ORACLE

Simplifying Your Data Audits With Oracle 11*g*'s Flashback Data Archive

Melanie Caffrey
Senior Development Manager, Unbreakable Linux Network, Oracle Linux

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 remain at the sole discretion of Oracle.

What Problem Are We Trying to Solve?

How Do You (unobtrusively) Meet Data Audit Requirements?

- May need to keep the history for a number of years
- Need a partitioning scheme (obviously)
- Want a compression scheme (for storage)
- How do you manage all of this cleanly without interfering (too much) in your user space?

But First ….. A Little History

A History of Flashback Introduction and Uses

- Version 3 (1983)
  - Provides multiple versions of data
- Version 4 (1984)
  - Provides read consistency
- Version 9 (2002)
  - Users get to start using Flashback for their own uses ….

A Quick Review of Other Flashback Technologies

- Flashback Query
  - select * from <table_name> as of timestamp to_timestamp …
- Flashback Version Query (10g 2003)
  - select <column_names>, versions_xid, versions_operation
    from <table_name>
    versions between timestamp to_timestamp …
    and to_timestamp …
- Flashback Transaction Query
  - select xid, logon_user, operation, undo_sql
    from flashback_transaction_query
    where xid in (select versions_xid from <table_name> …

A Quick Review of Other Flashback Technologies (continued)

- Flashback Transaction
  - DBMS_FLASHBACK.TRANSACTION_BACKOUT
- Flashback Table
  - Restore a table to a previous state
- Flashback Drop
  - Use the Recycle Bin to restore an erroneously dropped table
- Flashback Database
  - Return the database to a previous state

This All Sounds Great, So … Why Doesn't Everybody Use It?

Undo-based Flashback Drawbacks?

- Requires undo for *ALL* objects to be available
  - If you wanted to flashback 10 hours, 10 hours of undo must be online, no DBA would do that for 10 hours, let alone 10 years
- Not scalable
  - In order to flash back 1 hour, you rollback all changes to a block made in the last hour. 10 hours, all changes – one by one – for the last 10 hours. 5 years – it would never finish
- Not scalable Part 2
  - Undo for 5 years? For the entire database?

Undo-based Flashback Drawbacks?

- Typically not guaranteed – can be, but typically is not (and if guaranteed, can stop your database just like running out of archive space)
  - ORA-1555 snapshot too old, nothing you can do about it, not, therefore, suitable for auditing
- Would consume a ton of uncompressed, unpartitioned storage

Flashback Data Archive

- 11*g* introduced the new flashback technology, Flashback Data Archive, *aka* Oracle Total Recall
- What can it be used for?
  - Digital “Shredding”
  - Accessing Historical Data
  - Generating Reports
  - Auditing (our focus in this presentation)
  - Recovering Data (be aware that the SCN to Timestamp conversion data is still retained for only five days and the SCN conversion data is only good for +/- three seconds)

Flashback Data Archive Advantages

- Maintains data *only* for *certain* objects
  - Flashback Archive is implemented on a per table, not a per database, basis
- Very scalable
  - Instead of reading the undo per query, you can go directly to the archive and obtain the row, or sets of rows
- Very scalable Part 2
  - You are only limited by space needed for your retention requirements

Flashback Archive Advantages

- Is guaranteed
  - If enabled, will not overwrite undo needed for flashback archive purposes
- Provides partitioned storage
- If using EE and the Advanced Compression Option, then provides compressed storage, too

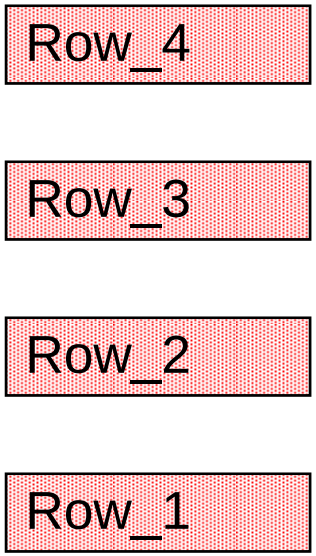
It’s (Almost) Free!

- Oracle Flashback Data Archive is no longer a priced option
- A feature of the Standard and Enterprise Editions
- Used to be part of the Oracle Total Recall option, then part of the Advanced Compression Option (ACO), but as of June 2013, anyone using Standard or Enterprise *(caveat explained shortly)*
- So, if you find your company’s data subject to more than a few data audits, it is well worth a look …

It’s (Almost) Free!

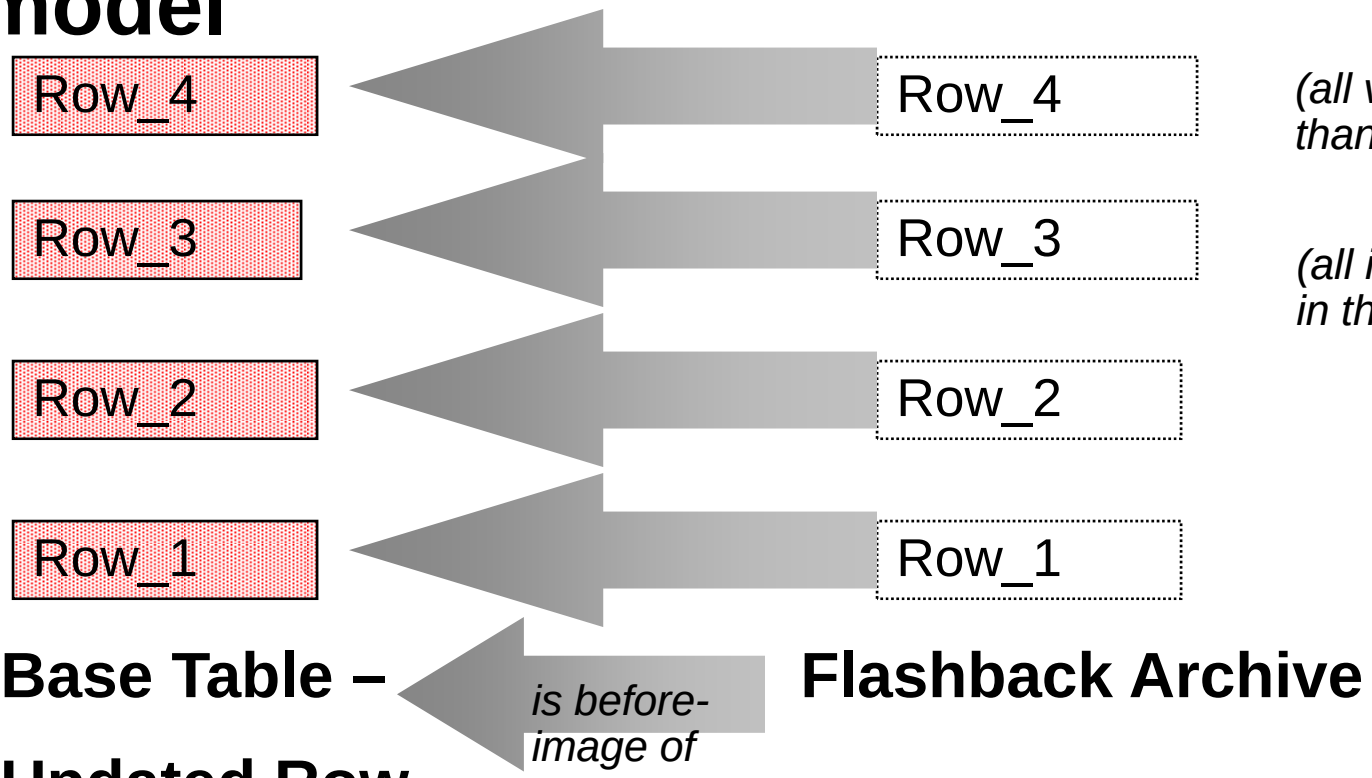
- Don’t Care About Compression?
  - FDA is available in Standard and Enterprise without Compression
- Otherwise …
  - Available in Enterprise with the Advanced Compression Option

Flashback Archive: implementation model

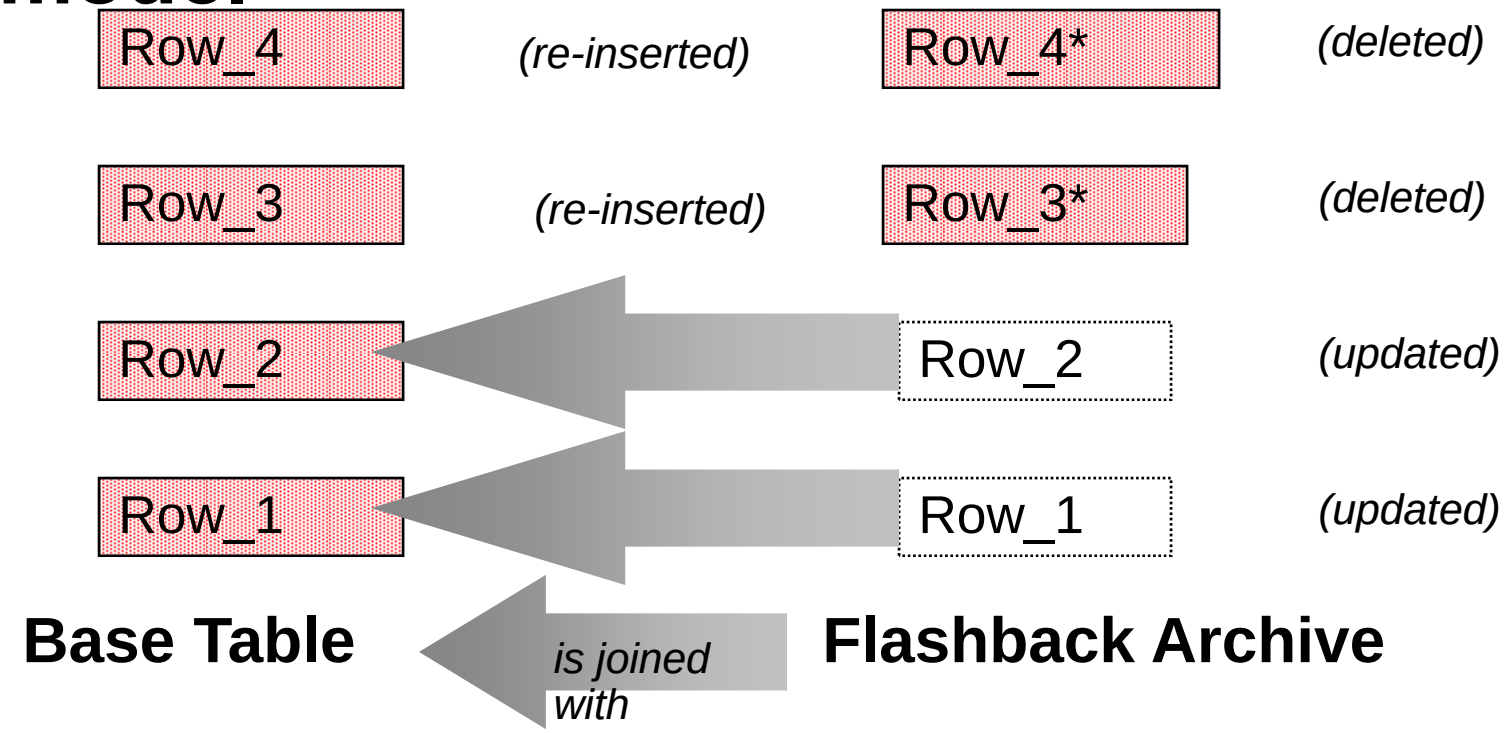


Base Table – New Row Data

Flashback Archive: implementation model



Flashback Archive: implementation model



flashback data archive vs. home-grown solution exercise

Case study

- The Oracle Linux entitlements, available to a customer when Unbreakable Linux Network Support is purchased, are stored as several key components (just to name a few):

CSI	Product	Start_Date	End_Date	Description
1783456	ENT LINUX	31-JAN-2008	30-JAN-2011	Premier 3 Year
1429073	ORACLE VM	01-OCT-2009	30-SEP-2010	Basic 1 Year

  - After a while, this data can become voluminous and it becomes necessary, especially once a contract has expired, to archive this data somewhere

Case study (continued)

- In a home-grown solution, we may make a simple backup of the entitlements table: *entitlements_history*, add a few extra audit columns (see below), and partition the table by time intervals

CSI	…	Updated_By	Updated_On	Action
1783456	…	RJOHNSON	28-FEB-2009	U
1429073	…	TWONG	15-JUN-2010	D

- Managing the partition scheme on such a table is something the developer/DBA must do as part of the home-grown solution

Such a home-grown solution usually involves triggers ….

- The trigger you may use for getting the data from your base table into your *backup* may look similar to the following:

```
SQL> create or replace trigger arud_entitlements
  2  before update or delete on entitlements
  3  for each row
  4    lv_status char(1) := null;
  5  begin
  6    if updating then
  7      lv_status := 'U';
  8    else
  9      lv_status := 'D';
 10    endif;
```

Home-grown Trigger (cont.)

```
11          insert into entitlements_history (csi, product, start_date,
12          end_date, description, … updated_by, updated_on,
13          action)
14        values (:old.csi, :old.product, :old.start_date,
15          :old.end_date, :old.description, … :old.updated_by,
16          :old.updated_on, lv_status);
17        end if;
18      end;
19    /
Trigger created.
```

Things to be aware of with the home-grown type of solution

- The elimination of old (no longer needed) partitions that have exceeded your retention period will need to be managed by you
- The access to this *history* table will need to be *fine grained*
  - Insert privileges for app users and/or trusted users only
  - No Update or Delete privileges
  - DDL (like DROP or TRUNCATE TABLE) may need to be managed/prevented by an event trigger in another schema by another user

Things to be aware of with the home-grown type of solution

- This solution requires a trigger, therefore you cannot perform any INSERT /* + APPEND */ statements on the base table
- SQL*Loader DIRECT PATH LOAD silently disables triggers before loading

Setting up Flashback Data Archive

- To begin to do anything with Flashback Data Archive you need one of the below four items to be in place
  - Logged on as SYSDBA
  - Have been granted the DBA role
  - Have been granted the Flashback Archive Administer system privilege
  - Have been granted a role with the Flashback Archive Administer system privilege

- Set up a Default Flashback Data Archive
  - **create flashback archive default fda tablespace fda_tbs quota 10g retention 2 year;**
  - **Note:** To create a default flashback archive you must be logged on as SYSDBA

Setting up Flashback Data Archive (continued)

- Or set up a non-default flashback data archive
  - **create flashback archive my_fda tablespace fda_tbs1 retention 5 year;**

- Enable your table(s) for archiving
  - **alter table entitlements flashback archive;** (will use the default flashback data archive *fda* in this example), or
  - **alter table entitlements flashback archive my_fda;** (will use a specific flashback data archive *my_fda* in this example)

Setting up Flashback Data Archive (continued)

- To enable a table for flashback data archiving, (at a minimum) you need to have been granted the Flashback Archive object privilege on the flashback data archive of interest
  - **grant flashback archive on my_fda to <user>;**
- The 11*g* background process FBDA (Flashback Data Archiver) takes care of archiving the data for each archive-enabled table

Things to be aware of with the FDA solution

- The FBDA process takes a before-image copy of each changed row directly from the undo segment(s), therefore undo must be available (maybe for longer than without FDA)
- Inserts are not stored in a flashback data archive since they are not (obviously) changed. Oracle uses set operations between your archive and your base table to obtain all data.
- (Pre-12*c*) Though metadata like *what* ('U' and 'D'), and *when* are stored as archive metadata, *who* is not. Consider using audit columns for your *who* metadata.

Archived Tables are Protected

Though you need only the FLASHBACK ARCHIVE object privilege to enable tables for archiving, you need the FLASHBACK ARCHIVE ADMINISTER system privilege to disable any table(s) currently being archived:

```
SQL> alter table entitlements no flashback archive;
alter table entitlements no flashback archive
*
ERROR at line 1:

ORA-55620: No privilege to use Flashback Archive
```

Inserting Into Your Archive

```
update entitlements
 set end_date = end_date + 30
where csi      = l_csi;

delete from entitlements
 where csi      = l_csi_del;

insert into entitlements (csi, product, start_date, description, end_date,
  order_number, license_id, org_id)
            values (l_csi_del, 'ENT LINUX', sysdate - 180, 'Enterprise Linux Premier
  Support 1 Year', sysdate + 265, i + 1, '12345', 1001);
```

Retrieving Data from Your Archive

Using flashback query or flashback versions query

```
select * from entitlements
  as of timestamp to_timestamp
  ('2012-05-10 21:00:00', 'YYYY-MM-DD
    HH24:MI:SS')
where product = ‘ORACLE VM';
```

```
select versions_xid, versions_operation, versions_starttime,
  versions_endtime from entitlements versions between timestamp
      to_timestamp('2012-04-30 00:00:00', 'YYYY-MM-DD HH24:MI:SS') and
      to_timestamp('2012-05-30 00:00:00', 'YYYY-MM-DD HH24:MI:SS')
  where end_date < sysdate + 265;
```

Retrieving Data from Your Archive (Pre-12*c*)

Using flashback query or flashback versions query

audit insert, update, delete on entitlements by access;

```
select versions_xid, versions_operation, versions_starttime,
   versions_endtime, uao.username
  from entitlements e, user_audit objects uao
 where e.versions_xid = uao.transactionid (+)
versions between timestamp
        to_timestamp('2012-04-30 00:00:00', 'YYYY-MM-DD HH24:MI:SS') and
        to_timestamp('2012-05-30 00:00:00', 'YYYY-MM-DD HH24:MI:SS')
   where end_date < sysdate + 265;
```

A Few Notes About Joining to SYS.AUD$

- You should not try to archive tables owned by SYS
- Joining to the aud$ table for this purpose should be used sparingly
- For purging or moving, you can use the DBMS_AUDIT_MGMT package
- For auditing *who* performed a particular delete, consider another table that stores (at a minimum), who, when and primary key
- Try to otherwise manage the archive of *who* performed a particular delete in your application logic

Flashback Transaction Query Configuration

Before you can run any sort of *meaningful* (undo SQL info) flashback transaction query, you'll need to enable supplemental logging:

```
alter database add supplemental log data;
```

And if you expect to run Flashback Transaction (DBMS_FLASHBACK package), you'll need column logging:

```
alter database add supplemental log data (primary key)
columns;
```

Retrieving Data from Your Archive (cont.)

Using flashback transaction query (only useful if logon_user is currently logged on)

```
select xid, operation, logon_user, undo_sql
  from flashback_transaction_query
where xid IN (
select  versions_xid FROM entitlements
versions between timestamp
to_timestamp('2012-04-30 00:00:00', 'YYYY-MM-DD HH24:MI:SS') and
to_timestamp('2012-05-30 00:00:00', 'YYYY-MM-DD HH24:MI:SS‘)
 );
```

Retrieving Data from Your Archive (cont.)

Using flashback transaction query result

```
XID                OPERATION                       LOGON_USER
---------------    ------------------------------- ------------------------------
UNDO_SQL
--------------------------------------------------------------------------------
06000E00ED0D0000   INSERT                          SCOTT
delete from "SCOTT"."ENTITLEMENTS" where ROWID = 'AAAS1XAAEAAAO1+AAZ';

06000E00ED0D0000   DELETE                          SCOTT
insert into "SCOTT"."ENTITLEMENTS"("CSI","PRODUCT","START_DATE","DESCRIPTION","END_DATE","ORDER_NUMBER",
"LICENSE_ID","ORG_ID","CREATED_BY","CREATED_ON","UPDATED_BY","UPDATED_ON") values ('755507','ENT
LINUX',TO_DATE('30-MAY-10', 'DD-MON-RR'),'Enterprise Linux Premier Support 1 Year',TO_DATE('17-AUG-13',
'DD-MON-RR'),'745508','12345','1001','SCOTT',TO_DATE('26-NOV-10', 'DD-MON-RR'),'SCOTT',TO_DATE(
'26-NOV-10', 'DD-MON-RR'));

06000E00ED0D0000   UPDATE                          SCOTT
update "SCOTT"."ENTITLEMENTS" set "END_DATE" = TO_DATE('22-JUN-18', 'DD-MON-RR'), "CREATED_BY" =
'SCOTT', "CREATED_ON" = TO_DATE('26-NOV-10', 'DD-MON-RR'), "UPDATED_BY" = 'SCOTT', "UPDATED_ON" =
TO_DATE('11-APR-13', 'DD-MON-RR') where ROWID = 'AAAS1XAAEAAANAWAAs';
```

FDA Data Dictionary Views

Retrieving metadata about your flashback data archives

```
SQL> column flashback_archive_name format a20
SQL> column quota_in_mb format a10
SQL> select flashback_archive_name, tablespace_name, quota_in_mb
  2    from dba_flashback_archive_ts;

FLASHBACK_ARCHIVE_NA TABLESPACE_NAME                QUOTA_IN_M
-------------------- ------------------------------ ----------
FDA1                 FDA_TBS1


SQL> column flashback_archive_name format a20
SQL> select flashback_archive_name, retention_in_days, status
  2    from dba_flashback_archive;

FLASHBACK_ARCHIVE_NA RETENTION_IN_DAYS STATUS
-------------------- ----------------- -------
FDA1                              3650 DEFAULT
```

FDA Data Dictionary Views (cont.)

Retrieving metadata about your flashback data archives

```
SQL> column table_name format a15
SQL> column owner_name format a15
SQL> column flashback_archive_name format a20
SQL> column archive_table_name format a20
SQL> select table_name, owner_name, flashback_archive_name, archive_table_name
  2    from dba_flashback_archive_tables;

TABLE_NAME      OWNER_NAME      FLASHBACK_ARCHIVE_NA ARCHIVE_TABLE_NAME
--------------- --------------- -------------------- --------------------
INVENTORY       SCOTT           FDA1                 SYS_FBA_HIST_77105
EMPLOYEE        SCOTT           FDA1                 SYS_FBA_HIST_77106
ENTITLEMENTS    SCOTT           FDA1                 SYS_FBA_HIST_77143
```

In 11*g*R2, but not 11*g*R1

- You can now
  - Perform DDL on a column or constraint (add, drop, rename)
  - Drop or truncate a partition
  - Rename or truncate a table
- But in neither R1 nor R2 nor 12*c*R1 can you
  - Move or exchange partitions
  - Drop a table
- To make unsupported changes consider using
  - DBMS_FLASHBACK_ARCHIVE.DISASSOCIATE_FBA to dissociate the table from its archive, make your changes, then
  - DBMS_FLASHBACK_ARCHIVE.REASSOCIATE_FBA

What does 12*c* bring?

- Database Hardening
  - The ability to associate certain security-sensitive tables with an application, then lock them all with one command (preventing further DML).
  - DBMS_FLASHBACK_ARCHIVE.REGISTER_APPLICATION

    (‘ULN_NO_CHANGE’, ‘FDA1’);
- Import of History
  - EXEC DBMS_FLASHBACK_ARCHIVE.extend_mappings();
  - EXEC DBMS_FLASHBACK_ARCHIVE.IMPORT_HISTORY(‘mel','temp_histt');

What does 12*c* bring?

- User Context Tracking (yes!)
  - You can more easily get the “who”

```
mel%ORA12CR1> select versions_xid, versions_operation, versions_starttime,
versions_endtime, dbms_flashback_archive.get_sys_context(versions_xid,
'USERENV', 'SESSION_USER')
  2   from test_fba versions between timestamp
  3   to_timestamp('2013-11-20 00:00:00', 'YYYY-MM-DD HH24:MI:SS') and
  4      to_timestamp('2013-11-20 19:19:00', 'YYYY-MM-DD HH24:MI:SS');

07000900DD0F0000 D 20-NOV-13 07.05.45.000000000 PM
20-NOV-13 07.05.45.000000000 PM
MEL
```

What does 12*c* bring?

- Temporal Validity
  - Specify a start and end period for which rows are valid

```
ashton%ORA12CR1> create table employee_valid (employee_id   NUMBER,
  2  first_name    VARCHAR2(30),
  3  last_name     VARCHAR2(30),
  4  hire_date     DATE
  5  departed_date DATE
  6  PERIOD FOR emp_valid_time (hire_date, departed_date));


ashton%ORA12CR1> select last_name, first_name, hire_date, departed_date
  2               from employee_valid;

LAST_NAME                     FIRST_NAME                    HIRE_DATE DEPARTED_
----------------------------- ----------------------------- --------- ---------
Eckhardt                       Emily                         07-JUL-04 01-JUL-05
Newton                         Frances                       14-SEP-05 07-DEC-06
Newton                         Donald                        24-SEP-06 03-OCT-10

3 rows selected.
```

What does 12*c* bring?

- Temporal Validity

```
ashton%ORA12CR1> select last_name, first_name, hire_date, departed_date
  2               from employee_valid
  3             AS OF PERIOD FOR emp_valid_time TO_DATE('07-JUL-2004', 'DD-MON-YYYY');

LAST_NAME                      FIRST_NAME                     HIRE_DATE DEPARTED_
------------------------------ ------------------------------ --------- ---------
Eckhardt                       Emily                          07-JUL-04 01-JUL-05

1 row selected.

ashton%ORA12CR1> select last_name, first_name, hire_date, departed_date
  2                from employee_valid
  3            VERSIONS PERIOD FOR emp_valid_time BETWEEN
  4            TO_DATE('15-SEP-2005', 'DD-MON-YYYY') AND
  5            TO_DATE('15-JAN-2006', 'DD-MON-YYYY');

LAST_NAME                      FIRST_NAME                     HIRE_DATE DEPARTED_
------------------------------ ------------------------------ --------- ---------
Newton                         Frances                        14-SEP-05 07-DEC-06


1 row selected.
```

In Summary

- Archiving for auditing is made simpler by using built-in technology rather than roll-your-own techniques
- The bulk of the work is delegated to the back-end, meaning this work does not interfere as much with your end users working
- Reduces the need for ongoing maintenance of this type of data for the developer or administrator
- Less prone to data errors or mishaps caused by DDL
- And best of all, partitioning and compression schemes are all managed by FBDA

Q&A