# Tactic

<u>Dependency</u>:
   For audits of data views and logins, the authentication and authorization functionality must be in place.

# Reporting & Recovery Capabilities

<u>Note</u>: A "record" in the requirements pertains to an object in the Sahana application, such as a person (including disaster victims, aid workers, volunteers, reporting person, missing persons, Sahana users), group, camp, organization.

### Chronological view

1. **Chronological view: Show chronological changes to a specific record**

**Sample screen**

## Chronological Changes For Missing Person

Name: Abdul Gandhi
Person UUID: 123456

| Time stamp | User making change | Affected table | Affected field | Type of change | Prior value | New value |
|---|---|---|---|---|---|---|
| 2005/11/15 22:01 | Anna Naan | person_status | p_uuid | Delete | 123456 | --- |
| 2005/11/15 22:01 | Anna Naan | person_status | isReliefWorker | Delete | 0 | --- |
| 2005/11/15 22:01 | Anna Naan | person_status | opt_status | Delete | Alive & Well | --- |
| 2005/11/15 22:01 | Anna Naan | person_status | updated | Delete | 2005/11/01 15:45 | --- |
| | | | | | | |
| 2005/11/10 10:15 | Anu Una | person_status | opt_status | Update | Missing | Alive & Well |
| 2005/11/10 10:15 | Anu Una | person_status | updated | Update | 2005/11/01 15:45 | 2005/11/10 10:15 |
| | | | | | | |
| 2005/11/01 15:45 | Anna Naan | person_status | p_uuid | Insert | --- | 123456 |
| 2005/11/01 15:45 | Anna Naan | person_status | isReliefWorker | Insert | --- | 0 |
| 2005/11/01 15:45 | Anna Naan | person_status | opt_status | Insert | --- | Missing |
| 2005/11/01 15:45 | Anna Naan | person_status | updated | Insert | --- | 2005/11/01 15:45 |

*Alternative implementation of "Update"*

| Time stamp | User making change | Affected table | Affected field | Type of change | Value |
|---|---|---|---|---|---|
| 2005/11/10 10:15 | Anu Una | person_status | opt_status | After Update | Alive & Well |
| 2005/11/10 10:15 | Anu Una | person_status | updated | After Update | 2005/11/10 10:15 |
| 2005/11/10 10:15 | Anu Una | person_status | opt_status | Before Update | Missing |
| 2005/11/10 10:15 | Anu Una | person_status | updated | Before Update | 2005/11/01 15:45 |

## 1.1 For an existing record

Initial: From search

    Input: (via search)
        From menu, click: [Show chronological changes to specific record]
        Search on record by name
        Select an existing record from list found

    Output:  In chronological order:
        Timestamp, Change type (insert, update, delete), Field changed, Data Before Change,
        Data After Change

Advanced: From any screen displaying record

    Input:
        From menu, click: [Show chronological changes to specific record]
        Search on record by name
        Select an existing record from list found

Output:  In chronological order:
Timestamp, Change type (insert, update, delete), Field changed, Data Before Change,
Data After Change


## 1.2  For a deleted record

### *Point in time view*

2. **Point in time view: Recreate view of a record at a specified point in time**

**Sample screen**

<div style="border:1px solid">

# Point In Time View For Missing Person

Name: Abdul Gandhi
Person UUID: 123456
View date: 2005/11/12

| Affected table | Affected field | Description | Value |
|---|---|---|---|
| person_status | p_uuid | Unique person ID | 123456 |
| person_status | isReliefWorker | 0=not relief worker | 0 |
| person_status | opt_status | Status | Alive & Well |
| person_status | updated | Last updated | 2005/11/01 15:45 |

</div>

2.1 **For an existing & deleted records**

    **2.3.1    View record at specified point in time**

    **2.3.2    Roll back record to specified point in time**

> There is a lot of complexity surrounding the ability to automatically restore a record to its state at a historical point in time. I would like to suggest that, for now, we concentrate on the ability to record the changes and report on them. If a record needs to be restored, this can be done manually using the existing application UI. The important thing is that the user is able to run a report to find out what the data values should be.

2.2 **For a deleted record**

    **2.3.3    View record just before it was deleted**

    **2.3.4    Roll back record to point just before it was deleted**

1. What did a certain record (such as: person, camp, group, org, etc.) look like at a specified point in time?

2. What changes have been made by a certain user during a certain time frame?

# 3    User Audit: Show all changes made by a certain user during a certain time frame

## 3.1 Show all logins for user

**Sample screen**

## Login Audit For A User

User: Anna Naan
User UUID: 234567
Logins between: 2005/10/01 and 2005/12/31

| Login timestamp |
|---|
| 2005/11/15 22:01 |
| 2005/11/10 15:45 |
| 2005/11/01 10:15 |

## 3.2 Show data viewed by certain user during a certain time frame

Caution: This type of audit has the potential of slowing down performance.  It is suggested that this level of logging only be turned on if appropriate.

**Sample screen**

## Data Viewed Audit By User

User: Anna Naan
User UUID: 234567
Logins between: 2005/10/01 and 2005/12/31

| Time stamp | Data for Person | Table | Field | Value |
|---|---|---|---|---|
| 2005/11/15 10:15 | Abdul Gandhi | person_status | opt_status | Alive & Well |
| 2005/11/15 10:15 | Abdul Gandhi | person_status | isReliefWorker | 0 |
|  |  |  |  |  |
| 2005/11/09 10:15 | Abdul Gandhi | person_status | opt_status | Missing |
| 2005/11/09 10:15 | Abdul Gandhi | person_status | isReliefWorker | 0 |

3.3 **Show all changes (inserts, updates, deletes) made by a certain user during a certain time frame**

Note: This view is similar to chronological changes by person (object), just filtered differently.

**Sample screen**

# Chronological Changes By A User

User: Anna Naan

| Time stamp | Affected person (object) | Affected table | Affected field | Type of change | Prior value | New value |
|---|---|---|---|---|---|---|
| 2005/11/15 22:01 | Abdul Gandhi | person_status | p_uuid | Delete | 123456 | --- |
| 2005/11/15 22:01 | Abdul Gandhi | person_status | isReliefWorker | Delete | 0 | --- |
| 2005/11/15 22:01 | Abdul Gandhi | person_status | opt_status | Delete | Alive & Well | --- |
| 2005/11/15 22:01 | Abdul Gandhi | person_status | updated | Delete | 2005/11/01 15:45 | --- |
| | | | | | | |
| 2005/11/01 15:45 | Abdul Gandhi | person_status | p_uuid | Insert | --- | 123456 |
| 2005/11/01 15:45 | Abdul Gandhi | person_status | isReliefWorker | Insert | --- | 0 |
| 2005/11/01 15:45 | Abdul Gandhi | person_status | opt_status | Insert | --- | Missing |
| 2005/11/01 15:45 | Abdul Gandhi | person_status | updated | Insert | --- | 2005/11/01 15:45 |

*General Audit*

4 **General Audit:**

4.1 **Show all logins during a specified time frame**

**Sample screen**

# Login Audit For All Users

Logins between: 2005/10/01 and 2005/12/31

| User | Login timestamp |
|---|---|
| Anna Naan | 2005/11/15 22:01 |
| Anu Una | 2005/11/10 15:45 |
| Anna Naan | 2005/11/01 10:15 |

4.2 **Show data viewed for a specified object (person, group, organization, etc.) during a certain time frame**

*Mass Loss*

## 5   Recover from loss of many records

The situation becomes more tricky if a lot of records need to be corrected. This might occur if, for example, a large number of records are somehow accidentally deleted. In this case, a technical person with SQL and ETL skills would need to be involved.

This may be more of an admin / backup function.

---

# Items to Log

DVR tables to track closely are:
     person_uuid
     identity_to_person
     users
     person_status
     contact
     location_details
     location
     pgroup
     person_to_pgroup
     person_details
     person_physical
     person_missing
     person_camp

But we probably want to also keep track of:
     camp
     camp_services
     camp_reg