Reversing Committed Transactions - CTRL-Z in SQL*Plus!

Peter Robson

(Independent)

peter.robson@justsql.com

Objectives

- Define the CTRL-Z and CTRL-Y process
- Demonstrate both processes
- Describe architecture to implement
- Describe elements of the components

demo…

– In Windows…..

What CTRL-Z is:

- Based on the Windows editor function
- Reverse the last immediate change
- Now applied to reversing last database changes
- Unit of change is the Row (not Commit)
- Like Windows, a 2-key operation

What CTRL-Y is:

- Based on the Windows editor function
- Reverse the last immediate CTRL-Z changes
- Only works against previous CTRL-Z changes
- Unit of change is the Row (not Commit)
- Like Windows, a 2-key operation

A Live Example…

- In SQL*Plus
- Several modifications to a Table
- Modifications are Committed
- CTRL-Z will reverse those modification
- CTRL-Y will reverse the reverses!

demo…

– In SQL*Plus…..

In the following examples…

colour coding will be used for the three major table-types:

Pale blue for the **DATA TABLE**

Pink for the **HISTORY TABLE**

In the following examples…

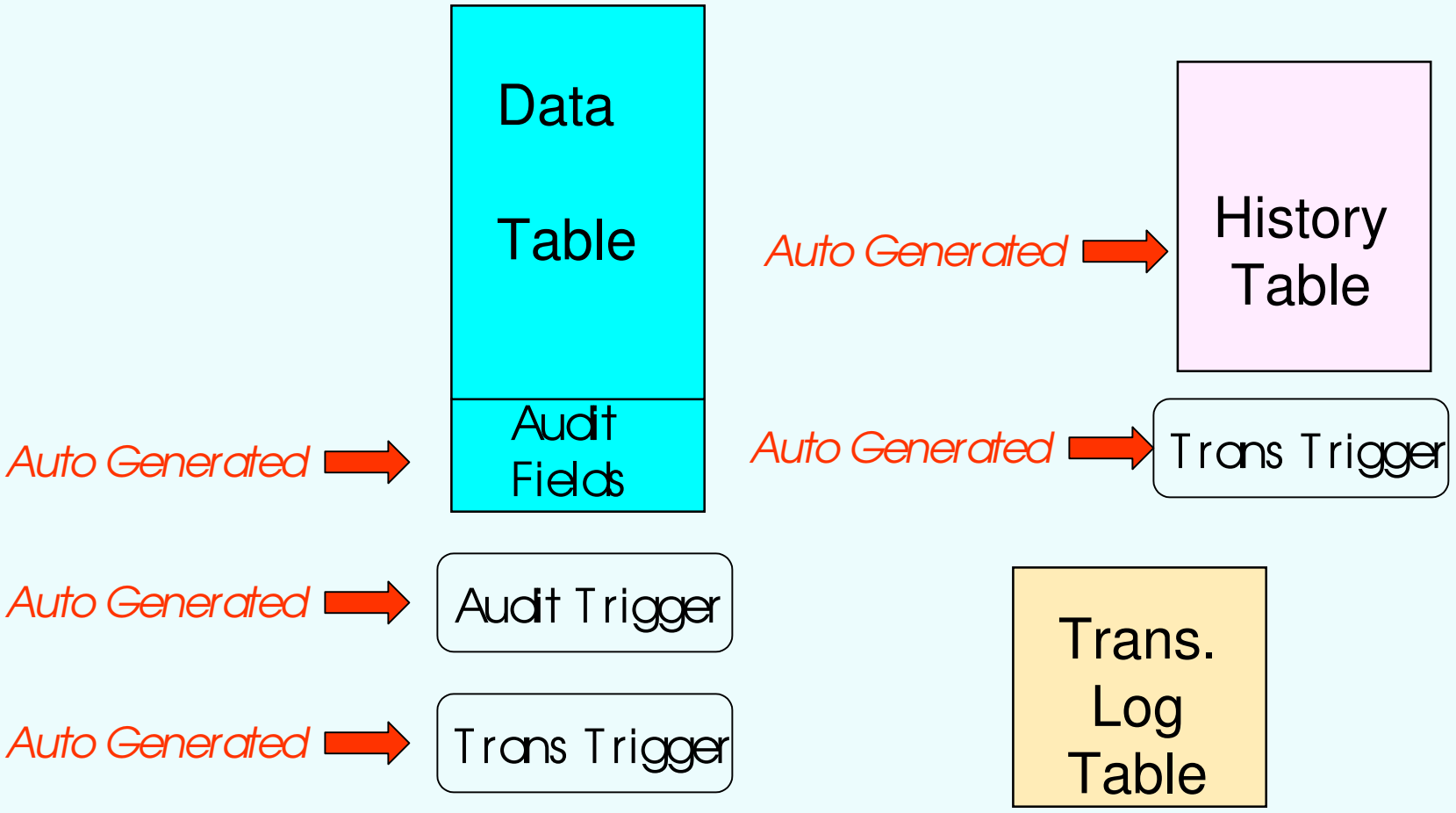
colour coding will be used for the three major table-types:

Pale blue for the **DATA TABLE**

Pink for the **HISTORY TABLE**

Brown for the **TRANSACTION LOG TABLE**

Building the Required Components



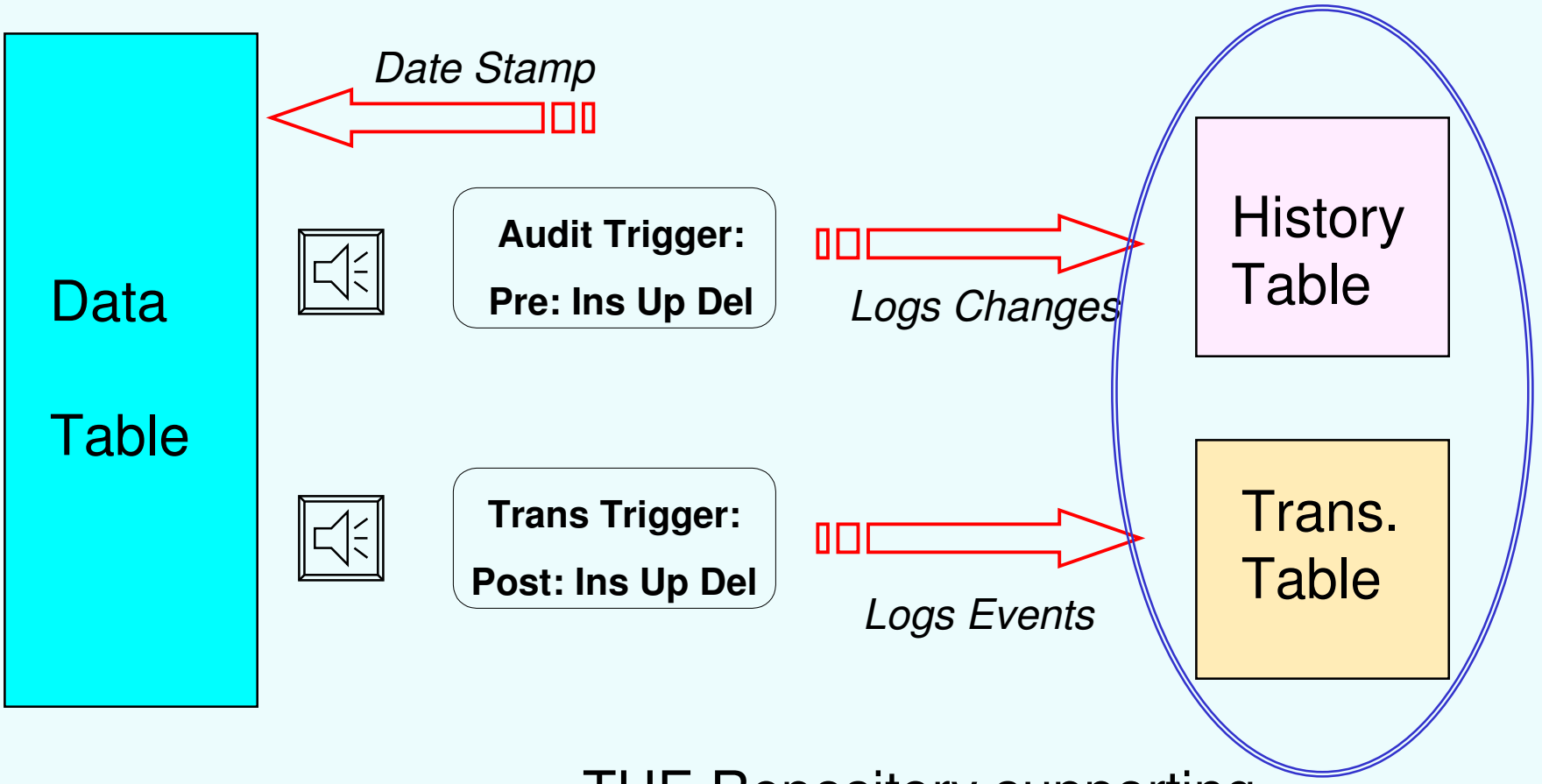
The Components:

- **The Data Table**
  - Audit trigger
  - Transaction trigger

- **History Table**
  - Transact Deletion trigger

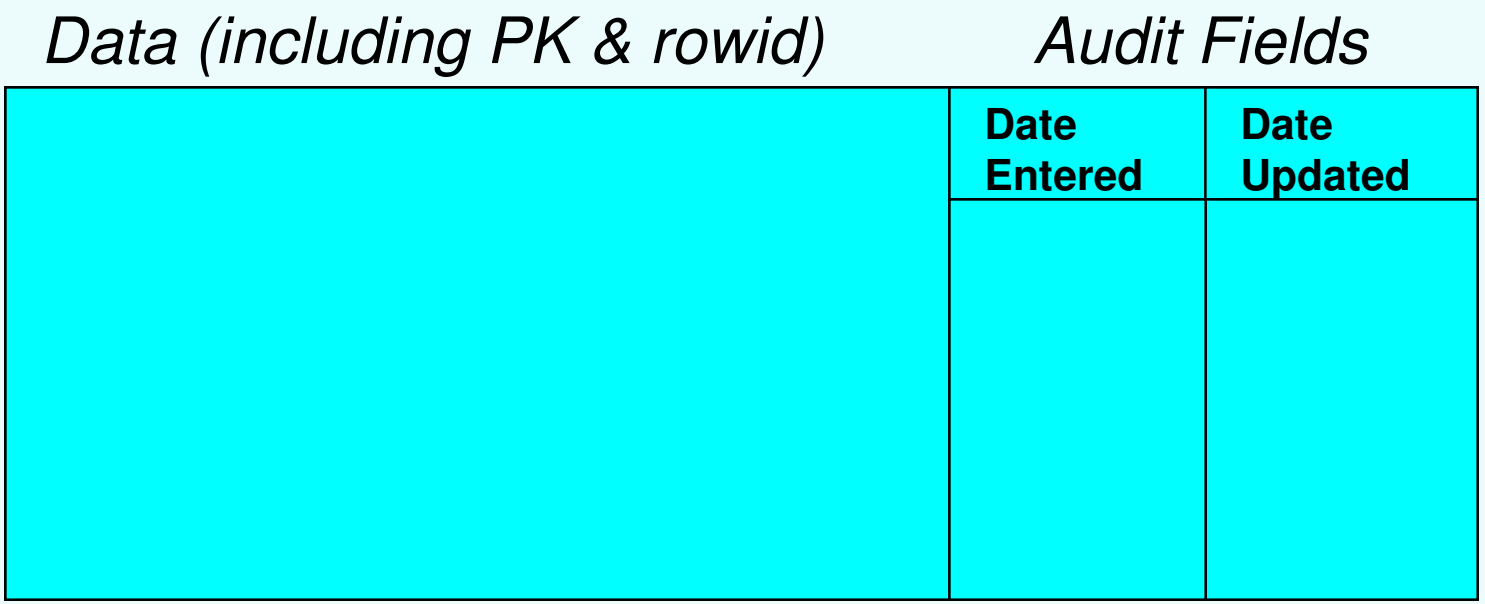
- **Transaction Log Table**
  - Audit trigger

The Components Working:



THE Repository supporting
Audit, Backup & Recovery

The Component Structures: Data Table



The Component Structures: History Table

Data (excl. rowid) + Audit Fields *Audit Table Fields*

Data Fields...	Date Entered	Date Updated	Event	Date
			‘ I ‘ ‘ U ‘ ‘ D ‘	

The Component Structures: Transaction Log Table

Table Name	Event	Date	Sequence	Rowid	
	‘ I ‘ ‘ U ‘ ‘ D ‘				

The Domain of values

The Three Tables:

Data:

Data Fields...	Date Entered	Date Updated

History:

Data Fields...	Date Entered	Date Updated	Event	Date

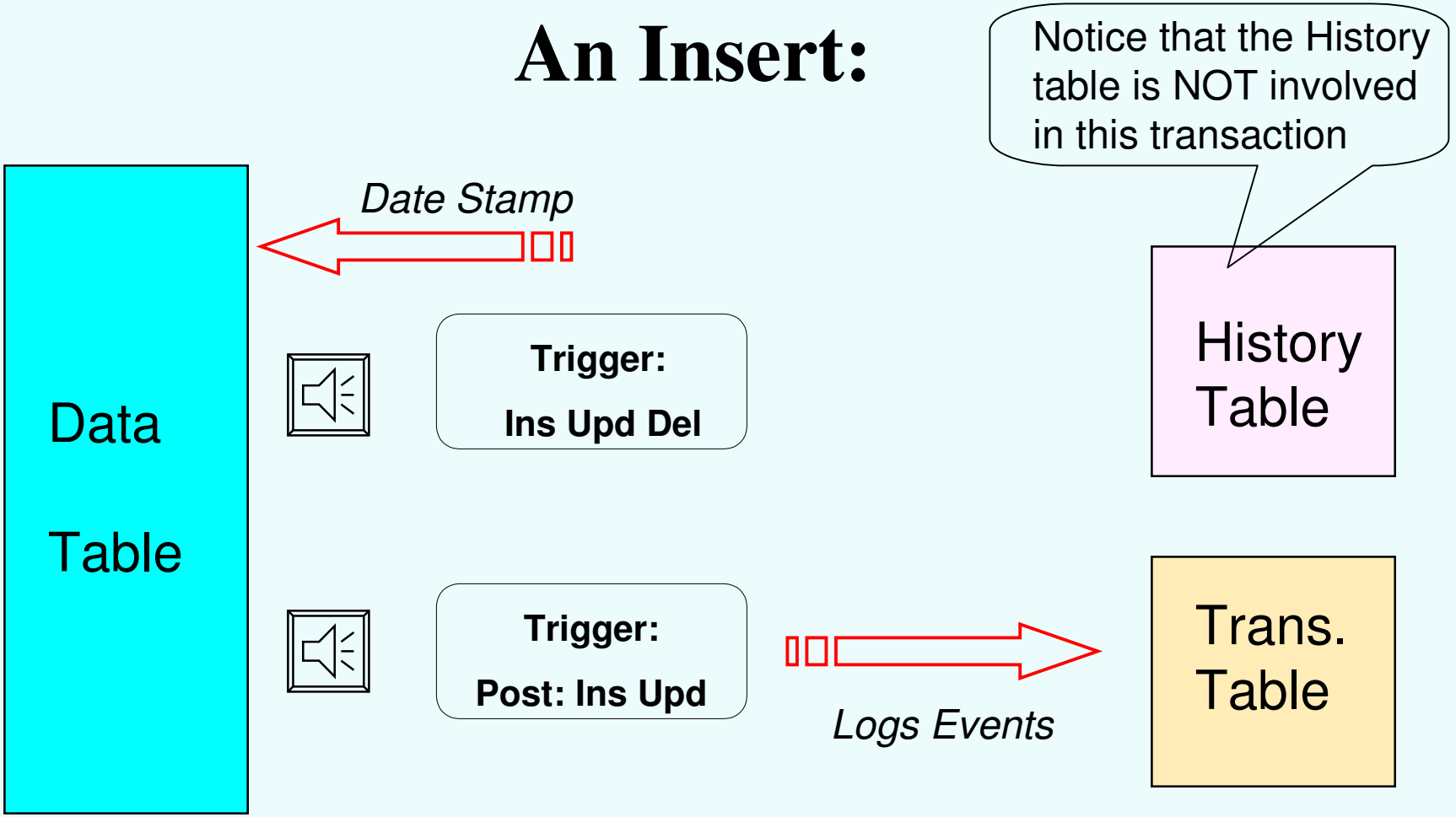
Transaction:

Table Name	Event	Date	Sequence	Rowid	

Insert Procedure:

- New Row *Inserted* into Data Table
  - Row Inserted
  - Audit trigger fires:
    - Timestamps new row (date_entered, etc)
  - Transaction trigger fires:
    - Populates Transaction Log Table
    - Captures *ROWID* from Data Table

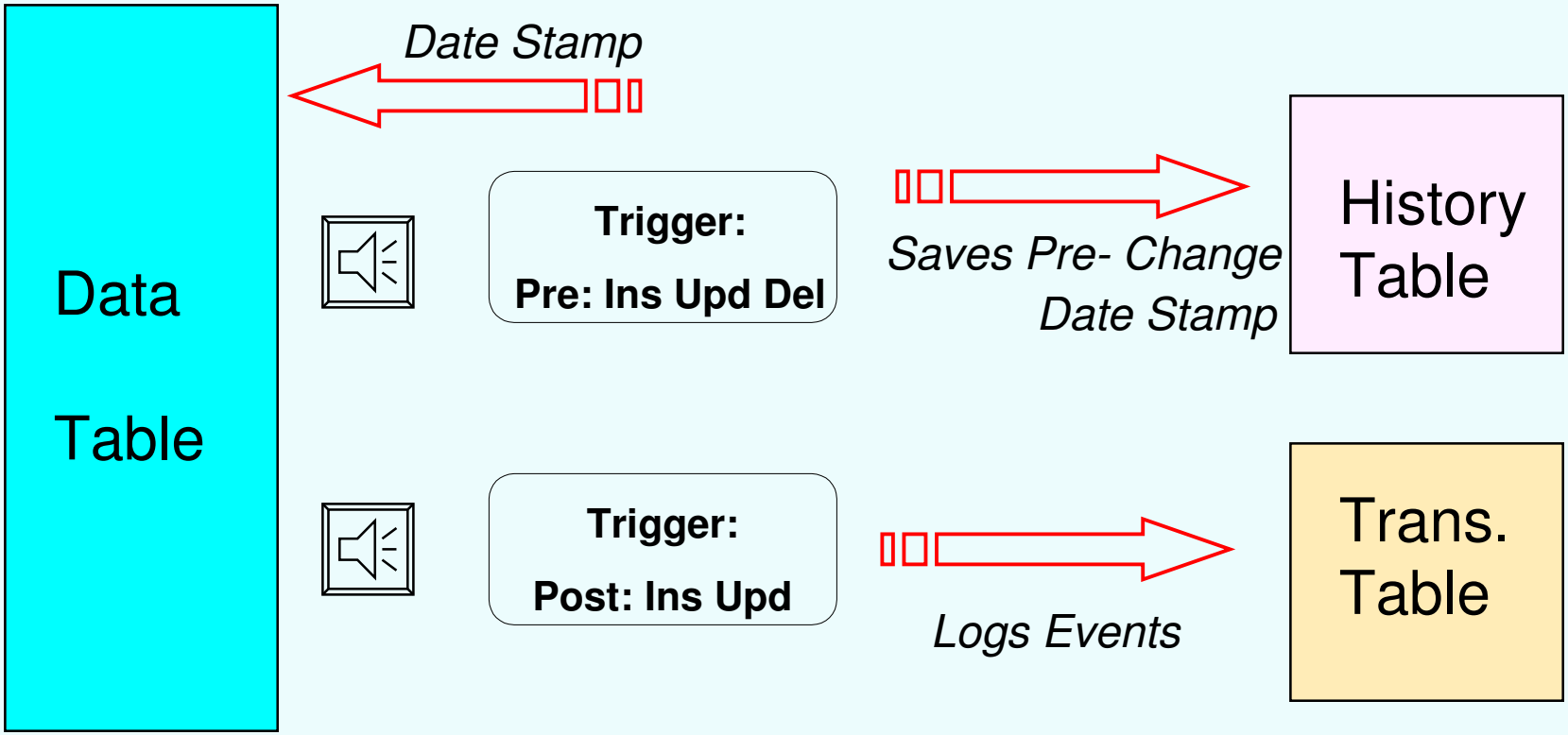
An Insert:


Notice that the History table is NOT involved in this transaction
Date Stamp
Data Table
Trigger: Ins Upd Del
History Table
Trigger: Post: Ins Upd
Logs Events
Trans. Table

Update Procedure:

- Old Row *Updated* in Data Table
  - Audit Trigger fires:
    - Copies pre-change row to History table
    - Timestamps pre-change row in History table
    - Timestamps update in Data Table
  - Transaction trigger fires
    - Populates Transaction Log Table
    - Captures *ROWID* from Data AND History Table

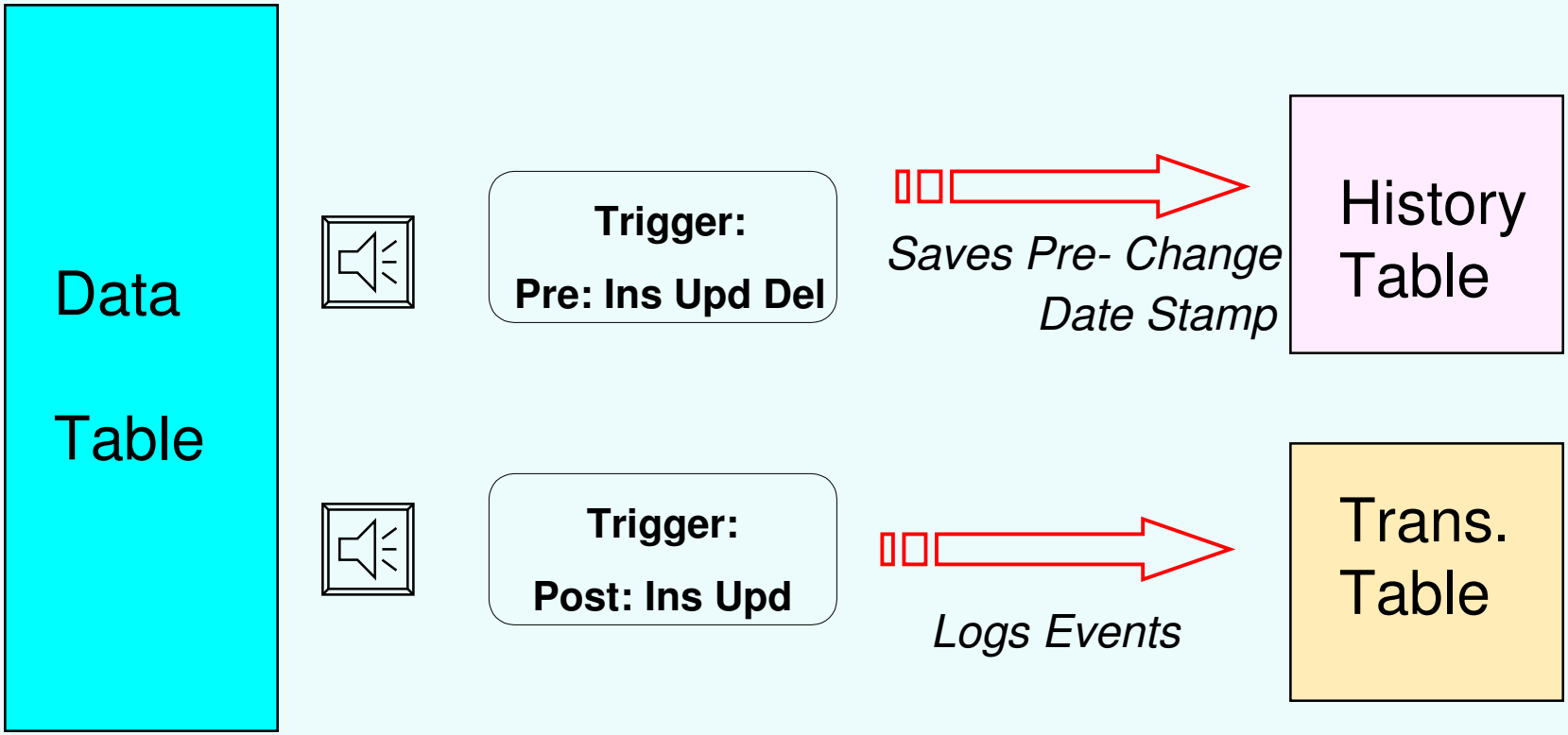
An Update:


Data
Table
Date Stamp
Trigger:
Pre: Ins Upd Del
Saves Pre- Change
Date Stamp
History
Table
Trigger:
Post: Ins Upd
Logs Events
Trans.
Table

Delete Procedure:

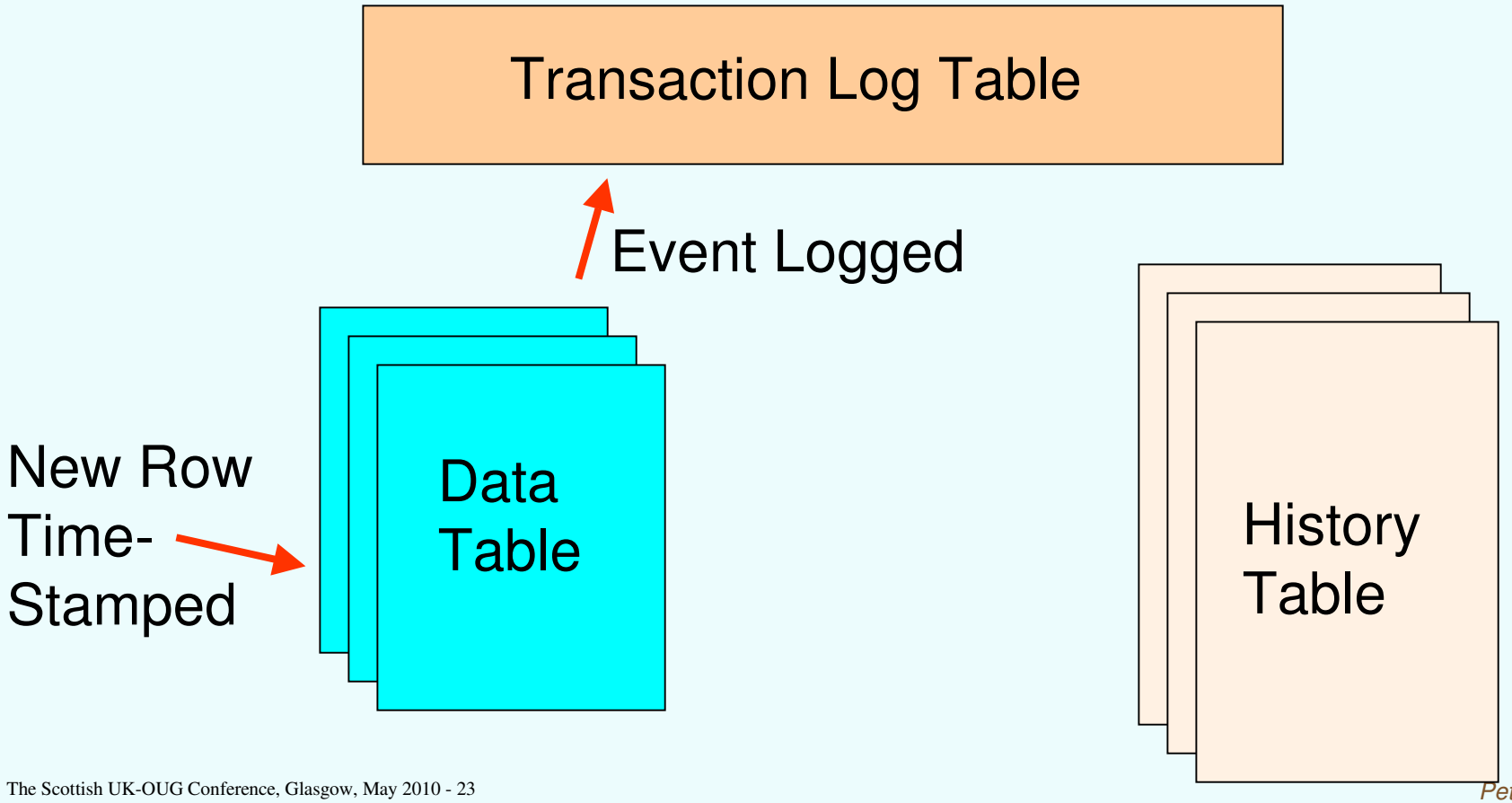
- Old Row *Deleted* from Data Table
  - Audit Trigger fires:
    - Copies pre-deleted row to History table
    - Timestamps pre-deleted row in History table
  - Audit Table Transaction Trigger fires:
    - Populates Transaction Log Table
    - Captures *ROWID* from History Table
  - Row Deleted

A Delete:


Data
Table
Trigger:
Pre: Ins Upd Del
Saves Pre- Change
Date Stamp
History
Table
Trigger:
Post: Ins Upd
Logs Events
Trans.
Table

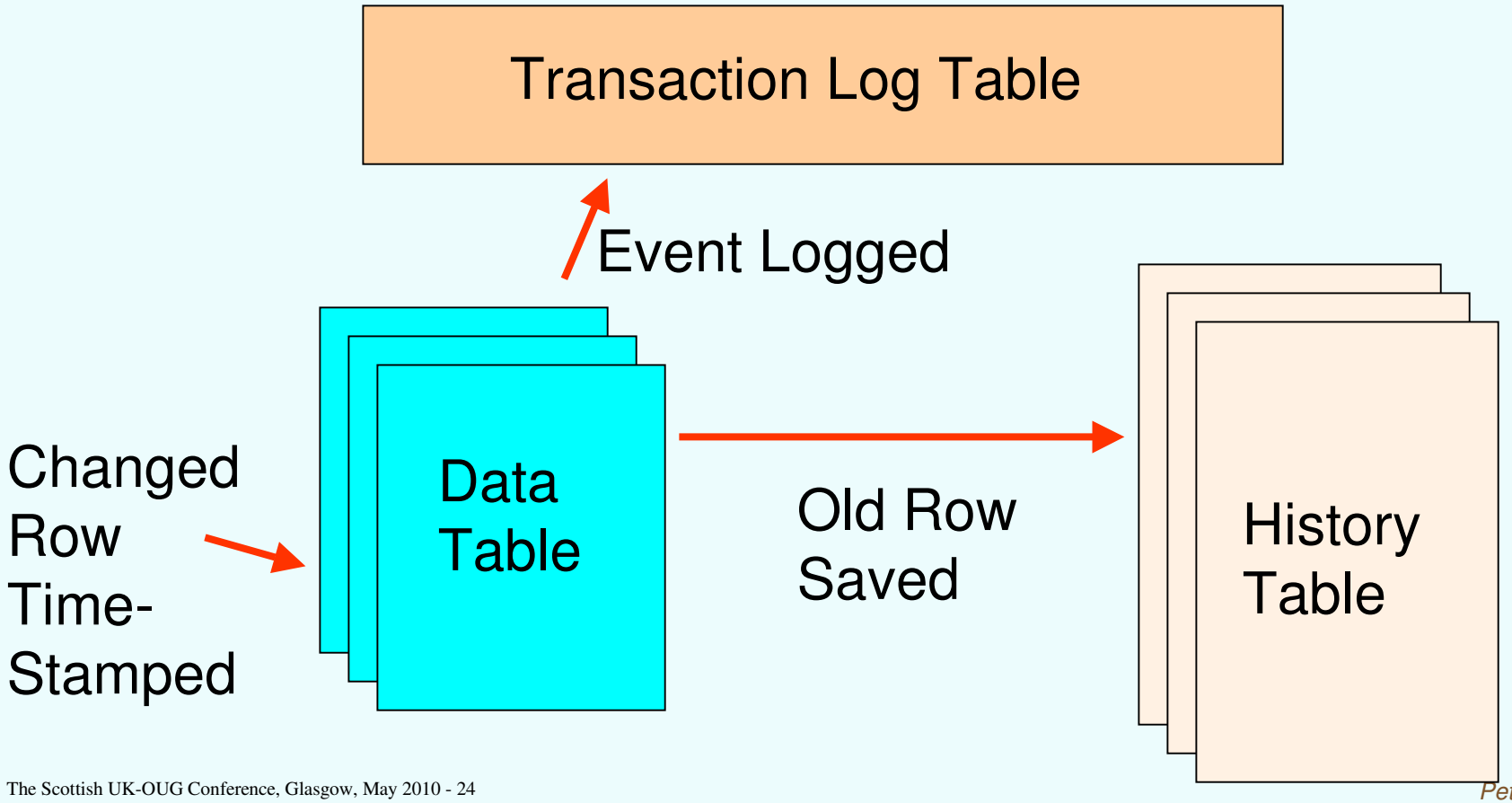
Structure in Summary

Insert:



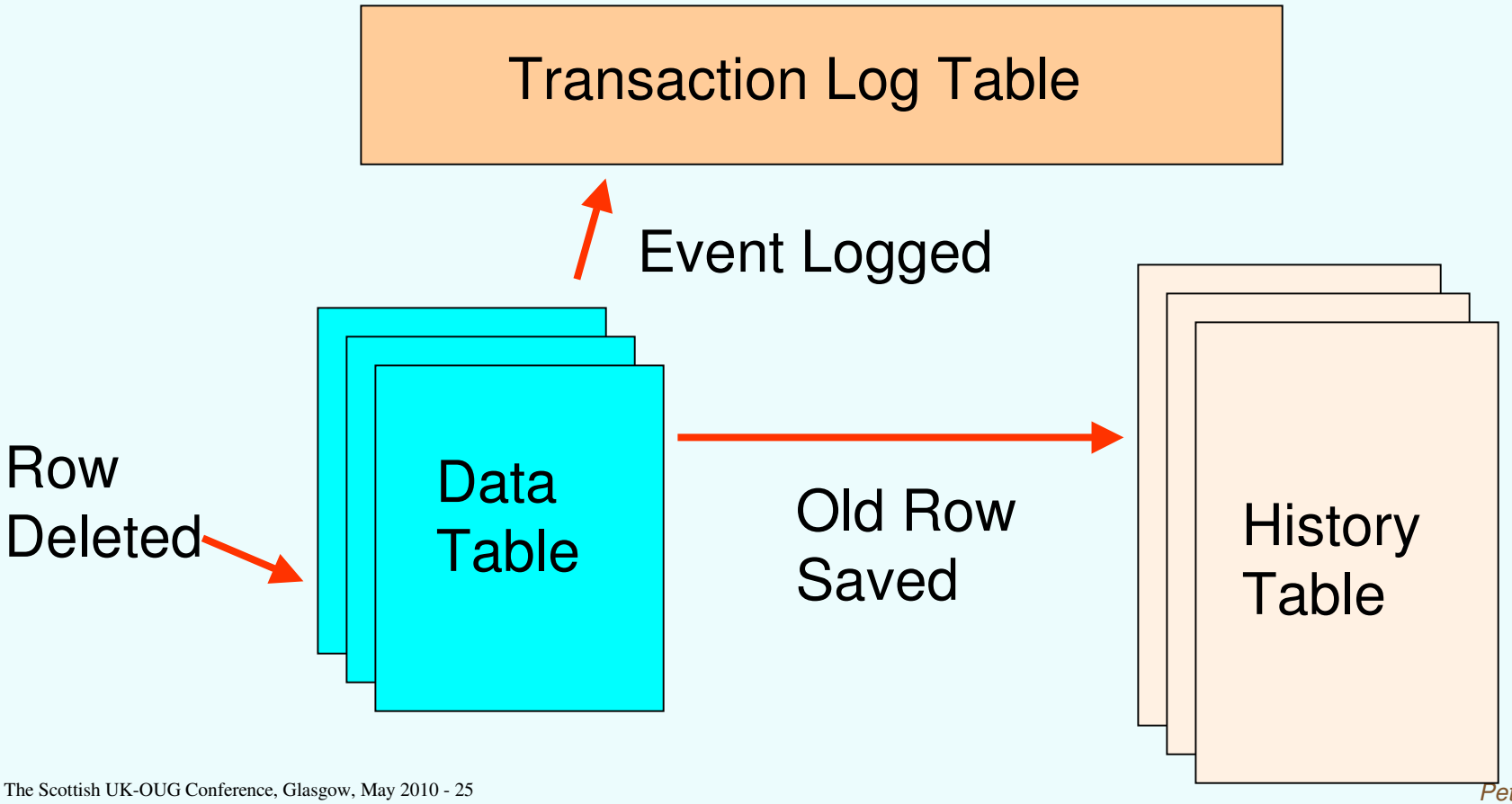
Structure in Summary

Update:

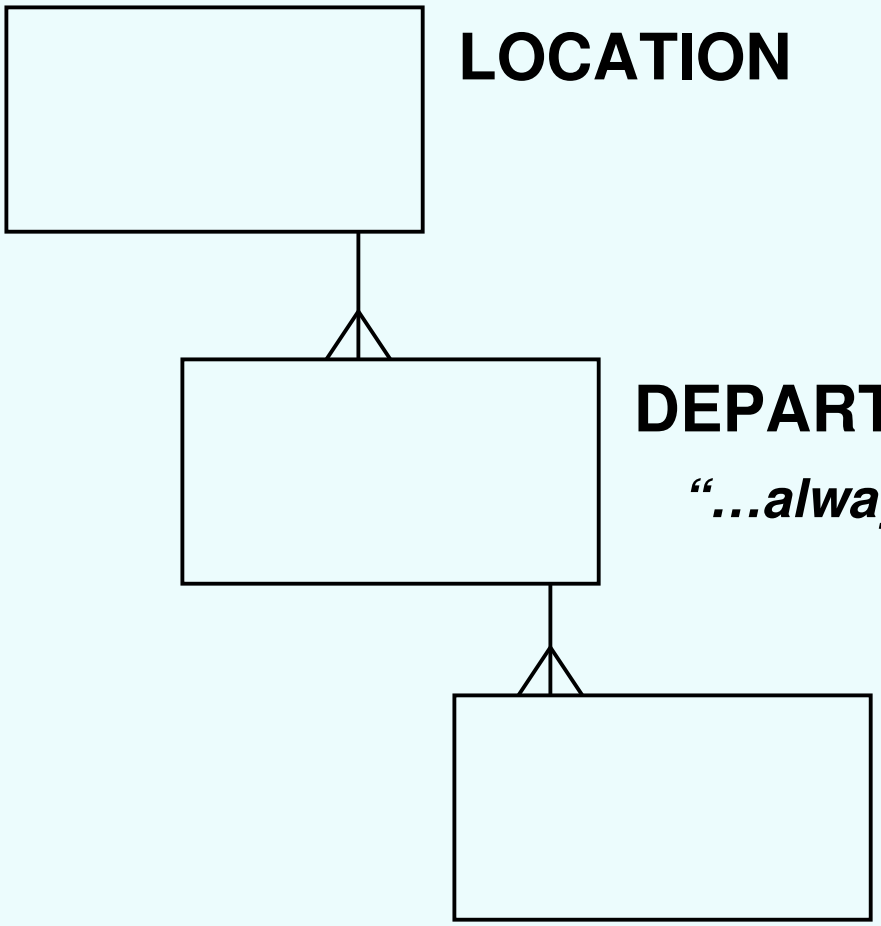


Structure in Summary

Delete:



The Demonstration Tables:



LOCATION

DEPARTMENT

"…always placed in a Location"

EMPLOYEE

"…always placed in a Department"

Table Structures (1)

The three demonstration tables have a predictable structure which is not of primary interest….

```
SQL> desc LOCATION
 Name                           Null?    Type
 ------------------------------ -------- ---------------
 LOCATION_ID                    NOT NULL CHAR(4)
 LOCAT_NAME                              VARCHAR2(30)
 DATE_ENTERED                   NOT NULL DATE
 USER_ENTERED                   NOT NULL VARCHAR2(10)
 DATE_UPDATED                            DATE
 USER_UPDATED                            VARCHAR2(10)
```

Table Structures (2)

```
SQL> desc DEPARTMENT
 Name                          Null?    Type
 ----------------------------- -------- -----------
 DEPT_ID                       NOT NULL CHAR(4)
 DEPARTMENT_NAME                        VARCHAR2(30
 LOCATION_ID      FK           NOT NULL CHAR(4)
 DATE_ENTERED                  NOT NULL DATE
 USER_ENTERED                  NOT NULL VARCHAR2(10
 DATE_UPDATED                           DATE
 USER_UPDATED                           VARCHAR2(10

SQL> desc EMPLOYEE
 EMP_ID                        NOT NULL NUMBER
 EMP_NAME                               VARCHAR2(30)
 DEPT_ID          FK           NOT NULL CHAR(4)
 DATE_ENTERED                  NOT NULL DATE
 USER_ENTERED                  NOT NULL VARCHAR2(10)
 DATE_UPDATED                           DATE
 USER_UPDATED                           VARCHAR2(10)
```

Structure of the Transaction Log Table

This is important…

```
SQL> desc trans_log
 Name                      Null?    Type
 ------------------------- -------- ----------------
 TABLE_NAME                         VARCHAR2(30)
 TRANS_FLAG                         CHAR(1)
 TRANS_DATE                         DATE
 ROW_ID                             ROWID
 PK_FIELDS                          VARCHAR2(2000)
 TRANS_SEQ                 NOT NULL NUMBER
 ROWID_FLAG                         CHAR(1)
 CTRLZ_FLAG                         CHAR(1)
 USER_ENTERED                       VARCHAR2(10)
 DATE_ENTERED                       DATE
 USER_UPDATED                       VARCHAR2(10)
 DATE_UPDATED                       DATE
```

The Transaction Log Table:

17 Inserts into

3 Tables

TRANS_SEQ	TRANS_DAT	TABLE_NAME	T	C	ROW_ID
1	09-DEC-03	LOCATION	I		AAABWJAABAAAEAaAAC
2	09-DEC-03	LOCATION	I		AAABWJAABAAAEAaAAA
3	09-DEC-03	LOCATION	I		AAABWJAABAAAEAaAAB
4	09-DEC-03	DEPARTMENT	I		AAABWKAABAAAEAiAAD
5	09-DEC-03	DEPARTMENT	I		AAABWKAABAAAEAiAAA
6	09-DEC-03	DEPARTMENT	I		AAABWKAABAAAEAiAAB
7	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAL
8	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAA
9	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAB
10	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAC
11	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAD
12	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAE
13	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAF
14	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAG
15	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAH
16	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAI
17	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAJ

The Transaction Log Table:

TRANS_SEQ	TRANS_DAT	TABLE_NAME	T	C	ROW_ID
1	09-DEC-03	LOCATION	I		AAABWJAABAAAEAaAAC
2	09-DEC-03	LOCATION	I		AAABWJAABAAAEAaAAA
3	09-DEC-03	LOCATION	I		AAABWJAABAAAEAaAAB
4	09-DEC-03	DEPARTMENT	I		AAABWKAABAAAEAiAAD
5	09-DEC-03	DEPARTMENT	I		AAABWKAABAAAEAiAAA
6	09-DEC-03	DEPARTMENT	I		AAABWKAABAAAEAiAAB
7	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAL
8	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAA
9	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAB
10	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAC
11	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAD
12	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAE
13	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAF
14	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAG
15	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAH
16	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAI
17	09-DEC-03	EMPLOYEE	I	Z	AAABWLAABAAADoaAAJ
18	09-DEC-03	EMPLOYEE	D	Z	AAABWgAABAAAD9iAAR

Transaction 17 Reversed - ctrl-z

The Transaction Log Table:

TRANS_SEQ	TRANS_DAT	TABLE_NAME	T	C	ROW_ID
1	09-DEC-03	LOCATION	I		AAABWJAABAAAEAaAAC
2	09-DEC-03	LOCATION	I		AAABWJAABAAAEAaAAA
3	09-DEC-03	LOCATION	I		AAABWJAABAAAEAaAAB
4	09-DEC-03	DEPARTMENT	I		AAABWKAABAAAEAiAAD
5	09-DEC-03	DEPARTMENT	I		AAABWKAABAAAEAiAAA
6	09-DEC-03	DEPARTMENT	I		AAABWKAABAAAEAiAAB
7	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAL
8	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAA
9	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAB
10	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAC
11	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAD
12	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAE
13	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAF
14	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAG
15	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAH
16	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAI
17	09-DEC-03	EMPLOYEE	I		AAABWLAABAAADoaAAJ
18	09-DEC-03	EMPLOYEE	D		AAABWgAABAAAD9iAAR
23	01-JAN-96	EMPLOYEE	I		AAABWLAABAAADoaAAI

Transaction 17 Restored

ctrl-y →

Live Demonstration...

In which the Following will be demonstrated:

- Start with a totally empty schema
- Build all tables, triggers, views, sequences
- Populate the core table
- Commit;
- Ctrl-Z (Reverse) several transactions
- Ctrl-Y (Restore) those transactions

demo……

Data tables

```
===============  Checking MASTER data table Employee:

    EMP_ID EMP_NAME                       DEPT
---------- ------------------------------ ----
        10 FRED                           D1
        20 BOB                            D1
        30 HENRY FROBISHER SMITH          D1
        40 JACK                           D2

===============  Checking MASTER HIST AUDIT data tables:

    EMP_ID EMP_NAME                       DEPT T THEDATE
---------- ------------------------------ ---- - ---------
        30 JOE                            D1   U 25-MAY-10

===============  Checking TRANS_LOG table:

 TRANS_SEQ TRANS_DAT TABLE_NAME                    T R C ROW_ID
---------- --------- ------------------------------ - - - ------------------
         1                                        I H   AAANLdAAEAAAADwAAA
         2                                        I H   AAANLdAAEAAAADwAAB
         3                                        I H   AAANLdAAEAAAADwAAC
         4                                        I H   AAANLdAAEAAAADwAAD
         5                                        U H   AAANLiAAEAAAAEIAAA
         6                                        U     AAANLdAAEAAAADwAAC
```

These rowids from table EMPLOYEE

But this rowid from table EMPLOYEE_HIST

Data tables

```
===============  Checking MASTER data table Employee:

    EMP_ID EMP_NAME                       DEPT
---------- ------------------------------ ----
        10 FRED                           D1
        20 BOB                            D1
        30 HENRY FROBISHER SMITH          D1
        40 JACK                           D2

===============  Checking MASTER HIST AUDIT data tables:

    EMP_ID EMP_NAME                       DEPT T THEDATE
---------- ------------------------------ ---- - ---------
        30 JOE                            D1   U 25-MAY-10

===============  Checking TRANS_LOG table:

 TRANS_SEQ TRANS_DAT TABLE_NAME                     T R C ROW_ID
---------- --------- ------------------------------ - - - ------------------
         1 25-MAY-10 EMPLOYEE                       I H   AAANLdAAEAAAADwAAA
         2 25-MAY-10 EMPLOYEE                       I H   AAANLdAAEAAAADwAAB
         3 25-MAY-10 EMPLOYEE                       I H   AAANLdAAEAAAADwAAC
         4 25-MAY-10 EMPLOYEE                       I H   AAANLdAAEAAAADwAAD
         5 25-MAY-10 EMPLOYEE                       U H   AAANLiAAEAAAAEIAAA
         6 25-MAY-10 EMPLOYEE                       U     AAANLdAAEAAAADwAAC
```

These rowids from table EMPLOYEE

Identical Rowids

continue demo…

Conclusions

- Very simple interface for the user
- Design respects all constraints
- Based on proven Backup / Restore / Replication
- All code in SQL*Plus and Triggers
- It Works!

End Note...

Thanks for your attendance!

Contact Details:

Peter Robson

peter.robson@justsql.com

www.justsql.com

(check www.justsql.com/html/publications for a paper
and PowerPoint file on this technique)