بسمه تعالی

رویدادنگاری، ممیزی و جرم‌شناسی در پایگاه داده‌ی اوراکل

فهرست مطالب

۱ مقدمه ۳

۲ نحوه‌ی ثبت وقایع در پایگاه داده‌ی اوراکل ۴

۲-۱ انواع فایل‌های رویدادنگاری ۴

۲-۱-۱ فایل Redo log ۴

۲-۱-۲ فایل‌های ردیابی ۱۰

۲-۱-۳ فایل Alert Log ۱۱

۲-۱-۴ فایل‌های DDL Log ۱۱

۲-۱-۵ فایل Debug log ۱۲

۳ نحوه‌ی انجام ممیزی در پایگاه داده‌ی اوراکل ۱۳

۳-۱ انواع ممیزی‌ها در پایگاه داده‌ی اوراکل ۱۴

۳-۱-۱ ممیزی یکپارچه ۱۴

۳-۱-۲ ممیزی ریزدانه ۲۰

۴ ابزارهای تحلیل‌کننده‌ی ممیزی ۲۴

۴-۱ logMiner ۲۴

۴-۱-۱ تنظیمات LogMiner ۲۵

۴-۱-۲ چگونگی کار با LogMiner ۲۸

۴-۱-۳ آیا LogMiner ابزار مناسبی برای جرم‌شناسی است؟ ۳۰

۴-۲ ابزار EventLog Analyzer ۳۱

۴-۲-۱ ممیزی با ابزار Eventlog Analyzer ۳۲

۵ جمع‌بندی ۳۴

۶ منابع ۳۴

۱ مقدمه

امروزه پایگاه‌های داده در برنامه‌های کاربردی مختلف کاربرد دارند. در پایگاه‌های داده اطلاعات متنوع و بعضاً حساسی توسط سازمان‌ها ذخیره می‌شوند که تغییر و از دست رفتن آن‌ها می‌تواند عواقب جبران‌ناپذیری را برای ادامه‌ی حیات سازمان‌ها در پی داشته باشد. از این‌رو سازمان‌ها نیاز به رهگیری و نظارت بر تغییرات و فعالیت‌های درون پایگاه‌های داده دارند. در این راستا، در پایگاه‌های داده مختلف روش‌های متفاوتی برای نظارت، تهیه‌ی گزارش از وقایع رخ‌داده و ممیزی وجود دارد که می‌توانند به ره‌گیری فعالیت‌های درون پایگاه داده کمک کنند. همچنین ابزارهایی به‌منظور تحلیل فایل‌های گزارش پیاده‌سازی شده‌اند.

با توجه به اهمیت ثبت وقایع و ممیزی برای نظارت بر اتفاقات درون پایگاه‌های داده، در این گزارش پایگاه‌های داده اوراکل و روش‌های متفاوت ثبت وقایع و ممیزی در آن‌ها و ابزارهای تحلیل اطلاعات ثبت‌شده که برای جرم‌شناسی پایگاه داده[1] مفید هستند، موردبررسی قرار گرفته‌اند.

در بخش ۲، انواع فایل‌های رویدادنگاری[2] و گزینه‌های قابل تنظیم برای ثبت اطلاعات و نظارت بر رویدادها در پایگاه داده‌ی اوراکل موردبررسی قرار گرفته است. پس‌ازآن در بخش ۳، ممیزی در پایگاه داده‌ی اوراکل و انواع آن ازجمله ممیزی یکپارچه و ممیزی ریزدانه توضیح داده شده است. در ادامه در بخش ۴، ابزارهای تحلیل فایل‌های رویدادنگاری ازجمله LogMiner و Oracle Eventlog Analyzer که برای جرم‌شناسی پایگاه داده می‌توانند مفید باشند، معرفی شده‌اند. لازم به ذکر است که در طول گزارش مثال‌هایی که به صورت عملی بر روی پایگاه داده‌ی اوراکل اجرا شده‌اند، برای واضح‌تر شدن مطالب آورده شده‌اند.

[1] Database forensic

[2] Log file

۲ نحوه‌ی ثبت وقایع در پایگاه داده‌ی اوراکل

در این بخش انواع فایل‌های رویدادنگاری موردبررسی قرار گرفته است. همچنین گزینه‌های قابل تنظیم برای نحوه‌ی نظارت و ثبت وقایع مشخص شده‌اند. این بخش از گزارش بر روی Oracle Database 12c نسخه‌ی Enterprise تهیه شده‌است.

۲-۱ انواع فایل‌های رویدادنگاری

در جدول ۱ انواع فایل‌های رویدادنگاری در پایگاه داده‌ی اوراکل به‌اختصار توضیح داده‌شده‌اند.

جدول ۱ انواع فایل‌های ثبت در اوراکل

نوع فایل رویدادنگاری	اطلاعات ثبت‌شده در فایل
فایل Redo log	ثبت تمامی تغییرات پایگاه داده. برای عملیات بازیابی مورد استفاده قرار می‌گیرد.
فایل ردیابی	شامل داده های تشخیصی برای بررسی مشکلات
فایل Alert log	گزارشی از خطاها و پیغام‌ها به ترتیب زمان وقوع
فایل DDL log	شامل عبارات DDL اجراشده توسط پایگاه داده
فایل Debug log	ثبت مشکلات احتمالی

در ادامه هر یک از این فایل‌ها موردبررسی قرار گرفته است.

۲-۱-۱ فایل Redo log

مهمترین ساختار در اوراکل که می‌توان از آن برای عملیات بازیابی[۳] استفاده کرد، redo log است. Redo log شامل دو یا چندین فایل است که تمامی تغییراتی که در پایگاه داده رخ می‌دهد را ذخیره می‌کند. در فایل‌های

[3] Recovery

redo log، رکوردهای redo ثبت می‌شوند. رکورد Redo یا redo entry از گروهی از بردارهای تغییر[4] ساخته شده است. بردار تغییر، توضیحی در مورد تغییر اعمال‌شده بر روی یک بلاک[5] پایگاه داده است. به‌عنوان نمونه درصورتی‌که مقدار حقوق کارمندی در جدول تغییر داده شود، یک رکورد redo ایجاد می‌شود که شامل بردارهای تغییر است که تغییرات اعمال‌شده را توصیف می‌کند [۱]. فایل‌های redo log فایل‌های دودویی[6] هستند که برای خواندن آن‌ها می‌توان از ویرایش‌گرهای هگزادسیمال[7] استفاده کرد و یا با استفاده از ابزارهایی همچون LogMiner می‌توان به بررسی محتوای آن‌ها پرداخت.

Redo log اهداف زیر را دنبال می‌کند [۲]:

- فراهم آوردن مکانیزمی برای ثبت تغییرات پایگاه داده تا در هنگام وقوع مشکل برای رسانه، روشی برای بازیابی تراکنش‌ها وجود داشته باشد.
- اطمینان از اینکه هنگام بروز مشکل حتی اگر داده‌های تثبیت[8] شده در فایل‌های داده نوشته نشده‌اند، بتوان تراکنش‌های تثبیت شده را بازیابی کرد.
- فراهم آوردن امکان بررسی تراکنش‌های پایگاه داده از طریق ابزار LogMiner
- مورد استفاده قرار گرفتن برای تکثیر[9] داده‌ها توسط ابزارهای مختلف

۱-۱-۱-۲ چگونگی ثبت اطلاعات در فایل‌های redo log

Redo log شامل دو یا چندین فایل است؛ پایگاه داده حداقل به دو فایل نیاز دارد تا مطمئن باشد همیشه یک فایل برای نوشتن و فایلی برای آرشیو شدن وجود دارد. رکوردهای redo ابتدا در بافرهای حافظه‌ی ناحیه‌ی سراسری سیستم[10] (SGA) بافر می‌شوند. پردازه‌ی نویسنده‌ی رویداد[11] (LGWR) که در پس‌زمینه اجرا

[4] Change vectors
[5] Block
[6] Binary
[7] Hexadecimal
[8] Commit
[9] Replicate
[10] System global area
[11] Log writer

می‌شود، اطلاعات را از بافرها بر روی فایل‌های redo log ثبت می‌کند و به رکوردهای redo یک تراکنشِ تثبیت شده، یک شماره‌ی تغییر سیستمی[۱۲] (SCN) تخصیص می‌دهد. بدین ترتیب می‌توان رکوردهای redo در یک تراکنش تثبیت شده را تشخیص داد. زمانی که فایل جاری پر شود، LGWR بر روی فایل موجود بعدی شروع به نوشتن و ثبت اطلاعات می‌کند. زمانی که آخرین فایل موجود پر شود، LGWR به اولین فایل بازمی‌گردد و این چرخه ادامه دارد (شکل ۱) [۱].

پردازه‌ی LGWR محتویات بافرها را تحت یکی از شرایط زیر بر روی فایل‌های redo log می‌نویسد [۲]:

- اجرای دستور COMMIT
- پس از سه ثانیه
- یک سوم بافر پر شده باشد
- یک مگابایت از بافر پر شده باشد
- Redo log فایل موجود پر شده باشد و شروع به ثبت در فایل بعدی کند.

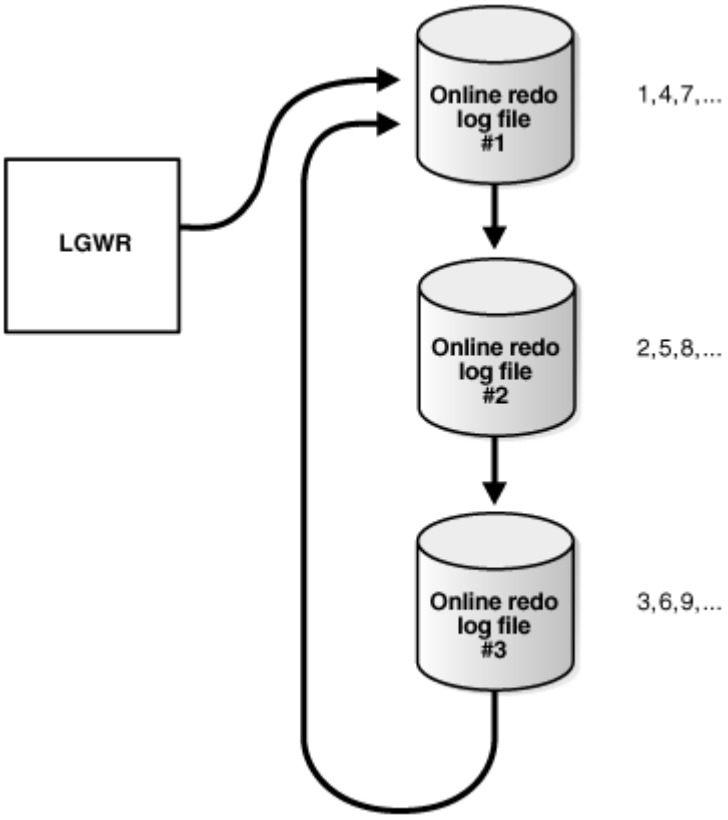
پایگاه داده‌ی اوراکل این امکان را می‌دهد که پیش از آنکه رکوردها مجددا بر روی فایل‌های redo log نوشته شوند، ابتدا آرشیو شوند به این فرآیند آرشیو کردن[۱۳] می‌گویند. درصورتی‌که پایگاه داده در مد ARCHIVELOG کار کند، یک فایل پر شده ابتدا باید آرشیو شود و مجددا روی آن نوشته شود [۱]. با استفاده از دستور زیر می‌توان متوجه شد که پایگاه داده در مد ARCHIVELOG یا NOARCHIVELOG فعالیت می‌کند:

```
SELECT LOG_MODE FROM V$DATABASE;
```

پایگاه داده‌ی اوراکل در یک زمان تنها از یک فایل redo log برای ذخیره‌ی رکوردهای redo استفاده می‌کند. به فایلی که LGWR به صورت فعال بر روی آن می‌نویسد فایل جاری می‌گویند [۱].

[12] System change number

[13] Archiving



شکل ۱ نحوه نوشتن پروسه LGWR بر روی فایل‌های redo log

با استفاده از دستور زیر می‌توان مسیر فایل‌های redo log را پیدا کرد:

```
SELECT * FROM v$logfile;
```

خروجی دستور بالا در شکل ۲ نشان داده شده است.

	GROUP#	STATUS	TYPE	MEMBER	IS_RECOVERY_DEST_FILE	CON_ID
1	3	(null)	ONLINE	C:\USERS\SEPIDEH\DESKTOP\ORACLE\ORADATA\ORCL\REDO03.LOG	NO	0
2	2	(null)	ONLINE	C:\USERS\SEPIDEH\DESKTOP\ORACLE\ORADATA\ORCL\REDO02.LOG	NO	0
3	1	(null)	ONLINE	C:\USERS\SEPIDEH\DESKTOP\ORACLE\ORADATA\ORCL\REDO01.LOG	NO	0

شکل ۲ خروجی دید v$logfile

خروجی دستور زیر و دستور بالا مشخص می‌کند که فایل جاری که LGWR روی آن می‌نویسد، کدام است:

```
SELECT * FROM v$log;
```

به‌عنوان نمونه با توجه به خروجی دستور فوق که در شکل ۳ نشان داده شده است، فایل REDO03.LOG فایل جاری است.

GROUP#	THREAD#	SEQUENCE#	BYTES	BLOCKSIZE	MEMBERS	ARCHIVED	STATUS	FIRST_CHANGE#	FIRST_TIME	NEXT_CHANGE#	NEXT_TIME
1	1	10	209715200	512	1	YES	INACTIVE	2473388	29-JUL-17	2573539	29-JUL-17
2	1	11	209715200	512	1	YES	INACTIVE	2573539	29-JUL-17	2702882	30-JUL-17
3	1	12	209715200	512	1	NO	CURRENT	2702882	30-JUL-17	18446744073709551615	(null)

شکل ۳ خروجی دید v$log

۲-۱-۱-۲ گزینه‌های قابل تنظیم در نحوه‌ی ثبت اطلاعات

در این قسمت گزینه‌های متفاوتی که برای ثبت اطلاعات در فایل‌های redo log وجود دارند توضیح داده می‌شوند.

۱. گزینه‌های LOGGING، NOLOGGING و FORCE LOGGING

با استفاده از گزینه‌های LOGGING و NOLOGGING می‌توان مشخص کرد که عملیات DML مشخصی در فایل redo log، ثبت شود یا خیر. این دو گزینه را می‌توان در عبارات زیر به‌کار برد [۵]:

- CREATE TABLE و ALTER TABLE
- CREATE INDEX و ALTER INDEX
- CREATE MATERIALIZED VIEW و ALTER MATERIALIZED VIEW
- CREATE TABLESPACE و ALTER TABLESPACE

درصورتی‌که LOGGING در عبارات بالا مشخص شود، ایجاد شیئ پایگاه داده و درج در آن شیئ، باعث ایجاد رکورد مربوط به آن رویداد، در فایل redo log می‌شود. در مقابل آن درصورتی‌که NOLOGGING در عبارات بالا به کار رود، به‌عنوان نمونه در CREATE TABLE … AS SELECT باعث می‌شود که حداقل اطلاعات redo در طول ایجاد جدول تولید شود. در نتیجه فضای کمتری توسط فایل‌های redo log اشغال می‌شود و همچنین زمان تولید جدول کاهش می‌یابد [۱].

همان‌طور که ذکر شد، به برخی از عبارات DDL مانند CREATE TABLE، می‌توان گزینه NOLOGGING را اضافه کرد. بدین ترتیب در برخی از عملیات پایگاه داده، رکورد redo در فایل‌های redo

log تولید نمی‌شود؛ لذا افزودن گزینه NOLOGGING، اثرات منفی بر روی بازیابی رسانه و پایگاه داده standby[۱۴] دارد [۱].

پایگاه داده‌ی اوراکل این امکان را فراهم می‌کند که حتی اگر NOLOGGING در عبارات DDL مشخص شده است، باز هم رکوردهای redo نوشته شوند، بدین منظور از FORCE LOGGING استفاده می‌شود. برای اینکه پایگاه داده در حالت FORCE LOGGING قرار گیرد، در عبارت CREATE DATABASE باید FORCE LOGGING اضافه شود. همچنین می‌توان پایگاه داده را پس از ایجاد با عبارت ALTER DATABASE و به‌کارگیری FORCE LOGGING در این حالت قرار داد [۱].

۲. رویدادنگاری تکمیلی

به صورت پیش‌فرض اطلاعات مختصری در فایل‌های redo log، ثبت می‌شود. درصورتی‌که نیاز به ثبت داده‌های بیشتری باشد، باید پیش از آن‌که فایل‌های رویدادنگاری تولید شوند، از رویدادنگاری تکمیلی استفاده کرد. رویدادنگاری تکمیلی دارای دو سطح است [۳]:

- **پایگاه داده:** سطح پایگاه داده انواع رویدادنگاری تکمیلی را فراهم می‌کند ازجمله:
  - حداقل رویدادنگاری تکمیلی[۱۵]: حداقل رویدادنگاری تکمیلی، بار قابل‌توجهی بر روی پایگاه داده برای تولید فایل‌های redo log وارد نمی‌کند.
  - Identification key logging: با استفاده از identification key logging، می‌توان اطلاعات بیشتری را ثبت کرد. به‌عنوان‌مثال، زمانی که سطری به‌روز می‌شود، تمامی ستون‌های آن سطر در فایل redo log قرار گیرند.
- **جدول:** رویدادنگاری تکمیلی در سطح جدول مشخص می‌کند که در سطح جدول از کدام‌یک از ستون‌های جدول به‌طور موثر رکوردهای مربوط به رویداد تهیه شود.

به‌منظور آزمون رویدادنگاری تکمیلی عبارات زیر اجرا شده‌اند.

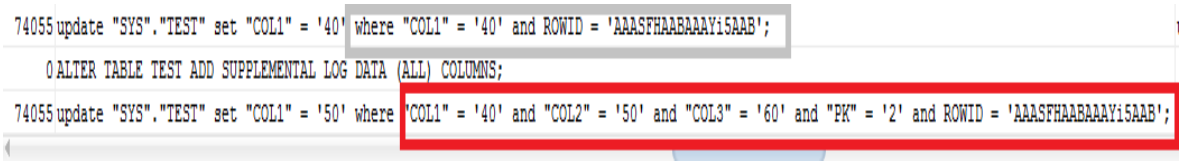
---

[۱۴] پایگاه داده standby، پایگاه داده‌ی کپی شده‌ای است که از فایل‌های پشتیبان پایگاه داده اصلی ایجادشده است. با قرار دادن فایل‌های redo log آرشیوشده از پایگاه داده اصلی بر روی پایگاه داده standby، می‌توان هر دو را با هم همگام کرد.

[15] Minimal supplemental logging

```
CREATE TABLE test (col1 NUMBER, col2 NUMBER, col3 NUMBER, pk NUMBER, CONSTRAINT test_pk PRIMARY KEY (pk));
INSERT INTO test VALUES(10,20,30,1);
INSERT INTO test VALUES (40,50,60,2);
UPDATE test SET COL1 = 40 WHERE PK=2;
ALTER TABLE TEST ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;
UPDATE test SET COL1 = 50 WHERE PK=2;
SELECT * FROM v$logmnr_contents WHERE TABLE_NAME='TEST';
```

زمانی که رویدادنگاری تکمیلی بر روی جدول فعال نشده باشد و به‌روزرسانی بر روی سطری از جدول انجام شود، اطلاعات کمتری در ستون SQL_REDO نمایش داده می‌شود؛ ولی با فعال شدن رویدادنگاری تکمیلی کلیه ستون‌های مربوط به سطر و مقادیر آن‌ها به نمایش درمی‌آیند (شکل ۶).



شکل ۴ خروجی دید v$logmnr_contents

۲-۱-۲ فایل‌های ردیابی

یک فایل ردیابی[16] فایلی است، شامل داده‌های تشخیصی که برای بررسی مشکلات مورد استفاده قرار می‌گیرد. هر کارگزار و فرایند[17] پس‌زمینه آن، می‌تواند در فایل ردیابی مرتبط با خود اطلاعاتی بنویسد. یک فرایند با شناسایی یک خطای داخلی، اطلاعاتی در مورد خطای رخ داده شده در فایل ردیابی خود می‌نویسد. برخی از اطلاعات نوشته شده در فایل ردیابی برای مدیر پایگاه داده و سایر اطلاعات برای سرویس‌های پشتیبانی اوراکل مفید هستند [۱]. دید V$DIAG_INFO مسیر فایل ردیابی با فرمت XML را با پارامتر Diag Trace مشخص می کند (شکل ۴).

[16] Trace

[17] Process

۲-۱-۳ فایل Alert Log

Alert log فایلی با فرمت XML است که در آن گزارشی از خطاها و پیغام‌ها به ترتیب زمان وقوع، ثبت شده است. برای پایگاه داده، Alert log شامل پیغام‌های زیر است [۱]:

- خطاهای بحرانی (حوادث)
- عملیات مدیریتی همچون راه‌اندازی و متوقف کردن پایگاه داده، بازیابی پایگاه داده، ایجاد یا حذف فضای جدول
- سایر رویدادهای پایگاه داده

دید V$DIAG_INFO مسیر فایل alert log با فرمت XML را با پارامتر Diag Alert مشخص می‌کند (شکل ۴).

```
SELECT * FROM V$DIAG_INFO;
```

ery Result

SQL | All Rows Fetched: 11 in 0.031 seconds

	INST_ID	NAME	VALUE	CON_ID
1	1	Diag Enabled	TRUE	0
2	1	ADR Base	C:\USERS\SEPIDEH\DESKTOP\ORACLE	0
3	1	ADR Home	C:\USERS\SEPIDEH\DESKTOP\ORACLE\diag\rdbms\orcl\orcl	0
4	1	Diag Trace	C:\USERS\SEPIDEH\DESKTOP\ORACLE\diag\rdbms\orcl\orcl\trace	0
5	1	Diag Alert	C:\USERS\SEPIDEH\DESKTOP\ORACLE\diag\rdbms\orcl\orcl\alert	0
6	1	Diag Incident	C:\USERS\SEPIDEH\DESKTOP\ORACLE\diag\rdbms\orcl\orcl\incident	0
7	1	Diag Cdump	C:\Users\Sepideh\Desktop\oracle\diag\rdbms\orcl\orcl\cdump	0
8	1	Health Monitor	C:\USERS\SEPIDEH\DESKTOP\ORACLE\diag\rdbms\orcl\orcl\hm	0
9	1	Default Trace File	C:\USERS\SEPIDEH\DESKTOP\ORACLE\diag\rdbms\orcl\orcl\trace\orcl_ora_2844.trc	0
10	1	Active Problem Count	2	0
11	1	Active Incident Count	44	0

شکل ۵ یافتن محل فایل‌های رویدادنگاری

۲-۱-٤ فایل‌های DDL Log

DDL Log فایلی است با فرمت و رفتاری مشابه alert log با این تفاوت که این فایل شامل عبارات DDL اجراشده توسط پایگاه داده است. درصورتی‌که پارامتر ENABLE_DDL_LOGGING مقدار TRUE داشته باشد، عبارات DDL در فایل DDL log ثبت می‌شوند. مقدار این پارامتر به صورت پیش‌فرض FALSE است [۱]. با دستور زیر وضعیت این پارامتر قابل بررسی است:

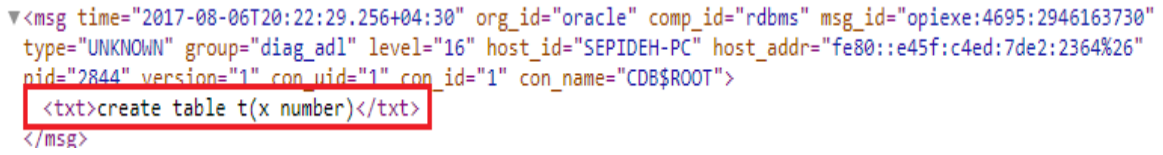
```
SELECT * FROM V$PARAMETER WHERE name = 'enable_ddl_logging';
```

DDL log شامل دو فایل با فرمت متنی و XML است. این فایل در مسیر ADR home (شکل ۴) در زیر دایرکتوری log/ddl قرار دارد. به‌منظور فعال کردن پارامتر ENABLE_DDL_LOGGIN از دستور زیر استفاده می‌شود:

```
ALTER SYSTEM SET enable_ddl_logging=true;
```

به‌عنوان نمونه با اجرای عبارت زیر در فایل DDL log اطلاعاتی مشابه شکل ۵ ذخیره می‌شود.

```
CREATE TABLE t(x NUMBER)
```



شکل ۶ اطلاعات ذخیره شده در فایل DDL log

۵-۱-۲ فایل Debug log

هر مؤلفه[18] در پایگاه داده‌ی اوراکل می‌تواند شرایط[19]، وضعیت‌ها[20] یا رویداد[21]هایی که به‌طور معمول با آن‌ها مواجه نمی‌شویم؛ ولی مؤلفه مانع آن‌ها نمی‌شود را تشخیص دهد. مؤلفه می‌تواند برای این‌گونه شرایط، وضعیت‌ها یا رویدادها هشداری تولید کند. این هشدارها در فایل Debug log ثبت می‌شوند. این دسته از هشدارها به قدری مهم نیستند که در فایل Alert log نوشته شوند. قالب و رفتار debug log مشابه alert log است با این تفاوت که مشکلات احتمالی در آن ثبت می‌شود.

[18] Component
[19] Conditions
[20] States
[21] Event

۳ نحوه‌ی انجام ممیزی در پایگاه داده‌ی اوراکل

ممیزی[۲۲]، نظارت و ثبت فعالیت‌های یک کاربر پایگاه داده است. ممیزی با اهداف زیر انجام می‌شود [۶]:

- اقدامات انجام‌شده بر روی یک شِماء، جدول، سطر یا محتوای مشخص قابل نظارت است.
- کاربران را از انجام اقدامات نامناسب بازمی‌دارد.
- امکان بررسی فعالیت‌های مشکوک وجود دارد.
- درصورتی‌که کاربر غیرمجازی مجوزهای بیش‌ازحد انتظار داشته باشد، یعنی بتواند داده‌ها را تغییر دهد یا حذف کند؛ می‌توان متوجه این موضوع شد و مجوزهای کاربر را مجددا بازنگری کرد.
- می‌توان مشکلات مربوط به پیاده‌سازی کنترل دسترسی یا مجوزها را شناسایی کرد. به‌عنوان نمونه می‌توان خط‌مشی[۲۳]های ممیزی را طوری تنظیم کرد که رکوردی از داده‌های حفاظت‌شده ثبت نشود. به‌عنوان‌مثال، درصورتی‌که رکوردهایی شامل داده‌های حفاظت‌شده تولید شوند، بیانگر این است که کنترل‌های امنیتی، به‌درستی پیاده‌سازی و تنظیم‌نشده‌اند.
- برای نیازمندی‌هایی که در کاربردهای خاص وجود دارد، می‌توان خط‌مشی‌های ممیزی مشخصی تنظیم کرد.
- نظارت و جمع‌آوری داده‌ها در مورد فعالیت‌های مشخصی در پایگاه داده قابل انجام است. به‌عنوان نمونه مدیر پایگاه داده می‌تواند اطلاعات آماری در مورد جداولی که به‌روزرسانی می‌شوند همچون تعداد کاربرانی که به‌طور هم‌زمان از سامانه استفاده می‌کنند را در زمان اوج مصرف جمع‌آوری کند.

دو نقش زیر در ممیزی وجود دارند [۶]:

- نقش AUDIT_ADMIN: دارای مجوز برای تنظیم خط‌مشی‌های ممیزی و مشاهده و تجزیه‌وتحلیل داده‌های جمع‌آوری‌شده است.
- نقش AUDIT_VIEWER: دارای مجوز برای مشاهده و تحلیل داده‌های ممیزی شده است.

---

[22] Auditing

[23] Policy

۳-۱ انواع ممیزی‌ها در پایگاه داده‌ی اوراکل

در این بخش انواع ممیزی ازجمله ممیزی یکپارچه و ممیزی ریزدانه در پایگاه داده‌ی اوراکل توضیح داده شده‌اند.

۳-۱-۱ ممیزی یکپارچه

با استفاده از خط‌مشی‌های ممیزی یکپارچه و عبارت AUDIT می‌توان از انواع فعالیت‌ها، ممیزی گرفت. اقداماتی که می‌توان از آن‌ها ممیزی گرفت شامل موارد زیر است [۷]:

- حساب‌های کاربری، نقش‌ها و مجوزهای کاربران
- اقدامات بر روی اشیاء به‌عنوان نمونه حذف یک جدول یا اجرای یک رویه

بدین منظور با توجه به هدف ممیزی از موارد زیر استفاده می‌شود [۷]:

- **خط‌مشی‌های ممیزی یکپارچه:** یک خط‌مشی ممیزی یکپارچه، یک گروه از تنظیمات ممیزی است که این امکان را فراهم می‌کند تا از جنبه‌ای مشخص از رفتار کاربر در پایگاه داده ممیزی گرفت. برای تعریف خط‌مشی می‌توان از عبارت CREATE AUDIT POLICY استفاده کرد. خط‌مشی تعریف‌شده می‌تواند به‌سادگی ثبت فعالیت‌های یک کاربر باشد و یا پیچیده و دارای شرایط خاصی باشد. درآن‌واحد می‌توان بیش از یک خط‌مشی تعریف کرد.
- **عبارات AUDIT و NOAUDIT:** با استفاده از عبارات AUDIT و NOAUDIT می‌توان یک خط‌مشی را فعال یا غیرفعال کرد. همچنین با این عبارت‌ها می‌توان کاربرانی را برای اعمال خط‌مشی بر روی آن‌ها افزود و یا حذف کرد.

در ادامه، مثالی برای بیان نحوه‌ی اعمال نظارت بر روی عبارت‌های SELECT بر روی جدول TEST شرح داده می‌شود. این مثال بر روی پایگاه داده‌ی اوراکل در سیستم‌عامل ویندوز انجام‌شده است و شامل مراحل زیر است [۶،۷].

۱. در گام اول فعال بودن ممیزی یکپارچه بررسی می‌شود و در صورت غیرفعال بودن، فعال می‌گردد. به این منظور باید اقدامات زیر انجام شوند:

- ابتدا باید با کاربر SYS و با مجوز مدیریتی SYSDBA وارد شد.

- با استفاده از پرسمان زیر فعال یا غیرفعال بودن ممیزی یکپارچه بررسی گردد:

```
SELECT VALUE FROM V$OPTION WHERE PARAMETER = 'Unified Auditing';
```

درصورتی‌که خروجی پرسمان بالا TRUE باشد، به گام دوم رفته، در غیر این صورت مراحل بعدی ادامه پیدا می‌کنند.

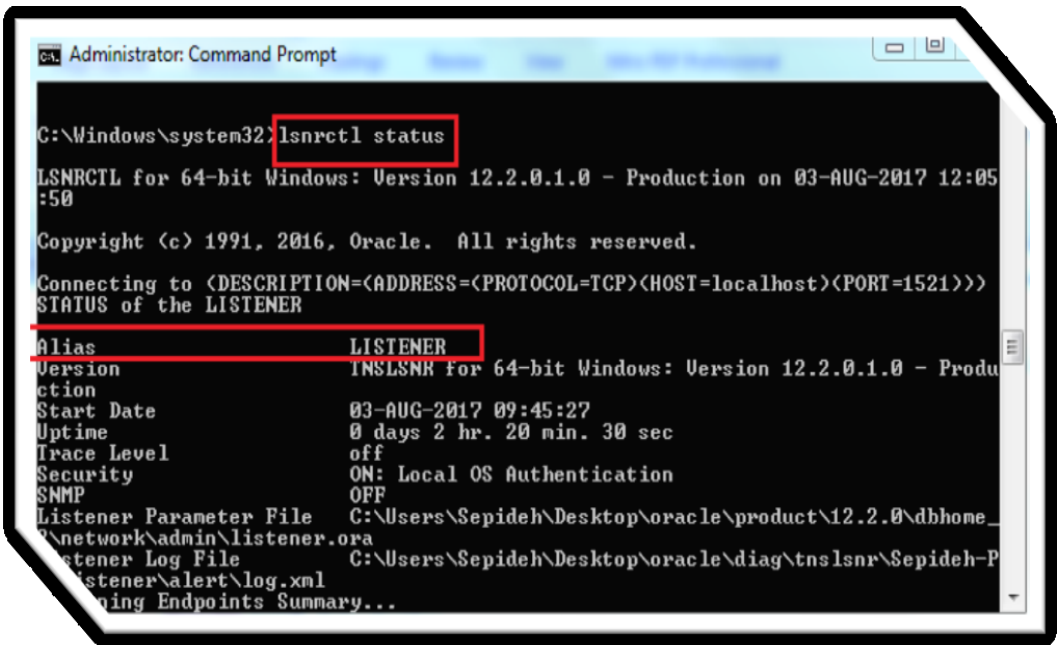
- متوقف کردن پایگاه داده: در سیستم‌های عامل ویندوزی، با استفاده از دستور زیر در خط فرمان می‌توان سرویس اوراکل را متوقف کرد:

```
net stop OracleService%ORACLE_SID%
net stop OracleServiceORCL
```

- سپس باید شنونده[۲۴] را با استفاده از دستور زیر متوقف کرد. نام شنونده را می‌توان با دستور lsnrctl status و گزینه‌ی Alias در خط فرمان به‌دست آورد (شکل ۷).

```
lsnrctl stop listener_name
lsnrctl stop LISTENER
```

[24] Listener



شکل ۷ خروجی lsnrctl status

- در ویندوز با تغییر نام %ORACLE_HOME%/bin/orauniaud12.dll.dbl به %ORACLE_HOME%/bin/orauniaud12.dll می‌توان ممیزی یکپارچه را فعال کرد.
- Listener با دستور lsnrctl start LISTENER مجددا راه‌اندازی می‌شود.
- پایگاه داده با دستور net start OracleServiceORCL مجددا راه‌اندازی می‌شود.
- حال خروجی دستور زیر بایستی TRUE باشد:

```
SELECT VALUE FROM V$OPTION WHERE PARAMETER = 'Unified Auditing'
```

- اعطای نقش AUDIT_ADMIN به کاربر SEC_ADMIN یا کاربر مورد نظر، در ادامه می‌توان از کاربر SYS که نقش AUDIT_ADMIN دارد نیز استفاده کرد.

۲. ایجاد و فعال‌سازی خط‌مشی ممیزی یکپارچه.

در اوراکل می‌توان در یک زمان چندین خط‌مشی را در پایگاه داده فعال کرد؛ ولی ایده‌آل آن است که تعداد خط‌مشی‌های فعال کم باشند. شکل کلی عبارت تعریف خط‌مشی، به گونه‌ای است که می‌توان خط‌مشی‌هایی

با تنظیمات مختلف را بر روی پایگاه داده تعریف کرد. تعریف یک خط‌مشی با گزینه‌های مرتبط به هم بهتر از تعریف چندین خط‌مشی کوچک‌تر است. بدین ترتیب امکان مدیریت بهتر خط‌مشی‌ها وجود دارد.

از دستور CREATE AUDIT POLICY برای ایجاد خط‌مشی استفاده می‌شود که شکل کلی آن به صورت زیر است:

```
CREATE AUDIT POLICY policy_name
  { {privilege_audit_clause [action_audit_clause] [role_audit_clause]}
    | { action_audit_clause [role_audit_clause] }
    | { role_audit_clause }
  }
  [WHEN audit_condition EVALUATE PER {STATEMENT|SESSION|INSTANCE}]
  [CONTAINER = {CURRENT | ALL}];
```

به‌عنوان نمونه خط‌مشی زیر عباراتی را ثبت می‌کند که مجوز CREATE ANY TABLE یا DROP ANY TABLE را نیاز دارند. همچنین خط‌مشی زیر عباراتی که با مجوزهای مستقیم نقش dba انجام می‌شوند را نیز ثبت می‌کند.

```
CREATE AUDIT POLICY table_pol PRIVILEGES CREATE ANY TABLE, DROP ANY TABLE ROLES dba;
```

۳. پس از ایجاد خط‌مشی، با استفاده از عبارت AUDIT، خط‌مشی فعال می‌شود. می‌توان خط‌مشی را بر روی یک یا چندین کاربر اعمال کرد یا می‌توان یک یا چندین کاربر را از خط‌مشی حذف کرد. همچنین می‌توان مشخص کرد، زمانی که فعالیت با موفقیت یا با خطا روبرو می‌شود، اطلاعات ثبت شوند. به‌عنوان نمونه برای اعمال خط‌مشی تعریف‌شده بر روی دو کاربر از دستور زیر استفاده می‌شود:

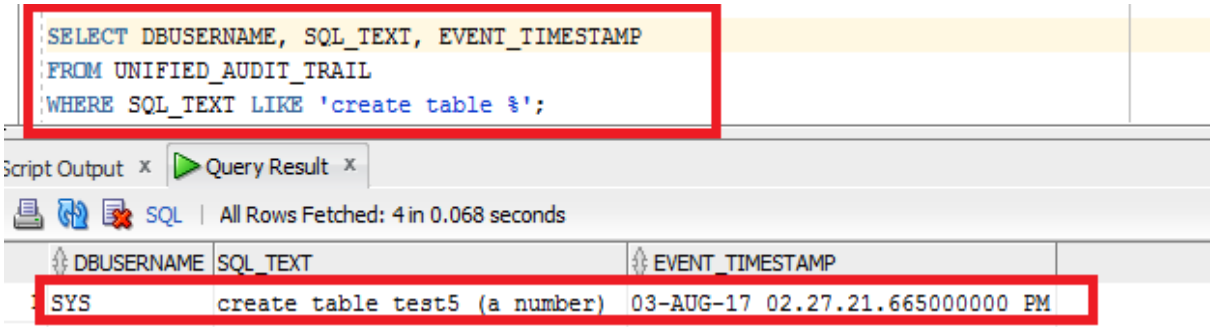
```
AUDIT POLICY table_pol BY SYS, SYSTEM;
```

۴. درصورتی‌که وضعیت متغیر audit trail، QUEUED باشد، رکوردهای ممیزی تا زمانی که صف‌های حافظه پر نشوند، در دیسک نوشته نمی‌شوند. با اجرای رویه‌ی زیر، صف‌ها در دیسک خالی‌شده و درنتیجه می‌توان رکوردها را در دید UNIFIED_AUDIT_TRAIL مشاهده کرد.

```
EXEC SYS.DBMS_AUDIT_MGMT.FLUSH_UNIFIED_AUDIT_TRAIL;
```

۵. حال با کاربر SYS به پایگاه داده متصل شده و با پرسمان CREATE TABLE جدولی ایجاد کرده و خروجی دستور زیر مشاهده می‌شود (شکل ۸):

```
SELECT DBUSERNAME, SQL_TEXT, EVENT_TIMESTAMP
FROM UNIFIED_AUDIT_TRAIL
WHERE SQL_TEXT LIKE 'create table%' ;
```



شکل ۸ خروجی دید UNIFIED_AUDIT_TRAIL

توصیفی از خط‌مشی‌های ممیزی یکپارچه را می‌توان با اجرای پرسمان بر روی دید AUDIT_UNIFIED_POLICIES به دست آورد (شکل ۹).

```
SELECT * FROM AUDIT_UNIFIED_POLICIES WHERE
POLICY_NAME='TABLE_POL';
```

```
SELECT * FROM AUDIT_UNIFIED_POLICIES WHERE POLICY_NAME='TABLE_POL';
```

Query Result | SQL | All Rows Fetched: 3 in 0.816 seconds

	POLICY_NAME	AUDIT_CONDITION	CONDITION_EVAL_OPT	AUDIT_OPTION	AUDIT_OPTION_TYPE	OBJECT_SCHEMA	OBJECT_NAME	OBJECT_TYPE	COMMON	INHERITED
1	TABLE_POL	NONE	NONE	DROP ANY TABLE	SYSTEM PRIVILEGE	NONE	NONE	NONE	NO	NO
2	TABLE_POL	NONE	NONE	CREATE ANY TABLE	SYSTEM PRIVILEGE	NONE	NONE	NONE	NO	NO
3	TABLE_POL	NONE	NONE	DBA	ROLE PRIVILEGE	NONE	NONE	NONE	NO	NO

شکل ۹ خروجی دید AUDIT_UNIFIED_POLICIES

توضیحی در مورد خط‌مشی‌های ممیزی یکپارچه‌ی فعال را می‌توان با اجرای پرسمان بر روی دید AUDIT_UNIFIED_ENABLED_POLICIES به دست آورد (شکل ۱۰).

```
SELECT * FROM audit_unified_enabled_policies;
```

Query Result | SQL | All Rows Fetched: 9 in 0.008 seconds

	USER_NAME	POLICY_NAME	ENABLED_OPT	ENABLED_OPTION	ENTITY_NAME	ENTITY_TYPE	SUCCESS	FAILURE
1	SYS	TABLE_POL	BY	BY USER	SYS	USER	YES	YES
2	SYSTEM	TABLE_POL	BY	BY USER	SYSTEM	USER	YES	YES
3	SYS	TABLE_POL3	BY	BY USER	SYS	USER	YES	YES
4	SYSTEM	TABLE_POL3	BY	BY USER	SYSTEM	USER	YES	YES
5	ALL USERS	ORA_SECURECONFIG	BY	BY USER	ALL USERS	USER	YES	YES
6	ALL USERS	ORA_LOGON_FAILURES	BY	BY USER	ALL USERS	USER	NO	YES
7	ALL USERS	TABLE_POL3	BY	BY USER	ALL USERS	USER	YES	YES
8	ALL USERS	TABLE_POL2	BY	BY USER	ALL USERS	USER	YES	YES
9	ALL USERS	TABLE_POL4	BY	BY USER	ALL USERS	USER	YES	YES

شکل ۱۰ خروجی دید AUDIT_UNIFIED_ENABLED_POLICIES

۲-۱-۳ ممیزی ریزدانه

ممیزی ریزدانه[۲۵] این امکان را فراهم می‌کند که خط‌مشی‌های ممیزی در سطح ریز و دقیق تعریف شوند. در این نوع ممیزی، می‌توان خط‌مشی‌هایی با شرایط مشخص تعریف کرد که در صورت تحقق آن شرایط، ممیزی اجرا شود.

با استفاده از ممیزی ریزدانه همچون ممیزی از تعداد دفعات دسترسی به داده، می‌توان تنظیمات سفارشی‌شده برای ممیزی ایجاد کرد. با ممیزی ریزدانه می‌توان بر دسترسی به داده بر اساس محتوا نظارت داشت. نمونه‌هایی از کاربردهای ممیزی ریزدانه عبارت‌اند از [۷]:

- دسترسی به جدول از ساعت نه صبح تا شش بعدازظهر یا در روزهای شنبه و یکشنبه
- استفاده از آدرس IP خارج از شبکه سازمان
- انتخاب یا به‌روزرسانی ستونی از جدول
- به‌روزرسانی مقدار ستونی از جدول

خط‌مشی‌های ممیزی که توسط خط‌مشی‌های ممیزی یکپارچه تعریف می‌شوند، اکثر عملیاتی که توسط خط‌مشی‌های ممیزی ریزدانه اجرا می‌شوند به‌جز موارد زیر را شامل می‌شوند:

- **ممیزی ستون‌های مشخص**: ممیزی از ستون‌های مشخصی که داده‌های حساس را شامل می‌شوند همچون ستون‌های حقوق.
- **استفاده از رسیدگی‌کننده[۲۶] رویداد**: به‌عنوان‌مثال می‌توان تابعی نوشت که هنگام تغییر ستون مشخص در نیمه‌شب، یک نامه الکترونیکی برای مدیر امنیتی ارسال کند.

رکوردهای ممیزی ریزدانه در شِمای AUDSYS و در فضای جدولSYSAUX ذخیره می‌شوند. این فضای جدول می‌تواند یک فضای جدول رمز شده باشد. برای یافتن رکوردهایی که توسط خط‌مشی‌های ممیزی تولیدشده‌اند می‌توان پرسمان‌هایی بر روی دید UNIFIED_AUDIT_TRAIL اجرا کرد. نقش‌های درگیر در ممیزی ریزدانه همچون ممیزی یکپارچه AUDIT_ADMIN و AUDIT_VIEWER هستند [۷].

[25] Fine-grained auditing

[26] Handler

۱-۲-۱-۳ مثالی از ممیزی ریزدانه

بسته‌ی[۲۷] PL/SQL به نام DBMS_FGA خط‌مشی‌های ممیزی ریزدانه را مدیریت می‌کند [۷].

در ادامه تمامی گام‌های طی شده به‌منظور آزمون ممیزی ریزدانه بیان می‌شوند:

- ابتدا باید با کاربر SYS و مجوزهای SYSDBA وارد پایگاه داده شد و کاربری جدیدی را با دستور زیر ایجاد کرد:

```
ALTER SESSION SET "_ORACLE_SCRIPT"=true;
CREATE USER test IDENTIFIED BY flintstone;
```

- سپس باید مجوزهای زیر را به کاربر جدید اعطا کرد:

```
GRANT create session TO test;
GRANT create table TO test;
GRANT create view TO test;
GRANT create any trigger TO test;
GRANT create any procedure TO test;
GRANT create sequence TO test;
GRANT create synonym TO test;
```

- در ادامه با کاربر جدید وارد پایگاه داده شده و جدولی مطابق با دستور زیر ساخته می‌شود:

```
CREATE SCHEMA AUTHORIZATION test
 CREATE TABLE people
 ( id NUMBER(10) NOT NULL,
salary NUMBER(10) NOT NULL,
name NUMBER(50) NOT NULL,
CONSTRAINT people_pk PRIMARY KEY (id));
```

- کاربر SYS می‌تواند با دستور زیر خط‌مشی ممیزی ریزدانه را ایجاد کند:

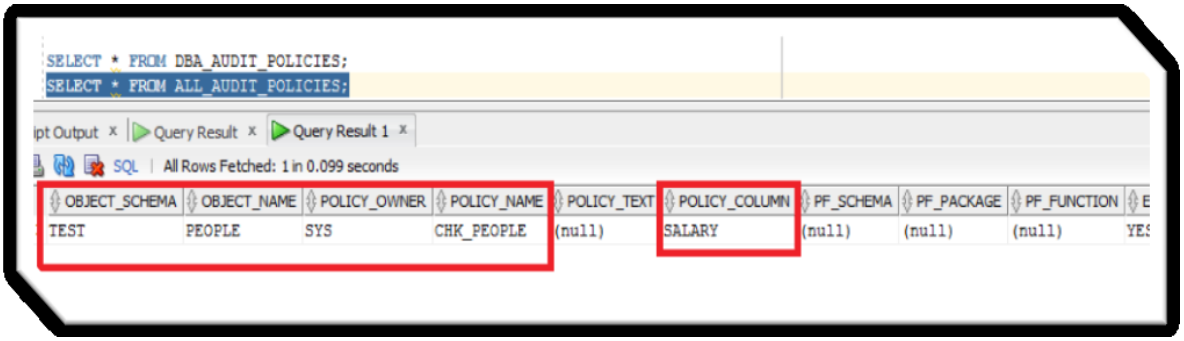
[27] Package

```
BEGIN
 DBMS_FGA.ADD_POLICY(
  object_schema      => 'test',
   object_name       => 'people',
   policy_name       => 'chk_people',
   audit_column      => 'SALARY',
   enable            =>  TRUE,
   statement_types   => 'INSERT, UPDATE, SELECT, DELETE') ;
END;
```

به صورت پیش‌فرض، خط‌مشی ممیزی ریزدانه با مجوزهای مالک خط‌مشی اجرا می‌شود. حداکثر تعداد خط‌مشی‌های ممیزی ریزدانه که می‌توان بر روی یک جدول یا دید تعریف کرد، ۲۵۶ عدد است. امکان تعریف خط‌مشی بر روی جداول و دیدهایی که در شِمای SYS هستند، وجود ندارد؛ به همین دلیل است که در گام‌های پیش، ابتدا کاربری ایجاد و جدولی توسط آن کاربر و در شِمای خارج از SYS تعریف شده است.

درصورتی‌که کاربر بخواهد خط‌مشی را تغییر دهد، امکان به‌روزرسانی آن وجود ندار؛ بنابراین ابتدا باید آن را حذف و مجددا ایجاد کند.

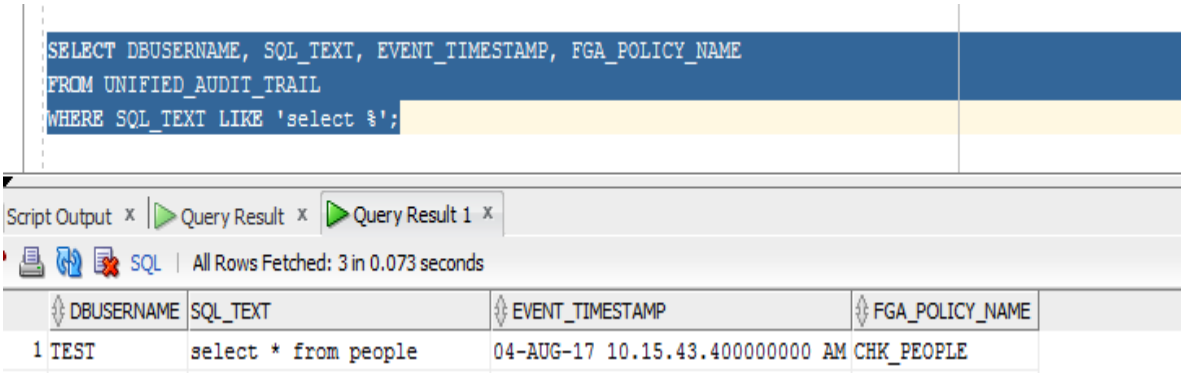
به‌منظور اطلاع از خط‌مشی‌های ممیزی ریزدانه می‌توان پرسمانی بر روی دیدهای ALL_AUDIT_POLICIES و DBA_AUDIT_POLICIES اجرا کرد (شکل ۱۱).



شکل ۱۱ خروجی دید ALL_AUDIT_POLICIES

- با توجه به آن‌که در عبارت تعریف خط‌مشی، audit_column ستون salary وجود دارد، درصورتی‌که عبارت درج، به‌روزرسانی، انتخاب و حذف (با توجه به گزینه statement_types) بر روی این ستون اجرا شود، رکورد ممیزی ثبت خواهد شد. با کاربر جدید، دستور SELECT * FROM people; وارد می‌شود.
- با کاربر sys، دستور زیر برای دیدن رکوردهای ممیزی از دید UNIFIED_AUDIT_TRAIL وارد می‌شود (شکل ۱۲):

```
SELECT DBUSERNAME, SQL_TEXT, EVENT_TIMESTAMP, FGA_POLICY_NAME
FROM UNIFIED_AUDIT_TRAIL
WHERE SQL_TEXT LIKE 'select%' ;
```



شکل ۱۲ خروجی دید UNIFIED_AUDIT_TRAIL

دید UNIFIED_AUDIT_TRAIL دارای ستونی با نام FGA_POLICY_NAME است که می‌توان از آن برای فیلتر کردن رکوردهایی که توسط خط‌مشی ممیزی ریزدانه مشخصی ایجاد شده‌اند، استفاده کرد (شکل ۱۳).

```
SELECT DBUSERNAME, SQL_TEXT, EVENT_TIMESTAMP, FGA_POLICY_NAME
FROM UNIFIED_AUDIT_TRAIL
WHERE FGA_POLICY_NAME = 'CHK_PEOPLE';
```

Script Output | Query Result | Query Result 1

SQL | All Rows Fetched: 3 in 0.055 seconds

	DBUSERNAME	SQL_TEXT	EVENT_TIMESTAMP	FGA_POLICY_NAME
1	TEST	select * from people	04-AUG-17 10.15.43.400000000 AM	CHK_PEOPLE
2	TEST	select * from people	04-AUG-17 10.18.41.121000000 AM	CHK_PEOPLE
3	TEST	select salary from people	04-AUG-17 10.21.01.767000000 AM	CHK_PEOPLE

شکل ۱۳ خروجی دید UNIFIED_AUDIT_TRAIL

توجه به این نکته ضروری است که درصورتی‌که عبارتی مشابه عبارت SELECT name FROM people; اجرا شود، سطر ممیزی اضافه نخواهد شد؛ به دلیل اینکه در این عبارت ستون salary استفاده نشده است.

٤ ابزارهای تحلیل‌کننده‌ی ممیزی

در این بخش سعی بر آن است که ابزارهای تحلیل‌کننده‌ی فایل‌های رویدادنگاری و رکوردهای ممیزی در اوراکل موردبررسی قرار گیرند. در این راستا دو ابزار LogMiner و Oracle EventLog Analyzer بررسی‌شده‌اند.

٤-١ logMiner

Oracle LogMiner که بخشی از پایگاه داده‌ی اوراکل است، امکان استخراج اطلاعات از درون فایل‌های redo log آنلاین و آرشیوشده را از طریق دستورات SQL فراهم می‌کند. به LogMiner از طریق خط فرمان[۲۸] یا واسط گرافیکی می‌توان دسترسی داشت. واسط گرافیکی Oracle LogMiner بخشی از Oracle Enterprise Manager است. تمامی تغییراتی که بر روی داده‌ها یا دیکشنری پایگاه داده رخ می‌دهد در فایل‌های redo log ثبت می‌شوند. LogMiner رابطی برای فایل‌های redo log است که از آن می‌توان

[28] Command line

به‌عنوان ابزار ممیزی و ابزار تحلیل داده استفاده کرد. در ادامه برخی از قابلیت‌های این ابزار معرفی می‌شوند [۳]:

- درصورتی‌که خطا و اشتباهی منطقی در پایگاه داده رخ دهد، می‌توان با استفاده از LogMiner پایگاه داده را به وضعیت پیش از خطا بازگرداند. به‌عنوان نمونه، درصورتی‌که مدیر پایگاه داده به اشتباه جدولی را پاک کند، می‌توان وضعیت پایگاه داده را به پیش از پاک شدن جدول برگرداند.
- LogMiner مشخص می‌کند برای بازیابی اطلاعات در سطح تراکنش چه اقداماتی بایستی صورت بگیرد. مثلاً می‌توان عملیاتی برای بازگرداندن یک جدول به وضعیت اصلی آن انجام داد. بدین منظور می‌توان از عبارات SQL مربوط به جدول مشخص که توسط LogMiner فراهم می‌شوند، استفاده کرد.
- با استفاده از LogMiner می‌توان متوجه شد در کدام‌یک از جداول میزان درج و به‌روزرسانی بیشتر است و پس‌ازآن با توجه به نتایج حاصل از تحلیل‌های انجام‌شده به تنظیم پایگاه داده برای بهبود عملکرد و کارایی پرداخت.
- با استفاده از این ابزار می‌توان عبارات DML و DDL اجراشده بر روی پایگاه داده، ترتیب اجرای آن‌ها و شخص اجراکننده‌ی آن‌ها را ردیابی کرد.

۱-۱-٤ تنظیمات LogMiner

در تنظیمات LogMiner، چهار مفهوم زیر مهم هستند [۳]:

- پایگاه داده مبدأ: پایگاه داده تولیدکننده‌ی فایل‌های redo log که logMiner قصد دارد آن‌ها را تحلیل کند.
- پایگاه داده‌ای که LogMiner هنگام تحلیل از آن استفاده می‌کند که به آن mining database می‌گویند. در حقیقت ممکن است فایل‌های redo log در پایگاه داده مبدأ آرشیو شوند و به پایگاه داده Mining منتقل شوند (شکل ۱۴).
- دیکشنری LogMiner: دیکشنری به LogMiner اجازه می‌دهد به‌جای شماره‌ی یکتای هر شیئ، نام آن را جایگزین کند. بدون دیکشنری، خروجی LogMiner شامل شماره‌های یکتای اشیا و داده‌های باینری خواهد بود.

به‌عنوان نمونه دستور SQL زیر را در نظر بگیرید:

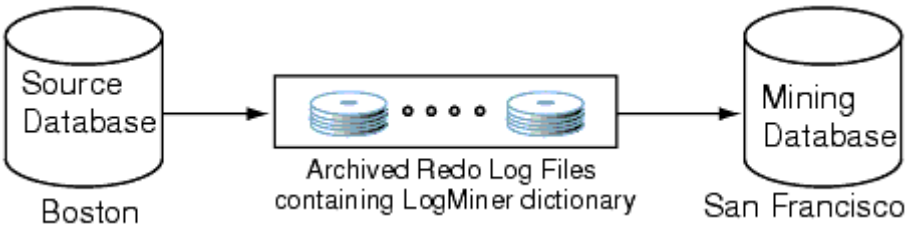
```
INSERT INTO HR.JOBS(JOB_ID, JOB_TITLE, MIN_SALARY, MAX_SALARY)
VALUES('IT_WT','Technical Writer', 4000, 11000);
```

بدون دیکشنری، LogMiner دستور بالا را به صورت زیر نمایش می‌دهد:

```
insert into "UNKNOWN"."OBJ# 45522"("COL 1","COL 2","COL 3","COL 4") values
(HEXTORAW('45465f4748'),HEXTORAW('546563686e6963616c20577269746572'),
HEXTORAW('c229'),HEXTORAW('c3020b'));
```

دیکشنری بایستی توسط پایگاه داده مبدایی که فایل‌های redo log را ساخته است، ایجاد شود.

- فایل‌های redo log که شامل تغییرات اعمال‌شده بر روی پایگاه داده یا دیکشنری پایگاه داده هستند.



شکل ۱۴ اجزای LogMiner

۱-۱-۱-٤ گزینه‌های دیکشنری LogMiner

زمانی که LogMiner داده‌های بازیابی شده را بازمی‌گرداند، شماره‌های یکتای اشیا را تبدیل به نام اشیا می‌کند. LogMiner این تبدیل را با استفاده از دیکشنری انجام می‌دهد. در ادامه دو گزینه برای تهیه دیکشنری به صورت خلاصه توضیح داده‌شده است [۳]:

- **استفاده از کاتالوگ آنلاین:** اوراکل پیشنهاد می‌دهد از این گزینه زمانی استفاده شود که به پایگاه داده مبدأ دسترسی وجود داشته باشد و هیچ تغییری در تعریف ستون جداول ایجاد نشده باشد. این گزینه کاراترین و آسان‌ترین گزینه است. این گزینه نیازمند آن است که پایگاه داده باز باشد.

بدین منظور هنگام راه‌اندازی LogMiner از گزینه‌ی مشخص‌شده در دستور زیر استفاده می‌شود:

```
EXECUTE DBMS_LOGMNR.START_LOGMNR(-
OPTIONS => DBMS_LOGMNR.DICT_FROM_ONLINE_CATALOG);
```

- **استخراج دیکشنری در فایل‌های redo log:** زمانی که به پایگاه داده‌ی مبدأ دسترسی وجود نداشته باشد یا در ستون جداول تغییر ایجادشده است، پیشنهاد می‌شود از این گزینه برای تهیه‌ی دیکشنری

استفاده شود. در این حالت باید پایگاه داده باز و در حالت ARCHIVELOG باشد. زمانی که دیکشنری در redo log استخراج می‌شود، هیچ دستوری از DDL نمی‌تواند اجرا شود. برای استخراج دیکشنری در فایل‌های redo log از دستور زیر استفاده می‌شود:

```
EXECUTE DBMS_LOGMNR.BUILD ( -
OPTIONS=> DBMS_LOGMNR_D.STORE_IN_REDO_LOGS);
```

۲-۱-۱-٤ گزینه‌های فایل redo log

به‌منظور استخراج داده‌ها از فایل‌های redo log، بایستی اطلاعاتی در مورد فایل‌های redo log به LogMiner داده شود. LogMiner می‌تواند به صورت خودکار و پویا یک فهرست از فایل‌های redo log برای تجزیه‌وتحلیل تهیه کند، همچنین امکان تهیه فهرستی از فایل‌های redo log توسط کاربر برای LogMiner وجود دارد؛ بنابراین دو روش زیر برای تعیین فایل‌های redo log برای logMiner وجود دارد:

- **خودکار**: درصورتی‌که LogMiner بر روی پایگاه داده مبدأ استفاده شود، می‌توان آن را برای یافتن و ایجاد خودکار فهرستی از فایل‌های redo log هدایت کرد.
- **غیرخودکار**: پیش از راه‌اندازی LogMiner می‌توان فهرستی از فایل‌های redo log را ایجاد کرد. در صورت استفاده از این روش نیازی نیست LogMiner به پایگاه داده مبدأ متصل باشد. برای مشخص شدن شروع فهرستی از فایل‌های redo log از گزینه New در رویه‌ی DBMS_LOGMNR.ADD_LOGFILE مطابق دستور زیر استفاده می‌شود:

```
EXECUTE DBMS_LOGMNR.ADD_LOGFILE( -
LOGFILENAME => '/oracle/logs/redo01.log', -
OPTIONS => DBMS_LOGMNR.NEW);
```

به‌منظور افزودن فایل‌های بیشتر به فهرست، از گزینه‌ی ADDFILE در رویه‌ی مشخص‌شده در دستور زیر استفاده می‌شود.

```
EXECUTE DBMS_LOGMNR.ADD_LOGFILE( -
   LOGFILENAME => '/oracle/logs/redo02.log ', -
    OPTIONS => DBMS_LOGMNR.ADDFILE);
```

با استفاده از دستور زیر می‌توان فهرست فایل‌های redo log که در نشست فعلی توسط LogMiner موردبررسی قرار می‌گیرند را مشاهده کرد.

```
SELECT * FROM V$LOGMNR_LOGS;
```

۲-۱-٤ چگونگی کار با LogMiner

در این بخش نحوه‌ی استفاده از LogMiner به‌منظور تجزیه‌وتحلیل فایل‌های redo log بیان می‌شود [۳].

۱. فعال‌سازی رویدادنگاری تکمیلی[۲۹]

۲. استخراج دیکشنری LogMiner یا استفاده از کاتالوگ آنلاین: برای استفاده از کاتالوگ آنلاین از گزینه DICT_FROM_ONLINE_CATALOG هنگام راه‌اندازی LogMiner استفاده می‌شود.

۳. تعیین فایل‌های redo log برای تحلیل: در بخش ۲-۱-۱-۴ (گزینه‌های فایل redo log) توضیح داده‌شده است.

۴. راه‌اندازی و شروع به کار LogMiner

۵. استفاده از دستور V$LOGMNR_CONTENTS به‌منظور استخراج اطلاعات

۶. بستن نشست LogMiner

در ادامه برخی از گام‌ها به صورت خلاصه توضیح داده می‌شوند.

۲-۱-۱-٤ فعال‌سازی رویدادنگاری تکمیلی

به‌منظور فعال‌سازی حداقل رویدادنگاری تکمیلی در سطح پایگاه داده از دستور زیر استفاده شده است:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

[29] Supplemental logging

۴-۱-۲-۲ راه‌اندازی و شروع به کار LogMiner

برای راه‌اندازی LogMiner بایستی رویه‌ی DBMS_LOGMNR.START_LOGMNR اجرا شود.

```
EXECUTE DBMS_LOGMNR.START_LOGMNR(-
OPTIONS => DBMS_LOGMNR.DICT_FROM_ONLINE_CATALOG;
```

می‌توان با استفاده از پارامتر OPTIONS ویژگی‌های دلخواهی به نشست LogMiner اضافه کرد. به‌عنوان نمونه می‌توان مشخص کرد که تنها تراکنش‌های تثبیت شده توسط دید[۳۰] V$LOGMNR_CONTENTS نشان داده شوند.

```
EXECUTE DBMS_LOGMNR.START_LOGMNR(OPTIONS => -
 DBMS_LOGMNR.DICT_FROM_ONLINE_CATALOG + -
 DBMS_LOGMNR.COMMITTED_DATA_ONLY);
```

۴-۱-۲-۳ استخراج اطلاعات با استفاده از دید V$LOGMNR_CONTENTS

به‌منظور دسترسی به اطلاعات موردنظر، می‌توان پرسمان‌هایی را بر روی دید V$LOGMNR_CONTENTS اجرا کرد. برای اجرای پرسمان بر روی این دید، نیاز به مجوزهای LOGMINING یا SYSDBA است. این دید شامل اطلاعات مربوط به تغییرات اعمال‌شده بر روی پایگاه داده ازجمله موارد زیر است:

- در ستون OPERATION نوع تغییر اعمال‌شده بر روی پایگاه داده، ازجمله INSERT، UPDATE، DELETE یا DDL مشخص می‌شود.
- نمایش SCN در ستون SCN
- ستون USERNAME شامل نام کاربری است که عبارت DDL یا DML را اجرا کرده است.

[30] View

- درصورتی‌که تغییر اعمال‌شده با عبارت DML انجام شده باشد، در ستون SQL_REDO عبارت SQL معادل عبارتی که رکوردهای redo را تولید کرده است، نشان داده می‌شود.
- درصورتی‌که تغییر اعمال‌شده با عبارت DML انجام گرفته شده باشد، در ستون SQL_UNDO عبارت SQL برای بازگرداندن[۳۱] تغییر اعمال‌شده، نشان داده می‌شود.

به‌عنوان نمونه به‌منظور یافتن عملیات delete اجراشده توسط کاربر Ron بر روی جدول orders، دستور زیر وارد می‌شود:

```
SELECT OPERATION, SQL_REDO, SQL_UNDO
 FROM V$LOGMNR_CONTENTS
 WHERE SEG_OWNER = 'OE' AND SEG_NAME = 'ORDERS' AND OPERATION = 'DELETE' AND USERNAME = 'RON';
```

٤-١-٢-٤ بستن نشست LogMiner

به‌منظور خاتمه‌ی نشست LogMiner، باید دستور زیر اجرا شود:

```
EXECUTE DBMS_LOGMNR.END_LOGMNR;
```

این رویه تمامی فایل‌های redo log را بسته و تمامی منابع سیستم و پایگاه داده که به LogMiner تخصیص داده شده‌اند را آزاد می‌کند. درصورتی‌که این دستور اجرا نشود، تمامی منابع تخصیص داده شده تا انتهای نشست اوراکل آزاد نخواهند شد.

٤-١-٣ آیا LogMiner ابزار مناسبی برای جرم‌شناسی است؟

در یکی از مطالعاتی که در سال ۲۰۰۵ بر روی این ابزار انجام شد [۲]، بررسی شدکه آیا logMiner می‌تواند به‌عنوان ابزاری برای جرم‌شناسی استفاده شود یا خیر. در این راستا توانایی این ابزار برای تولید نتایج تکرارپذیر، قابل‌اثبات[۳۲]، دقیق[۳۳] و صحیحی[۳٤] که در دادگاه قانونی قابل‌ارائه هستند، مورد ارزیابی قرارگرفت.

---

[31] Undo

[32] Verifiable

[33] Precise

مشکل اولی که در این مطالعه به آن اشاره‌شده است این است که ستون TIMESTAMP در خروجی دید V$LOGMNR_CONENTS برخلاف نامش نوع داده‌ی TIMESTAMP ندارد، بلکه نوع داده‌ی آن DATE است. دقت نوع داده‌ی DATE برخلاف TIMESTAMP در حد ثانیه است و نه کسری از ثانیه؛ بنابراین زمان‌ها با دقتی که در فایل‌های redo log ثبت‌شده‌اند، در خروجی LogMiner نمایش داده نمی‌شوند.

مشکل دوم LogMiner در زمان این مطالعه این بود که داده‌های ستون‌هایی از جداول که دارای نوع داده TIMESTAMP بودند را با دقت ثانیه نمایش می‌داد و درواقع در نمایش عبارت DML در ستون SQL_REDO، داده‌ها را تغییر می‌داد. توجه به این نکته ضروری است که در آزمون مشابهی که در سال ۲۰۱۷ توسط نویسنده این گزارش صورت گرفته است، این مشکل به‌طورکلی مرتفع شده است ولی مشکل اول کماکان باقی مانده است.

۲-٤ ابزار EventLog Analyzer

ابزار Oracle Eventlog analyzer امکان مشاهده‌ی آسان رکوردهای ممیزی را فراهم می‌کند. این ابزار امنیتی پایگاه داده، اطلاعاتی را در مورد گزارش‌های امنیتی از کارگزارهای پایگاه داده‌ی اوراکل استخراج کرده, آن‌ها را در محیط گرافیکی به کاربر نمایش می‌دهد و این امکان را فراهم می‌کند که کنترل‌های دسترسی و امنیتی مؤثری پیاده‌سازی شوند [۸].

این ابزار اطلاعات مرتبط را جمع‌آوری و تحلیل می‌کند و به کاربر در قالبی قابل‌فهم نمایش می‌دهد. بدین منظور ابزار Eventlog Analyzer از گزارش‌های پایگاه داده‌ی اوراکل، همچون تلاش‌های ناموفق برای ورود به سیستم، تلاش برای دسترسی با حساب‌های کاربری که وجود ندارند و دسترسی‌های غیرمجاز استفاده می‌کند. درصورتی‌که ممیزی ریزدانه بر روی پایگاه داده‌ی اوراکل فعال باشد، ابزار Eventlog Analyzer اطلاعات باارزشی در مورد اینکه چه کسی، چه‌کاری را بر روی کدام داده و در چه زمانی انجام داده است، فراهم می‌کند [۸].

[34] Accurate

ابزار Eventlog Analyzer ممیزی و نظارت بر پایگاه داده‌ی اوراکل و تهیه گزارش و ایجاد هشدار را تسهیل می‌بخشد. این ابزار موارد زیر را گزارش می‌دهد [۹]:

- فعالیت‌های کاربر در پایگاه داده
- تغییرات اعمال‌شده بر روی حساب کاربری
- دسترسی‌های کاربر به کارگزارهای پایگاه داده‌ی اوراکل
- تلاش برای نقض امنیت

با این ابزار می‌توان با پیام کوتاه و نامه‌ی الکترونیکی تغییراتی که در کارگزار پایگاه داده رخ داده است، رفتارهای غیرمعمول کاربران، دسترسی‌های غیرمجاز، تلاش برای نقض امنیت و دسترسی به داده‌های حساس را اطلاع‌رسانی کرد [۹].

۱-۲-٤ ممیزی با ابزار Eventlog Analyzer

ابزار Eventlog Analyzer برای ممیزی پایگاه داده در موارد زیر کاربرد دارد [۹]:

- **نظارت بر پایگاه داده‌ی اوراکل**: نظارت و ردگیری عملیاتی که در پایگاه داده رخ می‌دهد و تهیه گزارش و اطلاعات با جزئیات از موارد زیر:
  - تغییر در ساختار پایگاه داده مانند ایجاد، حذف یا تغییر در جداول
  - پرسمان‌هایی که بر روی پایگاه داده اجرا می‌شوند همچون انتخاب، درج در جدول

گزارش‌های تهیه‌شده اطلاعات دقیقی در مورد اینکه چه کسی، از کجا و چه زمانی تغییرات را اعمال و یا پرسمان را اجرا کرده است، فراهم می‌کنند.

- **ممیزی تغییرات در حساب‌های کاربری**: نظارت بر تغییرات در حساب‌های کاربری، ایجاد حساب‌های کاربری مجوزدار، تغییر آن‌ها، تغییر در نقش‌ها، کمک می‌کند که تلاش برای حملات امنیتی شناسایی شوند. در این مورد، گزارش‌های زیر تهیه می‌شوند:
  - ایجاد، تغییر و حذف حساب کاربری
  - ایجاد، تغییر و حذف نقش
  - اعطای نقش یا سلب نقش از حساب‌های کاربری

- چه کسی، از کجا و چه زمانی، نقش یا حساب کاربری را ایجاد کرده است، تغییر داده است یا حذف کرده است.

- **نظارت بر دسترسی به کارگزار اوراکل**: به دلیل آنکه کارگزارهای پایگاه داده، اطلاعات محرمانه را در خود ذخیره می‌کنند، ردگیری دسترسی کاربر و فعالیت‌هایی که در کارگزار انجام می‌دهد، کمک می‌کند که تلاش برای دسترسی غیرمجاز متوقف شود. در این مورد گزارش‌های زیر تهیه می‌شوند:
  - ورود به پایگاه داده، خروج از آن و ورود از راه دور
  - تلاش‌های ناموفق برای ورود به سیستم و حساب‌های کاربری با حداکثر تلاش ناموفق برای ورود
  - گزارش‌هایی در مورد ورود به سیستم، ورودهای ناموفق و خطا در ورود
  - جزئیاتی در مورد راه‌اندازی و خاموش کردن کارگزار

- **گزارش‌های امنیتی پایگاه داده‌ی اوراکل**: حملات امنیتی به طور معمول هدف خود را داده‌های مشتریان یا اطلاعات محرمانه‌ای که در پایگاه داده ذخیره شده‌اند، قرار می‌دهند. مدیران امنیتی وظیفه دارند به تلاش‌هایی که برای حمله انجام می‌شوند، عکس‌العمل نشان دهند و حملات را متوقف کنند یا امکان خرابی داده‌ها را کاهش دهند. بدین منظور باید بتوانند ریشه حمله را تحلیل و الگوی حمله را مجددا ایجاد کنند. ابزار Eventlog Analyzer گزارش‌های زیر را در این زمینه تهیه می‌کند:
  - حملات تزریق SQL و منع از سرویس یا تلاش برای حمله، که با تحلیل آن‌ها می‌توان امنیت پایگاه داده را تقویت کرد.
  - زمانی که حمله انجام می‌شود، فراهم کردن گزارشی در رابطه با منبع حمله، هدف حمله و سایر اطلاعاتی که به فهم الگوی حمله کمک می‌کنند.
  - حساب‌های کاربری که ممکن است توسط مهاجم مورد سوءاستفاده قرارگرفته باشند همچون حساب‌های کاربری که به دلیل تعداد دفعات تلاش ناموفق برای ورود، قفل‌شده‌اند[۳۵].

[35] Lock

۵ جمع‌بندی

ازآنجایی‌که نظارت بر فعالیت‌های درون پایگاه‌های داده از اهمیت بالایی برخوردار است، در پایگاه‌های داده مختلف روش‌های متفاوتی برای نظارت، ردگیری و ممیزی وجود دارد. با استفاده از اطلاعات ثبت‌شده‌ی حاصل از ممیزی می‌توان اتفاقات و دسترسی‌های غیرمجاز و حملات و تلاش برای حملات مختلف را ردگیری کرد و عکس‌العمل مناسب ازجمله بازگرداندن پایگاه داده به وضعیت پیش از اتفاقات نشان داد. در این گزارش روش‌های گزارش‌گیری، ممیزی و جرم‌شناسی در پایگاه‌های داده اوراکل موردبررسی قرارگرفت.

٦ منابع

[1]. Randy Urbano. 2017. Oracle Database Administrator’s Guide, 12*c* Release 2 (12.2)

[2]. Darl Kuhn. 2013. *Pro Oracle Database 12c Administration* (2nd ed.). Apress, Berkely, CA, USA.

[3]. Rhonda Day. 2017. *Oracle Database Utilities, 12c Release 2 (12.2)*

[4]. Wright-GSEC, P.M. and GCFW, G., 2005. Oracle Database Forensics using LogMiner.

[5]. Mary Beth Roeser. 2017. *Oracle Database SQL Language Reference, 12c Release 2 (12.2)*

[6]. Patricia Huey. 2017. *Oracle Database 2 Day + Security Guide, 12c Release 2 (12.2)*

[7]. Patricia Huey. 2017. *Oracle Database Security Guide, 12c Release 2 (12.2)*

[8]. http://www.secure-bytes.com/oracle-event-log-analyzer.php

[9]. https://www.manageengine.com/products/eventlog/oracle-database-auditing-tool.html

[10]. https://www.melissadata.com/enews/articles/0108/2.htm