WatchMQTT GRE – Database Schema (Current)

Last updated: 2025-08-31

This document describes the **current** schema populated by the GRE agent. It focuses on tables, columns, datatypes, constraints, relationships, indexes, and typical access patterns used by the dashboard/API.

Overview

Primary data sources - `$SYS/broker/#` → numeric metrics → **broker_metrics** - `$SYS/broker/log/#` → textual events → **events** → drives **sessions** & **subscriptions**; ensures **clients**

Core tables - `broker_metrics` - `events` - `clients` - `sessions` - `subscriptions`

Conceptual model

```
$SYS numeric ⟶ broker_metrics
$SYS logs    ⟶ events ⟶ sessions (open/close)
                        ↳⟶ subscriptions (active/inactive)
                    ↳⟶ clients (ensure/update)
```

Table: broker_metrics

Records numeric telemetry emitted by Mosquitto's `$SYS/broker/*` topics.

Column	Type	Nullable	Default	Notes
id	BIGSERIAL	NO		Primary key
broker	TEXT	NO		Logical broker name (e.g., `local`)
metric	TEXT	NO		`$SYS` path suffix (e.g., `clients/total`, `messages/sent/1min`)
value	DOUBLE PRECISION	NO		Parsed numeric value
ts	TIMESTAMP	NO		Ingest timestamp (from agent)

Indexes - `broker_metrics_ts_idx (ts)` - `broker_metrics_metric_ts_idx (metric, ts)`

Cardinality & notes - High write-rate (every ~10s per metric). Suitable for time-series charts.

Table: events

Immutable audit of broker log lines normalized into actions.

Column	Type	Nullable	Default	Notes
id	BIGSERIAL	NO		Primary key
ts	TIMESTAMP	NO		Parsed from log line (seconds epoch → timestamp)
action	TEXT	NO		One of: `connected`, `disconnected`, `subscribe`, `unsubscribe`, `checkpoint`, `conn_info`, `unknown`
client	TEXT	YES		Client identifier from broker logs (if present)
topic	TEXT	YES		Topic affected (subs/unsubs)
qos	INTEGER	YES		QoS for subs/unsubs (0 if missing)
username	TEXT	YES		Auth username if present in `connected` line
raw	TEXT	YES		Original log payload (for traceability)

Indexes - `events_ts_idx (ts)` - `events_action_ts_idx (action, ts)`

Usage - Recent activity feeds, churn trends, diagnostics.

Table: clients

Unique client identities observed by the agent.

Column	Type	Nullable	Default	Notes
id	BIGSERIAL	NO		Primary key
client	TEXT	NO		Broker client id (unique)
username	TEXT	YES		Last observed username
first_seen	TIMESTAMP	YES		First time seen by agent
last_seen	TIMESTAMP	YES		Updated on activity (e.g., connect)

Constraints & Indexes - Unique index on `(client)` → `clients_client_unique`.

Usage - Foreign reference from `sessions.client_id`; quick lookups for client metadata.

Table: sessions

Represents connected windows for each client; opened on `connected`, closed on `disconnected`.

Column	Type	Nullable	Default	Notes
id	BIGSERIAL	NO		Primary key
client_id	BIGINT	YES		FK to `clients.id` (nullable to tolerate out-of-order events)
client	TEXT	YES		Redundant client id for convenience/joins
username	TEXT	YES		Username recorded on connect
start_ts	TIMESTAMP	YES		When the session was opened
end_ts	TIMESTAMP	YES		When closed; `NULL` means **open**

Relationships - `subscriptions.session_id` (nullable) may refer to `sessions.id` (latest open session for that client at the time of sub/unsub).

Indexes - `sessions_client_ts_idx (client, start_ts)`

Usage - "Connected now" (rows with `end_ts IS NULL`), session timelines, reliability metrics (durations).

Table: subscriptions

Tracks per-client topic subscriptions; upserted by `(client, topic)`.

Column	Type	Nullable	Default	Notes
id	BIGSERIAL	NO		Primary key
session_id	BIGINT	YES		Nullable FK to `sessions.id` (best-effort link to the session active at change time)
client	TEXT	NO		Client id
topic	TEXT	NO		Subscribed topic
qos	INTEGER	NO	0	QoS level
active	BOOLEAN	NO	TRUE	TRUE on subscribe; FALSE on unsubscribe
created_at	TIMESTAMP	YES		First time seen
updated_at	TIMESTAMP	YES		Last change timestamp

Constraints & Indexes - Unique index `subscriptions_client_topic_idx (client, topic)` → supports idempotent upsert. - Optional FK: `subscriptions.session_id` → `sessions.id` (ON DELETE SET NULL) if declared.

Usage - “Active subs per topic”, client topic footprint, churn.

Relationships Summary

- `sessions.client_id` → `clients.id` (nullable to tolerate ordering)
- `subscriptions.session_id` → `sessions.id` (nullable)
- `events` has no formal FKs (by design: raw log history), but fields align (client/topic).

ER sketch

```
clients (1) ──< sessions (many)
   │              └─< subscriptions (many, via client; session link optional)
   └──────── events (loose association by client/topic)

broker_metrics (independent time series)
```

Typical Access Patterns (Read)

- **Now:**
- Current connected clients → `sessions` where `end_ts IS NULL`
- Live subscriptions → `subscriptions` where `active = TRUE`
- Latest metrics snapshot → latest row per metric in `broker_metrics`

- **Time series & Trends:**

- Messages per minute over range → `broker_metrics` filtered by metric + time window
- Connect vs. disconnect counts → `events` grouped by `action` and time bucket

- **Auditing & Diagnostics:**

- Recent activity feed → `events` ordered by `ts DESC`
- Checkpoints / connection-info → `events` filtered by action

Data Quality & Semantics

- **Ordering:** Broker logs may arrive in quick succession; parser derives timestamps from log prefixes. Session linkage is robust to minor ordering differences.
- **Idempotency:** `subscriptions` upsert on `(client, topic)` ensures the latest state is reflected without duplicates.
- **Granularity:** `broker_metrics` retains raw 10-second points; consider downsampling for long-range charts.

Operational Notes (Current)

- Write volume: moderate (`events`, `sessions`, `subscriptions`), high (`broker_metrics`).
- Growth: unbounded unless retention is applied (recommended next step).
- Indexes: present for common filters; monitor `btree` sizes and query plans as data grows.

Future Extensions (Not Yet Implemented)

- `pub_stats`, `sub_stats`, `session_stats` (derived aggregates)
- Additional event classes (ACL changes, retained messages, LWT triggers) as needed
- Partitioning `broker_metrics` by time for easier retention

End of document.