How Do You Know What You Know...

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

What we know, shapes how we do things...
New Ideas – Almost always come from those uncommitted to the status quo – from outsiders

People are committed to whatever has worked for them in the past.
“John Harrison was a clockmaker without any formal schooling or training. The son of a carpenter, he grew up in a small village far from London, Oxford or Cambridge. In 1727, began to work on the problem of determining longitude.”

Nevil Maskelyne
Arthur Fry
Church Choir Singer

“I need to bookmark my hymnal, but regular paper just falls out”
“Swiss Watches were the gold standard for watches. People will not want a watch with a microchip, they want it the way it has always been. **Well, not until the new watch costs less than $10**”
Educated Incapacity – A barrier to creative ideas can be experience.

“There is no reason for any individual to have a computer in their home.” – Ken Olson, DEC, 1977

Although experience is often valuable, it can be a liability in a search for creative ideas.
SQL> select dt, val
     2  from t
     3  order by
     4  dt;

<table>
<thead>
<tr>
<th>DT</th>
<th>VAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-AUG-07</td>
<td>232</td>
</tr>
<tr>
<td>02-AUG-07</td>
<td></td>
</tr>
<tr>
<td>03-AUG-07</td>
<td>228</td>
</tr>
<tr>
<td>04-AUG-07</td>
<td></td>
</tr>
<tr>
<td>05-AUG-07</td>
<td>213</td>
</tr>
<tr>
<td>06-AUG-07</td>
<td></td>
</tr>
</tbody>
</table>
How can you carry the last non-null value down?
SQL> select dt, val,
2          case when val is not null
3          then to_char(row_number() over (order by dt),'fm0000')||val
4          end max_val
5          from t
6          order by dt;

DT          VAL  MAX_VAL
---------  -------- ---------------
01-AUG-07   232 0001232
02-AUG-07
03-AUG-07   228 0003228
04-AUG-07
05-AUG-07   213 0005213
06-AUG-07
```sql
SQL> select dt, val,
    2       max(max_val) over (order by dt) max_val_str
    3  from ( select dt, val,
    4       case when val is not null
    5          then to_char(row_number() over (order by dt),'fm0000')||val
    6       end max_val
    7  from t ) order by dt;

DT               VAL MAX_VAL_STR
--------- ---------- ---------------------------------------------
01-AUG-07        232 0001232
02-AUG-07            0001232
03-AUG-07        228 0003228
04-AUG-07            0003228
05-AUG-07        213 0005213
06-AUG-07            0005213
```
SQL> select dt, val,
       2    to_number(substr(max(max_val) over (order by dt),5)) max_val
       3  from ( select dt, val,
       4         case when val is not null
       5         then to_char(row_number() over (order by dt),'fm0000')||val
       6         end max_val
       7  from t ) order by dt;

   DT        VAL    MAX_VAL
  --------- ---------- ----------
  01-AUG-07  232     232
  02-AUG-07                   232
  03-AUG-07  228     228
  04-AUG-07                   228
  05-AUG-07  213     213
  06-AUG-07                   213
SQL> select dt, val,
    2    last_value(val ignore nulls) over (order by dt) max_val
    3    from t
    4    order by dt;

DT             VAL      MAX_VAL
--------- ------- -------
01-AUG-07   232       232
02-AUG-07
03-AUG-07   228       228
04-AUG-07
05-AUG-07   213       213
06-AUG-07

“Dad, why can’t I see the pictures you took today?”

*His three year old daughter*

He didn’t let his experience cloud his creativity, *everyone knows you need a darkroom*...
Assumptions – Incorrect assumptions are barriers to creativity

Greetings card companies thought their competition was other card companies – but it wasn’t, it was FTD

What is a ‘requirement’ really? It sounds straightforward enough
Assumptions – Incorrect assumptions lead down the wrong roads

Where do you think they would put commit? Well, if you assume all application hangs are database problems, you might not even think about it.
Assumptions – Incorrect assumptions can be *harmful*

*When you assume the other person knows more than they know… When you assume a common terminology exists – ‘Save Changes’*

*When you assume something works the way you think it should work…*
Assumptions...

Oracle has a undo log pointer from the index tuple?? That really confuses me a little. I have referred to postgres source (8.3), … I believe an index structure should maintain all the versions of the index tuple in the b-tree itself. I am unable to think of an alternate design, where index tuple might need undo log pointer.

Continued…
Assumptions…

Actually Undo-log is used in oracle for two purposes - time-travel (to goto the previous version of records) and rollback. My last argument was that time-travel can't be achieved in the index through undo log. All the previous versions of a record has to be in the B-tree structure. Hence no need to store undo-log pointer for time travel purpose. If the statement needs more explanation, let me know.

And so I said…
Assumptions...

"You need to learn about Oracle then - multi-versioning, read consistency."

I have a fair idea on this. I just say that this can't work for index and it can only work for table.

......

• a) we build an index on 1, 2, 3, 4, .... N
• b) we measure it's size
• c) we delete everything from it
• d) we commit
• e) we populate the index again with new values 1.1, 2.1, ... (noting that index size stays same, therefore we MUST have overwritten everything ...)
• f) we commit
• g) we then offline the tablespaces for the table, we have access only to the index structure and run a query that uses the index and only the index AS OF (b) and a query that uses the index AS OF (f) and we see differing results. If that works – what then...
Assumptions...

Ambivalent

1: that does not matter one way or the other
2: of no importance or value one way or the other

1: simultaneous and contradictory attitudes or feelings (as attraction and repulsion) toward an object, person, or action
2: continual fluctuation (as between one thing and its opposite), uncertainty as to which approach to follow

How many times do you think misunderstood acronyms, internal abbreviations and nomenclature have gotten in your way...
Assumptions...

- It works the way it works.
- Shared Server
- Nulls – “I understand you and have no choice to accept that Status Quo, but the sense of my logic feels violated when I consider that the trivial query below, always returns nulls after the 'Z*'.
- Order by is the only way to – well – get ordered data **consistently**
Judgments – “We tried that once…” “It’ll never work…” “They will never buy that…”

Cars will never replace horses...  Birds were made to fly...  You’ll fall off the edge of the earth...
How to be creative…
How to be creative…

*First, Forget Everything You Know…*

Like Edwin Land did…
How to be creative…

Then, Remember Everything You Know…

You know a bunch of facts – you’ve let go of the preconceived relationships they had
How to be creative…

Then, Rearrange Everything You Know…

Edwin Land put the darkroom in the camera, instead of the camera in the darkroom… Out of the mouths of …
That was creative…

*How To Tune using the ‘Craig’ method…*
That was creative…

*Oracle is hanging… (but always at 7pm)*
That was creative…

**Uniquely index only ‘active’ records**

- **Fact:** Entirely NULL keys are not entered into a B*Tree Index
- **Fact:** We can index functions
- **Result:** `CREATE UNIQUE INDEX IDX ON TAB ( case when status = ‘ACTIVE’ then name end );`
But Hey, Let’s not get carried away…
But Hey, Let’s not get carried away…

Title: Light bulb changer

Document Type and Number: United States Patent 6826983

Filing Date: 02/10/2003
But Hey, Let’s not get carried away…

Title: Mouse device with a built-in printer

Document Type and Number: United States Patent 6650315

Filing Date: 05/21/2001
But Hey, Let’s not get carried away…

Title: Pedal operated mower

Document Type and Number: United States Patent 4455816

Filing Date: 06/29/1982
But Hey, Let’s not get carried away…

Title: Anti-eating face mask

Document Type and Number: United States Patent 4344424

Filing Date: 03/27/1980
So, What is the point?
DBA vs Developer
The Beginning...

- Data Model with Structure
- Data Independent of Code
- Set-oriented
- 1977 the work begins

“A Relational Model for Large Shared Databanks”
E.F. Codd - 1970
The Beginning...

When you were 12, did you know what you wanted to be when you grew up?...
Continuous Rethinking
Continuous Change
Putting it all Together
Questions?