

Adaptive Server Anywhere のロック制限

このマニュアルでは、数多くのローをトランザクションで操作する場合に発生する可能性があるエラーの原因と、その状況の回避策について説明します。

問題

非常に多くのローを含んだテーブルを持つデータベースでは、数多くの (何百万もの) ローをトランザクションで操作するとエラーが発生することがあります。これらのエラーは、ローの比較的小さなセットを操作するときには発生しません。たとえば、他のユーザがデータベースに接続していなくても、INSERT INTO table SELECT を実行すると以下のエラーが返されることがあります。

```
-210: "<table> のローは、ユーザ '別のユーザ' によってロックされています。"
```

説明

特定のテーブルに対するロックを管理するために使用するデータ構造には、エントリ可能な数の制限があります。この制限はページ・サイズをもとにした範囲です。各ページ・サイズに対するエントリの概数は以下のとおりです。

ページサイズ	制限
1024	500 万
2048	1100 万
4096	2200 万
8192	4400 万

回避策

最も単純な回避策は、トランザクションを分割し、より少ない数のローをロックすることです。これには、同時実行性の向上、ロールバック・ログのサイズの減少、トランザクション中にシステムが応答を停止した場合のリカバリ時間の減少という別の利点があります。

場合によっては、トランザクションの分割が容易でないことや望ましくないことがあります。その場合は、テーブルを明示的にロックすることを検討してください。SUBSUME_ROW_LOCKS オプションが ON (デフォルト) の場合、LOCK TABLE T IN EXCLUSIVE MODE でテーブル T が排他的にロックされていると、データベース・サーバはテーブル T の個々のロー・ロックを取得する必要がなくなります。これにより、単一のトランザクションで広範な更新が T に対して行われる場合はパフォーマンスが大幅に向上する可能性があります (特に、キャッシュ・サイズに比べて T が大きい場合)。

SUBSUME_ROW_LOCKS を ON に設定した場合の 1 つの欠点は、この方法でロックされたテーブルに対するキーセット・カーソルが期待どおりに機能しないということです。たとえば、これらのキーセット・カーソルは、変更されたデータベース内のローについてカーソル内のローごとに "row changed (ローが変更されました)" という警告を返すことがあります。また、データベース・サーバは、ORDER BY を使用した更新可能カーソルをキーセット・カーソルに変えることがあります。