



EDB

EDB Query Advisor

リリース 1.0.0

EnterpriseDB Corporation

2023 年 06 月 07 日

目次

第 1 章	EDB Query Advisor release notes	3
1.1	Release notes for Query Advisor version 1.0.0	3
第 2 章	Installing Query Advisor	5
2.1	前提条件	5
2.2	パッケージをインストールする	5
第 3 章	パラメーター	7
第 4 章	Using EDB Query Advisor	9
4.1	機能	9
4.1.1	query_advisor_index_recommendations (min_filter、min_selectivity)	9
第 5 章	postgres ユーザで実行	11
第 6 章	postgres ユーザで実行	13
6.1	ビュー	13
6.1.1	query_advisor_qualstat	13
6.1.2	query_advisor_qualstats_pretty	14
第 7 章	ro ユーザで実行	15

EDB Query Advisor は、WHERE ステートメント、JOIN 句、およびワークロードクエリで見つかった述語に関する統計を保持することにより、インデックスの推奨事項を提供する Postgres の拡張機能です。仮想インデックスでクエリを再計画することにより、実際のワークロードクエリに対する費用対効果を分析します。仮想インデックスは、ワークロードクエリから収集された述語に基づいて作成されます。

この拡張機能は、考えられるすべてのインデックスを作成せずに、ワークロードクエリに利益をもたらす最も有用なインデックスを分析する場合に役立ちます。

述語コレクションは、クエリ内の既知のパターンを探すことによって機能します。これらのパターンには以下が含まれます。

- `BinaryOpExpr` - 少なくとも一方はテーブルの列です。可能な場合は常に、述語がスワップされ、`CONST OP VAR` 式が `VAR COMMUTED_OP CONST` に変わります。AND 式および OR 式のメンバーは、`WHERE column1 = 2`, `WHERE column1 = column2`, `WHERE 3 = column3` など、個別のエントリとしてカウントされます。
- `ScalarArrayOpExpr` - `ScalarArrayOpExpr` 左辺が VAR、右辺が配列定数です。これらは、配列内の要素ごとに 1 回カウントされます。たとえば、`WHERE column1 IN (2, 3)` は、`(column1, '=')` 演算子のペアの 2 回のオカレンスとしてカウントされます。
- `BooleanTest` - 式は、`WHERE column1 IS TRUE` などの単純なブール列参照です。WHERE columns1, WHERE NOT column1 のような句は現在処理されていません。

この拡張機能は、ワークロードテーブルで実行される個別の queryid ごとに、最初のクエリテキストをそのまま保存します。

Postgres サーバーを再起動しても、収集されたデータは保存されません。

インデックス推奨を生成するために使用する `query_advisor_index_recommendations` ファンクションの説明については、*Using EDB Query Advisor* を参照してください。

制限事項

- 単一および 2 列のインデックスのみが考慮されます。
- ユーティリティコマンドは統計を取得しません。
- 述語とワークロードエントリ数は静的です。変更には再起動が必要です。

第1章 EDB Query Advisor release notes

EDB Query Advisor のドキュメントには、マイナーリリースとパッチを含む EDB Query Advisor の最新バージョンが記載されています。これらのリリースノートには、各リリースの新機能が記載されています。マイナーリリースまたはパッチリリースで導入された新機能については、その機能を導入したリリースに関するコンテンツにインジケータもあります。

Version	Release Date
<i>Release notes for Query Advisor version 1.0.0</i>	2023 May 10

1.1 Release notes for Query Advisor version 1.0.0

このリリースのクエリアドバイザーには以下が含まれます。

Type	Description
Feature	これは、最初のリリースです。

第2章 Installing Query Advisor

EDB Query Advisor は、使用している Postgres ディストリビューションと同じプラットフォームでサポートされています。EDB Query Advisor のサポートは Postgres 12 から始まります。詳細については、以下を参照してください。

- [EDB Postgres Advanced Server Product Compatibility](#)
- [PostgreSQL Product Compatibility](#)
- [EDB Postgres Extended Server Product Compatibility](#)

2.1 前提条件

インストールプロセスを開始する前に：

- Postgres をインストールします。参照： - [Installing EDB Postgres Advanced Server](#)
- [Installing PostgreSQL](#)
- [Installing EDB Postgres Extended Server](#)
- リポジトリをセットアップします。

リポジトリのセットアップは1回限りのタスクです。リポジトリを既にセットアップしている場合、この手順を実行する必要はありません。

リポジトリをセットアップするには、

[EDB repositories](#) に移動し、そこに記載されている指示に従います。

2.2 パッケージをインストールする

package install コマンドの構文は次のとおりです。

```
sudo <package-manager> -y install edb-<postgres><postgres_version>-query-advisor
```

そこで：

- `<package-manager>` は、オペレーティングシステムで使用するパッケージマネージャーです。

Package manager	Operating system
dnf	RHEL 8 and derivatives
yum	RHEL 7 and derivatives, CentOS 7
zypper	SLES
apt-get	Debian and derivatives

- <postgres> は、使用している Postgres のディストリビューションです。

Postgres distribution	Value
PostgreSQL	pg
EDB Postgres Advanced Server	as
EDB Postgres Extended Server	postgresextended

- <postgres_version> は、使用している Postgres のバージョンです。

たとえば、RHEL 8 プラットフォームに EDB Postgres Advanced Server 15 の EDB Query Advisor をインストールするには：

```
sudo dnf -y install edb-as15-query-advisor
```

- – title: EDB クエリアドバイザーの構成 navTitle: 構成 –

EDB Query Advisor を実行するには、それを `shared_preload_libraries` に追加し、データベースに拡張機能を作成する必要があります。

1. `postgresql.conf` ファイルで、`query_advisor` を `shared_preload_libraries` パラメーターに追加します。

```
shared_preload_libraries = query_advisor
```

注釈: `shared_preload_libraries` に他の拡張子がある場合、`query_advisor` をリストに追加できます。順番は関係ありません。 !!!

1. Postgres を再起動します。

1. データベースに EDB Query Advisor 拡張機能を作成します。

```
CREATE EXTENSION query_advisor;
```

第3章 パラメーター

次のカスタム GUC は、EDB Query Advisor 拡張機能の動作を制御します。

これらのパラメーターを変更する場合は、Postgres をリロードして変更を適用します。

- `query_advisor.enabled` — `query_advisor` を有効にするかどうかを指定します。ブール値を取ります。デフォルトは `true` です。
- `query_advisor.sample_rate` — サンプルングするクエリの割合を設定します。double 値を取ります。たとえば、0.1 は、10 個のクエリのうち 1 個のみをサンプルングすることを示します。デフォルトは -1 で、自動を指定し、 $1 / \text{max_connections}$ の値になります。

これらのパラメーターを変更する場合は、Postgres を再起動して変更を適用します。

- `query_advisor.max_qual_entries` — 追跡される述語の最大数を設定します。デフォルトは 100000 です。
- `query_advisor.max_workload_entries` — 追跡するワークロードクエリの最大数を設定します。デフォルトは 100000 です。
- `query_advisor.max_workload_query_size` — ワークロードクエリの最大サイズを設定します。デフォルトは 1024 バイトです。

第4章 Using EDB Query Advisor

EDB Query Advisor は、関数とビューを定義します。

4.1 機能

4.1.1 `query_advisor_index_recommendations` (`min_filter`、`min_selectivity`)

この関数は、グローバルインデックスの提案を実行します。

デフォルトでは、少なくとも 1000 行、平均して 30 %の行をフィルタリングする述語のみが考慮されます。`min_filter` および `min_selectivity` パラメーターを使用して、デフォルトをオーバーライドできます。

この関数は、収集する述語に基づいて、1 列および 2 列のインデックス候補を生成します。各候補に関して仮想インデックスが存在する場合、関連するすべてのワークロードクエリを再計画します。ワークロードに最大の価値をもたらすインデックスのリストを推奨します。また、ワークロードクエリの推定インデックスサイズとコスト削減の割合も示しています。サイズと利益率に基づいて、どのインデックスが最も役立つかを判断できます。

例：

第5章 postgres ユーザで実行

```
select * from query_advisor_index_recommendations(0,0);
__OUTPUT__
|                                     index                                     | estimated_size_in_
| bytes | estimated_pct_cost_reduction |
+-----+-----+-----+
| CREATE INDEX ON nation USING btree (n_name); | 8192
| 14.619857 |
| CREATE INDEX ON supplier USING btree (s_suppkey,s_nationkey); | 933888
| 0.63003576 |
| CREATE INDEX ON partsupp USING btree (ps_suppkey); | 50159616
| 13.254544 |
(3 rows)
```

この関数は、式ハッシュで識別されるすべての修飾子のカウントを返します。このハッシュは、各式を識別します。

Qualifier	Description
userid	クエリを実行したユーザーの ID。
dbid	クエリが実行されたデータベースの ID。
lrelid, lattnum	リレーションの ID と左側の VAR の属性番号（ある場合）。
opno	式で使用する演算子の ID
rrelid, rattnum	リレーションの ID と右側の VAR の属性番号（ある場合）。
qualid	親 AND 式がある場合、その正規化された識別子。この識別子は、定数を除外して計算されます。この修飾子は、一緒に使用される述語を識別するのに役立ちます。
uniquequalid	親 AND 式がある場合、それを表す一意識別子。この識別子は、定数を含めて計算されます。
qualnodeid	この単純な述語の正規化された識別子。この識別子は、定数を除外して計算されます。
uniquequalnodeid	この単純述語を表す一意識別子。この識別子は、定数を含めて計算されます。

次のページに続く

表 1 – 前のページからの続き

Qualifier	Description
occurrences	この述語が呼び出された回数、つまり、関連するクエリ実行の数。
execution_count	この述語が実行された回数、つまり処理された行数。
nbfiltered	この述語が破棄されたタプルの数。
queryid	クエリ識別子。
eval_typ	評価タイプ： スキャン後に評価される述語の場合は f 、インデックス述語の場合は i 。

例：

第6章 postgres ユーザで実行

```
select * from query_advisor_qualstats();
```

__OUTPUT__

userid	dbid	lrelid	lattnum	opno	rrelid	rattnum	qualid	uniquequalid
qualnodeid	uniquequalnodeid	occurrence	execution_count	nbfiltered	min_err_estimate_ratio	max_err_estimate_ratio	mean_err_estimate_ratio	stddev_err_estimate_ratio
min_err_estimate_num	max_err_estimate_num	mean_err_estimate_num	stddev_err_estimate_num	queryid	eval_type			
10	5	16501	1	96				
262905824	262905824	1	3000000	2999997	3			
	3		3	0				
	2		2		2			0
	-3469822585968758916	f						

格納されているすべてのクエリテキストを返します。

ワークロードと修飾子ハッシュテーブルの使用率を返します。

内部カウンターをリセットし、遭遇したすべての修飾子とワークロードエントリーについて忘れます。

6.1 ビュー

この拡張機能は、`pg_qualstats` 関数の上に次のビューを定義します。

6.1.1 query_advisor_qualstat

現在のデータベースで `query_advisor_qualstats()` への呼び出しをフィルタリングします。

6.1.2 query_advisor_qualstats_pretty

適切な結合を実行して、query_advisor_qualstats ビューのすべての属性の読み取り可能な集計フォームを表示します。

例：

