



PGECons
PostgreSQL Enterprise Consortium

確実な移行を目指して - DB移行試験手法の調査 -

2014年度活動成果報告

PostgreSQLエンタープライズ・コンソーシアム
WG2 (移行WG)

アジェンダ

- PGECcons WG2
- 2012, 2013年度の活動内容
- 2014年度の活動
- 2014年度の成果
- おわりに

PGECons WG2

- 活動内容
- PostgreSQLへの移行メリット

WG2(移行WG)活動内容

活動テーマ: 異種DBMSからPostgreSQLへの移行

課題認識

- ・ 異種DBMSシステムをPostgreSQLへ移行するプロセスが確立していないことが、普及を妨げる大きな障壁と認識
 - ・ 移行作業をどのように進めればよいか分からない。
 - ・ 初期段階で移行に必要なトータルコストを算出できない。
 - ・ 過去の経験則や点在するノウハウに依存しているのが現状



活動目標

- ・ 異種DBMSからPostgreSQLへの移行を検討する際のガイドラインを提示する。(難易度判断、留意すべき事項、移行手順)

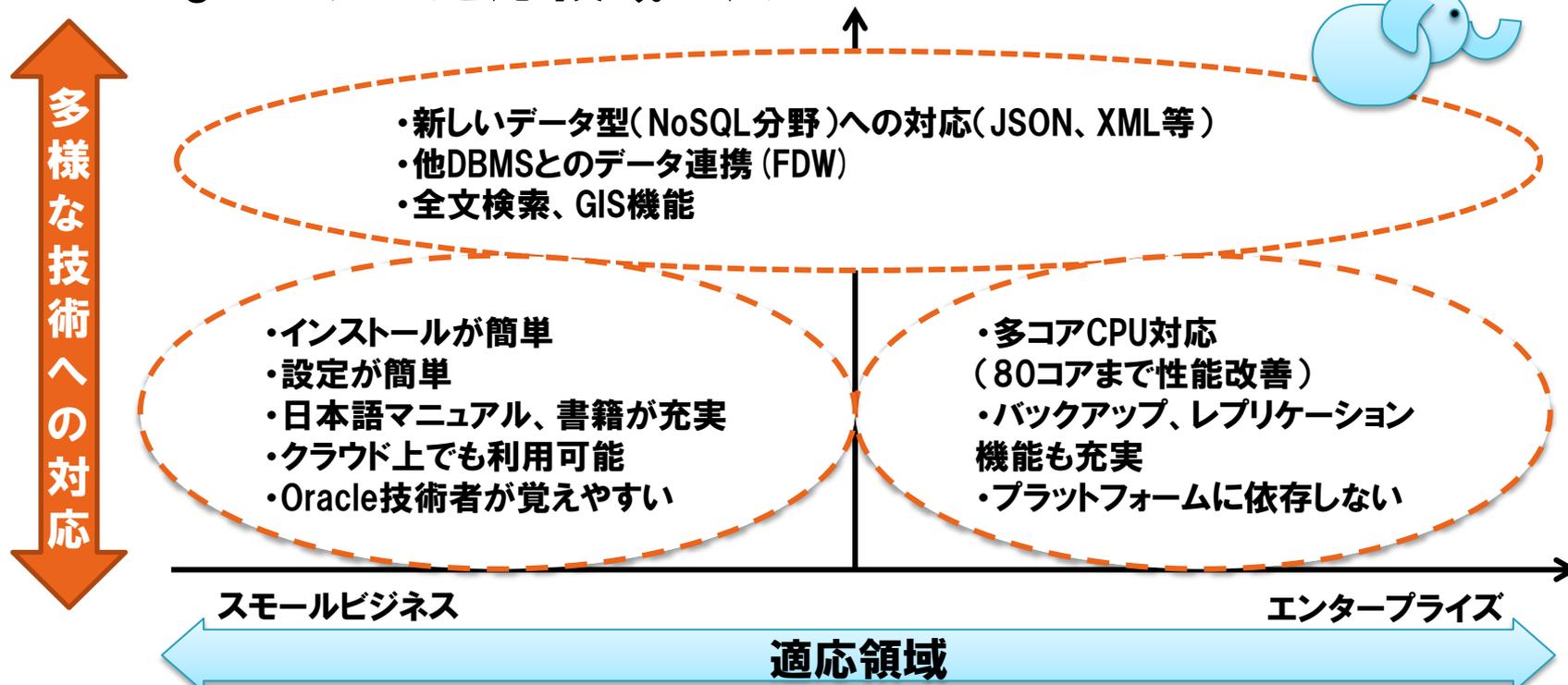


成果物

- ・ 「異種DBMSからPostgreSQLへの移行ガイド」を作成

商用DBからPostgreSQLに移行するメリット

- ベンダ・ロックインの回避
- 運用コストの削減
- PostgreSQLの適応領域が広い



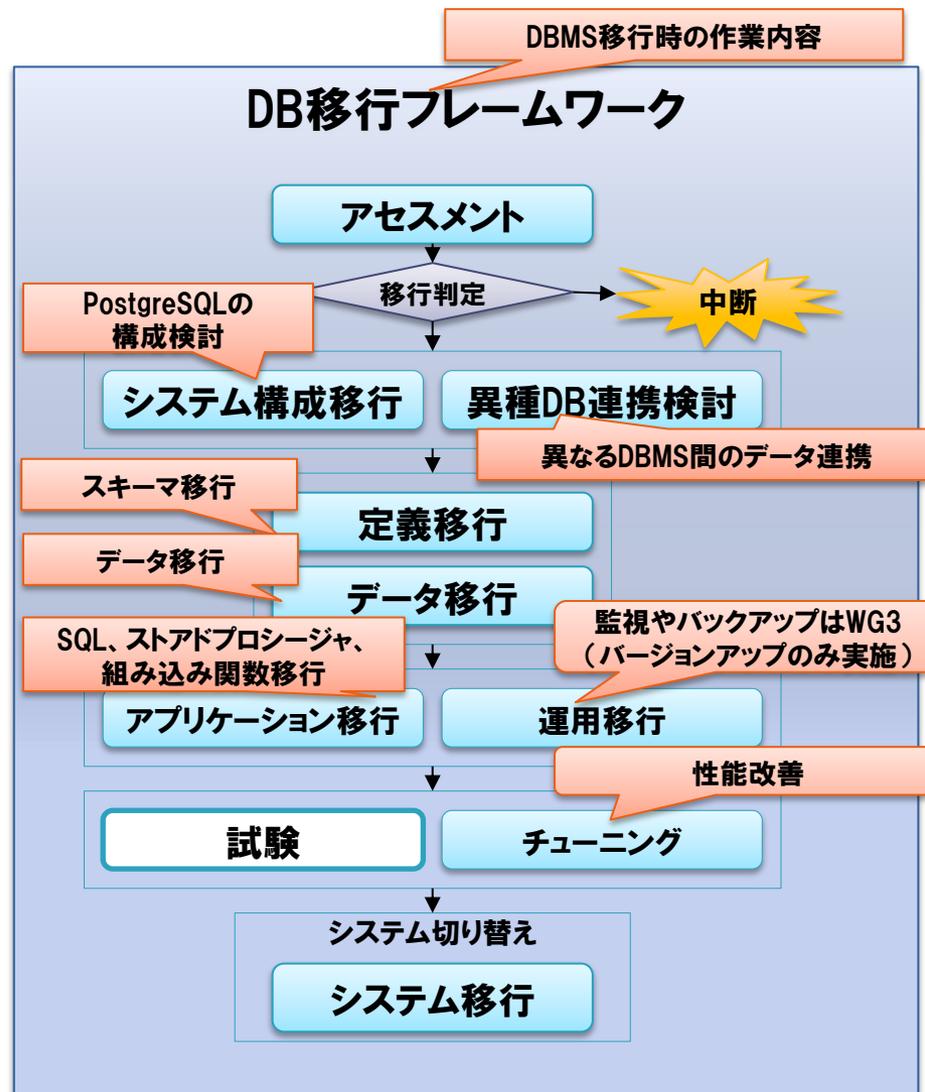
適応領域でデータベースを使い分けず、データベースを
PostgreSQLに統一することが可能

2012, 2013年度の成果

- 「異種DBMSからPostgreSQLへの移行ガイド」の文章構成
- 成果物一部紹介

「異種DBMSからPostgreSQLへの移行ガイド」の文章構成

- 移行作業の全体像を解説
 - DB移行フレームワーク編 (21ページ)
- 移行作業に含まれる作業内容、手順の調査
 - システム構成調査編 (29ページ)
 - 異種DB間連携調査編 (18ページ)
 - スキーマ移行調査編 (25ページ+別表)
 - データ移行・文字コード変換編 (49ページ)
 - ストアドプロシージャ移行調査編 (34ページ)
 - アプリケーション移行調査編 (10ページ)
 - SQL移行調査編 (18ページ+別表)
 - 組み込み関数移行調査編 (15ページ+別表)
 - チューニング編 (30ページ+別表)
 - バージョンアップ編 (39ページ+別表7)
- 移行作業を試行する検証
 - データ移行調査および実践編 (60ページ+別表3)
 - アプリケーション移行実践編 (25ページ+別表)



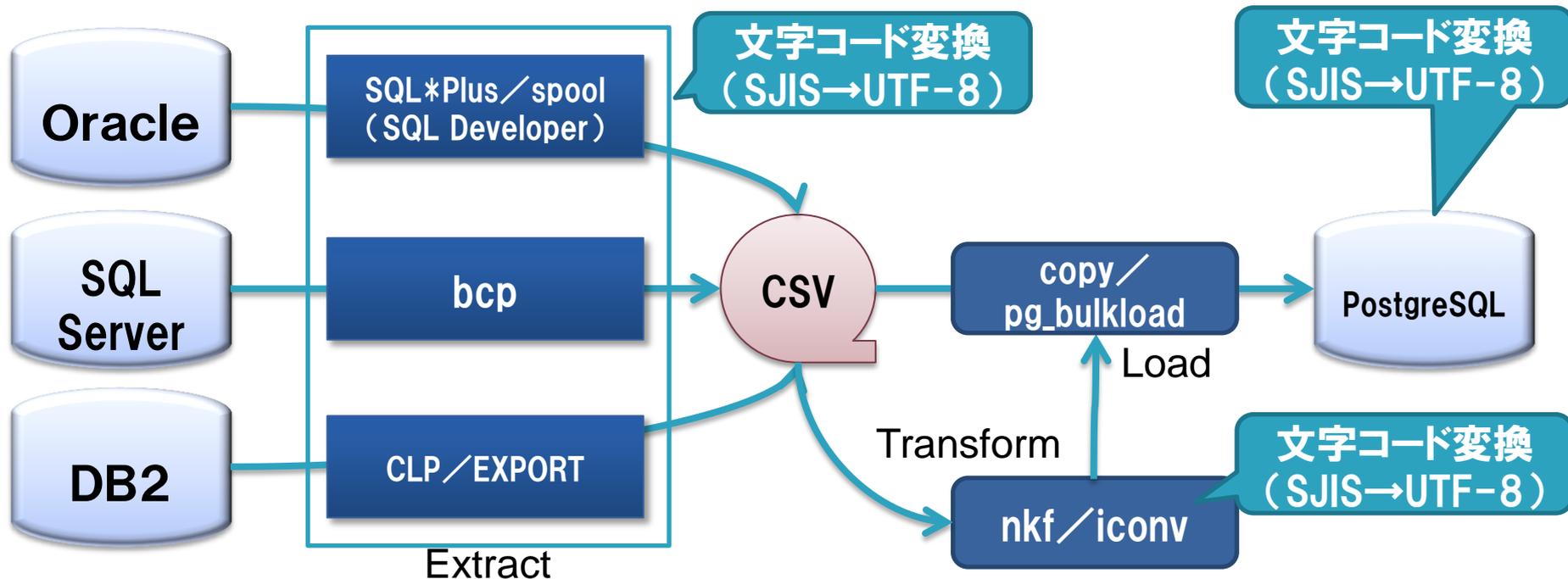
「異種DBMSからPostgreSQLへの移行ガイド」の各文章

| No. | 作成文書（作成年度） | 記載概要 |
|-----|---|--|
| 1 | DB移行フレームワーク編（2012年） | 移行作業全体の解説 |
| 2 | システム構成調査編（2012年） | DBMSの一般的なシステム構成とPostgreSQLの構成 |
| 3 | 異種DB間連携調査編（2012年） | 異種DBMSとPostgreSQLのデータ連携方法 |
| 4 | スキーマ移行調査編（2012年） | 異種DBMSとPostgreSQLのスキーマの違いと書き換え方針 |
| 5 | SQL移行調査編（2012年） | 異種DBMSとPostgreSQLのSQL互換性調査結果と書き換え方針 |
| 6 | ストアドプロシージャ移行調査編（2012、2013年） | 異種DBMSのストアドプロシージャの互換性調査結果と書き換え方針 |
| 7 | 組み込み関数移行調査編（2012年） | Oracle, PostgreSQLの組み込み関数互換性調査結果と書き換え方針 |
| 8 | チューニング編（2013年） | PostgreSQLのチューニング手法を記載 |
| 9 | バージョンアップ編（2013年） | PostgreSQLのバージョンアップ手法と検証結果 |
| 10 | データ移行調査および実践編（2012年） データ移行・文字コード変換編（2013年） | 異種DBMSからPostgreSQLへのデータ移行時の注意点と実施結果 |
| 11 | アプリケーション移行調査および実践編（2012年） | APIやトランザクションの差異と対処方法。および異種DBMSからPostgreSQLに移行した際のアプリケーションの移行結果 |

■ 赤字部分を本日紹介

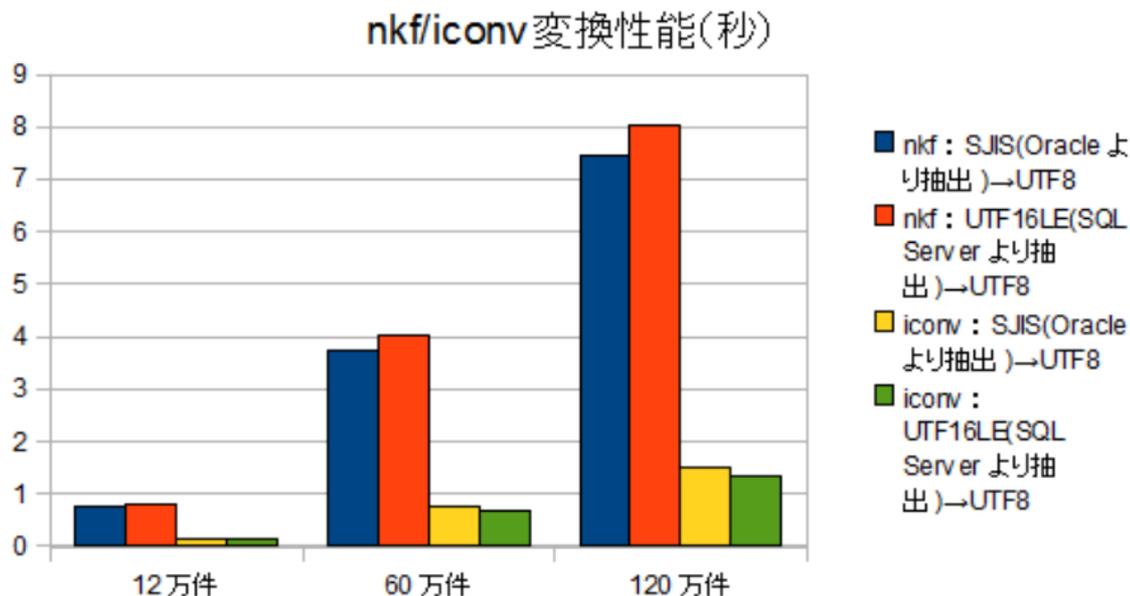
データ移行・文字コード変換編(1/2)

- 郵便番号のデータを利用(約12万件)
- Oracle、SQL Server、DB2からデータを抽出しPostgreSQLに投入
- 各DBMSのデータをファイル(CSV)に抽出
- コード変換のタイミング別検証
(データ抽出時／出力ファイル／データ投入時)



データ移行・文字コード変換編(2/2)

| 検証項目 | 検証結果 |
|--------------|--|
| データの抽出／投入 | <ul style="list-style-type: none">DBMS、ツール、文字コード変換のタイミングに関わらず、データ移行に成功出力時にデータを””でデータを囲うなど、ツールにより出力形式が異なる結果となった → 今回は、形式の違いがデータ投入の成否に影響しなかったデータ抽出時の文字コードはDBMSによる差を確認<ul style="list-style-type: none">Oracle(SQL*Plus)、DB2(CLP/EXPORT)はUTF-8出力SQL Server(bcp)はUTF-16出力 |
| ファイルの文字コード変換 | nkfとiconvには大きな変換性能差 |



アプリケーション移行実践編（1/2）

■ 移行対象アプリケーション

□ Commander4J

- バーコードラベルを作成するJavaのOSSアプリケーション
- 対応DB: Oracle, SQL Server, MySQL
- ストアドプロシージャ: なし

■ SQL変更箇所の抽出に db_syntax_diff を使用

- OracleからPostgreSQLへアプリケーションを移行する際に修正が必要となる箇所を報告するツール
- NTT OSSセンターで開発されたOSS(<https://github.com/db-syntax-diff>)
- アプリケーションのソースコードを指定して実行すると、修正が必要なSQLのパターンとマッチングして報告

| | | | | |
|---|-----|--------|-----------|--|
| /home/postgres/b4/src/com/commander4j/db/JDBTable.java | 83 | 7 SQL | WARNING | user_idのデータ・ディクショナリ・ビューまたは動的パフォーマンス・ビュー |
| /home/postgres/b4/src/com/commander4j/db/JDBTable.java | 83 | 15 SQL | WARNING | user_commentのデータ・ディクショナリ・ビューまたは動的パフォーマンス |
| /home/postgres/b4/src/com/commander4j/db/JDBDespatch.java | 428 | 7 SQL | LOW2 | FROMの省略は未サポートです。 |
| /home/postgres/b4/src/com/commander4j/db/JDBQuery.java | 67 | 13 SQL | ERROR LV1 | 全列指定は未レポートです。SELECT句にすべてのカラム名を指定して |
| /home/postgres/b4/src/com/commander4j/db/JDBQuery.java | 77 | 23 SQL | LOW1 | ROWNUM擬似列は未サポートです。 |

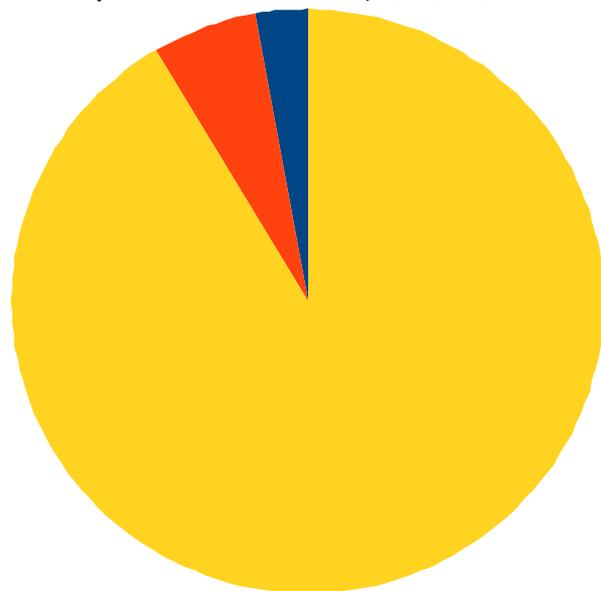
アプリケーション移行実践編（2/2）

■ 移行実施結果

□ 試験工程が全工程の90%以上を占めた

- ツールでSQL修正箇所を自動検出し、修正作業が短く済んだ。
- 修正有無に関わらず、関連するSQLはすべて試験を実施した。
- db_syntax_diff で検出できなかった箇所は試験工程で対応した。

各フェーズの所要時間比率



| 項番 | 作業項目 | 所要時間比率 |
|----|----------------------------|--------|
| #1 | AP修正(PostgreSQLへの接続) | 2.8% |
| #2 | AP修正(db_syntax_diffに基づく修正) | 5.8% |
| #3 | 試験およびエラー修正 | 91.4% |

- 移行作業項番#1
- 移行作業項番#2
- 移行作業項番#3

2014年度の活動

- 体制
- 活動内容

2014年度活動体制

- 本年度のWG2は下記体制で活動しました。

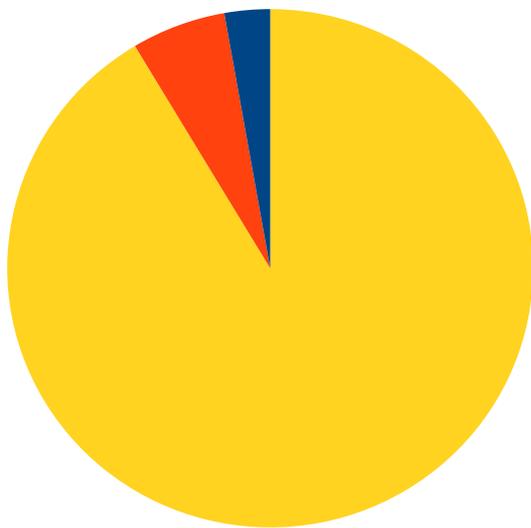
| 企業名 | 備考 |
|------------------------|-----|
| 株式会社富士通ソーシャルサイエンスラボラトリ | 主査 |
| NECソリューションイノベータ株式会社 | 副主査 |
| 日本電信電話株式会社 | - |
| 富士通株式会社 | - |



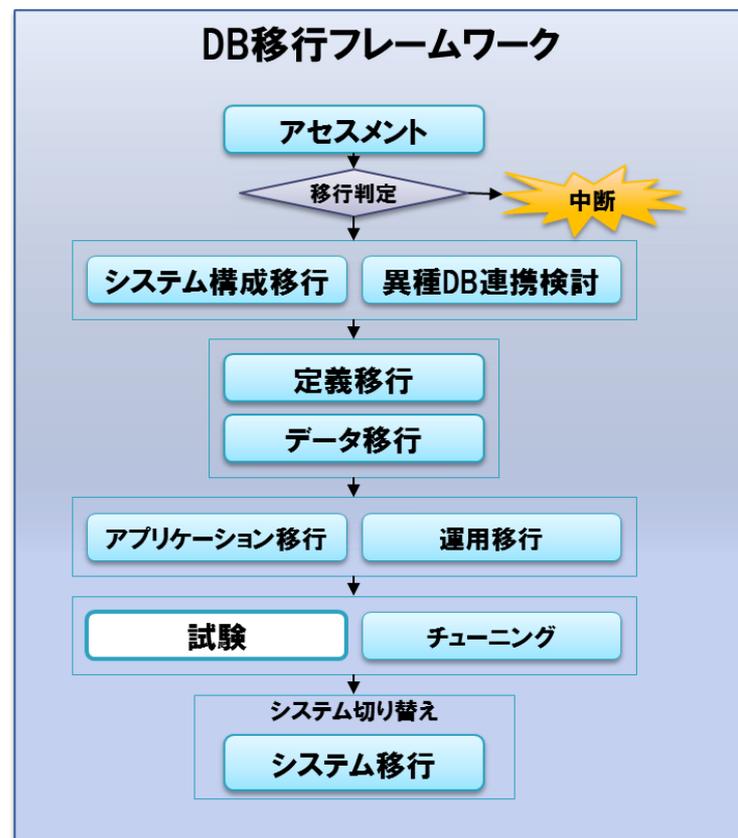
2014年度活動内容(1/2)

- これまでの移行検証における試験作業の割合が高かったことに加え、移行フレームワークにおいて試験手法を文書として公開できていなかったため、「試験編」の作成が重要な課題であると認識しました。

各フェーズの所要時間比率



- 試験およびエラー修正
- AP修正(db_syntax_diffに基づく修正)
- AP修正(PostgreSQLへの接続)



2014年度活動内容(2/2)

- 移行作業の全体像を解説
 - DB移行フレームワーク編 (21ページ)
- 移行作業に含まれる作業内容、手順の調査
 - システム構成調査編 (29ページ)
 - 異種DB間連携調査編 (18ページ)
 - スキーマ移行調査編 (25ページ+別表)
 - データ移行・文字コード変換編 (49ページ)
 - ストアドプロシージャ移行調査編 (34ページ)
 - アプリケーション移行調査編 (10ページ)
 - SQL移行調査編 (18ページ+別表)
 - 組み込み関数移行調査編 (15ページ+別表)
 - チューニング編 (30ページ+別表)
 - バージョンアップ編 (39ページ+別表7)
 - **試験編 <NEW! >**
- 移行作業を試行する検証
 - データ移行調査および実践編 (60ページ+別表3)
 - アプリケーション移行実践編 (25ページ+別表)



移行プロセス全体像

2014年度の成果

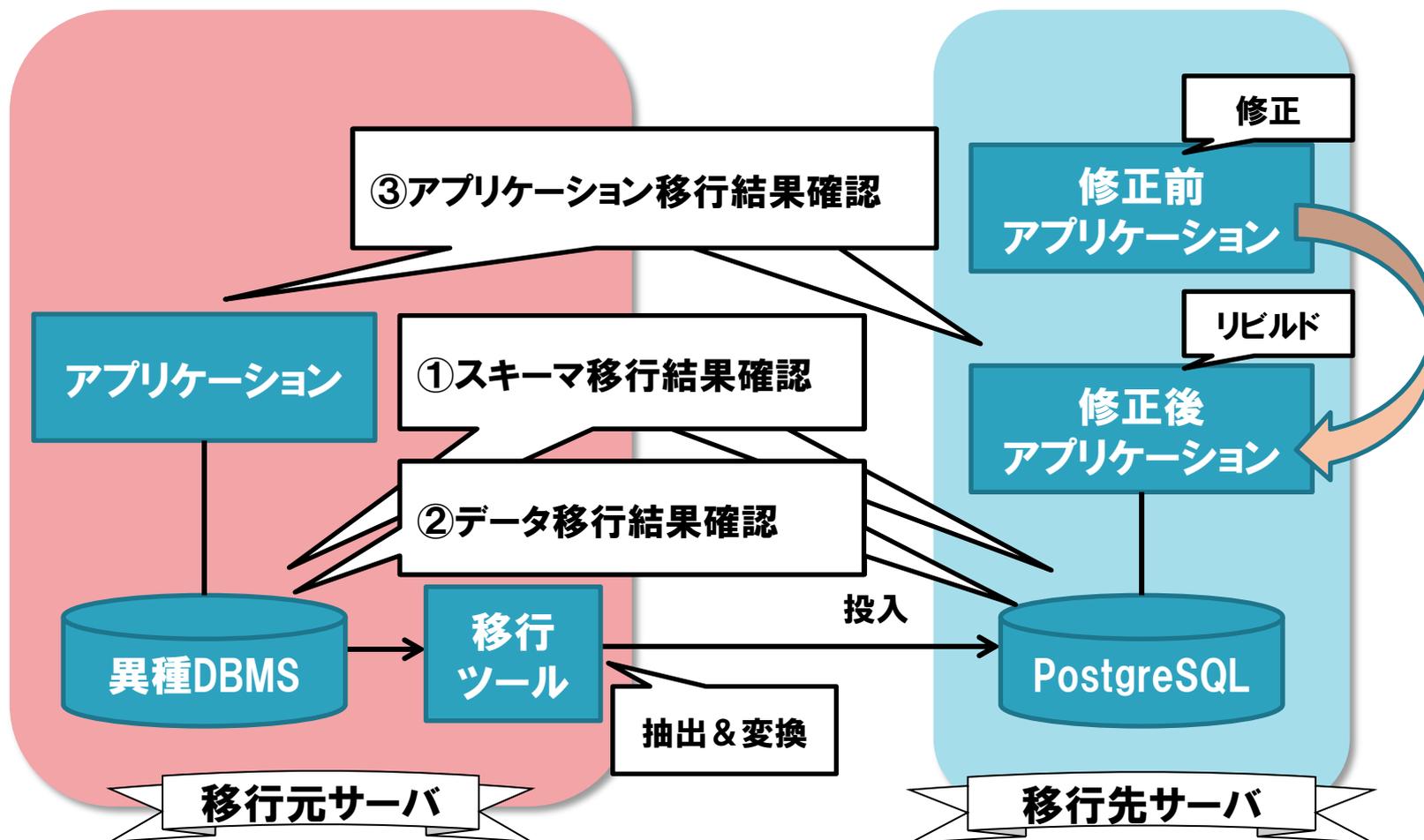
2014年度活動成果物

- 2014年度の成果物はPGECconsのサイトにて公開しております。
 - https://www.pgecons.org/download/works_2014/



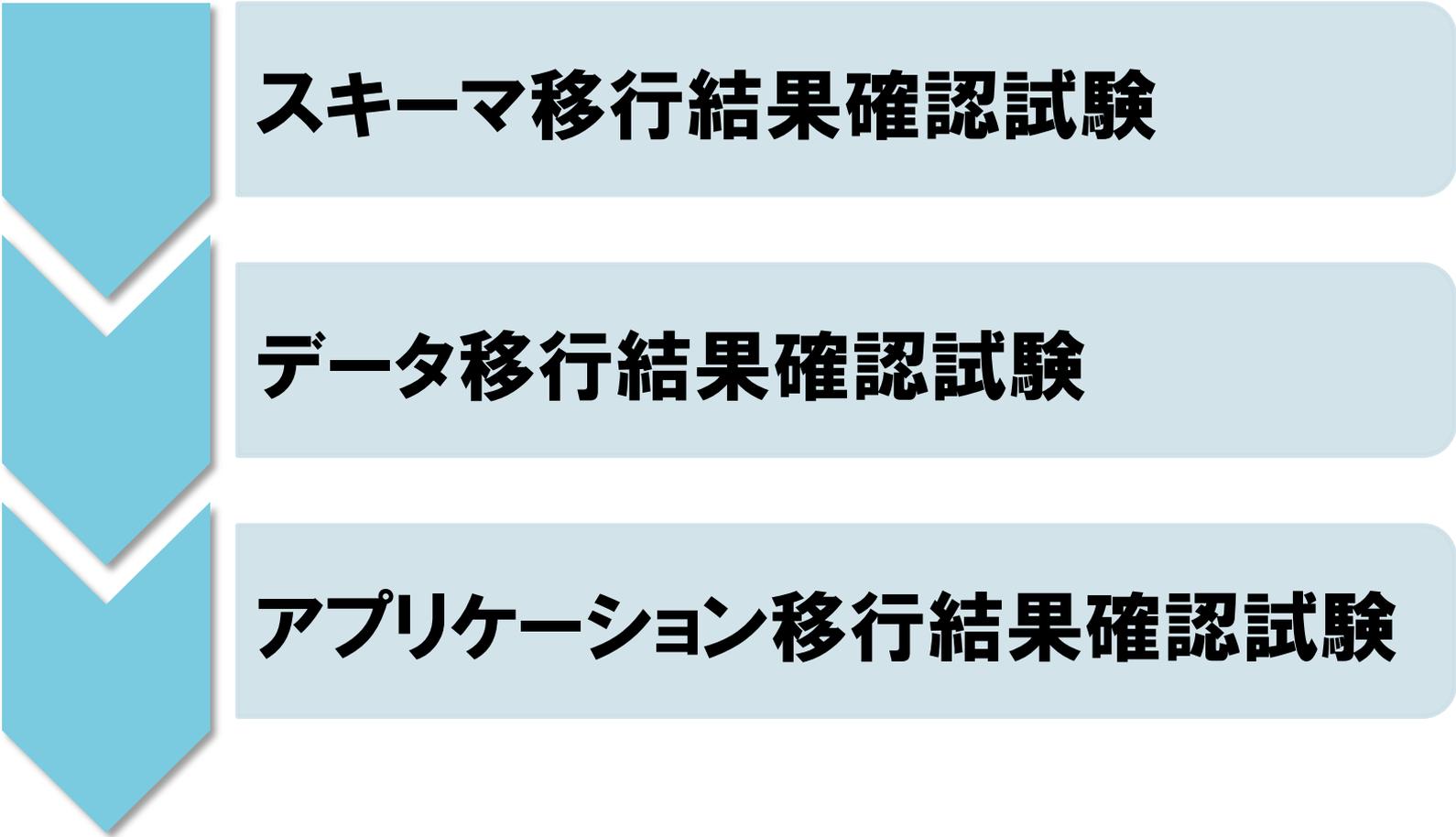
試験編の大項目

- 以下3項目についての調査を行いました。



発表の流れ

- 発表は以下の流れで実施します。



スキーマ移行結果確認試験

データ移行結果確認試験

アプリケーション移行結果確認試験

本編の紹介と併せて

- 本年度は「試験編」の調査に加えて、前年度までの成果物を利用して、OSSのアプリケーション「infoScoop」をPostgreSQLに移行(対応)する検証および、作成した試験を用いた移行結果確認試験を実施しました。
- 検証中に得られた気づきについては、右記のマークとともに紹介します。

検証結果

infoScoop採用の経緯(1/2)

- 移行対象アプリケーション選定のため、Oracleに対応したアプリケーションを検索した結果、下表が候補に挙がりました。
 - これまでに公開したWG2の文書がOracleからの移行を前提としていたため、Oracleに対応したアプリケーションを検索しています。

| No. | アプリケーション名 | 概要 |
|-----|-----------------------------------|---|
| 1 | infoScoop | 柔軟、軽快、直感的に使えるフリーの企業情報ポータル。 1万人以上の環境での稼働実績を持つ。 |
| 2 | Pentaho Business Analytics Server | 統合ビジネスインテリジェンス(BI)ツール。 |
| 3 | OpenInfo3W | ERP(統合業務パッケージ)、CRM(顧客関係管理)を自由に組み合わせることが可能なソフトウェア。 フランス語で開発されている。 |
| 4 | SugarCRM | CRM(顧客関係管理)ソフトウェア。 Enterprise Edition (EE) はOracleに対応。 |

infoScoop採用の経緯(2/2)

- 移行検証に適切かどうかを判断するため比較を実施しました。
 - PostgreSQL以外のDBMSには対応しており、アプリケーションの利用(構築)が容易であるということで、今回の実機検証の対象製品とさせていただきました。

| No. | 評価項目 | infoScoop | Pentaho | OpenInfo3W | SugarCRM (EE) |
|-----|--|------------------|------------------------------------|---------------------------------|-----------------------------|
| 1 | ストアドプロシージャの有無 | × | × | ○ | 有償製品のため、 情報が公開されて いない |
| 2 | データ型 (VARCHAR2等) | ○ | ○ | ○ | |
| 3 | Oracle の機能 SQL構文 (JOIN +) | Hibernateを利用 | Hibernateを利用 | ○ | |
| 4 | 地図情報、監査ログ | × | × | × | |
| 5 | アプリケーションの利用難易度 (アプリケーションを利用するまでの工数) | 容易 | 容易 | 困難 (ドキュメントが フランス語) | |
| 6 | 開発言語 | Java | Java | Java | |
| 7 | 対応DBMS | Oracle/MySQL/DB2 | Oracle/MySQL/ PostgreSQL | Oracle,その他不明 | |

infoScoopを採用

infoScoop動作画面

- 以下のように、様々なコンテンツウィンドウを好みに合わせてカスタマイズできたりと、利便性の高いポータルサイトを構築できます。

- <https://www.infoscoop.org/ja/>

The screenshot displays the infoScoop Cloud Enterprise portal interface. At the top, there is a navigation bar with the logo and user information (guest1). Below the navigation bar, a yellow notification banner indicates a new message. The main content area is divided into several widgets:

- Home:** Includes links for Home, Business, Operation Monitoring, Collaboration, Video Conference, Demo, and Map Search Demo.
- Company Noticeboard (All-Company Noticeboard):** Lists recent announcements such as server maintenance, personnel changes, and business hours.
- infoScoop OpenSource News:** Displays news items like the official website renewal and gadget gallery presentations.
- Schedule (スケジュール):** Shows a calendar view for the month of May, with the 10th (Sunday) highlighted.
- TODO List (TODOリスト):** A task management widget with an input field and an 'Add' button.
- Sports (スポーツナビ):** A widget for sports information, currently showing no data.
- Standard Time (標準時計):** Displays standard time for TYO (17:26:19), LDN (09:26:19*), and NYC (04:26:19*).
- Popular Gadgets (人気ガジェット):** A list of popular gadgets with 'Add' buttons for each.
- Calculator (電卓):** A standard numeric keypad calculator.

調査環境

- 試験編の調査や実機検証で利用したOS, DBMSおよびinfoScoop関連の主なソフトウェアのバージョンは以下の通りです。

| | ソフトウェア | バージョン | 備考 |
|-----------|----------------|---------------------|------------------------------|
| OS | CentOS | 6.5 | Oracle, PostgreSQL環境のOSとして利用 |
| | Windows Server | 2008 R2 | SQL Server環境のOSとして利用 |
| DBMS | PostgreSQL | 9.4.0, 9.4.1 | - |
| | Oracle | 11g Express Edition | - |
| | SQL Server | 2008 R2 Express | - |
| infoScoop | infoScoop | 3.4.0 | - |
| | Hibernate | 3.2.6 | オブジェクト関係マッピング(O/Rマッパ)ライブラリ |
| | Tomcat | 7.0.57 | - |

infoScoopの移行

- 各移行試験の前提条件として、事前に移行を完了させておく必要があります。
- 移行手法の詳細については過去ドキュメントに記載されているため、本年度の成果物には特に含めておりません。
- 今回infoScoopを使用した移行試験を行うにあたって実施した移行手順を次頁に記載します。

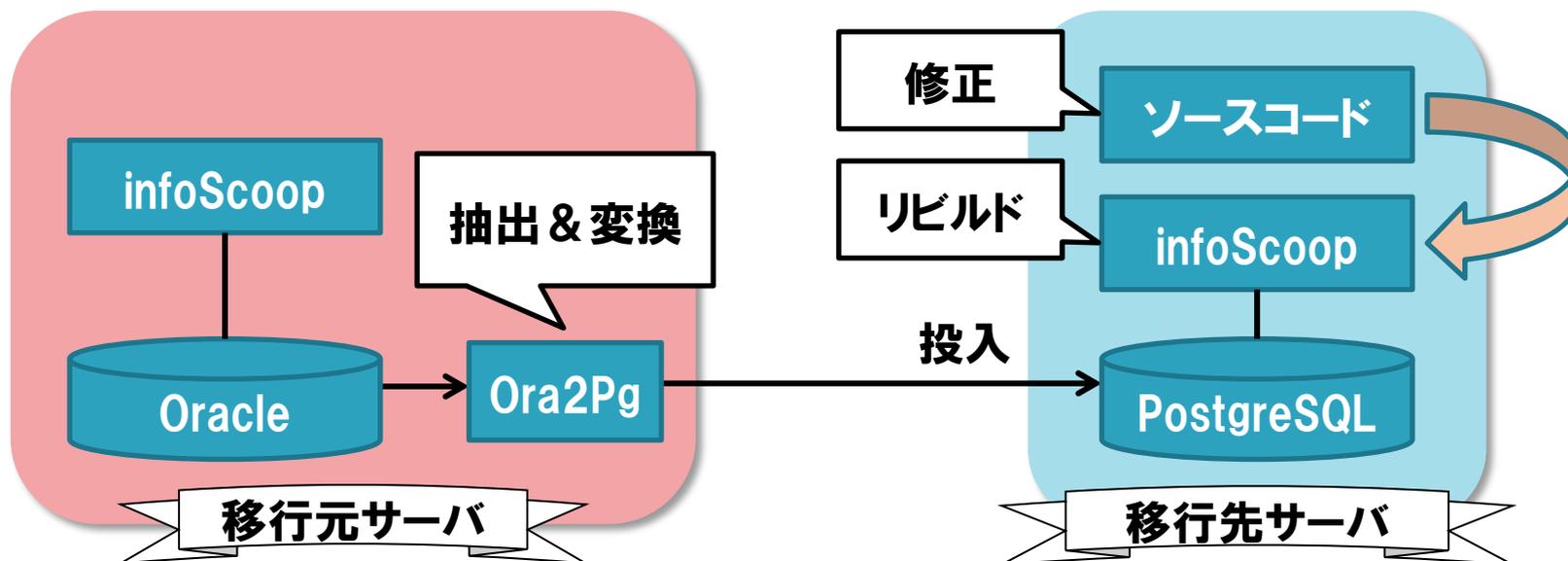
infoScoopの移行

検証結果

■ 以下手順で移行を実施しました。

1. 移行元サーバにinfoScoopを正規手順でインストール
2. 移行ツール「Ora2Pg(※)」を用いてスキーマ、データ等を抽出 & 自動整形
3. Ora2Pgだけでは取り除けなかった差異を手動、あるいはスクリプトで修正
4. infoScoopをPostgreSQL用にリビルドし移行先サーバに配置

※ Oracleからスキーマ・データ等を抽出し、PostgreSQL用に整形するツール



infoScoop移行で躓いた点(1/2)

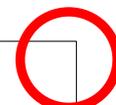
■ 予約語と同名のカラム名を持つテーブル

- 予約語と同名のカラムを定義する場合、ダブルクォートで囲う必要があります。(Ora2Pgによる自動整形では付与されません)

```
CREATE TABLE is_messages (  
  id bigint NOT NULL,  
  from varchar (150) NOT NULL,  
(以下略)
```



```
CREATE TABLE is_messages (  
  id bigint NOT NULL,  
  "from" varchar (150) NOT NULL,  
(以下略)
```



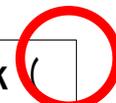
■ Ora2Pgによる自動型変換の修正

- boolean型として使用しているNUMBER(1)をsmallintに変換してしまうため、手動でboolean型に変更する必要があります。

```
CREATE TABLE databasechangeloglock (  
  id bigint NOT NULL,  
  locked smallint NOT NULL,  
(以下略)
```



```
CREATE TABLE databasechangeloglock (  
  id bigint NOT NULL,  
  locked boolean NOT NULL,  
(以下略)
```

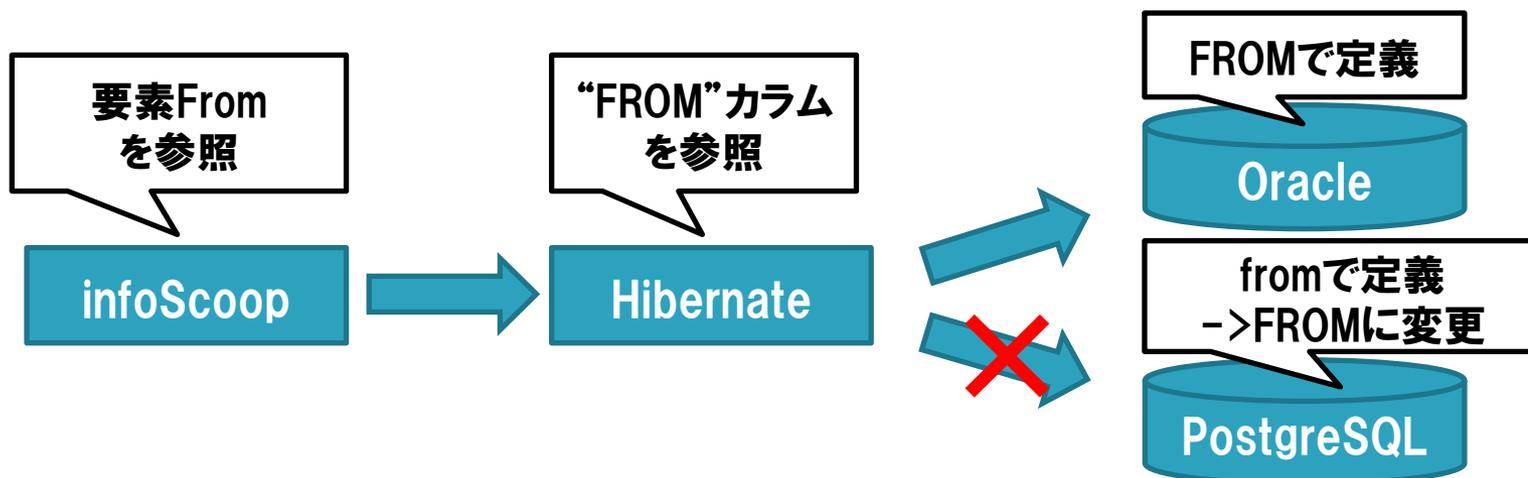


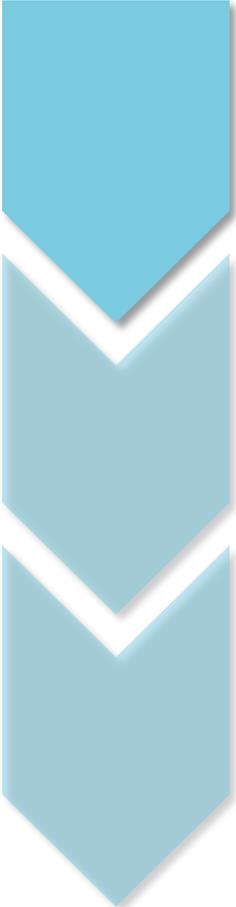
infoScoop移行で躓いた点(2/2)

検証結果

■ Hibernateの定義ファイルとの整合性

- infoScoopではDBMSの差異を吸収してくれるO/Rマツパ(Hibernate)を使用しているため、SQLの移行を行う必要がありませんでした。
- しかし移行にあたりカラム名を変更している場合は、Hibernateの定義ファイルと相違ないか確認する必要があります。
- 検証では大文字で参照されるカラムを小文字で移行してしまったため、PostgreSQLでも大文字に変更することで対処しました。
 - Ora2Pgが抽出した小文字のfromを単純にダブルクォーテーションで括ったため。





スキーマ移行結果確認試験

データ移行結果確認試験

アプリケーション移行結果確認試験

スキーマ移行結果確認試験

■ 試験観点

- スキーマ移行結果確認試験では、以下の観点で実施します。
 1. 異種DBMSおよびPostgreSQLのシステムカタログに登録されたスキーマオブジェクトを採取し、定義が一致していること。
 2. 移行設計時にあらかじめ差分(PostgreSQLに対応するための仕様変更など)として定義された箇所が定義通りになっていること。

■ 次スライドより、試験対象について説明します。

スキーマ移行結果確認試験

■ 試験対象と確認方法

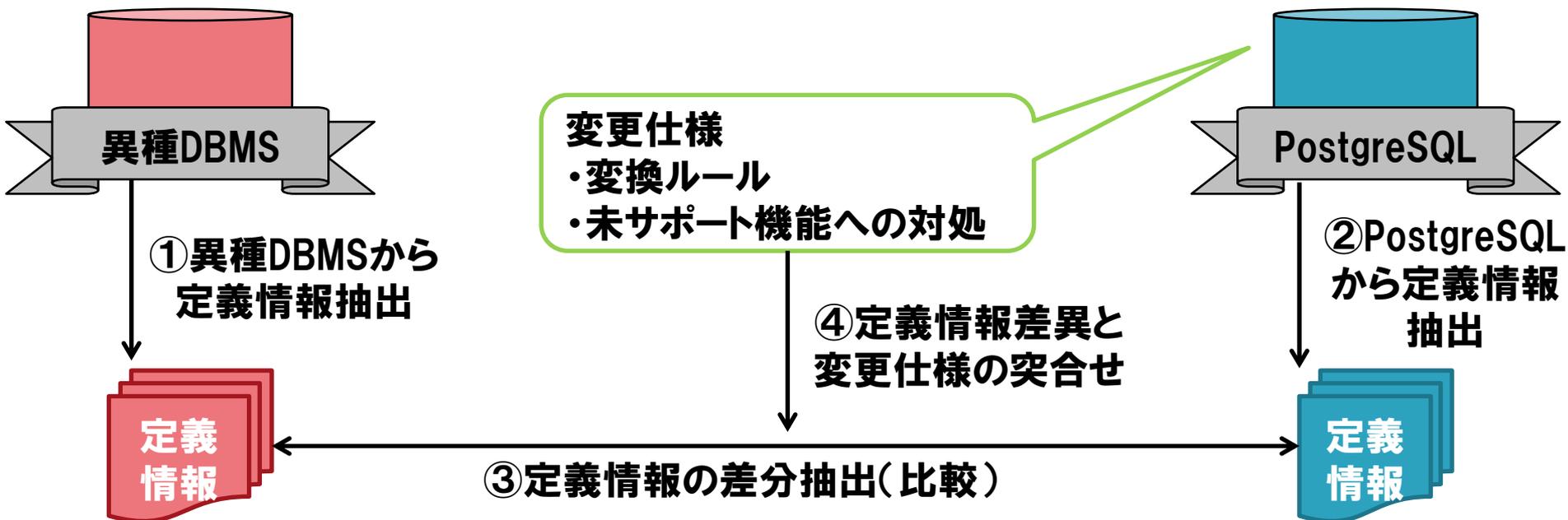
- スキーマ移行結果確認試験では、以下のスキーマオブジェクトに関する試験方法を記載しています。

| No. | 項目 | 確認方法 |
|-----|------------|--|
| 1 | スキーマ | 同名のスキーマが定義されていること。 |
| 2 | テーブル | 同名のテーブルが定義されていること。 |
| 3 | カラム | テーブルに同名のカラムが定義されていること。 カラムの属性が移行仕様通りの属性で定義されていること。 |
| 4 | インデックス | 同名のインデックスが、同じスキーマ、テーブルに対して定義されていること。 |
| 5 | ビュー | 同名のビューが、同じスキーマ、テーブルに対して定義されていること。 |
| 6 | シリアルオブジェクト | 同名のシリアルオブジェクトが定義されていること。 初期値、加算の単位、上限到達後の動作が同じであること。 |
| 7 | 制約条件 | 各種制約条件が定義されていること。 制約条件名で移行を確認しますが、NOT NULL や プライマリキー制約を除き、ロジックを含む定義自体は比較対象とはしません。 |

スキーマ移行結果確認試験

■ 試験手順

- 試験はおおまかに以下の手順で実施します。
- ① 異種DBMSから定義情報(スキーマオブジェクト)抽出
- ② PostgreSQLから定義情報(スキーマオブジェクト)抽出
- ③ 定義情報の差分抽出
- ④ 定義差異と移行時の変更仕様の突合せ



スキーマ移行結果確認試験

■ 各DBMSからの定義情報の抽出

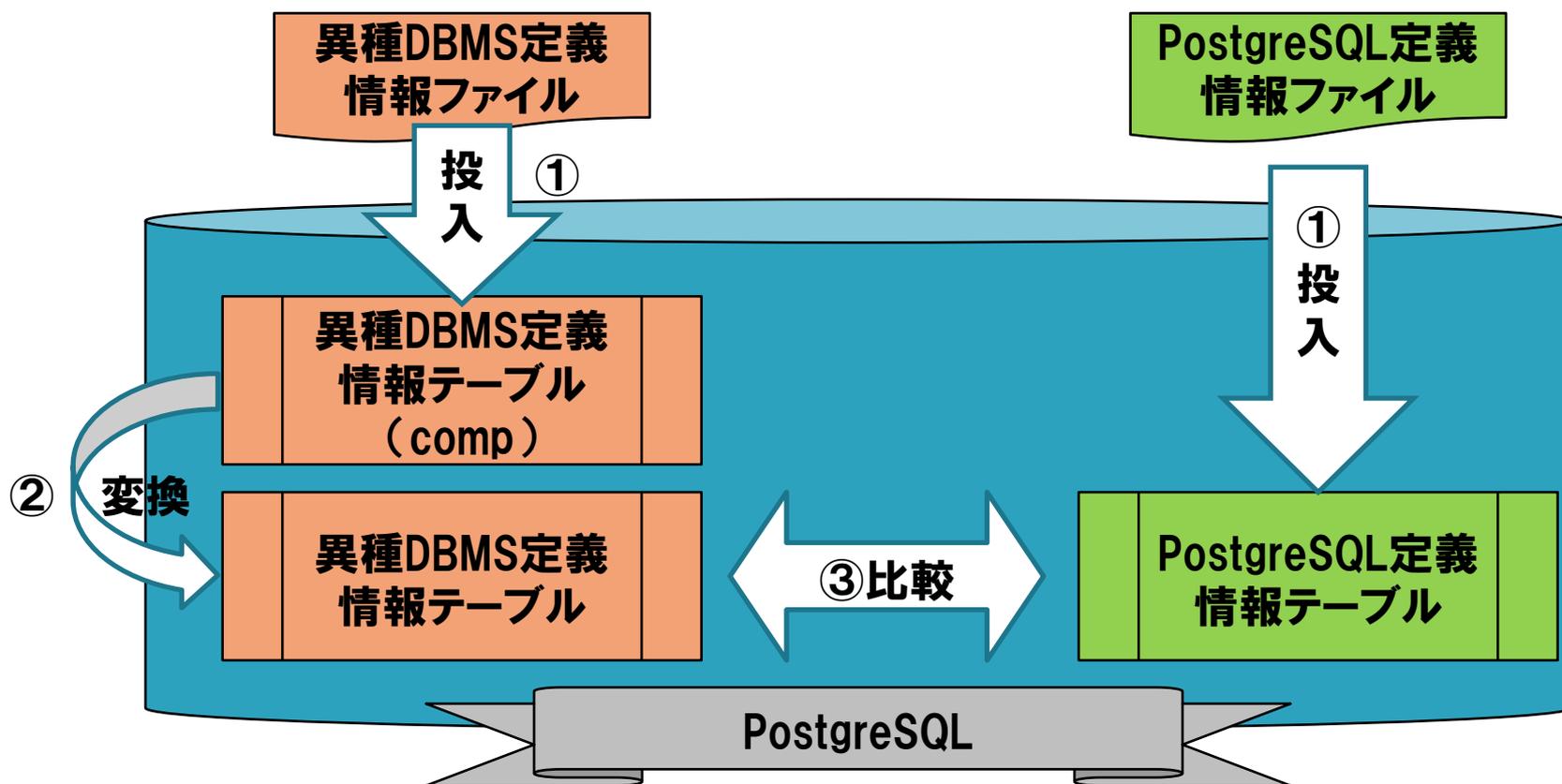
- 例として、テーブル名の比較に利用する情報の抽出方法を以下に記載します。

| No. | DBMS | システムカタログ | 抽出情報 | 抽出方法例 |
|-----|------------|---------------------------|---------------------|---|
| 1 | Oracle | ALL_TABLES | オーナ(スキーマ)名 テーブル名 | SELECT OWNER, TABLE_NAME FROM ALL_TABLES WHERE OWNER='オーナ名' |
| 2 | SQL Server | INFORMATION_SCHEMA.TABLES | スキーマ名 テーブル名 | SELECT TABLE_SCHEMA, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' |
| 3 | PostgreSQL | INFORMATION_SCHEMA.TABLES | スキーマ名 テーブル名 | SELECT TABLE_SCHEMA, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TABLE_SCHEMA NOT IN ('information_schema', 'pg_catalog') |

スキーマ移行結果確認試験

■ 定義情報の比較

- 例示する手法では一旦PostgreSQLに投入(①)し、その中でSP00L出力の空白を削るためのtrim等の形式の変換(②)を行った後、比較(③)を行います。



スキーマ移行結果確認試験

検証結果

■ 定義情報の比較(1/3)

- infoScoop検証時のスキーマ移行結果確認試験から、カラム定義比較試験を抜粋します。
- 以下は、移行元と移行先(PostgreSQL)のカラム定義を格納するためのPostgreSQLのテーブル情報です。

| comp | | | |
|------|------------------------|---------|------------------|
| No. | 列名 | 属性 | 説明 |
| 1 | table_schema | text | スキーマ名 |
| 2 | table_name | text | テーブル名 |
| 3 | column_name | text | カラム名 |
| 4 | data_type | text | 型 |
| 5 | character_octet_length | integer | 列長 |
| 6 | numeric_precision | integer | 数値(numeric)の最大精度 |
| 7 | numeric_scale | integer | 数値(numeric)の位取り |
| 8 | is_nullable | boolean | NULL値の格納可否 |
| 9 | column_default | text | デフォルト値 |

スキーマ移行結果確認試験

検証結果

■ 定義情報の比較(2/3)

- infoScoop検証における変換仕様を抜粋して紹介します。

| カテゴリ | Oracle | - | - | PostgreSQL |
|------|----------|-----|-----|------------|
| | 型 | 位取り | 精度 | 型 |
| 文字列 | VARCHAR2 | - | - | varchar |
| | LONG | - | - | text |
| 数値 | NUMBER | 0 | 9以下 | integer |
| バイナリ | BLOB | - | - | bytea |
| | CLOB | - | - | text |
| 日時 | DATE | - | - | timestamp |

スキーマ移行結果確認試験

検証結果

■ 定義情報の比較(3/3)

- カラム定義情報の比較試験を以下に抜粋します。

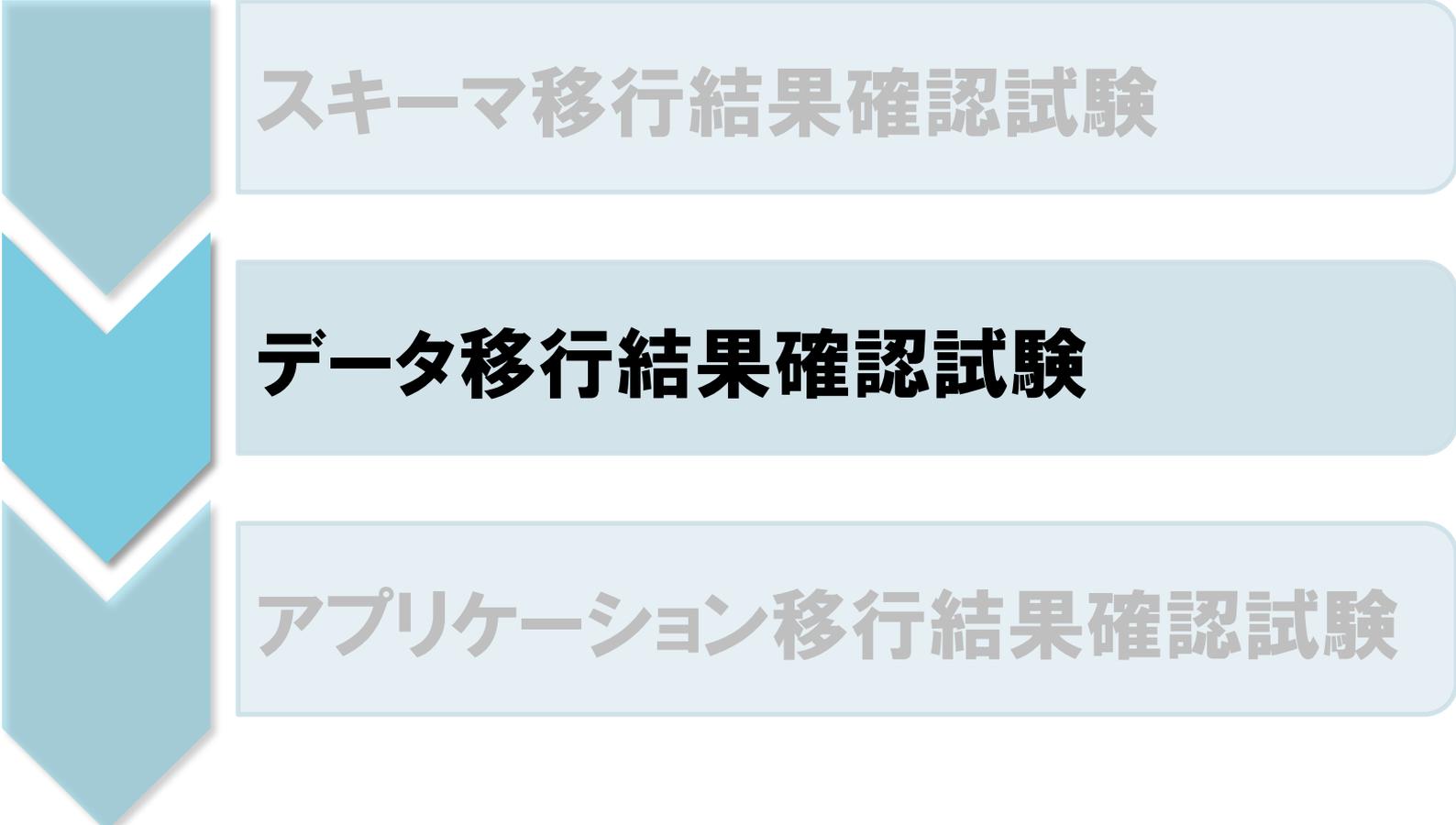
```
db=# select t1.table_name as oracle_tbl,t1.column_name as oracle_col,
db=#         t2.table_name as postgres_tbl,t2.column_name as postgres_col,
db=# case
db=#     when t1.column_name is null then 'NG'
db=#     when t2.column_name is null then 'NG '
db=#     when (t1.data_type = 'VARCHAR2' and t2.data_type = 'character varying' and
db=#           t1.character_octet_length = t2.character_octet_length) then 'OK'
db=#     when (t1.data_type = 'CLOB' and t2.data_type = 'text')           then 'OK'
```

(中略、詳細は本年度ドキュメントに記載)

```
db=#         t1.column_name = upper(t2.column_name))
db=# order by 1,2 ;
```

| oracle_tbl | oracle_col | postgres_tbl | postgres_col | case |
|---------------|------------|---------------|--------------|------|
| IS_ACCESSLOGS | DATE | is_accesslogs | date | OK |
| IS_ACCESSLOGS | ID | is_accesslogs | id | OK |
| (中略) | | | | |
| IS_WIDGETS | UID | is_widgets | uid | OK |
| IS_WIDGETS | WIDGETID | is_widgets | widgetid | OK |

(186 rows)



スキーマ移行結果確認試験

データ移行結果確認試験

アプリケーション移行結果確認試験

データ移行結果確認試験

■ 試験項目と概要

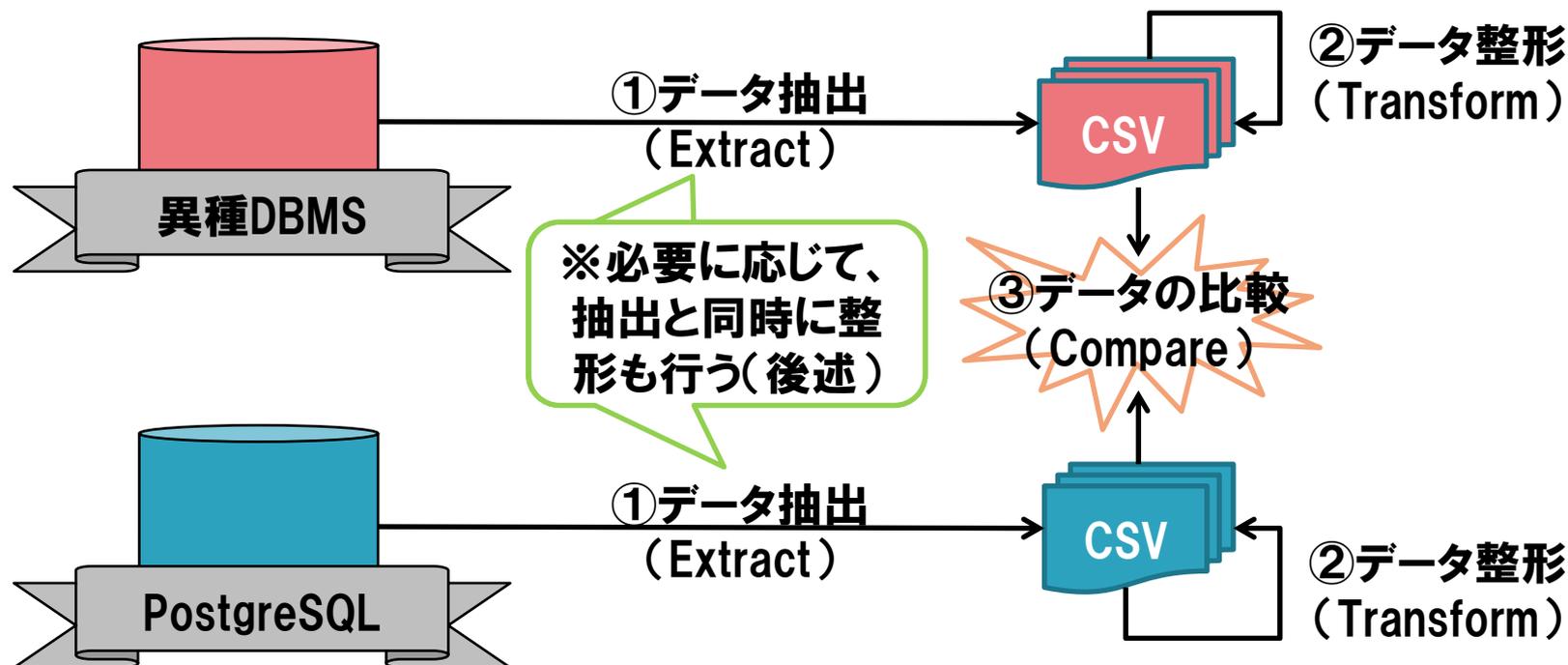
- データ移行結果確認試験の項目を以下に挙げます。
- 本日は**赤字部分**のデータ内容の確認についての成果を紹介します。

| No. | 試験項目 | 概要 |
|-----|-----------------|--|
| 1 | テーブル数の確認 | 異種DBMSに定義されたテーブル数と、PostgreSQLに定義されたテーブル数が同一であることを確認します。本試験項目で試験対象とするテーブルを決定します。 |
| 2 | データ件数の確認 | 異種DBMSに格納されている各テーブルのデータ件数と、PostgreSQLに格納されている各テーブルのデータ件数が同一であることを確認します。 |
| 3 | データサイズの確認 | 異種DBMSに格納されたデータ件数およびデータの種別等から、PostgreSQL移行後のデータサイズの想定値を算出します。PostgreSQLの実際のデータサイズと事前に算出した想定値が近似であることを確認します。 |
| 4 | 外字登録の確認 | 異種DBMSに登録された外字がPostgreSQLにも正しく登録されていることを確認します。 |
| 5 | データ内容の確認 | 異種DBMSに格納されたデータとPostgreSQLに格納されたデータが同一であることを確認します。成果物では、異種DBMSから出力したCSVファイルと、PostgreSQLから出力したCSVファイルの内容が同一であることを確認します。 |

データ移行結果確認試験

■ データ内容の確認試験

- 異種DBMSとPostgreSQLに格納されたデータが同一であることを以下の手順で確認します。
 - 異種DBMSとPostgreSQLのデータの一致を目指す、データを整形する必要があります。
 - どこまでデータを整形してよいか、データの一致を目指すかについては、プロジェクトごとに定義をお願いします。



データ移行結果確認試験

■ データ内容の確認試験(データの抽出)

- 成果物では、Oracle(SPOOL)、SQL Server(bcp)、PostgreSQL(COPY)についてのコマンド例を記載しています。
- 以下のような点に注意して、各DBMSからデータを抽出します。

| No. | DBMS | 項目 | 概要と対処 |
|-----|------------|--------------------------------------|---|
| 1 | Oracle | SELECT文の記述方法 | SPOOLを使用したCSV出力を行う際、「SELECT *」で出力してしまうと固定長のCSVファイルができてしまうため、セパレータをクエリ内に埋め込むことで対処(「 ',' 」等)します。 |
| 2 | PostgreSQL | COPYコマンドのオプション ”FORMAT CSV”を付与しない | COPYコマンドでは一般的にFORMAT CSVオプションを付与して出力しますが、ダブルクォーテーションが自動付与されるなど、データ比較には適さない出力となるため、当該オプションは付与しません。 |
| 3 | | タイムスタンプ型の整形 | Oracle, SQL Server, PostgreSQLでは日付型のミリ秒表示の形式が異なるため、to_char()関数を使用し、出力フォーマットをOracle, SQL Serverに合わせます。 |

データ移行結果確認試験

検証結果

■ データ内容の確認試験(データの抽出)

- infoScoopにはSPOOLとCOPYで出力形式の異なるカラムや、SPOOLの文字数上限を超過する可能性のあるカラムが含まれていたため、前スライドの対処のみではデータが一致しませんでした。

```
0,0,default,defaultRole,OrganizationPrincipal,default,<?xml version="1.0" encoding="UTF-8"?>
<widgets uid="default" tabId="0" tabName="ホーム" tabType="static">
  <panel type="StaticPanel">
    <widget column="1" href="" id="p_1_w_1" title="infoScoop OpenSource site" type="RssReader">
      <data>
        <property name="url">http://www.infoScoop.org/index.php/ja/news.feed</property>
      </data>
    </widget>
  </panel>
</widgets>
```

(以下略)

Oracle(SPOOL)抽出結果

タブや改行が
エスケープされない

SPOOLの出力文字
数制限を超過する
場合がある

```
0,0,default,defaultRole,OrganizationPrincipal,default,<?xml version="1.0" encoding="UTF-8"?>¥r¥n<widgets uid="default" tabId="0" tabName="ホーム" tabType="static">¥r¥n¥t<panel type="StaticPanel">¥r¥n¥t¥t<widget column="1" href="" id="p_1_w_1" title="infoScoop OpenSource site" type="RssReader">¥r¥n¥t¥t¥t<data>¥r¥n¥t¥t¥t<property name="url">http://www.infoScoop.org/index.php/ja/news.feed</property>¥r¥n¥t¥t¥t</data>¥r¥n¥t¥t</widget>¥r¥n¥r¥n¥t¥t<widget column="3" href="" id="p_1_w_5" title="Calendar" type="Calendar">
```

PostgreSQL(COPY)抽出結果

タブや改行が
エスケープされる

データ移行結果確認試験

検証結果

■ ハッシュ値による抽出

□ 今回は以下のカラムの値をハッシュ値に変換して比較を行いました。

1. SPOOLの制限値(3万字程度)を超過する可能性のあるカラム(BLOB、CLOB等)
2. タブや改行を含むカラム(VARCHAR等の文字列型)

■ CSVを抽出する際のSELECT文の出力カラムをハッシュ関数に通します。

■ Oracle

□ rawtohex (DBMS_CRYPTO.Hash (“カラム名”,2))

■ PostgreSQL

□ upper (md5 (“カラム名”))

```
0,0,C21F969B5F03D33D43E04F8F136
E7682,defaultRole,OrganizationPrincipal,d
efault,55FD0A9B2509049650C4172C3
360E3B2,E4F31D5048605EDAF51BF34
E66338576,-,0,,
```

Oracle(SPOOL)抽出結果例

```
0,0,C21F969B5F03D33D43E04F8F136
E7682,defaultRole,OrganizationPrincipal,d
efault,55FD0A9B2509049650C4172C3
360E3B2,E4F31D5048605EDAF51BF34
E66338576,-,0,¥N,¥N
```

PostgreSQL(COPY)抽出結果例

データ移行結果確認試験

■ データ内容の確認試験(データの整形)

- SQL ServerとPostgreSQLを比較する場合の整形方針例を以下に記載します。

| No. | 項目 | SQL Server(bcp) | PostgreSQL(COPY) | 対処 |
|-----|---------|-----------------|--------------------------------|--|
| 1 | NULLの出力 | 空文字に変換し出力 | NULL (¥Nまたは指定したNULLの代替文字列) を出力 | SQL Serverの空文字をNULLに置き換える必要があります。その際、CSVの出力時にPostgreSQL側で設定したNULLの代替文字列に置換します。 |
| 2 | 空文字の出力 | NULLに変換し出力 | 空文字を出力 | SQL Server側のNULLは本来は空文字ですので、空文字に変換します。 |
| 3 | 改行の出力 | 改行コードを出力 | エスケープされた改行文字を出力 | PostgreSQLの改行文字を改行コードに変換します。(SQL Serverの改行コードはCSVの終端改行コードと区別がつかないため。) |

データ移行結果確認試験

■ データ内容の確認試験(データの整形)

□ infoScoop移行検証では以下の整形を行いました。

■ ソート処理

□ OSコマンドのsortを利用して、1行1データとしてソートしました。

■ NULL文字(¥N)を空文字に置換

■ エスケープ文字の"¥"を空文字に置換

□ 以下は変換結果の一例になります。

```
0,0,C21F969B5F03D33D43E04F8F136
E7682,defaultRole,OrganizationPrincipal,d
efault,55FD0A9B2509049650C4172C3
360E3B2,E4F31D5048605EDAF51BF34
E66338576,-,0,,
```

```
0,0,C21F969B5F03D33D43E04F8F136
E7682,defaultRole,OrganizationPrincipal,d
efault,55FD0A9B2509049650C4172C3
360E3B2,E4F31D5048605EDAF51BF34
E66338576,-,0,¥N,¥N
```

```
0,0,C21F969B5F03D33D43E04F8F136
E7682,defaultRole,OrganizationPrincipal,d
efault,55FD0A9B2509049650C4172C3
360E3B2,E4F31D5048605EDAF51BF34
E66338576,-,0,,
```

```
0,0,C21F969B5F03D33D43E04F8F136
E7682,defaultRole,OrganizationPrincipal,d
efault,55FD0A9B2509049650C4172C3
360E3B2,E4F31D5048605EDAF51BF34
E66338576,-,0,,
```

一致

整形結果例(Oracle)

整形結果例(PostgreSQL)

データ移行結果確認試験

■ データ内容の確認試験(データの比較)

- 抽出および必要な整形が完了した後、データの比較を行います。
- 成果物にて、Linux系OSの「diff」とWindows Serverの「fc」コマンドについて記載しています。
(コマンド例はスライドでは割愛します。)

データ移行結果確認試験

検証結果

■ データ内容の確認試験(データの比較)

- 抽出した各CSVの差分ファイルをdiffコマンドで取得しました。

```
# diff [Oracle出力CSV] [PostgreSQL出力CSV] > /tmp/diff_result/[CSVファイル名].txt
```

- テーブル毎にCSV([テーブル名.csv])を出力し、それぞれについてdiffを実行した結果ファイルの「サイズ」を確認し、データが一致していることを確認しました。
 - diffは差異がなければ何も出力しないため。(デフォルト)

```
# ls -s /tmp/diff_result/  
合計 0  
0 DATABASECHANGELOG.txt          0 IS_OAUTHPROVIDER_CLIENT.txt  
0 DATABASECHANGELOGLOCK.txt      0 IS_OAUTHPROVIDER_REFRESH_TOKENS.txt  
0 IS_ACCESSLOGS.txt              0 IS_OAUTH_CERTIFICATE.txt  
0 IS_ACCOUNTS.txt                0 IS_OAUTH_CONSUMERS.txt  
0 IS_ADMINROLES.txt              0 IS_OAUTH_GADGET_URLS.txt  
0 IS_AUTHCREDENTIALS.txt         0 IS_OAUTH_TOKENS.txt  
0 IS_CACHES.txt                  0 IS_PORTALADMINS.txt  
(以下略)
```



スキーマ移行結果確認試験

データ移行結果確認試験

アプリケーション移行結果確認試験

アプリケーション移行結果確認試験

■ 試験項目

- アプリケーション移行結果確認試験の項目を以下に挙げます。

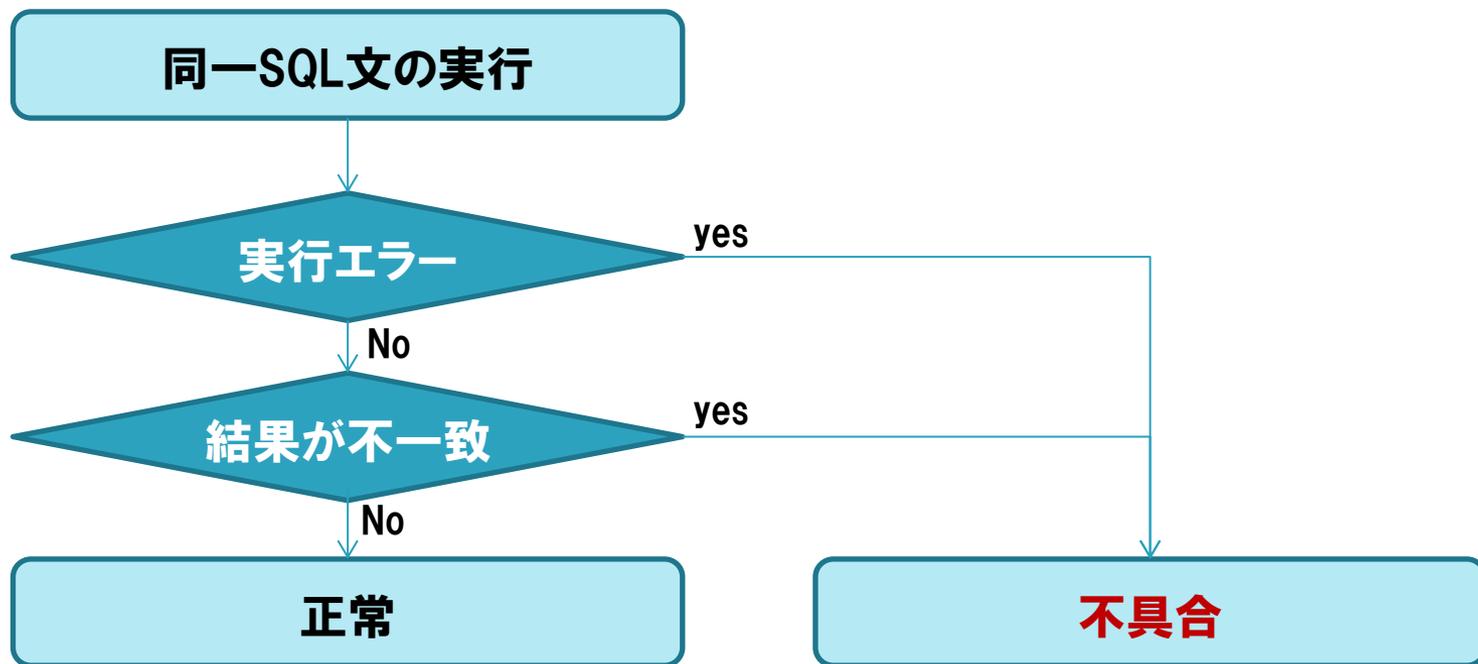
| No. | 試験項目 | 試験観点 |
|-----|---------|---|
| 1 | SQL文試験 | 異種DBMSとPostgreSQLの場合でSQL文の処理結果が変わらないことを確認します。またDBMSの仕様差異を埋めるため、SQL文の修正を行った場合は、修正したSQLの処理結果が想定通りであることを確認します。 |
| 2 | 業務バッチ試験 | 埋め込みSQLなどのバッチ処理が存在する場合、SQL文試験としてではなく、バッチ処理単位で試験を行うべきと考えました。 DBMS移行前後でバッチの実行結果に差異がないことを確認します。 日次や月次バッチなど、業務の種別ごとに試験を実施する必要があります。 |
| 3 | 画面試験 | DBMS移行前と同様に、画面表示・画面操作が正しく動作することを確認します。 |
| 4 | 性能試験 | アプリケーションがDBMS移行前と同等以上の性能を保持しているかを確認します。 |

- 以降のスライドで「SQL文試験」「画面試験」「性能試験」について説明を記載します。

アプリケーション移行結果確認試験

■ SQL文試験

- 移行元および移行先(PostgreSQL)にて同一処理のSQLを実行し、その結果を元に確認を行います。



- 結果が不一致となり「**不具合**」判定となる事象例を次スライドに記載します。

アプリケーション移行結果確認試験

■ SQL文試験

- 「予期せぬSQL実行結果の不一致」をとして以下のような例を記載しています。
- ソート順序の違い
 - SELECT ID, VALUE ORDER BY VALUE ASC FROM TESTTBL;

| ID | VALUE |
|----|--------|
| 1 | 空文字 |
| 2 | 非NULL値 |
| 3 | NULL値 |

PostgreSQL

| ID | VALUE |
|----|--------|
| 1 | 非NULL値 |
| 2 | NULL値 |
| 3 | 空文字 |

Oracle

Oracleでは
区別しない

| ID | VALUE |
|----|--------|
| 1 | NULL値 |
| 2 | 空文字 |
| 3 | 非NULL値 |

SQL Server

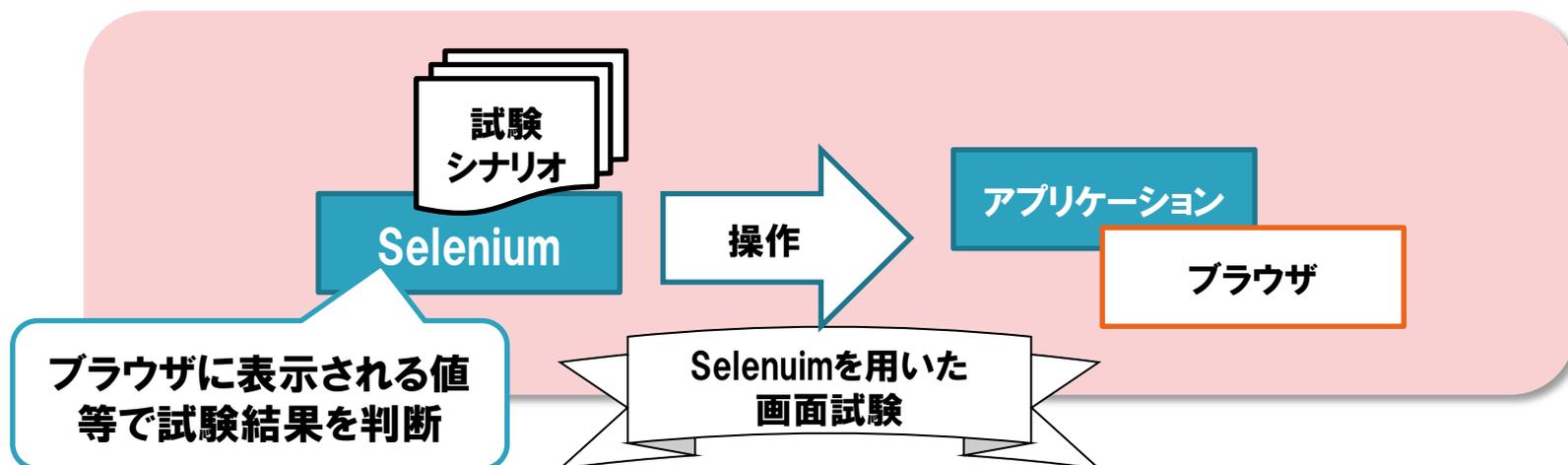
- 精度を省略した場合の違い

| No. | DBMS | 項目 |
|-----|------------|--|
| 1 | PostgreSQL | DECIMAL, NUMERICの精度を省略した場合、最大限の精度を取ります。 |
| 2 | Oracle | DECIMAL, NUMERICの精度を省略した場合、最小限の精度となります。 (小数点以下が切り上げられる) |
| 3 | SQL Server | DECIMAL, NUMERICの精度を省略した場合、最小限の精度となります。 (小数点以下が切り上げられる) |

アプリケーション移行結果確認試験

■ 画面試験

- 画面試験ではDBMS移行前後で画面遷移、画面の表示内容が変わらないことを確認します。
- 成果物では、画面試験はWEBアプリケーションのテストツールである、「Selenium」を紹介しています。
 - ブラウザの操作を自動化しWebアプリケーションのリグレッション試験を実施
 - 自動化する試験シナリオは、様々な言語 (Java、C#等) で実装可能
 - 試験結果をレポートに出力可能
 - <https://github.com/SeleniumHQ/selenium>



アプリケーション移行結果確認試験

検証結果

■ 画面試験

- infoScoop向けのSelenium用のシナリオが下記より入手可能でしたので、当該シナリオを利用して試験を実施しました。
 - <https://github.com/infoScoop/infoScoop-selenium>
- 試験結果のサマリーは下表の通りです。
 - 各シナリオの結果を確認し、PostgreSQLがエラーもしくは失敗になっている箇所を確認し、原因を調査しました。

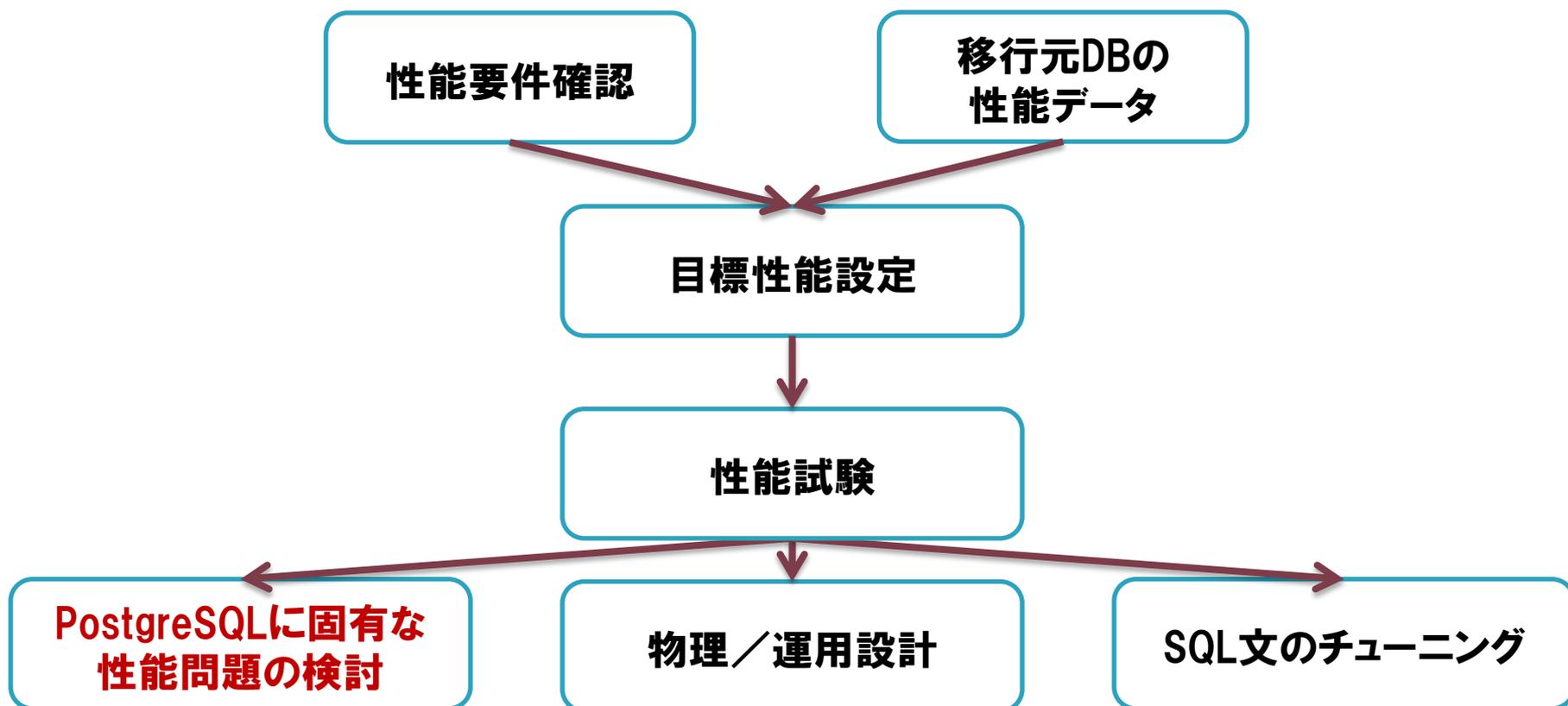
| データベース | シナリオ数 | エラー | 失敗 | スキップ | 成功率 |
|------------|-------|-----|----|------|--------|
| PostgreSQL | 176 | 18 | 11 | 0 | 83.52% |
| Oracle | 176 | 23 | 7 | 0 | 82.96% |

PostgreSQLがエラーもしくは失敗した原因を調査した結果、以下のいずれかであった。

- ・仮想環境のスペック不足によるタイムアウト(試験シナリオを手動で再実施すると問題なく動作する)
- ・他DBMSでも同一のエラー/失敗となる。

アプリケーション移行結果確認試験

- 性能試験
- PostgreSQLにおけるcheckpointやWALの書き込み競合等、性能面で影響を与える可能性がある点についての確認観点、方法を挙げています。



アプリケーション移行結果確認試験

■ 性能試験

- 例として、単位時間当たりのWAL書き込み量の確認方法を記載します。
- 遅延SQLが検出された際に、I/O負荷とWAL(トランザクションログ)の書き込み量が共に上昇していた場合はWAL領域(pg_xlog)とデータ領域を別のディスクにすることで改善する可能性があります。

```
--ある時点でのログの位置を算出
=# SELECT 'xlog' as xlog, pg_current_xlog_insert_location () as xlog_insert_loc;
xlog | xlog_insert_loc
-----+-----
xlog | 8/2E000020

--次の時点でのログの位置を算出
=# SELECT 'xlog' as xlog, pg_current_xlog_insert_location () as xlog_insert_loc;
xlog | xlog_insert_loc
-----+-----
xlog | 8/30625298

--上記ログ位置の差分を算出
=# select '8/30625298'::pg_lsn - '8/2E000020'::pg_lsn;
?column?
-----
39998072
(1 row)
```

まとめ

- スキーマ移行、データ移行、アプリケーション移行について確認試験を行うことにより、確実な移行に大きく寄与できる成果物になったと考えています。

【スキーマ移行試験】

さまざまなオブジェクトの抽出方法、比較方法についての調査

確実な移行

【データ移行試験】

バイナリ等、複雑なデータ型についての試験手法も調査

【アプリケーション移行試験】

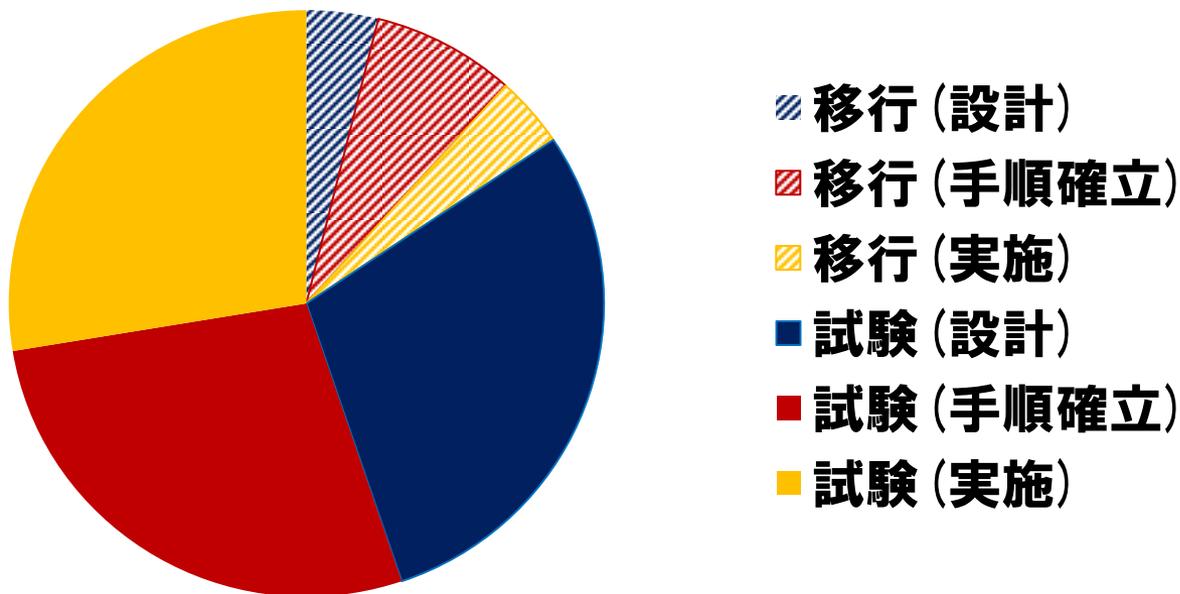
画面試験ツール等の利用方法についての調査



おわりに

おわりに

- infoScoopの移行検証においても、テスト対応(移行結果の確認)工数が85%以上となりました。
- 実際の移行時は、必要なテスト項目および対象オブジェクトを精査し、テスト工数が膨大にならない工夫が必要であると感じました。
 - 試験編調査では、可能な限り多くの項目の完全一致を目指しました。
 - 「試験」部分の青と赤が本年度の主な成果物となりますので、成果物を参照していただくことで、皆様の移行工数の削減につながれば幸いです。



おわりに

- また、本年度の成果物ではストアドプロシージャや関数など、ロジックを含むオブジェクトや地図情報等の移行難易度の高いオブジェクトの移行試験については本年度の記載は見送らせていただいているため、課題としては依然として残されています。
- 来年度以降も上記課題の解決、あるいは別の切り口から PostgreSQL への移行を促進するための調査及び検証を継続して行っていきたいと考えております。

参加企業からの感想など

- データ比較試験を突き詰めるとここまで工数がかかることは想定していなかった。しかし、調査の過程で多くの知見を得ることができました。
- 楽しかった。
- アプリケーション移行結果確認試験の観点洗い出しはもう少し深掘りできたように思いました。
- 物書きがつかったですが、合同で移行検証が行えたことは有益でした。

**(2014)2015年度は、具体的な作業や方法の
議論主体の活動になる
(かもしれません)**



PGECons

PostgreSQL Enterprise Consortium