



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 Microsoft SQL Server Database

Table `CHAT_SESSION_FACT`

# 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_ID     | varchar(64) | X | X |   |    |
| ADDED_TS            | int         |   | X |   |    |
| START_DATE_TIME_KEY | int         | X | X | X |    |

Table CHAT\_SESSION\_FACT

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

Table CHAT\_SESSION\_FACT

| Column                 | Data Type   | P | M | F | DV |
|------------------------|-------------|---|---|---|----|
| PARKING_QUEUE_COUNT    | numeric(19) |   |   |   |    |
| PARKING_QUEUE_DURATION | numeric(19) |   |   |   |    |
| 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.