

Fine Grained Auditing

Vijay Kumar Tiwari

IT Consultant, HCL America Inc, Texas, United States

How to cite this paper: Vijay Kumar Tiwari "Fine Grained Auditing" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-4 | Issue-1, December 2019, pp.1173, URL: www.ijtsrd.com/papers/ijtsrd29878.pdf



Copyright © 2019 by author(s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



INTRODUCTION

Through Fine-grained auditing was offered as a gritty audit option for SELECT statements from oracle version 9i. It was then improved in oracle 10g to also include DML. It was an enhancement, not as an auxiliary for standard auditing, but as a way to address audit supplies that explicitly specify situations for which an audit record needs to be created. Such supplies can sometimes be uttered based on data within certain columns or simply by which columns are being accessed. Using standard auditing you can specify what you want audited based on commands and objects.

Insight

FGA is carried through the DBMS_FGA package, and situation up FGA policies requires EXECUTE privileges on this package. The audit records are written to a different table than standard auditing or to operating system files. When written to a database table, they are written to SYS.FGA_LOG\$. You can then use the DBA_FGA_AUDIT_TRAIL view to look at the audit records.

One of the nice things with FGA is that you don't have a two-step initiation process. You don't need to set an initialization parameter, and you don't need to restart the database – you just have to define policies.

Express a FGA audit policy:

```
SQL> begin
dbms_fga.add_policy(
object_schema=>'TEST',
object_name=>'EMP_ACCOUNTS',
policy_name=>'EMP_SALARY'
);
end;
/
```

The audit policy created on the TEST.EMP_ACCOUNTS table. There is no condition specified in policy, all the DML operation will be audited by oracle.

FGA policies

You generate FGA policies using the DBMS_FGA.ADD_POLICY procedure.

```
SQL> begin
dbms_fga.add_policy(
object_schema=>'TEST',
object_name=>'EMP_ACCOUNTS',
policy_name=>'EMP_ACCESS',
audit_column=>'SAL,COMM');
end;
/
```

When you have more than one sensitive column you can control whether you want to audit access to any of the sensitive columns or if you only want to audit access that involves all of these columns. The default is to audit any access that involves any of these columns. To explicitly set the behavior use one of:

```
SQL> begin
dbms_fga.add_policy(
object_schema=>'SCOTT',
object_name=>'EMP_ACCOUNTS',
policy_name=>'EMP_ACCESS',
audit_column=>'SAL,COMM',
audit_column_opts=>DBMS_FGA.ANY_COLUMNS);
end;
/
```

Conclusion

FGA is very stretchy; the audit state is a PL/SQL expression which allows you to implement attractive any audit condition on DML and SELECT at a row level. A NULL as the audit state is interpreted as a null state and will match every row. Do not use a state such as 1=1 and do not use an empty string as a state.