リナックスアカデミー

LPI-JAPAN

OSS-DB Exam Silver 技術解説無料セミナー

2019/10/19

貴社名　　リナックスアカデミー

講師お名前 野口 庄一

アジェンダ

OSS-DB

- OSS-DB技術者認定資格
- PostgreSQL
- SQL (Structured Query Language)
- 演算子と関数
- トランザクション
- データベースオブジェクト
- バックアップとリカバリ
- メンテナンス

自己紹介

OSS-DB

- リカレント リナックスアカデミー　OSS-DBアカデミック認定校）
- 野口 庄一

- 2003年にリナックスアカデミーと契約。PostgreSQLトレーニングを含む開発系の講師に従事しています。
  その他、Webアプリケーションの開発などに携わっています。

オープンソースデータベース（OSS-DB）に関する技術と知識を認定するIT技術者認定

OSS-DB /Silver

データベースシステムの設計・開発・導入・運用ができる技術者

OSS-DB /Gold

大規模データベースシステムの
改善・運用管理・コンサルティングができる技術者

OSS-DB技術者認定資格の必要性

商用/OSSを問わず様々なRDBMSの知識を持ち、データベースの構築、運用ができる、または顧客に最適なデータベースを提案できる技術者が求められている

OSS-DB Exam Silver出題範囲

OSS-DB

■一般知識(16%)

- OSS-DBの一般的特徴 【重要度4】
- RDBMSに関する一般的知識【4】

■運用管理(52%)

- インストール方法【2】
- 設定ファイル【5】
- 基本的な運用管理作業【7】
- 標準付属ツールの使い方【5】
- バックアップ方法【7】

■開発/SQL(32%)

- SQLコマンド【13】
- トランザクションの概念【1】
- 組み込み関数【2】

Silver認定の基準

OSS-DB

- 下記のスキルと知識を持つエンジニアであることを証明する
  - RDBMSとSQLに関する知識を有する。
  - オープンソースデータベースに関する基礎的な知識を有する。
  - オープンソースを利用して小規模なデータベースの運用管理ができる。
  - オープンソースを利用して小規模なデータベースの開発を行う事ができる。
  - PostgreSQLなどのOSS-DBを使ったデータベースシステムの運用管理ができる。
  - PostgreSQLなどのOSS-DBを利用した開発でデータベース部分を担当することができる。

採用するRDBMS

OSS-DB

- 本試験は、OSS-DBを構築運用する能力を認定するために、OSS-DBのなかでも、特に商用データベースとの連携に優れ、エンタープライズ・システムでも多く活用されている

  PostgreSQL　10 以上

  を基準のRDBMSとして採用

■PostgreSQL

OSS-DB

- 無償利用可能 (BSDベースの PostgreSQLライセンス)
- ソースコードを公開
- 多くのプラットフォームで稼動
- 多くのプログラミング・インターフェースをサポート
- クライアント / サーバー・アーキテクチャー
- マルチバイトのサポート(日本語化、国際化に対応)
- データベースごとに文字エンコーディングを指定できる
- 必要に応じて文字エンコーディングを変換して利用できる
- ユーザー定義関数 / ユーザー定義データ型のサポート
- オブジェクト指向機能

用語

OSS-DB

- インスタンス
  - データベースを構成するファイルや共有メモリ、プロセスなどを合わせたもの
  - PostgreSQLサーバの起動単位
- データベースクラスタ
  - インスタンスに関連するファイルを格納する領域
  - インスタンスごとにひとつ
  - ディレクトリパスを環境変数$PGDATAに設定
  - ひとつのサーバ内に複数作成可能

アーキテクチャー

OSS-DB

- クライアントサーバー構成
- サーバー
  - マルチプロセス構成
  - 全体を管理するpostmasterプロセス
  - 目的別に複数のpostgresプロセス
  - クライアントひとつに対してひとつのpostgresプロセス
- ファイル構成
  - データベースクラスタの $PGDATA/base ディレクトリ下に格納

■PostgreSQLのインストール

OSS-DB

■ソースからインストール
- Cコンパイラとビルドツールを用いてソースコードをビルド
- PostgreSQLの公式サイトからソースコードをダウンロード
  - http://www.postgresql.org/ftp/source/

■インストーラを使ったインストール(ワンクリックインストール)
- EnterpriseDB社のサイトから、ビルド済みのパッケージをダウンロード
  - http://www.enterprisedb.com/products-services-training/pgdownload
- GUIの管理ツール(pgAdmin III)もあわせてインストール

■パッケージ管理システムを使ってインストール
- postgresql10-server

パッケージ管理システムによるインストール

OSS-DB

- インストール
  - # yum -y install postgresql10-server
- データベースクラスタの初期化
  - # /usr/pgsql-10/bin/postgresql-10-setup initdb
  - テンプレートデータベース（template0, template1)と postgres データベースが作成
    - template0は書き込み不可
    - template1は書き込み可（CREATE DATABASE のコピー元）

パッケージ管理システムによるインストール

OSS-DB

関連ファイル

- /usr/pgsql-10 以下に配置
- OSユーザーpostgresを作成
  - ホームディレクトリー　　/var/lib/pgsql
  - passwd でパスワード設定必要
  - .bash_profileに環境変数PGDATA設定済み
    - /var/lib/pgsql/10/data

インストール後の設定

OSS-DB

環境変数

- $ vi ~/.bash_profile
  - PGHOME=/usr/pgsql-10
  - export PATH=$PGHOME/bin:$PATH
  - export LD_LIBRARY_PATH=$PGHOME/lib:$LD_LIBRARY_PATH
  - export MANPATH=$PGHOME/share/man:$MANPATH
  - export PGDATA=/var/lib/pgsql/10/data
- 設定の反映
  - $. ~/.bash_profile

サービスの登録 (必要があれば)

PostgreSQL の起動と停止

OSS-DB

- 起動 $ pg_ctl start
- 停止 $ pg_ctl stop
  - 緊急停止 $ pg_ctl stop -m fast
  - 即時停止 $ pg_ctl stop -m immediate
- 再起動 $ pg_ctl restart
- 構成ファイルの再読込み $ pg_ctl reload
- 状態表示 $ pg_ctl status

標準の対話的ターミナル（コマンドラインツール）

- psql [オプション] . .　[DB 名 [ユーザ名]]

接続オプション

-h, --hostname	PGHOST	UNIXドメインソケット
-p, --port	PGPORT	5432
-U, --username	PGUSER	DBユーザー名
-d, --dbname	PGDATABASE	DB名

プロンプト

- 「=#」… 接続しているユーザはスーパーユーザ
- 「=>」… 接続しているユーザは一般ユーザ

psql の基本的なメタコマンド

OSS-DB

- 終了 => ¥q
- データベースの一覧表示 => ¥l
- テーブルなどの一覧表示 => ¥d
  - d[t|i|s|v|S] …
    - テーブル | インデックス | シーケンス | ビュー | システムテーブル
- テーブルなどの情報表示 => ¥dテーブル名
- ロール、ユーザーの一覧表示 => ¥du
- テーブルなどのアクセス権限の一覧表示 => ¥dp
- 関数の一覧表示 => ¥df

psql の基本的なメタコマンド

OSS-DB

- テーブルの表示モード変更 => ¥x
- ファイルのスクリプト実行 => ¥i ファイル名
- データベースの接続 => ¥c データベース名
- テーブルデータのコピー => ¥copy
- SQL ヘルプ表示 => ¥h [SQL コマンド]
- メタコマンド・ヘルプ表示 => ¥?
- OS コマンドの実行 => ¥![コマンド]
- クエリ実行時間表示のon/off => ¥timing [on|off]

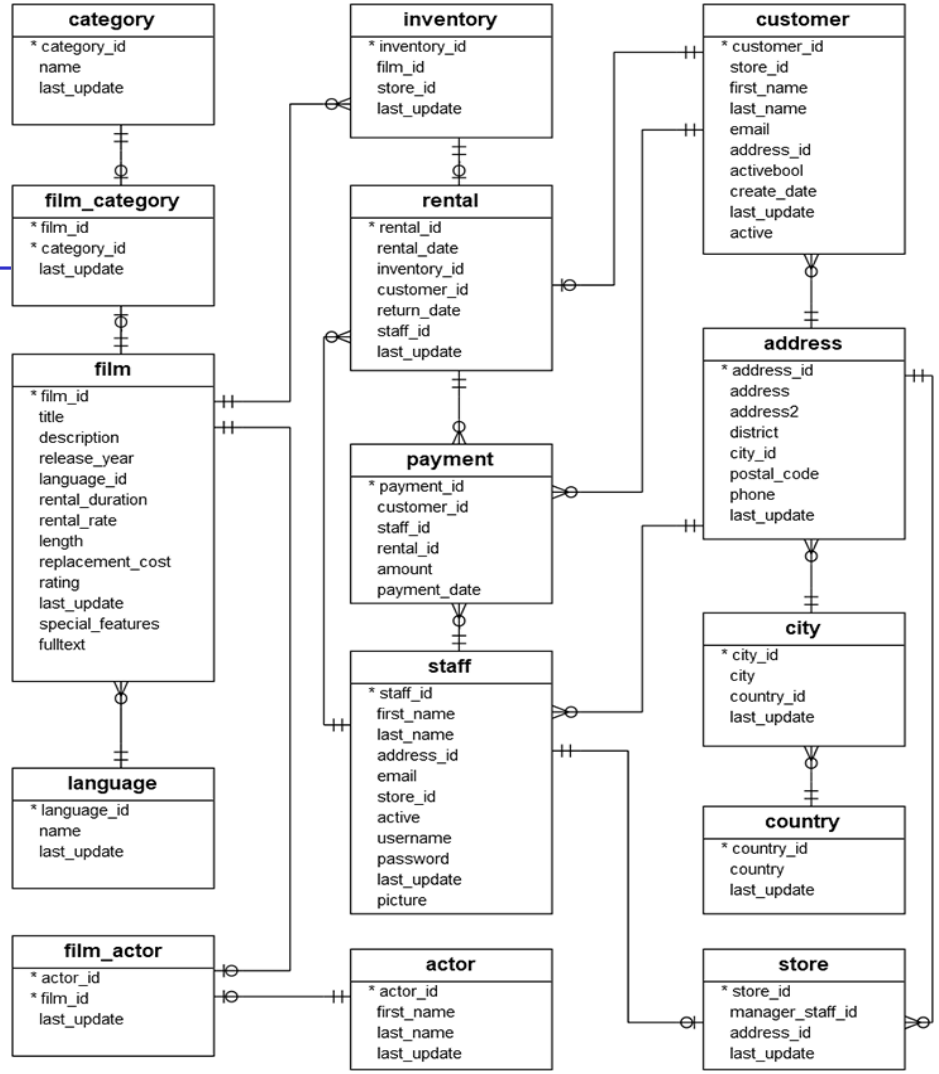
スクリプトの利用

OSS-DB

- psql コマンド実行時にリダイレクト
- psql ターミナルの、"¥i ファイル名" コマンドで実行
- psql 起動時に、"-e" オプションをつけると、スクリプトの SQL 文が、ターミナルに表示
  - 起動後につぎのメタコマンドで設定も可
    - ¥set ECHO all
    - ¥set ECHO_HIDDEN
- コメントの扱い
  - “--” コメントは、ターミナルに表示されない
  - "/*... */" コメントは、ターミナルに表示

サンプルDBの設定

- DVDレンタル
- PostgreSQL Sample Database
  - http://www.postgresqltutorial.com/postgresql-sample-database/
  - dvdrental.zip
- 解凍後、スクリプトを実行
  - restore.sql
  - 文字列 “$$PATH$$” を実際のパスに変更



PostgreSQL の標準添付ツール

OSS-DB

postmaster / postgres	サーバー・プロセス
psql	対話型の SQL 文実行ツール
pg_ctl	PostgreSQL の起動 / 停止
createdb / dropdb	データベースの追加 / 削除コマンド
createlang / droplang	PostgreSQL で使う手続き言語の定義 / 削除コマンド
createuser / dropuser	データベースのユーザーの作成 / 削除コマンド
vacuumdb	データベースに対する VACUUM 処理
libpq	PostgreSQL に接続するクライアントのためのライブラリー

PostgreSQL の設定ファイル

OSS-DB

- 変更の反映
  - リロード$ pg_ctl reload / 再起動 $ pg_ctl restart
- 実行時に設定値を変更
  - セッション中でSET文を実行
    - 一部のパラメータのみ変更可能
    - 影響範囲は実行したセッションのみ
  - SET パラメータ = {TO 値|'値'|DEFAULT};
    - 数値や論理値、列挙値でない場合はシングルクォートで囲む
    - 「DEFAULT」を指定するとデフォルト値に戻る

構成ファイル postgresql.conf

OSS-DB

- 最大接続数、ポート番号、ログ保存方式などの基本的な設定
- デフォルトでは $PGDATA/postgresql.conf
- 変数 = 値
  - データベースの実行時の設定項目
- SHOW / SETで閲覧・設定
  - SHOW 変数名;
  - SET 変数名 = 値;
- 変数によって、変更するタイミングが違う
  - 実行時、reload時、再起動時

postgresql.conf 接続・認証関係

OSS-DB

- listen_address = ‘localhost’
  - クライアント接続を監視するTCP/IPアドレス
  - リモート接続にする → ‘*’
- max_connections = 100
  - 同時接続可能なクライアント数

postgresql.conf クライアント関係

OSS-DB

- search_path = ‘”$user”, public’
  - スキーマ検索パス
- timezone = ‘Japan’
  - タイムスタンプ解釈用の時間帯
- client_encoding = sql_ascii
  - クライアント用の符号化方式(文字セット)を指定
  - LinuxのデフォルトはDBの符号化方式と同じ

postgresql.conf ログ関係

OSS-DB

- log_destination = ‘stderr’
- logging_collector = on
  - stderr/csvlog で出力されたログをリダイレクト
- log_filename = ‘postgresql-%a.log’
- log_connections = ‘off’
  - クライアントの接続認証をログに出力
- log_line_prefix = ‘< %m >’
  - 各ログ行の先頭に出力する書式文字列

クライアント認証ファイル pg_hba.conf

OSS-DB

- データベースに対するアクセス制限を設定
- デフォルトでは $PGDATA/pg_hba.conf
- TYPE　　接続方式
- DATABASE　　アクセス制限をかけるデータベース名
- USER　　アクセス制限をかけるユーザー名
- ADDRESS　　クライアントのIPアドレス
- METHOD　　データベースのアクセス認証方式
  peer, md5, ident, trust, reject など

■SQL (Structured Query Language)

OSS-DB

- データベース・オブジェクトに対する定義や変更、データ操作を行うための言語
- ANSI、ISO で標準化が進められている
- DBMSに依存しないが、一部独自拡張がある
  - PostgreSQLはドキュメントに標準・独自拡張を明記
- 「非手続き言語」とも呼ばれ、データを取り出す手順を考慮せず、何をしたいかのみを指定すればよい
- SQL 文を記述する際に、キーワードや名前は大文字・小文字を問わない
  - PostgreSQLの内部では小文字で扱う

SQLの種類

OSS-DB

- DDL: Data Definition Language（データ定義言語）
  - テーブルやインデックスの作成・変更・削除など
    - CREATE, ALTER, DROP
- DML: Data Manipulation Language（データ操作言語）
  - データの追加・検索・更新・削除など
    - INSERT, SELECT, UPDATE, DELETE
- DCL: Data Control Language（データ制御言語）
  - データのアクセス権設定・トランザクション制御など
    - GRANT, REVOKE
    - START TRANSACTION, COMMIT, ROLLBACK

NULL値

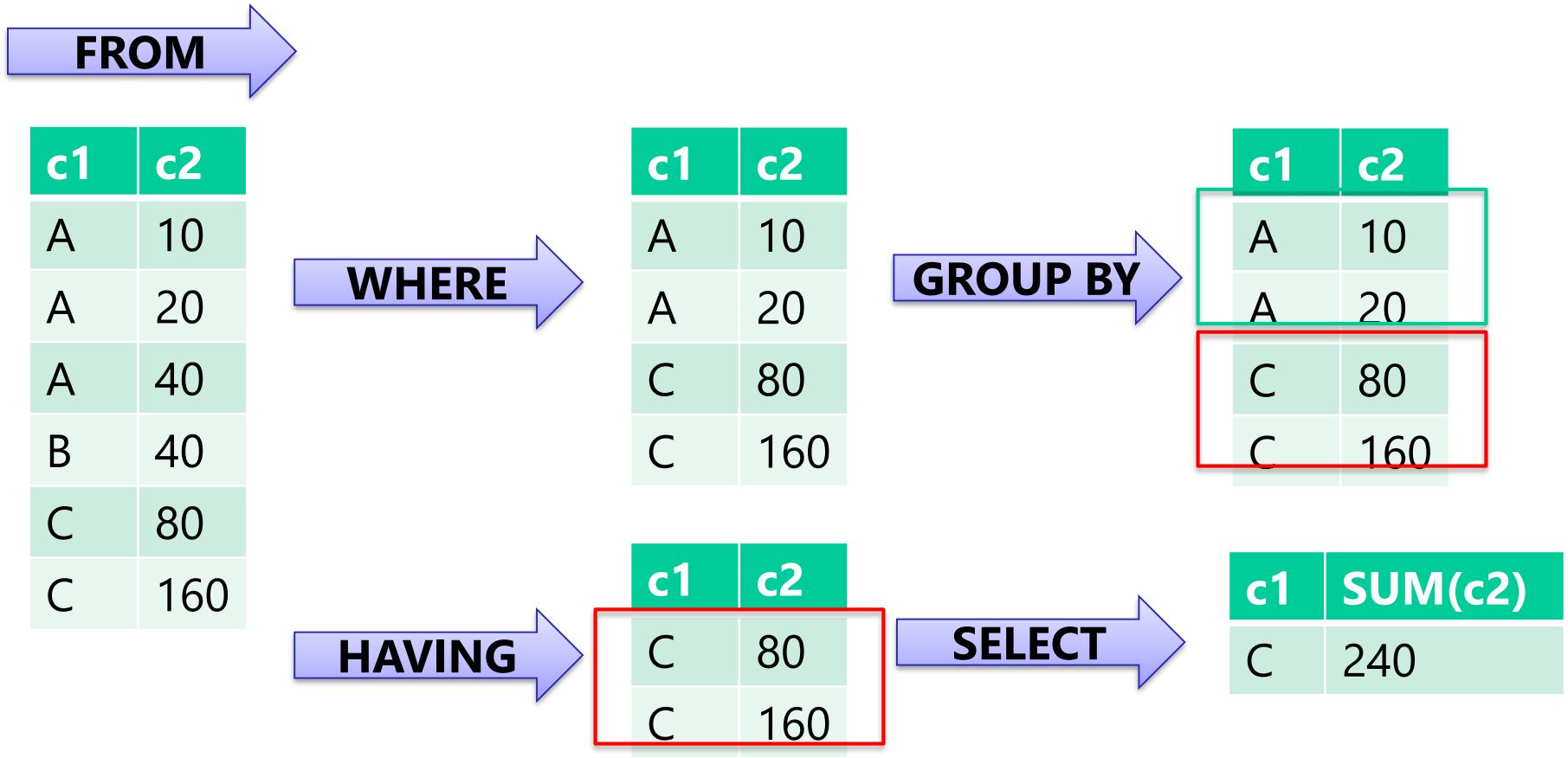
OSS-DB

- 「値不定」を表す特殊な値で、空文字列とは区別
- 一致、大小比較、文字列連結などのNULL値との演算の結果はNULL
- NULLであるか？ の判断は専用の演算子がある
  - 式 IS [NOT] NULL
- 多用しない方がよい
  - CREATE TABLE でのデフォルトはNULLを許すが、必要ない限り、NOT NULL制約をつける
- psqlでの表示
  - ¥pset null ‘(NULL)’とすると、(NULL) と表示

SELECT文

SELECT 句 ... 何を出力するのか
    FROM 句 ... どのテーブルから出力するのか
    WHERE 句 ... 行の選択条件
    GROUP BY 句 ... グループのキー
    HAVING 句 ... グループの選択条件
UNION | EXCEPT | INTERSECT .. テーブルの合併|差|積
SELECT 句 ... 何を出力するのか
    ...
ORDER BY 句 ... 結果のソート
LIMIT 句 ... 表示する行数
OFFSET 句 ... スキップする行数

SELECT文の評価順



テーブルの結合

OSS-DB

■A

a
1
2

■B

b
1
3

■A JOIN B ON a = b

a	b
1	1

■A FULL JOIN B ON a = b

a	b
1	1
2	NULL
NULL	3

■A LEFT JOIN B ON a = b

a	b
1	1
2	NULL

■A RIGHT JOIN B ON a = b

a	b
1	1
NULL	3

和結合、差結合、積結合

■A

a
1
2

■B

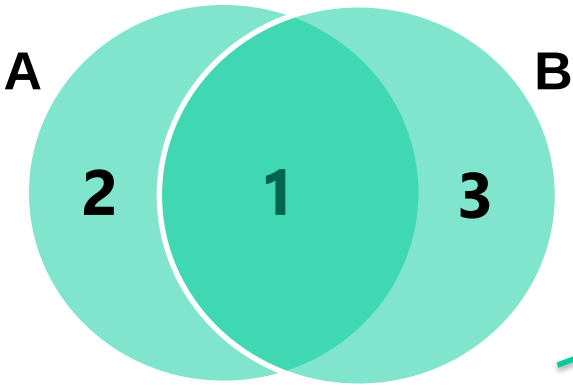
b
1
3

SELECT * FROM A UNION SELECT * FROM B

a
1
2
3

SELECT * FROM A UNION ALL SELECT * FROM B

a
1
1
2
3



SELECT * FROM A EXCEPT SELECT * FROM B

a
2

SELECT * FROM A INTERSECT SELECT * FROM B

a
1

副照会

SQL 文の中に SELECT 文を書く

副照会の結果表

- 単一行、単一列の結果表(スカラー値)
  - 通常の関係演算子
- 単一行、複数列の結果表(レコード値)
  - (x, y, z) = (a, b, c)
- 複数行、単一列の結果表(ベクター値)
  - つぎのキーワードとあわせて指定
    - IN, ANY, SOME, ALL
- 結果表がある / ない (0 行)
  - EXISTS

■演算子と関数

OSS-DB

- ■演算子
- ■スカラー関数
- ■集約関数(列関数)
- ■集合を返す関数
  - 戻り値が複数行

演算子

OSS-DB

- 型変換
  - cast(), ::
- 関係演算
  - =, != (<>), <, <=, >, >=,
  - BETWEEN, IS [NOT] NULL
- 算術演算　+, -, *, /, %(剰余), ^(べき乗), @(絶対値)
- 文字列演算
  - || (文字列結合)
  - LIKE, SIMILER TO, ~ (POSIX正規表現)

演算子

日付・時刻演算

- [日付|時刻] [+|-] [日付|時刻] = インターバル
- [日付|時刻] [+|-] インターバル = [日付|時刻]
- インターバル [*|/] 数値 = インターバル

インターバル

- (例) interval ‘2 month 3 day’
- YEAR, MONTH, DAY
- HOUR, MINUTE, SECOND

…

スカラー関数

OSS-DB

- NULL変換　　coalesce()
- 算術関数
- 文字列関数
- フォーマッティング関数
  - to_char(),　to_number(),
  - to_date(),　to_timestamp()
- 日付 / 時刻関数
  - current_date, current_time, current_timestamp
  - age(),　extract()

集約関数(列関数)

c
10
10
20
30
NULL
NULL

列名	DISTINCT	*
COUNT(c) = 4	COUNT(DISTINCT c) = 3	COUNT(*) = 6
SUM(c) = 70	SUM(DISTINCT c) = 60	
AVG(c) = 17.5	AVG(DISTINCT c) = 20	
MIN(c) = 10	MIN(DISTINCT c) = 10	
MAX(c) = 30	MAX(DISTINCT c) = 30	

■トランザクション

OSS-DB

- 複数の手順を単一の「すべてかなしか」の操作にまとめる
- 手順の進行途中の状態はほかの動いているトランザクションからは見えない
- 開始 START TRANSACTION / BEGIN
- 終了(確定) COMMIT / END
- 終了(取消し) ROLLBACK / ABORT
- psqlは自動コミット
- アプリケーションでは、各APIの仕様による
- トランザクションの途中で部分的にロールバックしたいときは、SAVEPOINTを使う(部分COMMIT不可)

トランザクションの特性

OSS-DB

A	原子性 (Atomicity)	トランザクション処理がすべて完了かまったく実行されていないかで終了すること
C	一貫性、同時性 (Concurrency)	処理の順番に関わらず結果が同じになる
I	分離性、隔離性、独立性 (Isolation)	中間結果は、ほかのトランザクションの処理内容に何の影響も与えない
D	耐久性 (Durability)	いったんトランザクションが完結したら障害が発生してもデータの状態が変化しない

トランザクション

OSS-DB

SQL文	パターン1	パターン2	パターン3
START TRANSACTION	OK	OK	OK
UPDATE	OK	OK	OK
UPDATE	OK	OK	OK
SELECT	OK	**NG**	OK
COMMIT/ ROLLBACK	ROLLBACK	**COMMIT**	COMMIT
結果	ROLLBACK	**ROLLBACK**	COMMIT

分離レベル

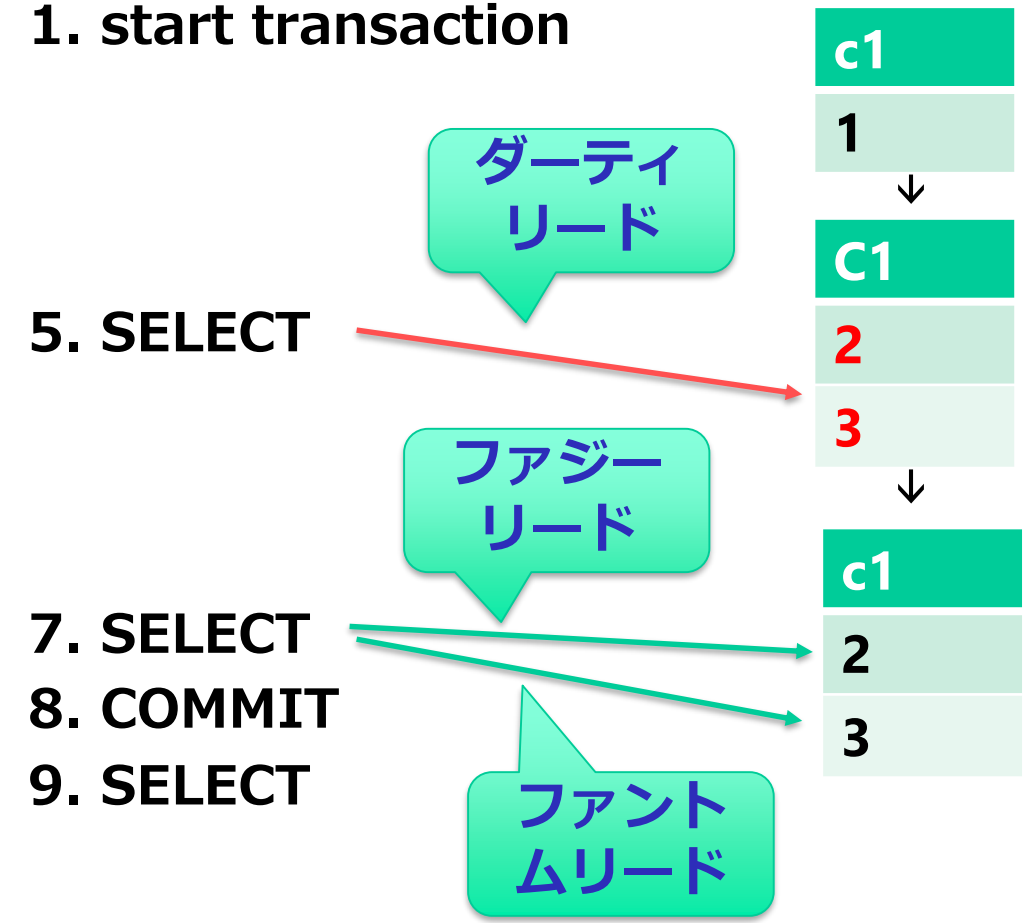
OSS-DB

分離レベル	ダーティ・リード	ファジー・リード	ファントム・リード
Read uncommitted	可能性あり	可能性あり	可能性あり
Read committed	安全	可能性あり	可能性あり
Repeatable read	安全	安全	可能性あり
Serializable	安全	安全	安全

- ダーティ・リード
  - 他のトランザクションによる未コミットの挿入/更新/削除結果が見える
- ファジー・リード (反復不能読み取り 、ノン・リピータブル・リード)
  - 他のトランザクションによるコミット済みの更新/削除結果が見える
- ファントム・リード
  - 他のトランザクションによるコミット済みの挿入結果が見える

分離レベル

1. start transaction



2. start transaction
3. INSERT … VALUES(3)
4. UPDATE … c1 = 2
   where c1 = 1
6. COMMIT

■データベースオブジェクト

OSS-DB

- ■データベース
- ■スキーマ
- ■ロール
- ■テーブル
- ■ビュー
- ■シーケンス
- ■ユーザー定義関数

- ■インデックス
  - データの検索を高速にする仕組み
- ■トリガー
  - テーブルに対してイベントが発生したときに指定した関数を実行するしくみ

データベース

OSS-DB

- データベース一覧
  - psql –l、　　¥l
- SQLコマンド
  - CREATE DATABASE　データベース名
  - ALTER DATABASE　データベース名 {SET | RESET }
  - DROP DATABASE　データベース名
- OSコマンド
  - createdb　データベース名
  - dropdb　データベース名

スキーマ

OSS-DB

- データベースオブジェクトが所属する名前空間
  - スキーマが異なれば同じ名前のオブジェクトも可能
  - スキーマ名.オブジェクト名
  - ひとつのデータベースに複数作成可能（ネストは不可）
  - デフォルトで「public」というスキーマがある
- 作成
  - CREATE SCHEMA スキーマ名;
- スキーマの探索パス
  - search_path = ‘“$user”,public’ パラメータ
  - SHOW search_path; で確認

システムカタログ/情報スキーマ

OSS-DB

- システムカタログ(pg_*)
  - DBの内部情報を格納するテーブルやビューの集合
  - PostgreSQL独自情報も含む
  - 移植性なし
- 情報スキーマ(information_schema.*)
  - DBで定義されたオブジェクトの情報を持つビューの集合
  - 標準SQLに準拠
  - PostgreSQL独自情報なし
  - 移植性高い

ロール / ユーザー

OSS-DB

- PostgreSQL データベースにアクセスできるユーザー
  - OS のユーザーとは別
  - データベースクラスタ初期化時にスーパーユーザpostgres作成済み
- PostgreSQL のユーザーの種類
  - 一般、DB 作成権限、スーパーユーザー (DB作成+ユーザー作成)
- ロールは、ユーザー+グループ の概念
- コマンド
  - #> CREATE ROLE ユーザー名 PASSWORD ‘password’
    [CREATEDB] [CREATEUSER] …
  - $ createuser ユーザー名

権限

- 特定のアクションを実行するために必要な権利
- 権限の種類
  - ロール属性としての権限
    - データベースクラスタで管理される対象に対する権限
    - ALTER ROLEコマンドなどで指定
    - ¥du, ¥du+ で確認
  - アクセス権限
    - データベースに存在するある特定のオブジェクトに対する権限
    - GRANT / REVOKE コマンドで指定
    - ¥dp で確認

アクセス権限

OSS-DB

権限	内容
SELECT	SELECT文、COPY TO文の実行
INSERT	INSERT文、COPY FROM文の実行
UPDATE	UPDATE文、SELECT FOR { UPDATE | SHARE }文の実行 DELETE 文の実行
REFERENCES	外部キー制約の作成
TRIGGER	トリガの作成
CREATE	スキーマ作成、オブジェクト作成
CONNECT	データベースへの接続
EXECUTE	指定の関数、演算子の実行

テーブル

データを格納する容器

- 複数の列からなり、各列に独立した値を格納
- 複数の行を保持できる

```
CREATE TABLE テーブル名 (
  列名 データ型 列制約 [, 列名 データ型 …]
  [, 表制約 … ]);
```

データ型

- 数値型
- 文字列型
- バイナリー列型
- 日付 / 時刻型
- 論理値型 （BOOLEAN）
- 擬似データ型
- OID
- ラージ・オブジェクト
- 配列

など

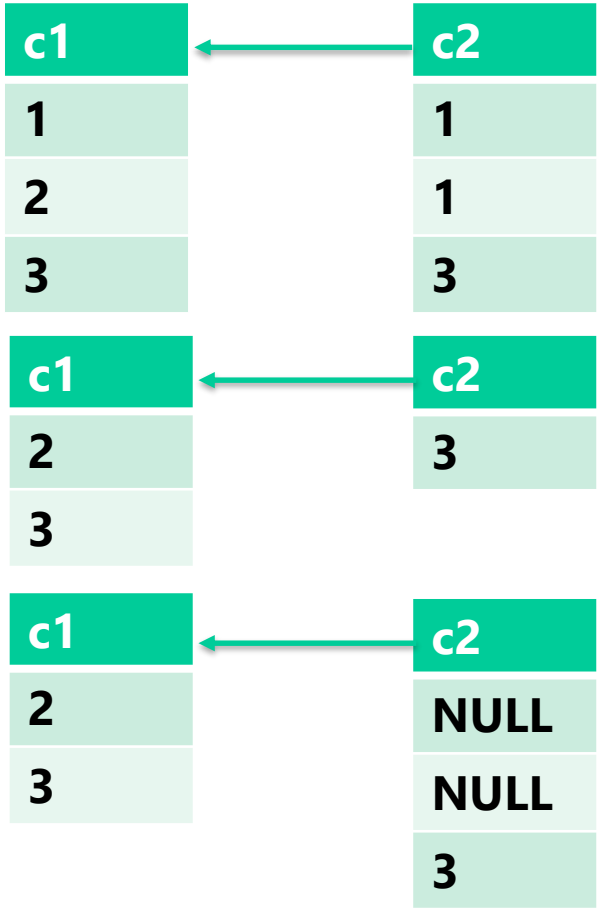
制約

OSS-DB

NOT NULL	列が NULL 値をとらないことを指定	NOT NULL
一意性	列や列のグループに含まれるデータが、テーブル内のすべての行で一意 暗黙でインデックスを作成	UNIQUE(列名, ...)
主キー	一意性制約と NOT-NULL 制約の組み合わせで、テーブルにひとつだけ 暗黙でインデックスを作成	PRIMARY KEY(列名, ...)
検査	特定の列の値が任意の条件式を満たす	CHECK(条件式)
外部キー	列や列のグループの値が、他のテーブルの行の値と一致しなければならない	FOREIGN KEY(列名, ...) REFERENCES 参照先テーブル名(列名, ...)

外部キー制約のACTION

- DELETE … WHERE c1 = 1
  - NO ACTION / RESTRICT
    - -> 失敗
  - CASCADE
    - -> 成功
  - SET NULL / SET DEFAULT
    - -> 成功



ALTER / DROP TABLE

OSS-DB

テーブル定義の変更

- ALTER TABLE
  - RENAME TO 新しい名前
  - RENAME COLUMN 列名 TO 新しい名前
  - ALTER COLUMN 列名
  - ALTER COLUMN 列名 SET DATA TYPE 新しいデータ型
  - ADD COLUMN 列名 データ型 制約
  - DROP COLUMN 列名

テーブルの削除

- DROP TABLE テーブル名

ビュー

OSS-DB

- SELECT文に名前をつけて、テーブルのように使えるようにしたもの
  - 基本的に読み取り専用
- 目的
  - 複雑な照会の単純化
  - セキュリティ
  - ビューの実装（具体的な検索方法）の隠蔽
  - アプリケーションとデータの独立
- 定義
  - CREATE VIEW ビュー名 AS SELECT …

シーケンス

OSS-DB

数列生成器

- 高速
- トランザクションをロールバックしても一度使用した値は戻らないので、完全な連番にはならない
- serial型、bigserial型はシーケンスを使って実現

定義

- CREATE SEQUENCE シーケンス名 [INCREMENT [BY] 増分] [START [WITH] 初期値];

シーケンスの操作関数

- 現在値の取得（進まない）　currval('シーケンス名');
- 次の値の取得（進む）　nextval('シーケンス名');
- 次の値の設定　setval('シーケンス名', 次の値);

ユーザー定義関数

OSS-DB

- ユーザーが、自分で関数を定義して使うことができる
- 関数の定義は、"CREATE FUNCTION" 文
- 関数の定義で使う言語は、LANGUAGE 句で指定

```
CREATE FUNCTION 関数名 ([ 引数の型 [, ...] ])
    RETURNS 戻り値の型
    AS $$関数の定義$$
    LANGUAGE '言語'
```

- 関数の一覧表示
  - ¥df

■バックアップとリカバリ

OSS-DB

- pg_dump / pg_dumpall コマンド
  - データベース単位 /データベースクラスタ全体
- PITR（Point In Time Recovery）
  - 任意の時点にリカバリ可能
- COPY 文、¥copy メタコマンド
  - テーブル単位でテキスト/CSV形式ファイルの入出力
- コールドバックアップ(ディレクトリコピー)
  - OSのコピー、アーカイブ用コマンドを使う
  - データベースの停止が必要

pg_dump/pg_dumpall

OSS-DB

- インスタンス稼働状態で論理バックアップを取得
- データはバックアップ開始時点のもので、一貫性あり
  - バックアップした時点にのみリカバリ可能
- pg_dump　　データベース単位
- pg_dumpall　データベースクラスタ全体
- 設定ファイルは別途バックアップ必要
- バックアップ取得元とリカバリ先でメジャーバージョンが異なっていてもリカバリ可能

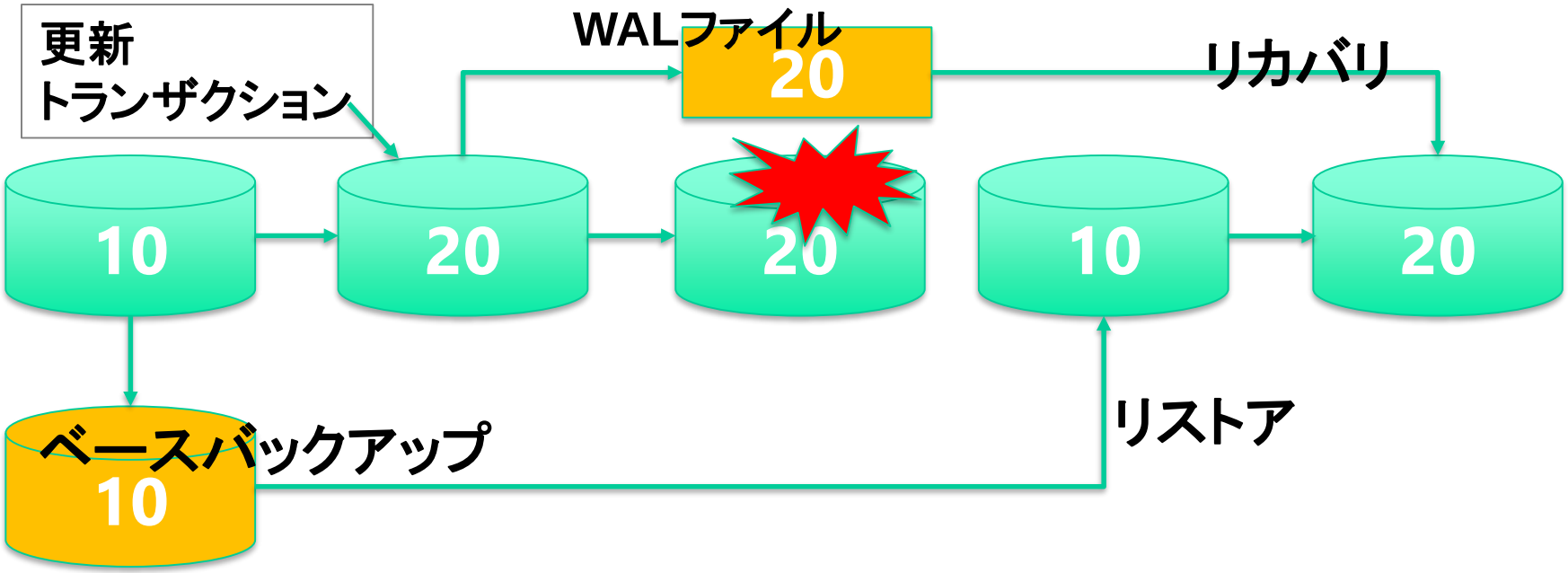
pg_dump/pg_dumpall

OSS-DB

- pg_dump のリカバリ
  - psql データベース名 < ダンプファイル名
  - pg_restore -d データベース名 ダンプファイル名
- pg_dumpall のリカバリ
  - psql -f ダンプファイル名 postgres

PITR（Point In Time Recovery）

- ベースバックアップとアーカイブログを使ってデータベースを最新の状態までリストアする手法
- 障害の直前の状態までデータを復旧(リカバリ)できる



¥copy メタコマンド （ファイル入出力）

OSS-DB

- クライアントマシン(psqlを実行しているマシン）上のファイルシステムにファイル出力（入力）する
  - デフォルトの形式はタブ区切りのテキストファイル
  - オプションに"csv"と指定すれば、CSVファイル
- エクスポート
  - ¥copy テーブル名 to ファイル名 [その他オプション]
- インポート
  - ¥copy テーブル名 from ファイル名 [その他オプション]

■メンテナンス

OSS-DB

■DMLにより生じた不要領域の回収

- データファイルの肥大化を抑制
- 8.3以降は自動バキュームがデフォルトでON (autovacuum = on)
- =# VACUUM オプション テーブル名
- $ vacuumdb 接続オプション オプション DB名

■プランナ統計情報の収集

- プランナ統計情報を最新に更新
- プランナ統計情報が古いと、SQLの処理パフォーマンスが適切にならない場合がある
  - SQLの処理方法（プラン）は統計情報をもとに決定するため
- =# ANALYZE テーブル名;

■まとめ(本セミナーのゴール)

OSS-DB

- 関係型データベースや PostgreSQL に関する基本的な知識を理解する
- PostgreSQL を使って自分自身で試しながらデータベースを学習する上で、基本的な操作方法を理解する

参考文献・教材

Webサイト

- PostgreSQL 最新版ドキュメント（日本語版）
  - https://www.postgresql.jp/document/current/index.html
  - 過去のセミナー資料
  - https://oss-db.jp/

オープンソースデータベース標準教科書 – PostgreSQL –

- つぎからダウンロード可
  - https://oss-db.jp/ossdbtext
- 製本版（書籍）の購入も可

ご清聴ありがとうございました。

■お問い合わせ■
株式会社リカレント リナックスアカデミー
〒160-0022　東京都新宿区新宿3-1-13
TEL. 03-5368-3889　FAX. 03-5368-3881