This document shows a simple ODBC access to sendQuick servers to send/receive SMS using a database record entry. SendQuick is using PostgreSQL database. Please ensure the relevant drivers are pre-installed in your system before accessing sendQuick. The ODBC driver is available from the Internet.

Login Access:
After configuring the ODBC driver in your database, you need to configure the permission and password in sendQuick before you can access the sendQuick database.

First, login to the sendQuick server administrator interface, and add the remote IP address (where you will be accessing sendQuick) into the DB host allowed text area under the "Security Setup". Then, make sure the password for smsapp DB (the DB that store all the tables below) has been changed in the "Change Password" (via the "Menu"). The default password for smsapp is "password". You can change it to suit your needs.

The login user name for remote access is "**smsapp**", and the database is "**spooldb**".

Remote access available to Outbox, Inbox, Queue and Unsent records. Below are the relation views created:

Outbox (Sent Messages) Name: public.outbox		
Field Description	***Field Name***	***Type***
Message ID	msgid	character varying(50)
Message Type	msg_type	character(1)
Date and Time message was sent	completed_dtm	timestamp with time zone
Modem IMEI used to send SMS	modem_imei	text
SMSC number of the SIM	smsc	text
Recipient mobile number	mobile_numb	character varying(20
Sender of the message	msg_from	character varying(500)
Username in Webapp application	webapp_user	character varying(20)
SMS message content	msg_content	character varying(1000)

Inbox (Received/Incoming Messages) Name: public.inbox		
Field Description	***Field Name***	***Type***
Message ID	msgid	character varying(50)
Date and Time message was received	completed_dtm	timestamp with time zone
Modem IMEI used to receive SMS	modem_imei	text
SMSC number of the SIM	smsc	text
Sender mobile number	mobile_numb	character varying(20
SMS message content received	msg_content	character varying(1000)

Unsent (Failed Messages) Name: public.unsent		
Field Description	***Field Name***	***Type***
Message ID	msgid	character varying(50)
Message Type	msg_type	character(1)
Date and Time message was unsent	completed_dtm	timestamp with time zone
Modem IMEI used to send SMS	modem_imei	text
SMSC number of the SIM	smsc	text
Recipient mobile number	mobile_numb	character varying(20
Sender of the message	msg_from	character varying(500)
Username in Webapp application	webapp_user	character varying(20)
SMS message content	msg_content	character varying(1000)

Queue (Messages Queuing in the Server) Name: public.queue		
Field Description	***Field Name***	***Type***
Message ID	msgid	character varying(50)
Message Type	msg_type	character(1)
Date and Time message was created	completed_dtm	timestamp with time zone
Recipient mobile number	mobile_numb	character varying(20
Sender of the message	msg_from	character varying(500)
Username in Webapp application	webapp_user	character varying(20)
SMS message content	msg_content	character varying(1000)

NOTE:

- msgid: the unique ID for each recorded message.
- Smsc: the SMSC number of the modem
- modem_imei: the modem used for sending or receiving the SMS
- msg_content: the content of the SMS message.
- msg_from: the sender information for outgoing messages.
- mobile_numb: the mobile number of sender in Inbox or recipient for Outbox
- msg_type: Please ignore this for now, we would be expanding this more as part of your additional requirements.
- created_dtm: The date and time the record was added into the system
- webapp_user: The user ID (of web interface) that sent the messages.

Send SMS via ODBC

You can also send SMS using ODBC via the same ***smsapp*** account. The database table for sending SMS is as below:

Send SMS (Send SMS via ODBC) Name: odbc_queue		
Field Description	***Field Name***	***Type***
Recipient mobile number	mobile_numb	character varying(20
Sender of the message	msg_from	text
SMS message content	msg_content	text

The system would take the message and send it accordingly. Once processed, the message would be deleted. You should use the "msg_from" and "mobile_numb" for referencing the message in the outbox/queue/unsent views.