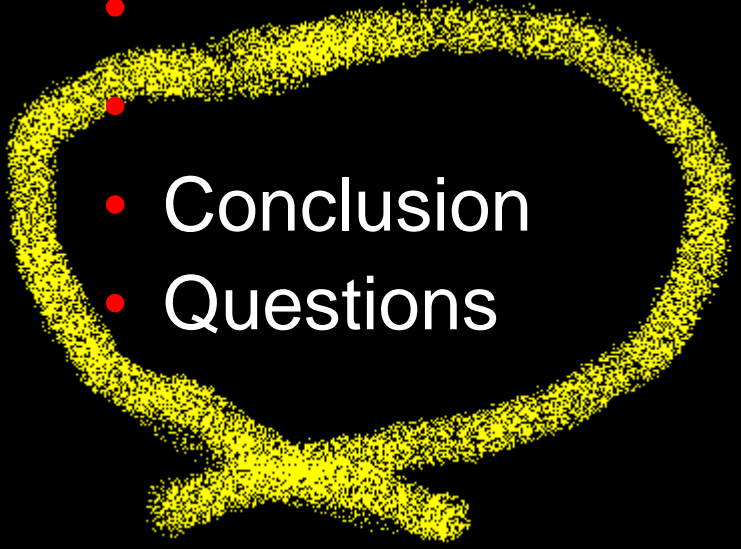


Agenda



- Introduction

Arial font



- Conclusion
- Questions

Agenda

- Introduction
-
-
- Conclusion
- Questions

bullet points

Advertisement

“buy my book”

“visit my website”

“hire our consultants”

“buy our services”

“spend your money on us”

“lease our hardware”



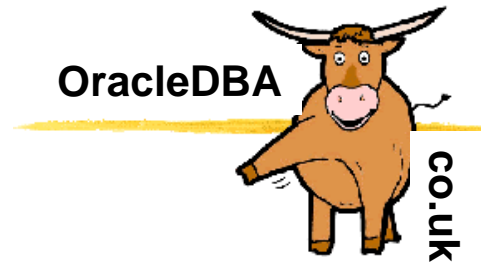
slightly
different

no templates

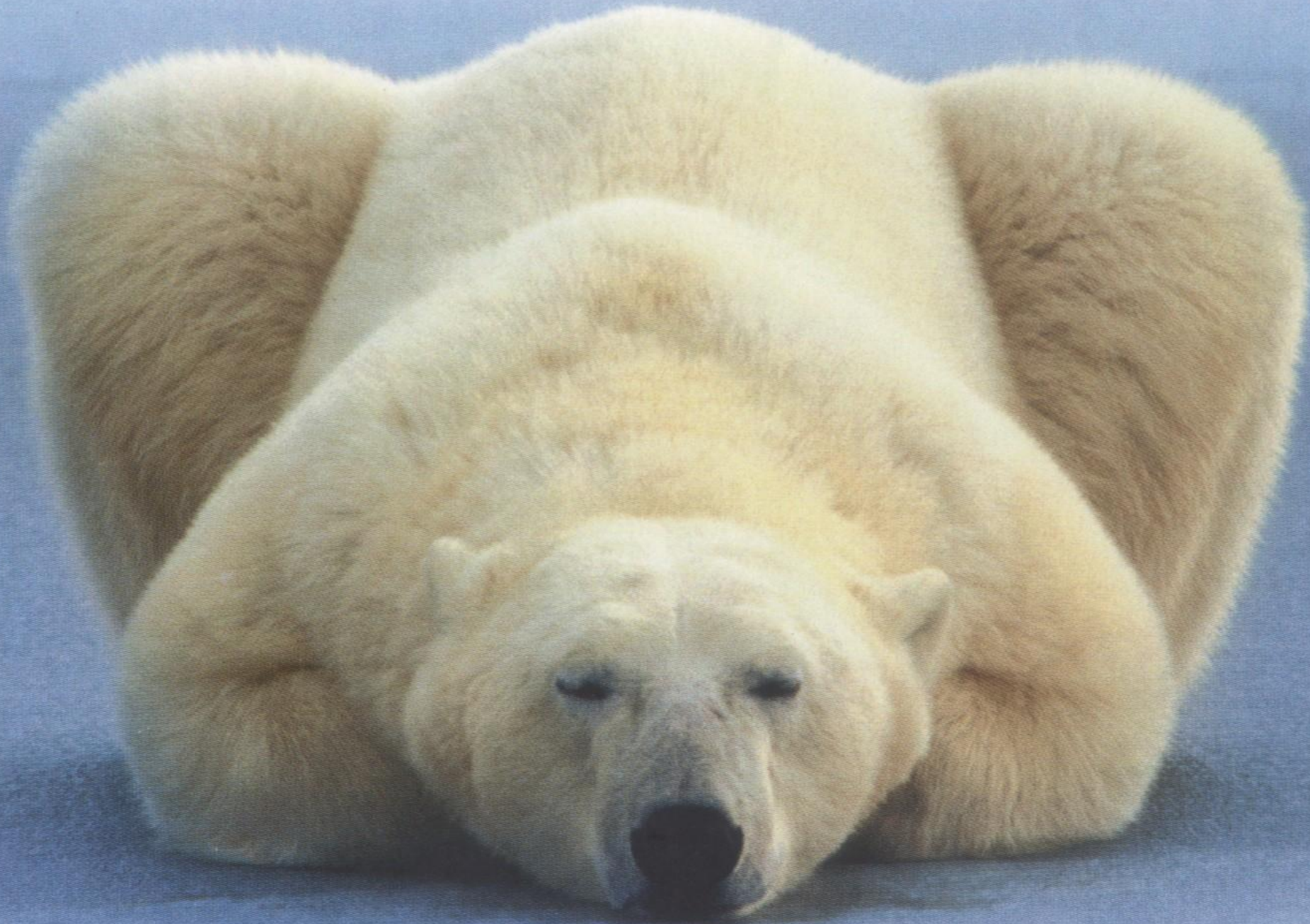
no rules

stuff

Connor McDonald









flashback

flashback query

flashback table

flashback drop

flashback database

flashback transaction

total recall

today's profound statement

you have

ALWAYS

been flashing back

revision

```
SQL> update EMP
  2  set SAL = SAL * 1.10
  3  /
```

14 rows updated.

```
SQL> rollback;
```

Rollback complete.

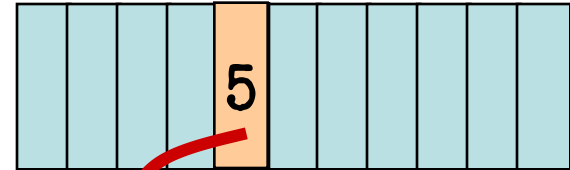
how ?

we remember stuff

```
SQL> update EMP
      2 set SAL = SAL * 1.10
      3 /
```

block 2754

uba 5.2		
1	smith	\$1,000
2	jones	\$990
3	brown	\$1,650
4	wells	\$2,200
5	wilson	\$1,000



undo header

block 2754, row 1,
sal \$1000

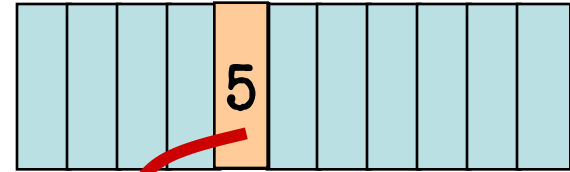
block 2754, row 2,
sal \$900

block 2754, row 3,
sal \$1500

block 2754, row 4,
sal \$2000

```
SQL> rollback;
```

SQL> rollback;



undo header

block 2754

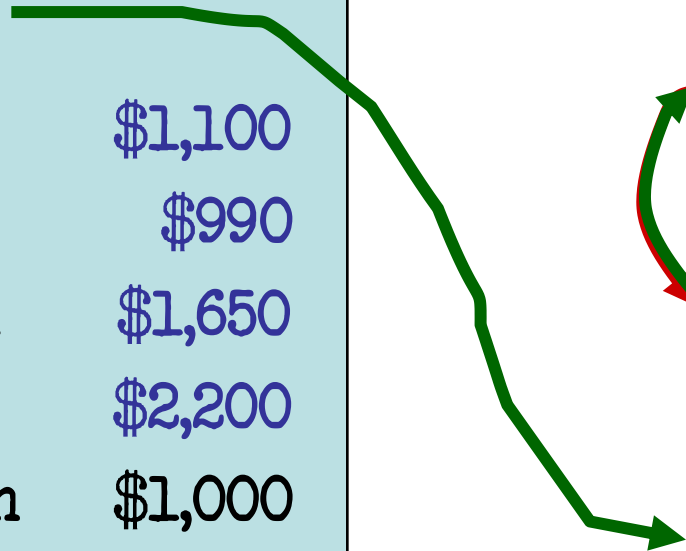
uba 5.2		
1	smith	\$1,100
2	jones	\$990
3	brown	\$1,650
4	wells	\$2,200
5	wilson	\$1,000

block 2754, row 1,
sal \$1000

block 2754, row 2,
sal \$900

block 2754, row 3,
sal \$1500

block 2754, row 4,
sal \$2000



we can use this stuff for queries!

(Oracle version 4)

read consistency

session 1

session 2

9:00

select * from emp
where hiredate > '01/01/2004'

9:03

9:05

"I need block 3217...
... **as it was at 9:00**"

update emp set ...
where empno = 1234;

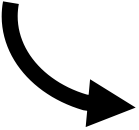


Block 3217



...and time = scn

Block 3217,
SCN 4567003
"XYZ"



Session 2
update emp
set ename = "ABC"
where empno = 1234;
commit;

Block 3217,
SCN 4567234
"ABC"



Block 3217,
SCN 4567003,
change ABC => XYZ

undo segment block(s)

Session 1
Request
Block 3217,
SCN 4567192

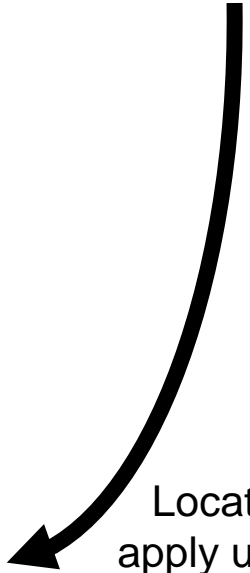
Block 3217,
SCN 4567234
"ABC"

No good..too new
take a copy of the block

Block 3217,
SCN 4567234
"ABC"



Block 3217,
SCN 4567003
"XYZ"



Locate
apply undo

Done ! ←



go back further...

flashback query

v9

```
SQL> select * from DEPT;
```

DEPTNO	DNAME	LOC
10	UNKNOWN	NEW YORK
20	UNKNOWN	DALLAS
30	UNKNOWN	CHICAGO
40	UNKNOWN	BOSTON

```
SQL> select * from DEPT AS OF SCN 995401;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON




```
SQL> select * from DEPT
2 AS OF TIMESTAMP systimestamp -
3 interval '20:00' minute to second;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

`smon_scn_time`

Contents - Mozilla Firefox

File Edit View History Bookmarks Tools Help

file:///C:/oracle/doc/11.1/B28359_01/server.111/b28320/toc.htm#

[DBA_MVIEW_DETAIL_SUBPARTITION](#)
[DBA_MVIEW_JOINS](#)
[DBA_MVIEW_KEYS](#)
[DBA_MVIEW_LOG_FILTER_COLS](#)
[DBA_MVIEW_LOGS](#)
[DBA_MVIEW_REFRESH_TIMES](#)
[DBA_MVIEWS](#)

5 Static Data Dictionary Views: DBA_NESTED_TABLE_COLS to USER_XML_VIEWS

[DBA_NESTED_TABLE_COLS](#)
[DBA_NESTED_TABLES](#)
[DBA_NETWORK_ACL_PRIVILEGES](#)
[DBA_NETWORK_ACLS](#)
[DBA_NETWORK_ACL_ENTRIES](#)

[ROLE_ROLE_PRIVS](#)
[ROLE_SYS_PRIVS](#)
[ROLE_TAB_PRIVS](#)
[SCHEDULER_BATCH_ERRORS](#)
[SEQ](#)
[SESSION_CONTEXT](#)
[SESSION_PRIVS](#)
[SESSION_ROLES](#)
[SOURCE_SIZE](#)
[STMT_AUDIT_OPTION_MAP](#)
[SYN](#)
[SYNONYMS](#)
[SYS_OBJECTS](#)
[SYSCATALOG](#)
[SYSFILES](#)

smon_scn_time ???

```
SQL> select TIMESTAMP_TO_SCN(systimestamp) from DUAL;
```

```
TIMESTAMP_TO_SCN(SYSTIMESTAMP)
```

```
-----
```

```
998332
```

```
SQL> select SCN_TO_TIMESTAMP(998314) from DUAL;
```

```
SCN_TO_TIMESTAMP(998314)
```

```
-----
```

```
29-SEP-08 09.12.48.000000000 PM
```

3 seconds

use in "normal" queries

```

SQL> select case
      2     when d1.deptno is null then 'DELETE'
      3     when d2.deptno is null then 'INSERT'
      4     end action,
      5     d1.deptno, d2.deptno, d1.dname, d1.loc
      6 from DEPT d1
      7 full outer join
      8 DEPT AS OF TIMESTAMP '03-SEP-08 11.51.02 PM' d2
      9 on d1.deptno = d2.deptno;

```

ACTION	DEPTNO	DEPTNO	DNAME	LOC
	10	10	ACCOUNTING	NEW YORK
	20	20	RESEARCH	DALLAS
	30	30	SALES	CHICAGO
DELETE		40		
INSERT	50		MARKETING	PERTH

grant flashback to ...

session level

```
dbms_flashback.enable_at_time(systimestamp-1);
```

just how far ?



`undo_retention`

```
SQL> show parameter undo_retention
```

NAME	TYPE	VALUE
undo_retention	integer	900

much much further

total recall...

no spanning DDL


```
SQL> select * from DEPT;
```

DEPTNO	DNAME	LOC
--------	-------	-----

10	UNKNOWN	NEW YORK
----	---------	----------

1 change ? 10 changes ?

```
SQL> select * from DEPT
```

```
2 as of TIMESTAMP SYSTIMESTAMP -
```

```
3 INTERVAL '20:00' MINUTE TO SECOND;
```

DEPTNO	DNAME	LOC
--------	-------	-----

10	ACCOUNTING	NEW YORK
----	------------	----------

flashback row versions

```

SQL> SELECT deptno, dname
2 FROM dept
3 VERSIONS BETWEEN
4     TIMESTAMP SYSTIMESTAMP -
5     INTERVAL '20:00' MINUTE TO SECOND
6     AND SYSTIMESTAMP
7 WHERE deptno = 10;

```

```

DEPTNO DNAME
-----
10 ACCOUNTING
10 MONEY GRABBERS
10 FINANCE
10 BEAN COUNTERS
10 UNKNOWN

```

1 row
5 versions

versions_starttime

versions_startscn

versions_endtime

versions_endscn

versions_xid

versions_operation

```

SQL> SELECT deptno, dname,
2         VERSIONS_STARTTIME
3         ,VERSIONS_XID
4         ,VERSIONS_OPERATION
5 FROM dept
6 VERSIONS BETWEEN
7     TIMESTAMP SYSTIMESTAMP - INTERVAL '20:00' MINUTE TO SECOND
8     AND SYSTIMESTAMP
9 WHERE deptno = 10;

```

DEPTNO	DNAME	VERSIONS_STARTTIME	VERSIONS_XID	V
10	UNKNOWN	03-SEP-08 11.53.45 PM	0200100060040000	U
10	MONEY GRABBERS	03-SEP-08 11.53.36 PM	0600050065040000	U
10	FINANCE	03-SEP-08 11.53.24 PM	09000D001D050000	U
10	BEAN COUNTERS	03-SEP-08 11.53.12 PM	01001A00EA030000	U
10	ACCOUNTING			

get the lot ...

```

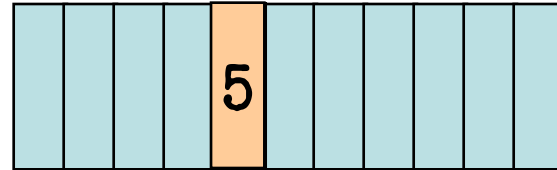
SQL> SELECT deptno, dname,
2         VERSIONS_STARTTIME
3         ,VERSIONS_XID
4         ,VERSIONS_OPERATION
5 FROM dept VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE;

```

DEPTNO	DNAME	VERSIONS_STARTTIME	VERSIONS_XID	V
50	UNKNOWN	03-SEP-08 11.08.15 PM	04000700EA030000	U
30	UNKNOWN	03-SEP-08 11.08.15 PM	04000700EA030000	U
20	NERDS	03-SEP-08 11.07.57 PM	090016001D050000	U
20	R&D	03-SEP-08 11.07.48 PM	05000B0074040000	U
20	GEEKS	03-SEP-08 11.07.36 PM	0300080071040000	U
20	UNKNOWN	03-SEP-08 11.07.14 PM	0A001600F8030000	U
10	ACCOUNTING	03-SEP-08 11.53.45 PM	0200100060040000	U
10	MONEY GRABBERS	03-SEP-08 11.53.36 PM	0600050065040000	U
10	FINANCE	03-SEP-08 11.53.24 PM	09000D001D050000	U
10	BEAN COUNTERS	03-SEP-08 11.53.12 PM	01001A00EA030000	U
10	UNKNOWN			
20	RESEARCH			
30	SALES			
50	MARKETING			
40	HR	03-SEP-08 11.08.42 PM	07000600E1030000	I

```
SQL> update DEPT
  2  set DNAME = 'FINANCE'
  3  where DEPTNO = 10;

SQL> rollback;
```



undo header

XID ?


```
SQL> desc FLASHBACK_TRANSACTION_QUERY
```

Name	Null?	Type
XID		RAW (8)
START_SCN		NUMBER
START_TIMESTAMP		DATE
COMMIT_SCN		NUMBER
COMMIT_TIMESTAMP		DATE
LOGON_USER		VARCHAR2 (30)
UNDO_CHANGE#		NUMBER
OPERATION		VARCHAR2 (32)
TABLE_NAME		VARCHAR2 (256)
TABLE_OWNER		VARCHAR2 (32)
ROW_ID		VARCHAR2 (19)
UNDO_SQL		VARCHAR2 (4000)

```
SQL> SELECT xid, UNDO_SQL
2 FROM flashback_transaction_query
2 WHERE xid = hextoraw('09000D001D050000');
```

XID	UNDO_SQL
09000D001D050000	update "SCOTT"."DEPT" set "DNAME" = 'BEAN COUNTERS' where ROWID = 'AAQ+hAAEAAAAAOAAA';

be careful

```
SQL> select text from dba_views
      2  where view_name = 'FLASHBACK_TRANSACTION_QUERY';
```

TEXT

```
-----
select
  xid, start_scn, start_timestamp,
  decode(commit_scn,
          0, commit_scn,
          281474976710655, NULL, commit_scn)
  commit_scn, commit_timestamp,
  logon_user, undo_change#, operation,
  table_name, table_owner,
  row_id, undo_sql
from SYS.X$KTUQQRY
```

Do NOT query without predicates

Do NOT forget "hextoraw"



UNDO_SQL

”rollback”
committed transaction

three words

delete

commit

uh-oh...

undo a transaction

flashback transaction

llg

```
SQL> BEGIN
  2     DBMS_FLASHBACK.TRANSACTION_BACKOUT (
  3         numtxns => 1,
  4         xids     => sys.xid_array('09000D001D050000')
  5     );
  6 END;
  7 /
```

```
BEGIN
```

```
*
```

```
ERROR at line 1:
```

```
ORA-55510: Mining could not start
```

```
ORA-06512: at "SYS.DBMS_FLASHBACK", line 37
```

```
ORA-06512: at "SYS.DBMS_FLASHBACK", line 70
```

```
ORA-06512: at line 2
```



```
SQL> ALTER DATABASE ADD  
2 SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
```

Database altered.

```
SQL> delete from DEPT where DEPTNO = 10;
```

```
1 row deleted.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> SELECT VERSIONS_XID  
2      FROM dept  
3      VERSIONS BETWEEN  
4          SCN MINVALUE AND MAXVALUE  
5  /
```

```
VERSIONS_XID  
-----  
080017006C040000
```

```

SQL> BEGIN
  2     DBMS_FLASHBACK.TRANSACTION_BACKOUT (
  3         numtxns => 1,
  4         xids     => sys.xid_array('080017006C040000')
  5     );
  6 END;
  7 /

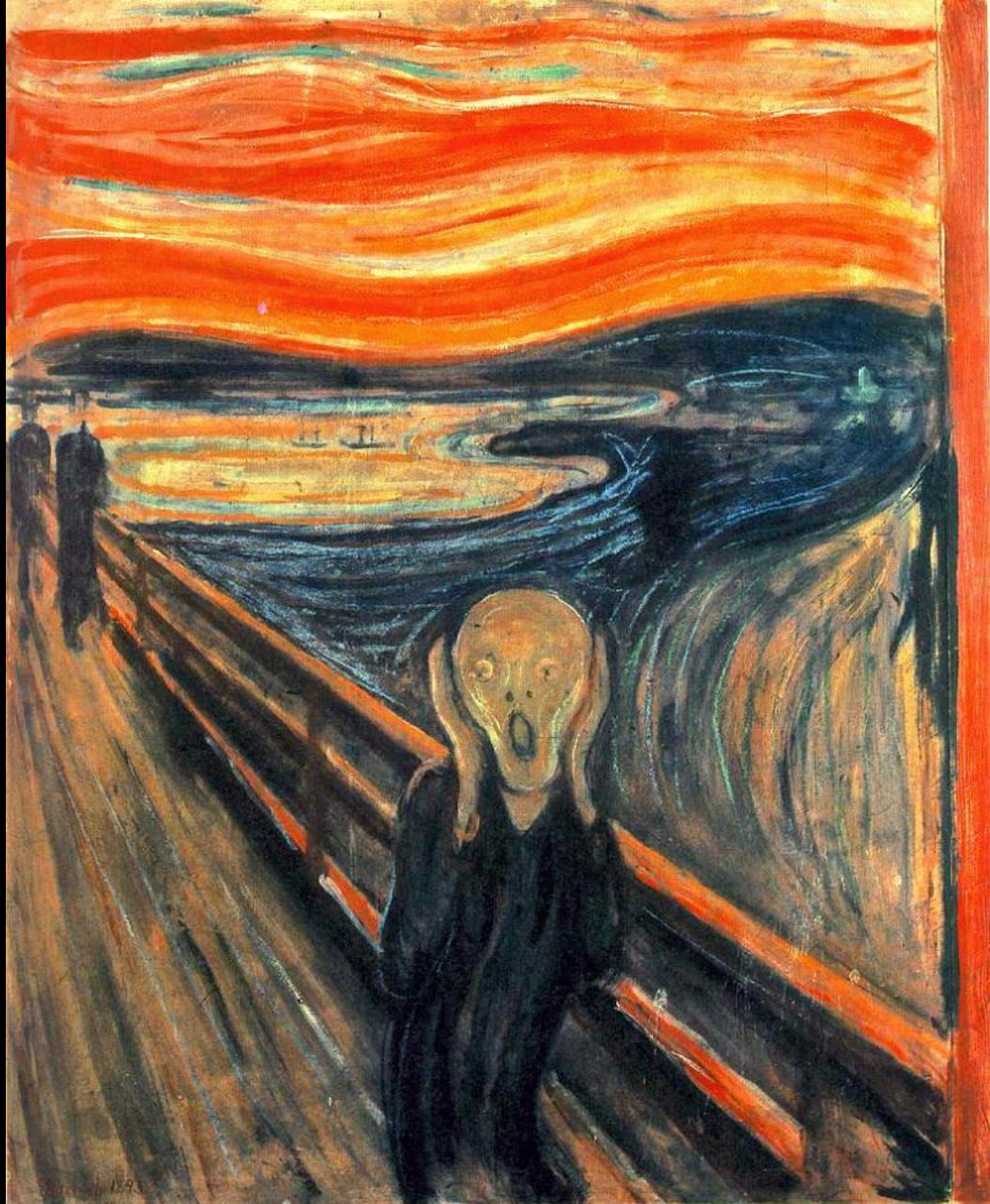
```

PL/SQL procedure successfully completed.

```
SQL> select * from DEPT;
```

DEPTNO	DNAME	LOC
20	RESEARCH	DALLAS
30	SALES	CHICAGO
50	MARKETING	PERTH
40	HR	US

????



```
SQL> SELECT COMPENSATING_XID
2 FROM DBA_FLASHBACK_TXN_STATE
3 WHERE xid = '080017006C040000';
```

```
COMPENSATING_XID
-----
03001C0072040000
```

```
SQL> SELECT xid_report
       2 FROM   DBA_FLASHBACK_TXN_REPORT
       3 WHERE  compensating_xid = '03001C0072040000';
```

XID_REPORT

```
-----
<COMP_XID_REPORT XID="03001C0072040000">
  <TRANSACTION XID="080017006C040000">
    <UNDO_SQL>
      <USQL exec="yes">
        insert into "SCOTT"."DEPT" ("DEPTNO", "DNAME", "LOC")
        values ('10', 'ACCOUNTING', 'NEW YORK')
      </USQL>
    </UNDO_SQL>
  </TRANSACTION>
  <EXECUTED_UNDO_SQL>
  <EXEC_USQL>
    insert into "SCOTT"."DEPT" ("DEPTNO", "DNAME", "LOC")
    values ('10', 'ACCOUNTING', 'NEW YORK')
  </EXEC_USQL>
</EXECUTED_UNDO_SQL>
</COMP_XID_REPORT>
```

```
SQL> commit;
```

```
Commit complete.
```

DEPTNO	DNAME	LOC
20	RESEARCH	DALLAS
30	SALES	CHICAGO
50	MARKETING	PERTH
40	HR	US
10	ACCOUNTING	NEW YORK

~~three~~ words
two

drop

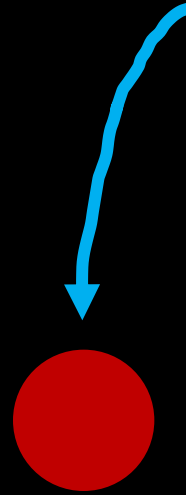
uh-oh...

```
SQL> desc EMP
```

```
ERROR:
```

```
ORA-04043: object EMP does not exist
```

You are here



Hard
Place

”undrop”



entropyzone.com

flashback table to before drop



```
SQL> drop table EMP;
```

Table dropped.

```
SQL> desc USER_RECYCLEBIN
```

Name	Null?	Type
OBJECT_NAME	NOT NULL	VARCHAR2 (30)
ORIGINAL_NAME		VARCHAR2 (32)
OPERATION		VARCHAR2 (9)
TYPE		VARCHAR2 (25)
TS_NAME		VARCHAR2 (30)
CREATETIME		VARCHAR2 (19)
DROPTIME		VARCHAR2 (19)
DROPSCN		NUMBER
PARTITION_NAME		VARCHAR2 (32)
CAN_UNDROP		VARCHAR2 (3)
CAN_PURGE		VARCHAR2 (3)
RELATED	NOT NULL	NUMBER
BASE_OBJECT	NOT NULL	NUMBER
PURGE_OBJECT	NOT NULL	NUMBER
SPACE		NUMBER

```
SQL> select OBJECT_NAME, ORIGINAL_NAME,
2          CAN_UNDROP, CAN_PURGE
3 from    USER_RECYCLEBIN;
```

OBJECT_NAME	ORIGINAL_NAME	CAN	CAN
-----	-----	---	---
BIN\$CmYCdNcITL6hp91266nskA==\$0	PK_EMP	NO	YES
BIN\$b/W75+c/Q/CeFMBIK7cXfw==\$0	EMP	YES	YES

```
SQL> show RECYCLEBIN
```

indexes / constraints / triggers

... all renamed


```
SQL> flashback table EMP to before drop;
```

```
Flashback complete.
```

```
SQL> flashback table EMP to before drop  
2      rename to OLD_EMP;
```

Flashback complete.

indexes / constraints / triggers

... not renamed back

how long ?

1 second...



10 years

```
SQL> PURGE TABLE "BIN$xyWe0+q+SniItJ0pn/u54A==$0";
```

```
SQL> PURGE TABLE emp;
```

```
SQL> PURGE RECYCLEBIN;
```

```
SQL> PURGE TABLESPACE user1;
```

```
SQL> PURGE TABLESPACE user1 USER fred;
```

```
SQL> PURGE INDEX "BIN$FTX34MN88J7==$0";
```

no untruncate

so far



flashback table

```
SQL> FLASHBACK TABLE DEPT  
2 TO TIMESTAMP TO_TIMESTAMP('2008-09-03 12:05:00')
```



“undo”

```
SQL> FLASHBACK TABLE DEPT  
2 TO TIMESTAMP TO_TIMESTAMP('2008-09-03 12:05:00');
```

deletes modified rows since SCN
inserts fresh rows as of SCN

... busy

```
SQL> FLASHBACK TABLE DEPT
      2 TO TIMESTAMP TO_TIMESTAMP('2008-09-03 12:05:00')
      3 ENABLE TRIGGERS;
```

rowid's change

... enable row movement

indexes existence unaffected

single transaction

(multi table allowed)

tables locked

R.I must still be valid

statistics unchanged

`sys_temp_fbt`

most DDL is a flashback boundary

```
SQL> flashback table DEPT to scn ...;
```

```
ORA-01466: unable to read data - table definition has changed
```

```
SQL> flashback table DEPT to scn ...;
```

```
ORA-08183: Flashback cannot be enabled in the middle  
of a transaction
```






flashback database

```
SQL> alter database flashback on;
```

```
Database altered.
```

data
changes

dbwr

redo

fbda

blocks
to disk

changes
to disk

old blocks
to disk

```
SQL> FLASHBACK DATABASE TO '2:05PM'
```

```
SQL> create restore point OK_SO_FAR;
```

```
...
```

```
SQL> flashback database to restore point OK_SO_FAR;
```

what does it cost ?


```
SQL> desc V$FLASHBACK_DATABASE_STAT
```

Name	Null?	Type
BEGIN_TIME		DATE
END_TIME		DATE
FLASHBACK_DATA		NUMBER
DB_DATA		NUMBER
REDO_DATA		NUMBER
ESTIMATED_FLASHBACK_SIZE		NUMBER

its all about blocks

```
SQL> update EMP
      2 set      JOB = ...
      3 where   rowid = ...
```

x 28,000

```
SQL> select * from V$FLASHBACK_DATABASE_STAT;
```

[delta]

FLASHBACK_DATA	REDO_DATA
-----	-----
3.9M	9.5M

```
SQL> update BIG_EMP
      2 set      JOB = ...
      3 where   rowid = ...
```

x 28,000

```
SQL> select * from V$FLASHBACK_DATABASE_STAT;
```

[delta]

FLASHBACK_DATA	REDO_DATA
-----	-----
9.8M	9.8M



guaranteed restore points

flashback logging optional

9:00 MyApp release 3 installation

9:40



9:45 SQL> FLASHBACK DATABASE...



combining the history



undo
(flashback
query)



flashback logs
(flashback
database)

total recall

llg

```
SQL> create tablespace SPACE_FOR_ARCHIVE
      2 datafile 'C:\ORACLE\DB11\ARCH_SPACE.DBF'
      3 size 100M;
```

Tablespace created.

```
SQL> CREATE FLASHBACK ARCHIVE longterm
      2 TABLESPACE space_for_archive
      3 RETENTION 1 YEAR;
```

Flashback archive created.

```
SQL> ALTER TABLE EMP FLASHBACK ARCHIVE LONGTERM;
```

```
Table altered.
```

```
[lots of DML]
```

```
SQL> select * from EMP;
```

```
-----  
| Id  | Operation                               | Name | Rows  | Bytes |  
-----  
|  0  | SELECT STATEMENT                       |      |   14  |  518  |  
|  1  | TABLE ACCESS FULL                     | EMP  |   14  |  518  |  
-----
```

```
SQL> select * from EMP
      2 AS OF TIMESTAMP SYSDATE-3;
```

Id	Operation	Name	Rows	Bytes
0	SELECT STATEMENT		14	518
1	TABLE ACCESS FULL	EMP	14	518

[lots more DML]

```
SQL> select * from EMP
      2 AS OF TIMESTAMP SYSDATE-3;
```

Id	Operation	Name	Rows
0	SELECT STATEMENT		446
1	VIEW		446
2	UNION-ALL		
* 3	FILTER		
4	PARTITION RANGE ITERATOR		445
* 5	TABLE ACCESS FULL	SYS_FBA_HIST_69539	445
* 6	FILTER		
* 7	HASH JOIN OUTER		1
* 8	TABLE ACCESS FULL	EMP	1
9	VIEW		14
* 10	TABLE ACCESS FULL	SYS_FBA_TCRV_69539	14

```
SQL> select table_name
      2 from user_tables
      3 /
```

TABLE_NAME

SYS_FBA_HIST_71036
SYS_FBA_TCRV_71036
SYS_FBA_DDL_COLMAP_71036
EMP


```
SQL> select dbms_metadata.get_ddl(
        'TABLE' ,
        'SYS_FBA_HIST_71036') from dual;
```

```
CREATE TABLE "SCOTT"."SYS_FBA_HIST_71036"
  ( "RID"          VARCHAR2(4000) ,
    "STARTSCN"    NUMBER ,
    "ENDSCN"      NUMBER ,
    "XID"         RAW(8) ,
    "OPERATION"   VARCHAR2(1) ,
    "EMPNO"       NUMBER(4,0) ,      "ENAME"    VARCHAR2(10) ,
    "JOB"         VARCHAR2(9) ,      "MGR"      NUMBER(4,0) ,
    "HIREDATE"    DATE ,            "SAL"      NUMBER(7,2) ,
    "COMM"        NUMBER(7,2) ,      "DEPTNO"   NUMBER(2,0)
  ) COMPRESS FOR ALL OPERATIONS
TABLESPACE "SPACE_FOR_ARCHIVE"
PARTITION BY RANGE ("ENDSCN")
  (
    PARTITION "HIGH_PART"  VALUES LESS THAN (MAXVALUE)
  )
```

```
SQL> select dbms_metadata.get_ddl(  
        'TABLE',  
        'SYS_FBA_TCRV_71036') from dual;
```

```
CREATE TABLE "SCOTT"."SYS_FBA_TCRV_71036"  
  (  
    "RID"          VARCHAR2(4000),  
    "STARTSCN"    NUMBER,  
    "ENDSCN"      NUMBER,  
    "XID"          RAW(8),  
    "OPERATION"   VARCHAR2(1)  
  )  
TABLESPACE "SPACE_FOR_ARCHIVE"
```

```
SQL> select * from EMP
      2 AS OF TIMESTAMP SYSDATE-3;
```

Id	Operation	Name	Rows
0	SELECT STATEMENT		446
1	VIEW		446
2	UNION-ALL		
* 3	FILTER		
4	PARTITION RANGE ITERATOR		445
* 5	TABLE ACCESS FULL	SYS_FBA_HIST_69539	445
* 6	FILTER		
* 7	HASH JOIN OUTER		1
* 8	TABLE ACCESS FULL	EMP	1
9	VIEW		14
* 10	TABLE ACCESS FULL	SYS_FBA_TCRV_69539	14

restrictions #1

```
SQL> alter table EMP RENAME column job to jobtitle;
```

```
*
```

```
ERROR at line 1:
```

```
ORA-55610: Invalid DDL statement on history-tracked table
```

```
SQL> alter table emp DROP column dummy;  
alter table scott.emp drop column dummy
```

```
*
```

```
ERROR at line 1:
```

```
ORA-55610: Invalid DDL statement on history-tracked table
```

```
SQL> alter table emp ADD dummy int;
```

```
Table altered.
```

be careful...

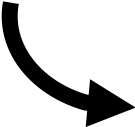
no spanning DDL might apply

restrictions #2



other stuff

Block 3217,
SCN 4567003
"XYZ"



Session 2
update emp
set ename = "ABC"
where empno = 1234;
commit;

Block 3217,
SCN 4567234
"ABC"



Block 3217,
SCN 4567003,
change ABC => XYZ

undo segment block(s)

Session 1
Request
Block 3217,
SCN 4567192

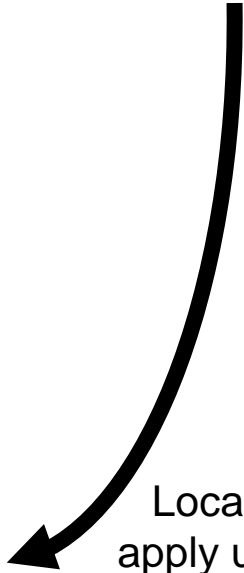
Block 3217,
SCN 4567234
"ABC"

No good..too new
take a copy of the block

Block 3217,
SCN 4567234
"ABC"



Block 3217,
SCN 4567003
"XYZ"



Locate
apply undo

Done ! ←

ORA_ROWSCN

```
SQL> SELECT ORA_ROWSCN, empno, ename, sal  
2 FROM emp;
```

ORA_ROWSCN	EMPNO	ENAME	SAL
1357038	7369	SMITH	800
1357038	7499	ALLEN	1600
1357038	7521	WARD	1250
1357038	7566	JONES	2975
1357038	7654	MARTIN	1250
1357038	7698	BLAKE	2850
1357038	7782	CLARK	2450
1357038	7788	SCOTT	3000
1357038	7839	KING	5000
1357038	7844	TURNER	1500
1357038	7876	ADAMS	1100
1357038	7900	JAMES	950
1357038	7902	FORD	3000
1357038	7934	MILLER	1300

block level SCN

```
SQL> UPDATE emp SET sal = 9999 WHERE empno = 7369;
```

```
SQL> SELECT ORA_ROWSCN, empno, ename, sal  
2 FROM emp;
```

ORA_ROWSCN	EMPNO	ENAME	SAL
1357484	7369	SMITH	9999
1357484	7499	ALLEN	1600
1357484	7521	WARD	1250
1357484	7566	JONES	2975
1357484	7654	MARTIN	1250
1357484	7698	BLAKE	2850
1357484	7782	CLARK	2450
1357484	7788	SCOTT	3000
1357484	7839	KING	5000
1357484	7844	TURNER	1500
1357484	7876	ADAMS	1100
1357484	7900	JAMES	950
1357484	7902	FORD	3000
1357484	7934	MILLER	1300

```
SQL> CREATE TABLE newemp ROWDEPENDENCIES  
2 AS SELECT * FROM emp;
```

```
SQL> UPDATE newemp SET sal = 9999 WHERE empno = 7369
```

```
SQL> SELECT ORA_ROWSCN, empno, ename, sal  
2 FROM newemp;
```

ORA_ROWSCN	EMPNO	ENAME	SAL
1357558	7369	SMITH	9999
1357545	7499	ALLEN	1600
1357545	7521	WARD	1250
1357545	7566	JONES	2975
1357545	7654	MARTIN	1250
1357545	7698	BLAKE	2850
1357545	7782	CLARK	2450
1357545	7788	SCOTT	3000
1357545	7839	KING	5000
1357545	7844	TURNER	1500
1357545	7876	ADAMS	1100
1357545	7900	JAMES	950
1357545	7902	FORD	3000
1357545	7934	MILLER	1300


```
SQL> select ORA_ROWSCN,  
           SCN_TO_TIMESTAMP(ORA_ROWSCN)  
2  FROM emp  
3  WHERE ...
```

wrap up

flashback query

flashback table

flashback drop

flashback database

flashback transaction

total recall

ORA-03113