

EDBウェビナー 2020年2月19日

EDB Postgres Advanced Server 12 新機能について

EDB技術本部長 高鶴勝治



概要 & インストール



EDB Postgres Advanced Server 12

- 2019年12月リリース

- サポート・プラットフォーム

Linux x86-64: RHEL / CentOS / OEL 7.x & 6.x, SLES 12, Ubuntu 18.04+, Debian 9.6.x

Windows x86-64: Windows2019, Windows2016, Windows 2012 R2

Linux on IBM Power 8/9 Little Endian (LE): RHEL 7.x

Kubernetes

*下記URLを参照ください

<https://www.enterprisedb.com/services/edb-supported-products-and-platforms>

- サポート期間: GAより**5年間**が標準サポート期間 (2024年12月)

- インストール方法

- Linux RPM * YUMリポジトリ・ログイン用のユーザID/パスワードが必要

- Windows <https://www.enterprisedb.com/software-downloads-postgres>

- アーキテクチャ

- **PostgreSQL12をベースに EDBがエンハンス**

- プロセス構成、データベース構造、設計、チューニング、運用管理 ⇨ PostgreSQL資料が参考になります

- SQL ⇨ PostgreSQLの構文 & Oracle構文

EDB Postgres Advanced Server 12

[RHEL7/CentOS7 インストール 手順]

1. Yum リポジトリ・ファイルの編集

- ❑ `yum install -y https://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.rpm`
- ❑ `/etc/yum.repos.d/edb.repo` 上の、リポジトリ “[edb]” において、`enable`を1に設定
- ❑ リポジトリキーのインストール

2. EPAS12インストール

- ❑ `yum -y install epel-release`
- ❑ `yum -y install edb-as12-server`

3. DBクラスタ作成 & サービス生成

- ❑ `PGSETUP_INITDB_OPTIONS="-E UTF-8" /usr/edb/as12/bin/edb-as-12-setup initdb`
- ❑ `systemctl start edb-as-12`
- ❑ `systemctl status edb-as-12`
- ❑ `systemctl enable edb-as-12`

4. enterprisedbユーザのパスワード設定

5. `pg_hba.conf` の編集 & RELOAD

* 詳細は、「[EDB_Postgres_Advanced_Server_Installation_Guide_Linux_v12.pdf](#)」をご参照

EDB Postgres Advanced Server 12

[EPAS12 でインストールされるもの]

Package	アーキテクチャー	バージョン	リポジトリ	容量
インストール中:				
edb-as12-server	x86_64	12.0.1-1.rhel7	edb-testing	5.7 k
依存性関連でのインストールをします:				
edb-as12-libicu	x86_64	53.1-1.rhel7	edb-testing	7.4 M
edb-as12-pgagent	x86_64	4.0.0-1.rhel7	edb-testing	174 k
edb-as12-server-client	x86_64	12.0.1-1.rhel7	edb-testing	1.4 M
edb-as12-server-cloneschema	x86_64	1.10-1.rhel7	edb-testing	23 k
edb-as12-server-contrib	x86_64	12.0.1-1.rhel7	edb-testing	660 k
edb-as12-server-core	x86_64	12.0.1-1.rhel7	edb-testing	5.7 M
edb-as12-server-devel	x86_64	12.0.1-1.rhel7	edb-testing	2.5 M
edb-as12-server-docs	x86_64	12.0.1-1.rhel7	edb-testing	14 k
edb-as12-server-indexadvisor	x86_64	12.0.1-1.rhel7	edb-testing	57 k
edb-as12-server-libs	x86_64	12.0.1-1.rhel7	edb-testing	611 k
edb-as12-server-llvmjit	x86_64	12.0.1-1.rhel7	edb-testing	11 M
edb-as12-server-parallel-clone	x86_64	1.5-1.rhel7	edb-testing	70 k
edb-as12-server-pldebugger	x86_64	1.1-1.rhel7	edb-testing	90 k
edb-as12-server-plperl	x86_64	12.0.1-1.rhel7	edb-testing	57 k
edb-as12-server-plpython	x86_64	12.0.1-1.rhel7	edb-testing	78 k
edb-as12-server-pltcl	x86_64	12.0.1-1.rhel7	edb-testing	38 k
edb-as12-server-sqlprofiler	x86_64	4.0-1.rhel7	edb-testing	88 k
edb-as12-server-sqlprotect	x86_64	12.0.1-1.rhel7	edb-testing	102 k
edb-as12-server-sslutils	x86_64	1.2-1.rhel7	edb-testing	43 k
llvm5.0	x86_64	5.0.1-7.el7	epel	2.6 M
llvm5.0-libs	x86_64	5.0.1-7.el7	epel	13 M

EDB Postgres Advanced Server 12

[EPAS12 のプロセス]

```
[root@epassrv03 data]# systemctl status edb-as-12
```

```
● edb-as-12.service - EDB Postgres Advanced Server 12
```

```
Loaded: loaded (/usr/lib/systemd/system/edb-as-12.service; disabled; vendor preset: disabled)
```

```
Active: active (running) since 金 2019-08-16 11:32:17 JST; 2s ago
```

```
Process: 11680 ExecStartPre=/usr/edb/as12/bin/edb-as-12-check-db-dir ${PGDATA} (code=exited, status=0/SUCCESS)
```

```
Main PID: 11688 (edb-postmaster)
```

```
Tasks: 9
```

```
CGroup: /system.slice/edb-as-12.service
```

```
├─11688 /usr/edb/as12/bin/edb-postmaster -D /var/lib/edb/as12/data
├─11692 postgres: logger
├─11694 postgres: checkpointer
├─11695 postgres: background writer
├─11696 postgres: walwriter
├─11697 postgres: autovacuum launcher
├─11698 postgres: stats collector
├─11699 postgres: dbms_aq launcher
└─11700 postgres: logical replication launcher
```

```
8月 16 11:32:17 epassrv03.enterprisedb.com systemd[1]: Starting EDB Postgres Advanced Server 12...
```

```
8月 16 11:32:17 epassrv03.enterprisedb.com edb-postmaster[11688]: 2019-08-16 11:32:17 JST LOG: starting PostgreSQL 12.0 (EnterpriseDB Advanced ...64-bit
```

```
8月 16 11:32:17 epassrv03.enterprisedb.com edb-postmaster[11688]: 2019-08-16 11:32:17 JST LOG: IPv4アドレス"0.0.0.0"、ポート5446で待ち受けています
```

```
8月 16 11:32:17 epassrv03.enterprisedb.com edb-postmaster[11688]: 2019-08-16 11:32:17 JST LOG: IPv6アドレス "::"、ポート5446で待ち受けています
```

```
8月 16 11:32:17 epassrv03.enterprisedb.com edb-postmaster[11688]: 2019-08-16 11:32:17 JST LOG: Unixソケット"/tmp/.s.PGSQL.5446"で待ち受けています
```

```
8月 16 11:32:17 epassrv03.enterprisedb.com edb-postmaster[11688]: 2019-08-16 11:32:17 JST LOG: ログ出力をログ収集プロセスにリダイレクトしています
```

```
8月 16 11:32:17 epassrv03.enterprisedb.com edb-postmaster[11688]: 2019-08-16 11:32:17 JST ヒント: ここからのログ出力はディレクトリ"log"に現れます。
```

```
8月 16 11:32:17 epassrv03.enterprisedb.com systemd[1]: Started EDB Postgres Advanced Server 12.
```

```
Hint: Some lines were ellipsized, use -l to show in full.
```

EPAS12の新機能



EDB Postgres Advanced Server 12 新機能

■ PostgreSQL12

- パーティション・テーブルの改善
 - パーティション・テーブルに対する参照整合性制約
 - Fast Run -Time Pruning
 - Attach partition w/ShareUpdateExclusiveLock
 - パーティション・テーブルへの高速COPY
- インデックス
 - B-Tree インデックスのよりサイズ縮小
 - REINDEX CONCURRENTLY
- SQL
 - 生成列のサポート
 - CTE のパフォーマンス向上
- 運用管理
 - pg_checksum
 - pg_stat_progress_cluster/pg_stat_progress_create_index
 - recovery.confの廃止 (postgresql.confへの統合)
- プラガバブル・ストレージ・エンジン

■ EPAS v12

- Oracle互換性
 - ✓ インターバル・パーティション
 - ✓ 複合トリガー
 - ✓ 集計関数の追加(LISTAGG/MEDIAN)
 - ✓ CAST関数のコレクション対応
 - ✓ SYS_GUIDのサポート
 - ✓ 追加されたデータ・ディクショナリ・ビュー
 - ✓ SELECT UNIQUE
 - ✓ PG12のOID廃止に伴うROWIDの挙動変更
- スタンバイDBのロジカル・レプリケーション・スロットのサポート

その他多数

パーティション・テーブル



EPAS12のパーティション機能

- サポートされるパーティション・タイプ

- ①レンジ・パーティション

- 一つ以上のパーティショニング・キー列。二つの極値間の値に基づく

- ②リスト・パーティション

- シングル・パーティショニング・キー列。正確な値に基づく

- ③ハッシュ・パーティション

PPAS9.4 ~

- 指定されたカラム名のハッシュ値に基いて、データを均等に分割

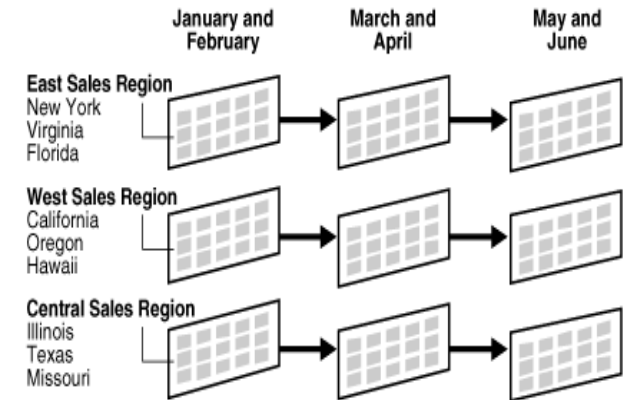
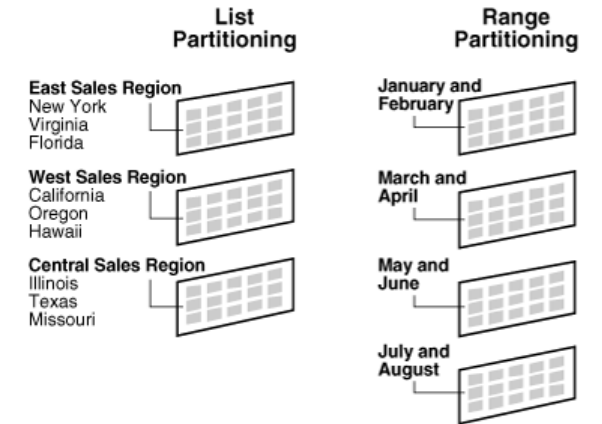
- ④インターバル・パーティション

EPAS12 ~

- 指定された時間間隔で、行挿入時に必要なパーティションを自動生成

- サブ・パーティション

- パーティション化されたテーブルより、より小さなサブセットへ分割可能
 - 親と異なる、パーティション・タイプを使用可能
 - 全てのデータは、サブ・パーティションに保管される



Images from http://docs.oracle.com/cd/E18283_01/server/e16541/partition.htm

レンジ・パーティション

-- MAXVALUEを持たないレンジ・パーティション 作成 --

```
CREATE TABLE employees_nomax(  
    empno numeric(8,0) check ( empno < 100000000)  
    ,ename varchar(32)  
    ,gender char(1)  
    ,birthday date  
    ,deptno number(2)  
    ,job varchar(9)  
    ,hiredate timestamp  
    ,postno char(7)  
    ,adress varchar2(1000)  
    ,telno varchar2(20)  
    ,mobileno varchar2(20)  
)  
PARTITION BY RANGE (hiredate)(  
    PARTITION employees_2001 VALUES LESS THAN('01-JAN-2002')  
    ,PARTITION employees_2004 VALUES LESS THAN('01-JAN-2005')  
    ,PARTITION employees_2007 VALUES LESS THAN('01-JAN-2008')  
    ,PARTITION employees_2010 VALUES LESS THAN('01-JAN-2011')  
    ,PARTITION employees_2013 VALUES LESS THAN('01-JAN-2014')  
    ,PARTITION employees_2014 VALUES LESS THAN('01-JAN-2015')  
    ,PARTITION employees_2015 VALUES LESS THAN('01-JAN-2016')  
    ,PARTITION employees_2016 VALUES LESS THAN('01-JAN-2017')  
);  
  
=====  
epas12=> insert into employees_nomax(empno, ename, job, hiredate ) values( 1 ,  
'Takatsuru' , 'CONSUL' , '01-JAN-1997');  
INSERT 0 1
```

```
epas12=> select * from employees_nomax;  
empno | ename | gender | birthday | deptno | job | hiredate | postno | adress |  
telno | mobileno  
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----  
1 | Takatsuru | | | | CONSUL | 01-JAN-97 00:00:00 | | | |  
(1 行)
```

```
epas12=> select * from employees_nomax_employees_2001;  
empno | ename | gender | birthday | deptno | job | hiredate | postno | adress |  
telno | mobileno  
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----  
1 | Takatsuru | | | | CONSUL | 01-JAN-97 00:00:00 | | | |  
(1 行)
```

```
=====  
epas12=> insert into employees_nomax(empno, ename, job, hiredate ) values( 2 , 'Hisano' ,  
'Developer' , '01-JAN-2019');  
psql: ERROR: 行に対応するパーティションがリレーション"employees_nomax"に見つかりません  
DETAIL: 失敗した行のパーティションキーは(hiredate) = (01-JAN-19 00:00:00)を含みます
```

• 該当のパーティションがない場合は、エラーとなる

レンジ・パーティション (続き)

-- MAXVALUEを持つレンジ・パーティション 作成 --

```
CREATE TABLE employees_nomax(  
  empno numeric(8,0) check ( empno < 100000000)  
  ,ename varchar(32)  
  ,gender char(1)  
  ,birthday date  
  ,deptno number(2)  
  ,job varchar(9)  
  ,hiredate timestamp  
  ,postno char(7)  
  ,adress varchar2(1000)  
  ,telno varchar2(20)  
  ,mobilenno varchar2(20)  
)  
PARTITION BY RANGE (hiredate)(  
  PARTITION employees_2001 VALUES LESS THAN('01-JAN-2002')  
  ,PARTITION employees_2004 VALUES LESS THAN('01-JAN-2005')  
  ,PARTITION employees_2007 VALUES LESS THAN('01-JAN-2008')  
  ,PARTITION employees_2010 VALUES LESS THAN('01-JAN-2011')  
  ,PARTITION employees_2013 VALUES LESS THAN('01-JAN-2014')  
  ,PARTITION employees_2014 VALUES LESS THAN('01-JAN-2015')  
  ,PARTITION employees_2015 VALUES LESS THAN('01-JAN-2016')  
  ,PARTITION employees_2016 VALUES LESS THAN('01-JAN-2017')  
  ,PARTITION employees_MAX VALUES LESS THAN(MAXVALUE)  
);  
  
=====  
epas12=> insert into employees_wmax(empno, ename, job, hiredate ) values( 1 ,  
'Takatsuru' , 'CONSUL' , '01-JAN-1997');  
INSERT 0 1
```

```
epas12=> insert into employees_wmax(empno, ename, job, hiredate ) values( 2 , 'Hisano' ,  
'Developer' , '01-JAN-2019');  
INSERT 0 1  
epas12=> insert into employees_wmax(empno, ename, job, hiredate ) values( 3 , 'Fujita' ,  
'President' , '01-JAN-2010');  
INSERT 0 1  
epas12=> insert into employees_wmax(empno, ename, job, hiredate ) values( 4 ,  
'Kitagawa' , 'Market' , '01-FEB-2018');  
INSERT 0 1  
epas12=> insert into employees_wmax(empno, ename, job, hiredate ) values( 5 ,  
'Makabe' , 'Sales' , '01-AUG-2018');  
INSERT 0 1
```

```
epas12=> select empno,ename, job,hiredate from employees_wmax;  
empno | ename | job | hiredate
```

```
-----+-----+-----+-----  
1 | Takatsuru | CONSUL | 01-JAN-97 00:00:00  
3 | Fujita | President | 01-JAN-10 00:00:00  
2 | Hisano | Developer | 01-JAN-19 00:00:00  
4 | Kitagawa | Market | 01-FEB-18 00:00:00  
5 | Makabe | Sales | 01-AUG-18 00:00:00  
(5 行)
```

```
epas12=> select empno,ename, job,hiredate from employees_wmax_employees_MAX;  
empno | ename | job | hiredate
```

```
-----+-----+-----+-----  
2 | Hisano | Developer | 01-JAN-19 00:00:00  
4 | Kitagawa | Market | 01-FEB-18 00:00:00  
5 | Makabe | Sales | 01-AUG-18 00:00:00  
(3 行)
```

- 該当のパーティションがなければ、MAXVALUEで定義されたパーティションに挿入される。
- 定期的にパーティションを追加するか、SPLITが必要

インターバル・パーティション

-- インターバル・パーティション 作成 --

```
CREATE TABLE employees_intval(
  empno numeric(8,0) check ( empno < 100000000)
  ,ename varchar(32)
  ,gender char(1)
  ,birthday date
  ,deptno number(2)
  ,job varchar(9)
  ,hiredate timestamp
  ,postno char(7)
  ,adress varchar2(1000)
  ,telno varchar2(20)
  ,mobilenno varchar2(20)
)
PARTITION BY RANGE (hiredate) interval(NUMTOYMINTERVAL(1,'year')) (
  PARTITION employees_2001 VALUES LESS THAN('01-JAN-2002')
  ,PARTITION employees_2004 VALUES LESS THAN('01-JAN-2003')
);

CREATE UNIQUE INDEX employees_intval_idx0 on employees_intval(hiredate,empno);
CREATE UNIQUE INDEX employees_intval_idx1 on employees_intval(mobilenno,hiredate);
CREATE INDEX employees_intval_idx2 on employees_intval(telno);
```

• 1年毎にパーティションを生成

epas12=> ¥d+ employees_intval

Partitioned table "public.employees_intval"

列	型	照合順序	Null 値を許容	デフォルト	ストレージ	統計の対象	説明
empno	numeric(8,0)				main		
ename	character varying(32)				extended		
gender	character(1)				extended		
birthday	timestamp without time zone				plain		
deptno	numeric(2,0)				main		
job	character varying(9)				extended		
hiredate	timestamp without time zone				plain		
postno	character(7)				extended		
adress	character varying(1000)				extended		
telno	character varying(20)				extended		
mobilenno	character varying(20)				extended		

パーティションキー: **RANGE (hiredate) INTERVAL ('1 year'::interval)**

インデックス:

"employees_intval_idx0" UNIQUE, btree (hiredate, empno)
 "employees_intval_idx1" UNIQUE, btree (mobilenno, hiredate)
 "employees_intval_idx2" btree (telno)

Check 制約:

"employees_intval_empno_check" CHECK (empno < 100000000::numeric)

パーティション: **employees_intval_employees_2001** FOR VALUES FROM (MINVALUE) TO ('01-JAN-02 00:00:00'),

employees_intval_employees_2004 FOR VALUES FROM ('01-JAN-02 00:00:00') TO ('01-JAN-03 00:00:00')

• インターバル・パーティション

• 初期状態では、CREATE時のものしかない

インターバル・パーティション (続き)

- データの挿入時、該当のパーティションがなければ、新たに、パーティションが作成される
- データが削除されても、パーティションは削除されない

```

epas12=> insert into employees_intval(empno, ename, job, hiredate ) values( 1 , 'Takatsuru' ,
'CONSUL' , '01-JAN-1997');
INSERT 0 1
epas12=> insert into employees_intval(empno, ename, job, hiredate ) values( 2 , 'Hisano' ,
'Developer' , '01-JAN-2019');
INSERT 0 1
epas12=> insert into employees_intval(empno, ename, job, hiredate ) values( 3 , 'Fujita' ,
'President' , '01-JAN-2010');
INSERT 0 1
epas12=> insert into employees_intval(empno, ename, job, hiredate ) values( 4 , 'Kitagawa' ,
'Market' , '01-FEB-2018');
INSERT 0 1
epas12=> insert into employees_intval(empno, ename, job, hiredate ) values( 5 , 'Makabe' , 'Sales' ,
'01-AUG-2018');
INSERT 0 1

```

```

epas12=> select empno,ename, job,hiredate from employees_intval;
empno | ename | job | hiredate
-----+-----+-----+-----
1 | Takatsuru | CONSUL | 01-JAN-97 00:00:00
3 | Fujita | President | 01-JAN-10 00:00:00
4 | Kitagawa | Market | 01-FEB-18 00:00:00
5 | Makabe | Sales | 01-AUG-18 00:00:00
2 | Hisano | Developer | 01-JAN-19 00:00:00
(5 行)

```

```

epas12=> desc employees_intval
Partitioned table "public.employees_intval"
 列 | 型 | 照合順序 | Null 値を許容 | デフォルト | ストレージ | 統計の対象 | 説明
-----+-----+-----+-----+-----+-----+-----+-----
empno | numeric(8,0) | | | | main | |
ename | character varying(32) | | | | extended | |
gender | character(1) | | | | extended | |
birthday | timestamp without time zone | | | | plain | |
deptno | numeric(2,0) | | | | main | |
job | character varying(9) | | | | extended | |
hiredate | timestamp without time zone | | | | plain | |
postno | character(7) | | | | extended | |
adress | character varying(1000) | | | | extended | |
telno | character varying(20) | | | | extended | |
mobileneno | character varying(20) | | | | extended | |
パーティションキー: RANGE (hiredate) INTERVAL ('1 year'::interval)
インデックス:

```

```

"employees_intval_idx0" UNIQUE, btree (hiredate, empno)
"employees_intval_idx1" UNIQUE, btree (mobileneno, hiredate)
"employees_intval_idx2" btree (telno)
Check 制約:
"employees_intval_empno_check" CHECK (empno < 100000000::numeric)
パーティション: employees_intval_employees_2001 FOR VALUES FROM (MINVALUE) TO ('01-JAN-
02 00:00:00'),
employees_intval_employees_2004 FOR VALUES FROM ('01-JAN-02 00:00:00') TO ('01-
JAN-03 00:00:00'),
employees_intval_sys89460103 FOR VALUES FROM ('01-JAN-19 00:00:00') TO ('01-JAN-
20 00:00:00'),
employees_intval_sys89460104 FOR VALUES FROM ('01-JAN-10 00:00:00') TO ('01-JAN-
11 00:00:00'),
employees_intval_sys89460105 FOR VALUES FROM ('01-JAN-18 00:00:00') TO ('01-JAN-
19 00:00:00')

```

- パーティション名は、自動で取得される
- パーティションの実テーブル名 = テーブル名+パーティション名

インターバル・パーティション (続き)

- ・索引も同時に追加される
- ・パーティションの索引名=パーティションのテーブル名+構成される索引キー+"idx"

```

epas12=> ¥di+
スキーマ |          名前          | リレーション一覧 | 所有者 | テーブル | サイズ |
説明
-----+-----+-----+-----+-----+-----
public | dept_idx0              | インデックス    | epas12 | dept      | 8192 bytes |
public | employees_intval_employees_2001_hiredate_empno_idx | インデックス    | epas12 | employees_intval_employees_2001_hiredate_empno_idx | 16 kB |
public | employees_intval_employees_2001_mobileno_hiredate_idx | インデックス    | epas12 | employees_intval_employees_2001_mobileno_hiredate_idx | 16 kB |
public | employees_intval_employees_2001_telno_idx           | インデックス    | epas12 | employees_intval_employees_2001_telno_idx           | 16 kB |
public | employees_intval_employees_2004_hiredate_empno_idx | インデックス    | epas12 | employees_intval_employees_2004_hiredate_empno_idx | 8192 bytes |
public | employees_intval_employees_2004_mobileno_hiredate_idx | インデックス    | epas12 | employees_intval_employees_2004_mobileno_hiredate_idx | 8192 bytes |
public | employees_intval_employees_2004_telno_idx           | インデックス    | epas12 | employees_intval_employees_2004_telno_idx           | 8192 bytes |
public | employees_intval_idx0                                  | partitioned index | epas12 | employees_intval_idx0                                  | 0 bytes |
public | employees_intval_idx1                                  | partitioned index | epas12 | employees_intval_idx1                                  | 0 bytes |
public | employees_intval_idx2                                  | partitioned index | epas12 | employees_intval_idx2                                  | 0 bytes |
public | employees_intval_sys89460103_hiredate_empno_idx      | インデックス    | epas12 | employees_intval_sys89460103_hiredate_empno_idx      | 16 kB |
public | employees_intval_sys89460103_mobileno_hiredate_idx   | インデックス    | epas12 | employees_intval_sys89460103_mobileno_hiredate_idx   | 16 kB |
public | employees_intval_sys89460103_telno_idx               | インデックス    | epas12 | employees_intval_sys89460103_telno_idx               | 16 kB |
public | employees_intval_sys89460104_hiredate_empno_idx      | インデックス    | epas12 | employees_intval_sys89460104_hiredate_empno_idx      | 16 kB |
public | employees_intval_sys89460104_mobileno_hiredate_idx   | インデックス    | epas12 | employees_intval_sys89460104_mobileno_hiredate_idx   | 16 kB |
public | employees_intval_sys89460104_telno_idx               | インデックス    | epas12 | employees_intval_sys89460104_telno_idx               | 16 kB |
(19行)
epas12=> explain select empno,ename, job,hiredate from employees_intval where hiredate between '01-JAN-2018' and '01-JAN-2019';
QUERY PLAN
-----
Append (cost=0.14..16.34 rows=2 width=140)
-> Index Scan using employees_intval_sys89460105_hiredate_empno_idx on employees_intval_sys89460105 (cost=0.14..8.16 rows=1 width=140)
    Index Cond: ((hiredate >= '01-JAN-18 00:00:00'::timestamp without time zone) AND (hiredate <= '01-JAN-19 00:00:00'::timestamp without time zone))
-> Index Scan using employees_intval_sys89460103_hiredate_empno_idx on employees_intval_sys89460103 (cost=0.14..8.16 rows=1 width=140)
    Index Cond: ((hiredate >= '01-JAN-18 00:00:00'::timestamp without time zone) AND (hiredate <= '01-JAN-19 00:00:00'::timestamp without time zone))
(5行)
  
```

・実際に、追加された索引を使用して検索の例



パーティション・テーブルに対する外部キー作成

EPAS11では、エラーになり、作成できない

-- 参照されるレンジ・パーティション作成 --

```
CREATE TABLE employees(  
    empno numeric(8,0) not null  
    ,ename varchar(32)  
    ,gender char(1)  
    ,birthday date  
    ,deptno number(2)  
    ,job varchar(9)  
    ,hiredate timestamp  
    ,postno char(7)  
    ,adress varchar2(1000)  
    ,telno varchar2(20)  
    ,mobilenno varchar2(20)  
)  
PARTITION BY RANGE (empno)(  
    PARTITION employees_0100 VALUES LESS THAN (100)  
    ,PARTITION employees_0200 VALUES LESS THAN (200)  
    ,PARTITION employees_MAX VALUES LESS THAN (MAXVALUE)  
);  
CREATE UNIQUE INDEX employees_idx0 on employees(empno);
```

-- レンジ・パーティションを参照するテーブル作成 --

```
CREATE TABLE emp_incident (  
    inci_no          number(8)  
    ,inc_date        date  
    ,inc_type        char(3)  
    ,empno           number(8)  
    ,description     clob  
    ,remark          clob  
);
```

[EPAS11]

```
epas11=> ALTER TABLE emp_incident ADD CONSTRAINT ref_empincident_employees  
FOREIGN KEY (empno) REFERENCES employees(empno);  
ERROR: パーティションテーブル"employees"は参照できません
```

[EPAS12]

```
epas12=> ALTER TABLE emp_incident ADD CONSTRAINT ref_empincident_employees  
FOREIGN KEY (empno) REFERENCES employees(empno);  
ALTER TABLE
```

!! パーティション・キーは、外部キーでもある必要がある

①外部キーを作成時は、参照先のカラムが、そのカラムの一意索引を持つ必要がある。ない場合は、下記エラー・メッセージが出力される。

psql: ERROR: 被参照テーブル "employees_wmax" に、指定したキーに一致する一意性制約がありません

②パーティション・テーブルで、一意索引を作成する場合は、パーティション・キーを含める必要がある。

psql: ERROR: UNIQUE 制約定義内の列が足りません

DETAIL: テーブル "employees" 上の UNIQUE制約にパーティションキーの一部である列 "hiredate" が含まれていません

・ 現状では、外部キーとパーティション・キーを組み合わせた一意索引を作成しても、上記①のエラーとなるため、外部キーをパーティション・キーとする必要がある



複合トリガー



複合トリガー

- 様々なタイミングで実行されるトリガー処理で、共通のデータを利用可能

```
CREATE OR REPLACE TRIGGER <複合トリガー名>  
FOR INSERT OR UPDATE OR DELETE ON <テーブル名>
```

COMPOUND TRIGGER

```
-- グローバル変数定義
```

```
BEFORE STATEMENT IS  
BEGIN
```

```
< 処理部分 # 1 >
```

```
END BEFORE STATEMENT;
```

```
BEFORE EACH ROW IS  
BEGIN
```

```
< 処理部分 # 2 >
```

```
END BEFORE EACH ROW;
```

```
AFTER EACH ROW IS  
BEGIN
```

```
< 処理部分 # 3 >
```

```
END AFTER EACH ROW;
```

```
AFTER STATEMENT IS  
BEGIN
```

```
< 処理部分 # 4 >
```

```
END AFTER STATEMENT;
```

```
END <複合トリガー名>;
```

```
/
```

グローバル変数宣言部分
ここで定義されたものは、個々のタイミン
グ・ポイントの処理部分で使用可能

Before Statement

Before Each Row

After Statement

After Each Row

複合トリガー (続き)

-- 通常テーブルに対し複合トリガー作成 --

```
epas12=> CREATE OR REPLACE TRIGGER trg_emp_incident
epas12-> FOR INSERT OR UPDATE OR DELETE ON emp_incident
epas12$> COMPOUND TRIGGER
epas12$> -- Global Declaration Section (optional)
epas12$> -- Variables declared here can be used inside any timing-point blocks.
epas12$> BEFORE STATEMENT IS
epas12$> BEGIN
epas12$> NULL;
epas12$> END BEFORE STATEMENT;
epas12$> BEFORE EACH ROW IS
epas12$> BEGIN
epas12$> NULL;
epas12$> END BEFORE EACH ROW;
epas12$> AFTER EACH ROW IS
epas12$> BEGIN
epas12$> NULL;
epas12$> END AFTER EACH ROW;
epas12$> AFTER STATEMENT IS
epas12$> BEGIN
epas12$> NULL;
epas12$> END AFTER STATEMENT;
epas12$> END trg_emp_incident;
CREATE TRIGGER
```

複合トリガー (続き)

```
--パーティション・テーブルに対し複合トリガー作成 --
epas12=> CREATE OR REPLACE TRIGGER trg_employees
epas12-> FOR INSERT OR UPDATE OR DELETE ON employees
epas12$> COMPOUND TRIGGER
epas12$> -- Global Declaration Section (optional)
epas12$> -- Variables declared here can be used inside any timing-point blocks.
epas12$> BEFORE STATEMENT IS
epas12$> BEGIN
epas12$> NULL;
epas12$> END BEFORE STATEMENT;
epas12$> BEFORE EACH ROW IS
epas12$> BEGIN
epas12$> NULL;
epas12$> END BEFORE EACH ROW;
epas12$> AFTER EACH ROW IS
epas12$> BEGIN
epas12$> NULL;
epas12$> END AFTER EACH ROW;
epas12$> AFTER STATEMENT IS
epas12$> BEGIN
epas12$> NULL;
epas12$> END AFTER STATEMENT;
epas12$> END trg_employees;
psql: ERROR: "employees"はパーティションテーブルです
DETAIL: パーティションテーブルは BEFORE / FOR EACH ROW トリガを持つことができません
```

パーティションに関しては、Afterトリガーのみ可能な点に注意

```
epas12=> CREATE OR REPLACE TRIGGER trg_employees
epas12-> FOR INSERT OR UPDATE OR DELETE ON employees
epas12$> COMPOUND TRIGGER
epas12$> -- Global Declaration Section (optional)
epas12$> -- Variables declared here can be used inside any timing-point blocks.
epas12$> AFTER EACH ROW IS
epas12$> BEGIN
epas12$> NULL;
epas12$> END AFTER EACH ROW;
epas12$> AFTER STATEMENT IS
epas12$> BEGIN
epas12$> NULL;
epas12$> END AFTER STATEMENT;
epas12$> END trg_employees;
CREATE TRIGGER
```

Afterトリガーのみ提議した場合は、正常に生成される

ファンクション



集計関数 : LISTAGG

- 複数行をまとめて 1 行に集約して表示

-- employees_intval テーブルの全データ --

```
epas12=> select empno,ename,job,hiredate from employees_intval order by empno;
```

```
empno | ename | job | hiredate
```

```
-----+-----+-----+-----  
1 | Takatsuru | Consult | 01-JAN-97 00:00:00  
2 | Hisano | Developer | 01-JAN-19 00:00:00  
3 | Fujita | President | 01-JAN-10 00:00:00  
4 | Kitagawa | Marketing | 01-FEB-18 00:00:00  
5 | Makabe | Sales | 01-AUG-18 00:00:00  
6 | Nishiyama | Analyst | 01-JAN-06 00:00:00  
7 | Hasegawa | Manager | 01-JAN-07 00:00:00  
8 | Terasawa | Expense | 01-JAN-08 00:00:00  
9 | Yamazaki | Clerk | 01-JAN-09 00:00:00  
10 | Yamada | Manager | 01-JAN-10 00:00:00  
11 | Ozeki | Analyst | 01-JAN-99 00:00:00  
12 | Sunada | Consult | 01-JAN-12 00:00:00  
13 | Tamura | Sales | 01-JAN-00 00:00:00  
14 | Teraoa | Sales | 01-JAN-13 00:00:00
```

(14 行)

-- JOB毎の従業員を表示 --

```
epas12=> SELECT job
```

```
epas12-> ,LISTAGG(ename ,',') WITHIN GROUP ( order by ename )
```

```
epas12-> FROM employees_intval
```

```
epas12-> GROUP BY job
```

```
epas12-> order by job;
```

```
job | listagg  
-----+-----  
Analyst | Nishiyama,Ozeki  
Clerk | Yamazaki  
Consult | Sunada,Takatsuru  
Developer | Hisano  
Expense | Terasawa  
Manager | Hasegawa,Yamada  
Marketing | Kitagawa  
President | Fujita  
Sales | Makabe,Tamura,Teraoa
```

(9 行)

集計関数：MEDIAN

- グループにおける中央値を取得

-- employees_intval テーブルの全データ --

```
epas12=> select empno,ename,job,hiredate from employees_intval order by empno;
```

```
empno | ename | job | hiredate
```

```
-----+-----+-----+-----  
1 | Takatsuru | Consult | 01-JAN-97 00:00:00  
2 | Hisano | Developer | 01-JAN-19 00:00:00  
3 | Fujita | President | 01-JAN-10 00:00:00  
4 | Kitagawa | Marketing | 01-FEB-18 00:00:00  
5 | Makabe | Sales | 01-AUG-18 00:00:00  
6 | Nishiyama | Analyst | 01-JAN-06 00:00:00  
7 | Hasegawa | Manager | 01-JAN-07 00:00:00  
8 | Terasawa | Expense | 01-JAN-08 00:00:00  
9 | Yamazaki | Clerk | 01-JAN-09 00:00:00  
10 | Yamada | Manager | 01-JAN-10 00:00:00  
11 | Ozeki | Analyst | 01-JAN-99 00:00:00  
12 | Sunada | Consult | 01-JAN-12 00:00:00  
13 | Tamura | Sales | 01-JAN-00 00:00:00  
14 | Teraoa | Sales | 01-JAN-13 00:00:00
```

(14 行)

-- Hiredate の平均、中央値、最大値を取得 --

```
epas12=> select job  
epas12-> ,avg(sysdate-hiredate)  
epas12-> ,median(sysdate-hiredate)  
epas12-> ,max(sysdate-hiredate)  
epas12-> from employees_intval  
epas12-> group by job  
epas12-> order by job;
```

```
job | avg | median | max  
-----+-----+-----+-----  
Analyst | 6267 days 23:10:50 | 6267 days 23:10:50 | 7546 days 11:10:50  
Clerk | 3893 days 11:10:50 | 3893 days 11:10:50 | 3893 days 11:10:50  
Consult | 5537 days 11:10:50 | 5537 days 11:10:50 | 8276 days 11:10:50  
Developer | 241 days 11:10:50 | 241 days 11:10:50 | 241 days 11:10:50  
Expense | 4259 days 11:10:50 | 4259 days 11:10:50 | 4259 days 11:10:50  
Manager | 4076 days 11:10:50 | 4076 days 11:10:50 | 4624 days 11:10:50  
Marketing | 575 days 11:10:50 | 575 days 11:10:50 | 575 days 11:10:50  
President | 3528 days 11:10:50 | 3528 days 11:10:50 | 3528 days 11:10:50  
Sales | 3335 days 27:10:50 | 2432 days 11:10:50 | 7181 days 11:10:50
```

(9 行)

CAST(MULTISET)

- 従来のバージョンでも、CAST()はサポート。EPAS12では、コレクションに対応

```
-- シンプルな型変換 --
epas12=> select ename, CAST(hiredate as varchar2) from employees_wmax;
  ename | hiredate
-----+-----
Takatsuru | 01-JAN-97 00:00:00
Fujita | 01-JAN-10 00:00:00
Hisano | 01-JAN-19 00:00:00
Kitagawa | 01-FEB-18 00:00:00
Makabe | 01-AUG-18 00:00:00
(5 行)
```

```
-- コレクションを変換 --
epas12=> CREATE OR REPLACE TYPE project_table_t AS TABLE OF
VARCHAR2(25);
CREATE TYPE
epas12=> CREATE TABLE projects (person_id NUMBER(10), project_name
VARCHAR2(20));
CREATE TABLE
epas12=> CREATE TABLE pers_short (person_id NUMBER(10), last_name
VARCHAR2(25));
CREATE TABLE
epas12=> INSERT INTO projects VALUES (1, 'Teach');
INSERT 0 1
epas12=> INSERT INTO projects VALUES (1, 'Code');
INSERT 0 1
```

```
epas12=> INSERT INTO projects VALUES (2, 'Code');
INSERT 0 1
epas12=> INSERT INTO pers_short VALUES (1, 'Morgan');
INSERT 0 1
epas12=> INSERT INTO pers_short VALUES (2, 'Kolk');
INSERT 0 1
epas12=> INSERT INTO pers_short VALUES (3, 'Scott');
INSERT 0 1
epas12=> COMMIT;
COMMIT
epas12-> SELECT e.last_name,
epas12-> ,CAST(MULTISET(
epas12(> SELECT p.project_name
epas12(> FROM projects p
epas12(> WHERE p.person_id = e.person_id
epas12(> ORDER BY p.project_name ) AS project_table_t)
epas12-> FROM pers_short
last_name | project_table_t
-----+-----
Morgan | {Code,Teach}
Kolk | {Code}
Scott | {}
(3 行)
```


SYS_GUID

- グローバルでユニークな16バイトのRAW型を生成
- uuid-ossplモジュールがベース

```
-- EPAS12 --  
epas12=> SELECT SYS_GUID(),SYS_GUID() from dual;  
      sys_guid      |      sys_guid  
-----+-----  
¥xa082cf80c0f14aefb956c5a75ebf053e | ¥x99f7fd4f098446cf9702eb3f692a4eec  
(1行)
```

```
-- Oracle --  
SQL> SELECT SYS_GUID(),SYS_GUID() from dual;  
  
SYS_GUID()          SYS_GUID()  
-----  
915022501EA63F09E0539600A8C0AD7C 915022501EA73F09E0539600A8C0AD7C
```

データ・ディクショナリ・ビュー



追加されたデータ・ディクショナリ・ビュー

- EPAS12では、下記データ・ディクショナリ・ビューを追加
 - {USER|ALL|DBA}_TAB_PRIVS
 - {USER|ALL|DBA}_COL_PRIVS
 - {USER|ALL|DBA}_DEPENDENCIES

{USER|ALL|DBA}_TAB_PRIVS

edb=# \d dba_tab_privs

ビュー "sys.dba_tab_privs"

列 | 型 | 照合順序 | Null 値を許容 | デフォルト

列	型	照合順序	Null 値を許容	デフォルト
grantee	character varying(128)	C		
owner	character varying(128)	C		
schema_name	character varying(128)	C		
table_name	character varying(128)	C		
grantor	character varying(128)	C		
privilege	character varying(40)	C		
grantable	character varying(3)	C		
hierarchy	character varying(3)	C		
common	character varying(3)			
type	character varying(24)			
inherited	character varying(3)			

edb=# \d user_tab_privs

ビュー "sys.user_tab_privs"

列 | 型 | 照合順序 | Null 値を許容 | デフォルト

列	型	照合順序	Null 値を許容	デフォルト
grantee	character varying(128)	C		
owner	character varying(128)	C		
schema_name	character varying(128)	C		
table_name	character varying(128)	C		
grantor	character varying(128)	C		
privilege	character varying(40)	C		
grantable	character varying(3)	C		
hierarchy	character varying(3)	C		
common	character varying(3)			
type	character varying(24)			
inherited	character varying(3)			

edb=# \d all_tab_privs

ビュー "sys.all_tab_privs"

列 | 型 | 照合順序 | Null 値を許容 | デフォルト

列	型	照合順序	Null 値を許容	デフォルト
grantor	character varying(128)	C		
grantee	character varying(128)	C		
table_schema	character varying(128)	C		
schema_name	character varying(128)	C		
table_name	character varying(128)	C		
privilege	character varying(40)	C		
grantable	character varying(3)	C		
hierarchy	character varying(3)	C		
common	character varying(3)			
type	character varying(24)			
inherited	character varying(3)			

{USER|ALL|DBA}_COL_PRIVS

edb=# \d dba_col_privs

ビュー "sys.dba_col_privs"

列 | 型 | 照合順序 | Null 値を許容 | デフォルト

列	型	照合順序	Null 値を許容	デフォルト
grantee	character varying(128)	C		
owner	character varying(128)	C		
schema_name	character varying(128)	C		
table_name	character varying(128)	C		
column_name	character varying(128)	C		
grantor	character varying(128)	C		
privilege	character varying(40)	C		
grantable	character varying(3)	C		
common	character varying(3)			
inherited	character varying(3)			

edb=# \d all_col_privs

ビュー "sys.all_col_privs"

列 | 型 | 照合順序 | Null 値を許容 | デフォルト

列	型	照合順序	Null 値を許容	デフォルト
grantor	character varying(128)	C		
grantee	character varying(128)	C		
table_schema	character varying(128)	C		
schema_name	character varying(128)	C		
table_name	character varying(128)	C		
column_name	character varying(128)	C		
privilege	character varying(40)	C		
grantable	character varying(3)	C		
common	character varying(3)			
inherited	character varying(3)			

edb=# \d user_col_privs

ビュー "sys.user_col_privs"

列 | 型 | 照合順序 | Null 値を許容 | デフォルト

列	型	照合順序	Null 値を許容	デフォルト
grantee	character varying(128)	C		
owner	character varying(128)	C		
schema_name	character varying(128)	C		
table_name	character varying(128)	C		
column_name	character varying(128)	C		
grantor	character varying(128)	C		
privilege	character varying(40)	C		
grantable	character varying(3)	C		
common	character varying(3)			
inherited	character varying(3)			

{USER|ALL|DBA}_DEPENDENCIES

edb=# \d dba_dependencies

ビュー "sys.dba_dependencies"

列	型	照合順序	Null 値を許容	デフォルト
owner	character varying(128)	C		
schema_name	character varying(128)	C		
name	character varying(128)	C		
type	character varying(18)			
referenced_owner	character varying(128)	C		
referenced_schema_name	character varying(128)	C		
referenced_name	character varying(128)	C		
referenced_type	character varying(18)			
referenced_link_name	character varying(128)			
dependency_type	character varying(4)			

edb=# \d all_dependencies

ビュー "sys.all_dependencies"

列	型	照合順序	Null 値を許容	デフォルト
owner	character varying(128)	C		
schema_name	character varying(128)	C		
name	character varying(128)	C		
type	character varying(18)			
referenced_owner	character varying(128)	C		
referenced_schema_name	character varying(128)	C		
referenced_name	character varying(128)	C		
referenced_type	character varying(18)			
referenced_link_name	character varying(128)			
dependency_type	character varying(4)			

edb=# \d user_dependencies

ビュー "sys.user_dependencies"

列	型	照合順序	Null 値を許容	デフォルト
schema_name	character varying(128)	C		
name	character varying(128)	C		
type	character varying(18)			
referenced_owner	character varying(128)	C		
referenced_schema_name	character varying(128)	C		
referenced_name	character varying(128)	C		
referenced_type	character varying(18)			
referenced_link_name	character varying(128)			
schemaid	numeric			
dependency_type	character varying(4)			

{USER|ALL|DBA}_DEPENDENCIES (続き)

```
edb=# select unique type, referenced_type from dba_dependencies;
```

```
type | referenced_type
-----+-----
FUNCTION | FUNCTION
FUNCTION | LANGUAGE
FUNCTION | PACKAGE
FUNCTION | SCHEMA
FUNCTION | TYPE
OPERATOR | FUNCTION
OPERATOR | SCHEMA
PACKAGE | FUNCTION
PACKAGE | PACKAGE
PACKAGE | TYPE
PACKAGE BODY | FUNCTION
PACKAGE BODY | PACKAGE
```

```
PACKAGE BODY | TYPE
PROCEDURE | LANGUAGE
PROCEDURE | SCHEMA
RULE | VIEW
TABLE | PACKAGE
TABLE | SCHEMA
TABLE | TYPE
TYPE | SCHEMA
TYPE | TYPE
TYPE BODY | TYPE
VIEW | FUNCTION
VIEW | SCHEMA
VIEW | TABLE
VIEW | TYPE
VIEW | VIEW
(27 行)
```

```
epas12=> select * from user_dependencies; schema_name | name | type | referenced_owner | referenced_schema_name | referenced_name |
referenced_type | referenced_link_name | schemaid | dependency_type
```

```
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
PUBLIC | EMP_INCIDENT | TABLE | ENTERPRISEDB | SYS | CLOB | TYPE | | | HARD
PUBLIC | TRG_EMPLOYEES | TRIGGER | EPAS12 | PUBLIC | EMPLOYEES | TABLE | | | 0 | HARD
PUBLIC | TRG_EMPLOYEES | TRIGGER | EPAS12 | PUBLIC | EMPLOYEES_EMPLOYEES_0100 | TABLE PARTITION | | | 0 | HARD
PUBLIC | TRG_EMPLOYEES | TRIGGER | EPAS12 | PUBLIC | EMPLOYEES_EMPLOYEES_0200 | TABLE PARTITION | | | 0 | HARD
PUBLIC | TRG_EMPLOYEES | TRIGGER | EPAS12 | PUBLIC | EMPLOYEES_EMPLOYEES_MAX | TABLE PARTITION | | | 0 | HARD
PUBLIC | TRG_EMP_INCIDENT | TRIGGER | EPAS12 | PUBLIC | EMP_INCIDENT | TABLE | | | 0 | HARD
PUBLIC | TRG_EMP_INCIDENT_EMP_INCIDENT | TYPE BODY | EPAS12 | PUBLIC | TRG_EMP_INCIDENT_EMP_INCIDENT | TYPE | | | 0 | HARD
PUBLIC | TRG_EMPLOYEES_EMPLOYEES | TYPE BODY | EPAS12 | PUBLIC | TRG_EMPLOYEES_EMPLOYEES | TYPE | | | 0 | HARD
(8 行)
```

その他



ROWIDの実装変更

- PG12でのテーブルの隠しOIDの廃止に伴い、EPAS12での実装方法を変更

```
-- EPAS11 --
epas11=> alter session set default_with_rowids =TRUE;
ALTER SESSION
epas11=> create table test_data (a number(1),b varchar2(2000));
CREATE TABLE
epas11=> %d+ test_data
          テーブル "public.test_data"
列 |      型      | 照合順序 | Null 値を許容 | デフォルト | ストレージ | 統計の対象 | 説明
-----+-----+-----+-----+-----+-----+-----+-----
a | numeric(1,0) |          |               |            | main       |            |
b | character varying(2000) |          |               |            | extended   |            |
インデックス:
"pg_oid_42119_index" UNIQUE, btree (oid)
OID あり: はい

epas11=> %d+ pg_oid_42119_index
          インデックス "public.pg_oid_42119_index"
列 | 型 | キー? | 定義 | ストレージ | 統計の対象
-----+-----+-----+-----+-----+-----
oid | oid | はい | oid | plain |
ユニーク,btree, テーブル "public.test_data" 用

epas11=> insert into test_data values(1,'AAA');
INSERT 42126 1
epas11=> insert into test_data values(2,'BBB');
INSERT 42127 1
epas11=> select *,rowid from test_data;
a | b | rowid
-----+-----+-----
1 | AAA | 42126
2 | BBB | 42127
(2 行)
```

• EPAS11までは、OID型として定義されていた。

```
-- EPAS12 --
epas12=> alter session set default_with_rowids =TRUE;
ALTER SESSION
epas12=> create table test_data (a number(1),b varchar2(2000));
CREATE TABLE
epas12=> %d+ test_data
          テーブル "public.test_data"
列 |      型      | 照合順序 | Null 値を許容 | デフォルト | ストレージ | 統計の対象 | 説明
-----+-----+-----+-----+-----+-----+-----+-----
a | numeric(1,0) |          |               |            | main       |            |
b | character varying(2000) |          |               |            | extended   |            |
インデックス:
"edb_rowid_17765_index" UNIQUE, btree (rowid)
Access method: heap

epas12=> %d+ edb_rowid_17765_index
          インデックス "public.edb_rowid_17765_index"
列 | 型 | キー? | 定義 | ストレージ | 統計の対象
-----+-----+-----+-----+-----+-----
rowid | bigint | はい | rowid | plain |
ユニーク,btree, テーブル "public.test_data" 用

epas12=> insert into test_data values(1,'AAA');
INSERT 0 1
epas12=> insert into test_data values(2,'BBB');
INSERT 0 1
epas12=> select *,rowid from test_data;
a | b | rowid
-----+-----+-----
1 | AAA | 1
2 | BBB | 2
(2 行)
```

• EPAS12では、bigint列として定義される

• テーブル個々で、1からの連番となる

SELECT UNIQUE

- EPAS12では、DISTINCTに加え、UNIQUEにも対応

```
-- EPAS11 -
epas11=> select distinct job from employees_wmax;
job
-----
President
Market
CONSUL
Developer
Sales
(5行)
```

```
epas11=> select unique job from employees_wmax;
ERROR: "job"またはその近辺で構文エラー
行 1: select unique job from employees_wmax;
      ^
```

```
-- EPAS12 -
epas12=> select distinct job from employees_wmax;
job
-----
President
Market
CONSUL
Developer
Sales
(5行)
```

```
epas12=> select unique job from employees_wmax;
job
-----
President
Market
CONSUL
Developer
Sales
(5行)
```

• EPAS11ではNGだが、EPAS12ではOK

CTE(Common Table Expressions)

```
-- EPAS11 --
epas11=> explain (analyze) with d_cte as (select * from dept), e_cte as (select * from employees_wmax)
epas11-> select e.empno , e.ename, e.job, d.dname, d.location
epas11-> from e_cte e
epas11-> left outer join d_cte d
epas11-> on d.deptno = e.deptno;
```

QUERY PLAN

```
-----
Hash Left Join (cost=115.98..138.85 rows=810 width=968) (actual time=0.221..0.329 rows=14 loops=1)
Hash Cond: (e.deptno = d.deptno)
CTE d_cte
-> Seq Scan on dept (cost=0.00..10.90 rows=90 width=848) (actual time=0.013..0.015 rows=5 loops=1)
CTE e_cte
-> Append (cost=0.00..102.15 rows=810 width=832) (actual time=0.020..0.102 rows=14 loops=1)
-> Seq Scan on employees_wmax_employees_2001 (cost=0.00..10.90 rows=90 width=832) (actual time=0.019..0.020 rows=3 loops=1)
-> Seq Scan on employees_wmax_employees_2004 (cost=0.00..10.90 rows=90 width=832) (actual time=0.004..0.004 rows=0 loops=1)
-> Seq Scan on employees_wmax_employees_2007 (cost=0.00..10.90 rows=90 width=832) (actual time=0.007..0.008 rows=2 loops=1)
-> Seq Scan on employees_wmax_employees_2010 (cost=0.00..10.90 rows=90 width=832) (actual time=0.015..0.016 rows=4 loops=1)
-> Seq Scan on employees_wmax_employees_2013 (cost=0.00..10.90 rows=90 width=832) (actual time=0.030..0.031 rows=2 loops=1)
-> Seq Scan on employees_wmax_employees_2014 (cost=0.00..10.90 rows=90 width=832) (actual time=0.004..0.004 rows=0 loops=1)
-> Seq Scan on employees_wmax_employees_2015 (cost=0.00..10.90 rows=90 width=832) (actual time=0.003..0.003 rows=0 loops=1)
-> Seq Scan on employees_wmax_employees_2016 (cost=0.00..10.90 rows=90 width=832) (actual time=0.003..0.003 rows=0 loops=1)
-> Seq Scan on employees_wmax_employees_max (cost=0.00..10.90 rows=90 width=832) (actual time=0.007..0.008 rows=3 loops=1)
-> CTE Scan on e_cte e (cost=0.00..16.20 rows=810 width=144) (actual time=0.024..0.114 rows=14 loops=1)
-> Hash (cost=1.80..1.80 rows=90 width=848) (actual time=0.037..0.037 rows=5 loops=1)
    Buckets: 1024 Batches: 1 Memory Usage: 9kB
    -> CTE Scan on d_cte d (cost=0.00..1.80 rows=90 width=848) (actual time=0.019..0.024 rows=5 loops=1)
Planning Time: 0.506 ms
Execution Time: 0.419 ms
(21 行)
```

**Materialized (実体) を作成後に
ジョインを実行**

CTE(Common Table Expressions) (続き)

```
-- EPAS12 --
epas11=> select distinct job from employees_wmax;
epas12=> explain (analyze) with d_cte as (select * from dept), e_cte as (select * from employees_intval)
epas12-> select e.empno , e.ename, e.job, d.dname, d.location
epas12-> from e_cte e
epas12-> left outer join d_cte d
epas12-> on d.deptno = e.deptno;
```

QUERY PLAN

```
-----
Hash Left Join (cost=12.03..97.25 rows=635 width=968) (actual time=0.050..0.151 rows=14 loops=1)
  Hash Cond: (employees_intval_employees_2001.deptno = dept.deptno)
  -> Append (cost=0.00..83.53 rows=635 width=144) (actual time=0.019..0.104 rows=14 loops=1)
    -> Seq Scan on employees_intval_employees_2001 (cost=0.00..1.01 rows=1 width=144) (actual time=0.018..0.019 rows=3 loops=1)
    -> Seq Scan on employees_intval_employees_2004 (cost=0.00..10.90 rows=90 width=144) (actual time=0.003..0.003 rows=0 loops=1)
    -> Seq Scan on employees_intval_sys89460106 (cost=0.00..10.90 rows=90 width=144) (actual time=0.007..0.008 rows=1 loops=1)
    -> Seq Scan on employees_intval_sys89460107 (cost=0.00..10.90 rows=90 width=144) (actual time=0.017..0.017 rows=1 loops=1)
    -> Seq Scan on employees_intval_sys89460108 (cost=0.00..10.90 rows=90 width=144) (actual time=0.007..0.008 rows=1 loops=1)
    -> Seq Scan on employees_intval_sys89460109 (cost=0.00..10.90 rows=90 width=144) (actual time=0.007..0.007 rows=1 loops=1)
    -> Seq Scan on employees_intval_sys89460104 (cost=0.00..1.01 rows=1 width=144) (actual time=0.007..0.008 rows=2 loops=1)
    -> Seq Scan on employees_intval_sys89460110 (cost=0.00..10.90 rows=90 width=144) (actual time=0.006..0.007 rows=1 loops=1)
    -> Seq Scan on employees_intval_sys89460111 (cost=0.00..10.90 rows=90 width=144) (actual time=0.007..0.007 rows=1 loops=1)
    -> Seq Scan on employees_intval_sys89460105 (cost=0.00..1.02 rows=2 width=144) (actual time=0.008..0.008 rows=2 loops=1)
    -> Seq Scan on employees_intval_sys89460103 (cost=0.00..1.01 rows=1 width=144) (actual time=0.007..0.007 rows=1 loops=1)
  -> Hash (cost=10.90..10.90 rows=90 width=848) (actual time=0.018..0.018 rows=5 loops=1)
    Buckets: 1024 Batches: 1 Memory Usage: 9kB
    -> Seq Scan on dept (cost=0.00..10.90 rows=90 width=848) (actual time=0.007..0.009 rows=5 loops=1)
Planning Time: 0.948 ms
Execution Time: 0.227 ms
(19 行)
```

**Materialized (実体) を作成せずに
ジョインを実行**

生成列のサポート

- テーブル作成時に導出カラムを作成可能。但し、実際に領域を確保する。

```
-- EPAS12 --
epas12=> create table test_table ( a number(5), b varchar(20), c varchar(20), d varchar(20) GENERATED ALWAYS AS (b || c) STORED );
```

• 生成列の定義

```
CREATE TABLE
epas12=> ¥d+ test_table
```

列	型	照合順序	Null 値を許容	デフォルト	ストレージ	統計の対象	説明
a	numeric(5,0)				main		
b	character varying(20)				extended		
c	character varying(20)				extended		
d	character varying(20)			generated always as (((b::text c::text))) stored	extended		

インデックス:
"edb_rowid_17808_index" UNIQUE, btree (rowid)
Access method: heap

```
epas12=> insert into test_table(a,b,c,d) values(1,'AA','BB','CC');
psql: ERROR: 列"d"への挿入はできません
DETAIL:  Column "d" is a generated column.
```

• 生成列への挿入は不可

```
epas12=> insert into test_table(a,b,c) values(1,'AA','BB');
INSERT 0 1
```

```
epas12=> select * from test_table;
a | b | c | d
---+---+---+---
1 | AA | BB | AABB
(1 行)
```

```
epas12=> update test_table set b='ZZ';
UPDATE 1
epas12=> select * from test_table;
a | b | c | d | e
```

```
1 | ZZ | BB | ZZBB | ZZBB
(1 行)
```

• 生成列の変更は不可

```
epas12=> update test_table set d='XXXXX';
psql: ERROR: 列"d"はDEFAULTにのみ更新可能です
DETAIL:  Column "d" is a generated column.
```

```
epas12=> alter table test_table add e varchar(30) generated always as (b || c) stored;
ALTER TABLE
epas12=> ¥d test_data
```

• ALTER TABLEによる生成列追加可能。

列	型	照合順序	Null 値を許容	デフォルト
a	numeric(1,0)			
b	character varying(2000)			

インデックス:
"edb_rowid_17765_index" UNIQUE, btree (rowid)

```
epas12=> ¥d test_table
```

列	型	照合順序	Null 値を許容	デフォルト
a	numeric(5,0)			
b	character varying(20)			
c	character varying(20)			
d	character varying(40)			generated always as (((b::text c::text))) stored
e	character varying(30)			generated always as (((b::text c::text))) stored

インデックス:
"edb_rowid_17815_index" UNIQUE, btree (rowid)



pg_checksums

- チェックサムの整合性の有効化/無効化/チェックが可能

```
[enterprisedb@epassrv03 ~]$ pg_checksums  
pg_checksums: error: cluster must be shut down
```

・V12では、オフライン中の時だけ、チェック可能

```
[enterprisedb@epassrv03 ~]$ pg_checksums --check  
pg_checksums: error: data checksums are not enabled in cluster
```

```
[enterprisedb@epassrv03 ~]$ pg_checksums --enable  
Checksum operation completed  
Files scanned: 2233  
Blocks scanned: 8353  
pg_checksums: syncing data directory  
pg_checksums: updating control file  
Checksums enabled in cluster
```

・チェック・サムの有効化

```
[enterprisedb@epassrv03 ~]$ pg_checksums --check  
Checksum operation completed  
Files scanned: 2233  
Blocks scanned: 8353  
Bad checksums: 0  
Data checksum version: 1
```

・チェック・サムのチェック

```
[enterprisedb@epassrv03 ~]$ pg_checksums --check --progress  
65/65 MB (100%) computed  
Checksum operation completed  
Files scanned: 2233  
Blocks scanned: 8353  
Bad checksums: 0  
Data checksum version: 1
```

・チェック・サムのチェック(進捗確認)

```
[enterprisedb@epassrv03 ~]$ pg_checksums --disable  
pg_checksums: syncing data directory  
pg_checksums: updating control file  
Checksums disabled in cluster
```

・チェック・サムの無効化

EDB Postgres Advanced Server 12 新機能

■ PostgreSQL12

- パーティション・テーブルの改善
 - パーティション・テーブルに対する参照整合性制約
 - Fast Run -Time Pruning
 - Attach partition w/ShareUpdateExclusiveLock
 - パーティション・テーブルへの高速COPY
- インデックス
 - B-Tree インデックスのよりサイズ縮小
 - REINDEX CONCURRENTLY
- SQL
 - 生成列のサポート
 - CTE のパフォーマンス向上
- 運用管理
 - pg_checksum
 - pg_stat_progress_cluster/pg_stat_progress_create_index
 - recovery.confの廃止 (postgresql.confへの統合)
- プラガバブル・ストレージ・エンジン

■ EPAS v12

- Oracle互換性
 - ✓ インターバル・パーティション
 - ✓ 複合トリガー
 - ✓ 集計関数の追加(LISTAGG/MEDIAN)
 - ✓ CAST関数のコレクション対応
 - ✓ SYS_GUIDのサポート
 - ✓ 追加されたデータ・ディクショナリ・ビュー
 - ✓ SELECT UNIQUE
 - ✓ PG12のOID廃止に伴うROWIDの挙動変更
- スタンバイDBのロジカル・レプリケーション・スロットのサポート

その他多数



The End