This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Info Mart Physical Data Model for a PostgreSQL Database

Table CHAT_SESSION_FACT

4/14/2025

Table CHAT_SESSION_FACT

Description

Introduced: 8.5.011
Modified: 8.5.015.19 (PRODUCER_BATCH_ID added); 8.5.014.26 (PARKING_QUEUE_COUNT and PARKING_QUEUE_DURATION columns added); 8.5.014.09 (THREAD_ID column added); 8.5.011.14 (8 new columns added specific to eServices asynchronous chat, as identified in the column descriptions)

In partitioned databases, this table is partitioned.

In on-premises deployments with Genesys Chat managed by Chat Server and in cloud deployments with Genesys Chat or Advanced Chat, each row in this table describes a chat session. A chat session is a single chat interaction from the point of view of the server that manages chat activity, and a single conversation from the point of view of the customer. Multiple agents can participate in a single chat interaction (session).

Each fact is based on user data sent in an Interaction Server reporting event when the chat session ends. Genesys Info Mart extracts the KVP data from the G_USERDATA_HISTORY table in IDB, and the transformation job combines the statistics in each event into a single CHAT_SESSION_FACT record. Rows are inserted on receipt of the reporting event and are not updated. The chat statistics reported in each record are summarized by session and are not connected to specific agents or, in deployments that include Bot Gateway Server (BGS), bots.

The MEDIA_SERVER_IXN_GUID links the CHAT_SESSION_FACT record with the related INTERACTION_FACT (IF). In deployments that include BGS, the MEDIA_SERVER_IXN_GUID also links the CHAT_SESSION_FACT record with the related BGS_SESSION_FACT records. In this way, Genesys Info Mart enables you to generate reports that provide details about Genesys Chat or Advanced Chat activity at the interaction level, session level, and chat bot level:

- The MEDIA_SERVER_IXN_GUID links the CHAT_SESSION_FACT record with the related INTERACTION_FACT (IF).
- In deployments that include BGS, the MEDIA_SERVER_IXN_GUID also links the CHAT_SESSION_FACT record with the related BGS_SESSION_FACT records.
- In Genesys Engage cloud deployments with Advanced Chat, the THREAD_ID links the CHAT_SESSION_FACT record with the related CHAT_THREAD_FACT. To get interaction details associated with a thread, you can join CHAT_SESSION_FACT and IF (via the MEDIA_SERVER_IXN_GUID) and then filter the results by CHAT_SESSION_FACT.THREAD_ID.

Terminology note

The meanings of terms such as *interaction, session, thread,* and *conversation* have evolved with Genesys chat implementations, and these terms might have different technical meanings in different contexts, depending on the type and version of chat implementation in your deployment.

- For the CHAT_SESSION_FACT table, the reporting entity is a set of chat messages with a particular customer on a single topic. The messages occur in close time proximity to each other. From the point of view of the server managing the chat activity, the messages occur within a single interaction.

  In the Genesys Info Mart documentation, the reporting entity that is the subject of CHAT_SESSION_FACT records is always referred to as a *session.* In certain chat implementations in cloud deployments and, therefore, in documentation describing those deployments, such a set of messages could be referred to as an *interaction,* and the term *session* could have a different meaning (see next bullet).

- For the CHAT_THREAD_FACT table, the reporting entity is a thread of multiple chat interactions with a particular customer over time.

  In the Genesys Info Mart documentation, the reporting entity that is the subject of CHAT_THREAD_FACT records is always referred to as a *thread.* In certain chat implementations in cloud deployments and, therefore, in documentation describing those deployments, these linked interactions, or threads, are referred to as *sessions* or *conversations.* As noted in the previous bullet, in the Genesys Info Mart documentation the term *session* always refers to the individual interactions in a thread.

- The term *asynchronous chat* or *asynchronous interactions* refers to chat interactions when both parties are not present in the chat at the same time. The types of supported asynchronous chat scenarios depend on the type and version of chat implementation in your deployment and determine what columns are populated in the CHAT_SESSION_FACT table.

Tip

To assist you in preparing supplementary documentation, click the following link to download a comma-separated text file containing information such as the data types and descriptions for all columns in this table: Download a CSV file.

Hint: For easiest viewing, open the downloaded CSV file in Excel and adjust settings for column widths, text wrapping, and so on as desired. Depending on your browser and other system settings, you might need to save the file to your desktop first.

Column List

Legend

Column	Data Type	P	M	F	DV
MEDIA_SERVER_IXN_GUID	varchar(64)	X	X		
ADDED_TS	integer		X		
START_DATE_TIME_KEY	integer	X	X	X	

Column	Data Type	P	M	F	DV
END_DATE_TIME_KEY	integer		X	X	
TENANT_KEY	integer		X	X	-2
SESSION_DURATION	integer		X		0
MSG_FROM_AGENTS_COUNT	integer		X		0
MSG_FROM_AGENTS_SIZE	integer		X		0
MSG_FROM_CUSTOMERS_COUNT	integer		X		0
MSG_FROM_CUSTOMERS_SIZE	integer		X		0
AGENT_REPLY_COUNT	integer		X		0
AGENT_REPLY_MAX_DURATION	integer		X		0
AGENT_REPLY_DURATION	integer		X		0
AGENT_WAIT_COUNT	integer		X		0
AGENT_WAIT_MAX_DURATION	integer		X		0
AGENT_WAIT_DURATION	integer		X		0
CUSTOMER_REPLY_COUNT	integer		X		0
CUSTOMER_REPLY_MAX_DURATION	integer		X		0
CUSTOMER_REPLY_DURATION	integer		X		0
CUSTOMER_WAIT_COUNT	integer		X		0
CUSTOMER_WAIT_MAX_DURATION	integer		X		0
CUSTOMER_WAIT_DURATION	integer		X		0
UNTIL_FIRST_AGENT_DURATION	integer		X		0
UNTIL_FIRST_REPLY_DURATION	integer		X		0
AGENTS_COUNT	integer		X		0
MSG_FROM_BOTS_COUNT	integer		X		0
MSG_FROM_BOTS_SIZE	integer		X		0
UNTIL_FIRST_BOT_DURATION	integer		X		0
BOTS_COUNT	integer		X		0
ASYNC_DORMANT_COUNT	integer				
ASYNC_DORMANT_DURATION	integer				
ASYNC_IDLE_COUNT	integer				
ASYNC_IDLE_DURATION	integer				
ACTIVE_IDLE_COUNT	integer				
ACTIVE_IDLE_DURATION	integer				
HANDLE_COUNT	integer				
HANDLE_DURATION	integer				
THREAD_ID	varchar(64)				
CHAT_SESSION_DIM_KEY	integer		X	X	-2
MEDIA_TYPE_KEY	integer		X	X	-2

Column	Data Type	P	M	F	DV
PARKING_QUEUE_COUNT	integer				
PARKING_QUEUE_DURATION	integer				
CREATE_AUDIT_KEY	numeric(19)		X	X	
UPDATE_AUDIT_KEY	numeric(19)			X	
PRODUCER_BATCH_ID	numeric(19)				

MEDIA_SERVER_IXN_GUID

The interaction GUID, as reported by Interaction Server. This value is the ID of the chat session. This GUID might not be unique. The value allows you to associate interaction details with the chat session details by using the following references:

```
INTERACTION_FACT.MEDIA_SERVER_IXN_GUID =
CHAT_SESSION_FACT.MEDIA_SERVER_IXN_GUID

AND INTERACTION_FACT.START_DATE_TIME_KEY =
CHAT_SESSION_FACT.START_DATE_TIME_KEY
```

In combination with START_DATE_TIME_KEY, MEDIA_SERVER_IXN_GUID forms the value of the composite primary key for this table in nonpartitioned as well as partitioned databases.

Note that in practice the size limit of column data is 50 characters, which corresponds to the data type size of the MEDIA_SERVER_IXN_GUID in the INTERACTION_FACT table.

ADDED_TS

The UTC-equivalent value of the date and time at which the event with chat data is received.

START_DATE_TIME_KEY

Based on KVP: ChatServerSessionStartedAt

Identifies the start of a 15-minute interval in which the chat session began. Use this value as a key to join the fact tables to any configured DATE_TIME dimension, in order to group the facts that are related to the same interval and/or convert the timestamp from the KVP to an appropriate time zone. In combination with MEDIA_SERVER_IXN_GUID, START_DATE_TIME_KEY forms the value of the composite primary key for this table in nonpartitioned as well as partitioned databases.

END_DATE_TIME_KEY

Based on KVP: ChatServerSessionClosedAt

Identifies the start of a 15-minute interval in which the chat session ended. Use this value as a key to join the fact tables to any configured DATE_TIME dimension, in order to group the facts that are related to the same interval and/or convert the timestamp from the KVP to an appropriate time zone.

TENANT_KEY

Based on KVP: csg_TenantId

The surrogate key that is used to join the TENANT dimension to the fact tables.

SESSION_DURATION

Based on KVP: csg_SessionTotalTime

The duration, in seconds, of the Chat Server session. Note that async chat sessions could last for a few days.

MSG_FROM_AGENTS_COUNT

Based on KVP: csg_MessagesFromAgentsCount

The total number of all messages visible to the customer that were sent by all agents involved in the chat. A chat session might involve several agents (for example, in the case of a conference or transfer).

MSG_FROM_AGENTS_SIZE

Based on KVP: csg_MessagesFromAgentsSize

The total size of all messages visible to the customer that were sent by all agents involved in the chat. The size is expressed as number of characters, including spaces.

MSG_FROM_CUSTOMERS_COUNT

Based on KVP: csg_MessagesFromCustomersCount

The total number of messages sent by the customer.

MSG_FROM_CUSTOMERS_SIZE

Based on KVP: csg_MessagesFromCustomersSize

The total size of the messages sent by the customer. The size is expressed as number of characters, including spaces.

AGENT_REPLY_COUNT

Based on KVP: cse_AgentReplyTotalCount

The total number of agent replies to the customer.

AGENT_REPLY_MAX_DURATION

Based on KVP: cse_AgentReplyMaxTime

The maximum amount of time, in seconds, the agent(s) spent replying to the customer (in other words, the maximum amount of time that elapsed between the customer's response and the time the first agent actually sent a reply). If the customer's response was a set of messages, the reply interval is calculated from the time the first message in the set was received.

Note: For asynchronous (async) chat interactions, if a chat session was in a dormant state while a customer message was received, the time until the agent(s) rejoined the session is excluded.

AGENT_REPLY_DURATION

Based on KVP: cse_AgentReplyTotalTime

The total amount of time, in seconds, the agent(s) spent replying to the customer.

Note: For async chat interactions, if a chat session was in a dormant state while a customer message was received, the time until the agent(s) rejoined the session is excluded.

AGENT_WAIT_COUNT

Based on KVP: cse_AgentWaitTotalCount

The number of times the agent(s) waited for a reply from the customer.

AGENT_WAIT_MAX_DURATION

Based on KVP: cse_AgentWaitMaxTime

The maximum amount of time, in seconds, the agent(s) spent waiting for a reply from the customer (in other words, the maximum amount of time that elapsed between the last response from any agent and the customer's reply). If the agent was waiting for a reply to a set of messages, the wait interval is calculated from the time the last message in the set was sent.

Note: For async chat interactions, cumulative dormant time until a customer's reply is received is excluded.

AGENT_WAIT_DURATION

Based on KVP: cse_AgentWaitTotalTime

The total amount of time, in seconds, the agent(s) spent waiting for a reply from the customer. If there were multiple agents on the chat, a time interval is counted only once.

Note: For async chat interactions, cumulative dormant time until a customer's reply is received is excluded.

CUSTOMER_REPLY_COUNT

Based on KVP: cse_CustomerReplyTotalCount

The number of times the customer replied to the agent(s).

CUSTOMER_REPLY_MAX_DURATION

Based on KVP: cse_CustomerReplyMaxTime

The maximum amount of time, in seconds, the customer spent replying to the agent(s). If the customer was replying to a set of messages, the reply interval is calculated from the time the first message in the set was received.

CUSTOMER_REPLY_DURATION

Based on KVP: cse_CustomerReplyTotalTime

The total amount of time, in seconds, the customer spent replying to the agent(s).

CUSTOMER_WAIT_COUNT

Based on KVP: cse_CustomerWaitTotalCount

The number of times the customer waited for a reply from an agent.

CUSTOMER_WAIT_MAX_DURATION

Based on KVP: cse_CustomerWaitMaxTime

The maximum amount of time, in seconds, the customer spent waiting for a reply from an agent. If the customer was waiting for a reply to a set of messages, the wait interval is calculated from the time the last message in the set was sent.

CUSTOMER_WAIT_DURATION

Based on KVP: cse_CustomerWaitTotalTime

The total amount of time, in seconds, the customer spent waiting for a reply from an agent.

UNTIL_FIRST_AGENT_DURATION

Based on KVP: csg_SessionUntilFirstAgentTime

The amount of time, in seconds, the customer waited until the first agent visible to the customer joined the session. An agent is not visible to the customer until the interaction has been successfully routed to and accepted by the agent.

The meaning of a value of `0` (zero) depends on the value of AGENTS_COUNT:

- If AGENTS_COUNT = 0, no agent ever joined the session.
- If AGENTS_COUNT > 0, an agent joined very quickly or existed on the session from the start.

UNTIL_FIRST_REPLY_DURATION

Based on KVP: csg_SessionUntilFirstReplyTime

The amount of time since the start of the session, in seconds, until the first agent submits into the chat session the first greeting/message that is visible to the customer.

AGENTS_COUNT

Based on KVP: csg_PartiesAsAgentCount

The number of unique parties that participated in the chat session as agents.

MSG_FROM_BOTS_COUNT

Based on KVP: csg_MessagesFromBotsCount

The total number of messages visible to the customer that were sent by all bots that participated in the chat session.

MSG_FROM_BOTS_SIZE

Based on KVP: csg_MessagesFromBotsSize

The total size of all messages visible to the customer that were sent by all bots that participated in the chat session. The size is expressed as number of characters, including spaces.

UNTIL_FIRST_BOT_DURATION

Based on KVP: csg_SessionUntilFirstBotTime

The amount of time, in seconds, the customer waited until the first bot visible to the customer joined the session.

BOTS_COUNT

Based on KVP: csg_PartiesAsBotCount

The number of unique parties that participated in the chat session as bots.

ASYNC_DORMANT_COUNT

Introduced: Release 8.5.011.14
Based on KVP: cse_AsyncDormantTotalCount

The total number of times that the async chat session was put in a dormant state (no agent was connected to the async chat session with the customer).

ASYNC_DORMANT_DURATION

Introduced: Release 8.5.011.14
Based on KVP: cse_AsyncDormantTotalTime

The total amount of time, in seconds, that the async chat session spent in a dormant state (no agent was connected to the async chat session with the customer). Routing time is excluded from this value.

ASYNC_IDLE_COUNT

Introduced: Release 8.5.011.14
Based on KVP: cse_AsyncIdleTotalCount

The total number of times when an inactivity period exceeded a configured threshold while no agent was connected to the async chat session (that is, while the chat session was in a dormant state).

ASYNC_IDLE_DURATION

Introduced: Release 8.5.011.14
Based on KVP: cse_AsyncIdleTotalTime

The total time of inactivity, in seconds, in the async chat session while no agent was connected (that is, while the chat session was in a dormant state).

ACTIVE_IDLE_COUNT

Introduced: Release 8.5.011.14
Based on KVP: cse_ActiveIdleTotalCount

The total number of times when an inactivity period exceeded a configured threshold while at least one agent was connected to the async chat session (that is, while the chat session was technically in an active state).

ACTIVE_IDLE_DURATION

Introduced: Release 8.5.011.14
Based on KVP: cse_ActiveIdleTotalTime

The total time of inactivity, in seconds, in the async chat session while at least one agent was connected (that is, while the chat session was technically in an active state).

HANDLE_COUNT

Introduced: Release 8.5.011.14
Based on KVP: cse_SessionHandleTotalCount

The total number of times a session was in an active state, with at least one agent connected to the chat session.

HANDLE_DURATION

Introduced: Release 8.5.011.14
Based on KVP: cse_SessionHandleTotalTime

The total time (in seconds) that at least one agent was connected to a chat session.

THREAD_ID

Introduced: Release 8.5.014.09
Based on KVP: thread_Id

Identifier of the thread that the chat session is part of. This field is populated in cloud deployments with Advanced Chat. The value allows you to associate interaction, or chat session, details with chat thread details by using the following reference:

```
CHAT_SESSION_FACT.THREAD_ID=CHAT_THREAD_FACT.THREAD_ID
```

CHAT_SESSION_DIM_KEY

Based on KVP: csg_SessionEndedByand csg_SessionEndedReasonand csg_LanguageNameand csg_MediaOriginand csg_ChatAsyncMode

The surrogate key that is used to join the CHAT_SESSION_DIM dimension to the fact table, to identify typical characteristics of the chat session.

MEDIA_TYPE_KEY

Based on KVP: csg_MediaType

The surrogate key that is used to join the MEDIA_TYPE dimension to the fact tables. The MEDIA_TYPE_KEY references the MEDIA_TYPE dimension record where the value of the KVP matches MEDIA_TYPE.MEDIA_NAME_CODE.

PARKING_QUEUE_COUNT

Introduced: Release 8.5.014.26
Based on KVP: cse_ParkingQueueCount

The total number of times that the interaction was placed in a parking queue. This column is populated only in deployments with Advanced Chat.

PARKING_QUEUE_DURATION

Introduced: Release 8.5.014.26
Based on KVP: cse_ParkingQueueCount

The total amount of time that the interaction spent in the parking queue. This column is populated only in deployments with Advanced Chat.

CREATE_AUDIT_KEY

The surrogate key that is used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data creation. This value can be useful for aggregation, enterprise application integration (EAI), and ETL tools—that is, applications that need to identify newly added data.

UPDATE_AUDIT_KEY

The surrogate key that is used to join to the CTL_AUDIT_LOG control table. The key specifies the lineage for data update. This value can be useful for aggregation, enterprise application integration

(EAI), and ETL tools—that is, applications that need to identify recently modified data.

PRODUCER_BATCH_ID

Introduced: Release 8.5.015.19
Reserved for internal use.

Index List

CODE	U	C	Description
I_CHAT_SESSION_FACT_SDT			Improves access time, based on the Start Date Time key.

Index I_CHAT_SESSION_FACT_SDT

Field	Sort	Comment
START_DATE_TIME_KEY	Ascending	

Subject Areas

No subject area information available.