

# DateTime-Bestellproblem aus SQL-Abfrageergebnissen mit Upgrade von SQL Version 6.5 auf 7.0

## Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Problem](#)

[Lösung](#)

[Zugehörige Informationen](#)

## Einführung

Dieses Dokument beschreibt, warum die SQL-Abfrageergebnisreihenfolge nach **DateTime** für die `Route_Call_Detail` oder die `Termination_Call_Detail`-Tabelle zwischen Microsoft SQL Version 6.5 und 7.0 unterschiedlich ist und eine Problemumgehung in einer Cisco Intelligent Contact Management (ICM)-Umgebung bietet.

## Voraussetzungen

### Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- Cisco ICM
- Microsoft SQL

### Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- Cisco ICM
- Microsoft SQL Server Version 6.5 und 7.0

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie

die potenziellen Auswirkungen eines Befehls verstehen.

## Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter [Cisco Technical Tips Conventions](#) (Technische Tipps zu Konventionen von Cisco).

## Problem

Wenn Sie ein Upgrade auf Cisco ICM Version 4.6.2 oder höher durchführen, wird Microsoft SQL Server Version 6.5 auf Version 7 aktualisiert. Nach dem Upgrade gibt die Ausführung der SQL-Abfrage in der `Route_Call_Detail`- oder der `Termination_Call_Detail`-Tabelle auf dem ICM-System, das SQL Version 7 ausführt, andere Ergebnisse als die SQL-Version 6.5 zurück. Siehe diese [SQL-Abfrage](#):

Abbildung 1: Microsoft SQL Server-Abfrage

```
SELECT      DialedNumberID,
            Day,
            RouterCallKey,
            RouteID,
            DateTime,
            RequestType,
            RoutingClientID,
            OriginatorType,
            RoutingClientCallKey,
            Priority,
            MsgOrigin,
            ScriptID
            RecoveryDay,
            RecoveryKey,
            TimeZone,
            RouterCallKeySequenceNumber

From  t_Route_Call_Detail

Where DateTime between '2004-05-12 06:00:00.827' and '2004-05-12 08:23:35.827'
```

Wenn Sie die Ergebnisse der Ausführung derselben SQL-Abfrage auf dem alten ICM-System vergleichen, auf dem SQL Version 6.5 ausgeführt wird, ist der Inhalt identisch. Die neuen Ergebnisse befinden sich jedoch nicht in der aufsteigenden **DateTime**-Reihenfolge wie die ursprünglichen Ergebnisse. Vor dem Upgrade gab diese Abfrage Daten in **DateTime**-Reihenfolge zurück. Seit der Aktualisierung werden Daten nicht in der **DateTime**-Reihenfolge zurückgegeben, wie [hier](#) gezeigt.

Abbildung 2: SQL-Abfrageergebnisse in DateTime-Reihenfolge

2004-05-12 07:34:11.327

2004-05-12 07:34:11.327

2004-05-12 07:34:11.827

2004-05-12 07:34:11.827

2004-05-12 07:34:11.827

2004-05-12 07:34:11.827

2004-05-12 07:34:12.327

**Out of Order**

2004-05-12 06:04:31.827

2004-05-12 06:04:33.327

2004-05-12 06:04:33.827

2004-05-12 06:04:33.827

2004-05-12 06:04:34.327

2004-05-12 06:04:34.327

2004-05-12 06:04:34.327

## Lösung

Nach dem Upgrade von SQL Version 6.5 auf Version 7.0 befinden sich die Ergebnisse der ausgewählten Abfragen, die im Rahmen von `Route_Call_Detail` oder `Termination_Call_Detail` abgeschlossen wurden, nicht mehr in der **DateTime**-Reihenfolge. Um die **DateTime**-Ergebnisse zu erhalten, muss eine **Order by**-Klausel eingefügt werden. Dies ist ein Problem, da die **Order by**-Klausel erhebliche Overhead zur `Route_Call_Detail`- und `Termination_Call_Detail`-Abfragen hinzufügen kann, wodurch sehr große Resultsets entstehen können.

Die Bestellung des Primärschlüssels in SQL Server Version 6.5 stammt aus dem alten Sybase-System, von dem Microsoft SQL stammt. Microsoft hat die Konformität mit dem SQL-Standard in SQL Server Version 7.0 verschärft, wodurch eine Bestellung ohne eine **Order by**-Klausel in der SQL-Abfrage nicht garantiert wird. Dies ist eine relationale Datenbank, keine physische sequenzielle Datei. In einer relationalen Datenbank gibt es keine angenommenen Sequenzen, wie es in einer physischen sequenziellen Datei der Fall ist. Daher ist es notwendig, eine **order by**-Klausel zu verwenden, um eine Sequenz im Ergebnis festzulegen.

**Hinweis:** Dies ist kein Cisco Problem. Es handelt sich um ein Standardproblem von Microsoft SQL Server.

## Zugehörige Informationen

- [Technischer Support und Dokumentation - Cisco Systems](#)