DB2 for LUW V9.5
セキュリティー・デザイン・ガイド
監査機能編

＜第1.0版　2008年6月＞

お断り：当資料は、DB2 for Linux, UNIX and Windows V9.5 をベースに作成されています。

本書に含まれている情報は、正式なIBMのテストを受けておらず、全体または一部においていかなる保証責任を負わないものとし、保証対象とは見なされません。この情報の使用またはこれらの技術の実施は、いずれも、使用先の責任において行われるべきものであり、それらを評価し、実際に使用する環境に統合する使用先の判断に依存しています。それぞれの項目は、ある特定の状態において正確であることがIBMによって調べられていますが、他のところで同じまたは同様の結果が得られる保証はありません。これらの技術を自身の環境に適用することを試みる使用先は、自己の責任において行う必要があります。

監査機能編　目次

- データベースにおける監査ログ
- DB2 LUW監査機能
- DB2 LUW監査運用
- DB2 LUW監査レポート例
- 別紙
  - ClientInformationAPIと監査機能の組み合わせ
  - AUDITパフォーマンス検証結果
  - 監査レコードのフォーマット

データベースにおける監査ログ

＜第1.0版　2008年6月＞

お断り：当資料は、DB2 for Linux, UNIX and Windows V9.5 をベースに作成されています。

データベースにおける監査ログ取得の重要性

- 監査証跡としてログを取得・保管することはセキュリティー一般として重要
  - ITシステム上での行動をロギング、トレースすることで改ざん等の不正アクセスや、否認に対する抑止力となる
  - ログを定期的に、あるいは、リアルタイムでチェックすることで問題の早期発見に繋がる
  - 問題が発生、発覚した時の証拠として重要
    - 不正アクセス禁止法
      - 監査ログは都道府県公安委員会の援助を得るための資料となる
    - J-SOX法
      - 財務関係システムにおいて監査強化が必要

    ⇒ どんなログをどのように取得し、どこに保管するかが大切
    DBにおいても、ログの取得、保管が求められている

解説：DBにおける監査ログ取得の重要性

- セキュリティー意識の高まりとともに、DBにおける監査ログ取得の要求も高まっています。
- 監査証跡としてログを取得・保管することはセキュリティーー一般として重要です。
  - 一般に監査ログをとることによって、以下の効果があると言われています。
    - ITシステム上での行動をロギング、トレースすることで改ざん等の不正アクセスや、様々な行動の否認(*1)に対する抑止力となる。
    - ログを定期的に、あるいは、リアルタイムでチェックすることで問題の早期発見に繋がる。
    - 問題が発生、発覚した時の証拠として重要
  - ログの取得自体は、権限管理や暗号化などのように、実際に不正なアクセスを制限したり、情報を隠蔽する効果はありません。つまり、ログをとっているということだけで、不正なアクセスを防止することはできません。
    しかし、ログをとっていることで、問題が起こったときの証拠となり得ますし、ログをとっているということを知らしめることで、不正なアクセスをしても検知されると思わせ、不正なアクセスを抑止することが期待されます。
    また、不正アクセス禁止法では、不正アクセスがあった場合、「当該特定電子計算機の作動状況及び管理状況その他の参考となるべき事項に関する書類その他の物件」を添えて都道府県公安委員会の援助を申請することができると定めており、監査ログはそのための資料となり得ます。
    また、いわゆるJ-SOX法では財務関係の情報を扱うシステムでの監査強化が必要となると予想されます。
- 以上のことから、情報システムにおいては、どんなログをどのように取得し、どこに保管するか決めて、ログを取得、管理しておくことが大切です。DBシステムにおいても、ログの取得、保管が求められています。

注1：否認とは(Repudiation)とは、当事者が行った行為を、行為が行われなかった、他人による なりすましである、改ざん された、等々と主張して否定することを言います。

DBにおける監査ログ取得の考慮点(1)

- 何の目的で(Why)
  - アクセスの証拠になり、否認防止などにも利用可
- 何を(What)
  - 証跡となるもの
    - 利用者が当システム上でどのような操作を行なったかを記録
    - 利用目的によって取得する内容が異なる
    - 基本的な内容は
      - 誰が
      - いつ
      - 何を行ったか
    - パフォーマンスとのトレードオフ
    - ログ量の見積もりが必要
    - 「誰が」行ったかを監査するためにはユーザーID管理が必須
- どこで(Where)
  - DBで取得するのか、DB以外の場所で取得するのか
    - アプリケーションでのログ
    - 機能の制約、監査のしやすさ、ログ取得の目的も考慮

解説：DBにおける監査ログ取得の考慮点（1）

- DBシステムにおいて監査ログを取得する際の考慮点としては、以下が挙げられます。
  これらの事柄について、事前によく考え、ログの取得、管理を行う必要があります。

- 何の目的で(Why)
  - ログは、一般にアクセスの証拠になり、否認防止などにも利用できますが、何の目的でログをとるのかによって何を取得しておくべきがが異なります。
  - 目的の例としては、次のようなものがあります。
    - 不正アクセスを試みた者がいないかチェック
      - アクセスの連続失敗などがないか？
      - パスワードを何回も間違えていないか？
      - 主に認証に関する失敗記録が必要
    - 改ざんがあった場合発見する、更新について否認を防止する
      - データの更新記録を保存
    - データの盗難について証跡を残す
      - 読み取りも含めすべてのアクセスを記録

解説：DBにおける監査ログ取得の考慮点（1）

- 何を(What)
  - ログとして何をとるのか
    - 監査ログはアクセスの証跡となるもので、利用者が当システム上でどのような操作を行なったかを記録します。
    - 上述のログをとる目的によって、取得する内容は異なります。
  - 基本的な内容は以下の内容となります。
    - 誰が
    - いつ
    - 何を行ったか
  - すべてのアクセスを取得しようとすれば、ログ量は膨大となり、ログ取得のシステム負荷も高まります。一方、一部だけ取得すれば、抜けが生じ、証拠能力として劣ることになります。
    - たとえば、オンライン時間帯のみログをとっている場合、バッチ時間帯に不正がなかったことを証明できない。
    - 全件のSQLをログすると膨大なログ量になるが、更新のみ記録すると不正な読み取りは発見できない、など。
    - ログ採取に必要な資源(CPU負荷、ディスク容量)などの見積もりも必要となります。
    - 「誰が」を監視するためには、操作を行った個人が特定できるようにIDを管理することが前提となります。
      - IDの共有が行われていると、あるIDで行われた操作を行った人を特定することはできません。
      - WASからの接続のように、同じIDで実際のユーザーは多数であるような場合、各ユーザーの処理を区別することはできません。情報を特殊レジスタに設定することによりDBサーバー側で各ユーザーの処理を区別することが可能となります。
      - 特殊レジスタを使わない場合には、アプリケーション側でログを取得することを検討してください。

- どこで(Where)
  - DBで取得するのか、DB以外で取得するのかの決定には、パフォーマンス要件の他、ユーザーIDの問題、ログデータの監査のしやすさなども関係します。上記のように、ユーザーIDが共有されたセッションがある場合には、DB側でのログは監査上あまり役立ちません。
    また、DB側でのログでは、ユーザーが実行している事柄とログの内容の対応付けが難しいこともあります。これらのことを考慮すると、アプリケーションでログをとるということも1つの案となります。

DBにおける監査ログ取得の考慮点(2)

- どうやって(How)
  - 採取方法
    - どんなツールで
  - 解析方法
    - 監視項目
    - レポートツール
    - 解析のタイミング
  - 保存方法、保存場所、保存期間
    - 保存方法
      - ファイル
      - DB
      - アーカイブ
    - ログを一箇所に収集すればログ管理に役立つほか、ログ改竄を防げる可能性が高まる
      - ローカル
        - 各サーバごとにローカルに保存
      - リモート
        - ログ収集サーバを設置し、ログをそのサーバ上に収集
        - 収集のタイミング
    - 保存期間を決める
    - 保存に必要な資源(ディスクなど)の見積もりも必要

解説：DBにおける監査ログ取得の考慮点（2）

- 監査ログを実際に取る上では、以下のどのように(How)とるかについて具体的に検討していかなければなりません。
- 採取方法
  - まず、どのようなツールでログを採取するか決定することが必要です。DB2　for z/OSではトレース機能、DB2　for Linux, Unix, Windows (LUW)ではdb2auditが監査ログ採取用のツールとして標準に提供されています。
    標準機能以外でログを採取する場合は、使用したいツールのベンダーにお問い合わせください。
- 解析方法
  - 採取したログの解析方法としては、以下の事柄について検討してください。
  - 監視項目
    - どんな目的で、何のログを採取するかと関連します。目的に応じた項目を監視するよう計画します。
  - レポートツール
    - 監視のためのレポートはどのような形式にするか、レポートツールがあるのか、レポートツールで監視項目の要件が満足されるのか、作りこみが必要なのか検討します。
  - 解析のタイミング – リアルタイムか／バッチか
    - 上記のDB2提供ツールで監査ログを取得する場合、採取したログを加工する必要があるので、リアルタイムの解析は困難です。
      ただし、レポート期間は一日一回なのか、一時間遅れでデータを見たいのか、あるいは何かあったときにだけ過去ログを見るのかにより運用が異なりますので、どのようなタイミングで解析するのか検討が必要です。
- 保存方法、保存場所、保存期間
  - 保存方法
    - 保存方法としては、ファイルとして保存する、DBに保存する、加工したり、別の場所、別のメディアなどにアーカイブするなどの選択肢があります。
  - 保存場所
    - 保存場所では、ローカル（ログを取得したサーバ上)に置くのか、別にログ管理サーバなどを用意し、そこに転送して保管するのを検討します。(アーカイブの場合にはメディアを保管)
    - ログを一箇所に収集すれば、ログ管理に役立つほか、ログ管理サーバのセキュリティーを強化することにより、ログ改ざんを防げる可能性が高まります。
      ログをリモートで保管する場合には、転送タイミングも決定する必要があります。
  - 保存期間
    - お客様の方針、法律の要請などにしたがって、保存期間を決めます。
  - 保存方法、保存期間などにしたがって、保存に必要な資源の見積もりも必要です。

DB2 LUW 監査機能

お断り：当資料は、DB2 for Linux, UNIX and Windows V9.5 をベースに作成されています。

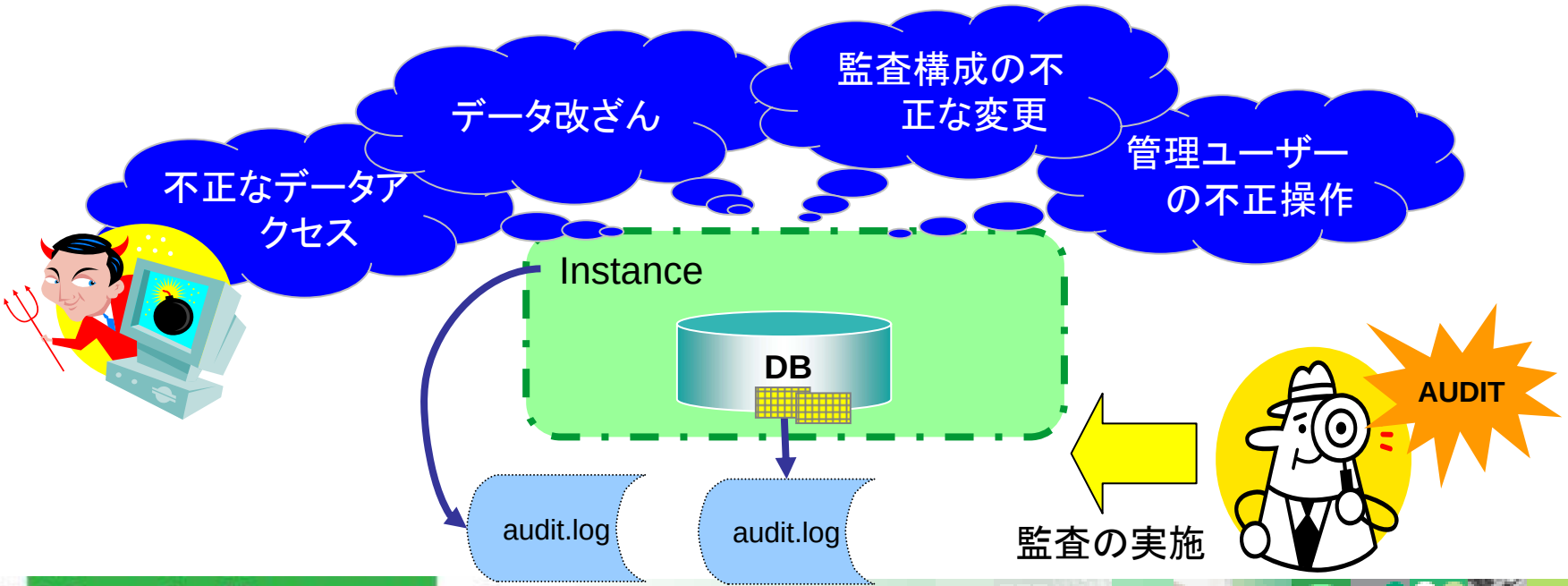
- この文書はDB2 9.5 を前提にして記述されています。
  - 文書内のテスト結果では以下のFix Levelが使用されています。
    - DB2 9.5 FP1

目次

- DB2 for LUW 監査機能
- DB2 for LUW 監査運用
  - 監査の構成
  - 監査ログファイルのアーカイブ
  - 監査ログの抽出
  - 監査ログの解析

DB2 for LUWの監査機能

- DB2 for LUWの監査機能としてDB2AUDIT機能が提供される。
  - 監査を取得するカテゴリーを選択する。
    - 監査構成の変更、オブジェクトの作成/削除、監査構成操作、実行ステートメントなどの監査が可能
  - インスタンス・レベルの監査とデータベースレベルの監査（V9.5～）がある。
  - 監査対象となるデータベースや、オブジェクトを絞り込むことができる。（V9.5～）
  - 監査ログはファイル、またはCSV形式として抽出し、データベース中の表にデータを投入、解析することができる。



DB2 for LUWの監査機能

- DB2 for LUWの監査ログ取得
  - インスタンス・レベルの監査とデータベースレベルの監査（V9.5～）がある
- 監査対象となるカテゴリー
  - 以下のカテゴリーについて監査を取得することができる。

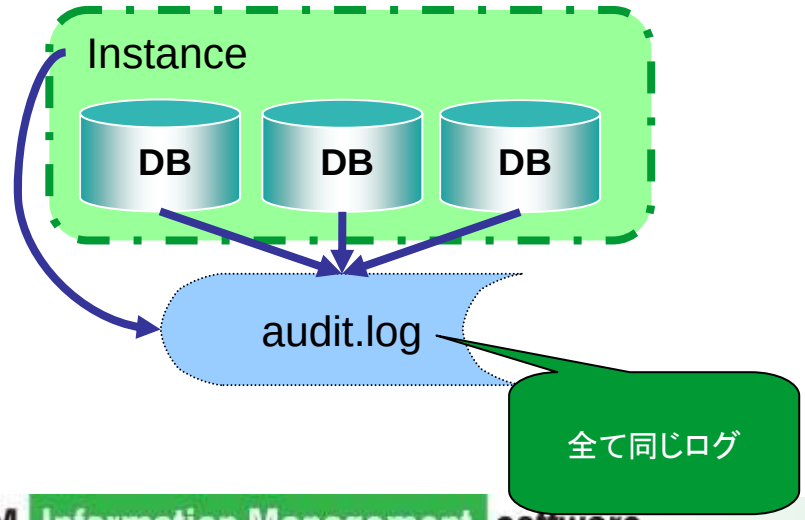
CATEGORY	説明	インスタンスレベル監査	データベースレベル監査（V9.5～）
ALL	全てのカテゴリ	○	○
AUDIT	監査設定が変更されたとき、または監査ログにアクセスされたときに	○	○
CHECKING	データベース・オブジェクトまたは関数へのアクセス試行またはその操作試行の許可検査中	○	○
OBJMAINT	オブジェクトの作成、除去時	○	○
SECMAINT	オブジェクト、データベースの特権またはDBADM権限の付与、取り消し時	○	○
SYSADMIN	SYSADM、SYSMAINT、SYSCTRL権限が必要とされる操作の実行時	○	○
VALIDATE	ユーザーの認証時、SECURITY情報の検索時	○	○
CONTEXT	データベース操作実行時のログ	○	○
EXECUTE（V9.5～）	SQL ステートメントの実行時 （データベースレベル監査のみで有効）	×	○

DB2 9.5　AUDIT機能の変更点(概要)

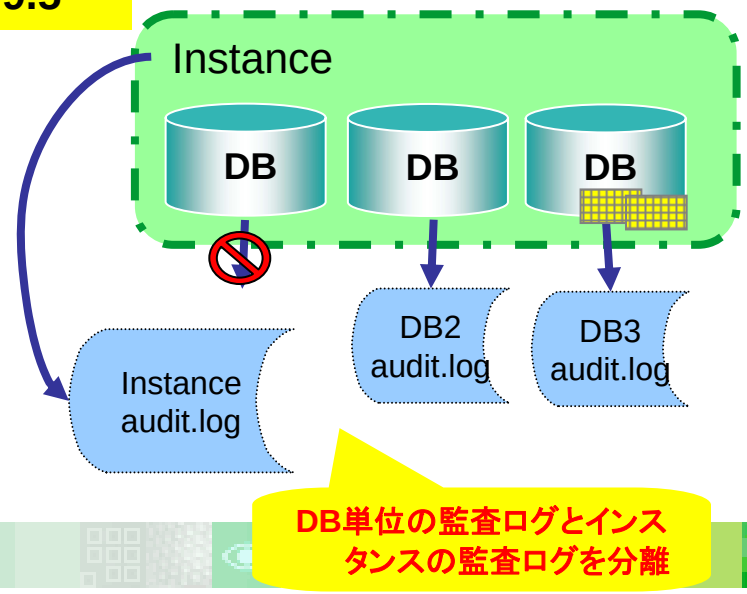
- 背景
  - 同一インスタンスの監査を一つの構成ファイルで管理し、一つのログファイルに出力していた
  - 監査対象としたいDBが複数あるうちの一つであっても、全てのDBを監査対象としてロギングされていた

- 新しい監査機能の特徴
  - インスタンスと各データベースで個別のログを出力することが可能
  - 新しい監査区分の登場や、監査構成のカスタマイズにより、より細かい監査構成が可能
  - 監査対象となるデータベース内のオブジェクトを詳細に選択できるようになった
    - 監査不要なデータベース・オブジェクトのイベントを監査しなくてよい
  - 監査ログ取得処理によるシステム負荷を抑制

9.1以前



9.5～



DB2 9.5　AUDIT機能の変更点（補足）

- データベース単位に監査ログファイルを作成可能
- 監査ログのアーカイブ
  - 監査ログを、アクティブな監査ログとは別のパスにアーカイブすることが可能
  - アーカイブされた監査ログに対して抽出（extract)するように変更された。
    - 監査ログの書き込みと、抽出操作の干渉を防ぐ
- 監査ログパスの変更が可能
  - 以前は監査ログパスが固定であったが、任意のパスを指定することができるようになった。
- AUDIT　CATEGORYの導入
  - データベース単位の監査を実施するカテゴリー。より細かいレベルで取りたい情報のみを監査するために新しいデザインが導入された。
  - EXECUTEカテゴリーの追加
    - データベースレベルで実行されたステートメントを監査する
    - 取得したいオブジェクトのみに限定できる
    - パラメータマーカー、ホスト変数も取得可能。
- db2audit コマンドの変更
  - Pruneコマンドがなくなった。
    - Archiveコマンドを利用するよう変更
  - Extractコマンド
    - アクティブ監査ログに対しては使用できない
  - Configureコマンド
    - datapath/archivepathを指定して監査ログパスの変更を行う
  - Start/stopコマンド
    - インスタンス・レベルの監査のみに有効。

DB2 9.5「監査ポリシー」を使用した監査構成

監査対象を特定のオブジェクトに絞ることができる

要件	設定	対象
特定テーブル上での実行SQLを記録したい	EXECUTE category	特定表（EMPLOYEE表）
特定権限の管理者の操作を記録したい	SYSADMIN category	特定権限（SYSADM）
特定ユーザーの表へのアクセスを記録したい	CHECKING category	特定USER（BOB）
特定ロールのユーザー認証を記録したい	VALIDATE category	特定ROLE（ACCOUNTING）

AUDITポリシーを作成し、AUDITコマンドでオブジェクトに関連付ける

CATEGORY

- ALL
- AUDIT
- CHECKING
- OBJMAINT
- SECMAINT
- SYSADMIN
- VALIDATE
- CONTEXT
- EXECUTE

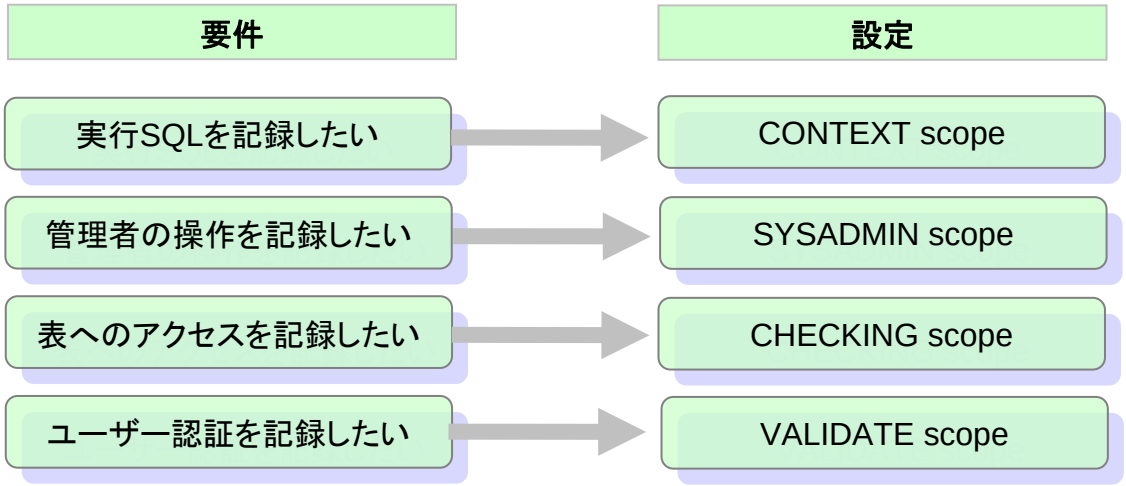
AUDIT POLICY → AUDIT →

- Database
- Tables
- Authorities
- Users
- Groups
- Roles
- Trusted Connections

DB2 9.5「監査ポリシー」を使用した監査構成

- 監査ポリシー（AUDIT POLICY）
  - 必要なデータやオブジェクトに関する情報だけを収集するためのデータベース・オブジェクト
  - セキュリティー管理者（SECADM）のみ作成可能
  - 監査を行いたい実施操作のカテゴリーを選択する
  - AUDITコマンドを使用して、作成された監査ポリシーを監査対象とするデータベース・オブジェクトに関連付ける
  - 個々のデータベース内で何を監査の対象にするかを制御する
- 監査ポリシーを関連付けることができるオブジェクト
  - データベース全体
  - 表
  - トラステッド・コンテキスト
  - ユーザー、グループ、またはロールを示す許可 ID
  - 権限
    - SYSADM、SECADM、DBADM、SYSCTRL、SYSMAINT、SYSMON

参考：V9.1までのdb2auditの機能



インスタンス単位の監査構成。

監査対象を絞ることはできなかった。

Scope	
ALL	全てのカテゴリ
AUDIT	Auditの構成変更、Auditログへのアクセスなどのイベント発生時 またはAUDIT_BUF_SZの変更時
CHECKING	DB2オブジェクトまたは機能にアクセス時または処理時の許可検査
OBJMAINT	オブジェクトの作成、除去時
SECMAINT	オブジェクト、データベースの特権またはDBADM権限の付与、取り消し時
SYSADMIN	SYSADM、SYSMAINT、SYSCTRL権限が必要とされる操作の実行時
VALIDATE	ユーザーの認証時、SECURITY情報の検索時
CONTEXT	データベース操作実行時のログ

DB2 9.5 監査ツール

- DB2AUDITコマンド
  - DB2 for LUWの監査ログ取得用の標準ツール
    - V9.5以前は監査の構成、開始、停止含めて全てdb2auditコマンドを使用していた。
    - V9.5以降もインスタンス・レベルの監査はdb2auditを利用して構成する
  - 以下の操作が可能
    - インスタンス・レベルの監査構成
      - データベース・レベルの監査構成は監査ポリシー作成とAUDITコマンドによって行う。(次頁参照)
    - インスタンス・レベルの監査開始・停止
    - 監査ログ・パスの指定
    - 監査ログのアーカイブ
    - 監査ログレコードの抽出
- AUDIT POLICY(監査ポリシー)
  - データベース・レベル監査の監査に利用、監査を行いたい実施操作のカテゴリーを選択する(詳細は後述)
- AUDIT コマンド
  - データベース・レベル監査の監査に利用、作成した監査ポリシーを監査したいオブジェクトに関連付ける(詳細は後述)

データベース・レベル監査の構成例①

- 特定テーブルへのアクセスを監査する
  - EMPLOYEE 表に極めて機密性の高い情報が含まれており、その表のデータに対する SQL アクセスを監査する
  - 表に対する全アクセスのトラッキングには、EXECUTE 区分を使用する。
  - SQL ステートメント、およびオプションとしてそのステートメントの実行時に提供される入力データの値も監査可能

```
CREATE AUDIT POLICY SENSITIVEDATAPOLICY
   CATEGORIES EXECUTE
   STATUS BOTH
   ERROR TYPE AUDIT;

COMMIT;

AUDIT TABLE EMPLOYEE USING POLICY SENSITIVEDATAPOLICY;

COMMIT;
```

EXECUTEカテゴリーで AUDIT POLICYを作成

AUDITコマンドで、監査ポリシーとオブジェクトを関連づける。

データベース・レベル監査の構成例②

- SYSADM または DBADM によるアクションすべてを監査する例
  - SYSADM または DBADM によるデータベース内でのすべてのアクションを収集するためには、EXECUTE 区分と SYSADMIN 区分の両方を監査する。
  - SYSADM または DBADM 権限を保持しているすべてのユーザーについて、すべての監査可能イベントのログが記録される。

```
CREATE AUDIT POLICY ADMINSPOLICY
  CATEGORIES
    EXECUTE STATUS BOTH,
    SYSADMIN  STATUS BOTH
  ERROR TYPE AUDIT;

COMMIT;

AUDIT SYSADM, DBADM USING POLICY ADMINSPOLICY;

COMMIT;
```

CATEGORIESを複数指定した場合、そのOR条件で監査される。指定すれば指定するほどログは多くなる。

DB2 LUW 監査運用

お断り：当資料は、DB2 for Linux, UNIX and Windows V9.5 をベースに作成されています。

DB2 for LUW 監査運用

- データベースレベルの監査（V9.5～）とインスタンス・レベルの監査の運用項目

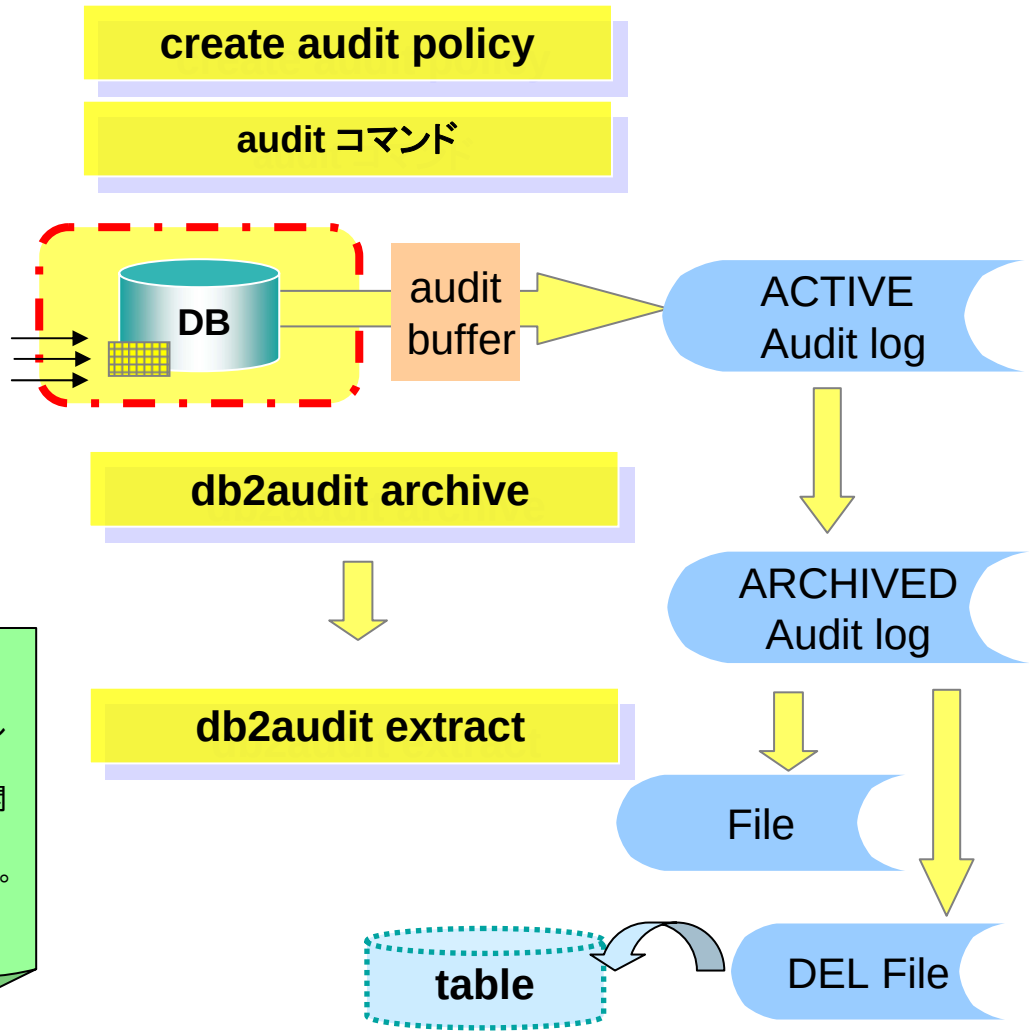
運用項目		データベースレベル監査（V9.5～）	インスタンス・レベル監査
監査の構成	監査ログパスの構成	•db2audit configure datapath / archivepath	
	監査の構成	•監査ポリシーの作成 –CREATE AUDIT POLICY 監査対象オブジェクトの決定 – AUDIT	•監査の構成 –db2audit configure •監査の開始・停止 –db2audit start/stop
監査ログファイルのアーカイブ		•db2audit archive	
監査ログの抽出		•db2audit extract	
監査ログの解析		•TEXTファイルの解析、またはDEL形式データを表にロード	

V9.5　データベース・レベル監査の実施方法

- 監査の構成
  - 監査ポリシー作成
    - CREATE AUDIT POLICY
  - 監査対象オブジェクトの決定
    - AUDITコマンド
- 監査ログファイルのアーカイブ
  - db2audit archive
- 監査レコードの抽出
  - db2audit extract
- 監査ログの解析

MEMO:　以前のバージョンからの変更点:

- db2audit configureは今までどおりのインスタンスレベルの監査構成では使用可能。
- db2audit start/stopはインスタンス・レベルでの監査の開始・停止に必要。DBレベルの監査には必要ない。
- Extractはアーカイブしたファイルに対してのみ実行可能。
- Prune　historyコマンドはなくなった。

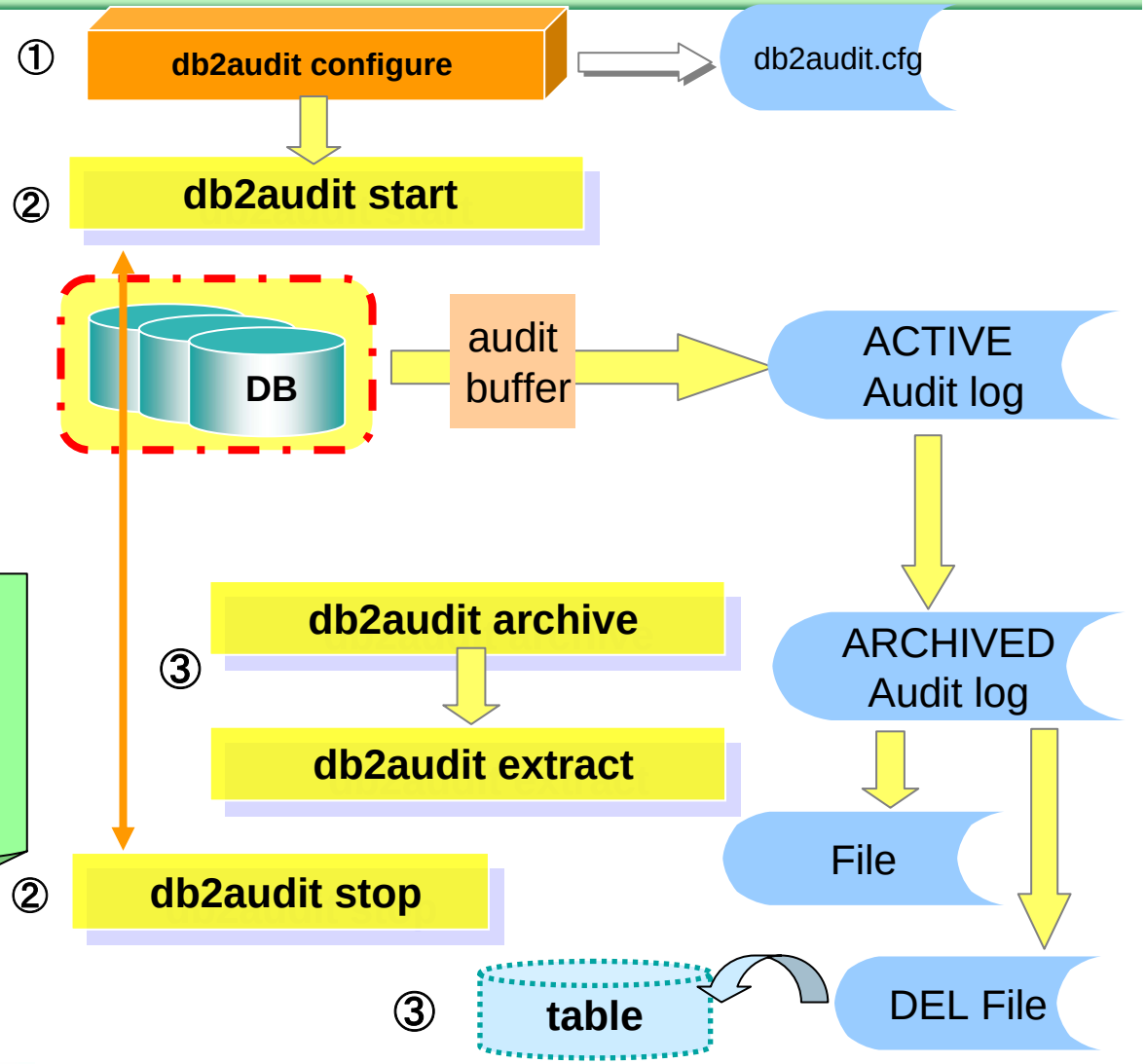


V9.5　インスタンス・レベル監査の実施方法

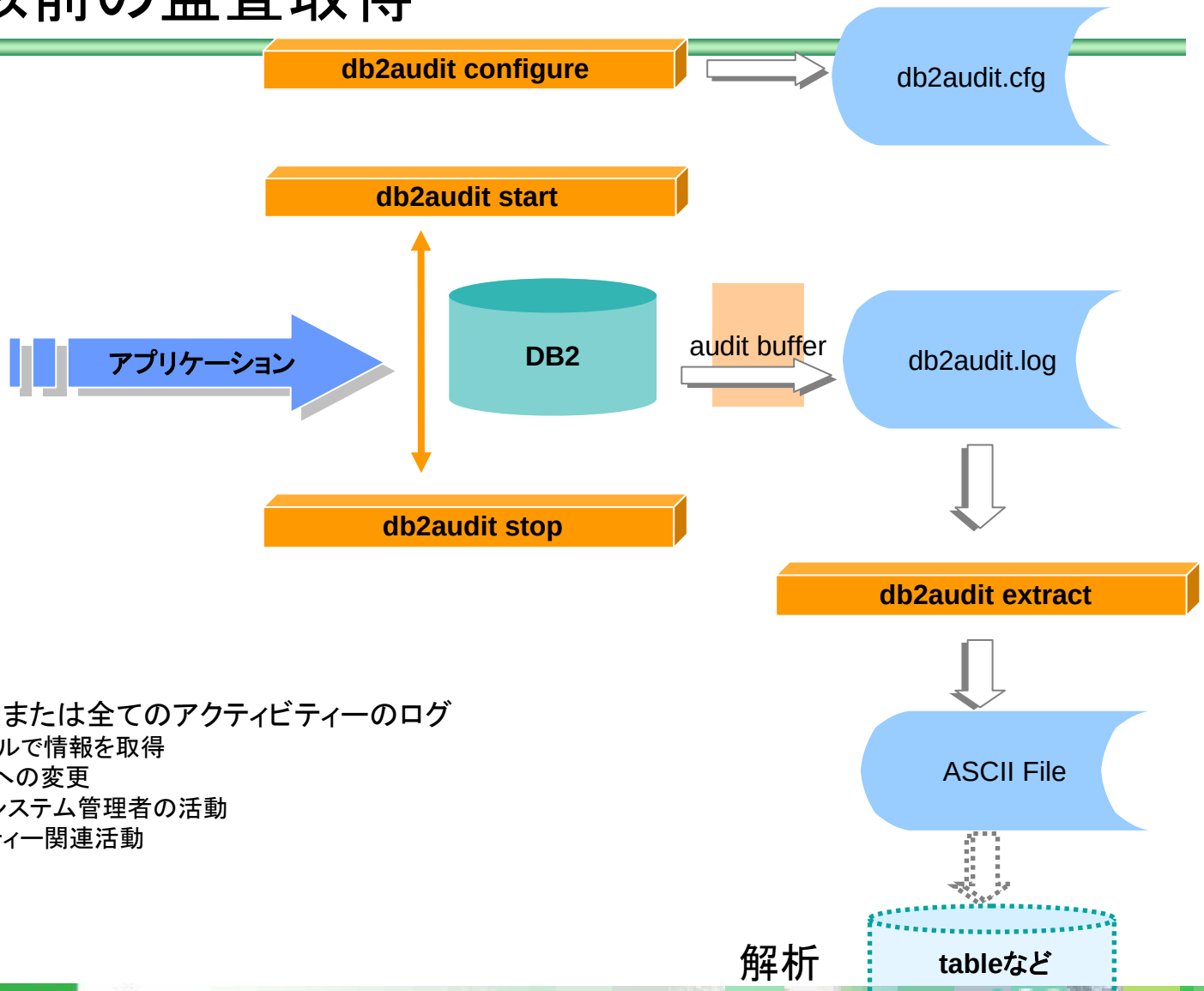
- 監査の構成
  - db2audit configure
- 監査の開始・停止
  - db2audit start/stop
- 監査ログファイルのアーカイブ
  - db2audit archive
- 監査レコードの抽出
  - db2audit extract
- 監査ログの解析

MEMO:　以前のバージョンからの変更点:

- db2audit configureは今までどおりのインスタンスレベルの監査構成で使用可能。
- Extractはアーカイブしたファイルに対してのみ実行可能。
- Prune　historyコマンドはなくなった。



参考：V9.1以前の監査取得



- DB2で発生する特定または全てのアクティビティーのログ
  - インスタンスレベルで情報を取得
  - DB2オブジェクトへの変更
  - データベースやシステム管理者の活動
  - 全てのセキュリティー関連活動

監査ログファイルと監査ログパスの構成

- アクティブ監査ログとアーカイブ監査ログ
  - V9.5より、2種類の監査ログが存在する。監査ログの抽出(EXTRAXT)を行う場合、アクティブ監査ログを一度db2audit archiveコマンドによりアーカイブし、作成されたアーカイブ監査ログに対してログの抽出を行う。
- アクティブ監査ログ
  - 監査情報がバイナリーで記録される。ログのネーミングは以下

インスタンス監査ログ(アクティブ)
db2audit.instance.log.node_number
データベース監査ログ(アクティブ)
db2audit.db.dbname.log.node_number

非区分化DBでは
node_numberは"0"
となる

  - デフォルトの監査ログパス
    - Windows
      - C:¥Documents and Settings¥All Users¥Application Data¥ibm¥DB2¥DB2コピー名¥DB2インスタンス名¥security¥auditdata
    - Linuxおよび UNIX
      - *instance*/sqllib/security/auditdata
- アーカイブ監査ログ
  - アクティブ監査ログがアーカイブされる。
  - アーカイブコマンドが実行されたタイムスタンプが付加される。
  - デフォルトではアクティブ監査ディレクトリと同一パスに作成される。

インスタンス監査ログ(アクティブ)
db2audit.instance.log.node_number.timestamp
データベース監査ログ(アクティブ)
db2audit.db.dbname.log.node_number.timestamp

ローカル時刻が
YYYYMMDDHHMMSS形
式で付く

監査ログファイルと監査ログパスの構成

- 監査ログパスの変更
  - アクティブ監査ログ、アーカイブ監査ログともにパスを変更することができる。
  - アクティブ監査ログの変更
    - db2audit configure datapath <log_path>
  - アーカイブ監査ログパスの変更
    - db2audit configure archivepath <log_path>
  - 監査ログパスに区分化ノードのノード表現が使用できる
    - db2audit configure datapath “/auditForNode $N”

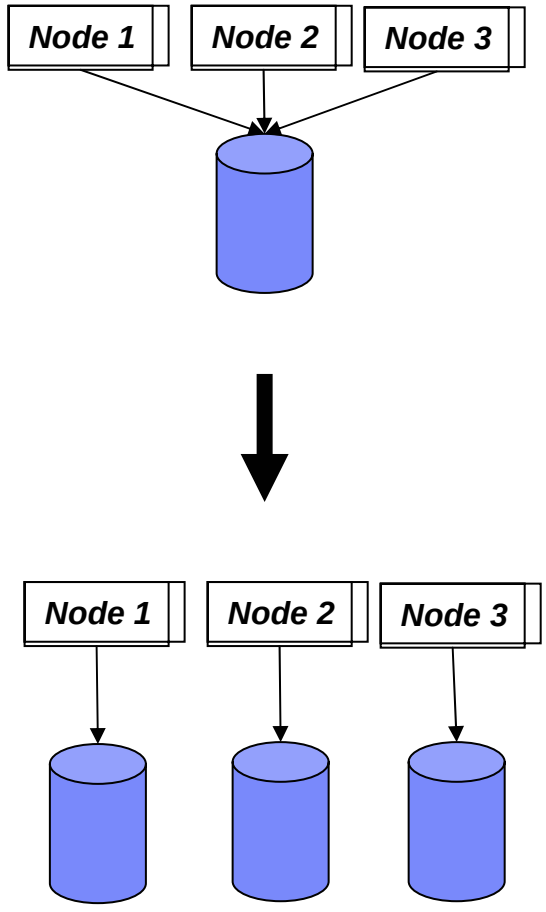
“ $N”がノード番号を示す。頭にブランクが必要。

```
$ db2audit configure datapath /log/audit/active
AUD0000I  Operation succeeded.
$ db2audit configure archivepath  /log/audit/archive
AUD0000I  Operation succeeded
$ db2audit describe
DB2 AUDIT SETTINGS:
......
Audit Data Path:  “/log/audit/active/”
Audit Archive Path: ”/log/audit/archive/”
AUD0000I  Operation succeeded.
```

describe コマンドで確認

監査ログファイルと監査ログパスの構成

- 区分化環境では、監査ログパスに”partition expression”を含むことができる。
  - 2つの”partititon expressions”をパスに含む例：
    - db2audit configure datapath ″D:¥auditLogs $N+1 _ $N+1″
  - 現在のデータベース区画が0であれば、監査ログパスは以下になる
    - D:¥auditLogs1_1”.
  - ”partititon expressionsはブランクで始まり、$Nを続ける。

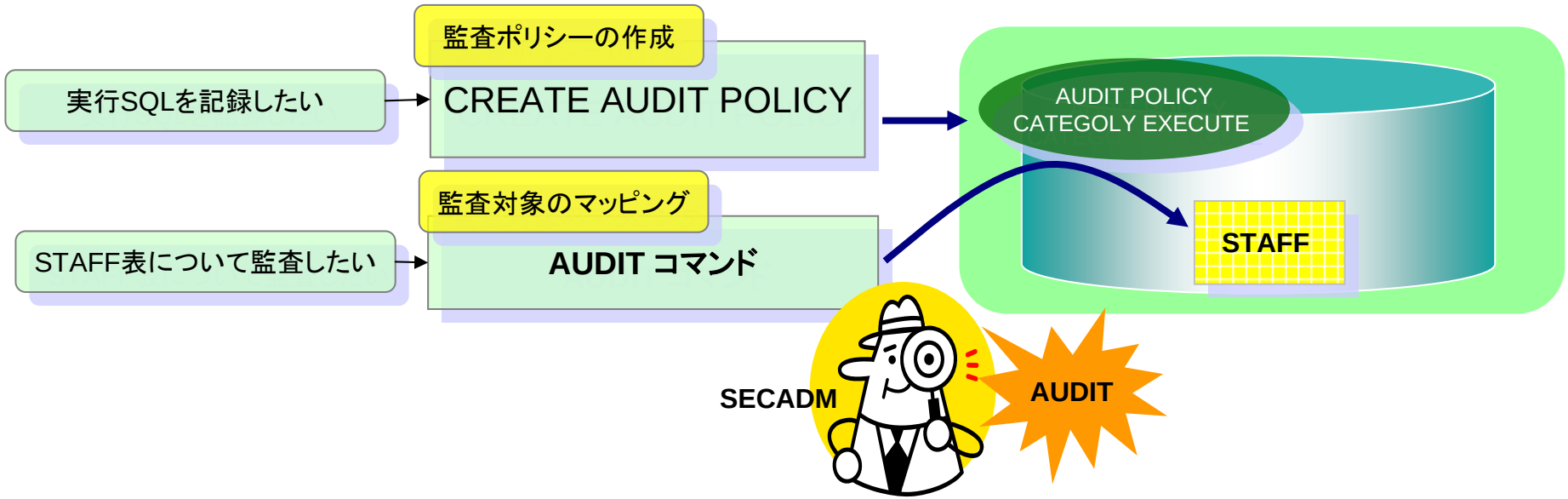


サポートされる”partition expression”のフォーマット

Syntax	Example	Result
[blank]$N	" $N"	"23"
[blank]$N+[number]	" $N+1000"	"1023"
[blank]$N%[number]	" $N%4"	"3"
[blank]$N+[number]%[number]	" $N+19%8	"2"
[blank]$N%[number]+[number]	" $N%3+20"	"22"

データベース・レベル監査の構成

- データベース・レベルの監査を行うためには、監査ポリシー（AUDIT POLICY)を作成し、どのような監査情報カテゴリーを取得するか決定する。
- 次に監査ポリシーを、AUDITコマンドによって監査対象オブジェクトにマッピングし、監査を実行する。
- データベース・レベル監査の構成はSECADM権限ユーザーで実施する必要があるため、SECADM権限ユーザーを決定し、このユーザーで監査の構成を行う。



監査ポリシーの作成

- データベースに接続し監査ポリシーを作成する。
  - CREATE AUDIT POLICY ステートメント: 監査ポリシーを作成
  - ALTER AUDIT POLICY ステートメント: 監査ポリシーの変更
  - DROP AUDIT POLICY ステートメント: 監査ポリシーの削除

- 監査カテゴリー
  - 個別に監査することができるイベントのタイプ。 監査ポリシー作成時に指定する。

CATEGORIES	
ALL	全てのカテゴリ
AUDIT	監査設定が変更されたとき、または監査ログにアクセスされたときに
CHECKING	データベース・オブジェクトまたは関数へのアクセス試行またはその操作試行の許可検査中
OBJMAINT	オブジェクトの作成、除去時
SECMAINT	オブジェクト、データベースの特権またはDBADM権限の付与、取り消し時
SYSADMIN	SYSADM、SYSMAINT、SYSCTRL権限が必要とされる操作の実行時
VALIDATE	ユーザーの認証時、SECURITY情報の検索時
CONTEXT	データベース操作実行時のログ
EXECUTE NEW!	SQL ステートメントの実行 時 （データベースレベル監査のみで有効）

EXECUTE カテゴリー

- ユーザーが発行する SQL ステートメントを的確にトラッキングすることが可能
  - SQLステートメントの実行の監査が目的の場合、以前使用されていた、CONTEXTカテゴリーの代わりとなる。
- SQL ステートメントのテキストに加えて、コンパイル環境や、後でステートメントを再現するのに必要なその他の値も収集可能
- オプションによって入力ホスト変数や、パラメータ・マーカーの値も取得可能
- 静的 SQL と動的 SQL の両方のステートメント・テキストが記録される
- 主な収集可能情報
  - ステートメント・テキスト
  - データ・タイプ、長さ
  - 入力ホスト変数、パラメータ・マーカ （WITH DATAオプションをつけた場合）
    - LOBS, LONG, XML, 構造化タイプは含まれない
  - コンパイル環境、変更または、リターンされた行数

CREATE AUDIT POLICY シンタックス

- 監査ポリシーの作成をおこなう

CATEGORIES:

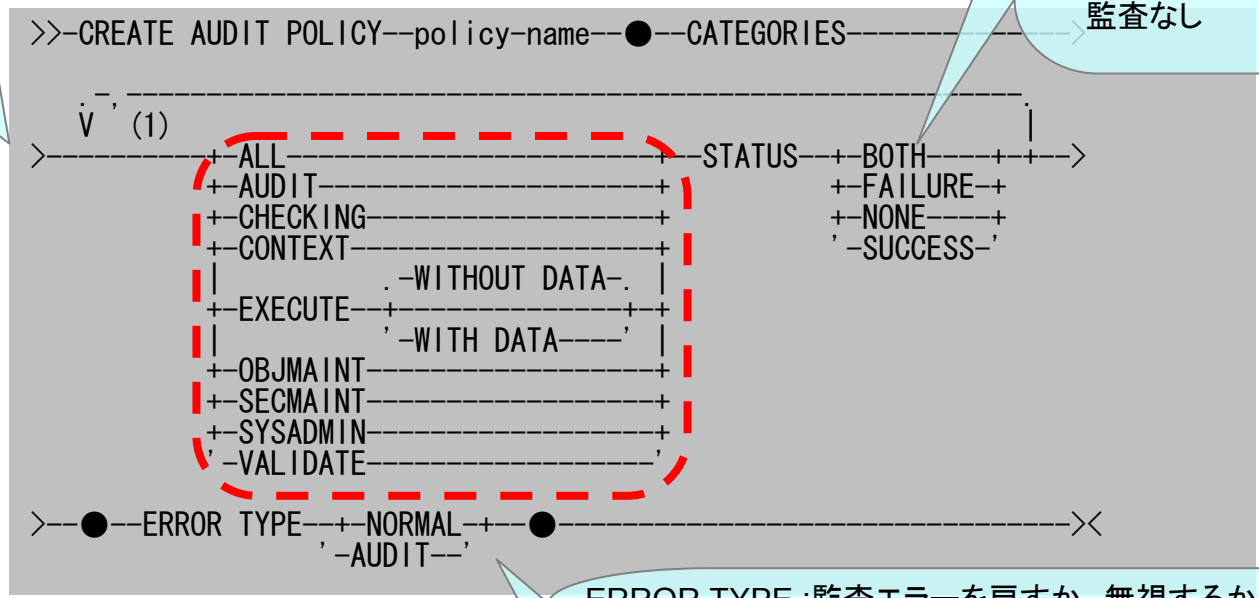
以前のバージョンのdb2audit configure scopeと似ているが、SQLステートメント操作を監査するための“EXECUTE”カテゴリーが追加されている。

BOTH :
　成功、失敗イベント両方監査

FAILURE :
　失敗イベントのみ監査

SUCCESS :
　成功イベントのみ監査

NONE :
　監査なし



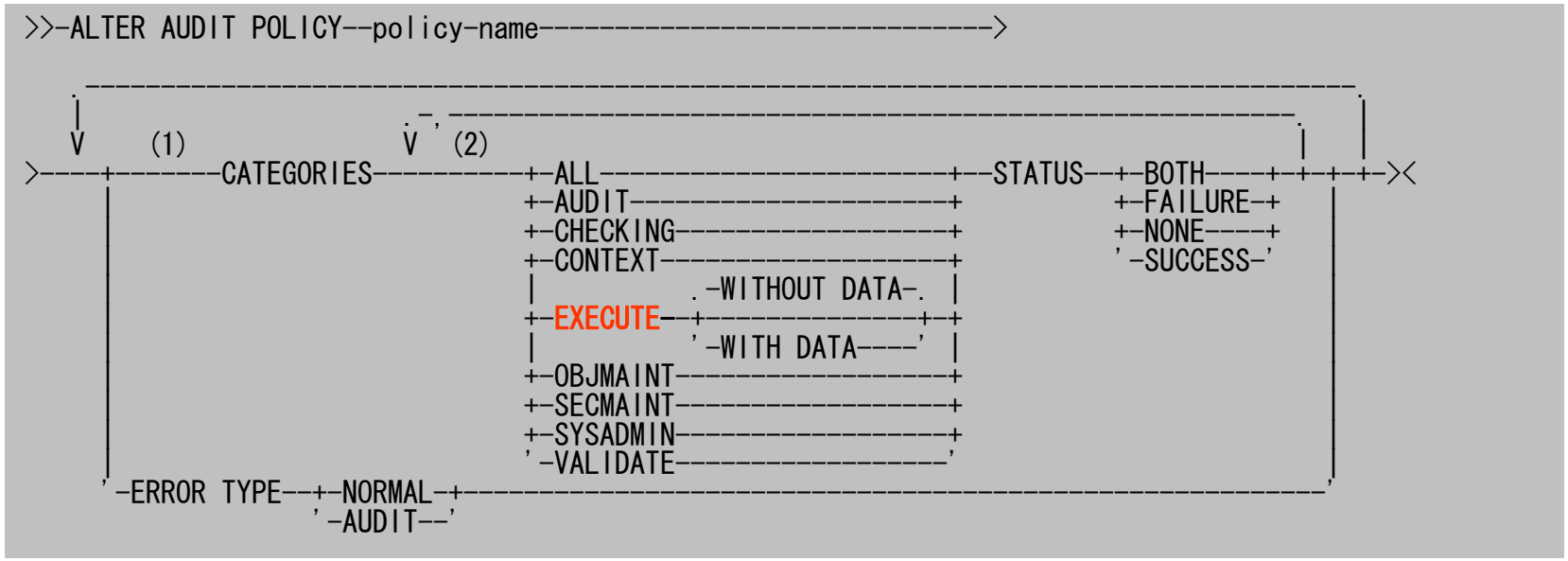
ERROR TYPE :監査エラーを戻すか、無視するかを指定

NORMAL : 監査によって生成されたエラーはすべて無視され、
　実行される操作に関連したエラーのみ

AUDIT : 監査機能自体で発生したエラーを含む、すべてのエラー

ALTER AUDIT POLICY　シンタックス

- 監査ポリシーの変更をおこなう



監査ポリシーの作成例

- SECADM権限の付与

```
$GRANT SECADM ON DATABASE TO USER DB2SEC
DB20000I  The SQL command completed successfully.
```

- 監査ポリシー作成DDLの例

```
CONNECT TO SAMPLE USER DB2SEC USING xxxxxxxxx;
CREATE AUDIT POLICY auditdb CATEGORIES ALL STATUS BOTH ERROR TYPE NORMAL;
COMMIT;
CREATE AUDIT POLICY admins CATEGORIES SYSADMIN STATUS BOTH ERROR TYPE NORMAL;
COMMIT;
CREATE AUDIT POLICY powerusers   CATEGORIES OBJMAINT STATUS BOTH,SECMAINT STATUS BOTH ERROR TYPE AUDIT;
COMMIT;
CREATE AUDIT POLICY context CATEGORIES CONTEXT STATUS BOTH ERROR TYPE NORMAL;
COMMIT;
ALTER AUDIT POLICY context CATEGORIES CONTEXT STATUS FAILURE ERROR TYPE NORMAL;
COMMIT;
ALTER AUDIT POLICY auditdb CATEGORIES EXECUTE WITH DATA STATUS BOTH ERROR TYPE NORMAL;
COMMIT;
DROP AUDIT POLICY context;
COMMIT;
```

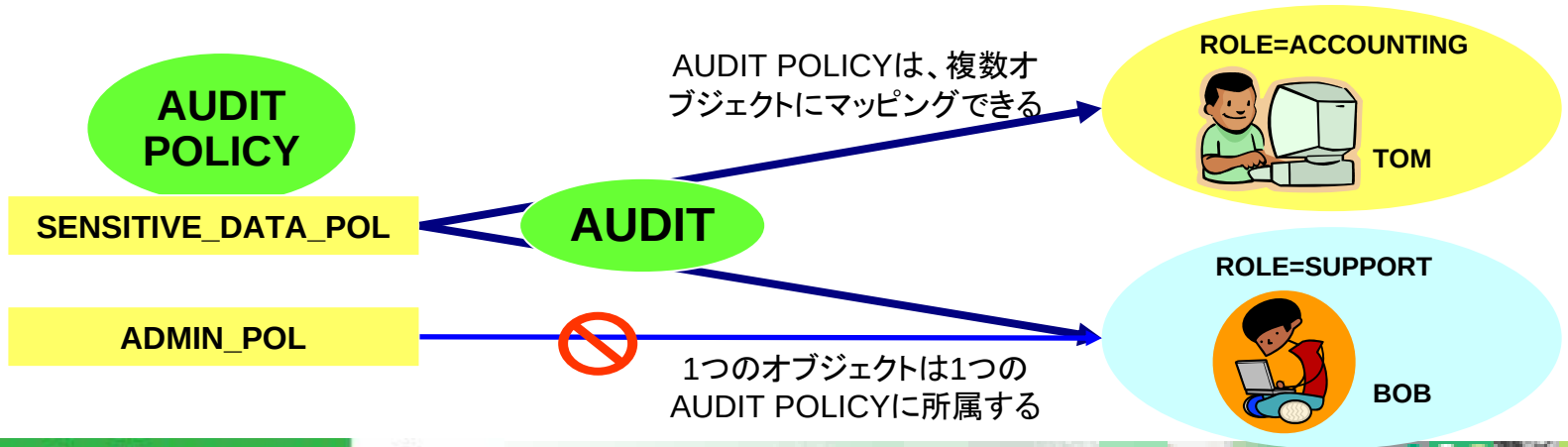
COMMITをしない場合以下のエラー
DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned:
SQL4708N Only a COMMIT or ROLLBACK statement is allowed at this time for this unit of work. SQLSTATE=5U021

CREATE/ALTER AUDIT POLICY　注意点

- AUDIT 排他 SQL ステートメントの後は、COMMIT または ROLLBACK ステートメントでなければいけない（SQLSTATE 5U021）
- AUDIT 排他 SQL ステートメント
  - AUDIT
  - CREATE AUDIT POLICY、ALTER AUDIT POLICY、または DROP AUDIT POLICY
  - DROP ROLE および DROP TRUSTED CONTEXT
    - そのロールやトラステッド・コンテキストが監査ポリシーと関連付けられる場合
- AUDIT 排他 SQL ステートメントをグローバル・トランザクション（例えば XA トランザクション）内で発行することはできない（SQLSTATE 51041）
- データベース・パーティション全体を通じて、同時に実行できる非コミットの AUDIT 排他 SQL ステートメントは 1 つのみ。非コミットの AUDIT 排他 SQL ステートメントが実行されている場合、後続の AUDIT 排他 SQL ステートメントは、現行の AUDIT 排他 SQL ステートメントがコミットまたはロールバックされるまで待機する。
- 変更はシステム・カタログに書き込まれるが、コミットされるまでは有効にならない。これは、ステートメントを発行する接続の場合でも当てはまる。

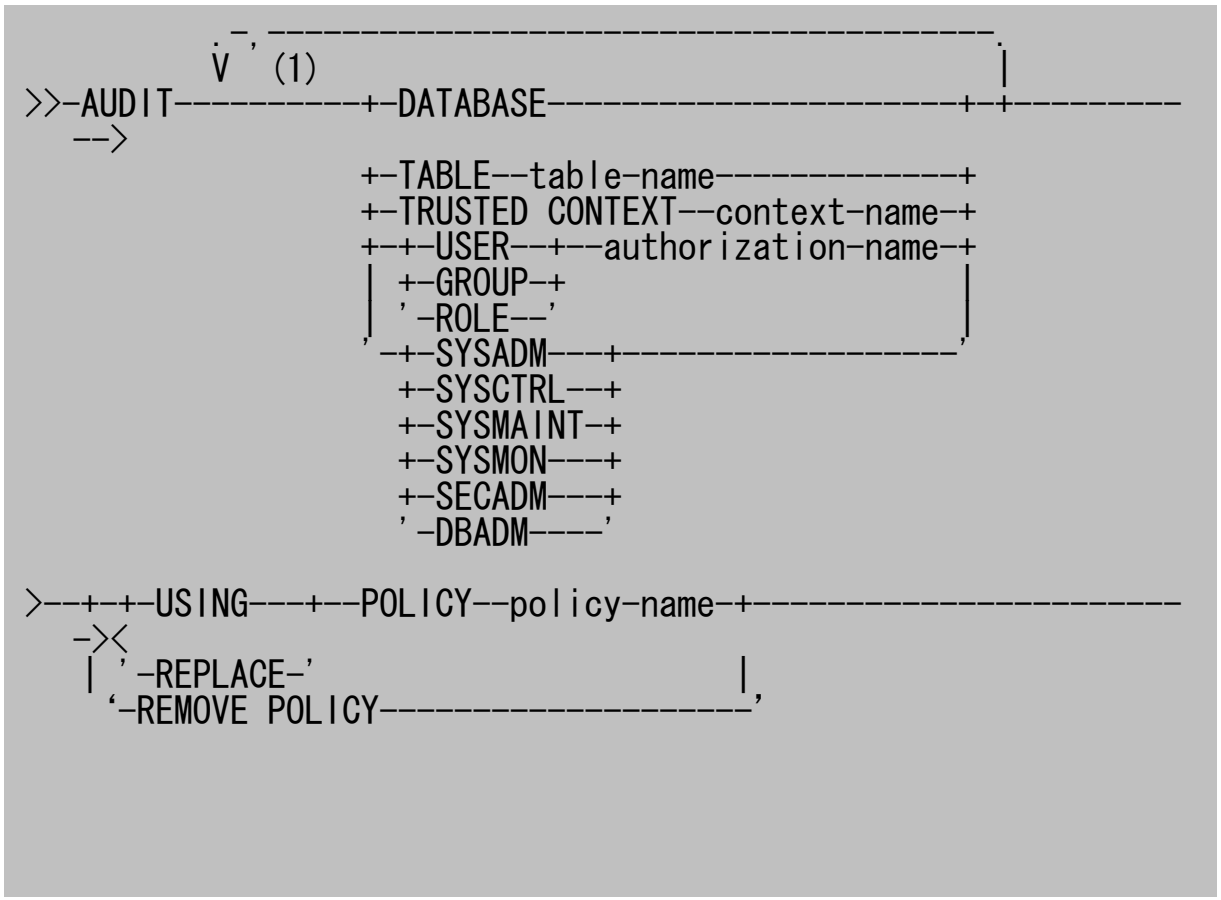
監査対象オブジェクトの決定

- AUDITコマンドで、監査ポリシーを監査対象のオブジェクトにマッピングする。
  - 監査対象のオブジェクトが使用される際に、関連付けられた監査ポリシーに従って監査ログが取得される。
    - 監査対象を絞り、本当に欲しい情報のみを収集することが可能
  - SECADM 権限で実行する必要がある
  - 1つのAUDIT POLICYは、複数のデータベースオブジェクトにマッピングできる
  - 1つのオブジェクトは1つのAUDIT POLICYに所属する
    - Ex. AUDIT ROLE ACCOUNTING, ROLE SUPPORT USING POLICY SENSITIVE_DATA_POL;
  - 複数のAUDIT POLICYが有効になる場合、マージされる
    - 例えば、POLICY1 → group1 、POLICY2 → group2 と定義されており、あるユーザーが2つのグループに属する場合は、論理和をとる



AUDITステートメント　シンタックス

- 作成した監査ポリシーをAUDITステートメントを使用してオブジェクトにマッピングする



AUDITステートメント　監査対象

- DATABASE
  - データベースの中で発生する、監査可能なイベントは全て、関連付けられたポリシーに従って監査が行われる

```
例）AUDIT DATABASE USING POLICY auditdb
```

AUDITステートメント　監査対象

- TABLE
  - 監査対象の表に対する全てのDML、Xqueryによるアクセスが監査される。
  - サポートされる表のタイプ
    - 表
    - Materialized Query Tables (MQT)
    - ニックネーム
  - サポートされない表のタイプ
    - ビュー
    - カタログ表
    - 宣言済み一時表
    - 型付き表
  - EXECUTEカテゴリーのみ監査記録される。
    - EXECUTEカテゴリー以外のポリシーが指定されている場合もEXECUTEカテゴリーのイベントのみ記録される。

```
例) AUDIT TABLE le_employee USING POLICY auditdb
```

AUDITステートメント　監査対象

- Trusted　Context
  - 監査ポリシーに関連付けられたtrusted connection内で発生する監査可能なイベントがロギングされる

```
例）AUDIT TRUSTED CONTEXT appserver USING POLICY admins
```

- ユーザー、グループ、ロール
  - 監査ポリシーに関連付けられたユーザーから実行されたイベントが記録される。
  - 監査ポリシーに関連付けられているグループ、またはロールのメンバーであるユーザーから実行されるイベントがロギングされる。
    - 間接的なメンバーシップについても監査対象となる。


```
例）AUDIT TRUSTED CONTEXT appserver USING POLICY admins
```

AUDITステートメント　監査対象

- 権限
  - 指定された権限を持つユーザーから実行されたイベントがロギングされる
  - サポートされる権限
    - SYSADM
    - DBADM
    - SECADM
    - SYSCTRL
    - SYSMAINT
    - SYSMON

```
例）AUDIT SYSADM, DBADM, SECADM, GROUP DBAS USING POLICY powerusers
```

AUDITステートメント 注意事項

- 変更は、監査ポリシーが適用されるオブジェクトを参照する次の作業単位が適用されるまでは有効にならない。例えば、監査ポリシーがデータベースで使用されている場合、現行の作業単位は COMMIT または ROLLBACK ステートメントが完了するまではポリシーに従って監査を開始しない。
- 監査ポリシーと関連付けられている表にビューがアクセスする場合、そのビューは基礎表のポリシーに従って監査される。
- 監査ポリシーが表に適用される場合、その監査ポリシーはその表に基づくマテリアライズ照会表 (MQT) には適用されない。監査ポリシーを表と関連付ける場合はそのポリシーをその表に基づく MQT にも関連付けること。SQL ステートメントが基本表を参照していても、コンパイラーは MQT を自動的に使用する可能性がある。ただし、基本表で使用されている監査ポリシーは引き続き有効。
- トラステッド・コンテキスト内でユーザーの切り替え操作が実行される際、監査ポリシーはすべて新規ユーザーに応じて再評価され、古いユーザーのポリシーは現行セッションでは使用されない。これは特に、ユーザー、ユーザーのグループまたはロールのメンバーシップ、およびユーザーの権限と直接関連付けられている監査ポリシーに当てはまる。例えば、切り替え前のユーザーが監査対象のロールのメンバーだったために現行セッションが監査されたとすると、切り替え先のユーザーがそのロールのメンバーではない場合、そのポリシーはセッションに適用されなくなる。
- SET SESSION USER ステートメントが実行される際、元のユーザー (およびそのユーザーのグループとロールのメンバーシップと権限) と関連付けられている監査ポリシーは、SET SESSION USER ステートメントで指定されたユーザーと関連付けられているポリシーに結合される。元のユーザーと関連付けられている監査ポリシーは引き続き有効となり、SET SESSION USER ステートメントで指定されたユーザーのポリシーも同じように有効となる。1 つのセッション内で複数の SET SESSION USER ステートメントが発行される場合、元のユーザーおよび現行ユーザーと関連付けられている監査ポリシーだけが考慮される。
- 監査ポリシーが関連付けられているオブジェクトがドロップされると、監査ポリシーとの関連付けはカタログから除去される。その後いつかそのオブジェクトが再作成される場合、そのオブジェクトがドロップされた時点で関連付けられていたポリシーによる監査は、そのオブジェクトに対して実行されない。

インスタンス・レベル監査の構成

- インスタンス・レベルの監査も引き続き使用可能
- インスタンス・レベル監査の構成はdb2audit configureコマンドを利用
  - SYSADMIN権限で実行可能
  - 構成可能なカテゴリー
    - AUDIT
    - CHECKING
    - CONTEXT
    - OBJMAINT
    - SECMAINT
    - SYSADMIN
    - VALIDATE

※EXECUTEカテゴリーは利用できない（データベース・レベル監査のみ）

```
|>- db2audit -+- configure -+- reset -----------------+----------+
                            '-( Audit Configuration )-'          |

Audit Configuration:
>--------+-----------------------------------------------------+->
         |         .-----,-------------------------------------. |
         |         V                                           | |
         +- scope ---+-- all -------+-- status -+- both ----+-+-+
                     +-- audit -----+           +- none ----+
                     +-- checking --+           +- failure -+
                     +-- context  --+           '- success -'
                     +-- objmaint --+
                     +-- secmaint --+
                     +-- sysadmin --+
                     '-- validate --'

>--+--------------------------+---------------------------------->
   '- errortype -+- audit --+--'
                 '- normal -'

>--+--------------------------------+---------------------------->
   '- datapath--<audit-data-path>---'

>--+-----------------------------------------+-------------------->
   '- archivepath--<audit-archive-path>-----'
```

インスタンス・レベル監査の構成

- db2audit configure

① 監査対象のカテゴリ

Scope	
ALL	全てのカテゴリ
AUDIT	Auditの構成変更、Auditログへのアクセスなどのイベント発生時 またはAUDIT_BUF_SZの変更時
CHECKING	DB2オブジェクトまたは機能にアクセス時または処理時の許可検査
OBJMAINT	オブジェクトの作成、除去時
SECMAINT	オブジェクト、データベースの特権またはDBADM権限の付与、取り消し時
SYSADMIN	SYSADM、SYSMAINT、SYSCTRL権限が必要とされる操作の実行時
VALIDATE	ユーザーの認証時、SECURITY情報の検索時
CONTEXT	データベース操作実行時のログ

② 監査対象イベントが成功か失敗か

Status(* Contextの場合区別なし)	
success	正常終了したイベントをログ
failure	失敗したイベントをログ
both	その両方をログ

③ Auditのエラーも含めるか

Errortype	
audit	Audit機能内で起こったエラーを含めて 全てのエラーはDB2によって管理、 エラー時のSQLCODEは呼び出し元に返される
normal	db2auditで生成された全てのエラーは無視。

インスタンス・レベル監査の構成

- audit機能を使用する際には、まず、audit機能でどのようなデータを取得するか決定し、auditの構成を行っておく必要があります。audit機能の構成は、db2audit configureコマンドで行われ、情報はsecurityディレクトリのdb2audit.cfg ファイルに保存されます。
  指定できるパラメータは以下のとおりです。
  - Scope
    - どのタイプの活動をログするか指定します。
    - デフォルトではcontext以外のscopeがすべて選択されています。
    - 1つまたは複数のscopeの組み合わせが可能です。
  - Status
    - 成功、失敗、またはその両方をログすることを指定します。
  - ErrorType
    - Auditの場合は監査機能内部でエラーが生じた場合に負のSQLCODEが返ります。
    - Normalの場合は監査機能内部のエラーは無視されます。
- db2audit configure resetは、設定を初期状態に戻します。
  - 初期状態は SCOPEはCONTEXT以外すべて
    STATUSはfailure（失敗のみログ）
    ERRORTYPEはNORMALです。
- db2audit describeは現在の設定を表示す。

インスタンス・レベル監査の開始

- 監査機能を開始
  - db2audit start
- 監査機能の停止
  - db2audit stop
- DBM構成パラメータAUDIT_BUF_SZにより、監査バッファー・サイズを決定
  - インスタンス共有メモリにアロケートされる。
  - デフォルトは0(監査バッファーなし)
    - 0のときは、監査ログへの書き込みは同期処理
      - パフォーマンスに影響する。
    - 監査バッファーを設定すると、監査ログへの書き込みは非同期処理
      - ログ処理用のプロセスdb2auditdが開始(Unix, Linux)
      - 障害時に監査ログが失われる可能性あり

監査ログファイルのアーカイブ

- 監査ログファイルのアーカイブ
  - 活動中の監査ログをアーカイブ監査ログへと移動する。
  - アーカイブされた監査ログから監査ログの抽出を行う。
  - 監査ログのアーカイブにはdb2audit archive コマンドを使用する。
  - アーカイブ先のパスはデフォルトの監査ログ・ディレクトリまたは、db2audit configure arhivepath で指定された、アーカイブ監査ログパスに作成される。
  - db2 archive コマンド（toオプション）で出力パスを指定することもできる。

```
|>- db2audit -+- archive ---( Audit Archive )--------------------+-><|

Audit Archive:
>--+-----------------------------+-+---------------------------------+->
   '- database -<database name>-' '- node -+-----------------------+
                                           '-<current node number>-'

>--+-----------------------------+-----------------------------------><
   '- to --<audit-archive-path>--'
```

アーカイブ実施手順(データベース・レベル監査)

監査ログのアーカイブ （データベース・レベル監査）

- アーカイブ前の監査ログパス

アクティブ監査ログ

```
[bashii95@yumiko /home/bashii95/audit/log/active]$ ls -l  /home/bashii95/audit/log/active
-rw-------   1 bashii95 staff         11686 Jun 12 15:14 db2audit.db.SECURITY.log.0
[bashii95@yumiko /home/bashii95/audit/log/active]$ ls -l  /home/bashii95/audit/log/archive
```

- db2audit archive database <DB_NAME>の実行

```
[bashii95@yumiko /home/bashii95/audit/log/active]$ db2audit archive database security
Node     AUD      Archived or Interim Log File
         Message
-------- -------- ----------------------------------------------------
       0 AUD0000I db2audit.db.SECURITY.log.0.20080612215157
AUD0000I  Operation succeeded.
```

アーカイブされた監査ログの名
日付が付加される

- アーカイブ後の監査ログパス

アーカイブ監査ログパスに移動される。

```
[bashii95@yumiko /home/bashii95/audit/log/active]$ ls -l  /home/bashii95/audit/log/active

[bashii95@yumiko /home/bashii95/audit/log/active]$ ls -l  /home/bashii95/audit/log/archive
-rw-------   1 bashii95 staff         11686 Jun 12 21:57 db2audit.db.SECURITY.log.0.20080612215157
```

アーカイブ実施手順(インスタンス・レベル監査)

監査ログのアーカイブ (インスタンス・レベル監査)

- アーカイブ前の監査ログパス

アクティブ監査ログ(インスタンスレベル)

```
[bashii95@yumiko /home/bashii95/audit/log/active]$ ls -l  /home/bashii95/audit/log/active
-rw-------   1 bashii95 staff         8643 Jun 12 22:0 db2audit.instance.log.0
[bashii95@yumiko /home/bashii95/audit/log/active]$ ls -l  /home/bashii95/audit/log/archive
```

- db2audit archive の実行

```
[bashii95@yumiko /home/bashii95/audit/log/active]$ db2audit archive

Node     AUD      Archived or Interim Log File
         Message
-------- -------- ----------------------------------------------------
       0 AUD0000I db2audit.instance.log.0.20080612220617
```

アーカイブされた監査ログの名
日付が付加される

- アーカイブ後の監査ログパス

アーカイブ監査ログパスに移動される。

```
[bashii95@yumiko /home/bashii95/audit/log/active]$ ls -l  /home/bashii95/audit/log/active

[bashii95@yumiko /home/bashii95/audit/log/active]$ ls -l  /home/bashii95/audit/log/archive
-rw-------   1 bashii95 staff        10955 Jun 12 22:0 db2audit.instance.log.0.20080612220617
```

監査ログの抽出

- 監査ログの抽出
  - 監査ログのアーカイブにはdb2audit extract コマンドを使用する。
  - アーカイブ監査ログに対して抽出を行う。(アクティブ監査ログに対してextractはできない)
  - データ抽出方式は以下の2つ。
    - TEXT形式
    - DEL形式(delimiterを指定したtext)
      - 表にloadするかexcelで使用
      - DEL形式での出力の場合、区切りファイルの書き込み場所のパスを指定可能

DB2AUDIT EXTRACT シンタックス

- db2audit extractのシンタックス

ファイル形式またはDEL形式を指定

抽出するカテゴリーを指定

Files: アーカイブ監査ログ（1つまたは複数）を指定。

監査ログの抽出(ファイル)

アーカイブ監査ログを指定

監査レコードの抽出例 (ファイル)

```
$db2audit  extract file './audit.file' from files db2audit.db.SAMPLE.log.0.20071119231623
AUD0000I  Operation succeeded.
```

抽出ファイルの中身

EXECUTEカテゴリー

STATEMENT実行

ユーザー情報

```
timestamp=2007-11-19-23.12.58.836143;
  category=EXECUTE;
  audit event=STATEMENT;
  event correlator=5;
  event status=100;
  database=SAMPLE;
  userid=iwahashi;
  authid=IWAHASHI;
  session authid=IWAHASHI;
  origin node=0;
  coordinator node=0;
  application id=9.188.198.118.37372.07111914125;
  application name=db2jcc_application;
  client workstation name=horiken;
  package schema=BASHI195;
  package name=SYSSH200;
  package section=1;
  local transaction id=0x00000000000197c3;
  global transaction
id=0x000000000000000000000000000000000000000000;
  uow id=1;
```

EXECUTEカテゴリーで
STAFF表の監査を取得した例。

次ページに続く

監査ログの抽出(ファイル)

抽出ファイルの中身 (続き)

```
activity id=1;
statement invocation id=0;
statement nesting level=0;
activity type=READ_DML;
statement text=SELECT NAME FROM BASHI195.STAFF WHERE ID = ?;
statement isolation level=CS;
Compilation Environment Description
  isolation: CS
  query optimization: 5
  min dec div 3: NO
  degree: 1
  SQL rules: DB2
  refresh age: +00000000000000.000000
  resolution timestamp: 2007-11-19-23.12.58.000000
  federated asynchrony: 0
  maintained table type: SYSTEM;
rows modified=0;
rows returned=1;
value index = 1
  type = SMALLINT
  data = 10;
```

実施ステートメント

分離レベル

操作対象行

パラメータマーカー値

```
timestamp=2007-11-19-23.12.58.905313;
  category=EXECUTE;
  audit event=COMMIT;
  event correlator=5;
  event status=0;
  database=SAMPLE;
  userid=iwahashi;
  authid=IWAHASHI;
  session authid=IWAHASHI;
  origin node=0;
  coordinator node=0;
  application id=9.188.198.118.37372.07111914125;
  application name=db2jcc_application;
  client workstation name=horiken;
  package schema=NULLID;
  package name=SYSSH200;
  package section=0;
  local transaction id=0x00000000000197c4;
  global transaction
id=0x00000000000000000000000000000000000000000000;
  activity type=OTHER;
```

COMMIT実行

※ 監査ログ解析の詳細については次章を参照のこと

監査ログの抽出(DEL形式)

- カテゴリ毎にDEL形式(CSV)で出力。
  - 監査に含まれるLOBデータがauditlobsファイルに出力される。
    - audit.del
    - checking.del
    - objmaint.del
    - secmaint.del
    - sysadmin.del
    - validate.del
    - context.del
    - execute.del
    - auditlobs
  - デフォルトでは、区切り文字にダブルクオーテーション(")、列区切り文字にカンマ(,)が使用される。
    - delimiter オプションを使用し、監査ログからの抽出時に、デフォルトの監査文字ストリング区切り (二重引用符 ") をオーバーライド可能。 監査レコードが入る表の中にロードするための準備段階で使用する新しい区切り文字を、delimiter の後に指定できる。新しいロード区切り文字として、単一の文字 (例えば !)、または 16 進数表記の 4 文字ストリング (例えば 0xff) が可能。
  - デフォルトでは、アーカイブ監査ログパスに作成される。
    - toオプションにフルパスを指定して、出力先ディレクトリを指定可能

監査ログの抽出(DEL形式)

- 監査レコードの抽出例 (DEL形式)

```
$db2audit extract delasc to 20080612150252 from files db2audit.db.SECURITY.log.0.20080612150252
AUD0000I  Operation succeeded
```

アーカイブ監査ログを指定

DEL形式出力の指定

- 抽出先ディレクトリ
  - カテゴリーごとにファイルが作成される

```
-rw-rw-rw-   1 bashii95 db2adm           0 Jun 13 02:36 audit.del
-rw-rw-rw-   1 bashii95 db2adm          28 Jun 13 02:36 auditlobs
-rw-rw-rw-   1 bashii95 db2adm         801 Jun 13 02:36 checking.del
-rw-rw-rw-   1 bashii95 db2adm         869 Jun 13 02:36 context.del
-rw-rw-rw-   1 bashii95 db2adm         620 Jun 13 02:36 execute.del
-rw-rw-rw-   1 bashii95 db2adm           0 Jun 13 02:36 objmaint.del
-rw-rw-rw-   1 bashii95 db2adm           0 Jun 13 02:36 secmaint.del
-rw-rw-rw-   1 bashii95 db2adm           0 Jun 13 02:36 sysadmin.del
-rw-rw-rw-   1 bashii95 db2adm         371 Jun 13 02:36 validate.del
```

中身はDEL形式

```
"2008-06-12-
15.02.39.006874","CHECKING","CHECKING_OBJECT",3,0,"
SECURITY","iwahashi","IWAHASHI",0,0,"*LOCAL.bashii95.0
80612060210","db2bp","NULLID","SQLC2G13"
0000000000040","0x0000000000002000",,,"
"2008-06-12-
15.02.39.094888","CHECKING","CHECKING_OBJECT",3,-
551,"SECURITY","iwahashi","IWAHASHI",0,0,"*LOCAL.bashi
i95.080612060210","db2bp","NULLID","SQLC2G
000000000000001","0x0000000000000020",,,"
```

監査ログの表へのロード

- DELASC形式のデータを、DB2 の表に格納することができる。
  - 格納手順や表の作成方法は、DB2 Information Center 「DB2 監査データを保持する表の作成 」を参照。
    - http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.sec.doc/doc/t0011542.html

監査ログの表へのロード（実行例）

- 監査データ格納表の作成

セキュリティーの観点から、スキーマを分け、SYSADM,SECADMのみにアクセス権限を与えることをお奨め。

■ 監査テーブル用のスキーマを作成
```
$ db2 "CREATE SCHEMA AUDIT"
DB20000I  The SQL command completed successfully.
$ db2 " SET CURRENT SCHEMA = 'AUDIT'"
DB20000I  The SQL command completed successfully.
```

■ db2audit.ddl スクリプトの実行
```
$db2 -tf $HOME/sqllib/misc/db2audit.ddl
```

- 以下のテーブルが作成される。

Table/View	Schema	Type	Creation time
AUDIT	AUDIT	T	
CHECKING	AUDIT	T	
CONTEXT		AUDIT	T
EXECUTE		AUDIT	T
OBJMAINT	AUDIT	T	
SECMAINT	AUDIT	T	
SYSADMIN	AUDIT	T	
VALIDATE		AUDIT	T

監査ログの表へのロード(実行例)

- 監査データのロード

```
LOAD FROM audit.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE INSERT
INTO AUDIT.AUDIT;
LOAD FROM checking.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
INSERT INTO AUDIT.CHECKING;
LOAD FROM objmaint.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
INSERT INTO AUDIT.OBJMAINT;
LOAD FROM secmaint.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
INSERT INTO AUDIT.SECMAINT;
LOAD FROM sysadmin.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
INSERT INTO AUDIT.SYSADMIN;
LOAD FROM validate.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
INSERT INTO AUDIT.VALIDATE;
LOAD FROM context.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
INSERT INTO AUDIT.CONTEXT;
LOAD FROM execute.del OF DEL MODIFIED BY DELPRIORITYCHAR LOBSINFILE
INSERT INTO AUDIT.EXECUTE;
```

- DELPRIORITYCHARの指定:
  BINARYデータを適切に区切る
- LOBINFILEの指定:インラインLOBは
  32K に制限される。

監査ログの表へのロード（実行例）

- SQLによる解析例 （詳細は後述 監査レポート例参照）
  - 接続認証の失敗を監査するステートメント例

```
SELECT TIMESTAMP,APPNAME, APPID,USERID,EXECID
FROM AUDIT.VALIDATE
WHERE STATUS <0
ORDER BY EXECID,USERID,TIMESTAMP;
```

監査ログのメンテナンス

- 監査バッファーの構成
  - AUDIT_BUF_SZ　（DBM構成パラメータ）
    - 0の場合、同期的に監査ログ書き込みが行われる（デフォルト）
    - 0以上の場合、非同期的に監査ログ書き込みが行われる。

- db2audit flushコマンドの実行
  - 非同期モードでは、監査レコードがディスクに書き込まれる前にバッファーに入れられるので、いくつかのレコードが失われる可能性がある。
  - db2audit flushコマンドによって、データベース・マネージャーは定期的に監査レコードの書き込みを強制可能。
  - 監査機能の許可ユーザーもまた、明示的な要求により監査バッファーをフラッシュすることができる
  - アーカイブ操作中は、バッファーが自動的にフラッシュされる。

新しいストアード・プロシージャおよび表関数

- SYSPROC.AUDIT_ARCHIVE
  - 接続中のデータベースの監査ログ・ファイルをアーカイブする
    - 例：プロシージャーを使用して、すべてのデータベース・パーティションの監査ログをデフォルト・ディレクトリーにアーカイブする
      CALL SYSPROC.AUDIT_ARCHIVE(NULL, NULL)

- SYSPROC.AUDIT_LIST_LOGS　表関数
  - アーカイブされた監査ログをリストする
    - 例：デフォルトの監査アーカイブ・ディレクトリーにあるアーカイブ対象監査ログをすべてリストする
      SELECT * FROM TABLE(SYSPROC.AUDIT_LIST_LOGS('')) AS T1

- SYSPROC.AUDIT_DEL_EXTRACT　ストアード・プロシージャー
  - アーカイブされた監査ログから、DEL形式での抽出を行う

Note: SYSPROC.AUDIT_ARCHIVE

```
>>-AUDIT_ARCHIVE--(--directory--,--dbpartitionnum--)-----------><
```

- directory（オプション）
  - アーカイブ対象監査ファイルが書き込まれるディレクトリーを指定 ： VARCHAR(1024)
  - ディレクトリーがサーバー上に存在しており、インスタンス所有者がそのディレクトリーにファイルを作成できる必要がある
  - 引数が NULL または空ストリングである場合、デフォルト・ディレクトリーを使用
- dbpartitionnum（オプション）
  - 有効なデータベース・パーティション番号を指定 ： INTEGER
  - 現行のデータベース・パーティションには -1、すべてのデータベース・パーティションの集約には NULL または -2 を指定する
- 戻り値

列名	データ・タイプ	説明
DBPARTITIONNUM	INTEGER	アーカイブ対象ファイルのパーティション番号
PATH	VARCHAR(1024)	アーカイブ対象ファイルのディレクトリー位置
FILE	VARCHAR(1024)	アーカイブ対象ファイルの名前
SQLCODE	INTEGER	ファイルのアーカイブ試行中に受信した SQLCODE
SQLSTATE	VARCHAR(5)	ファイルのアーカイブ試行中に受信した SQLSTATE
SQLERRMC	VARCHAR(70) FOR BIT DATA	ファイルのアーカイブ試行中に受信した sqlerrmc

Note: SYSPROC.AUDIT_LIST_LOGS

```
>>-AUDIT_LIST_LOGS--(--directory--)-------------------------><
```

- directory (オプション)
  - アーカイブ対象監査ファイルが書き込まれるディレクトリーを指定 : VARCHAR(1024)
  - ディレクトリーがサーバー上に存在しており、インスタンス所有者がそのディレクトリーにファイルを作成できる必要がある
  - 引数が NULL または空ストリングである場合、検索デフォルト・ディレクトリーを使用

- 戻り値

列名	データ・タイプ	説明
PATH	VARCHAR(1024)	アーカイブ対象ファイルのディレクトリー位置
FILE	VARCHAR(1024)	アーカイブ対象ファイルの名前
SIZE	BIGINT	アーカイブ対象ファイルのファイル・サイズ

Note: SYSPROC.AUDIT_DEL_EXTRACT

```
>>-AUDIT_DELIM_EXTRACT--(--delimiter--,--target_directory--,--source_directory--,-->
>--file_mask--,--event_options--)----------------------------><
```

- delimiter（オプション）
  - 区切り文字付きファイルで使用される区切り文字を指定 ： VARCHAR(1)
  - 引数が NULL または空ストリングである場合、二重引用符が区切り文字として使用される
- target_directory （オプション）
  - 区切り文字付きファイルが保管されるディレクトリーを指定 ： VARCHAR(1024)
  - 引数が NULL または空ストリングである場合、*source_directory* と同じディレクトリーが使用される
- source_directory （オプション）
  - アーカイブ対象監査ログ・ファイルが保管されるディレクトリーを指定 ： VARCHAR(1024)
  - 引数が NULL または空ストリングである場合、監査デフォルトが使用される
- file_mask （オプション）
  - どのファイルを抽出するかについてのマスク ： VARCHAR(1024)
  - 引数が NULL または空ストリングである場合、ソース・ディレクトリーのすべての監査ログ・ファイルから抽出される
- event_options （オプション）
  - どのイベントを抽出するかを定義するストリングを指定 ： VARCHAR(1024)
  - db2audit ユーティリティーの同じストリングと一致する
  - 引数が NULL または空ストリングである場合、すべてのイベントが抽出される

Note：監査に関するカタログ表

- データベース・レベル監査に関する情報が以下のカタログ表に保管される。
  - SYSCAT.AUDITPOLICIES
    - 監査ポリシーに関する情報
  - SYSCAT.AUDITUSE
    - USER、GROUP、または権限（SYSADM、SYSCTRL、SYSMAINT）などの非データベース・オブジェクトと関連付けられている監査ポリシーの情報

- データベースに定義されている監査ポリシーを表示する。（SYSCAT.AUDITPOLICIES）

```
$ db2 -tvf  view_AUDITPOLICIES

select substr(AUDITPOLICYNAME,1,16) as  AUDITPOLICYNAME, AUDITSTATUS, CONTEXTSTATUS, VALIDATESTATUS, CHECKINGSTATUS, SECMAINTSTATUS,
OBJMAINTSTATUS, SYSADMINSTATUS, EXECUTESTATUS, EXECUTEWITHDATA, ERRORTYPE REMARKS from  SYSCAT.AUDITPOLICIES


AUDITPOLICYNAME  AUDITSTATUS CONTEXTSTATUS VALIDATESTATUS CHECKINGSTATUS SECMAINTSTATUS OBJMAINTSTATUS SYSADMINSTATUS EXECUTESTATUS EXECUTEWITHDATA
REMARKS
---------------- ----------- ------------- -------------- -------------- -------------- -------------- -------------- ------------- ---------------
------
ALLPOLICY        B           B             B              B              B              B              B              B             N               A
SENSITIVEDATAPOL N           N             N              N              N              N              N              B             Y               A


  2 record(s) selected.
```

監査対象ステータス
B = 両方
F = 失敗 のみ
N = なし
S = 成功のみ

Note：監査に関するカタログ表

- データベース・レベル監査に関する情報が以下のカタログ表に保管される。
  - SYSCAT.AUDITPOLICIES
    - 監査ポリシーに関する情報
  - SYSCAT.AUDITUSE
    - USER、GROUP、または権限 (SYSADM、SYSCTRL、SYSMAINT) などの非データベース・オブジェクトと関連付けられている監査ポリシーの情報

- 監査対象となっているオブジェクトと監査ポリシーを表示する。 (SYSCAT.AUDITUSE)

```
[bashii95@yumiko /home/bashii95/audit/config]$ db2 -tvf  view_AUDITUSE
select substr(AUDITPOLICYNAME,1,16) as  AUDITPOLICYNAME, AUDITPOLICYID, OBJECTTYPE, SUBOBJECTTYPE, substr(OBJECTSCHEMA,1,16) as
OBJECTSCHEMA, substr(OBJECTNAME,1,16) as OBJECTNAME from SYSCAT.AUDITUSE


AUDITPOLICYNAME  AUDITPOLICYID OBJECTTYPE SUBOBJECTTYPE OBJECTSCHEMA     OBJECTNAME
---------------- ------------ ---------- ------------ ---------------- ----------------
ALLPOLICY                 100                          -                SECURITY
SENSITIVEDATAPOL          101 T                        BASHII95         STAFF


  2 record(s) selected.
```

OBJECT TYPE
S = MQT
T = 表
g = 権限
i = 許可 ID
x = トラステッド・コンテキスト
ブランク = データベース

DB2 LUW 監査レポート例

＜第1.0版　2008年6月＞

お断り：当資料は、DB2 for Linux, UNIX and Windows V9.5 をベースに作成されています。

db2auditによる監査の例

1. 接続の監査
2. 権限のないオブジェクトへのアクセス
3. オブジェクトの作成、削除の監査
4. データアクセスに対する監査
5. 特権、権限の与奪、AuthID変更の監査
6. 管理者権限操作の監査
7. 監査構成の変更

以下のページで監査例を示しています。
監査レコードは主要部分を抜粋しています。

1. 接続の監査

不正な接続要求は発生
していないか？
パスワードハッキング？

- DBに対する接続失敗 （パスワード間違い）
  - 認証操作はVALIDATEカテゴリで監査
    - Validateの他、接続時には次の監査カテゴリーのレコードも生成される
      - CHECKING: DBに対するCONNECT権限の検査
      - EXECUTE: CONNECTイベント
      - CONTEXT: CONNECTイベント
  - 失敗のみ記録すればよいのであれば、AUDIT POLICY 作成時、VALIDATE カテゴリにSTATUS FAILUREを指定することで監査ログ量を減らすことができる。

監査対象の操作例

```
$ db2 connect to secdb user XXXXXX using xxxxxxx
SQL30082N  Security processing failed with reason "24" ("USERNAME AND/OR
PASSWORD INVALID").  SQLSTATE=08001
```

1. 接続の監査

- DBに対する接続失敗 （パスワード間違い）
  - 認証操作はValidateカテゴリで監査

-30082 (SQL30082N セキュリティー処理中にエラーが発生しました) が発生したことが記録される。

不正な接続要求が発生したクライアントのIPアドレスが16進数で表示される。

VALIDATE

	TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	EXECID	NODENUM	COORDNUM	APPID	APPNAME	AUTHTYPE
NG	2008-06-23-14.24.58.761034	VALIDATE	AUTHENTICATION	2	-30082	SECURITY	iwahashi		osuser	0	0	09BCC677.ABDB.080623052500	db2bp	SERVER
NG	2008-06-23-14.25.14.573033	VALIDATE	AUTHENTICATION	2	-30082	SECURITY	iwahashi		osuser	0	0	09BCC677.ABE5.080623052516	db2bp	SERVER
NG	2008-06-23-14.25.21.215353	VALIDATE	AUTHENTICATION	2	-30082	SECURITY	iwahashi		osuser	0	0	09BCC677.ABE9.080623052523	db2bp	SERVER
OK	2008-06-23-14.25.32.852404	VALIDATE	AUTHENTICATION	2	0	SECURITY	iwahashi	IWAHASHI	osuser	0	0	9.188.198.119.44016.08062305240	db2bp	SERVER

CONNECT時指定ユーザー

OSログインユーザー

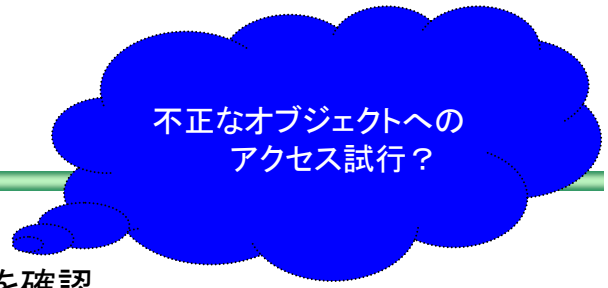
1. 接続の監査

- 接続の失敗に関して、接続を試行したユーザーごとに、その時に使用したID,
  エラーコード、失敗の回数をリストする
  - execid(実行ID):connect実施時に使用(login)していたOSユーザーID
  - userid: connectの際に指定したユーザーID
  - authid: DB2内でユーザー識別のために使用されるID(ユーザーIDにマップされる)
    OS上にユーザーID=osuserでログインし
    DB2 CONNECT TO DB USER kayoko using xxxx(パスワード)
    を指定したらexecidはosuser, useridはkayokoとなる

```
>db2 "select substr(execid,1,10) as "EXECID", substr(userid,1,10) as "USERID", status, count(*) as
   "失敗回数" from audit.validate where event='AUTHENTICATION' and status < 0 group by execid, userid,
   status"

USERID     EXECID           STATUS      失敗回数
---------- ---------- ----------- -----------
db2inst1   admusr          -30082           1
db2inst1   ctrlusr         -30082           1
kayoko     osuser          -30082           1
```

2. 権限のないオブジェクトへのアクセス



- 権限チェックの監査はCHECKINGカテゴリで監査
  - 権限チェックに失敗した場合、STATUS ＜0 となっているところを確認
  - どのようなアクセスを試みたかは、アクセス試行タイプ（ACCESSATT）を確認し、その意味は以下マニュアルを参照。
    - http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.sec.doc/doc/r0005643.html
  - どのような理由で拒否されたかは、アクセス承認理由（ACCESSATT）で数値を確認し、その意味は以下マニュアル参照
    - http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.sec.doc/doc/r0005641.html
  - 失敗のみ記録すればよいのであれば、AUDIT POLICY 作成時、CHECKINGカテゴリにSTATUS FAILUREを指定すると監査ログ量が減る。

監査対象の操作例

```
$ db2 "select * from bashii95.staff"
SQL0551N  "IWAHASHI" does not have the privilege to perform operation "SELECT"
on object "BASHII95.STAFF".  SQLSTATE=42501

$ db2 "select * from bashii95.employee"
SQL0551N  "IWAHASHI" does not have the privilege to perform operation "SELECT"
on object "BASHII95.EMPLOYEE".  SQLSTATE=42501
```

2. 権限のないオブジェクトへのアクセス

- 権限チェックの監査はCHECKINGカテゴリで監査

CHECKING

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	APPID	APPNAME	PKGSCHEMA	PKGNAME	PKGSECNUM	OBJSCHEMA	OBJNAME	OBJTYPE	ACCESSAPP	ACCESSATT
2008-06-23-14.25.32.873171	CHECKING	CHECKING_OBJECT	2	0	SECURITY	iwahashi	IWAHASHI	9.188.198.119.44016.080623052400	db2bp					SECURITY	DATABASE	0x0000000000000020	0x0000000000004000
2008-06-23-14.25.39.232465	CHECKING	CHECKING_OBJECT	3	0	SECURITY	iwahashi	IWAHASHI	9.188.198.119.44016.080623052400	db2bp	NULLID	SQLC2G13	0	NULLID	SQLC2G13	PACKAGE	0x0000000000000040	0x0000000000002000
2008-06-23-14.25.39.292981	CHECKING	CHECKING_OBJECT	3	-551	SECURITY	iwahashi	IWAHASHI	9.188.198.119.44016.080623052400	db2bp	NULLID	SQLC2G13	201	BASHII95	STAFF	TABLE	0x0000000000000001	0x0000000000000020
2008-06-23-14.25.45.113443	CHECKING	CHECKING_OBJECT	4	-551	SECURITY	iwahashi	IWAHASHI	9.188.198.119.44016.080623052400	db2bp	NULLID	SQLC2G13	201	BASHII95	EMPLOYEE	TABLE	0x0000000000000001	0x0000000000000020

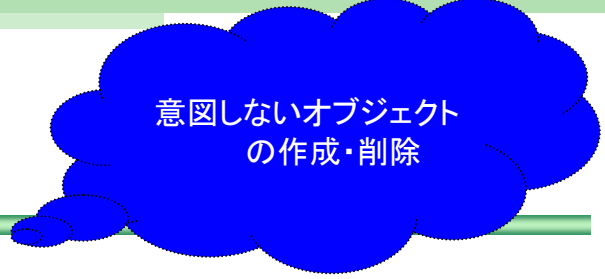
-551 (SQL0551N authorization-ID は、オブジェクト object-name で処理 operation を実行する特権を持っていません。)

誰から？ アクセス権のないユーザーによる表へのアクセス試行があった。

アクセス承認理由
ACCESS DENIED

アクセス試行タイプ
SELECT

3. オブジェクトの作成、削除の監査



- オブジェクトの作成、削除はOBJMAINTで監査
  - 操作の実行(DDLの発行)はEXECUTE(CONTEXT)に記録
  - オブジェクトを作成したことに伴う暗黙の権限のGRANTはSECMAINTに記録
  - 必要な権限チェックはCHECKINGに記録
  - 最終的な実行の有無は、EXECUTE(CONTEXT)中のCOMMITレコードで判断
  - 以上のDDL発行、各種チェック、最終COMMIT/ROLLBACKまでは同一のEvent Correlatorによって関連付けを判断する
  - オブジェクトの変更について、TABLESPACE, CONTAINERなど上位オブジェクトはSYSADMINに記録

監査対象の操作例

```
$ db2 create table test.tab1(c1 int, c2 char(8))
$ db2 drop table test.tab1
```

3. オブジェクトの作成、削除の監査

- オブジェクトの作成、削除はOBJMAINTで監査
  - 実行されたDDLはEXECUTEカテゴリー（次頁参照）

表の作成に伴う暗黙的なスキーマの作成

表の作成

表のドロップ

OBJMAINT

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	NODENUM	COORDNUM	APPID	APPNAME	PKGSCHEMA	PKGNAME	PKGSECNUM	OBJSCHEMA	OBJNAME	OBJTYPE
2008-06-23-19.40.42.949738	OBJMAINT	CREATE_OBJECT	3	0	SECURITY	bashii95	BASHII95	0	0	*LOCAL.bashii95.080623103948	db2bp	NULLID	SQLC2G13	0		TEST	SCHEMA
2008-06-23-19.40.43.773498	OBJMAINT	CREATE_OBJECT	3	0	SECURITY	bashii95	BASHII95	0	0	*LOCAL.bashii95.080623103948	db2bp	NULLID	SQLC2G13	0	TEST	TAB1	TABLE
2008-06-23-19.41.15.209673	OBJMAINT	DROP_OBJECT	4	0	SECURITY	bashii95	BASHII95	0	0	*LOCAL.bashii95.080623103948	db2bp	NULLID	SQLC2G13	0	TEST	TAB1	TABLE

次頁　EXECUTE
カテゴリーへ

3. オブジェクトの作成、削除の監査

- オブジェクトの作成、削除はOBJMAINTで監査
  - 実行されたDDLはEXECUTEカテゴリー、ステートメントイベントを参照

前頁　OBJMAINT
カテゴリーへ

実行ステートメント

EXECUTE

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	SESSNAUTHID	APPID	APPNAME	ACTIVITYTYPE	STMTTEXT
2008-06-23-19.40.43.793568	EXECUTE	STATEMENT	3	0	SECURITY	bashii95	BASHII95	BASHI95	*LOCAL.bashii95.080623103948	db2bp	DDL	create table test.tab1(c1 int, c2 char(8))
2008-06-23-19.40.43.833515	EXECUTE	COMMIT	3	0	SECURITY	bashii95	BASHII95	BASHI95	*LOCAL.bashii95.080623103948	db2bp	OTHER	
2008-06-23-19.41.15.229571	EXECUTE	STATEMENT	4	0	SECURITY	bashii95	BASHII95	BASHI95	*LOCAL.bashii95.080623103948	db2bp	DDL	drop table test.tab1
2008-06-23-19.41.15.277515	EXECUTE	COMMIT	4	0	SECURITY	bashii95	BASHII95	BASHI95	*LOCAL.bashii95.080623103948	db2bp	OTHER	

3. オブジェクトの作成、削除の監査

- DDL発行に関して、AUTHIDごとに、オブジェクトの作成、削除、名前変更について成功、失敗、オブジェクトのスキーマ、名称、タイプをリストする

```
>db2 "select substr(authid,1,10) as authid, event, status, substr(objschema,1,16) as
   objectschema, substr(objname,
1,16) as objectname, objtype from audit.objmaint group by authid, event, status,
   objschema, objname, objtype "

AUTHID     EVENT            STATUS      OBJECTSCHEMA     OBJECTNAME       OBJTYPE
---------- ---------------- ----------- ---------------- ---------------- --------
DBUSR1     CREATE_OBJECT           -552                  DBUSR1           SCHEMA
DBUSR1     CREATE_OBJECT           -552 DBUSR1           TEST             TABLE
DBUSR1     CREATE_OBJECT              0 MONUSR           TEST2            TABLE
DBUSR1     DROP_OBJECT                0 MONUSR           TEST2            TABLE
MONUSR     CREATE_OBJECT              0                  MONUSR           SCHEMA
MONUSR     CREATE_OBJECT              0 MONUSR           SALESFACT        TABLE
MONUSR     CREATE_OBJECT              0 MONUSR           TEST             TABLE
MONUSR     DROP_OBJECT                0 MONUSR           SALESFACT        TABLE
```

4. データアクセスに対する監査

機密テーブルへのアクセス。
データ盗聴？
データ改ざん？

- オブジェクトへのデータアクセスはEXECUTEカテゴリーで取得
  - STATEMENTイベントに注目する
  - パラメータマーカー、ホスト変数の中身を監査する場合、AUDIT POLICY作成時、EXECUTEカテゴリーにWITH DATAオプションを指定する。
  - 他の監査区分とは異なり、EXECUTE 区分の場合は、監査ログが表形式で表示される際に 1 イベントの記述が複数行にわたって示される。
    - 1 行目のレコードは主要なイベントについて記述しており、イベント列にはキーワード STATEMENT が含まれる。
    - 残りの行は、パラメーター・マーカーやホスト変数について、パラメーターごとに 1 行を使用して記述します。これらの行のイベント列には、キーワード DATA が含まれる。
    - 監査ログがレポート形式で表示される際は、レコードは 1 行になるが、「Statement Value」には複数の項目が表示される。DATA キーワードは、表形式の場合にのみ表示。
  - 監査を行いたいテーブルに限定してAUDITすることで、監査ログ量を少なくする。

（例1） JAVA動的SQL プログラムの例

```
・・・・
 con = DriverManager.getConnection(url,
   "bashii95", "bashii95");
 pstmt = con.prepareStatement( “SELECT NAME FROM
   BASHII95.STAFF  WHERE ID >= ? WITH UR");
pstmt.setInt(1,10);
rs = pstmt.executeQuery()
・・・・
```

（例2） 組み込み静的SQL プログラムの例

```
・・・・
char role[8];
・・・・
strcpy (role, argv[3]);
・・・・
 con = DriverManager.getConnection(url,
   "bashii95", "bashii95");
EXEC SQL DECLARE c1 CURSOR FOR
SELECT name, dept FROM BASHII95.staff WHERE
   job=:role
FOR UPDATE OF job;
EXEC SQL OPEN c1;
・・・・
```

4. データアクセスに対する監査

- （例1） JAVA動的SQL
  - オブジェクトへのデータアクセスはEXECUTEカテゴリーで取得
  - STATEMENT　イベントに注目する

JAVA動的SQL
プログラムの例

IPアドレスを特定

パラメータ・マーカー入り
ステートメント

EXECUTE

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	SESSNAUTHID	APPID	APPNAME	PKGSCHEMA	PKGNAME	ACTIVITYTYPE	STMTTEXT	STMTISOLATIONLVL
2008-06-24-03.44.48.454388	EXECUTE	STATEMENT	5	100	SECURITY	bashii95	BASHII95	BASHII95	9.188.198.119.35598.0806231844	db2jcc_application	BASHII95	SYSSH200	READ_DML	SELECT NAME FROM BASHII95.STAFF WHERE ID = ? WITH UR	UR

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	SESSNAUTHID	APPID	APPNAME	PKGSCHEMA	PKGNAME	ACTIVITYTYPE	STMTVALINDEX	STMTVALTYPE		
2008-06-24-03.44.48.454388	EXECUTE	STATEMENT	5	100	SECURITY	bashii95	BASHII95	BASHII95	9.188.198.119.35598.0806231844					SMALLINT	10		

パラメータ・マーカーに指定された値

パラメータ・マーカーの値が複数行にわたって記録される

注）監査レコードの中から一部の列のみ取り出して掲載します。

4. データアクセスに対する監査

- （例2） 組み込み静的SQL
  - プログラムの例オブジェクトへのデータアクセスはEXECUTEカテゴリーで取得
  - STATEMENT イベントに注目する

組み込み静的SQL
プログラムの例

EXECUTE

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	SESSNAUTHID	APPID	APPNAME	PKGSCHEMA	PKGNAME	ACTIVITYTYPE	STMTTEXT	
2008-06-25-19.08.12.782900	EXECUTE	STATEMENT	5	0	SECURITY	bashi95	BASHII95	BASHII95	9.188.198.119.41132.080625100081	fch_audit	BASHII95	FCH_AUDI	READ_DML	DECLARE C1 CURSOR FOR SELECT name, dept FROM BASHII95.staff WHERE job='H00005' FOR UPDATE OF job	
TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	SESSNAUTHID	APPID	APPNAME	PKGSCHEMA	PKGNAME	ACTIVITYTYPE	STMTVALINDEX	STMTVALTYPE
2008-06-25-19.08.12.782900	EXECUTE	STATEMENT	5	0	SECURITY	bashi95	BASHII95	BASHII95	9.188.198.119.41132.080625100081					VARCHAR	Mgr

ホスト変数に代入された値が表示される、

注）監査レコードの中から一部の列のみ取り出して掲載します。

5. 特権、権限の与奪、AUTHID変更の監査

不正な権限与奪？
ユーザー成りすまし？

- （例1）特権、権限の与奪
  - SECMAINTカテゴリーで監査
    - 与奪された権限の種類はPRIVAUTHフィールドを確認。意味は下記を参照
      - http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.admin.sec.doc/doc/r0005678.html
      - 一度に多数の権限がGRANT, REVOKEされたときは、PRIVAUTHの値はAND（論理和）になる
    - EXECUTEカテゴリーに実行コマンドが書かれる
      - CORRELATORで関連付ける。
    - CHECKINGカテゴリーにも権限チェックのレコードが書かれる
- GRANT/REVOKEの例

SECMAINT

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	APPID	OBJSCHEMA	OBJNAME	OBJTYPE	GRANTOR	GRANTEE	GRANTEETYPE	PRIVAUTH	GRANTORTYPE
2008-06-24-17.12.53.666172	SECMAINT	GRANT	5	0	SECURITY	bashii95	BASHII95	*LOCAL.bashii95.080624081113	BASHII95	STAFF	TABLE	BASHII95	IWAHASHI	USER	0x0000000000000200	USER
2008-06-24-17.14.09.326212	SECMAINT	REVOKE	6	0	SECURITY	bashii95	BASHII95	*LOCAL.bashii95.080624081113	BASHII95	STAFF	TABLE	BASHII95	IWAHASHI	USER	0x0000000000000200	USER

STAFF表に対してGRANTEEにSELECT権限を与える

STAFF表に対してGRANTEEからSELECT権限をREVOKE

→次頁EXECUTEに続く

0x0000000000000200 ＝Table SELECT 権限を意味する。

5. 特権、権限の与奪、AUTHID変更の監査

- （例1）特権、権限の与奪
  - SECMAINTカテゴリーで監査
    - EXECUTEカテゴリーにコマンドが書かれる
      - CORRELATORで関連付ける。

- GRANT/REVOKEの例

→前頁SECMAINTから続く

EXECUTE

実行ステートメントはEXECUTEカテゴリに記録

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	SESSNAUTHID	ACTIVITYTYPE	STMTTEXT
2008-06-24-17.12.53.691415	EXECUTE	STATEMENT	5	0	SECURITY	bashii95	BASHII95	BASHII95	DDL	grant select on table bashii95.staff to iwahashi
2008-06-24-17.12.53.739312	EXECUTE	COMMIT	5	0	SECURITY	bashii95	BASHII95	BASHII95	OTHER	
2008-06-24-17.14.09.347978	EXECUTE	STATEMENT	6	0	SECURITY	bashii95	BASHII95	BASHII95	DDL	revoke select on table bashii95.staff from iwahashi
2008-06-24-17.14.09.395883	EXECUTE	COMMIT	6	0	SECURITY	bashii95	BASHII95	BASHII95	OTHER	

5. 特権、権限の与奪、AUTHID変更の監査

- （例2） SET SESSION AUTHORIZATION
  - SECMAINTで監査
    - レコードの書かれる場所は通常の特権、権限の与奪と同じ
  - SET SESSION AUTHORIZATIONの例

set session authorizationによって bashii95の権限で表をアクセス。

```
$ date; db2 "set session authorization bashii95"
Tue Jun 24 18:20:54 JST 2008
DB20000I  The SQL command completed successfully.

$ db2 "select count(*) from bashii95.staff"
```

- 補足
- V9.1 の新機能: SETSESSIONUSER 特権の追加
  - SET SESSION AUTHORIZATION ステートメントを使用してセッション許可 ID を新しい値に変更するには、SQL ステートメントの許可 ID に新しい SETSESSIONUSER 特権があることが必要です。セキュリティー管理者 (新しい SECADM 権限を持つ) は、新しい GRANT SETSESSIONUSER ステートメントを使用してこの特権を付与できます。
    - http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp?topic=/com.ibm.db2.udb.rn.doc/doc/c0023706.htm
  - SET SESSION AUTHORIZATION ステートメント
    - http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.sql.ref.doc/doc/r0020073.html

5. 特権、権限の与奪、AUTHID変更の監査

- （例2） SET SESSION AUTHORIZATION 実行時の監査ログ
  - SECMAINTに” SET_SESSION_USER”イベントが記録される。
  - EXECUTEカテゴリーに実行したステートメントが記録される。
  - そのほかCHECKINGカテゴリーにも記録される。

SEC_MAINTカテゴリーに、
SET_SESSION_USERイベント

SECMAINT

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	APPID	GRANTEE	GRANTEETYPE	PRIVAUTH
2008-06-24-18.20.54.992543	SECMAINT	SET_SESSION_USER	5	0	SECURITY	iwahashi	IWAHASHI	9.188.198.119.61942.08062409173	BASHII95	USER	0x0000000000000000

EXECUTE

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	SESSNAUTHID	APPID	APPNAME	ACTIVITYTYPE	STMTTEXT
2008-06-24-18.20.55.012049	EXECUTE	STATEMENT	5	0	SECURITY	iwahashi	BASHII95	BASHII95	9.188.198.119.61942.08062409173	db2bp	OTHER	set session authorization bashii95

EXECUTEカテゴリーに発行ステートメント

5. 特権、権限の与奪、AUTHID変更の監査

- （例2） SET SESSION AUTHORIZATION後の権限での表アクセス
  - SET SESSION AUTHORIZATIONを実施したユーザーのその後の操作は、AUTHIDは変更したID, USERIDはもとのIDで記録されるため、使用中のAuthIDに関わりなくもとのユーザーを特定することが可能

実際の接続ユーザー

SETSESSION AUTHORIZAION後のユーザー

EXECUTE

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	SESSNAUTHID	APPID	ACTIVITYTYPE	STMTTEXT	STMTISOLATIONLVL	COMPENVDESC	ROWSMODIFIED
2008-06-24-18.21.02.406166	EXECUTE	STATEMENT	9	0	SECURITY	iwahashi	BASHII95	BASHII95	9.188.198.119.61942.0806240 9173	READ_DML	select count(*) from bashii95 .staff	CS	0	1

CHECKING

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	APPID	OBJSCHEMA	OBJNAME	OBJTYPE	ACCESSAPP	ACCESSATT
2008-06-24-18.21.02.261127	CHECKING	CHECKING_OBJECT	6	0	SECURITY	iwahashi	BASHII95	9.188.198.119.61942.08062409173	NULLID	SQLC2G13	PACKAGE	0x0000000000000040	0x0000000000002000
2008-06-24-18.21.02.306171	CHECKING	CHECKING_OBJECT	6	0	SECURITY	iwahashi	BASHII95	9.188.198.119.61942.08062409173	BASHII95	STAFF	TABLE	0x0000000000000010	0x0000000000000020
2008-06-24-18.21.02.326447	CHECKING	CHECKING_OBJECT	7	0	SECURITY	iwahashi	BASHII95	9.188.198.119.61942.08062409173		SYSDEFAULTUSERWORKLOAD	WORKLOAD	0x0000000000000002	0x0000040000000000

SETSESSION AUTHORIZAION後のユーザー権限でチェックされる

V9.5ではWORKLOAD（WLM)に対するUSAGE権限もチェック

5. 特権、権限の与奪、AUTHID変更の監査

- 実行された、特権、権限の与奪、セッション中のAuthID変更を時系列にリスト

```
select
   timestamp,  substr(event,1,16) as event,  substr(userid,1,10) as UserID,
   substr(authid,1,10) as AuthID,
   status,  substr(objtype,1,10) as  ObjectType,  substr(objschema,1,10) as  ObjectSchema,
   substr(objname,1,10) as  ObjectName,
   substr(grantor,1,10) as grantor,  substr(grantee,1,10) as grantee,  privauth
from audit.secmaint
where event in ('GRANT', 'REVOKE', 'SET_SESSION_USER')
order by TIMESTAMP
;
```

```
TIMESTAMP                  EVENT            USERID     AUTHID     STATUS      OBJECTTYPE   OBJECTSCHEMA   OBJECTNAME  GRANTOR    GRANTEE    PRIVAUTH
-------------------------- ---------------- ---------- ---------- ----------- ------------ -------------- ----------- ---------- ---------- --------------------
2008-06-24-17.12.53.666172 GRANT            bashi95    BASHI195             0 TABLE        BASHI195       STAFF       BASHI195   IWAHASHI   0x0000000000000200
2008-06-24-17.14.09.326212 REVOKE           bashi95    BASHI195             0 TABLE        BASHI195       STAFF       BASHI195   IWAHASHI   0x0000000000000200
2008-06-24-18.20.54.992543 SET_SESSION_USER iwahashi   IWAHASHI             0 NONE         -              -           -          BASHI195   0x0000000000000000
2008-06-25-11.27.23.155496 GRANT            bashi95    BASHI195             0 TABLE        BASHI195       EMPLOYEE    BASHI195   PUBLIC     0x0000000000000200
2008-06-25-11.27.43.881203 GRANT            bashi95    BASHI195             0 TABLE        BASHI195       EMPLOYEE    BASHI195   DB2SEC     0x0000000000000200
2008-06-25-11.28.10.970974 REVOKE           bashi95    BASHI195             0 TABLE        BASHI195       EMPLOYEE    BASHI195   DB2SEC     0x0000000000000200
2008-06-25-11.28.20.934233 REVOKE           bashi95    BASHI195          -567 TABLE        BASHI195       EMPLOYEE    BASHI195   PUBLIC     0x0000000000000200
2008-06-25-11.28.39.730331 REVOKE           bashi95    BASHI195             0 TABLE        BASHI195       EMPLOYEE    BASHI195   PUBLIC     0x0000000000000200
2008-06-25-11.30.04.156214 GRANT            bashi95    BASHI195             0 DATABASE     -              SECURITY    BASHI195   IWAHASHI   0x0004000000000000
2008-06-25-11.30.24.623158 REVOKE           bashi95    BASHI195             0 DATABASE     -              SECURITY    BASHI195   IWAHASHI   0x0004000000000000

  10 record(s) selected.
```

6. 管理者権限操作の監査

管理者権限、パワーユーザーで不正行為？

- 管理者権限で実行された操作はSYSADMINカテゴリーで監査できる。
  - SYSADM、SYSMAINT、SYSCTRL権限が必要とされる操作の実行時

- DBレベルでの監査

SYSADMIN

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	APPID
2008-06-25-13.13.59.317931	SYSADMIN	PRUNE_RECOVERY_HISTORY	3	0	SECURITY	bashii95	BASHII95	*LOCAL.bashii95.080625041353
2008-06-25-13.13.59.337998	SYSADMIN	BACKUP_DB	3	0	SECURITY	bashii95	BASHII95	*LOCAL.bashii95.080625041353
2008-06-25-13.43.17.091360	SYSADMIN	LOAD_TABLE	31	0	SECURITY	bashii95	BASHII95	*LOCAL.bashii95.080625044016
2008-06-25-13.43.33.179426	SYSADMIN	LOAD_TABLE	49	0	SECURITY	bashii95	BASHII95	*LOCAL.bashii95.080625044016
2008-06-25-13.45.16.329771	SYSADMIN	CREATE_TABLESPACE	57	0	SECURITY	bashii95	BASHII95	*LOCAL.bashii95.080625044016
2008-06-25-13.45.27.642875	SYSADMIN	DROP_TABLESPACE	58	0	SECURITY	bashii95	BASHII95	*LOCAL.bashii95.080625044016

→次頁EXECUTEに続く

6. 管理者権限操作の監査

- 管理者権限で実行された操作はSYSADMINカテゴリーにて監査できる。
  - 実際のステートメントはEXECUTEカテゴリーで一部確認できる

- DBレベルでの監査

EXECUTE

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	SESSNAUTHID	APPID	ACTIVITYTYPE	STMTTEXT
2008-06-25-13.43.17.111358	EXECUTE	STATEMENT	31	0	SECURITY	bashii95	BASHII95	BASHII95	*LOCAL.bashii95.0806250440 16	LOAD	
2008-06-25-13.43.33.199534	EXECUTE	STATEMENT	49	0	SECURITY	bashii95	BASHII95	BASHII95	*LOCAL.bashii95.0806250440 16	LOAD	
2008-06-25-13.45.16.413787	EXECUTE	STATEMENT	57	0	SECURITY	bashii95	BASHII95	BASHII95	*LOCAL.bashii95.0806250440 16	DDL	create tablespace ts1 managed by automatic storage
2008-06-25-13.45.27.718980	EXECUTE	STATEMENT	58	0	SECURITY	bashii95	BASHII95	BASHII95	*LOCAL.bashii95.0806250440 16	DDL	drop tablespace ts1

6. 管理者権限操作の監査（インスタンスレベル）

- 管理者権限で実行された操作はSYSADMINカテゴリーにて監査できる。
  - SYSADM、SYSMAINT、SYSCTRL権限が必要とされる操作の実行時
  - インスタンスレベルの監査に記録されるイベントもある。

- インスタンスレベルでの監査

SYSADMIN

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	NODENUM	COORDNUM	APPID	APPNAME
2008-06-25-13.13.00.413549	SYSADMIN	FORCE_APPLICATION	2	0		bashii95	BASHII95	0	0	*LOCAL.bashii95.080625041300	db2bp
2008-06-25-13.14.14.761762	SYSADMIN	ACTIVATE_DB	2	0	SECURITY	bashii95	BASHII95	0	0	*LOCAL.bashii95.080625041413	db2bp
2008-06-25-13.14.22.543826	SYSADMIN	DEACTIVATE_DB	2	0	SECURITY	bashii95	BASHII95	0	0	*LOCAL.bashii95.080625041422	db2bp
2008-06-25-13.16.26.494496	SYSADMIN	UPDATE_DB_CFG	6	-5153		bashii95	BASHII95	0	0	*LOCAL.bashii95.080625041435	db2bp
2008-06-25-13.16.46.280698	SYSADMIN	UPDATE_DB_CFG	8	0		bashii95	BASHII95	0	0	*LOCAL.bashii95.080625041435	db2bp

FORCE APPLICATION, ACTIVATE,DEACTIVATE, UPDATE DB CFGなどはインスタンスレベルでの監査に記録される

7. 監査構成変更の監査

監査自体を不正に停止？
監査構成の変更？

- 監査構成自体を変更しようとした場合に、どのような監査レコードが記録されるか。
  - AUDITカテゴリーにて監査する

■　現在のAUDIT設定を確認する

```
$ db2 -tvf auditchk.txt
select substr(AUDITPOLICYNAME,1,16) as AUDITPOLICYNAME, AUDITPOLICYID, OBJECTTYPE, SUBOBJECTTYPE,
    substr(OBJECTSCHEMA,1,16) as OBJECTSCHEMA, substr(OBJECTNAME,1,16) as OBJECTNAME from
    syscat.AUDITUSE

AUDITPOLICYNAME  AUDITPOLICYID OBJECTTYPE SUBOBJECTTYPE OBJECTSCHEMA     OBJECTNAME
---------------- ------------ ---------- ------------- ---------------- ----------------
ALLPOLICY                 100                          -                SECURITY
SENSITIVEDATAPOL          101 T                        BASHI195         STAFF
```

POLICYを REMOVE し、監査を停止

■　STAFF表の監査を停止する。

```
$ date; db2 "audit table bashii95.staff remove policy"
Wed Jun 25 15:26:16 JST 2008

$ db2 -tvf auditchk.txt
select substr(AUDITPOLICYNAME,1,16) as AUDITPOLICYNAME, AUDITPOLICYID, OBJECTTYPE, SUBOBJECTTYPE,
    substr(OBJECTSCHEMA,1,16) as OBJECTSCHEMA, substr(OBJECTNAME,1,16) as OBJECTNAME from
    syscat.AUDITUSE

AUDITPOLICYNAME  AUDITPOLICYID OBJECTTYPE SUBOBJECTTYPE OBJECTSCHEMA     OBJECTNAME
---------------- ------------ ---------- ------------- ---------------- ----------------
ALLPOLICY                 100                          -                SECURITY
```

7. 監査構成変更の監査

- 監査構成自体を変更しようとした場合に、どのような監査レコードが記録されるか。
  - AUDITカテゴリーにて監査する
  - 発行されたステートメントはEXECUTEカテゴリーで確認できる

AUDITのREMOVEが行われた

AUDIT

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	USERID	AUTHID	DATABASE	APPID	APPNAME	POLNAME	POLASSOCOBJTYPE	POLASSOCSUBOBJTYPE	POLASSOCNAME	POLASSOCOBJSCHEMA
2008-06-25-15.26.16.236786	AUDIT	AUDIT_REMOVE	40	0	db2sec	DB2SEC	SECURITY	*LOCAL.bashii.080625060827	db2bp	SENSITIVEDATAPOLICY	T		STAFF	BASHII95

EXECUTE

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATABASE	USERID	AUTHID	SESSNAUTHID	APPID	APPNAME	PKGSCHEMA	ACTIVITYTYPE	STMTTEXT
2008-06-25-15.26.16.261084	EXECUTE	STATEMENT	40	0	SECURITY	db2sec	DB2SEC	DB2SEC	*LOCAL.bashii.080625060827	db2bp	BASHII95	DDL	audit table bashii95.staff remove policy

発行されたコマンド

特記事項

- この資料に含まれる情報は可能な限り正確を期しておりますが、IBMの正式なレビューを受けておらず、当資料に記載された内容に関してIBMは何ら保証するものではありません。この情報の使用、評価、実施は使用者の責任で使用者の環境に合わせて行ってください。
- 当資料の他社情報は一般公開されている資料を参照し、一般的な視点から論じたものであり、IBMは内容および実際の稼動を保証しません。
- この情報には、技術的に不適切な記述や誤植を含む場合があります。IBM は予告なしに、随時、この文書に記載されている内容に対して、改良または変更を行うことがあります。
- 当資料をコピー等で複製することは、IBMおよび執筆者の承諾なしではできません。
- 当資料に記載された製品名または会社名はそれぞれの各社の商標または登録商標です。