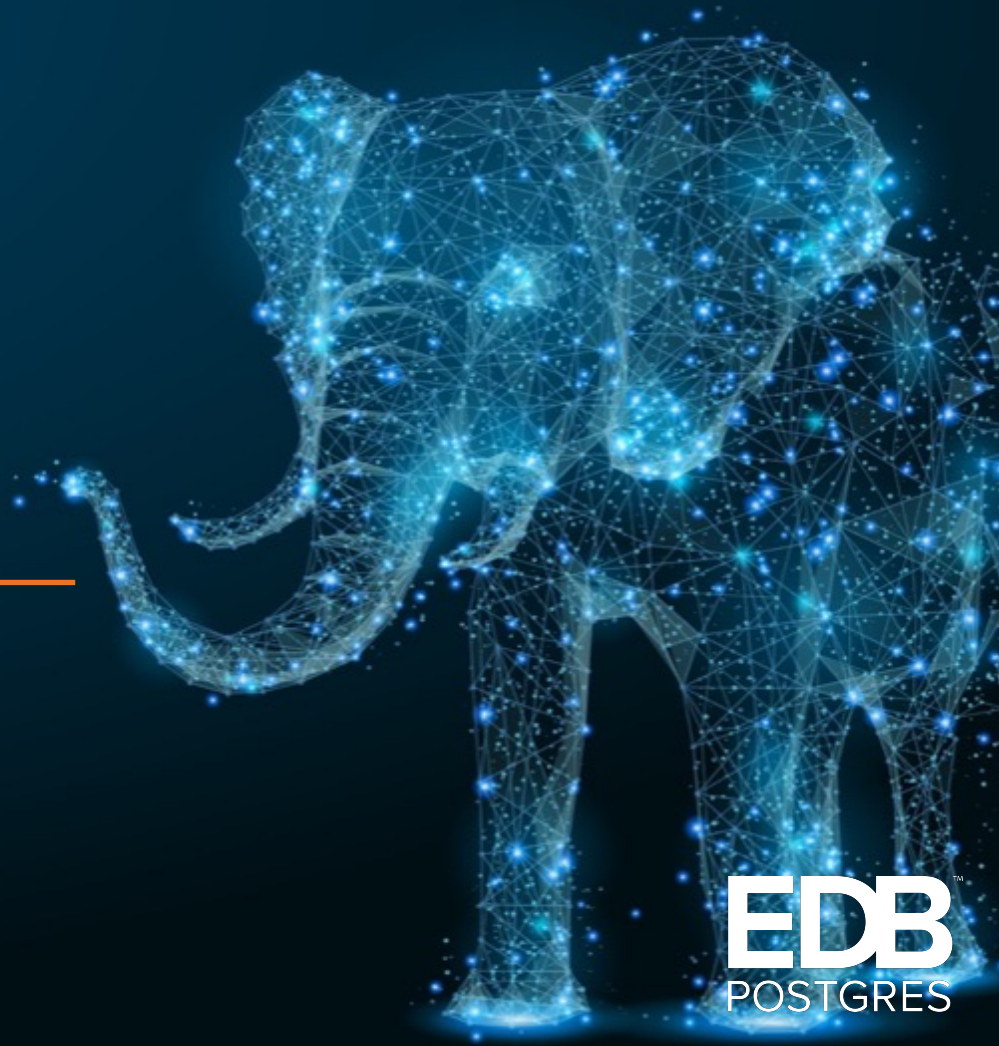
Do More with EDB Postgres



김지훈 이사, APAC Solution Architect

2019-05-18

EDB POSTGRES

OPEN SOURCE IS THE NEW DATA CENTER STANDARD!

가트너, 오픈소스 RDBMS의 현주소, 2015

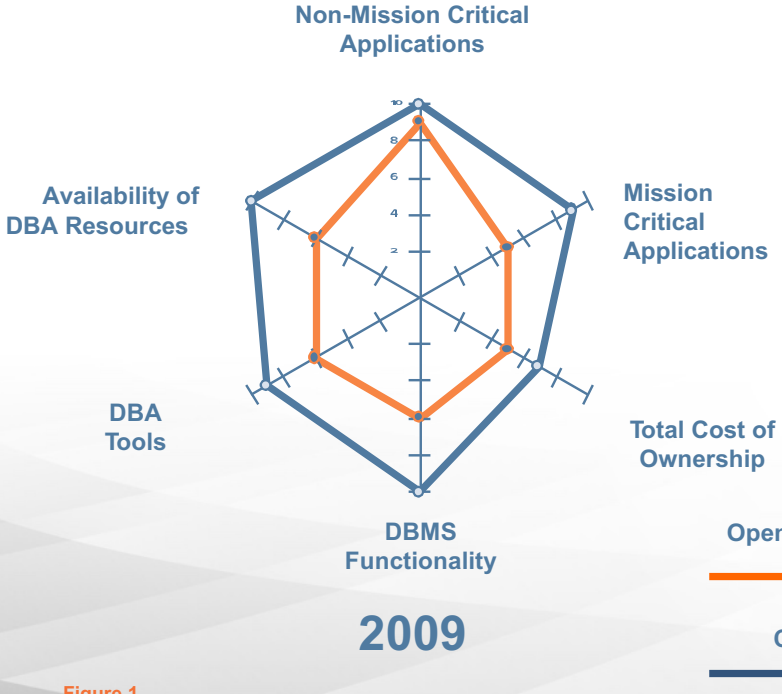
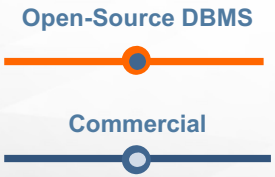


Figure 1
Relational Open-Source DBMS Maturity Evaluation, 2015
Source: Gartner (April 2015)



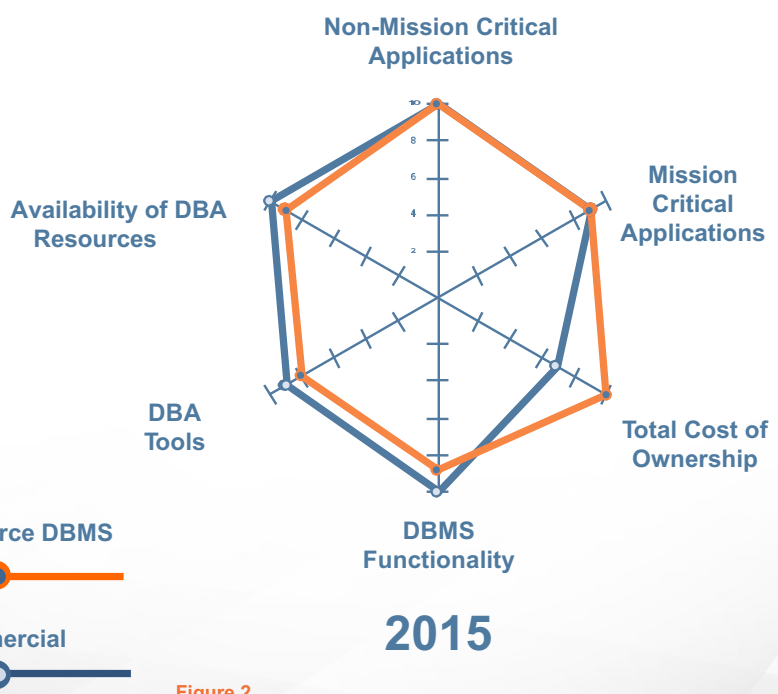


Figure 2
Relational Open-Source DBMS Maturity Evaluation,2015
Source: Gartner (April 2015)

2022년까지 신규 개발 어플리케이션의 **70%** 이상은 오픈소스 **DB**를 기반으로 개발될 것이고, 기존 상용 **DB**의 **50%**는 오픈소스 **DB**로 교체되거나 그 과정에 있을 것이다.

— Gartner,
State of the Open-Source DBMS Market,
Merv Adrian, Donald Feinberg, February 28, 2018

EDB POSTGRES

POSTGRES: 2017,18년 올해의 DBMS!

PostgreSQL is the DBMS of the Year 2018

by Paul Andlinger, Matthias Gelbmann, 2 January 2019
Tags: DBMS of the year, MongoDB, PostgreSQL, Redis

> *PostgreSQL is the database management system that gained more popularity in our DB-Engines Ranking within the last year than any of the other 343 monitored systems.*
> *We thus declare* **PostgreSQL** *as the* **DBMS of the Year 2018**.

DB-Engines Ranking

345 systems in ranking, March 2019

Rank Mar 2019	Rank Feb 2019	Rank Mar 2018	DBMS	Database Model	Score Mar 2019	Score Feb 2019	Score Mar 2018
1.	1.	1.	Oracle	Relational, Multi-model	1279.14	+15.12	-10.47
2.	2.	2.	MySQL	Relational, Multi-model	1198.25	+30.96	-30.62
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1047.85	+7.79	-56.94
4.	4.	4.	PostgreSQL	Relational, Multi-model	469.81	-3.75	+70.46
5.	5.	5.	MongoDB	Document	401.34	+6.24	+60.82
6.	6.	6.	IBM Db2	Relational, Multi-model	177.20	-2.23	-9.47
7.	↑ 9.	7.	Microsoft Access	Relational	146.20	+2.18	+14.26
8.	↓ 7.	8.	Redis	Key-value, Multi-model	146.12	-3.32	+14.90
9.	↓ 8.	9.	Elasticsearch	Search engine, Multi-model	142.79	-2.46	+14.25
10.	10.	↑ 11.	SQLite	Relational	124.87	-1.29	+10.06

EDB POSTGRES

WHO IS EDB?

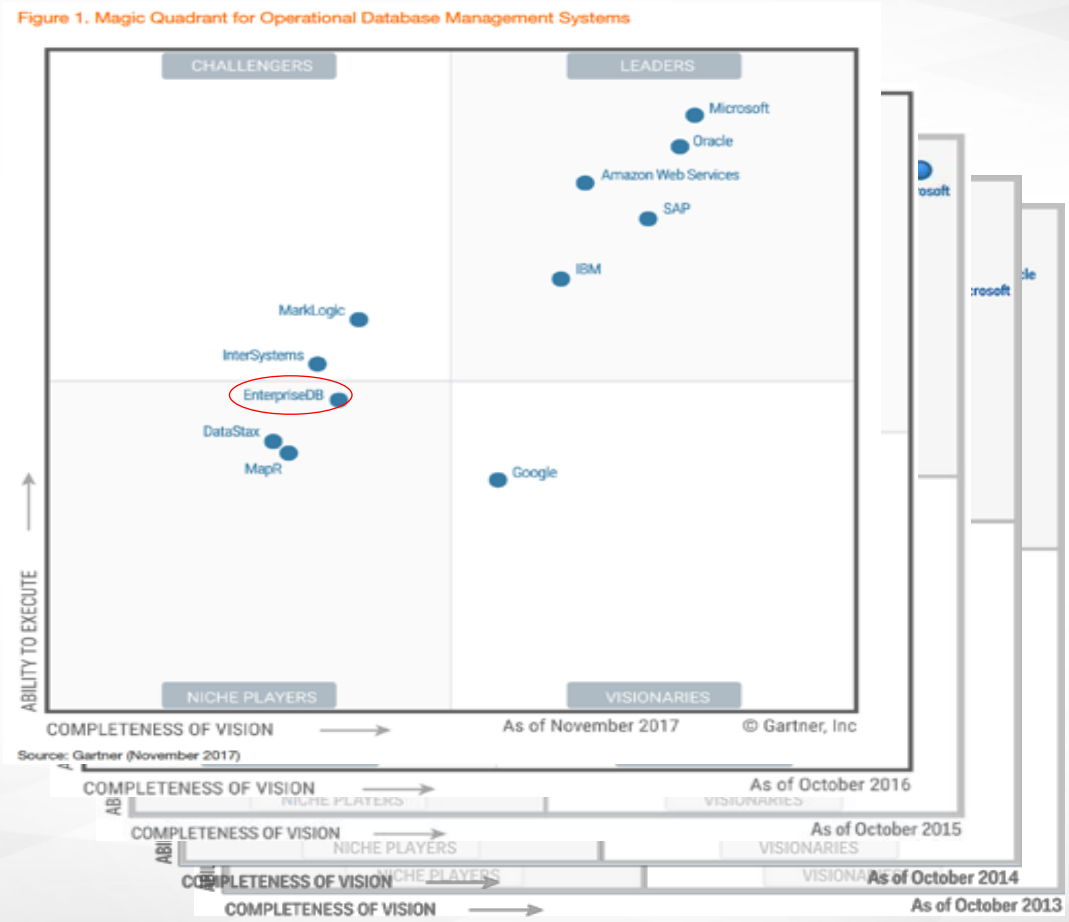
The world leader in open-source based Postgres software and services.

- **Founded in 2004**
- **Recognized RDBMS leader by:**
  - Gartner
  - Forrester
- **Customer base > 4000**
- **300+ employees**
- **Offices worldwide**
- **Largest PostgreSQL community leader**

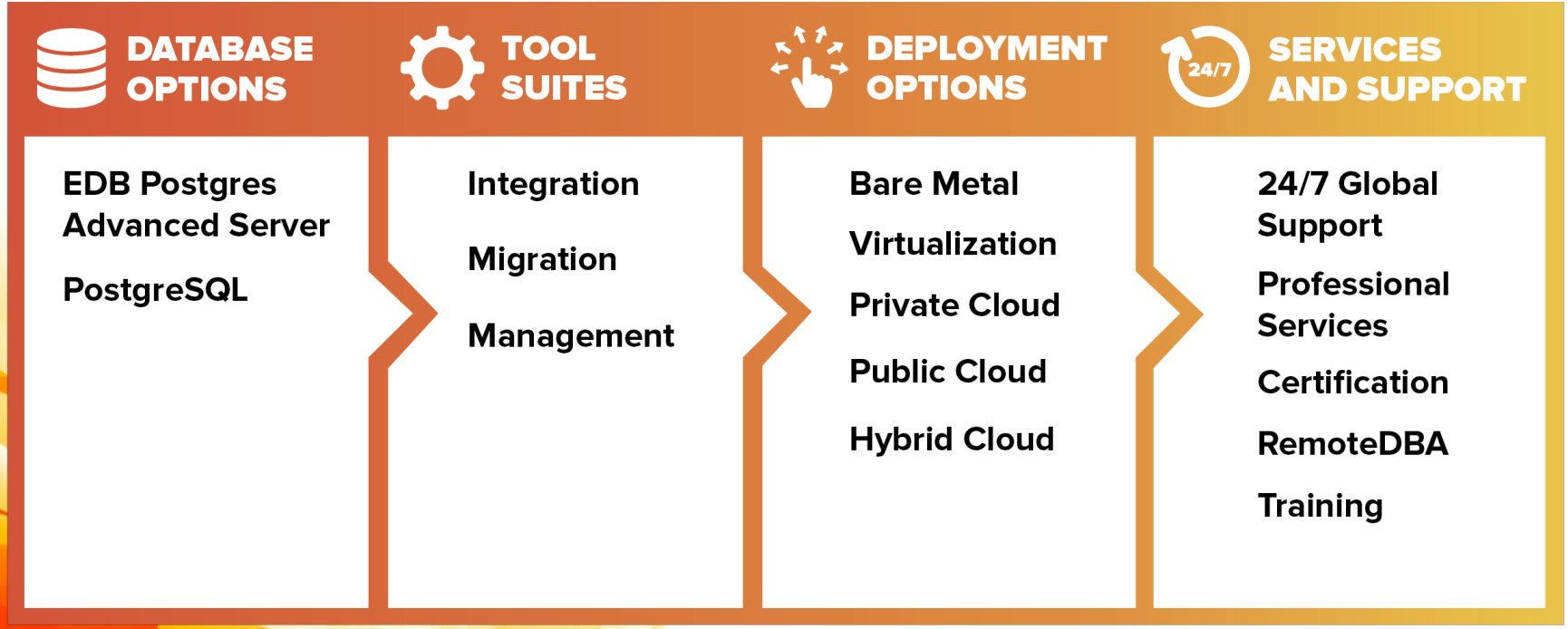
EDB POSTGRES

GARTNER MQ의 유일한 오픈소스 RDBMS

2013년부터 5년 연속 Gartner's Magic Quadrant 포함



EDB POSTGRES PLATFORM



EDB POSTGRES SOLUTION USE CASES

New Applications

- DevOps, schema-less rapid development, and multiple programming language support

Application Modernization

- Multi-model flexibility and integration with popular data sources

Legacy Migration

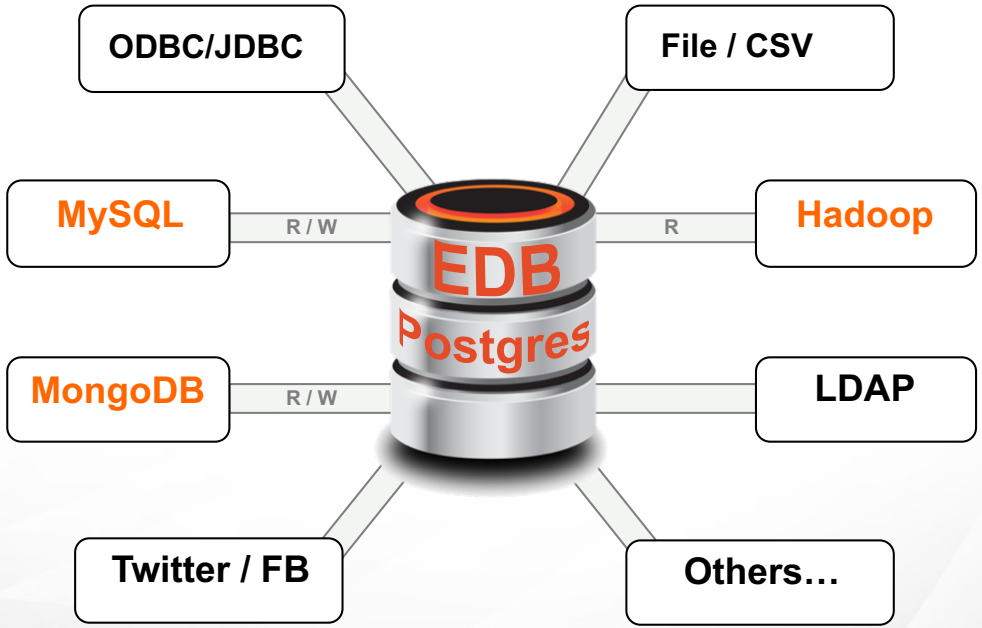
- Compatibility with Oracle leverages existing DBA and developer skills

Migration to Cloud

- Flexible deployment options and simple business terms for moving to the cloud

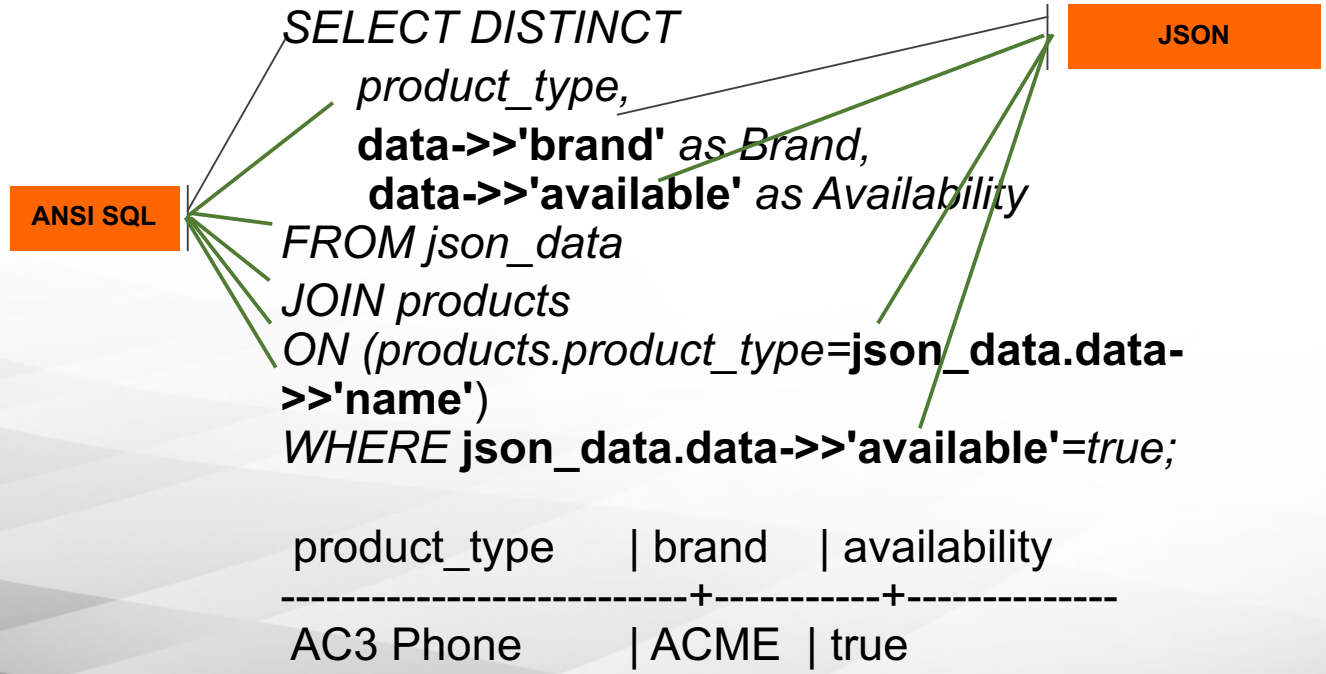
웹, 빅데이터, 도큐먼트 데이터와 **EDB POSTGRES**

- Foreign Data Wrapper 와 SQL/MED 기반
- MySQL, MongoDB & Hadoop FDW는 서브스크립션에 기본 포함
- 커뮤니티를 통한 기타 FDWs 지원



NOSQL & RDBMS를 동시에 사용?

- SQL과 NoSQL이 동시에 사용되는 경우 **별도의 프로그램 로직 불필요**
  – **Postgres does it all!**



JSON OPERATORS

Operator	Right Operand Type	Description	Example
->	int	Get JSON array element (indexed from zero, negative integers count from the end)	'[{"a":"foo"},{"b":"bar"},{"c":"baz"}]'::json->2
->	text	Get JSON object field by key	'{"a": {"b":"foo"}}'::json->'a'
->>	int	Get JSON array element as text	'[1,2,3]'::json->>2
->>	text	Get JSON object field as text	'{"a":1,"b":2}'::json->>'b'
#>	text[]	Get JSON object at specified path	'{"a": {"b":{"c": "foo"}}}'::json#>'{a,b}'
#>>	text[]	Get JSON object at specified path as text	'{"a":[1,2,3],"b":[4,5,6]}'::json#>>'{a,2}'
@>	jsonb	Does the left JSON value contain the right JSON path/value entries at the top level?	'{"a":1, "b":2}'::jsonb @> '{"b":2}'::jsonb
<@	jsonb	Are the left JSON path/value entries contained at the top level within the right JSON value?	'{"b":2}'::jsonb <@ '{"a":1, "b":2}'::jsonb
?	text	Does the string exist as a top-level key within the JSON value?	'{"a":1, "b":2}'::jsonb ? 'b'
?|	text[]	Do any of these array strings exist as top-level keys?	'{"a":1, "b":2, "c":3}'::jsonb ?| array['b', 'c']
?&	text[]	Do all of these array strings exist as top-level keys?	'["a", "b"]'::jsonb ?& array['a', 'b']
||	jsonb	Concatenate two jsonb values into a new jsonb value	'["a", "b"]'::jsonb || '["c", "d"]'::jsonb
-	text	Delete key/value pair or string element from left operand. Key/value pairs are matched based on their key value.	'{"a": "b"}'::jsonb - 'a'
-	text[]	Delete multiple key/value pairs or string elements from left operand. Key/value pairs are matched based on their key value.	'{"a": "b", "c": "d"}'::jsonb - '{a,c}'::text[]
-	integer	Delete the array element with specified index (Negative integers count from the end). Throws an error if top level container is not an array.	'["a", "b"]'::jsonb - 1
#-	text[]	Delete the field or element with specified path (for JSON arrays, negative integers count from the end)	'["a", {"b":1}]'::jsonb #- '{1,b}'

JSON FUNCTIONS

Function	Description	Example	Example Result
to_json(anyelement)	Returns the value as json or jsonb.	to_json('Fred said "Hi."'::text)	"Fred said \"Hi.\""
array_to_json(anyarray [, pretty_bool])	Returns the array as a JSON array.	array_to_json('{{1,5},{99,100}}'::int[])	[[1,5],[99,100]]
row_to_json(record [, pretty_bool])	Returns the row as a JSON object.	row_to_json(row(1,'foo'))	{"f1":1,"f2":"foo"}
json_build_array(VARIADIC "any")	Builds a possibly-heterogeneously-typed JSON array out of a variadic argument list.	json_build_array(1,2,'3',4,5)	[1, 2, "3", 4, 5]
json_build_object(VARIADIC "any")	Builds a JSON object out of a variadic argument list.	json_build_object('foo',1,'bar',2)	{"foo": 1, "bar": 2}
json_object(text[])	Builds a JSON object out of a text array.	json_object('{a, 1, b, "def", c, 3.5}')	{"a": "1", "b": "def", "c": "3.5"}
json_object(keys text[], values text[])	This form of json_object takes keys and values pairwise from two separate arrays. In all other respects it is identical to the one-argument form.	json_object('{a, b}', '{1,2}')	{"a": "1", "b": "2"}

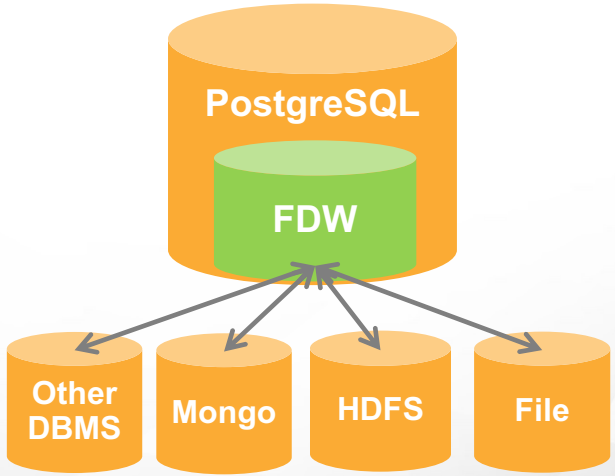
참고 : https://www.postgresql.org/docs/current/functions-json.html

JSON, JSONB DEMO

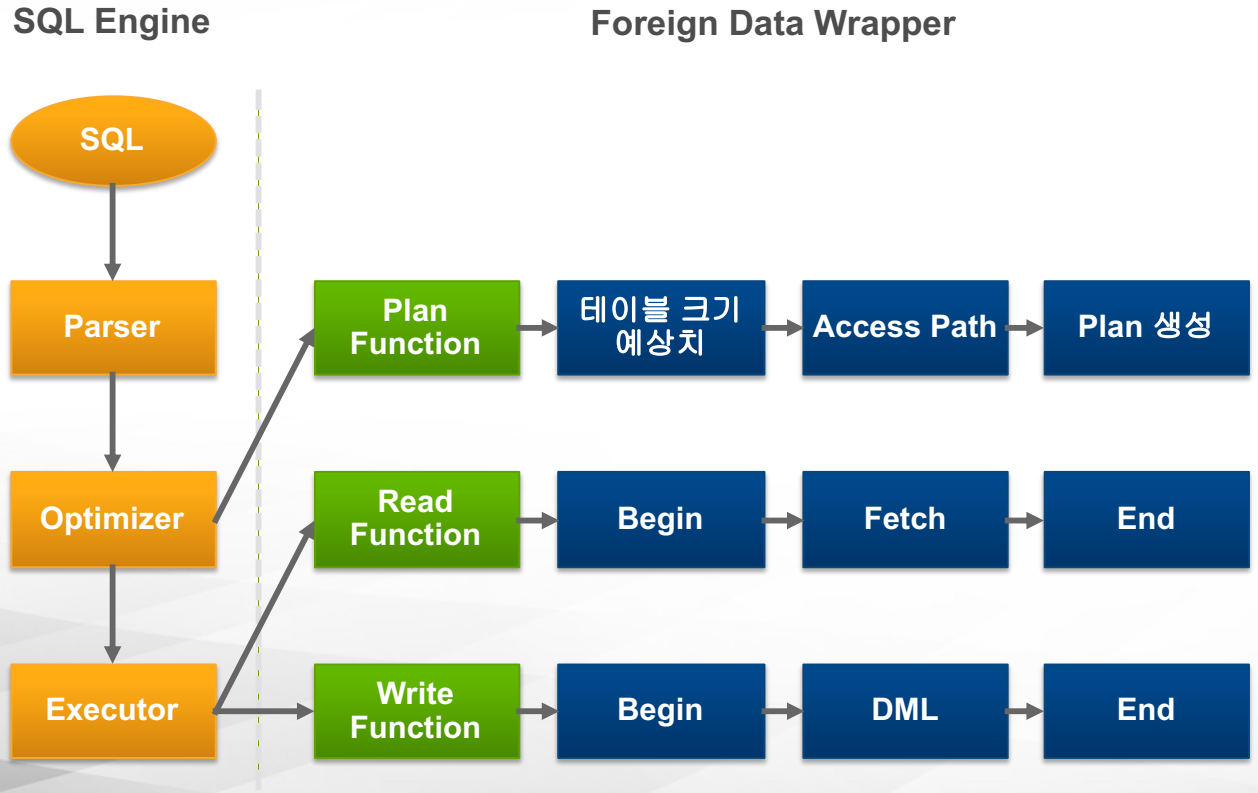
- Demo 진행 절차
  - http://examples.citusdata.com/customer_reviews_nested_1998.json.gz
  - 209MB 60만 건 가량의 리뷰 데이터
  - 테이블 생성 & 로딩
  - JSON 오퍼레이터를 이용한 쿼리

FDW (FOREIGN DATA WRAPPER)란?

- **PostgreSQL**을 다른 시스템 / **DB** 연결을 통한 데이터 수집이나 쿼리/조인 등을 위한 통합 인터페이스로 활용
- 원격의 다양한 형식의 데이터소스를 로컬 테이블처럼 사용할 수 있도록 하는 기능
- 데이터소스는 **DB**, **NoSQL**, 하둡, 파일, 웹 등 형식이 정해져있지 않음
- 지원 기능
  - WHERE 조건절 필터 Push-down
  - DISTINCT, ORDER BY, GROUP BY
  - 내부 테이블과 조인
  - 비교, 수학, 문자, 패턴 매칭, 날짜/시간 등 다양한 함수 지원
  - DML 지원



FDW 작동 원리



FDW 설정 방법

- Extension 생성

```
CREATE EXTENSION dummy_fdw;
```

- Foreign Server 생성

```
CREATE SERVER dummy_server
FOREIGN DATA WRAPPER dummy_fdw
OPTIONS (host '127.0.0.1', port '3306');
```

- User-Mapping 생성

```
CREATE USER MAPPING FOR postgres
SERVER dummy_server
OPTIONS (username 'foo', password 'bar');
```

- Foreign Table 생성

```
CREATE FOREIGN TABLE f_table(id int, name text)
SERVER mysql_server
OPTIONS (dbname 'db', table_name 'r_table');
```

FDW 사용 방법

- Foreign Table 사용법은 일반 테이블의 사용법과 동일
- Foreign Table 조회

```
SELECT id, name FROM f_table WHERE id = 1;
```

- Foreign Table 데이터 입력

```
INSERT INTO f_table values (1, 'foor',);
INSERT INTO f_table values (2, 'bar',);
```

- Foreign Table 데이터 삭제

```
DELETE FROM f_table where id= 2;
```

- Foreign Table 데이터 업데이트

```
UPDATE f_table set name= 'bar' WHERE id = 1;
```

- 실행 계획 조회

```
EXPLAIN SELECT id, name FROM f_table WHERE name LIKE 'foo' limit 1;
```

JSON, JSONB DEMO

- **Demo 진행 절차**
  - Load MongoDB Extension
  - Create MongoDB FDW
  - Execute DML

USE CASE #1 : MONGODB FDW

***Undisclosed* 미국계 글로벌 카드사**

	내용
당면 과제	① MongoDB와 유연한 인터페이스 ② JSON 쿼리 사용가능 및 DML 지원
기술 요소	① MongoDB FDW로 PPAS에 외부 테이블로 생성 ② 외부 테이블에 JSON 쿼리, 타 테이블과 조인 및 DML
효과	① MongoDB와 RDBMS 간의 쉬운 인터페이스 및 높은 이식성 ② SQL 문장으로 MongoDB와 PPAS 상호간의 데이터 마이그레이션 용이 (MongoDB ⇔ PPAS)
버전	Postgres Plus Advanced Server 9.3 MongoDB FDW

USE CASE #2 : CASSANDRA FDW

Samsung SDS Smart Home

		내용
당면 과제		① IoT 플랫폼의 Cassandra에 BLOB으로 저장된 IoT 단말 데이터를 활용해 단말 통계 생성 ② Cassandra에 JSON 쿼리로 필요 데이터만 추출
기술 요소		① Cassandra 데이터를 FDW로 DB 외부 테이블로 생성 ② 외부 테이블에 JSON 쿼리를 사용하여 원본인 BLOB 데이터에서 필요 항목만 추출
효과		① 순수 SQL 구문만으로 별도의 프로그래밍없이 NoSQL 데이터 핸들링이 가능하여 개발 생산성 극대화 ② DB의 메타 정보와 조인하여 다양한 쿼리 작성 가능
업무	업무	IoT 단말 통계 정보 생성
	제품	Postgres Plus Advanced Server 9.4 Cassandra FDW

THANK YOU

The Most Complete Open Source Database Platform

sales-kr@enterprisedb.com
www.enterprisedb.com

EDB POSTGRES