**2. Tables Definition**

**1. calls**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='calls’;

* id (integer) - Primary Key, auto\_increment
* queue\_id (integer) - Foreign Key referencing queues(id)
* service\_id (integer) - Foreign Key referencing services(id)
* counter\_id (integer) - Foreign Key referencing counters(id)
* user\_id (integer) - Foreign Key referencing users(id)
* token\_letter (varchar (191)) - Token letter associated with the call
* token\_number (integer) - Token number associated with the call
* called\_date (date) - Date when the call was made
* started\_at (datetime) - Timestamp when the call started, defaults to current timestamp
* ended\_at (datetime) - Timestamp when the call ended, defaults to NULL
* waiting\_time (time) - Time spent waiting, defaults to NULL
* served\_time (time) - Time spent being served, defaults to NULL
* turn\_around\_time (time) - Total turnaround time, defaults to NULL
* created\_at (timestamp) - Timestamp when the call was created, defaults to NULL
* updated\_at (timestamp) - Timestamp when the call was last updated, defaults to NULL
* call\_status\_id (integer) - Foreign Key referencing call\_statuses(id), defaults to NULL

**2. call\_statuses**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='call\_statuses’;

* id (integer) - Primary Key, auto\_increment
* name (varchar(191)) - Name of the call status
* created\_at (timestamp) - Timestamp when the call status was created, defaults to NULL
* updated\_at (timestamp) - Timestamp when the call status was last updated, defaults to NULL

**3. Counters**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='counters’;

* id (integer) - Primary Key, auto\_increment
* name (varchar(191)) - Name of the counter
* status (integer) - Status of the counter
* created\_at (timestamp) - Timestamp when the counter was created, defaults to NULL
* updated\_at (timestamp) - Timestamp when the counter was last updated, defaults to NULL

**4.demo**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='demo’;

* ID (integer) - Primary Key, auto\_increment
* Name (varchar(20)) - Name of the demo
* Hint (text) - Hint text for the demo

**5.jobs**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='jobs’;

* id (integer) - Primary Key, auto\_increment
* queue (varchar(191)) - Queue name
* payload (longtext) - Payload of the job
* attempts (integer) - Number of attempts
* reserved\_at (integer) - Timestamp when the job was reserved, defaults to NULL
* available\_at (integer) - Timestamp when the job is available
* created\_at (integer) - Timestamp when the job was created

**6.Languages**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='languages';

* id (integer) - Primary Key, auto\_increment
* code (varchar(191)) - code for the language
* name (varchar(191)) - Name of the language
* display (varchar(191)) - Display name of the language
* image (varchar(191)) - Image associated with the language, defaults to NULL
* created\_at (timestamp) - Timestamp when the language was created, defaults to NULL
* updated\_at (timestamp) - Timestamp when the language was last updated, defaults to NULL
* token\_translation (varchar(191)) - Token translation text, defaults to NULL
* please\_proceed\_to\_translation (varchar(191)) - Translation for "please proceed to", defaults to NULL

**7. permisssions**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='permissions';

* id (integer) - Primary Key, auto\_increment
* name (varchar(191)) - Name of the permission
* guard\_name (varchar(191)) - Guard name associated with the permission
* created\_at (timestamp) - Timestamp when the permission was created, defaults to NULL
* updated\_at (timestamp) - Timestamp when the permission was last updated, defaults to NULL

**8. queues**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='queues';

* id (integer) - Primary Key, auto\_increment
* service\_id (integer) - Foreign Key referencing services(id)
* number (integer) - Queue number
* called (integer) - Indicator if the queue has been called
* reference\_no (varchar(191)) - Reference number for the queue
* letter (varchar(191)) - Letter associated with the queue
* name (varchar(191)) - Name of the person in the queue, defaults to NULL
* email (varchar(191)) - Email of the person in the queue, defaults to NULL
* phone (varchar(191)) - Phone number of the person in the queue, defaults to NULL
* position (integer) - Position in the queue, defaults to NULL
* created\_at (timestamp) - Timestamp when the queue was created, defaults to NULL
* updated\_at (timestamp) - Timestamp when the queue was last updated, defaults to NULL

**9. roles**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='roles';

* id (integer) - Primary Key, auto\_increment
* name (varchar(191)) - Name of the role
* guard\_name (varchar(191)) - Guard name associated with the role
* created\_at (timestamp) - Timestamp when the role was created, defaults to NULL
* updated\_at (timestamp) - Timestamp when the role was last updated, defaults to NULL

**10. role\_has\_permissions**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='role\_has\_permissions';

* permission\_id (integer) - Foreign Key referencing permissions(id)
* role\_id (integer) - Foreign Key referencing roles(id)

**11. services**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='services';

* id (integer) - Primary Key, auto\_increment
* name (varchar(191)) - Name of the service
* letter (varchar(191)) - Letter associated with the service
* start\_number (integer) - The starting number for the service
* status (integer) - Status of the service
* sms\_enabled (integer) - Indicator if SMS is enabled (0 or 1)
* optin\_message\_enabled (integer) - Indicator if opt-in message is enabled (0 or 1)
* call\_message\_enabled (integer) - Indicator if call message is enabled (0 or 1)
* noshow\_message\_enabled (integer) - Indicator if no-show message is enabled (0 or 1)
* completed\_message\_enabled (integer) - Indicator if completed message is enabled (0 or 1)
* status\_message\_enabled (integer) - Indicator if status message is enabled (0 or 1)
* optin\_message\_format (varchar(191)) - Format of the opt-in message
* call\_message\_format (varchar(191)) - Format of the call message
* noshow\_message\_format (varchar(191)) - Format of the no-show message
* completed\_message\_format (varchar(191)) - Format of the completed message
* status\_message\_format (varchar(191)) - Format of the status message
* status\_message\_positions (varchar(191)) - Positions for the status message
* ask\_name (integer) - Indicator if name is asked (0 or 1)
* name\_required (integer) - Indicator if name is required (0 or 1)
* ask\_email (integer) - Indicator if email is asked (0 or 1)
* email\_required (integer) - Indicator if email is required (0 or 1)
* ask\_phone (integer) - Indicator if phone is asked (0 or 1)
* phone\_required (integer) - Indicator if phone is required (0 or 1)
* created\_at (timestamp) - Timestamp when the service was created
* updated\_at (timestamp) - Timestamp when the service was last updated

**12. sqlite\_sequence**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='sqlite\_sequence';

* name (text) - Name of the table
* seq (integer) - Current sequence number for the table's auto-increment column

**13. users**

Query: SELECT sql FROM sqlite\_master WHERE type='table' AND name='users';

id (integer) - Primary Key, auto-increment

name (varchar(191)) - Name of the user

email (varchar(191)) - Email of the user, unique

email\_verified\_at (timestamp) - Timestamp when the email was verified

password (varchar(191)) - Password for the user

remember\_token (varchar(100)) - Token used for remembering the user

created\_at (timestamp) - Timestamp when the user was created

updated\_at (timestamp) - Timestamp when the user was last updated

image (varchar(191)) - Path or URL to the user's image

**2. Table Relationship**

Here are the relationships:

* calls & queues: Linked by queue\_id in the calls table referencing queues(id).
* calls & services: Linked by service\_id in the calls table referencing services(id).
* calls & counters: Linked by counter\_id in the calls table referencing counters(id).
* calls & users: Linked by user\_id in the calls table referencing users(id).
* calls & call\_statuses: Linked by call\_status\_id in the calls table referencing call\_statuses(id).
* queues & services: Linked by service\_id in the queues table referencing services(id).
* role\_has\_permissions & permissions: Linked by permission\_id in the role\_has\_permissions table referencing permissions(id).
* role\_has\_permissions & roles: Linked by role\_id in the role\_has\_permissions table referencing roles(id).