



External Database Tables

This chapter provides information about the external database tables that are created in your schema to support the IM and Presence Service node.



Note

By default, the IM and Presence Service generates 27 tables in the external database but at present it only uses the tables described in this module.

- [TC_ROOMS Table, page 1](#)
- [TC_USERS Table, page 2](#)
- [TC_MESSAGES Table, page 3](#)
- [TC_TIMELOG Table, page 3](#)
- [TC_MSGARCHIVE Table, page 4](#)
- [JM Table, page 5](#)

TC_ROOMS Table

The TC_ROOMS table contains information for group chat rooms.

Column Name	Postgres Datatype	Oracle Datatype	Not Null	Description
ROOM_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The ID of the room.
CREATOR_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The ID of the user who created the room.
SUBJECT	VARCHAR (255)	VARCHAR2 (255)	Yes	The current subject for the room.

Column Name	Postgres Datatype	Oracle Datatype	Not Null	Description
TYPE	VARCHAR (32)	VARCHAR2 (32)	Yes	The constraint check_type. This value must be either “ad-hoc” or “persistent”.
CONFIG	TEXT	CLOB	Yes	The entire packet from the last time the room was configured. This information enables the room to be reconfigured when the room is recreated (for example, at start-up).
SPACKET	TEXT	CLOB	Yes	The entire packet from the last time the subject was set for the room. This information enables the room subject to be displayed when the room is recreated.
START_MSG_ID	BIGINT	NUMBER (19)	Yes	A sequence number that is used to populate the MSG_ID column in the TC_MSGARCHIVE table. Do not modify this value.
NEXT_MSG_ID	BIGINT	NUMBER (19)	Yes	A sequence number that is used to populate the MSG_ID column in the TC_MSGARCHIVE table. Do not modify this value.

TC_USERS Table

The TC_USERS table contains roles and affiliations, alternate names, and other data associated with group chat room users.

Column Name	Postgres Datatype	Oracle Datatype	Not Null	Description
ROOM_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The ID of the room.
REAL_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The ID of a user in the room. This value is the actual ID of the user, rather than an alternate name.
ROLE	VARCHAR (32)	VARCHAR2 (32)	Yes	The role of the user in the room. This value is constrained to one of the following: “none”, “hidden”, “visitor”, “participant”, or “moderator”.
AFFILIATION	VARCHAR (32)	VARCHAR2 (32)	Yes	The affiliation of the user in the room. This value is constrained to one of the following: “none”, “outcast”, “member”, “admin”, or “owner”.
NICK_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The ID of the room, plus the alternate name for the user. The format is room@tc-server/nick.

Column Name	Postgres Datatype	Oracle Datatype	Not Null	Description
REASON	VARCHAR (255)	VARCHAR2 (255)	Yes	The reason entered when the user's affiliation was last changed.
INITIATOR_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The ID of the room in which the configuration change occurred.

TC_MESSAGES Table

The TC_MESSAGES table contains messages that are sent in group chat rooms.

Column Name	Postgres Datatype	Oracle Datatype	Not Null	Description
MSG_ID	BIGINT	NUMBER (19)	Yes	The ID of the message. The MSG_ID is a unique identifier for each message per chat room; it is not globally unique.
ROOM_JID	VARCHAR (3071)	VARCHAR (3071)	Yes	The ID of the room to which the message was sent.
STAMP	TIMESTAMP	TIMESTAMP	Yes	The date and time the message was sent.
MSG	TEXT	CLOB	Yes	The entire message.

TC_TIMELOG Table

The TC_TIMELOG table contains the time that users enter and exit specific group chat rooms. This table may be used in conjunction with the other TC tables to recreate group chat conversations and to determine which users viewed the conversations.

Column Name	Postgres Datatype	Oracle Datatype	Not Null	Description
REAL_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The ID of the user who is entering or leaving the room.
NICK_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The ID of the room, plus the alternate name for the user. The format is room@tc-server/nick.
DIRECTION	VARCHAR (1)	VARCHAR2 (1)	Yes	Indicates whether the user entered (E) or left (L) the room. Constrained to the values "E" and "L".
STAMP	TIMESTAMP	TIMESTAMP	Yes	The date and time at which the user entered or left the room.

TC_MSGARCHIVE Table

The TC_MSGARCHIVE table stores messages and associated information for group chat rooms.



Note

This table archives all messages if you turn on group chat on IM and Presence Service. Choose the option Archive all room messages on the **Cisco Unified CM IM and Presence Administration** user interface. Choose **Messaging > Conferencing and Persistent Chat**. See *Configuration and Administration of IM and Presence Service on Cisco Unified Communications Manager* for information on the group chat feature.

Column Name	Postgres Datatype	Oracle Datatype	Not Null	Description
MSG_ID	BIGINT	NUMBER (19)	Yes	A unique identifier for the message.
TO_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The ID of the room that received the message.
FROM_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The ID of the user who sent the message.
NICK_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The ID of the room, plus the alternate name of the sender; for example: room@conference.exmpl.com/nick
SENT_DATE	TIMESTAMP	TIMESTAMP	Yes	The date on which the message was sent.
MSG_TYPE	VARCHAR (1)	VARCHAR2 (1)	Yes	The first character of the type attribute of the message. The possible values are “c” (chat), “n” (normal), “g” (groupchat), “h” (headline), and “e” (error).
BODY_LEN	INT	NUMBER (9)	Yes	The length in characters of the message body.
MESSAGE_LEN	INT	NUMBER (9)	Yes	The length in characters of the message, including the subject and body.
BODY_STRING	VARCHAR (4000)	VARCHAR2 (4000)	Yes	The message body.
MESSAGE_STRING	VARCHAR (4000)	VARCHAR2 (4000)	Yes	The entire raw packet.
BODY_TEXT	TEXT	CLOB	Yes	If the message body exceeds 4000 characters, it is stored in this field rather than the BODY_STRING field.

Column Name	Postgres Datatype	Oracle Datatype	Not Null	Description
MESSAGE_TEXT	TEXT	CLOB	Yes	If the entire raw packet exceeds 4000 characters, it is stored in this column rather than in the MESSAGE_STRING column.
SUBJECT	VARCHAR (255)	VARCHAR2 (255)	Yes	The current subject of the room.

JM Table

The JM table stores conversations and associated information for the message archiver component. The message archiver component provides the native compliance functionality on the IM and Presence Service.

Column Name	Postgres Datatype	Oracle Datatype	Not Null	Description
TO_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The Jabber ID (JID) of the user who is sending the message being archived.
FROM_JID	VARCHAR (3071)	VARCHAR2 (3071)	Yes	The JID of the user who is receiving the message being archived.
SENT_DATE	TIMESTAMP	TIMESTAMP	Yes	The date the message was sent.
SUBJECT	VARCHAR (128)	VARCHAR2 (128)	Yes	The subject line of the message that is being archived.
THREAD_ID	VARCHAR (128)	VARCHAR2 (128)	Yes	The thread ID of the message that is being archived.
MSG_TYPE	VARCHAR (1)	VARCHAR2 (1)	Yes	The first character of the message's type attribute. The possible values are: <ul style="list-style-type: none"> • “c” — chat • “n” — normal • “g” — groupchat • “h” — headline • “e” — error
DIRECTION	VARCHAR (1)	VARCHAR2 (1)	Yes	Indicates whether the message is “O” — outgoing or “I” — incoming. If the message is sent between users on the same server, it is logged twice: once as outgoing and once as incoming.
BODY_LEN	INT	NUMBER (9)	Yes	The number of characters in the message body.

Column Name	Postgres Datatype	Oracle Datatype	Not Null	Description
MESSAGE_LEN	INT	NUMBER (9)	Yes	The number of characters in the message, including the subject and the body.
BODY_STRING	VARCHAR (4000)	VARCHAR2 (4000)	Yes	The message body.
MESSAGE_STRING	VARCHAR (4000)	VARCHAR2 (4000)	Yes	The entire raw packet.
BODY_TEXT	TEXT	CLOB	Yes	If the message body exceeds 4000 characters, it is stored in this field rather than the BODY_STRING field.
MESSAGE_TEXT	TEXT	TEXT	Yes	If the entire raw packet exceeds 4000 characters, it is stored in this field rather than in the MESSAGE_STRING field.
HISTORY_FLAG	VARCHAR (1)	VARCHAR2 (1)	Yes	Used when room history messages are sent to new participants (upon entering an existing room). This allows you to distinguish between messages received while actively participating in a room and those received as part of a history push. The latter message type is flagged with HISTORY_FLAG="H" in the database. Otherwise, this column is set to "N."

Sample SQL Queries for the JM Table

This section contains some sample SQL queries that you can run on the JM table to extract specific information. The following queries select all columns from the table but you can be more selective about which information you want to include in your SQL queries.

All Instant Messages Sent by a Specific User

The following SQL query returns all instant messages sent by a specific user:

```
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%';
```

All Instant Messages Received by a Specific User

The following SQL query returns all instant messages received by a specific user:

```
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%';
```

All Instant Messages That Contain a Specific Word

The following SQL query returns all instant messages that contain a specific word:

```
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%');
```

All Instant Messages Conversations and Chat Rooms From a Specific Date

The following SQL query returns all instant messages, conversations and chat rooms from a specific date:

```
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%';
```

