NSCC

Disaster Scenario

How to recover missing records

Disclaimer:

This document should only be used as reference. In a real scenario, please test the steps suggested here before implementing it

Nascimento,Luiza
13/3/2019

Contents

Disaster Scenario

A user calls you and say that they delete some records by mistake three days ago and now they need the data back.

Recovery Plan

The disaster recovery plan should contain an analysis of the impact the disaster would have to the business. It should also contain communication plan, risk assessment, recovery procedure and more. With the help of this document, the DBA should be able of recovering the records deleted and mitigate the impact the mistake could cause to the business. Here are some steps it should be followed:

Asses the risk

The DBA team must evaluate the importance of the records missing and the impact a recovery would cause to decide the priority of solving this incident. With the help of the documentation available - DRP, source-to-target mapping, data dictionary or other document from the information governance program - the DBA should look for answers to the following questions:

- How many people could have been impacted and from what departments are they?
- Which systems/reports could have been impacted?
- Is the data from recent transactions or is it historical?
- Could this be a liability for the organization? (i.e. compliance reports were generated, and this mistake could cause issues with the regulatory agency?)

If documentation is not available, an alternative solution would be looking into the logs to find out who or what systems queried the tables from where the records where deleted in the last 3 days. This method could take a great amount of time and management must evaluate if spending time on this activity is essential at this moment. If it is known that the data is missing is highly important to the business, then the disaster should be treated as high priority.

After evaluating the impact of the missing records to the business, it is time to evaluate the impact of the recovery to the business. In some systems, the recovery can be seamless and cause zero or little impact. In other cases, it can be necessary to block the transactions in that table or schema for some time to fix the issue. To get these answers we need to know the architecture of the database and what kind of backup we have available.

In my example below, I used Oracle 12c with archive log mode enabled and the recovery of the deleted records was seamless.

Communication

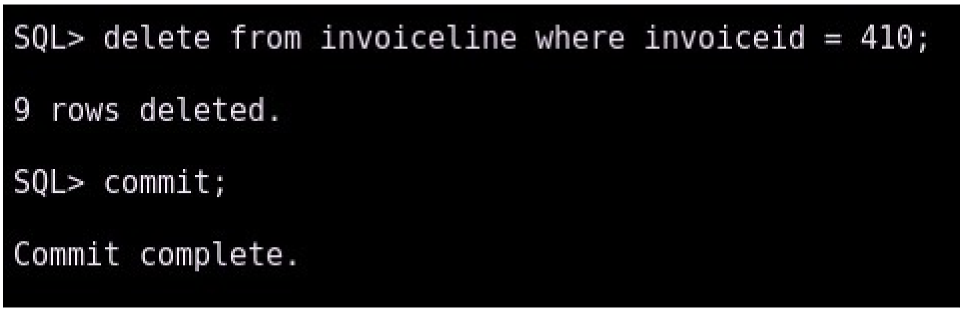
All the relevant stakeholders must be informed about possible problems in the reports generated in the last 3 days and give an estimated time to have the correct data available.

Refer to the recovery plan documentation to find out more about the communication plan and possible templates to communicate the issue.

Fix the problem

Before starting the recovery of deleted records, it is recommended to back up the database and datafiles, to mitigate any problems the recovery process may cause. You could also do a export dump to have an extra backup of the data before recovering the records.

To simulate the user's mistake, I deleted 9 rows of the table invoiceLine and then confirmed that the data was no longer available:

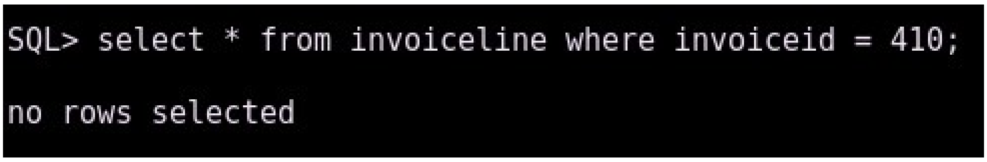


```
SQL> delete from invoiceline where invoiceid = 410;

9 rows deleted.

SQL> commit;

Commit complete.
```

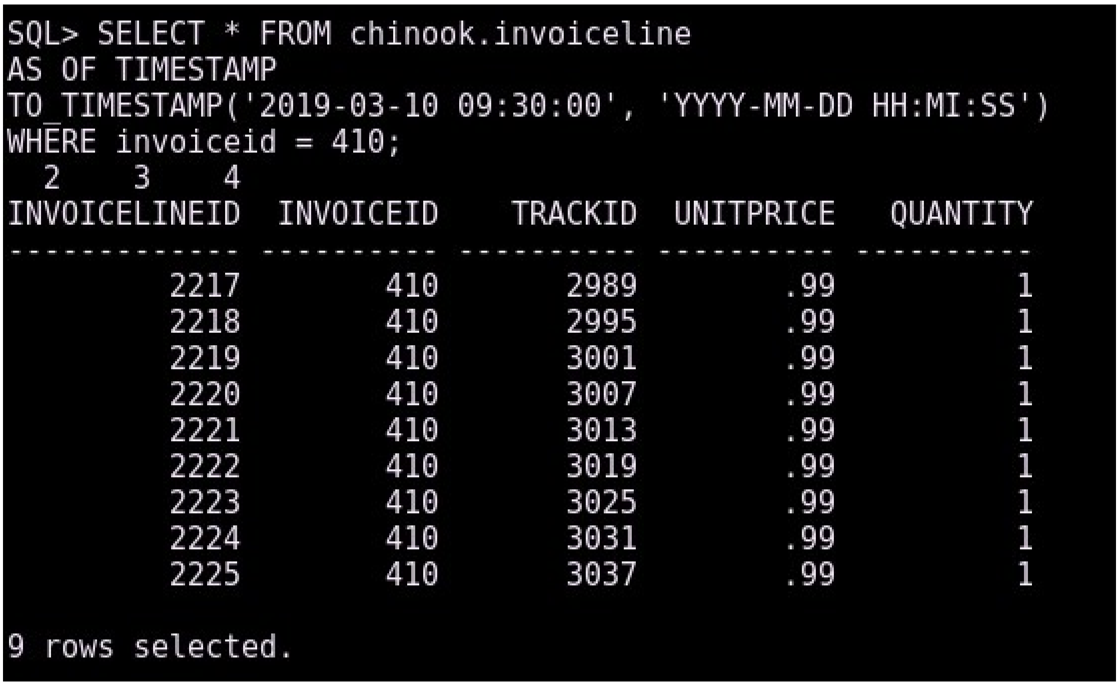


```
SQL> select * from invoiceline where invoiceid = 410;

no rows selected
```

In a real scenario, you would ask the user more information about the rows they deleted, day and time it happened and if you have an audit trail activated, you can confirm by checking the queries they ran to guarantee that the correct data was recovered, no more, no less.

Then I confirmed that the lines were available through the archive log, using the flashback command:

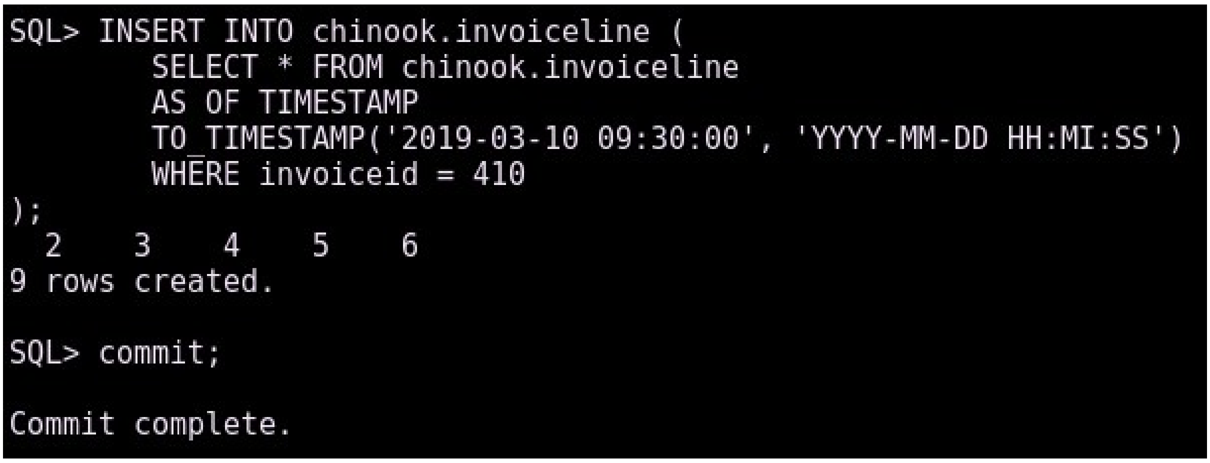


```
SQL> SELECT * FROM chinook.invoiceline
AS OF TIMESTAMP
TO_TIMESTAMP('2019-03-10 09:30:00', 'YYYY-MM-DD HH:MI:SS')
WHERE invoiceid = 410;
  2    3    4
INVOICELINEID  INVOICEID    TRACKID  UNITPRICE   QUANTITY
------------- ---------- ---------- ---------- ----------
         2217        410       2989        .99          1
         2218        410       2995        .99          1
         2219        410       3001        .99          1
         2220        410       3007        .99          1
         2221        410       3013        .99          1
         2222        410       3019        .99          1
         2223        410       3025        .99          1
         2224        410       3031        .99          1
         2225        410       3037        .99          1

9 rows selected.
```

At this point, you can validate with the logs and with the user if that's is the correct data to recover. It is important to get sign off from the business before inserting this data back to the table.

I then used the same flashback query to insert the data back to the table:



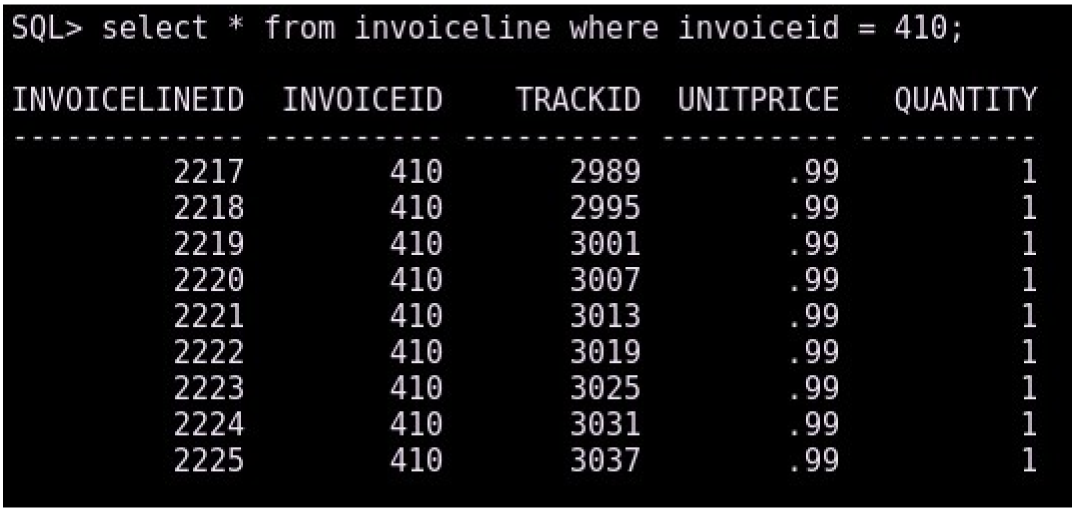
```
SQL> INSERT INTO chinook.invoiceline (
        SELECT * FROM chinook.invoiceline
        AS OF TIMESTAMP
        TO_TIMESTAMP('2019-03-10 09:30:00', 'YYYY-MM-DD HH:MI:SS')
        WHERE invoiceid = 410
);
  2    3    4    5    6
9 rows created.

SQL> commit;

Commit complete.
```

Validation

Querying the table invoiceLine I confirmed that the missing rows were there again. At this point we can get confirmation from the user if the problem has been fixed.



```
SQL> select * from invoiceline where invoiceid = 410;

INVOICELINEID  INVOICEID    TRACKID  UNITPRICE   QUANTITY
------------- ---------- ---------- ---------- ----------
         2217        410       2989        .99          1
         2218        410       2995        .99          1
         2219        410       3001        .99          1
         2220        410       3007        .99          1
         2221        410       3013        .99          1
         2222        410       3019        .99          1
         2223        410       3025        .99          1
         2224        410       3031        .99          1
         2225        410       3037        .99          1
```

Backup after the fix

After getting confirmation from the user, I would do a new incremental differential backup to make sure that all the logs and the new state are safe and sound.

End of task Communication

Using the communication plan from DRP and the information gathered from the risk assessment, you should inform all relevant stakeholders that the correct data is now available.

Root cause analysis and mitigation plan

After the recovery is completed and everybody is happy again, you should investigate why that happen and take actions to prevent that from happening again in the future. In this case, you must understand what made the user make that mistake, and here are some things to consider:

- **Audit privileges.** It is important to confirm if the user is enjoying of more permissions than they need. If that is the case, you could work with a business analyst to understand what privileges the user really needs.
- **Lack of knowledge.** Check with the team the user is from to confirm if they have all the knowledge they need to manipulate the data. You could offer them some training, if that is the case.
- **Lack of process.** It could be a problem of not having enough validation before committing to a delete command. You could suggest improving the deletion process by adding an extra validation.

- **Lack of automation.** Maybe the process of deleting records is done frequently, so implementing an automated process would mitigate the problems with human error.
- **Interface issues.** The interface may be poorly designed, like the button to save is to close to the one to cancel or it does not have a confirmation before deleting a record. In this case, you should engage the users to give more feedback on the interface and then engage the development team to fix the issues.

Report

The last thing to do is to create a report describing the problem, possible consequences of the problem, the actions taken to solve the problem and the consequences and suggestions to avoid that from happening again.

References

Auditing Database Activity. (2019, March 14). Retrieved from Oracle Help Center: https://docs.oracle.com/cd/E11882_01/server.112/e10575/tdpsg_auditing.htm#TDPSG50521

Using Oracle Flashback Technology. (2019, March 13). Retrieved from Oracle Help Center: https://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm#ADFNS618