

# システム利用でこんなお悩みはありませんか？

- ・性能劣化の問題を改善・防止したい
- ・画面操作後の動作パフォーマンスが遅いので改善したい

- ・障害発生に対して、後手ではなく先手に対応したい
- ・問題解決までの改善スピードを向上したい
- ・DB チューニングを簡易的に最適化したい



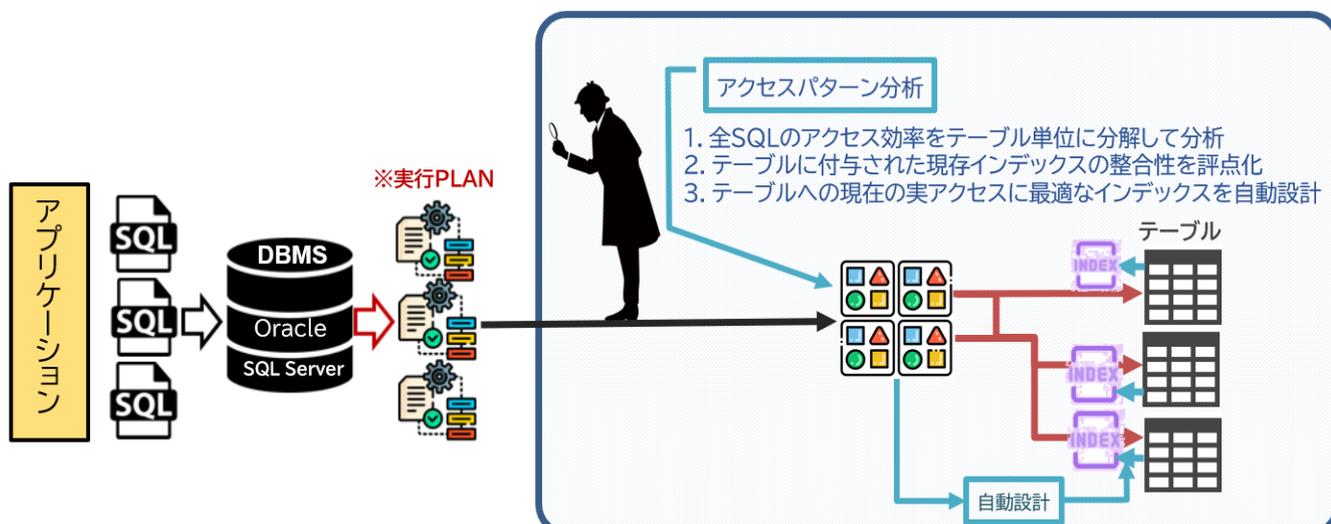
- ・ハードウェアの設備投資やクラウド費用、ライセンス数を適正にしたい
- ・DB 対応の属人化を回避したい

## 「MAJESTY」アクセスパターン分析技法について

これらのお悩みについては、アプリケーションの改修で改善できない場合、データベースのパフォーマンスに起因する問題が大半となります。このような問題に対して、迅速にデータベースの診断・問題解決するツールが「MAJESTY」となります。日米欧中で特許取得の新技术「アクセスパターン分析」は、データベースの安定運用とコスト削減を同時に実現する最新の解決手法となります。国内導入実績もトップ企業 80 社以上に採用されています。

### アクセスパターン分析(特許技術)とは？

アクセスパターン分析とは SQL の実際の動きである実行 PLAN が、テーブルから目的のデータを取得した挙動の全てを収集・分析し、テーブル単位に効率性を可視化する分析手法です。専門家が多くの時間を割いて分析してきた実行 PLAN を、自動でわかり易く可視化することで、客観的なデータに基づいた分析や改善が可能となります。



# DB 運用の課題と解決策について

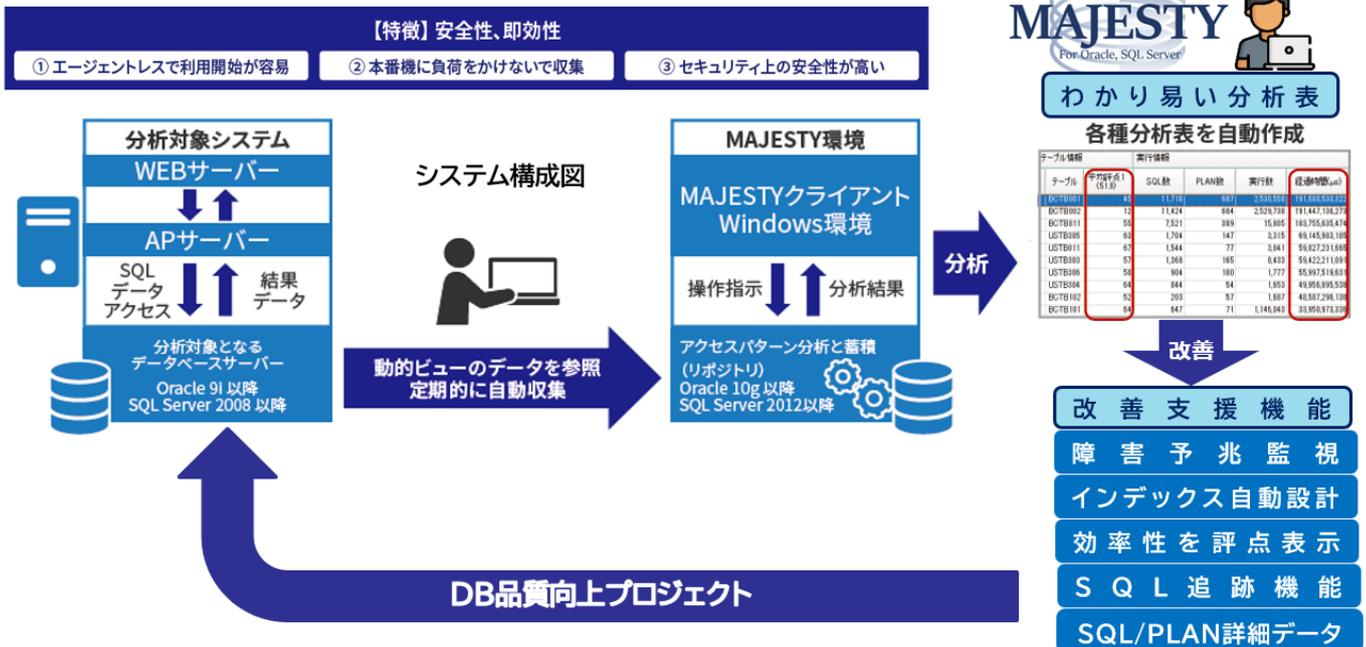


## アクセスパターン分析が提供する解決策

1. すべての SQL をシステムに負荷を掛けずに自動収集し分かりやすく分析・可視化するので**専門家に依存せずに改善**
2. SQL の動きをテーブル単位で纏めて分析することで、最適なインデックスを自動設計し、既存インデックスの削除や統合を安全に行うことで、**想像を超えた DB 性能の大幅な改善**
3. 顕在的な課題のみならず潜在的な非効率も評点で可視化し全体を最適化、HW リソースの効率を高め**運用コストの削減**
4. バッチ処理など重要な SQL の予兆監視ができ、先手に対応ができることで、**障害の予防保全と改善スピードの向上**

## 「MAJESTY」の仕組みについて

全てのSQL処理(動的ビュー)を分析し、わかり易く可視化



# 「MAJESTY」機能について

## ① テーブル単位の分析と効果

- ◆ テーブル単位で見ると、全体経過時間の9割が上位20個のテーブル改善で、全体の9割を最適化！
- ◆ テーブル、SQL単位で、適正なアクセスルートになっているかを評点で見える化することで、問題点をすぐにキャッチアップ！

※経過時間1番のテーブル：11,716個のSQLが51個のアクセスパターンに纏められ、平均評点が45点/100点

新しいアプローチ	テーブル情報		アクセスパターン情報					実行情報					改善優先順位	
	テーブル	タイプ	パーティション	パターン数	ルート数	パス数	平均評点1 (51.9)	平均評点2 (62.4)	SQL数	PLAN数	実行数	経過時間(μs)	経過時間(μs) (AVG)	
BCTB001	NO	5	47	10	45	48	11,716	607	2,530,558	191,688,533,322	75,679			
BCTB002	NO	164	156	11	12	49	11,424	604	2,529,738	191,447,136,273	75,679			
BCTB811	NO	9	9	3	55	91	7,521	389	15,805	103,755,635,474	6,564,735			
USTB305	NO	6	6	3	63	52	1,704	147	3,315	69,145,983,185	20,858,517			
USTB011	NO	6	5	3	67	60	1,544	77	3,041	59,827,231,665	19,673,539			
USTB303	NO	8	7	5	57	94	1,368	165	8,433	59,422,211,091	7,046,391			
USTB306	NO	6	6	2	58	45	904	100	1,777	55,997,519,631	31,512,391			
USTB304	NO	6	6	3	64	2	844	54	1,653	49,956,095,538	30,221,473			
BCTB102	NO	25	22	8	52	43	203	57	1,687	48,587,296,130	28,801,005			
BCTB101	NO	54	49	13	64	100	647	71	1,146,043	33,950,973,336	29,625			

## ② インデックス自動設計とシミュレーション

- ◆ アクセスパターン情報を基にテーブル一括でインデックスの自動設計が可能！
- ◆ 自動設計前・後をシミュレーションし改善効果を素早く把握が可能！

改善インデックス

改善シミュレーション

自動設計ボタン

自動設計後										自動設計前									
アクセスルート	INDEX_NAME	Selectivity	Cardinality	評点	SQL数	PLAN数	実行	アクセスルート	アクセスパス	Selectivity	Cardinality	評点	SQL数	PLAN数	実行				
1 (YMD=>#)	INDEX_01	10.001%	6,166,135	0	1	1	1	1 (YMD=>#)	BCTB811 TABLE ACCESS FULL	10.001%	6,166,135	0	1	1	1				
2 NONE	INDEX_02	100	61,661,344	0	3	3	3	2 NONE	BCTB811 TABLE ACCESS FULL	100.000%	61,661,344	0	3	3	3				
3 (IDX=>)	INDEX_05	0.002%	1,011	100	1	1	1	3 (NAL_TENPO_CD=>XNAL_TENPO_CD=>#)	BCTB811 SK2 INDEX SKIP SCAN	0.002%	59,924	75	1	1	1				
4 (NAL_TENPO_CD=>XNAL_TENPO_CD=>#)	INDEX_02	0.002%	59,924	100	1	1	1	4 (NAL_TENPO_CD=>XNAL_TENPO_CD=>#)	BCTB811 INDEX SKIP SCAN	0.001%	441	75	10	4	4				
5 (NAL_TENPO_CD=>XSHOHN_CD=>#)	INDEX_02	0.001%	441	100	10	4	4	5 (SHOHN_CD=>XSHOHN_CD=>#)	BCTB811 INDEX SKIP SCAN	0.735%	453,393	75	1	1	1				
6 (NAL_TENPO_CD=>XSHOHN_CD=>XKYM=>)	INDEX_01	0.001%	100	100	34	8	8	6 (IDX=>)	BCTB811 SK1 INDEX RANGE SCAN	0.002%	1,011	100	1	1	1				
7 (NAL_TENPO_CD=>XKYM=>)	INDEX_01	0.001%	100	100	2	2	2	7 (NAL_TENPO_CD=>XSHOHN_CD=>XKYM=>)	BCTB811 INDEX UNBUQ SCAN	0.001%	1	100	34	8	8				
8 (SHOHN_CD=>XSHOHN_CD=>#)	INDEX_04	0.736%	453,393	100	1	1	1	8 (NAL_TENPO_CD=>XKYM=>)	BCTB811 INDEX RANGE SCAN	0.001%	55	100	2	2	2				
9 (YMD=>)	INDEX_03	2.381%	1,468,128	100	3	3	3	9 (YMD=>)	BCTB811 SK2 INDEX RANGE SCAN	2.381%	1,468,128	100	3	3	3				
10 (YMD=>)	INDEX_01	0.001%	55,652	100	3	7	7	10 (YMD=>)	BCTB811 SK3 INDEX RANGE SCAN	0.001%	55,652	100	3	7	7				

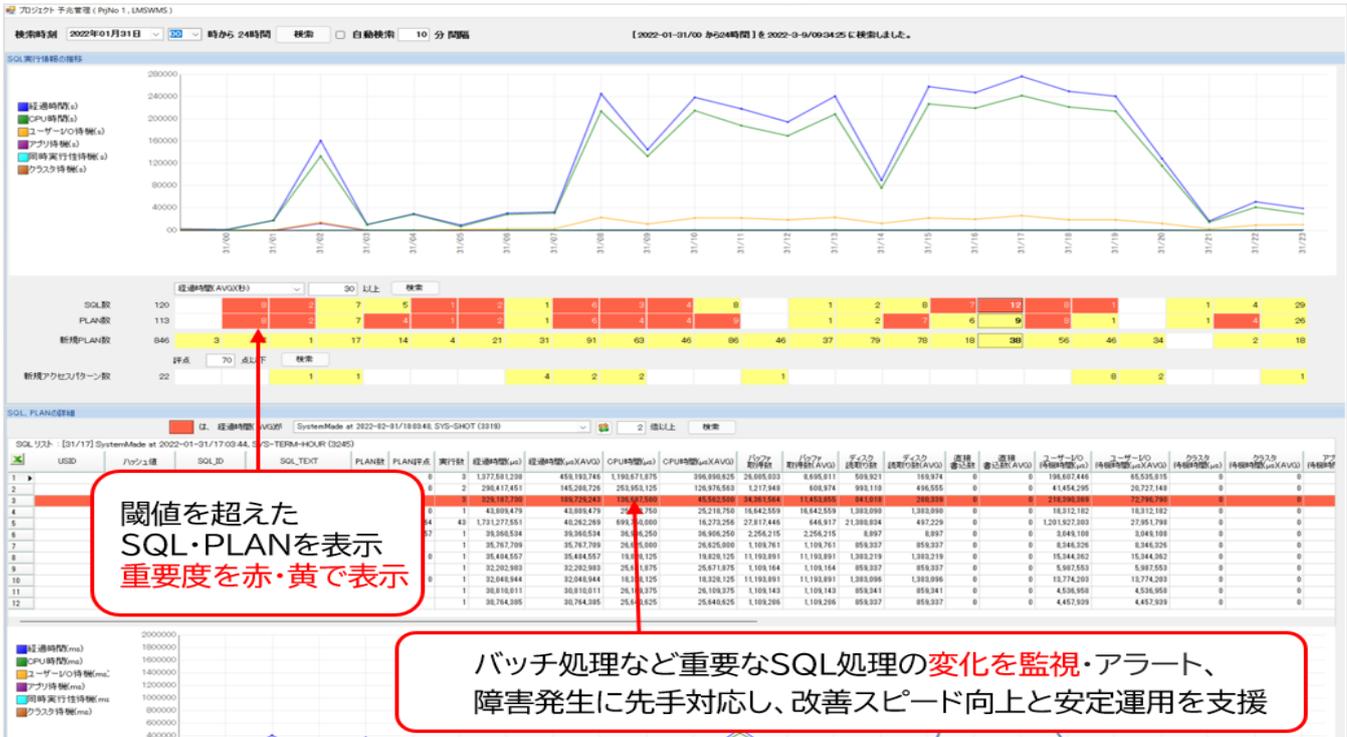
自動設計前・後の改善効果をシミュレーションし、比較して確認できる

既存アクセスパターン

既存インデックス

### ③ 予兆監視で障害に先手対応

◆ 監視で重要な、SQL 処理の変化の監視を「差分分析」の機能で実現。



## MAJESTY・基本性能分析サービスの提供の流れ

