



データベース テーブル

この章では、IM and Presence Service ノードをサポートするために、スキーマで作成される外部データベースのテーブルについて説明します。



(注) IM and Presence Service は、デフォルトで外部データベースに 27 個のテーブルを生成しますが、ここでは、このモジュールで説明されるテーブルのみを使用します。



(注) 外部データベースのデータを変更する必要がある場合は、変更後に Cisco XCP Text Conference Manager サービスを再起動してください。

- [AFT_LOG テーブル \(1 ページ\)](#)
- [TC_ROOMS テーブル \(4 ページ\)](#)
- [TC_USERS テーブル \(6 ページ\)](#)
- [TC_MESSAGES テーブル \(7 ページ\)](#)
- [TC_TIMELOG テーブル \(8 ページ\)](#)
- [TC_MSGARCHIVE テーブル \(9 ページ\)](#)
- [JM テーブル \(11 ページ\)](#)

AFT_LOG テーブル

AFT_LOG テーブルには、Cisco Unified Communications Manager IM and Presence Service のマネージド ファイル転送機能の使用中に発生したファイル転送に関する情報が格納されます。

Indexes: "aft_log_pkey" PRIMARY KEY, btree (aft_index)

列名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
AFT_INDEX	BIGINT	NUMBER (19)	bigint	○	トランザクションを識別するシーケンス番号。

AFT_LOG テーブルのサンプル SQL クエリ

列名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	ファイルをアップロードまたはダウンロードしたユーザの Jabber ID (JID)。この列の内容は、METHOD 列の内容によって異なります。 <ul style="list-style-type: none"> • METHOD 列に "POST" が含まれている場合は、ファイルをアップロードしたユーザの JID になります。 • METHOD 列に "GET" が含まれている場合は、ファイルをダウンロードしたユーザの JID になります。
TO_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	ファイル転送の意図された受信者であるユーザ、グループチャット、またはパーシステントルームの JID。
METHOD	VARCHAR (63)	VARCHAR2 (63)	nvarchar(63)	○	この列には、ユーザがファイルをアップロードしたことを示す POST か、ユーザがファイルをダウンロードしたことを示す GET のどちらかを含めることができます。
FILENAME	VARCHAR (511)	VARCHAR2 (511)	nvarchar(511)	○	アップロードまたはダウンロードされたファイルのリソース名。リソース名は HTTP 要求内のファイルを識別します。IM and Presence Service で自動生成されます。
REAL_FILENAME	VARCHAR (511)	VARCHAR2 (511)	nvarchar(511)	○	ユーザによってアップロードされたファイルの実際の名前。
FILE_TYPE	VARCHAR (10)	VARCHAR2 (10)	nvarchar(10)	○	jpg、txt、pptx、docx などのファイル拡張子。
CHAT_TYPE	VARCHAR (10)	VARCHAR2 (10)	nvarchar(10)	○	1 対 1 の IM 会話中にファイルが転送された場合は "im"。 アドホック グループチャットの会話中にファイルが転送された場合は "groupchat"。 ファイルがパーシステントチャットルームに転送された場合は "persistent"。
FILE_SERVER	VARCHAR (511)	VARCHAR2 (511)	nvarchar(511)	○	ファイルが保存されるファイルサーバのホスト名または IP アドレス。
FILE_PATH	VARCHAR (511)	VARCHAR2 (511)	nvarchar(511)	○	ファイルサーバ上のファイルへの絶対パス (ファイル名を含む)。リポジトリに保存されるファイル名は、一意で、IM and Presence Service によって自動生成されます。
FILESIZE	BIGINT	NUMBER (19)	bigint	○	バイト単位のファイルサイズ。
BYTES_TRANSFERRED	BIGINT	NUMBER (19)	bigint	○	転送されたバイト数。この数値は、転送中にエラーが発生した場合にだけ FILESIZE と異なります。
TIMESTAMPVALUE	TIMESTAMP	TIMESTAMP	timestamp	○	ファイルがアップロードまたはダウンロードされた日付と時刻 (UTC)。

AFT_LOG テーブルのサンプル SQL クエリ

ここでは、特定の情報を抽出するために AFT_LOG テーブルで実行可能なサンプル SQL クエリを紹介します。

すべてのアップロードされたファイル

次の SQL クエリは、マネージドファイル転送機能を使用してアップロードされたすべてのファイルと画面キャプチャのレコードを返します。

```
SELECT file_path
FROM aft_log
WHERE method = 'Post';
```

特定の受信者にアップロードされたすべてのファイル

次の SQL クエリは、マネージドファイル転送機能を使用して、ユーザ <userid> にアップロードされたすべてのファイルと画面キャプチャのレコードを返します。



- (注) ダウンロードされたファイルと画面キャプチャのレコードには *to_jid* フィールドにデータが含まれていません。

```
SELECT file_path
FROM aft_log
WHERE to_jid = '<userid>@<domain>';
```

特定の送信者によってアップロードされたすべてのファイル

次の SQL クエリは、ユーザ <userid> がマネージドファイル転送機能を使用してアップロードしたすべてのファイルと画面キャプチャのレコードを返します。

```
SELECT file_path
FROM aft_log
WHERE jid LIKE '<userid>@<domain>%' AND method = 'Post';
```

特定のユーザによってダウンロードされたすべてのファイル

次の SQL クエリは、ユーザ <userid> がマネージドファイル転送機能を使用してダウンロードしたすべてのファイルと画面キャプチャのレコードを返します。

```
SELECT file_path
FROM aft_log
WHERE jid LIKE '<userid>@<domain>%' AND method = 'Get';
```

IM 会話中にアップロードまたはダウンロードされたすべてのファイル

次の SQL クエリは、マネージドファイル転送機能を使用した IM 会話でアップロードまたはダウンロードされたすべてのファイルと画面キャプチャのレコードを返します。

```
SELECT file_path
```

```
FROM aft_log
WHERE chat_type = 'im';
```

特定の時間の経過後に特定のユーザによってアップロードされたすべてのファイル

次の SQL クエリは、特定の時間の経過後にユーザ <userid> がマネージドファイル転送機能を使用してアップロードしたすべてのファイルと画面キャプチャのレコードを返します。

```
SELECT file_path
FROM aft_log
WHERE jid LIKE '<userid>@<domain>%' AND method = 'Post' AND timestampvalue > '2014-12-18
11:58:39';
```

AFT_LOG テーブルの SQL クエリの出力例

これらのクエリのいずれかからの出力は次のようになります。

```
/opt/mftFileStore/node_1/files/im/20140811/15/file_name1
/opt/mftFileStore/node_1/files/im/20140811/15/file_name2
/opt/mftFileStore/node_1/files/im/20140811/15/file_name3
/opt/mftFileStore/node_1/files/im/20140811/15/file_name4
...
/opt/mftFileStore/node_1/files/im/20140811/15/file_name99
/opt/mftFileStore/node_1/files/im/20140811/15/file_name100
```

外部ファイルサーバをクリーンアップするための出力の使用

この出力と **rm** コマンドを使用して、外部ファイルサーバから不要なファイルを削除することができます。たとえば、外部ファイルサーバ上で次のコマンドを実行できます。

```
rm /opt/mftFileStore/node_1/files/im/20140811/15/file_name1
rm /opt/mftFileStore/node_1/files/im/20140811/15/file_name2
rm /opt/mftFileStore/node_1/files/im/20140811/15/file_name3
その他
```

TC_ROOMS テーブル

TC_ROOMS テーブルには、グループ チャット ルームの情報が含まれます。

カラム名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
ROOM_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	ルームの ID です。
CREATOR_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	ルームを作成したユーザの ID です。
件名	VARCHAR (255)	VARCHAR2 (255)	nvarchar(255)	○	ルームの現在の件名です。
TYPE	VARCHAR (32)	VARCHAR2 (32)	nvarchar(32)	○	制約 check_type です。この値は、「ad-hoc」または「persistent」のいずれかである必要があります。
CONFIG	TEXT	CLOB	nvarchar(MAX)	Yes	ルームが最後に設定された時刻からのパケット全体です。この情報は、ルームがもう一度作成される場合に（例：起動時）、ルームを再設定できるようにします。
SPACKET	TEXT	CLOB	nvarchar(MAX)	Yes	件名がルームに最後に設定された時刻からのパケット全体です。この情報は、ルームがもう一度作成される場合に、ルームの件名を表示できるようにします。
START_MSG_ID	BIGINT	NUMBER (19)	bigint	○	TC_MSGARCHIVE テーブルの MSG_ID カラムに入力する際に使用される連続番号です。 この値は変更しないでください。

TC_USERS テーブル

カラム名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
NEXT_MSG_ID	BIGINT	NUMBER (19)	bigint	○	TC_MSGARCHIVE テーブルの MSG_ID カラムに入力する際に使用される連続番号です。 この値は変更しないでください。

TC_USERS テーブル

TC_USERS テーブルには、役割や所属、代替名などの、グループ チャット ルームのユーザに関連付けられているデータが含まれます。

カラム名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
ROOM_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	ルームの ID です。
REAL_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	ルームでのユーザの ID です。この値は、代替名ではなく、ユーザの実際の ID です。
ROLE	VARCHAR (32)	VARCHAR2 (32)	nvarchar(32)	○	ルームでのユーザの役割です。この値は、次のいずれかに限定されます: 「none」、 「hidden」、 「visitor」、 「participant」、または 「moderator」。

カラム名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
AFFILIATION	VARCHAR (32)	VARCHAR2 (32)	nvarchar(32)	○	ルームでのユーザの所属です。この値は、次のいずれかに限定されます：「none」、「outcast」、「member」、「admin」、または「owner」。
NICK_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	ルームの ID とユーザの代替名です。形式は room@tc-server/nick です。
REASON	VARCHAR (255)	VARCHAR2 (255)	nvarchar(255)	○	ユーザの所属が最後に変更されたときに入力した理由です。
INITIATOR_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	設定変更が行われたルームの ID です。

TC_MESSAGES テーブル

TC_MESSAGES テーブルには、グループチャットルームで送信されるメッセージが含まれません。

カラム名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
MSG_ID	BIGINT	NUMBER (19)	bigint	○	メッセージの ID です。MSG_ID はチャットルームごとの各メッセージの一意な識別子です。これは、グローバルに一意ではありません。

TC_TIMELOG テーブル

カラム名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
ROOM_JID	VARCHAR (3071)	VARCHAR (3071)	nvarchar(3071)	○	メッセージの送信先となったルームの ID です。
STAMP	TIMESTAMP	TIMESTAMP	datetime	○	メッセージが送信された日時です。
MSG	TEXT	CLOB	nvarchar(MAX)	Yes	メッセージ全体です。

TC_TIMELOG テーブル

TC_TIMELOG テーブルには、ユーザが特定のグループチャットルームに入退室した時刻が含まれます。グループチャットの会話をもう一度作成したり、会話を表示したユーザを確定するために、このテーブルはその他の TC テーブルと併用される場合があります。

カラム名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
REAL_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	ルームに入っているユーザ、またはルームから離れているユーザの ID です。
NICK_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	ルームの ID とユーザの代替名です。形式は room@tc-server/nick です。
DIRECTION	VARCHAR (1)	VARCHAR2 (1)	nvarchar(1)	○	ユーザがルームに入った (E) か、ルームから離れた (L) かを示します。値は、「E」および「L」に限定されます。

カラム名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
STAMP	TIMESTAMP	TIMESTAMP	datetime	○	ユーザがルームに入った日時、またはルームから離れた日時です。IMP サーバからの UTC 形式です。

TC_MSGARCHIVE テーブル

TC_MSGARCHIVE テーブルには、グループチャットルームのメッセージと関連情報が保管されます。



- (注) IM and Presence Service でグループチャットをオンにすると、すべてのメッセージがこのテーブルにアーカイブされます。**Cisco Unified CM IM and Presence Administration** のユーザインターフェイスで、[Archive all room messages (すべてのルームメッセージをアーカイブする)] オプションを選択します。[Messaging (メッセージング)] > [Conferencing and Persistent Chat (会議と永続的なチャット)] を選択します。グループチャット機能については、『*Configuration and Administration of IM and Presence Service on Cisco Unified Communications Manager*』を参照してください。

カラム名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
MSG_ID	BIGINT	NUMBER (19)	bigint	○	メッセージの一意の識別子です。
TO_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	メッセージを受信したルームの ID です。
FROM_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	メッセージを送信したユーザの ID です。
NICK_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	ルームの ID と送信者の別名です。 例： <code>room@conference.plmnick</code>

カラム名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
SENT_DATE	TIMESTAMP	TIMESTAMP	datetime	○	メッセージが送信された日付です。IMP サーバからの UTC 形式です。
MSG_TYPE	VARCHAR (1)	VARCHAR2 (1)	nvarchar(1)	○	メッセージのタイプ属性の最初の文字です。可能な値は、「c」（チャット）、「n」（標準）、「g」（グループチャット）、「h」（見出し）、および「e」（エラー）です。
BODY_LEN	INT	NUMBER (9)	int	○	メッセージ本文の文字の長さです。
MESSAGE_LEN	INT	NUMBER (9)	int	○	件名と本文を含むメッセージの文字の長さです。
BODY_STRING	VARCHAR (4000)	VARCHAR2 (4000)	nvarchar(4000)	○	メッセージ本文です。
MESSAGE_STRING	VARCHAR (4000)	VARCHAR2 (4000)	nvarchar(4000)	○	未加工パケット全体です。
BODY_TEXT	TEXT	CLOB	nvarchar(MAX)	Yes	メッセージ本文が 4000 文字を超えている場合は、BODY_STRING フィールドの代わりにこのフィールドに保管されます。

カラム名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
MESSAGE_TEXT	TEXT	CLOB	nvarchar(MAX)	Yes	未加工パケット全体が4000文字を超えている場合は、MESSAGE_STRING列の代わりにこの列に保管されます。
SUBJECT	VARCHAR (255)	VARCHAR2 (255)	nvarchar(255)	○	ルームの現在の件名です。

JM テーブル

JM テーブルには、メッセージアーカイバコンポーネントの会話と関連情報が保存されます。メッセージアーカイバコンポーネントは、IM and Presence Service上でネイティブなコンプライアンス機能を提供します。

列名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
TO_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	アーカイブされたメッセージを送信しているユーザの Jabber ID (JID)。
FROM_JID	VARCHAR (3071)	VARCHAR2 (3071)	nvarchar(3071)	○	アーカイブされたメッセージを受信しているユーザの JID。
SENT_DATE	TIMESTAMP	TIMESTAMP	datetime	○	メッセージが送信された日付です。IMP サーバからの UTC 形式です。
SUBJECT	VARCHAR (128)	VARCHAR2 (128)	nvarchar(128)	○	アーカイブされることになるメッセージの件名です。

列名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
THREAD_ID	VARCHAR (128)	VARCHAR2 (128)	nvarchar(128)	○	アーカイブされることになるメッセージのスレッド ID です。メッセージスレッドが開始されると、IM クライアントによって値が提供され、スレッドのすべての関連メッセージにこの値が使用されます。これらの値は固有であり、関連付けられたアーカイブメッセージのグループを識別するものです。
MSG_TYPE	VARCHAR (1)	VARCHAR2 (1)	nvarchar(1)	○	メッセージのタイプ属性の最初の文字です。設定可能な値は次のとおりです。 <ul style="list-style-type: none"> • 「c」 : チャット • 「n」 : 標準 • 「g」 : グループチャット • 「h」 : 見出し • 「e」 : エラー

列名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
DIRECTION	VARCHAR (1)	VARCHAR2 (1)	nvarchar(1)	○	メッセージが「O」(発信)か、「I」(着信)かを示します。同じサーバ上のユーザ間でメッセージが送信される場合は、2回(発信としての1回と着信としての1回)記録されます。
BODY_LEN	INT	NUMBER (9)	int	○	メッセージ本文の文字の長さです。
MESSAGE_LEN	INT	NUMBER (9)	int	○	件名と本文を含むメッセージの文字の長さです。
BODY_STRING	VARCHAR (4000)	VARCHAR2 (4000)	nvarchar(4000)	○	メッセージ本文です。
MESSAGE_STRING	VARCHAR (4000)	VARCHAR2 (4000)	nvarchar(4000)	○	未加工パケット全体です。
BODY_TEXT	TEXT	CLOB	nvarchar(MAX)	Yes	メッセージ本文が4000文字を超えている場合は、 BODY_STRING フィールドの代わりにこのフィールドに保管されます。
MESSAGE_TEXT	TEXT	TEXT	nvarchar(MAX)	Yes	未加工パケット全体が4000文字を超えている場合は、 MESSAGE_STRING フィールドの代わりにこのフィールドに保管されます。

列名	Postgres のデータ型	Oracle のデータ型	Microsoft SQL のデータ型	Null 以外	説明
HISTORY_FLAG	VARCHAR (1)	VARCHAR2 (1)	nvarchar(1)	○	ルールの履歴メッセージが新しい参加者に送信される場合に使用します (既存ルールに入った場合)。これにより、ルールに実際に参加している間に受信したメッセージと、履歴送信の一部として受信したメッセージとを区別することができます。後者のメッセージタイプには、データベースで HISTORY_FLAG=「H」のフラグが付けられます。そうでない場合、この列は「N」に設定されます。

JM テーブルへのサンプル SQL クエリ

この項では、特定の情報を抽出するための、JM テーブルで実行できるサンプル SQL クエリを紹介합니다。次のクエリは、テーブルからすべての列を選択しますが、SQL クエリにどの情報を含めるかについて、さらに選択することができます。

特定のユーザによって送信されたすべてのインスタント メッセージ

次の SQL クエリは、特定のユーザによって送信されたすべてのインスタント メッセージを返します。

```
SELECT to_jid, sent_date, subject, thread_id, msg_type, direction, body_len, message_len,
body_string, message_string, body_text, message_text, history_flag
FROM jm
WHERE from_jid like 'bob@cisco.com%';
```

特定のユーザが受信するすべてのインスタントメッセージ

以下の SQL クエリは特定のユーザが受信するすべてのインスタントメッセージを返します。

```
SELECT from_jid, sent_date, subject, thread_id, msg_type, direction, body_len,  
message_len, body_string, message_string, body_text, message_text, history_flag  
FROM jm  
WHERE to_jid like 'bob@cisco.com%';
```

特定のワードを含むすべてのインスタントメッセージ

次の SQL クエリは、特定のワードを含むすべてのインスタントメッセージを返します。

```
SELECT to_jid, from_jid, sent_date, subject, thread_id, msg_type, direction, body_len,  
message_len, body_string, message_string, body_text, message_text, history_flag  
FROM jm  
WHERE LOWER(body_string) like LOWER('%hello%');
```

特定の日付以降のすべてのインスタントメッセージの会話とチャットルーム

次の SQL クエリは、特定の日付以降のすべてのインスタントメッセージの会話とチャットルームを返します。

```
SELECT to_jid, from_jid, sent_date, subject, thread_id, msg_type, direction, body_len,  
message_len, body_string, message_string, body_text, message_text, history_flag  
FROM jm  
WHERE CAST(sent_date AS Character(32)) like '2011-01-31%';
```

