3  select t1.visit_date, max(t2.visit_date) last_visit_date
  4    from patient_visits t1, patient_visits t2
  5   where t1.patient_id = t2.patient_id
  6     and t2.visit_date < t1.visit_date
  7   group by t1.patient_id, t1.visit_date
  8         )
  9  /

AVG(VISIT_DATE-LAST_VISIT_DATE)
-------------------------------
                     23.5868092
```

ORACLE

```
call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.00       0.00          0         16          0           0
Execute      1      0.00       0.00          0          0          0           0
Fetch        2      5.69       5.78          0        598          0           1
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total        4      5.70       5.79          0        614          0           1


Rows (1st) Row Source Operation
---------- ---------------------------------------------------
         1 SORT AGGREGATE (cr=598 pr=0 pw=0 time=5782851 us)
     89224  VIEW  (cr=598 pr=0 pw=0 time=6920605 us cost=37226 size=1612980 card
     89224   HASH GROUP BY (cr=598 pr=0 pw=0 time=6021974 us cost=37226 size=340
  11045765    HASH JOIN  (cr=598 pr=0 pw=0 time=3936745 us cost=488 size=3952522
     89610     TABLE ACCESS FULL PATIENT_VISITS (cr=299 pr=0 pw=0 time=11514 us
     89610     TABLE ACCESS FULL PATIENT_VISITS (cr=299 pr=0 pw=0 time=38901 us
TKYTE@ORA12C> select avg( visit_date-last_visit_date )
  2    from (
  3  select visit_date,
  4         (select max(visit_date)
  5            from patient_visits t2
  6           where t2.patient_id = t1.patient_id
  7             and t2.visit_date < t1.visit_date) last_visit_date
  8     from patient_visits t1
  9          )
 10  /

AVG(VISIT_DATE-LAST_VISIT_DATE)
-------------------------------
                     23.5868092
```

```
call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.00       0.00          0          0          0           0
Execute      1      0.00       0.00          0          0          0           0
Fetch        2      6.85       6.90        360      30919          0           1
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total        4      6.85       6.91        360      30919          0           1


Rows (1st) Row Source Operation
---------- ---------------------------------------------------
     89610 SORT AGGREGATE (cr=30620 pr=360 pw=0 time=5432914 us)
     89224  FIRST ROW  (cr=30620 pr=360 pw=0 time=3154562 us cost=2 size=19 card
     89224   INDEX RANGE SCAN (MIN/MAX) PATIENT_VISITS_PK (cr=30620 pr=360 pw=0
         1 SORT AGGREGATE (cr=30919 pr=360 pw=0 time=6908877 us)
     89610  TABLE ACCESS FULL PATIENT_VISITS (cr=299 pr=0 pw=0 time=403891 us co
TKYTE@ORA12C> select avg(visit_date-last_visit_date)
  2    from (
  3  select visit_date,
  4         lag(visit_date) over
  5               (partition by patient_id
  6                    order by visit_date)
  7         as last_visit_date
  8    from patient_visits
  9         )
 10  /

AVG(VISIT_DATE-LAST_VISIT_DATE)
-------------------------------
                     23.5868092
```

```
call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.00       0.00          0          0          0           0
Execute      1      0.00       0.00          0          0          0           0
Fetch        2      0.19       0.19          0        361          0           1
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total        4      0.19       0.20          0        361          0           1

Rows (1st) Row Source Operation
---------- ---------------------------------------------------
         1 SORT AGGREGATE (cr=361 pr=0 pw=0 time=197264 us)
     89610  VIEW  (cr=361 pr=0 pw=0 time=2751836 us cost=362 size=1612980 card=8
     89610   WINDOW BUFFER (cr=361 pr=0 pw=0 time=771339 us cost=362 size=170259
     89610    INDEX FULL SCAN PATIENT_VISITS_PK (cr=361 pr=0 pw=0 time=640345 us
```

Model

Model Clause

- A feature of the database since 10g (2003!)
- Spreadsheet like construct
- Procedural processing in a non-procedural language

I need to have running totals that group rows into groups such that the total for that group does not exceed some threshold

Model Clause

Threshold = 65,000

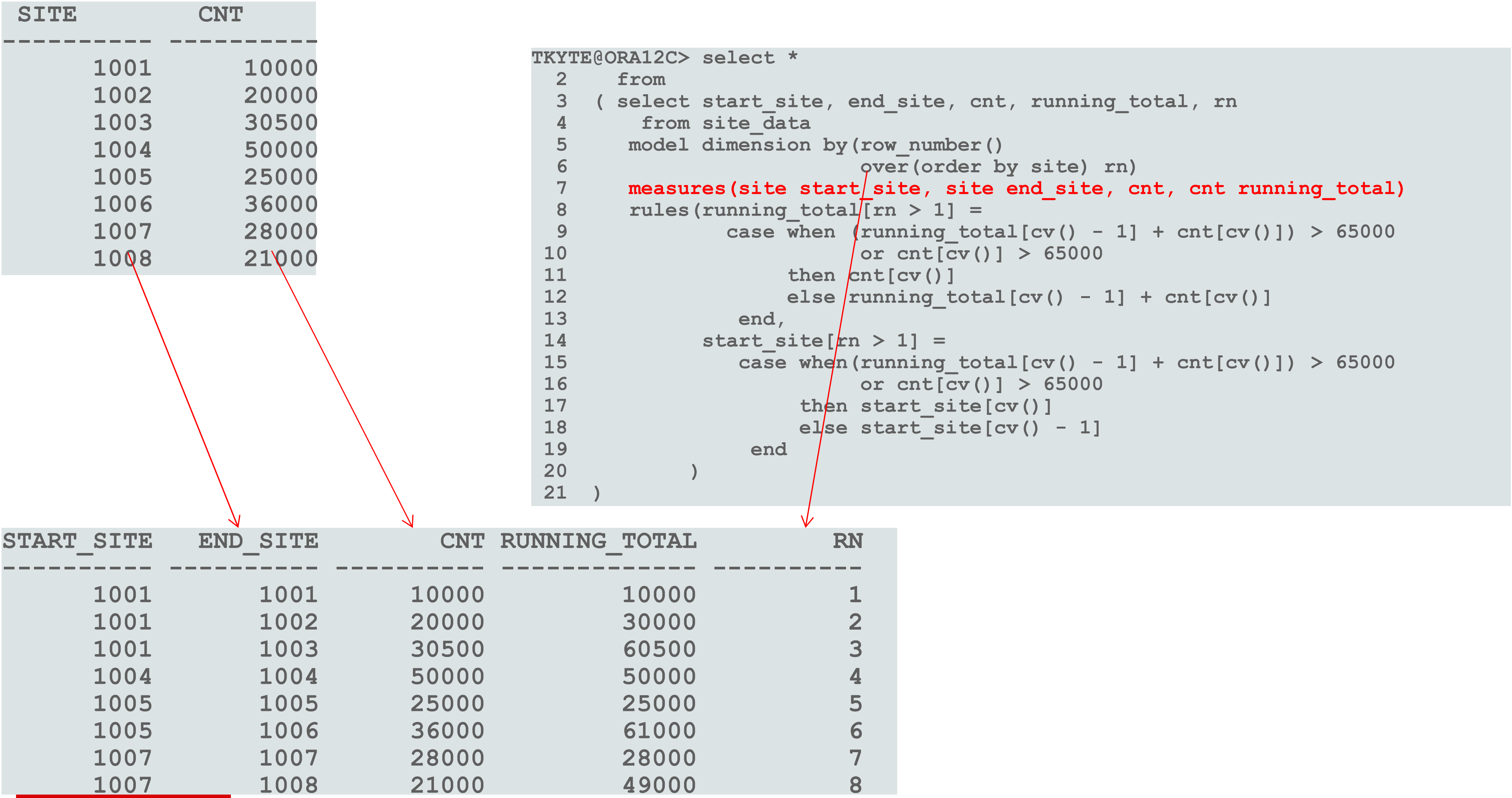
```
Site    cnt
-----   --------
1001       10000
1002       20000
1003       30500        60,500
1004       50000        50,000
1005       25000
1006       36000        61,000
1007       28000
1008       21000        49,000
```

```
St_key    end_key   total
------    -------   -----
1001      1003      60500
1004      1004      50000
1005      1006      61000
1007      1008      49000
```

```
TKYTE@ORA12C> select start_site, max(end_site), max(running_total)
  2    from
  3  (
  4  select *
  5    from
  6  ( select start_site, end_site, cnt, running_total, rn
  7      from site_data
  8     model dimension by(row_number()
  9                        over(order by site) rn)
 10     measures(site start_site, site end_site, cnt, cnt running_total)
 11     rules(running_total[rn > 1] =
 12                     case when (running_total[cv() - 1] + cnt[cv()]) > 65000
 13                                or cnt[cv()] > 65000
 14                          then cnt[cv()]
 15                          else running_total[cv() - 1] + cnt[cv()]
 16                      end,
 17                   start_site[rn > 1] =
 18                     case when(running_total[cv() - 1] + cnt[cv()]) > 65000
 19                               or cnt[cv()] > 65000
 20                          then start_site[cv()]
 21                          else start_site[cv() - 1]
 22                      end
 23             )
 24  )
 25  )
 26  group by start_site
 27  order by start_site
 28  /

START_SITE MAX(END_SITE) MAX(RUNNING_TOTAL)
---------- ------------- ------------------
      1001          1003              60500
      1004          1004              50000
      1005          1006              61000
      1007          1008              49000
 SITE          CNT
---------- ----------
      1001      10000
      1002      20000
      1003      30500
      1004      50000
      1005      25000
      1006      36000
      1007      28000
      1008      21000
```

```
TKYTE@ORA12C> select *
  2      from
  3    ( select start_site, end_site, cnt, running_total, rn
  4        from site_data
  5       model dimension by(row_number()
  6                          over(order by site)  rn)
  7       measures(site start_site, site end_site, cnt, cnt running_total)
  8       rules(running_total[rn > 1] =
  9               case when (running_total[cv() - 1] + cnt[cv()]) > 65000
 10                          or cnt[cv()] > 65000
 11                    then cnt[cv()]
 12                    else running_total[cv() - 1] + cnt[cv()]
 13                end,
 14             start_site[rn > 1] =
 15                case when(running_total[cv() - 1] + cnt[cv()]) > 65000
 16                          or cnt[cv()] > 65000
 17                     then start_site[cv()]
 18                     else start_site[cv() - 1]
 19                 end
 20            )
 21    )
```

```
START_SITE   END_SITE        CNT RUNNING_TOTAL         RN
---------- ---------- ---------- ------------- ----------
      1001       1001      10000         10000          1
      1001       1002      20000         30000          2
      1001       1003      30500         60500          3
      1004       1004      50000         50000          4
      1005       1005      25000         25000          5
      1005       1006      36000         61000          6
      1007       1007      28000         28000          7
      1007       1008      21000         49000          8
```



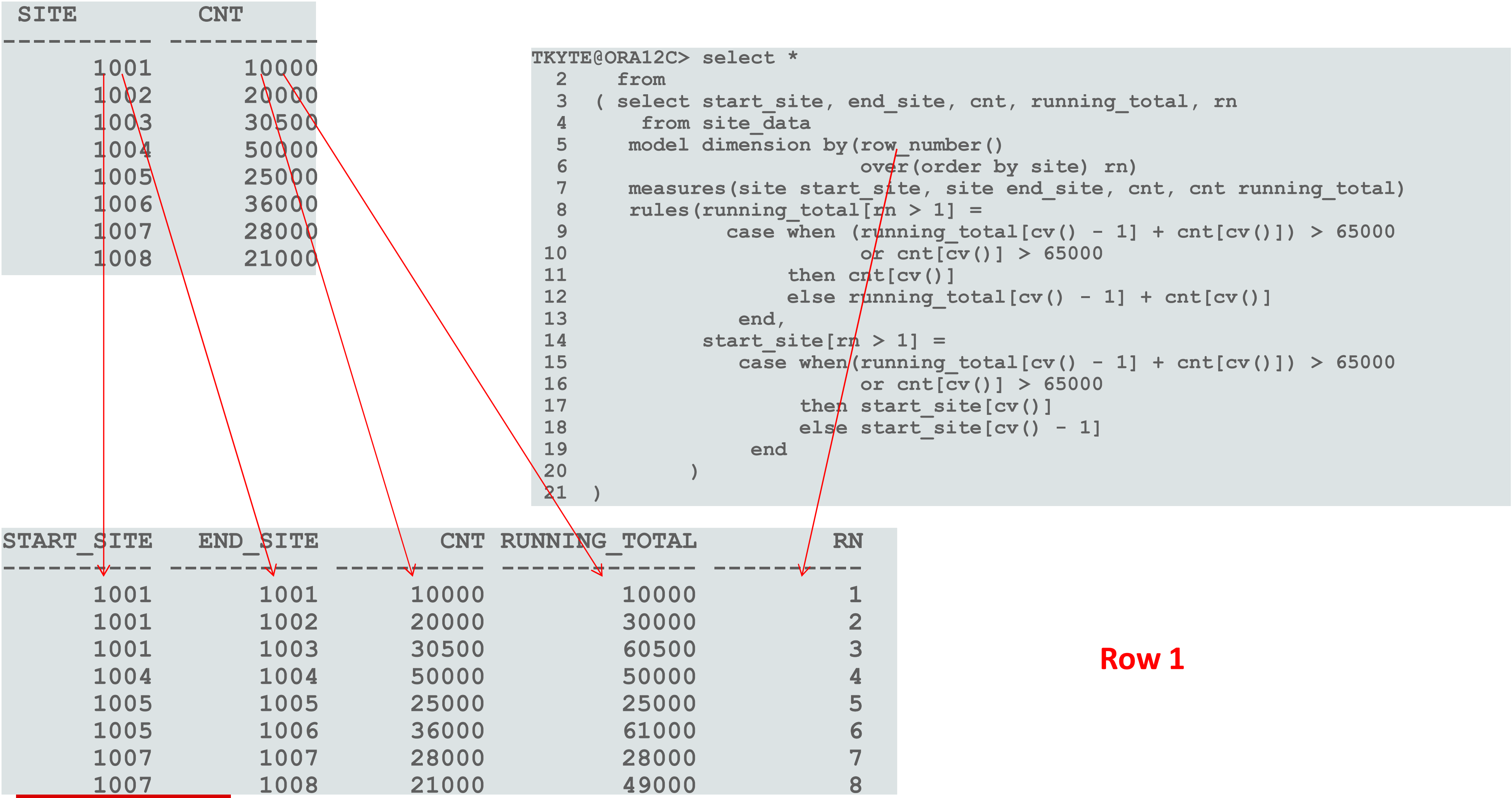
```
 SITE         CNT
---------- ----------
      1001      10000
      1002      20000
      1003      30500
      1004      50000
      1005      25000
      1006      36000
      1007      28000
      1008      21000
```

```
TKYTE@ORA12C> select *
  2    from
  3  ( select start_site, end_site, cnt, running_total, rn
  4      from site_data
  5     model dimension by(row_number()
  6                        over(order by site) rn)
  7     measures(site start_site, site end_site, cnt, cnt running_total)
  8     rules(running_total[rn > 1] =
  9             case when (running_total[cv() - 1] + cnt[cv()]) > 65000
 10                        or cnt[cv()] > 65000
 11                  then cnt[cv()]
 12                  else running_total[cv() - 1] + cnt[cv()]
 13              end,
 14           start_site[rn > 1] =
 15              case when(running_total[cv() - 1] + cnt[cv()]) > 65000
 16                        or cnt[cv()] > 65000
 17                   then start_site[cv()]
 18                   else start_site[cv() - 1]
 19               end
 20          )
 21  )
```

```
START_SITE   END_SITE        CNT RUNNING_TOTAL         RN
---------- ---------- ---------- ------------- ----------
      1001       1001      10000         10000          1
      1001       1002      20000         30000          2
      1001       1003      30500         60500          3
      1004       1004      50000         50000          4
      1005       1005      25000         25000          5
      1005       1006      36000         61000          6
      1007       1007      28000         28000          7
      1007       1008      21000         49000          8
      SITE        CNT
---------- ----------
      1001      10000
      1002      20000
      1003      30500
      1004      50000
      1005      25000
      1006      36000
      1007      28000
      1008      21000
```

```
TKYTE@ORA12C> select *
       from
     ( select start_site, end_site, cnt, running_total, rn
         from site_data
        model dimension by(row_number()
                           over(order by site) rn)
        measures(site start_site, site end_site, cnt, cnt running_total)
        rules(running_total[rn > 1] =
                case when (running_total[cv() - 1] + cnt[cv()]) > 65000
                           or cnt[cv()] > 65000
                     then cnt[cv()]
                     else running_total[cv() - 1] + cnt[cv()]
                 end,
              start_site[rn > 1] =
                 case when(running_total[cv() - 1] + cnt[cv()]) > 65000
                           or cnt[cv()] > 65000
                      then start_site[cv()]
                      else start_site[cv() - 1]
                  end
             )
     )
```

```
START_SITE   END_SITE        CNT RUNNING_TOTAL         RN
---------- ---------- ---------- ------------- ----------
      1001       1001      10000         10000          1
      1001       1002      20000         30000          2
      1001       1003      30500         60500          3
      1004       1004      50000         50000          4
      1005       1005      25000         25000          5
      1005       1006      36000         61000          6
      1007       1007      28000         28000          7
      1007       1008      21000         49000          8
```



```
 SITE             CNT
---------- ----------
      1001      10000
      1002      20000
      1003      30500
      1004      50000
      1005      25000
      1006      36000
      1007      28000
      1008      21000
```

```
TKYTE@ORA12C> select *
  2    from
  3  ( select start_site, end_site, cnt, running_total, rn
  4      from site_data
  5     model dimension by(row_number()
  6                        over(order by site) rn)
  7     measures(site start_site, site end_site, cnt, cnt running_total)
  8     rules(running_total[rn > 1] =
  9             case when (running_total[cv() - 1] + cnt[cv()]) > 65000
 10                        or cnt[cv()] > 65000
 11                  then cnt[cv()]
 12                  else running_total[cv() - 1] + cnt[cv()]
 13               end,
 14            start_site[rn > 1] =
 15               case when(running_total[cv() - 1] + cnt[cv()]) > 65000
 16                        or cnt[cv()] > 65000
 17                   then start_site[cv()]
 18                   else start_site[cv() - 1]
 19                end
 20           )
 21  )
```

```
START_SITE   END_SITE        CNT RUNNING_TOTAL         RN
---------- ---------- ---------- ------------- ----------
      1001       1001      10000         10000          1
      1001       1002      20000         30000          2
      1001       1003      30500         60500          3
      1004       1004      50000         50000          4
      1005       1005      25000         25000          5
      1005       1006      36000         61000          6
      1007       1007      28000         28000          7
      1007       1008      21000         49000          8
```

Row 1

ORACLE

```
 SITE              CNT
---------- ----------
      1001      10000
      1002      20000
      1003      30500
      1004      50000
      1005      25000
      1006      36000
      1007      28000
      1008      21000
```

```
TKYTE@ORA12C> select *
       from
    ( select start_site, end_site, cnt, running_total, rn
        from site_data
       model dimension by(row_number()
                          over(order by site) rn)
       measures(site start_site, site end_site, cnt, cnt running_total)
       rules(running_total[rn > 1] =
               case when (running_total[cv() - 1] + cnt[cv()]) > 65000
                          or cnt[cv()] > 65000
                    then cnt[cv()]
                    else running_total[cv() - 1] + cnt[cv()]
                end,
             start_site[rn > 1] =
                case when(running_total[cv() - 1] + cnt[cv()]) > 65000
                          or cnt[cv()] > 65000
                     then start_site[cv()]
                     else start_site[cv() - 1]
                 end
             )
   )
```

```
START_SITE   END_SITE         CNT  RUNNING_TOTAL         RN
---------- ---------- ---------- ------------- ----------
      1001       1001      10000         10000          1
      1001       1002      20000         30000          2
      1001       1003      30500         60500          3
      1004       1004      50000         50000          4
      1005       1005      25000         25000          5
      1005       1006      36000         61000          6
      1007       1007      28000         28000          7
      1007       1008      21000         49000          8
```

Row 2

```
SITE              CNT
---------- ----------
      1001      10000
      1002      20000
      1003      30500
      1004      50000
      1005      25000
      1006      36000
      1007      28000
      1008      21000
```

```
TKYTE@ORA12C> select *
  2    from
  3   ( select start_site, end_site, cnt, running_total, rn
  4       from site_data
  5      model dimension by(row_number()
  6                         over(order by site) rn)
  7      measures(site start_site, site end_site, cnt, cnt running_total)
  8      rules(running_total[rn > 1] =
  9              case when (running_total[cv() - 1] + cnt[cv()]) > 65000
 10                        or cnt[cv()] > 65000
 11                   then cnt[cv()]
 12                   else running_total[cv() - 1] + cnt[cv()]
 13               end,
 14            start_site[rn > 1] =
 15               case when(running_total[cv() - 1] + cnt[cv()]) > 65000
 16                         or cnt[cv()] > 65000
 17                    then start_site[cv()]
 18                    else start_site[cv() - 1]
 19                end
 20           )
 21   )
```

```
START_SITE   END_SITE        CNT RUNNING_TOTAL         RN
---------- ---------- ---------- ------------- ----------
      1001       1001      10000         10000          1
      1001       1002      20000         30000          2
      1001       1003      30500         60500          3
      1004       1004      50000         50000          4
      1005       1005      25000         25000          5
      1005       1006      36000         61000          6
      1007       1007      28000         28000          7
      1007       1008      21000         49000          8
```

Row 3

```
 SITE         CNT
---------- ----------
      1001      10000
      1002      20000
      1003      30500
      1004      50000
      1005      25000
      1006      36000
      1007      28000
      1008      21000
```

```
TKYTE@ORA12C> select *
  2    from
  3  ( select start_site, end_site, cnt, running_total, rn
  4      from site_data
  5     model dimension by(row_number()
  6                    over(order by site) rn)
  7     measures(site start_site, site end_site, cnt, cnt running_total)
  8     rules(running_total[rn > 1] =
  9             case when (running_total[cv() - 1] + cnt[cv()]) > 65000
 10                        or cnt[cv()] > 65000
 11                  then cnt[cv()]
 12                  else running_total[cv() - 1] + cnt[cv()]
 13              end,
 14           start_site[rn > 1] =
 15              case when(running_total[cv() - 1] + cnt[cv()]) > 65000
 16                        or cnt[cv()] > 65000
 17                   then start_site[cv()]
 18                   else start_site[cv() - 1]
 19               end
 20          )
 21  )
```

```
START_SITE   END_SITE        CNT RUNNING_TOTAL         RN
---------- ---------- ---------- ------------- ----------
      1001       1001      10000         10000          1
      1001       1002      20000         30000          2
      1001       1003      30500         60500          3
      1004       1004      50000         50000          4
      1005       1005      25000         25000          5
      1005       1006      36000         61000          6
      1007       1007      28000         28000          7
      1007       1008      21000         49000          8
```

Row 4

```
SITE              CNT
---------- ----------
      1001      10000
      1002      20000
      1003      30500
      1004      50000
      1005      25000
      1006      36000
      1007      28000
      1008      21000
```

```
TKYTE@ORA12C> select *
  2    from
  3   ( select start_site, end_site, cnt, running_total, rn
  4       from site_data
  5      model dimension by(row_number()
  6                         over(order by site) rn)
  7      measures(site start_site, site end_site, cnt, cnt running_total)
  8      rules(running_total[rn > 1] =
  9              case when (running_total[cv() - 1] + cnt[cv()]) > 65000
 10                         or cnt[cv()] > 65000
 11                   then cnt[cv()]
 12                   else running_total[cv() - 1] + cnt[cv()]
 13               end,
 14            start_site[rn > 1] =
 15               case when(running_total[cv() - 1] + cnt[cv()]) > 65000
 16                         or cnt[cv()] > 65000
 17                    then start_site[cv()]
 18                    else start_site[cv() - 1]
 19                end
 20           )
 21   )
```

```
START_SITE   END_SITE        CNT RUNNING_TOTAL         RN
---------- ---------- ---------- ------------- ----------
      1001       1001      10000         10000          1
      1001       1002      20000         30000          2
      1001       1003      30500         60500          3
      1004       1004      50000         50000          4
      1005       1005      25000         25000          5
      1005       1006      36000         61000          6
      1007       1007      28000         28000          7
      1007       1008      21000         49000          8
```

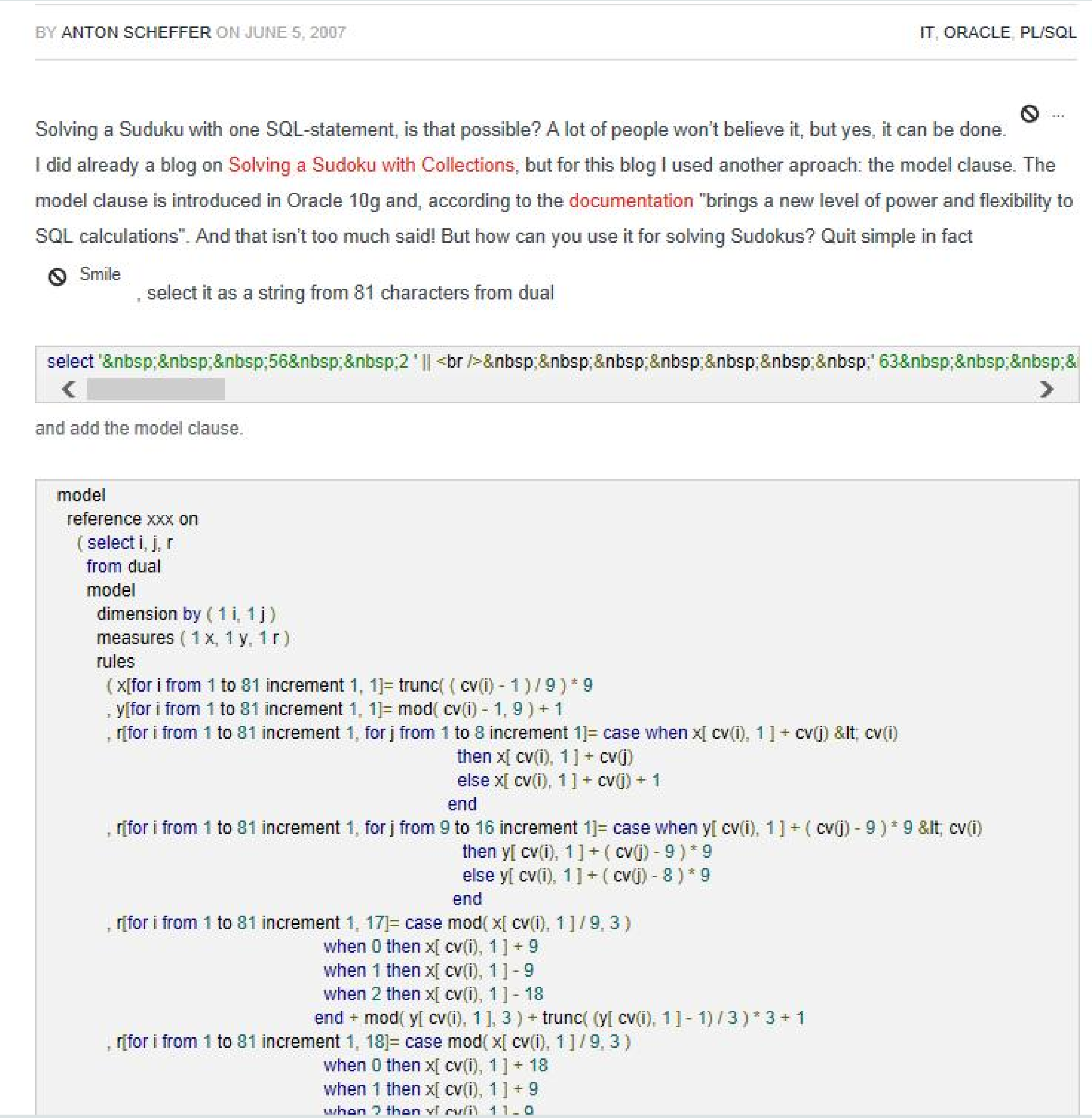
Row 5

```
 SITE        CNT
---------- ----------
      1001      10000
      1002      20000
      1003      30500
      1004      50000
      1005      25000
      1006      36000
      1007      28000
      1008      21000
```

```
TKYTE@ORA12C> select *
  2    from
  3   ( select start_site, end_site, cnt, running_total, rn
  4       from site_data
  5      model dimension by(row_number()
  6                         over(order by site) rn)
  7      measures(site start_site, site end_site, cnt, cnt running_total)
  8      rules(running_total[rn > 1] =
  9              case when (running_total[cv() - 1] + cnt[cv()]) > 65000
 10                         or cnt[cv()] > 65000
 11                   then cnt[cv()]
 12                   else running_total[cv() - 1] + cnt[cv()]
 13               end,
 14            start_site[rn > 1] =
 15               case when(running_total[cv() - 1] + cnt[cv()]) > 65000
 16                         or cnt[cv()] > 65000
 17                    then start_site[cv()]
 18                    else start_site[cv() - 1]
 19                end
 20           )
 21   )
```

```
START_SITE   END_SITE        CNT RUNNING_TOTAL         RN
---------- ---------- ---------- ------------- ----------
      1001       1001      10000         10000          1
      1001       1002      20000         30000          2
      1001       1003      30500         60500          3
      1004       1004      50000         50000          4
      1005       1005      25000         25000          5
      1005       1006      36000         61000          6
      1007       1007      28000         28000          7
      1007       1008      21000         49000          8
```

Row 6



BY ANTON SCHEFFER ON JUNE 5, 2007 IT, ORACLE, PL/SQL

Solving a Suduku with one SQL-statement, is that possible? A lot of people won't believe it, but yes, it can be done. ⃠ ...
I did already a blog on Solving a Sudoku with Collections, but for this blog I used another aproach: the model clause. The model clause is introduced in Oracle 10g and, according to the documentation "brings a new level of power and flexibility to SQL calculations". And that isn't too much said! But how can you use it for solving Sudokus? Quit simple in fact

⃠ Smile , select it as a string from 81 characters from dual

```
select '&nbsp;&nbsp;&nbsp;56&nbsp;&nbsp;2 ' || <br />&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;' 63&nbsp;&nbsp;&nbsp;&
```

and add the model clause.

```
model
 reference xxx on
  ( select i, j, r
   from dual
   model
    dimension by ( 1 i, 1 j )
    measures ( 1 x, 1 y, 1 r )
    rules
     ( x[for i from 1 to 81 increment 1, 1]= trunc( ( cv(i) - 1 ) / 9 ) * 9
     , y[for i from 1 to 81 increment 1, 1]= mod( cv(i) - 1, 9 ) + 1
     , r[for i from 1 to 81 increment 1, for j from 1 to 8 increment 1]= case when x[ cv(i), 1 ] + cv(j) &lt; cv(i)
                                          then x[ cv(i), 1 ] + cv(j)
                                          else x[ cv(i), 1 ] + cv(j) + 1
                                         end
     , r[for i from 1 to 81 increment 1, for j from 9 to 16 increment 1]= case when y[ cv(i), 1 ] + ( cv(j) - 9 ) * 9 &lt; cv(i)
                                           then y[ cv(i), 1 ] + ( cv(j) - 9 ) * 9
                                           else y[ cv(i), 1 ] + ( cv(j) - 8 ) * 9
                                          end
     , r[for i from 1 to 81 increment 1, 17]= case mod( x[ cv(i), 1 ] / 9, 3 )
                             when 0 then x[ cv(i), 1 ] + 9
                             when 1 then x[ cv(i), 1 ] - 9
                             when 2 then x[ cv(i), 1 ] - 18
                            end + mod( y[ cv(i), 1 ], 3 ) + trunc( (y[ cv(i), 1 ] - 1) / 3 ) * 3 + 1
     , r[for i from 1 to 81 increment 1, 18]= case mod( x[ cv(i), 1 ] / 9, 3 )
                             when 0 then x[ cv(i), 1 ] + 18
                             when 1 then x[ cv(i), 1 ] + 9
                             when 2 then x[ cv(i), 1 ] - 9
```

BY ANTON SCHEFFER ON OCTOBER 13, 2009 DATABASE, IT, ORACLE, PL/SQL

Oracle Database 11g Release 2 introduces a new feature called Recursive Subquery Factoring. My collegue Lucas sees it as a substitute for Connect By based hierarchical querying, Oracle RDBMS 11gR2 – new style hierarchical querying using Recursive Subquery Factoring. When I first was thinking about a pratical use for this feature I couldn't come up with anything, but on second thought:: solving Sudokus!

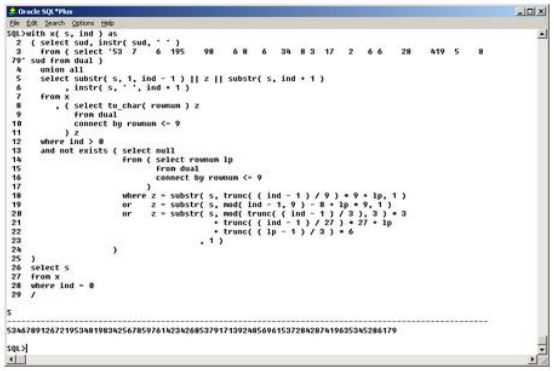
Say you have a sudoku like:

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

To solve this sudoku you first have to transforms this to a single string by appending all rows together:

"53 7 6 195 98 6 8 6 34 8 3 17 2 6 6 28 419 5 8 79"

Past this string into a Recursive Subquery, run it and you get a new string with your solved sudoku:



```
SQL>with x( s, ind ) as
 ( select sud, instr( sud, ' ' )
   from ( select '53  7    6  195    98    6 8    6    34  8 3  17    2    6 6    28    419  5    8
79' sud from dual )
   union all
   select substr( s, 1, ind - 1 ) || z || substr( s, ind + 1 )
        , instr( s, ' ', ind + 1 )
   from x
      , ( select to_char( rownum ) z
          from dual
          connect by rownum <= 9
        ) z
   where ind > 0
   and not exists ( select null
                    from ( select rownum lp
                           from dual
                           connect by rownum <= 9
                         )
                    where z = substr( s, trunc( ( ind - 1 ) / 9 ) * 9 + lp, 1 )
                    or    z = substr( s, mod( ind - 1, 9 ) - 8 + lp * 9, 1 )
                    or    z = substr( s, mod( trunc( ( ind - 1 ) / 3 ), 3 ) * 3
                                   + trunc( ( ind - 1 ) / 27 ) * 27 + lp
                                   + trunc( ( lp - 1 ) / 3 ) * 6
                                 , 1 )
                  )
 )
 select s
 from x
 where ind = 0
 /

S
--------------------------------------------------------------------------------
534678912672195348198342567859761423426853791713924856961537284287419635345286179

SQL>
```

Pattern Matching

I need to group a series of audit trail records together based on how close they are to each other. All records within three seconds of each other should be in a group

Row Pattern Matching

- This can be done with analytics
- Requires three passes
  - Tag records with ROW_NUMBER as RN that have a timestamp more then 3 seconds away from the prior record
  - Carry down the maximum RN, all records in a group will have the same RN now
  - Aggregate
- Can be done with vanilla SQL
  - To say it is hard would be one thing
  - To say it does not perform well is another
- Can be done in a single pass *easily* with pattern matching

Set Up

```
ops$tkyte%ORA12CR1> create table t ( x date primary key, y int );
Table created.
```

Set Up

```
ops$tkyte%ORA12CR1> select x,y
  2     from t
  3    order by x;


X            Y
-------- ----
12:22:03   94
12:22:04   96
12:22:05   87
12:22:06   94
12:22:07   55
12:22:08   57
12:22:09  100
12:22:10   97
12:22:11   40
12:22:12   13
```

```
12:22:18   54
12:22:19   84
12:22:20   26
12:22:36   47
12:22:37   57
12:22:38   52
12:22:39   11
12:22:40   39
12:22:41    4
12:22:42   86
12:22:43   54
12:22:44   47
12:22:45   97
12:22:46   44
12:22:47   17

25 rows selected.
```

```
insert into t2 (tag,min_x,max_x,sum_y)
SELECT 'old sql' "'OLDSQL'",MIN("A1"."X") "MIN(X)",MAX("A1"."X") "MAX(X)",SUM("A1"."Y") "SUM(Y)"
  FROM  (SELECT "A2"."X" "X","A2"."Y" "Y","A2"."GRP" "GRP",
                (SELECT MAX("A3"."GRP") "MAX(GRP)"
                   FROM  (SELECT "A4"."X" "X","A4"."Y" "Y","A4"."GRP" "GRP"
                            FROM (SELECT "A5"."X" "X","A5"."Y" "Y",
                                          CASE  WHEN "A5"."FIRST_X"="A5"."X" THEN "A5"."X"
                                                WHEN "A5"."X"-"A5"."NEXT_X">3/24/60/60 THEN "A5"."X"
                                           END  "GRP"
                                    FROM  (SELECT "A6"."X" "X","A6"."Y" "Y",
                                                  (SELECT MAX("A8"."X") "MAX(T2.X)"
                                                     FROM OPS$TKYTE."T" "A8"
                                                    WHERE "A8"."X"<"A6"."X") "NEXT_X",
                                                  (SELECT MIN("A7"."X") "MIN(T3.X)"
                                                     FROM OPS$TKYTE."T" "A7") "FIRST_X"
                                             FROM OPS$TKYTE."T" "A6")
                                   "A5")
                            "A4") "A3"
          WHERE "A3"."X"<="A2"."X") "NEW_GRP"
   FROM  (SELECT "A9"."X" "X","A9"."Y" "Y" ,"A9"."GRP" "GRP"
            FROM  (SELECT "A10"."X" "X","A10"."Y" "Y",
                           CASE  WHEN "A10"."FIRST_X"="A10"."X" THEN "A10"."X"
                                 WHEN "A10"."X"-"A10"."NEXT_X">3/24/60/60 THEN "A10"."X"
                            END  "GRP"
   FROM  (SELECT "A11"."X" "X","A11"."Y" "Y",
                (SELECT MAX("A13"."X") "MAX(T2.X)"
                   FROM OPS$TKYTE."T" "A13"
                  WHERE "A13"."X"<"A11"."X") "NEXT_X",
                (SELECT MIN("A12"."X") "MIN(T3.X)"
                   FROM OPS$TKYTE."T" "A12") "FIRST_X"
    FROM OPS$TKYTE."T" "A11") "A10") "A9") "A2") "A1"
   GROUP BY "A1"."NEW_GRP"
```

Analytics

```
ops$tkyte%ORA12CR1> select x, y,
  2         lag(x) over (order by x),
  3         case
  4            when abs(lag(x) over (order by x) - x) > 3/24/60/60
  5                 then row_number() over (order by x)
  6         end rn
  7    from t
  8  /
```

Analytics

```
…



X            Y LAG(X)OV   RN
--------- ---- --------- ---
12:22:03    94
12:22:04    96 12:22:03
12:22:05    87 12:22:04
12:22:06    94 12:22:05
12:22:07    55 12:22:06
12:22:08    57 12:22:07
12:22:09   100 12:22:08
12:22:10    97 12:22:09
12:22:11    40 12:22:10
12:22:12    13 12:22:11
```

```
12:22:18    54 12:22:12  11
12:22:19    84 12:22:18
12:22:20    26 12:22:19
12:22:36    47 12:22:20  14
12:22:37    57 12:22:36
12:22:38    52 12:22:37
12:22:39    11 12:22:38
12:22:40    39 12:22:39
12:22:41     4 12:22:40
12:22:42    86 12:22:41
12:22:43    54 12:22:42
12:22:44    47 12:22:43
12:22:45    97 12:22:44
12:22:46    44 12:22:45
12:22:47    17 12:22:46

25 rows selected.
```

Analytics

```
ops$tkyte%ORA12CR1> select x, y,
  2         max(rn) over (order by x) max_rn
  3    from (
  4  select x, y,
  5         lag(x) over (order by x),
  6         case
  7            when abs(lag(x) over (order by x) - x) > 3/24/60/60
  8                 then row_number() over (order by x)
  9         end rn
 10    from t
 11         )
 12  /
```

Analytics

```
…

X            Y     MAX_RN
--------- ---- ----------
12:22:03    94
12:22:04    96
12:22:05    87
12:22:06    94
12:22:07    55
12:22:08    57
12:22:09   100
12:22:10    97
12:22:11    40
12:22:12    13
```

```
12:22:18   54           11
12:22:19   84           11
12:22:20   26           11
12:22:36   47           14
12:22:37   57           14
12:22:38   52           14
12:22:39   11           14
12:22:40   39           14
12:22:41    4           14
12:22:42   86           14
12:22:43   54           14
12:22:44   47           14
12:22:45   97           14
12:22:46   44           14
12:22:47   17           14

25 rows selected.
```

Analytics

```
ops$tkyte%ORA12CR1> select min(x), max(x), sum(y)
  from (
select x, y,
       max(rn) over (order by x) max_rn
  from (
select x, y,
       lag(x) over (order by x),
       case
          when abs(lag(x) over (order by x) - x) > 3/24/60/60
               then row_number() over (order by x)
       end rn
  from t
       )
       )
 group by max_rn
 order by 1;
```

Analytics

```
MIN(X)   MAX(X)       SUM(Y)
-------- -------- ----------
12:22:03 12:22:12        733
12:22:18 12:22:20        164
12:22:36 12:22:47        555
```

```
ops$tkyte%ORA12CR1> select *
  2     from t
  3   match_recognize
  4   ( order by x
  5     measures first(x) start_time,
  6              last(x)  end_time,
  7              sum(y)   sum_y
  8     one row per match
  9     after match skip past last row
 10     pattern (any_row another_row_within_3_secs*)
 11     define
 12       another_row_within_3_secs as (x-prev(x)) <= 3/24/60/60
 13   );
```

```
ops$tkyte%ORA12CR1> select *
  2    from t
  3  match_recognize
  4  ( order by x
  5    measures first(x) start_time,
  6             last(x)  end_time,
  7             sum(y)   sum_y
  8    one row per match
  9    after match skip past last row
 10    pattern (any_row another_row_within_3_secs*)
 11    define
 12      another_row_within_3_secs as (x-prev(x)) <= 3/24/60/60
 13  );
```

```
ops$tkyte%ORA12CR1> select *
  2    from t
  3  match_recognize
  4  ( order by x
  5    measures first(x) start_time,
  6             last(x)  end_time,
  7             sum(y)   sum_y
  8    one row per match
  9    after match skip past last row
 10    pattern (any_row another_row_within_3_secs*)
 11    define
 12      another_row_within_3_secs as (x-prev(x)) <= 3/24/60/60
 13  );
```

```
ops$tkyte%ORA12CR1> select *
  2    from t
  3  match_recognize
  4  ( order by x
  5    measures first(x) start_time,
  6             last(x)  end_time,
  7             sum(y)   sum_y
  8    one row per match
  9    after match skip past last row
 10    pattern (any_row another_row_within_3_secs*)
 11    define
 12      another_row_within_3_secs as (x-prev(x)) <= 3/24/60/60
 13  );
```

```
ops$tkyte%ORA12CR1> select *
  2     from t
  3   match_recognize
  4   ( order by x
  5     measures first(x) start_time,
  6              last(x)  end_time,
  7              sum(y)   sum_y
  8     one row per match
  9     after match skip past last row
 10     pattern (any_row another_row_within_3_secs*)
 11     define
 12       another_row_within_3_secs as (x-prev(x)) <= 3/24/60/60
 13   );
```

```
ops$tkyte%ORA12CR1> select *
  2    from t
  3  match_recognize
  4  ( order by x
  5    measures first(x) start_time,
  6             last(x)  end_time,
  7             sum(y)   sum_y
  8    one row per match
  9    after match skip past last row
 10    pattern (any_row another_row_within_3_secs*)
 11    define
 12      another_row_within_3_secs as (x-prev(x)) <= 3/24/60/60
 13  );
```

```
ops$tkyte%ORA12CR1> select *
  2     from t
  3   match_recognize
  4    ( order by x
  5      measures first(x) start_time,
  6               last(x)  end_time,
  7               sum(y)   sum_y
  8      one row per match
  9      after match skip past last row
 10      pattern (any_row another_row_within_3_secs*)
 11      define
 12        another_row_within_3_secs as (x-prev(x)) <= 3/24/60/60
 13   );
```

```
ops$tkyte%ORA12CR1> select *
  2    from t
  3  match_recognize
  4  ( order by x
  5    measures first(x) start_time,
  6             last(x)  end_time,
  7             sum(y)   sum_y
  8    one row per match
  9    after match skip past last row
 10    pattern (any_row another_row_within_3_secs*)
 11    define
 12      another_row_within_3_secs as (x-prev(x)) <= 3/24/60/60
 13  );
```

ORACLE

```
ops$tkyte%ORA12CR1> select *
  2    from t
  3  match_recognize
  4  ( order by x
  5    measures first(x) start_time,
  6             last(x)  end_time,
  7             sum(y)   sum_y
  8    one row per match
  9    after match skip past last row
 10    pattern (any_row another_row_within_3_secs*)
 11    define
 12      another_row_within_3_secs as (x-prev(x)) <= 3/24/60/60
 13  );
```

Old SQL

```
call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.00       0.00          0          0          0           0
Execute      1    506.99     524.81          0     230292         56         501
Fetch        0      0.00       0.00          0          0          0           0
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total        2    506.99     524.82          0     230292         56         501


Rows (1st) Rows (avg) Rows (max)  Row Source Operation
---------- ---------- ----------  ---------------------------------------------------
         0          0          0  LOAD TABLE CONVENTIONAL  (cr=230290 pr=0 pw=0 time=524810816
         1          1          1   SORT AGGREGATE (cr=2 pr=0 pw=0 time=46 us)
         1          1          1    INDEX FULL SCAN (MIN/MAX) SYS_C0015724 (cr=2 pr=0 pw=0 tim
   8455484    8455484    8455484   SORT AGGREGATE (cr=203546 pr=0 pw=0 time=407371174 us)
   8455484    8455484    8455484    FIRST ROW  (cr=203546 pr=0 pw=0 time=223073082 us cost=2 s
   8455484    8455484    8455484     INDEX RANGE SCAN (MIN/MAX) SYS_C0015724 (cr=203546 pr=0 p
      5000       5000       5000   SORT AGGREGATE (cr=230270 pr=0 pw=0 time=524705644 us)
  12502500   12502500   12502500    INDEX RANGE SCAN SYS_C0015724 (cr=26722 pr=0 pw=0 tim
       501        501        501   HASH GROUP BY (cr=230286 pr=0 pw=0 time=524812282 us cos
      5000       5000       5000    VIEW  (cr=230286 pr=0 pw=0 time=498097951 us cost=640158
      5000       5000       5000     TABLE ACCESS FULL T (cr=16 pr=0 pw=0 time=17660 us cost=5
```

Analytics vs Pattern Matching

```
call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.00       0.01          0          0          0           0
Execute      2     46.75      50.92     111960      32908      45801     1000001
Fetch        0      0.00       0.00          0          0          0           0
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total        3     46.75      50.94     111960      32908      45801     1000001


Rows (1st) Rows (avg) Rows (max)  Row Source Operation
---------- ---------- ----------  ---------------------------------------------------
         0          0          0  LOAD TABLE CONVENTIONAL  (cr=34061 pr=112144 pw=83551 time=50950237
   1000001    1000001    1000001   HASH GROUP BY (cr=24896 pr=111952 pw=83551 time=55427557 us cost=5
  10000000   10000000   10000000    VIEW  (cr=24896 pr=107798 pw=79397 time=350242521 us cost=54354 siz
  10000000   10000000   10000000     WINDOW BUFFER (cr=24896 pr=107798 pw=79397 time=109267393 us cost=
  10000000   10000000   10000000      VIEW  (cr=24896 pr=73444 pw=49453 time=332314841 us cost=54354 si
  10000000   10000000   10000000       WINDOW SORT (cr=24896 pr=73444 pw=49453 time=77674611 us cost=5
  10000000   10000000   10000000        TABLE ACCESS FULL T (cr=24896 pr=23991 pw=0 time=52818046 us co
```

Analytics vs Pattern Matching

```
call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.00       0.01          0          0          0           0
Execute      1     20.01      23.09      74332      32711      43012     1000001
Fetch        0      0.00       0.00          0          0          0           0
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total        2     20.01      23.11      74332      32711      43012     1000001


Rows (1st) Rows (avg) Rows (max)  Row Source Operation
---------- ---------- ----------  ---------------------------------------------------
         0          0          0  LOAD TABLE CONVENTIONAL  (cr=33281 pr=74332 pw=49448 time=23103728 u
   1000001    1000001    1000001   VIEW  (cr=24888 pr=74332 pw=49448 time=51930592 us cost=54354 size=
   1000001    1000001    1000001    MATCH RECOGNIZE SORT DETERMINISTIC FINITE AUTOMATON (cr=24888 pr=7
  10000000   10000000   10000000     TABLE ACCESS FULL T (cr=24888 pr=24884 pw=0 time=55735750 us cost=
```

External Tables

External Tables
Sqlldr is the legacy data loading tool from the 20th century

- Query flat files
- Query datapump format files
- Query output of programs (10.2.0.5 and above)
  - Load compressed files without uncompressing
  - Query program output, like ls, ps, df, etc
- Query HDFS/Hive
- Infinite possibilities

```
ops$tkyte%ORA12C> CREATE TABLE EMP_ET
  2   (
  3     "EMPNO" NUMBER(4),
  4     "ENAME" VARCHAR2(10),
  5     "JOB" VARCHAR2(9),
  6     "MGR" NUMBER(4),
  7     "HIREDATE" DATE,
  8     "SAL" NUMBER(7,2),
  9     "COMM" NUMBER(7,2),
 10     "DEPTNO" NUMBER(2)
 11   )
 12   ORGANIZATION external
 13   ( TYPE oracle_loader
 14     DEFAULT DIRECTORY load_dir
 15     ACCESS PARAMETERS
 16     ( RECORDS DELIMITED BY NEWLINE
 17           preprocessor  exec_dir:'run_gunzip.sh'
 18       FIELDS TERMINATED BY "|" LDRTRIM
 19     )
 20     location ( 'emp.dat.gz')
 21   )
 22   /
Table created.
```

```
ops$tkyte%ORA12C> !file emp.dat.gz
emp.dat.gz: gzip compressed data, was "emp.dat", from Unix, last …

ops$tkyte%ORA12C> !cat run_gunzip.sh
#!/bin/bash

/usr/bin/gunzip -c $*

ops$tkyte%ORA11GR2> select empno, ename from emp_et where rownum <= 5;

     EMPNO ENAME
---------- ----------
      7369 SMITH
      7499 ALLEN
      7521 WARD
      7566 JONES
      7654 MARTIN
```

```
SQL> !cat /home/tkyte/df
#!/bin/bash
/bin/df -Pl

SQL> !/home/tkyte/run_df.sh
Filesystem          1024-blocks       Used  Available   Capacity  Mounted on
/dev/mapper/VolGr...   18156292  10827600    6391528    63%       /
/dev/sda1                101086     12062      83805    13%       /boot
tmpfs                    517520         0     517520    0%        /dev/shm
```

```
SQL> create table df
  2  (
  3   fsname   varchar2(100),
  4   blocks   number,
  5   used     number,
  6   avail    number,
  7   capacity varchar2(10),
  8   mount    varchar2(100)
  9  )
 10  organization external
 11  (
 12    type oracle_loader
 13    default directory exec_dir
 14    access parameters
```

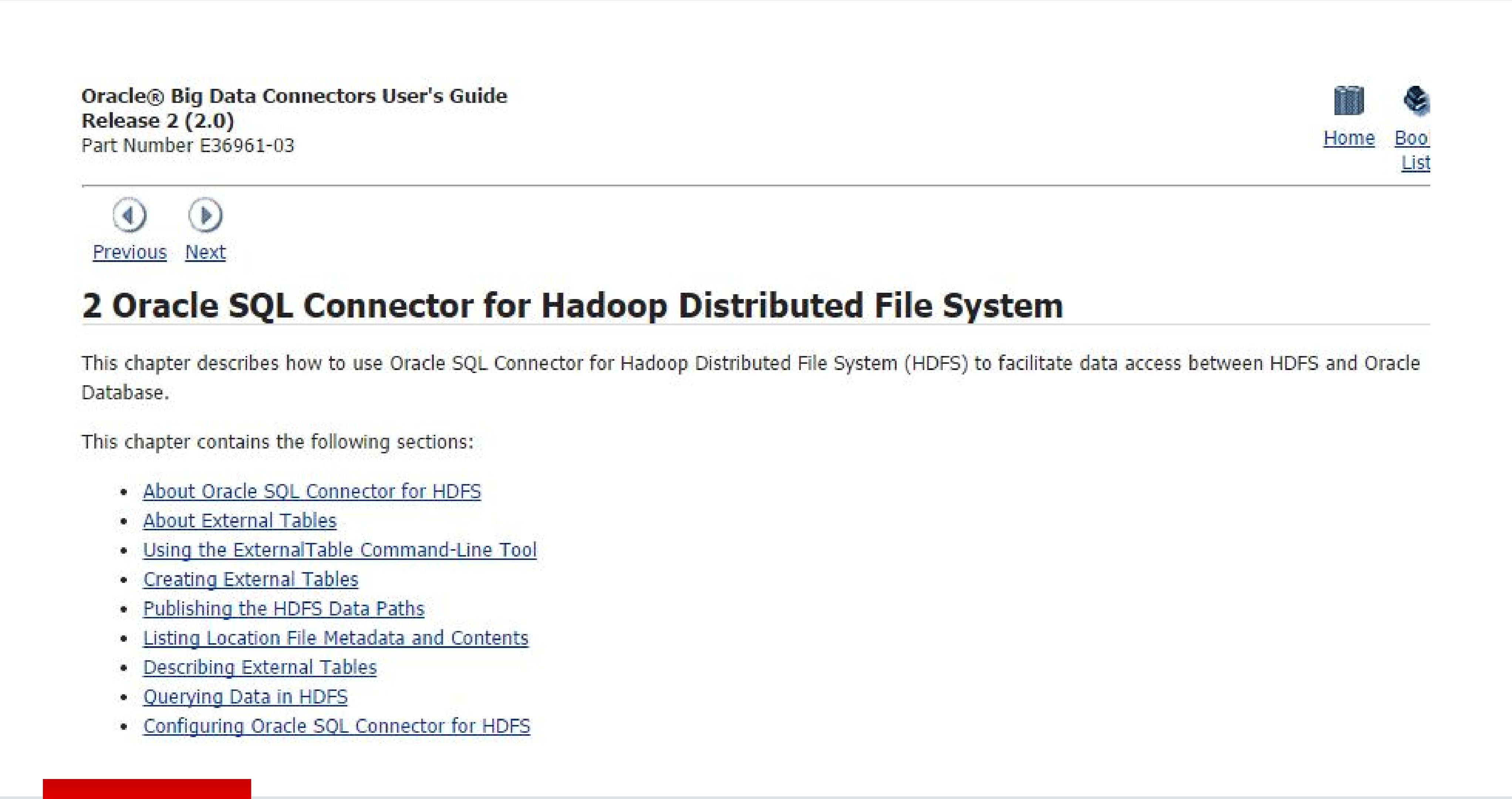
```
15    (
16      records delimited
17      by newline
18      preprocessor
19      exec_dir:'run_df.sh'
20      skip 1
21      fields terminated by
22      whitespace ldrtrim
23    )
24    location
25    (
26      exec_dir:'run_df.sh'
27    )
28  )
29  /
Table created.
```

```
SQL> select * from df;

FSNAME                           BLOCKS   USED      AVAIL    CAPACITY MOUNT
-------------------------------- -------- --------- -------- -------- ------
/dev/mapper/VolGroup00-LogVol00  18156292 10827600  6391528    63%    /
/dev/sda1                          101086     12062   83805    13%    /boot
tmpfs                              517520         0  517520     0%    /dev/shm
```

```
with fs_data
as
(select /*+ materialize */ * from df)
select mount,
       file_name,
       bytes,
       tot_bytes,
       avail_bytes,
       case
         when 0.2 * tot_bytes < avail_bytes
         then 'OK'
         else 'Short on disk space'
       end status
  from (
select file_name, mount, avail_bytes, bytes,
       sum(bytes) over
         (partition by mount) tot_bytes
```

```
 from (
select a.file_name,
       b.mount,
       b.avail*1024 avail_bytes, a.bytes,
       row_number() over
         (partition by a.file_name
          order by length(b.mount) DESC) rn
  from dba_data_files a,
       fs_data b
 where a.file_name
           like b.mount || '%'
       )
 where rn = 1
       )
 order by mount, file_name
```

ORACLE



Oracle® Big Data Connectors User's Guide
Release 2 (2.0)
Part Number E36961-03

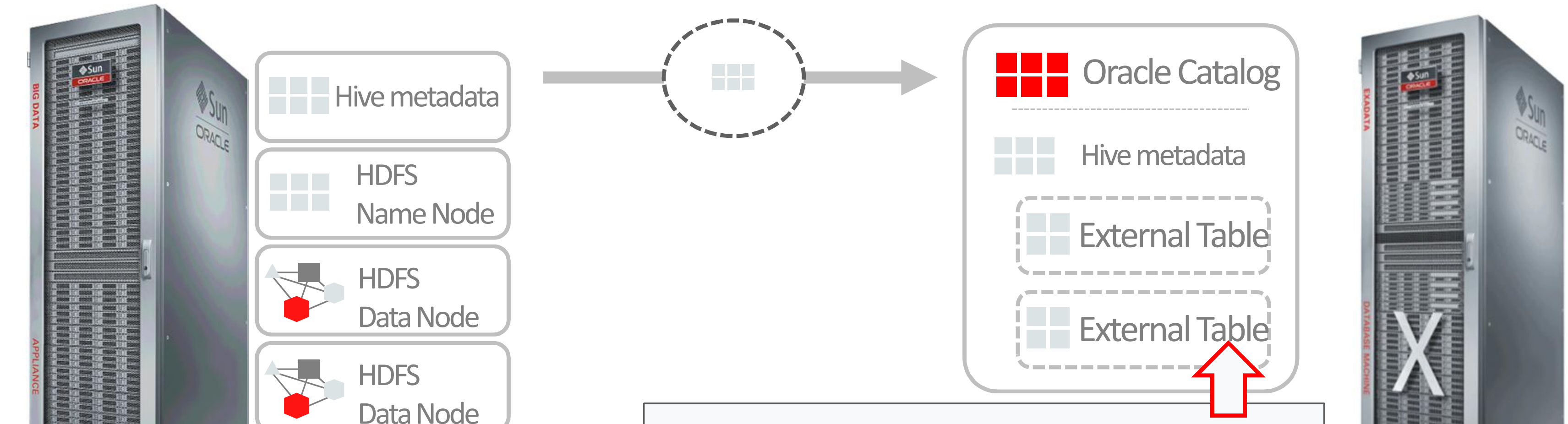
2 Oracle SQL Connector for Hadoop Distributed File System

This chapter describes how to use Oracle SQL Connector for Hadoop Distributed File System (HDFS) to facilitate data access between HDFS and Oracle Database.

This chapter contains the following sections:

SQL Over Hadoop (and others)

Publish Hadoop Metadata to Oracle Catalog



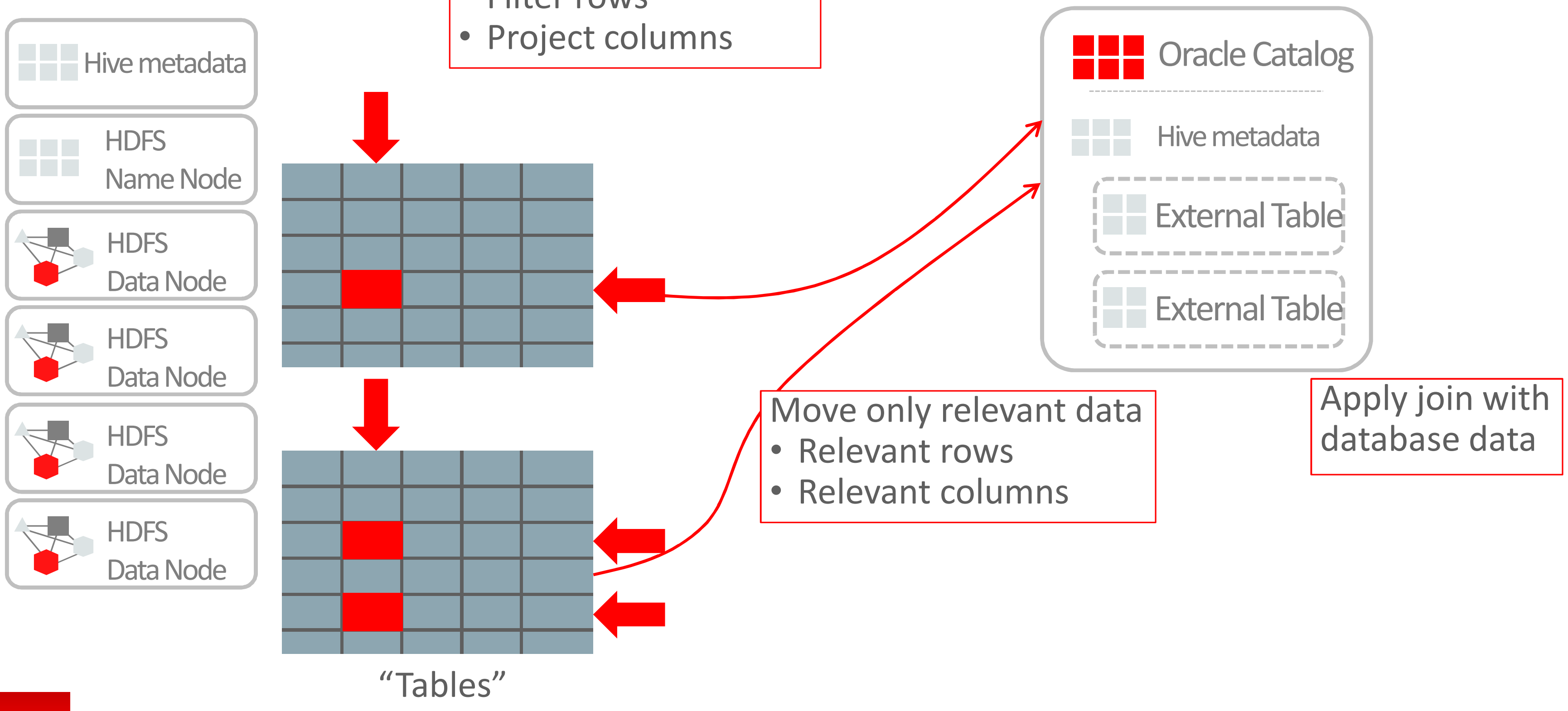
Big Data Appliance
+
Hadoop

```
create table customer_address
( ca_customer_id       number(10,0)
, ca_street_number  char(10)
, ca_state          char(2)
, ca_zip            char(10)
)
organization external (
TYPE ORACLE_HIVE
DEFAULT DIRECTORY DEFAULT_DIR
ACCESS PARAMETERS
    (com.oracle.bigdata.cluster hadoop_cl_1)
LOCATION ('hive://customer_address')
)
```

Exadata
+
Oracle Database

Executing Queries on Hadoop

```
Select c_customer_id
,      c_customer_last_name
,      ca_county
From   customers
,      customer_address
where  c_customer_id = ca_customer_id
and    ca_state = ‘CA’
```



Do I/O and Smart Scan:
- Filter rows
- Project columns

Move only relevant data
- Relevant rows
- Relevant columns

Apply join with database data

“Tables”

Data lives in even more places

The magic of Storage Handlers

SQL

Hadoop

Relational

N

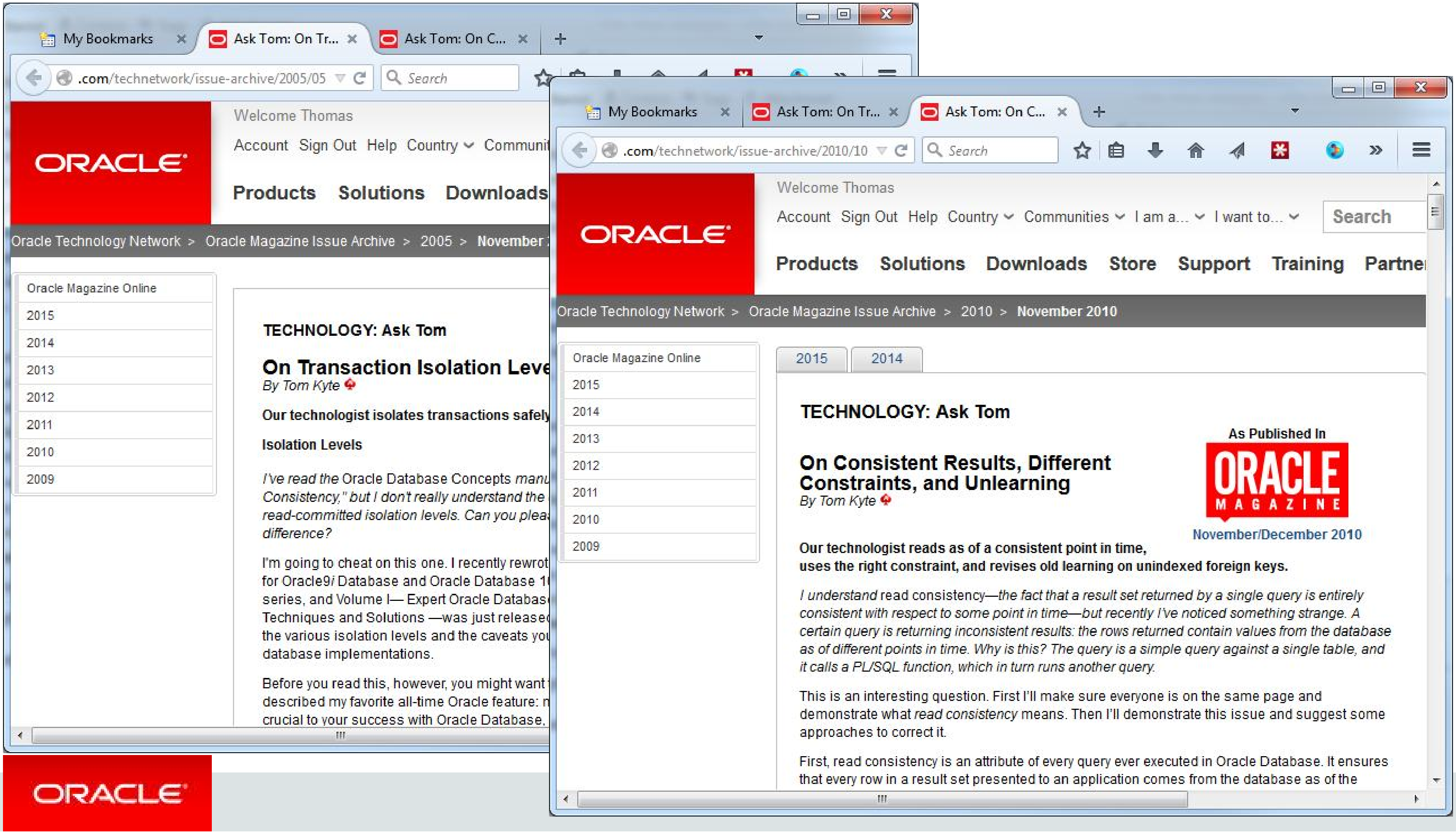
ORACLE NOSQL DATABASE

mongoDB

Cassandra

And more…

My Favorite Oracle Feature



My Bookmarks
Ask Tom: On Tr...
Ask Tom: On C...
.com/technetwork/issue-archive/2005/05
Search
Welcome Thomas
Account Sign Out Help Country Communi
Products Solutions Downloads
Oracle Technology Network > Oracle Magazine Issue Archive > 2005 > November
Oracle Magazine Online
2015
2014
2013
2012
2011
2010
2009

TECHNOLOGY: Ask Tom

On Transaction Isolation Leve
By *Tom Kyte*

Our technologist isolates transactions safely

Isolation Levels

I've read the Oracle Database Concepts *manu* *Consistency," but I don't really understand the* *read-committed isolation levels. Can you plea* *difference?*

I'm going to cheat on this one. I recently rewrot for Oracle9*i* Database and Oracle Database 1 series, and Volume I— Expert Oracle Databas Techniques and Solutions —was just release the various isolation levels and the caveats yo database implementations.

Before you read this, however, you might want described my favorite all-time Oracle feature: n crucial to your success with Oracle Database.

My Bookmarks
Ask Tom: On Tr...
Ask Tom: On C...
.com/technetwork/issue-archive/2010/10
Search
Welcome Thomas
Account Sign Out Help Country Communities I am a... I want to...
Search
Products Solutions Downloads Store Support Training Partne
Oracle Technology Network > Oracle Magazine Issue Archive > 2010 > November 2010
Oracle Magazine Online
2015
2014
2013
2012
2011
2010
2009
2015
2014

TECHNOLOGY: Ask Tom

As Published In ORACLE MAGAZINE November/December 2010

On Consistent Results, Different Constraints, and Unlearning
By *Tom Kyte*

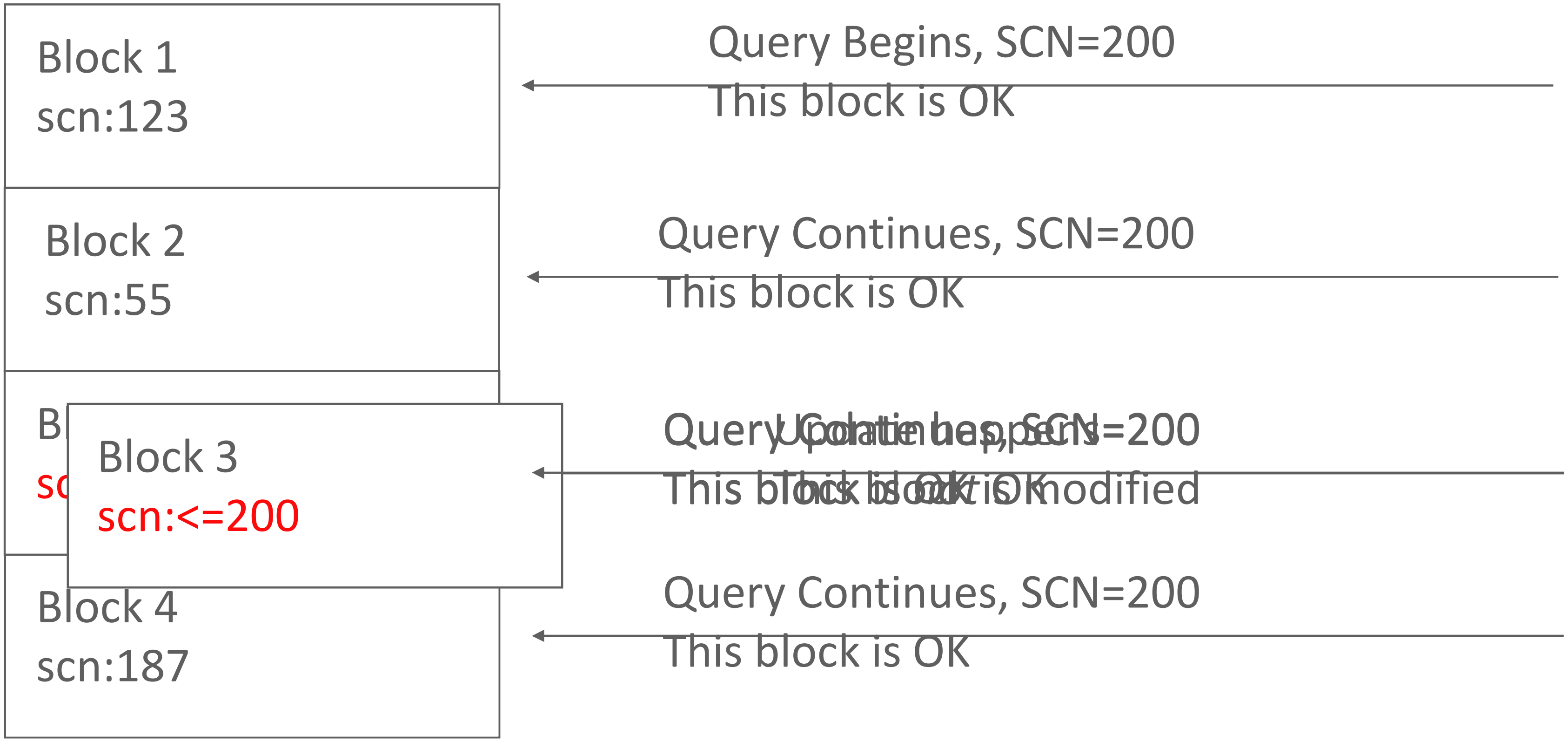
Our technologist reads as of a consistent point in time, uses the right constraint, and revises old learning on unindexed foreign keys.

I understand read consistency*—the fact that a result set returned by a single query is entirely consistent with respect to some point in time—but recently I've noticed something strange. A certain query is returning inconsistent results: the rows returned contain values from the database as of different points in time. Why is this? The query is a simple query against a single table, and it calls a PL/SQL function, which in turn runs another query.*

This is an interesting question. First I'll make sure everyone is on the same page and demonstrate what *read consistency* means. Then I'll demonstrate this issue and suggest some approaches to correct it.

First, read consistency is an attribute of every query ever executed in Oracle Database. It ensures that every row in a result set presented to an application comes from the database as of the

What does it do…



Block 1
scn:123

Query Begins, SCN=200
This block is OK

Block 2
scn:55

Query Continues, SCN=200
This block is OK

B
sc

Block 3
scn:<=200

Query Continues, SCN=200
This block is OK

Update happens
This block is modified

Block 4
scn:187

Query Continues, SCN=200
This block is OK

Why do you care

- Suppose you were pulling changes….

T-1: update some data at 11:59, do not commit, time recorded though

T0: 12 noon, start initial 'pull', remember this time (for refreshing later).

T.5: commit the T-1 transaction. It was not visible at T0, it is at T1

T1: Later on, go pull everything updated since 12 noon

T1.5: But – we will not see the T-1 update!!

T2: After committing T1's work, you have all of the data right?

ORACLE

```
ops$tkyte%ORA11GR2> create table t
  2  as
  3  select *
  4    from all_users
  5   where rownum <= 5;

Table created.
```

```
ops$tkyte%ORA11GR2> create or replace function f
  2  return number
  3  as
  4      pragma autonomous_transaction;
  5      l_cnt  number;
  6  begin
  7      select count(*) into l_cnt from t;
  8
  9      insert into t (username, user_id, created )
 10      values ( 'hello', 123, sysdate );
 11      commit;
 12
 13      return l_cnt;
 14  end;
 15  /
```

```
ops$tkyte%ORA11GR2> select count(*) over () cnt1,
  2          (select count(*) from t) cnt2,
  3          f() cnt3,
  4          (select f() from dual) cnt4
  5     from t;
```

```
ops$tkyte%ORA11GR2> select count(*) over () cnt1,
  2          (select count(*) from t) cnt2,
  3          f() cnt3,
  4          (select f() from dual) cnt4
  5     from t;

      CNT1       CNT2       CNT3       CNT4
---------- ---------- ---------- ----------
         5          5          5          6
         5          5          7          6
         5          5          8          6
         5          5          9          6
         5          5         10          6
```

```
ops$tkyte%ORA11GR2> create or replace
                    function f(p_scn in number)
  2  return number
  3  as
  4      pragma autonomous_transaction;
  5      l_cnt  number;
  6  begin
  7      select count(*) into l_cnt from t
  8          as of scn p_scn;
  9
 10      insert into t (username, user_id, created )
 11      values ( 'hello', 123, sysdate );
 12      commit;
 13
 14      return l_cnt;
 15  end;
 16  /
```

```
ops$tkyte%ORA11GR2> variable scn number
ops$tkyte%ORA11GR2> exec :scn :=
               dbms_flashback.get_system_change_number

PL/SQL procedure successfully completed.
```

```
ops$tkyte%ORA11GR2> select count(*) over () cnt1,
  2          (select count(*) from t) cnt2,
  3          f(:scn) cnt3,
  4          (select f(:scn) from dual) cnt4
  5    from t;

      CNT1       CNT2       CNT3       CNT4
---------- ---------- ---------- ----------
         5          5          5          5
         5          5          5          5
         5          5          5          5
         5          5          5          5
         5          5          5          5
```

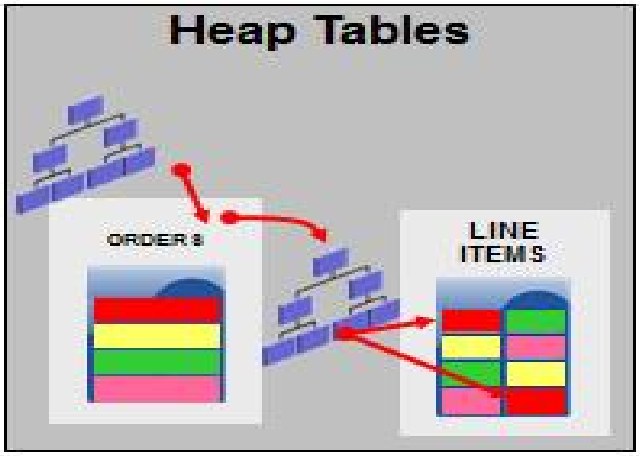
```
ops$tkyte%ORA11GR2> insert into t
(username, user_id, created ) values
( 'x', 1, sysdate );

1 row created.

ops$tkyte%ORA11GR2> exec :scn :=
             dbms_flashback.get_system_change_number

PL/SQL procedure successfully completed.
```

```
ops$tkyte%ORA11GR2> select count(*) over () cnt1,
  2         (select count(*) from t) cnt2,
  3         f(:scn) cnt3,
  4         (select f(:scn) from dual) cnt4
  5    from t;

      CNT1       CNT2       CNT3       CNT4
---------- ---------- ---------- ----------
         6          6          5          5
         6          6          5          5
         6          6          5          5
         6          6          5          5
         6          6          5          5
         6          6          5          5
6 rows selected.
```

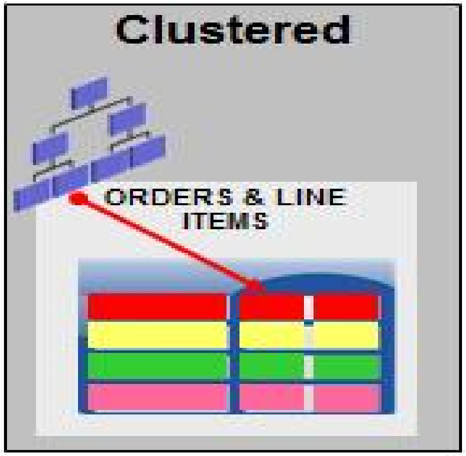
So many structures

So transparent to the application

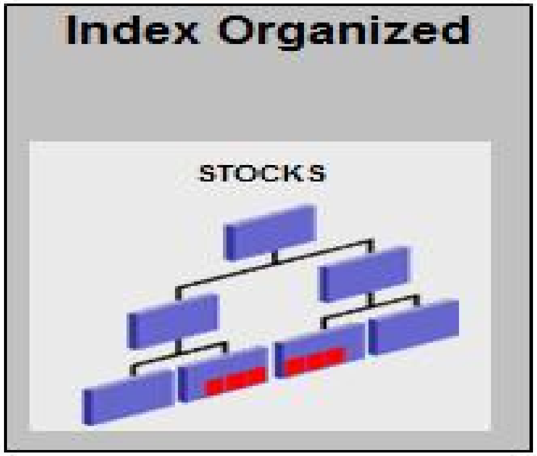
Tables



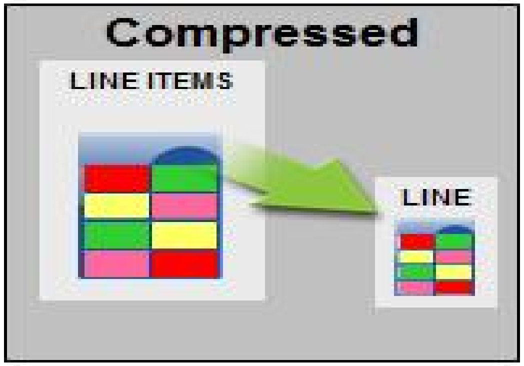
Heap Tables



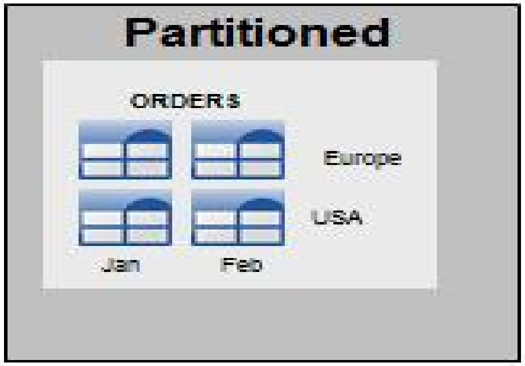
Clustered



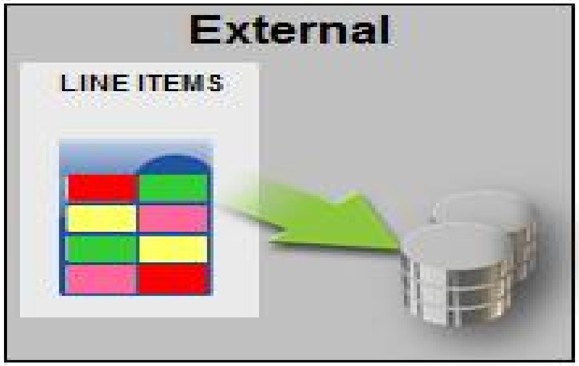
Index Organized



Compressed



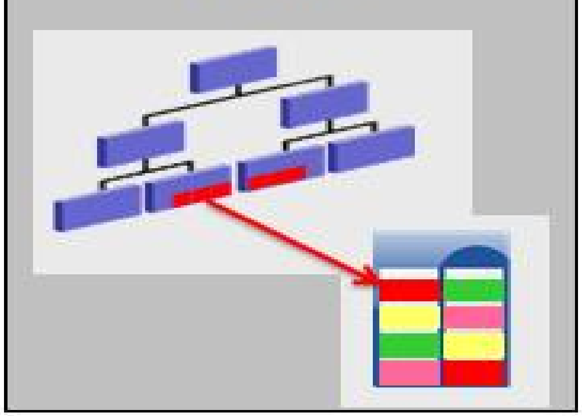
Partitioned



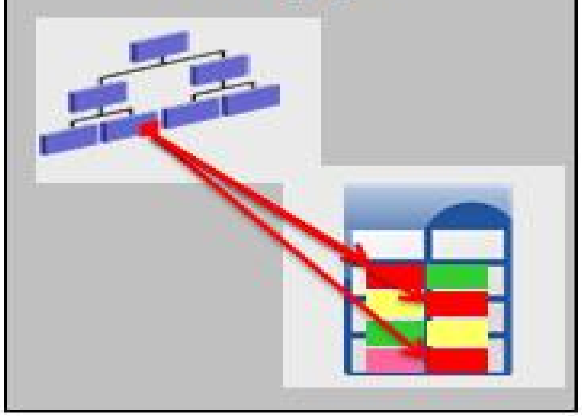
External

Indexes

B*Tree

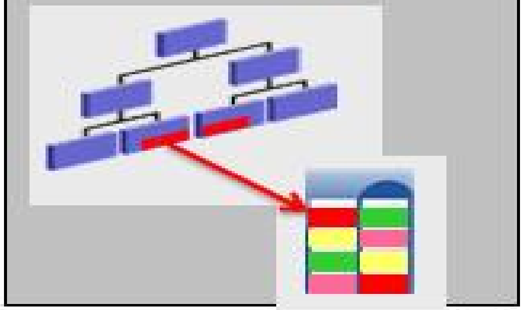


Bitmapped

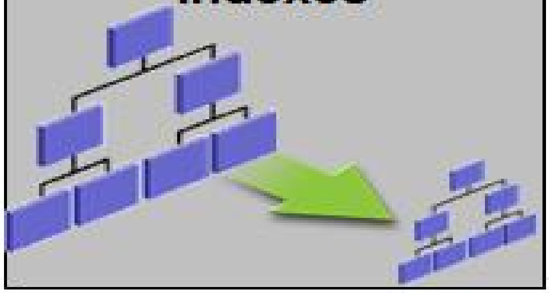


Function Based

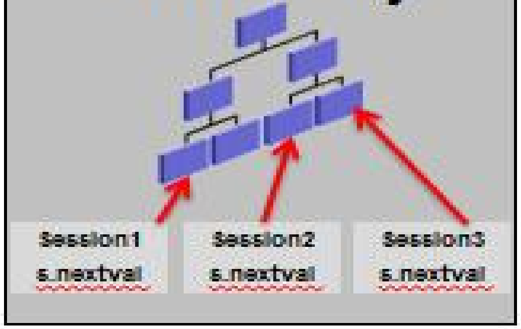
```
Create index
on((a+b)/2)
```



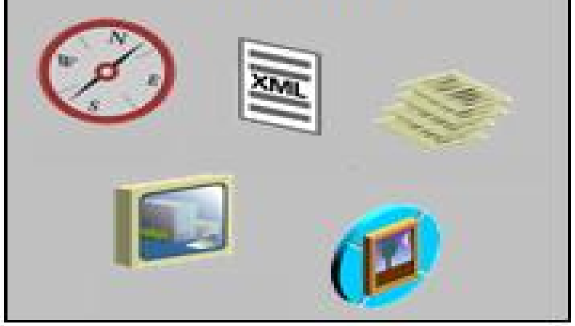
Compressed Indexes



Reverse Key



Domain



Think about Slow by Slow (aka Row by Row)

Versus Sets

http://oracle.com/goto/oll/rwp

ETL @1ms per item

```
• ITEMS                SP                   FMT
• -------------------- -------------------- --------------------
•            1,000,000 One million          16.67 minutes
•           10,000,000 Ten million          2.78 hours
```

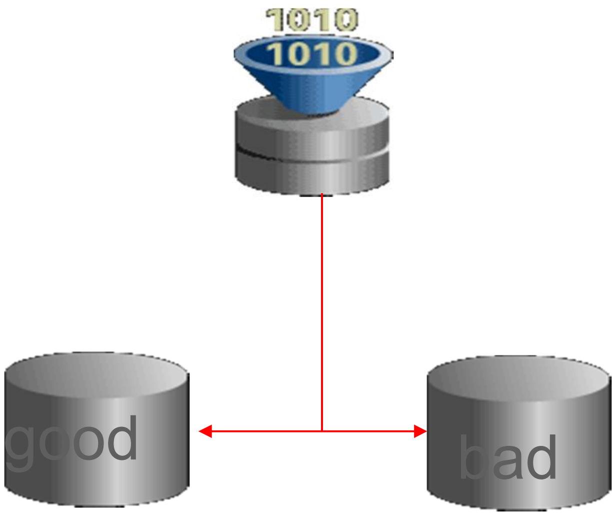
http://www.oracle.com/goto/oll/rwp

```
•       10,000,000,000 Ten billion          3.86 months
•      100,000,000,000 One Hundred billion  3.17 years
•    1,000,000,000,000 One trillion         31.69 years
•   10,000,000,000,000 Ten trillion         316.88 years
•  100,000,000,000,000 One Hundred trillion 3168.81 years
```

I can’t do it set based – because error handling

LOG ERRORS

- This will change the way ETL is done.
- Modifications are no longer “*all for one, and one for all*”
- Errors can be captured in a logging table, for easy resolution afterwards
- Like a bad file from sqlldr, only better.



Where…

- Virtually any place you are doing slow by slow processing
- Less code = Less bugs

Also…

- What about a multi-table insert

```
Insert /*+ append */ all
When (this that and the other thing)
Then
Into my_table values ( … )
Else
Into my_bad values ( … )
Select ….
```

DML is not the only way

Would you like to load and validate one row 5,000,000 times

or load and validate 5,000,000 rows *once*?

```
create or replace procedure slow_by_slow
as
begin
    for x in (select rowid rid, object_name
                from t t_slow_by_slow)
    loop
        x.object_name := substr(x.object_name,2)
                      ||substr(x.object_name,1,1);
        update t
           set object_name = x.object_name
         where rowid = x.rid;
    end loop;
end;
/
```

```
create or replace procedure bulk
as
    type ridArray is table of rowid;
    type onameArray is table
             of t.object_name%type;

    cursor c is select rowid rid, object_name
                  from t t_bulk;
    l_rids      ridArray;
    l_onames    onameArray;
    N           number := 100;
begin
    open c;
    loop
        fetch c bulk collect
        into l_rids, l_onames limit N;
      for i in 1 .. l_rids.count
      loop
          l_onames(i) := substr(l_onames(i),2)
                      ||substr(l_onames(i),1,1);
      end loop;
      forall i in 1 .. l_rids.count
          update t
             set object_name = l_onames(i)
           where rowid = l_rids(i);
      exit when c%notfound;
   end loop;
   close c;
end;
```

```
update t
   set object_name = substr(object_name,2) ||
                     substr(object_name,1,1);
```

```
create table new_table
as
select OWNER,
substr(object_name,2) || substr(object_name,1,1) OBJECT_NAME,
SUBOBJECT_NAME, OBJECT_ID, DATA_OBJECT_ID, OBJECT_TYPE,
CREATED, LAST_DDL_TIME, TIMESTAMP, STATUS, TEMPORARY,
GENERATED, SECONDARY, NAMESPACE, EDITION_NAME
from t;
```

Method	CPU units of time
Slow by Slow	495
Bulk	193 (39%)
Single SQL statement	91 (18%) *near order of magnitude
CTAS	28 (5%) *two orders of magnitude

Fear of NULLS

Nulls and Indexes

- There is a pervasive myth that indexes and NULLs are like matter and anti-matter
- There is the thought that “where column is null” cannot use an index
- There is a thought that NULLs are not indexed

- None of that is true…

NULLs and Indexes

```
ops$tkyte%ORA11GR2> create table t
  2  as
  3  select a.*,
  4         case when mod(rownum,100) > 1
  5              then object_type
  6          end otype
  7    from all_objects a;

Table created.
```

NULLs and Indexes

```
ops$tkyte%ORA11GR2> select count(*) from t where
otype is null;

  COUNT(*)
----------
      1445
```

NULLs and Indexes

```
ops$tkyte%ORA11GR2> begin
  2     dbms_stats.gather_table_stats( user, 'T' );
  3   end;
  4   /

PL/SQL procedure successfully completed.
```

NULLs and Indexes

```
ops$tkyte%ORA11GR2> create index t_idx
                        on t(otype,owner);

Index created.
```

NULLs and Indexes

```
ops$tkyte%ORA11GR2> set autotrace traceonly explain
ops$tkyte%ORA11GR2> select * from t where otype is
null;

Execution Plan
-------------------------------------------------------
-
Plan hash value: 470836197
```

NULLs and Indexes

```
-----------------------------------------------------------------------------------
| Id  | Operation                   | Name  | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |       |  1445 |   149K|    96   (0)| 00:00:02 |
|   1 |  TABLE ACCESS BY INDEX ROWID| T     |  1445 |   149K|    96   (0)| 00:00:02 |
|*  2 |   INDEX RANGE SCAN          | T_IDX |  1445 |       |     7   (0)| 00:00:01 |
-----------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("OTYPE" IS NULL)
```

NULLs and Indexes

```
ops$tkyte%ORA11GR2> drop index t_idx;

Index dropped.

ops$tkyte%ORA11GR2> create index t_idx
                        on t(otype,0);

Index created.
```

NULLs and Indexes

```
-------------------------------------------------------------------------------------
| Id  | Operation                   | Name  | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |       |  1445 |   149K|    96   (0)| 00:00:02 |
|   1 |  TABLE ACCESS BY INDEX ROWID| T     |  1445 |   149K|    96   (0)| 00:00:02 |
|*  2 |   INDEX RANGE SCAN          | T_IDX |  1445 |       |     7   (0)| 00:00:01 |
-------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("OTYPE" IS NULL)
```

Nulls and Indexes

- What is true is that entirely NULL key entries are not made in B*Tree indexes
- Therefore, an index on just OTYPE cannot be used to find NULLs
- But – what about B*Tree cluster indexes and Bitmap indexes?

Fear of Nulls

- Use some out of range value
  - Which obviously changes the high/low values
  - Which impacts cardinality estimates

- Could the use of fake values lead to data integrity issues?

Fake Values

```
ops$tkyte%ORA11GR2> create table t
  2  as
  3  select *
  4    from (
  5  select add_months(sysdate,-100) + mod( rownum, 3000 ) dt
  6    from dual
  7   connect by level <= 1000000
  8         )
  9   where dt < trunc(sysdate,'y')
 10  /

Table created.

ops$tkyte%ORA11GR2> insert into t
  2  select *
  3    from (
  4  select null dt
  5    from dual
  6   connect by level <= 1000000
  7         )
  8  /

1000000 rows created.
```

Fake Values

```
ops$tkyte%ORA11GR2> exec dbms_stats.gather_table_stats( user, 'T' );

PL/SQL procedure successfully completed.
```

Fake Values

```
ops$tkyte%ORA11GR2> select count(*)
  2    from t
  3   where dt between to_date( '01-jun-2013' ) and to_date( '30-jun-2013' );

  COUNT(*)
----------
      9657


Execution Plan
----------------------------------------------------------
Plan hash value: 2966233522

---------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
---------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |     1 |     4 |   949   (2)| 00:00:12 |
|   1 |  SORT AGGREGATE    |      |     1 |     4 |            |          |
|*  2 |   TABLE ACCESS FULL| T    | 10337 | 41348 |   949   (2)| 00:00:12 |
---------------------------------------------------------------------------
```

Fake Values

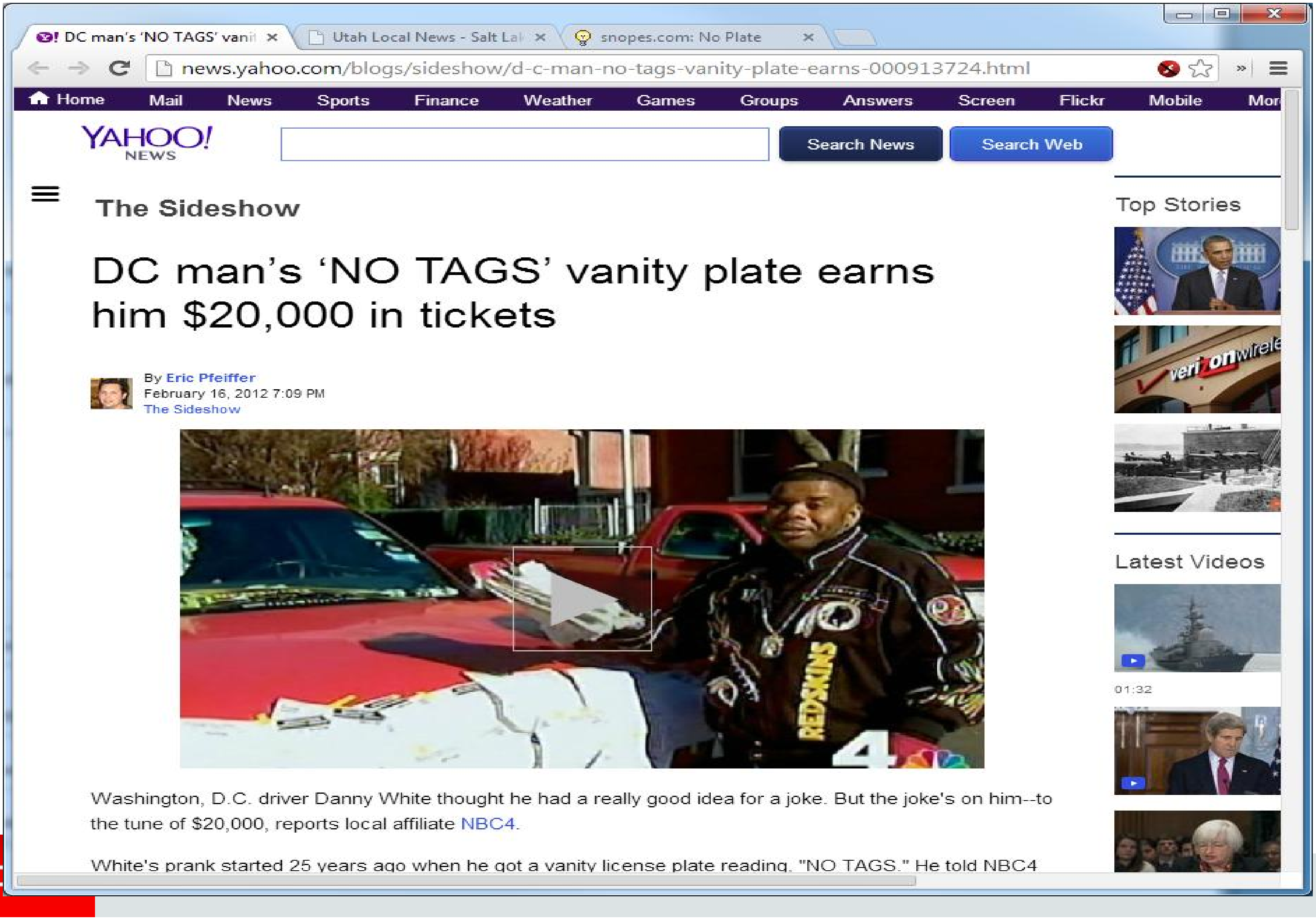
```
ops$tkyte%ORA11GR2> create table t
  2  as
  3  select *
  4    from (
  5  select add_months(sysdate,-100) + mod( rownum, 3000 ) dt
  6    from dual
  7   connect by level <= 1000000
  8          )
  9   where dt < trunc(sysdate,'y')
 10  /

Table created.

ops$tkyte%ORA11GR2> insert into t
  2  select *
  3    from (
  4  select to_date( '01-jan-9999') dt
  5    from dual
  6   connect by level <= 1000000
  7          )
  8  /

1000000 rows created.
```

Fake Values

```
ops$tkyte%ORA11GR2> exec dbms_stats.gather_table_stats( user, 'T' );

PL/SQL procedure successfully completed.
```

Fake Values

```
ops$tkyte%ORA11GR2> select count(*)
  2    from t
  3   where dt between to_date( '01-jun-2013' ) and to_date( '30-jun-2013' );

  COUNT(*)
----------
      9657


Execution Plan
----------------------------------------------------------
Plan hash value: 2966233522

---------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
---------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |     1 |     8 |  1018   (2)| 00:00:13 |
|   1 |  SORT AGGREGATE    |      |     1 |     8 |            |          |
|*  2 |   TABLE ACCESS FULL| T    |  1356 | 10848 |  1018   (2)| 00:00:13 |
---------------------------------------------------------------------------
```



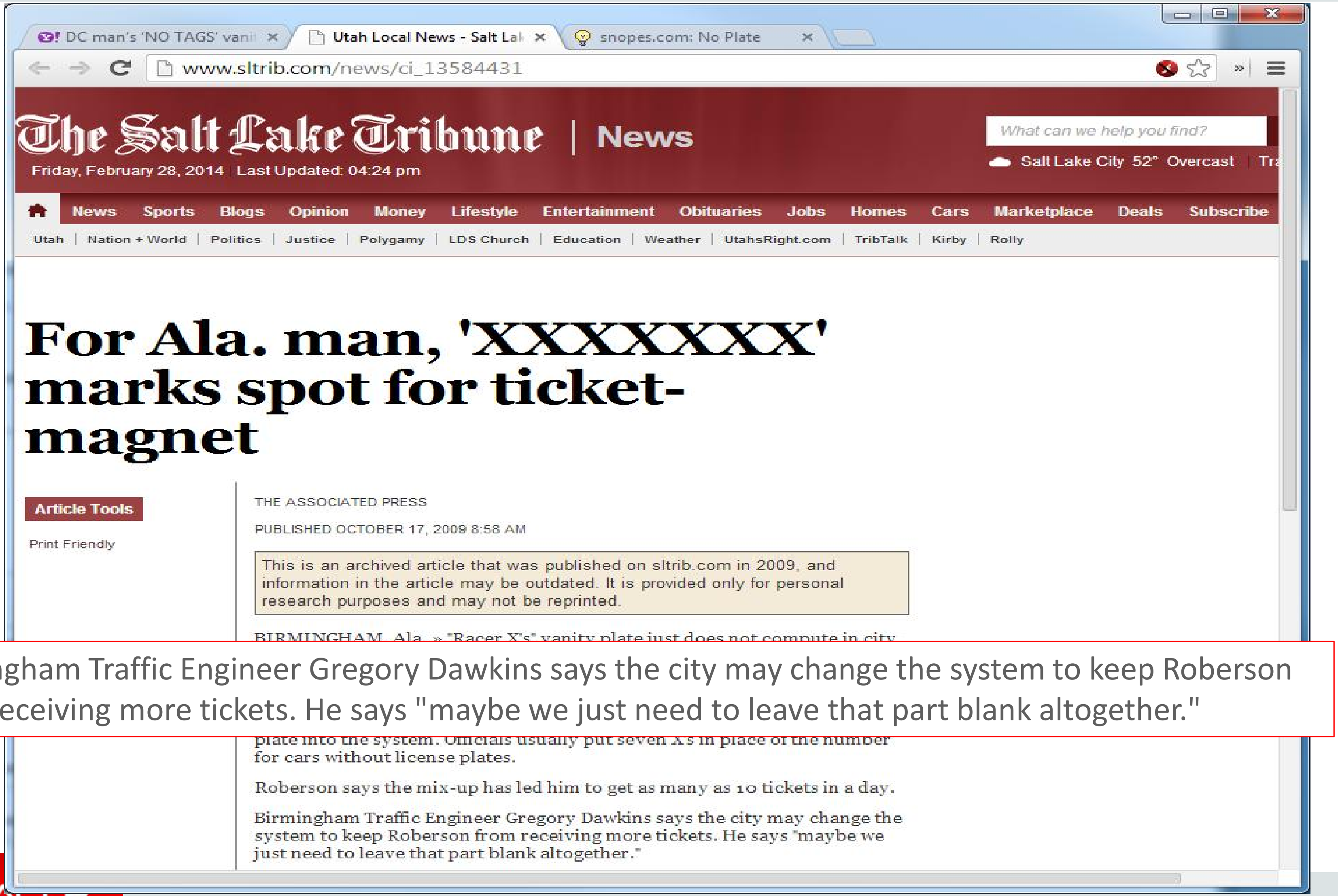
The Sideshow

DC man's 'NO TAGS' vanity plate earns him $20,000 in tickets

By Eric Pfeiffer
February 16, 2012 7:09 PM
The Sideshow

Washington, D.C. driver Danny White thought he had a really good idea for a joke. But the joke's on him--to the tune of $20,000, reports local affiliate NBC4.

White's prank started 25 years ago when he got a vanity license plate reading, "NO TAGS." He told NBC4


DC man's 'NO TAGS' vani
Utah Local News - Salt Lal
snopes.com: No Plate
www.sltrib.com/news/ci_13584431
What can we help you find?
Salt Lake City 52° Overcast
News Sports Blogs Opinion Money Lifestyle Entertainment Obituaries Jobs Homes Cars Marketplace Deals Subscribe
Utah | Nation + World | Politics | Justice | Polygamy | LDS Church | Education | Weather | UtahsRight.com | TribTalk | Kirby | Rolly
Article Tools
Print Friendly

The Salt Lake Tribune | News

Friday, February 28, 2014 Last Updated: 04:24 pm

For Ala. man, 'XXXXXXX' marks spot for ticket-magnet

THE ASSOCIATED PRESS

PUBLISHED OCTOBER 17, 2009 8:58 AM

This is an archived article that was published on sltrib.com in 2009, and information in the article may be outdated. It is provided only for personal research purposes and may not be reprinted.

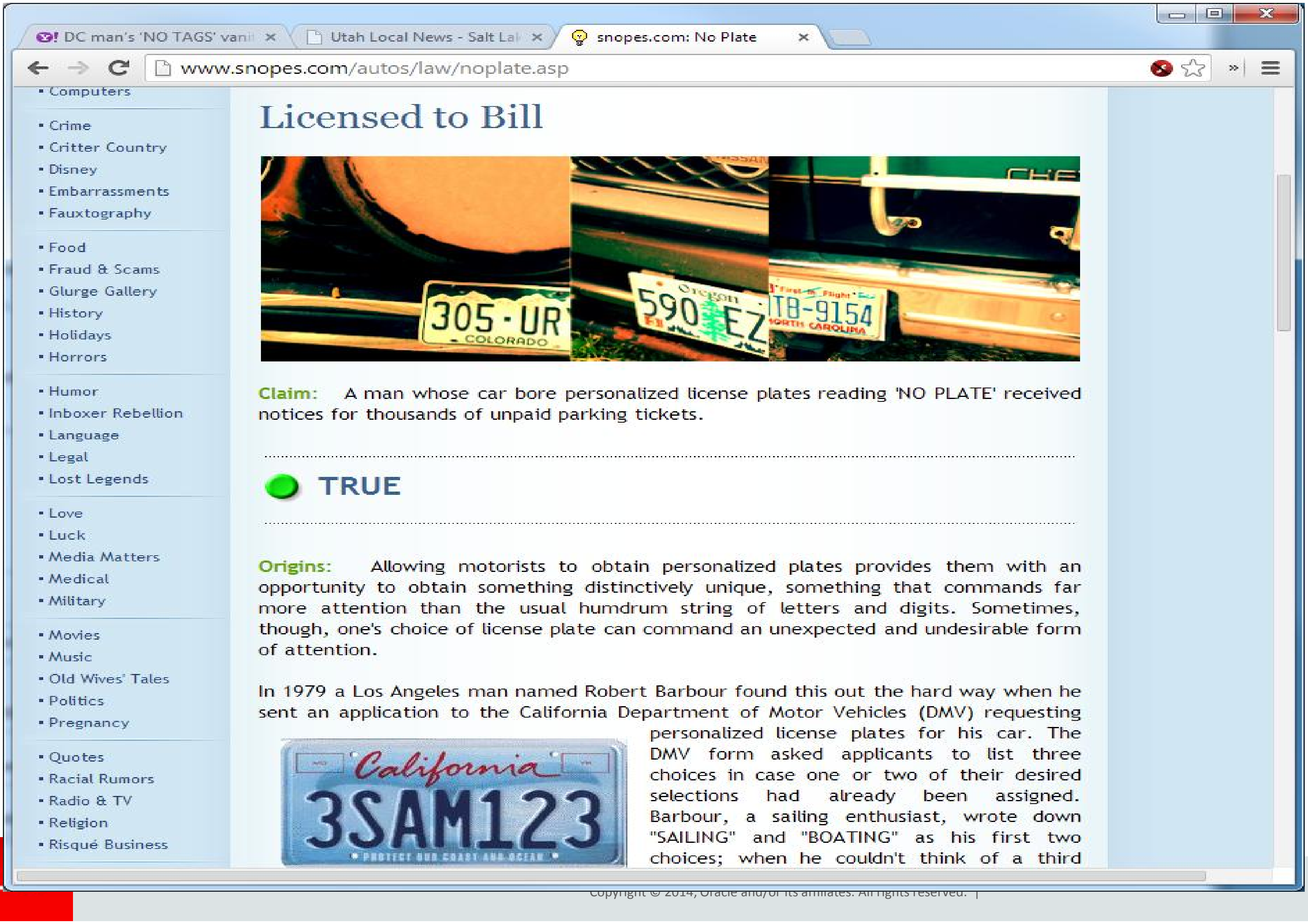
BIRMINGHAM, Ala. » "Racer X's" vanity plate just does not compute in city

plate into the system. Officials usually put seven X's in place of the number for cars without license plates.

Roberson says the mix-up has led him to get as many as 10 tickets in a day.

Birmingham Traffic Engineer Gregory Dawkins says the city may change the system to keep Roberson from receiving more tickets. He says "maybe we just need to leave that part blank altogether."

Birmingham Traffic Engineer Gregory Dawkins says the city may change the system to keep Roberson from receiving more tickets. He says "maybe we just need to leave that part blank altogether."



DC man's 'NO TAGS' vani
Utah Local News - Salt Lal
snopes.com: No Plate
www.snopes.com/autos/law/noplate.asp

Licensed to Bill

Claim: A man whose car bore personalized license plates reading 'NO PLATE' received notices for thousands of unpaid parking tickets.

● **TRUE**

Origins: Allowing motorists to obtain personalized plates provides them with an opportunity to obtain something distinctively unique, something that commands far more attention than the usual humdrum string of letters and digits. Sometimes, though, one's choice of license plate can command an unexpected and undesirable form of attention.

In 1979 a Los Angeles man named Robert Barbour found this out the hard way when he sent an application to the California Department of Motor Vehicles (DMV) requesting personalized license plates for his car. The DMV form asked applicants to list three choices in case one or two of their desired selections had already been assigned. Barbour, a sailing enthusiast, wrote down "SAILING" and "BOATING" as his first two choices; when he couldn't think of a third

Self Learning…

```
tkyte%ORA12C> select /*+ gather_plan_statistics */ count(*)
  2     from cities_state
  3    where name = 'New York city'
  4      and state = 'New York'
  5   /

  COUNT(*)
----------
   8175132
```

```
-------------------------------------------------------------
| Id  | Operation          | Name         | E-Rows | A-Rows |
-------------------------------------------------------------
|   0 | SELECT STATEMENT   |              |        |      1 |
|   1 |  SORT AGGREGATE    |              |      1 |      1 |
|*  2 |   TABLE ACCESS FULL| CITIES_STATE |   3398 |   8175K|
-------------------------------------------------------------

Predicate Information (identified by operation id):
----------------------------------------------------

   2 - filter(("NAME"='New York city' AND "STATE"='New York'))
```

```
tkyte%ORA12C> select /*+ gather_plan_statistics */ count(*)
  2      from cities_state
  3     where name = 'New York city'
  4       and state = 'New York'
  5  /

  COUNT(*)
----------
   8175132
```

```
-------------------------------------------------------------
| Id  | Operation          | Name          | E-Rows | A-Rows |
-------------------------------------------------------------
|   0 | SELECT STATEMENT   |               |        |      1 |
|   1 |  SORT AGGREGATE    |               |      1 |      1 |
|*  2 |   TABLE ACCESS FULL| CITIES_STATE  |   8175K|   8175K|
-------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------


   2 - filter(("NAME"='New York city' AND "STATE"='New York'))

Note
-----
   - statistics feedback used for this statement
```

```
tkyte%ORA12C> exec dbms_spd.flush_sql_plan_directive;

PL/SQL procedure successfully completed.

tkyte%ORA12C> select o.object_name, o.subobject_name col_name,
                     o.object_type, d.type, d.state, d.reason
  2    from dba_sql_plan_directives d, dba_sql_plan_dir_objects o
  3   where d.directive_id = o.directive_id
  4     and o.owner = user
  5   order by 1, 2, 3, 4, 5;

OBJECT_NAME  COL_NAME   OBJECT TYPE             STATE      REASON
------------ ---------- ------ ---------------- ---------- ------------------------------------
CITIES_STATE NAME       COLUMN DYNAMIC_SAMPLING USABLE     SINGLE TABLE CARDINALITY MISESTIMATE
CITIES_STATE STATE      COLUMN DYNAMIC_SAMPLING USABLE     SINGLE TABLE CARDINALITY MISESTIMATE
CITIES_STATE            TABLE  DYNAMIC_SAMPLING USABLE     SINGLE TABLE CARDINALITY MISESTIMATE
```

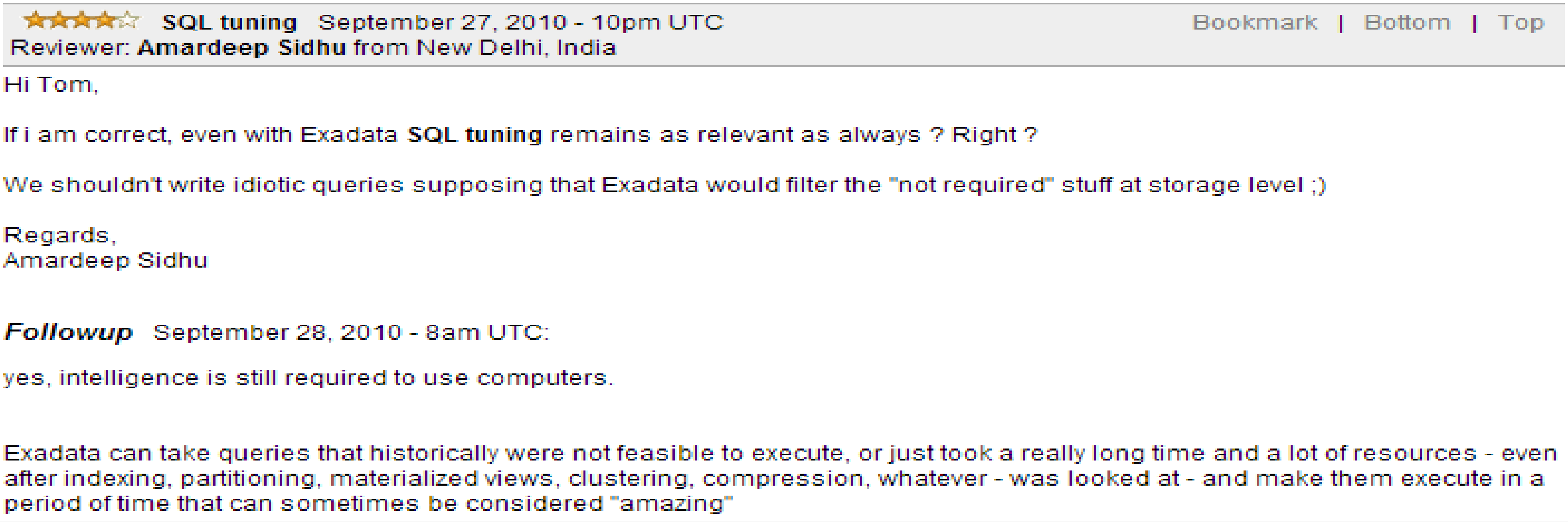
```
tkyte%ORA12C> select /*+ gather_plan_statistics */ count(*)
  2    from cities_state
  3   where name = 'Los Angeles city'
  4     and state = 'California'
  5  /

  COUNT(*)
----------
   3792620
```

```
-------------------------------------------------------------
| Id  | Operation          | Name         | E-Rows | A-Rows |
-------------------------------------------------------------
|   0 | SELECT STATEMENT   |              |        |      1 |
|   1 |  SORT AGGREGATE    |              |      1 |      1 |
|*  2 |   TABLE ACCESS FULL| CITIES_STATE |   5196K|   3792K|
-------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - filter(("NAME"='Los Angeles city' AND "STATE"='California'))

Note
-----
   - dynamic statistics used: dynamic sampling (level=2)
   - 1 Sql Plan Directive used for this statement
```

```
tkyte%ORA12C> select column_name, num_distinct, histogram
  2     from user_tab_col_statistics
  3    where table_name = 'CITIES_STATE';

COLUMN_NAME NUM_DISTINCT HISTOGRAM
----------- ------------ ---------------
NAME                 675 NONE
STATE                 50 NONE
```

```
tkyte%ORA12C> exec dbms_stats.gather_table_stats(user,'CITIES_STATE');

PL/SQL procedure successfully completed.

tkyte%ORA12C> select column_name cname, num_distinct, histogram
  2    from user_tab_col_statistics
  3   where table_name = 'CITIES_STATE';

CNAME                          NUM_DISTINCT HISTOGRAM
------------------------------ ------------ ---------------
NAME                                    675 HYBRID
STATE                                    50 FREQUENCY
SYS_STS652AVX5KJJE5OOY9V6#UOGP          714 HYBRID
```

But Still



★★★★☆ **SQL tuning** September 27, 2010 - 10pm UTC

Reviewer: **Amardeep Sidhu** from New Delhi, India

Hi Tom,

If i am correct, even with Exadata **SQL tuning** remains as relevant as always ? Right ?

We shouldn't write idiotic queries supposing that Exadata would filter the "not required" stuff at storage level ;)

Regards,
Amardeep Sidhu

Followup September 28, 2010 - 8am UTC:

yes, intelligence is still required to use computers.

Exadata can take queries that historically were not feasible to execute, or just took a really long time and a lot of resources - even after indexing, partitioning, materialized views, clustering, compression, whatever - was looked at - and make them execute in a period of time that can sometimes be considered "amazing"

Transforming so you don't have to…

Subquery Unnesting

A correlated subquery is one that refers to a column from a table outside the subquery

In this case C.cust_id is referenced in the subquery

Without subquery unnesting the correlated subquery must be evaluated for each row in the Customers tables

```
SELECT C.cust_last_name, C.country_id
  FROM  customers C
 WHERE Exists(SELECT null
               FROM sales S
              WHERE C.cust_id=S.cust_id
                AND s.quantity_sold > 1000);
```

Subquery Unnesting

Query gets transformed as follows

Transformation rewrites the EXISTS subquery to an ANY subquery

ANY subquery is no longer correlated

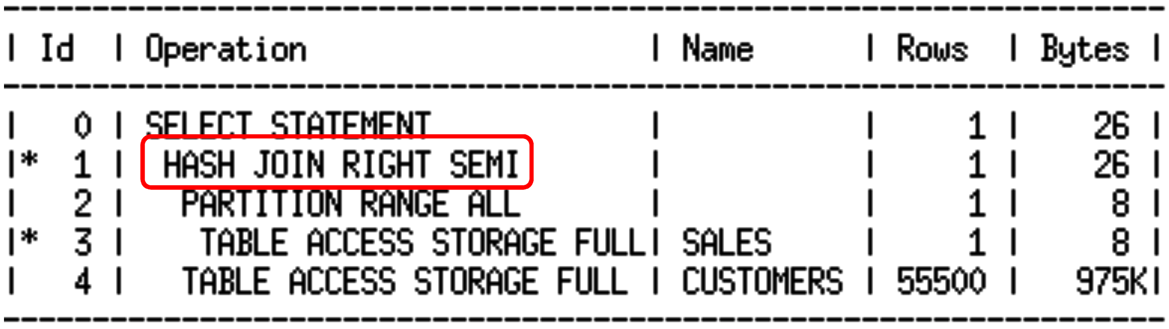
Any subquery returns a set of cust_id if any match the predicate will return true

```
SELECT C.cust_last_name, C.country_id
  FROM customers C
 WHERE C.cust_id =
           ANY(SELECT S.cust_id
                 FROM sales S
                WHERE S.quantity_sold > 1000);
```

Subquery Unnesting

Transformation allows subquery to be evaluated as a SEMI join

Subquery returns a set of cust_id and those cust_id are joined to the customers table via a SEMI Hash Join



```
------------------------------------------------------------------
| Id  | Operation                    | Name      | Rows  | Bytes |
------------------------------------------------------------------
|   0 | SELECT STATEMENT             |           |     1 |    26 |
|*  1 |  HASH JOIN RIGHT SEMI        |           |     1 |    26 |
|   2 |   PARTITION RANGE ALL        |           |     1 |     8 |
|*  3 |    TABLE ACCESS STORAGE FULL| SALES     |     1 |     8 |
|   4 |   TABLE ACCESS STORAGE FULL  | CUSTOMERS | 55500 |   975K|
------------------------------------------------------------------
```

Complex View Merging

Complex view merging refers to the merging of group by and distinct views

Allows the optimizer to consider additional join orders and access paths

Group-by/distinct operations can be delayed until after the joins have been evaluated

```
Create View cust_prod_totals_v as
Select sum(s.quantity_sold) total, s.cust_id, s.prod_id
  from sales s
 group by s.cust_id, s.prod_id;


Select c.cust_id, c.cust_first_name, c.cust_last_name
  from customers c, cust_prod_totals_v v, products p
 where c.country_id = 'US'
   and c.cust_id = v.cust_id
   and v.total > 100
   and v.prod_id = p.prod_id
   and p.prod_name = 'T3 Faux Fur-Trimmed Sweater';
```

Complex View Merging

Query gets transformed as follows

After transformation group by operation occurs after Sales is joined to Products and Customers

Number of rows in group by greatly reduced after join

May not always be best to delay the group by / distinct operation so transformation is cost based*

```
select c.cust_id, cust_first_name,
cust_last_name
  from customers c, products p, sales s
 where c.country_id = 'US'
   and c.cust_id = s.cust_id
   and s.prod_id = p.prod_id
   and p.prod_name = 'T3 Faux Fur-Trimme…’
  group by s.cust_id, s.prod_id, p.rowid,
        c.rowid, c.cust_last_name,
        c.cust_first_name, c.cust_id
 having sum(s.quantity_sold) > 100;
```

*Compare the cost of the best plan with and without the transformation

OR Expansion

Transforms queries that contain OR predicates into the form of a UNION ALL query of two or more branches

Without the transformation Optimizer treats OR predicates as a single unit

Can’t use index on either column

```
SELECT *
  FROM products
 WHERE prod_category ='Photo'
    OR prod_subcategory ='Camera Media';
```

OR Expansion

Query gets transformed as follows

The transformation adds an LNNVL() function to the second branch in order to avoid duplicates being generated across branches

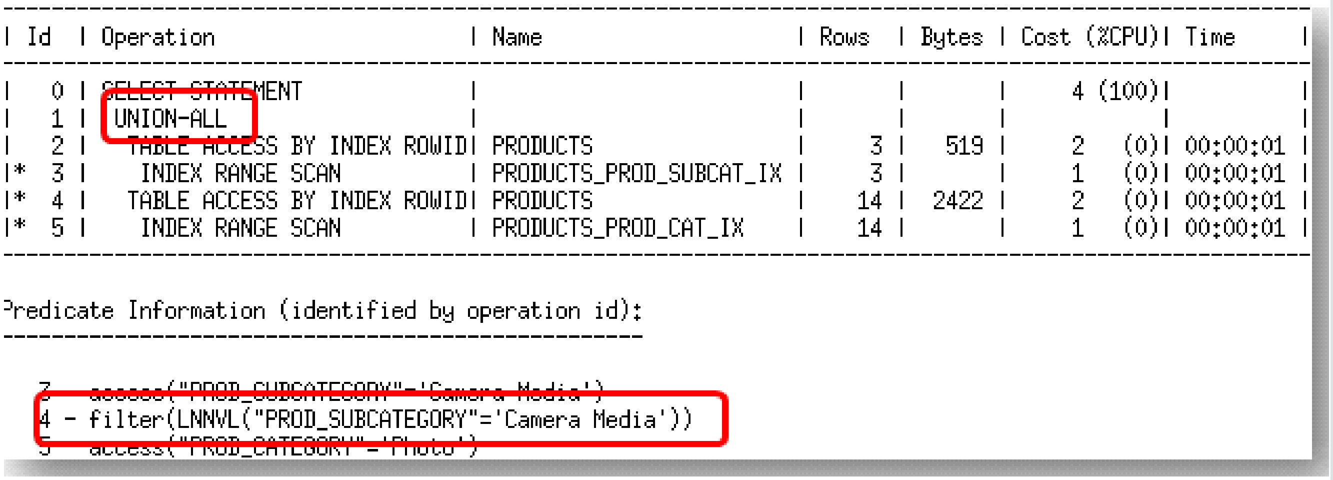
The LNNVL function returns TRUE, if the predicate evaluates to FALSE or if the predicate involves NULL; otherwise it will return FALSE

```
SELECT *
  FROM products
 WHERE prod_subcategory ='Camera Media‘
 UNION ALL
SELECT *
  FROM products
 WHERE prod_category ='Photo'
   AND lnnvl(prod_subcategory =
                    'Camera Media');
```

**lnnvl(true) is FALSE, lnnvl(false||null) is TRUE*

OR Expansion

Transformation allows an index access to be considered for each branch of the UNION ALL



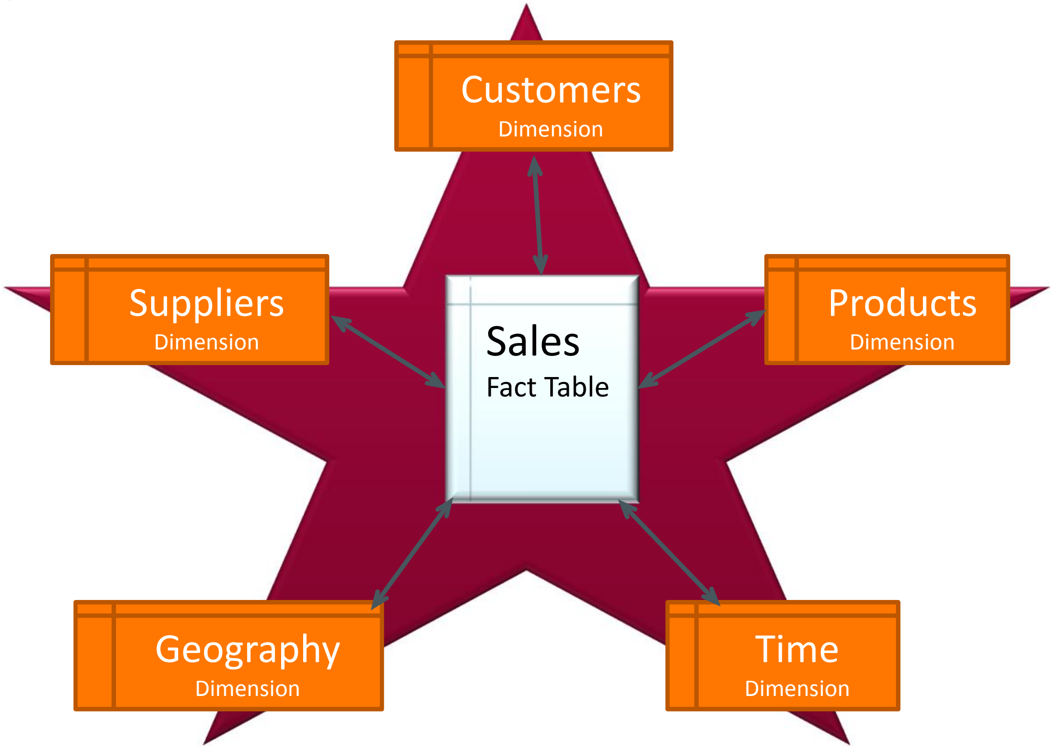
```
---------------------------------------------------------------------------------------------------
| Id  | Operation                    | Name                     | Rows  | Bytes | Cost (%CPU)| Time     |
---------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |                          |       |       |     4 (100)|          |
|   1 |  UNION-ALL                   |                          |       |       |            |          |
|   2 |   TABLE ACCESS BY INDEX ROWID| PRODUCTS                 |     3 |   519 |     2   (0)| 00:00:01 |
|*  3 |    INDEX RANGE SCAN          | PRODUCTS_PROD_SUBCAT_IX  |     3 |       |     1   (0)| 00:00:01 |
|*  4 |   TABLE ACCESS BY INDEX ROWID| PRODUCTS                 |    14 |  2422 |     2   (0)| 00:00:01 |
|*  5 |    INDEX RANGE SCAN          | PRODUCTS_PROD_CAT_IX     |    14 |       |     1   (0)| 00:00:01 |
---------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   3 - access("PROD_SUBCATEGORY"='Camera Media')
   4 - filter(LNNVL("PROD_SUBCATEGORY"='Camera Media'))
   5 - access("PROD_CATEGORY"='Photo')
```

Star Query Transformation

Cost-based* transformation designed to execute star queries more efficiently

Relies on bitmap indexes on foreign key columns to access rows in the fact table

Controlled by parameter
STAR_TRANSFORMATION_ENABLED



Customers
Dimension

Suppliers
Dimension

Sales
Fact Table

Products
Dimension

Geography
Dimension

Time
Dimension

Star Schema - one or more large fact table and many smaller dimension tables

*Compares the cost of the best plan with and without the transformation

Star Query Transformation

Traditionally a star query only defines predicates on the dimension tables

No efficient way to access rows in the fact table

By rewriting the query new access paths become available on the fact table

```
SELECT *
  FROM fact, d1, d2, d3
 WHERE fact.c1 = d1.c1
   AND fact.c2 = d2.c1
   AND fact.c3 = d3.c1
   AND d1.c2 IN (1, 2, 3, 4)
   AND d2.c2 < 100
   AND d3.c2 = 35;
```

Star Query Transformation

Query gets transformed as follows

Converts original query to include 3 sub-queries on the fact

Fact table accessed first via bitmap index and then joins out to dimension tables

Result of sub-queries may be saved in temp tables

```
SELECT *
  FROM    fact, d1, d2, d3
… exactly as before …
   AND fact.c1 IN (SELECT d1.c1
                     FROM d1
                    WHERE d1.c2 IN (1, 2, 3, 4) )
   AND fact.c2 IN (select d2.c1
                     FROM d2
                    WHERE d2.c2 < 100)
   AND fact.c3 IN (SELECT d3.c1
                     FROM d3
                    WHERE d3.c2 = 35)
```

The optimizer is getting smarter than we are…

```
ops$tkyte%ORA11GR2> CREATE TABLE T1
  2  (
  3    ORDER_ID           NUMBER(18)    NOT NULL,
  4    ACCOUNT_NO         NUMBER(10)    NOT NULL,
  5    ORDER_NUMBER       VARCHAR2(20)  NOT NULL,
  6    data               varchar2(1000)
  7  );
Table created.

ops$tkyte%ORA11GR2> ALTER TABLE T1 ADD CONSTRAINT T1_PK1 PRIMARY KEY (ORDER_ID);
Table altered.
```

The optimizer is getting smarter than we are…

```
ops$tkyte%ORA11GR2> CREATE TABLE T2
  2  (
  3    SERVICE_ORDER_ID          NUMBER(18)          NOT NULL,
  4    ORDER_ID                  NUMBER(18)          NOT NULL,
  5    ORDER_STATUS_ID           NUMBER(6)           NOT NULL,
  6    data                      varchar2(1000)
  7  );
Table created.

ops$tkyte%ORA11GR2> ALTER TABLE T2 ADD CONSTRAINT T2_PK1
  2  PRIMARY KEY (SERVICE_ORDER_ID);
Table altered.

ops$tkyte%ORA11GR2> ALTER TABLE T2 ADD CONSTRAINT T2_OSO_FK1
  2  FOREIGN KEY (ORDER_ID)  REFERENCES T1 (ORDER_ID);
Table altered.
```

The optimizer is getting smarter than we are…

```
ops$tkyte%ORA11GR2> CREATE TABLE T3
  2  (
  3    SERVICE_ORDER_ID               NUMBER(18)      NOT NULL,
  4    RELATED_SERVICE_ORDER_ID       NUMBER(18),
  5    data                           varchar2(1000)
  6  );
Table created.

ops$tkyte%ORA11GR2> ALTER TABLE T3 ADD CONSTRAINT T3_ORDER_PK1
  2  PRIMARY KEY (SERVICE_ORDER_ID);
Table altered.

ops$tkyte%ORA11GR2> ALTER TABLE T3 ADD CONSTRAINT T3_OLS_S_FK1
  2  FOREIGN KEY (SERVICE_ORDER_ID)  REFERENCES T2 (SERVICE_ORDER_ID);
Table altered.

ops$tkyte%ORA11GR2> CREATE INDEX T3_OLS_RS_1
  2  ON T3 (RELATED_SERVICE_ORDER_ID);
Index created.
```

The optimizer is getting smarter than we are…

```
ops$tkyte%ORA11GR2> SELECT COUNT(*)
  2    FROM T1, T2, T3
  3   WHERE T2.order_id = T1.order_id
  4     AND T2.service_order_id = T3.service_order_id (+)
  5     AND T3.related_service_order_id = TO_NUMBER(:v0);

---------------------------------------------------------------------------------
| Id  | Operation         | Name        | Rows  | Bytes | Cost (%CPU)| Time     |
---------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |             |     1 |    26 |     1   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE   |             |     1 |    26 |            |          |
|*  2 |   INDEX RANGE SCAN| T3_OLS_RS_1 |     1 |    26 |     1   (0)| 00:00:01 |
---------------------------------------------------------------------------------
```

The optimizer is getting smarter than we are…

```
ops$tkyte%ORA11GR2> set autotrace traceonly explain
ops$tkyte%ORA11GR2> SELECT COUNT(*)
  2    FROM T1, T2, T3
  3   WHERE T2.order_id = T1.order_id
  4     AND T2.service_order_id = T3.service_order_id (+)
  5     AND T3.related_service_order_id = TO_NUMBER(:v0);
```

```
ops$tkyte%ORA11GR2> set autotrace traceonly explain
ops$tkyte%ORA11GR2> SELECT COUNT(*)
  2    FROM T1, T2, T3
  3   WHERE T2.order_id = T1.order_id
  4     AND T2.service_order_id = T3.service_order_id
  5     AND T3.related_service_order_id = TO_NUMBER(:v0);
```

- First, it knows the outer join is not necessary
  - Where t2.col = t3.col(+) and t3.anything = 'something'
  - Implies the (+) is not necessary
    - If the outer join 'happened', then t3.anything would be NULL! And t3.anything = to_number(:v0) would never be satisfied

The optimizer is getting smarter than we are…

```
ops$tkyte%ORA11GR2> set autotrace traceonly explain
ops$tkyte%ORA11GR2> SELECT COUNT(*)
  2    FROM T1, T2, T3
  3   WHERE T2.order_id = T1.order_id
  4     AND T2.service_order_id = T3.service_order_id (+)
  5     AND T3.related_service_order_id = TO_NUMBER(:v0);
```

```
ops$tkyte%ORA11GR2> set autotrace traceonly explain
ops$tkyte%ORA11GR2> SELECT COUNT(*)
  2    FROM T2, T3
  3   WHERE T2.service_order_id = T3.service_order_id
  4     AND T3.related_service_order_id = TO_NUMBER(:v0);
```

- Second, it knows that T1 is not relevant to the query
  - Nothing is selected from T1 in the output
  - T1(order_id) is the primary key, joined to T2(order_id) – so T2 is “key preserved”
  - T2(order_id) is NOT NULL and is a foreign key to T1
  - Therefore, when you join T1 to T2 – every row in T2 appears at least once and at most once in the output

The optimizer is getting smarter than we are…

```
ops$tkyte%ORA11GR2> set autotrace traceonly explain
ops$tkyte%ORA11GR2> SELECT COUNT(*)
  2    FROM T1, T2, T3
  3   WHERE T2.order_id = T1.order_id
  4     AND T2.service_order_id = T3.service_order_id (+)
  5     AND T3.related_service_order_id = TO_NUMBER(:v0);
```

```
ops$tkyte%ORA11GR2> set autotrace traceonly explain
ops$tkyte%ORA11GR2> SELECT COUNT(*)
  2    FROM T3
  3   WHERE T3.related_service_order_id = TO_NUMBER(:v0);
```

- Lastly, it knows that T2 is not relevant to the query
  - Nothing is selected from T2 in the output
  - T2(service_order_id) is the primary key, joined to T3(service_order_id) – so T3 is “key preserved”
  - T3(service_order_id) is NOT NULL and is a foreign key to T2
  - Therefore, when you join T2 to T3 – every row in T3 appears at least once and at most once in the output

The optimizer is getting smarter than we are…

```
ops$tkyte%ORA11GR2> SELECT COUNT(*)
  2    FROM T1, T2, T3
  3   WHERE T2.order_id = T1.order_id
  4     AND T2.service_order_id = T3.service_order_id (+)
  5     AND T3.related_service_order_id = TO_NUMBER(:v0);
```

Is the same as…. But only because of the constraints in place…

```
ops$tkyte%ORA11GR2> SELECT COUNT(*)
  2    FROM T3
  3   WHERE T3.related_service_order_id = TO_NUMBER(:v0);
```

Not Necessarily SQL, but think about the coolest, most underutilized feature (that has been around for over five years!)

Quick Story…

I’m going to write a PL/SQL parser…

```
TKYTE%ORA12C> create user demo
  2     identified by demo;

User created.

TKYTE%ORA12C> grant create session,
  2        create procedure
  3     to demo;

Grant succeeded.

TKYTE%ORA12C> create edition version2
  2      as child of ora$base;

Edition created.
```

```
TKYTE%ORA12C> connect demo/demo
Connected.

DEMO%ORA12C> create or replace
  2  procedure my_procedure
  3  as
  4  begin
  5     dbms_output.put_line( 'I am buggy version 1.0' );
  6  end;
  7  /
Procedure created.

DEMO%ORA12C> create or replace
  2  procedure my_procedure2
  3  as
  4  begin
  5     my_procedure;
  6  end;
  7  /
Procedure created.
```

```
DEMO%ORA12C> exec my_procedure2
I am buggy version 1.0

PL/SQL procedure successfully completed.
```

```
DEMO%ORA12C> alter session
  2    set edition = version2;
ERROR:
ORA-38802: edition does not exist
```

```
DEMO%ORA12C> connect tkyte/tkyte
Connected.

TKYTE%ORA12C> alter user demo
  2    enable editions;
User altered.

TKYTE%ORA12C> grant use
  2   on edition version2
  3   to demo;
Grant succeeded.

TKYTE%ORA12C> grant use
  2   on edition version2
  3   to scott;
Grant succeeded.
```

```
TKYTE%ORA12C> connect demo/demo
Connected.

DEMO%ORA12C> alter session
  2  set edition = version2;
Session altered.

DEMO%ORA12C> select object_name, object_type, status, edition_name
  2    from user_objects;

OBJECT_NAME                    OBJECT_TYPE STATUS EDITION_NAME
------------------------------ ----------- ------ ------------
MY_PROCEDURE                   PROCEDURE   VALID  ORA$BASE
MY_PROCEDURE2                  PROCEDURE   VALID  ORA$BASE
```

```
DEMO%ORA12C> create or replace
  2  procedure my_procedure
  3  as
  4  begin
  5      dbms_output.put_line( 'I am fixed in version 2.0' );
  6  end;
  7  /

Procedure created.
```

```
DEMO%ORA12C> select object_name, edition_name
  2    from user_objects;

OBJECT_NAME                    EDITION_NAME
------------------------------ ------------
MY_PROCEDURE2                  ORA$BASE
MY_PROCEDURE                   VERSION2

DEMO%ORA12C> select object_name, edition_name
  2    from user_objects_AE;

OBJECT_NAME                    EDITION_NAME
------------------------------ ------------
MY_PROCEDURE                   ORA$BASE
MY_PROCEDURE2                  ORA$BASE
MY_PROCEDURE                   VERSION2
```

```
DEMO%ORA12C> grant execute  on my_procedure2 to scott;
Grant succeeded.

DEMO%ORA12C> select object_name, edition_name from user_objects;

OBJECT_NAME                    EDITION_NAME
------------------------------ ------------
MY_PROCEDURE2                  VERSION2
MY_PROCEDURE                   VERSION2

DEMO%ORA12C> select object_name, edition_name from user_objects_AE;

OBJECT_NAME                    EDITION_NAME
------------------------------ ------------
MY_PROCEDURE2                  ORA$BASE
MY_PROCEDURE                   ORA$BASE
MY_PROCEDURE2                  VERSION2
MY_PROCEDURE                   VERSION2
```

```
DEMO%ORA12C> SELECT SYS_CONTEXT ('userenv', 'current_edition_name') sc
  2     FROM DUAL;

SC
---------------
VERSION2

DEMO%ORA12C> exec my_procedure2
I am fixed in version 2.0

PL/SQL procedure successfully completed.
```

```
DEMO%ORA12C> connect demo/demo
Connected.

DEMO%ORA12C> SELECT SYS_CONTEXT ('userenv', 'current_edition_name') sc
  2    FROM DUAL;

SC
---------------
ORA$BASE

DEMO%ORA12C> exec my_procedure2
I am buggy version 1.0

PL/SQL procedure successfully completed.
```

```
DEMO%ORA12C> alter session set edition = version2;

Session altered.

DEMO%ORA12C> SELECT SYS_CONTEXT ('userenv', 'current_edition_name') sc
  2    FROM DUAL;

SC
---------------
VERSION2

DEMO%ORA12C> exec my_procedure2
I am fixed in version 2.0

PL/SQL procedure successfully completed.
```

```
DEMO%ORA12C> connect scott/tiger
Connected.

SCOTT%ORA12C> SELECT SYS_CONTEXT ('userenv', 'current_edition_name') sc
  2    FROM DUAL;

SC
---------------
ORA$BASE

SCOTT%ORA12C> exec demo.my_procedure2
BEGIN demo.my_procedure2; END;

      *
ERROR at line 1:
ORA-06550: line 1, column 7:
PLS-00201: identifier 'DEMO.MY_PROCEDURE2' must be declared
ORA-06550: line 1, column 7:
PL/SQL: Statement ignored
```

```
SCOTT%ORA12C> alter session
  2    set edition = version2;

Session altered.

SCOTT%ORA12C> exec demo.my_procedure2
I am fixed in version 2.0

PL/SQL procedure successfully completed.
```

ORACLE®

ORA-03113

end-of-file on communication channel