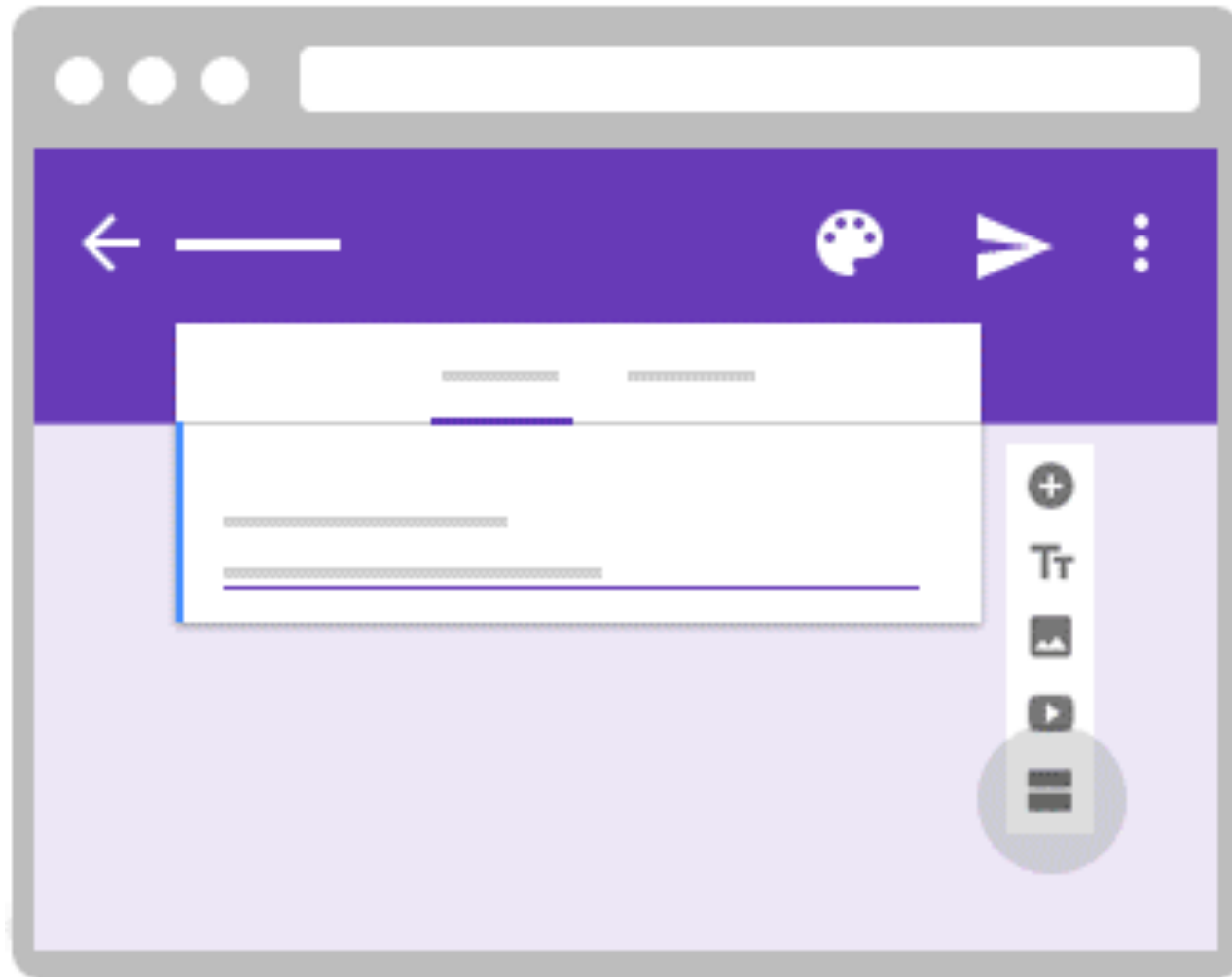


surveydown: An Open-Source, Markdown-Based Platform for Interactive and Reproducible Surveys

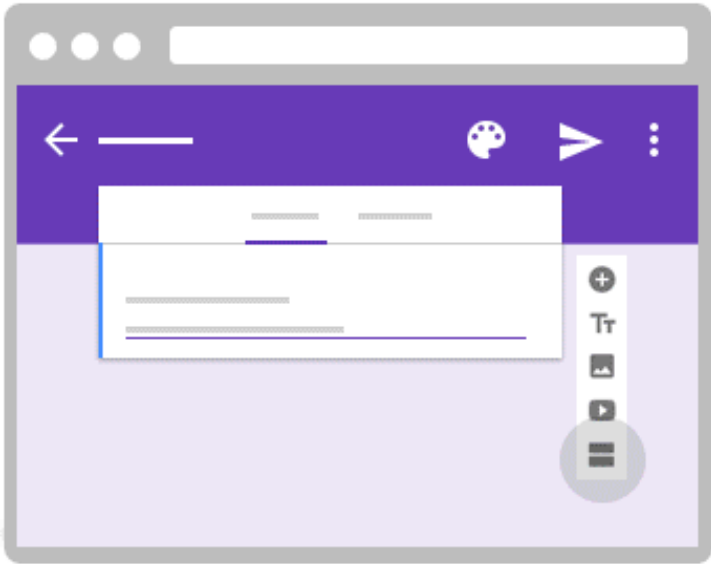
Pingfan Hu, Bogdan Bunea, John Paul Helveston



Typical experience making a survey



Typical experience making a survey



✗ Reproducible

✗ Data Control

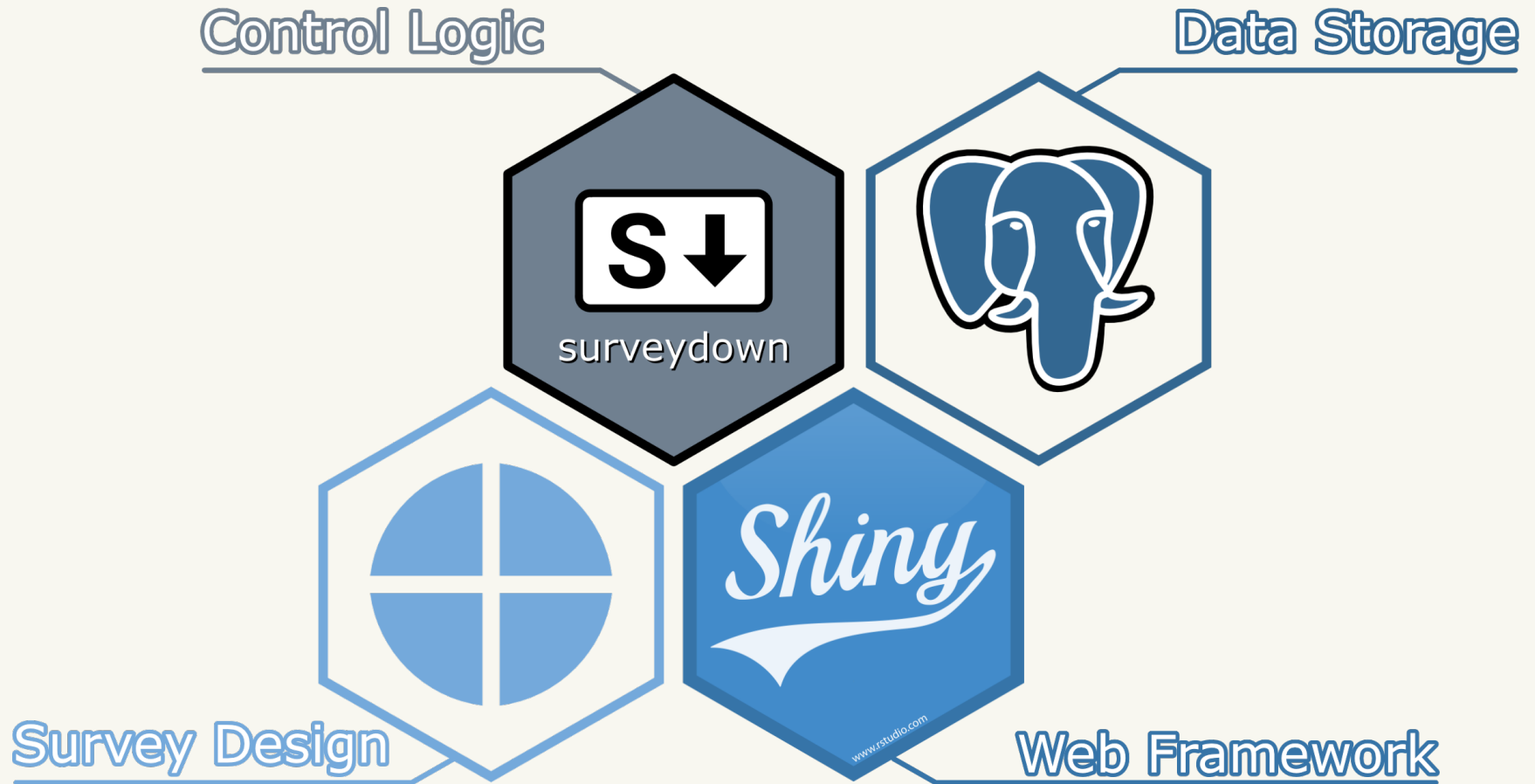
⚠ Free to Use

⚠ Open Source

⚠ Easy Collaboration

⚠ Feature Packed

Introducing surveydown



Components

`survey.qmd`

A **Quarto doc** defining the main survey content (pages, text, images, questions, etc).

`app.R`

An **R script** defining the survey Shiny app. It sets up the database and server, and launches the survey.

survey.qmd

```
1 ---
2 format: html
3 echo: false
4 warning: false
5 ---
6
7 ```{r}
8 library(surveydown)
9 ```
10
11 ::: {#welcome .sd-page}
12
13 # Welcome to our survey!
14
15 ```{r}
16 sd_question(
17   type = "mc",
18   id   = "penguins",
19   label = "What's your favorite penguin?",
20   option = c(
21     "Adélie" = "adelie",
22     "Chinstrap" = "chinstrap",
23     "Gentoo" = "gentoo"
24   )
25 )
26
27 sd_next()
28 ```
29
30 :::
31
32 ::: {#end .sd-page}
33
34 This is the last page of the survey.
35
36 ```{r}
37 sd_close()
38 ```
39
40 :::
```

survey.qmd

```
1 ---
2 format: html
3 echo: false
4 warning: false
5 ---
6
7 ```{r}
8 library(surveydown)
9 ```
10
11 ::: {#welcome .sd-page}
12
13 # Welcome to our survey!
14
15 ```{r}
16 sd_question(
17   type = "mc",
18   id   = "penguins",
19   label = "What's your favorite penguin?",
20   option = c(
21     "Adélie" = "adelie",
22     "Chinstrap" = "chinstrap",
23     "Gentoo" = "gentoo"
24   )
25 )
26
27 sd_next()
28 ```
29
30 :::
31
32 ::: {#end .sd-page}
33
34 This is the last page of the survey.
35
36 ```{r}
37 sd_close()
38 ```
39
40 :::
```

YAML Header

Load Package

survey.qmd

```
1 ---
2 format: html
3 echo: false
4 warning: false
5 ---
6
7 ```{r}
8 library(surveydown)
9 ```
10
11 ::: {#welcome .sd-page}
12
13 # Welcome to our survey!
14
15 ```{r}
16 sd_question(
17   type = "mc",
18   id    = "penguins",
19   label = "What's your favorite penguin?",
20   option = c(
21     "Adélie"      = "adelie",
22     "Chinstrap"   = "chinstrap",
23     "Gentoo"      = "gentoo"
24   )
25 )
26
27 sd_next()
28 ```
29
30 :::
31
32 ::: {#end .sd-page}
33
34 This is the last page of the survey.
35
36 ```{r}
37 sd_close()
38 ```
39
40 :::
```

Define pages using “fences”:

```
::: {#id .sd-page}
:::
```

Page 1 (Welcome)

Page 2 (End)

survey.qmd

```
1 ---
2 format: html
3 echo: false
4 warning: false
5 ---
6
7 ```{r}
8 library(surveydown)
9 ```
10
11 ::: {#welcome .sd-page}
12
13 # Welcome to our survey!
14
15 ```{r}
16 sd_question(
17   type = "mc",
18   id    = "penguins",
19   label = "What's your favorite penguin?",
20   option = c(
21     "Adélie"    = "adelie",
22     "Chinstrap" = "chinstrap",
23     "Gentoo"    = "gentoo"
24   )
25 )
26
27 sd_next()
28 ```
29
30 :::
31
32 ::: {#end .sd-page}
33
34 This is the last page of the survey.
35
36 ```{r}
37 sd_close()
38 ```
39
40 :::
```

Contents of Page 1

survey.qmd

```
1 ---
2 format: html
3 echo: false
4 warning: false
5 ---
6
7 ```{r}
8 library(surveydown)
9 ```
10
11 ::: {#welcome .sd-page}
12
13 # Welcome to our survey!
14
15 ```{r}
16 sd_question(
17   type = "mc",
18   id    = "penguins",
19   label = "What's your favorite penguin?",
20   option = c(
21     "Adélie"    = "adelie",
22     "Chinstrap" = "chinstrap",
23     "Gentoo"    = "gentoo"
24   )
25 )
26
27 sd_next()
28 ```
29
30 :::
31
32 ::: {#end .sd-page}
33
34 This is the last page of the survey.
35
36 ```{r}
37 sd_close()
38 ```
39
40 :::
```

Question Definition

app.R

```
1 # Load Package
2 library(surveydown)
3
4 # Database Credentials
5 sd_db_config()
6
7 # Connect to Database
8 db <- sd_db_connect()
9
10 # Server Setup
11 server <- function(input, output, session) {
12
13   # Skip logic
14   sd_skip_if(...)
15
16   # Conditional display
17   sd_show_if(...)
18
19   # Server settings
20   sd_server(
21     db = db,
22     ...
23   )
24 }
25
26 # Launch Survey
27 shiny::shinyApp(ui = sd_ui(), server = server)
```

app.R

```
1 # Load Package
2 library(surveydown)
3
4 # Database Credentials
5 sd_db_config()
6
7 # Connect to Database
8 db <- sd_db_connect()
9
10 # Server Setup
11 server <- function(input, output, session) {
12
13   # Skip logic
14   sd_skip_if(...)
15
16   # Conditional display
17   sd_show_if(...)
18
19   # Server settings
20   sd_server(
21     db = db,
22     ...
23   )
24 }
25
26 # Launch Survey
27 shiny::shinyApp(ui = sd_ui(), server = server)
```

Load Package

```
1 # Load Package
2 library(surveydown)
3
4 # Database Credentials
5 sd_db_config()
6
7 # Connect to Database
8 db <- sd_db_connect()
9
10 # Server Setup
11 server <- function(input, output, session) {
12
13   # Skip logic
14   sd_skip_if(...)
15
16   # Conditional display
17   sd_show_if(...)
18
19   # Server settings
20   sd_server(
21     db = db,
22     ...
23   )
24 }
25
26 # Launch Survey
27 shiny::shinyApp(ui = sd_ui(), server = server)
```

Database Setup

We recommend using **Supabase** or any **PostgreSQL** database for the database setup.

app.R

```
1 # Load Package
2 library(surveydown)
3
4 # Database Credentials
5 sd_db_config()
6
7 # Connect to Database
8 db <- sd_db_connect()
9
10 # Server Setup
11 server <- function(input, output, session) {
12
13   # Skip logic
14   sd_skip_if(...)
15
16   # Conditional display
17   sd_show_if(...)
18
19   # Server settings
20   sd_server(
21     db = db,
22     ...
23   )
24 }
25
26 # Launch Survey
27 shiny::shinyApp(ui = sd_ui(), server = server)
```

Server Setup

app.R

```
1 # Load Package
2 library(surveydown)
3
4 # Database Credentials
5 sd_db_config()
6
7 # Connect to Database
8 db <- sd_db_connect()
9
10 # Server Setup
11 server <- function(input, output, session) {
12
13   # Skip logic
14   sd_skip_if(...)
15
16   # Conditional display
17   sd_show_if(...)
18
19   # Server settings
20   sd_server(
21     db = db,
22     ...
23   )
24 }
25
26 # Launch Survey
27 shiny::shinyApp(ui = sd_ui(), server = server)
```

Launch Survey



Templates and demos are provided on
surveydown.org for an easy start.

Demo: Built-in Question Types

Demo - Question Types

Hi! This is a showcase of different question types and question formatting that surveydown supports.

Please make your choice below.

Where do you want to go to?

- ☐ Take me to the page on Question Types (the next page)
- ☐ Take me to the page on Question Formatting
- ☐ Skip to the end

Next

Made with [surveydown](#)

[Source Code](#)

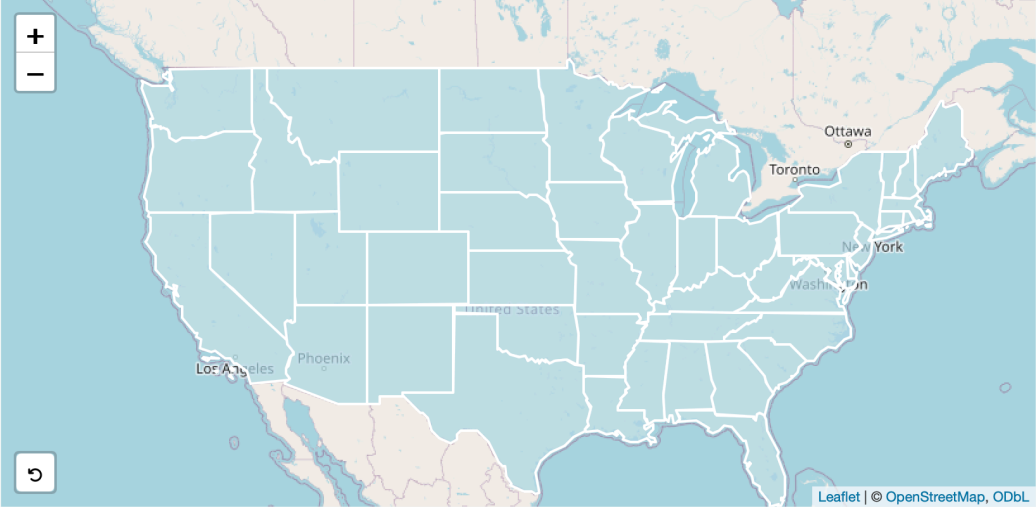
Demo: Custom Question (leaflet Map)

Demo - Custom Leaflet Map

Click on the state you live in:

+

-



↺

Leaflet | © OpenStreetMap, ODbL

You live in:

Next

Made with [surveydown](#)[Source Code](#)

Data Management in Supabase

Some platforms are IPv4-only:

Transaction pooler

Supervisor

Ideal for stateless applications like serverless functions where each interaction with Postgres is brief and isolated.

```
postgresql://postgres.hwumdvjnymbrgeemvrhq:[YOUR-PASSWORD]@
```

Does not support PREPARE statements

View parameters

host:

port: 6543

database: postgres

user:

pool_mode: transaction

For security reasons, your database password is never shown.

- Suitable for a large number of connected clients
- Pre-warmed connection pool to Postgres
- IPv4 compatible
Transaction pooler connections are IPv4 proxied for free.

Session pooler

Supervisor

Only recommended as an alternative to Direct Connection, when connecting via an IPv4 network.

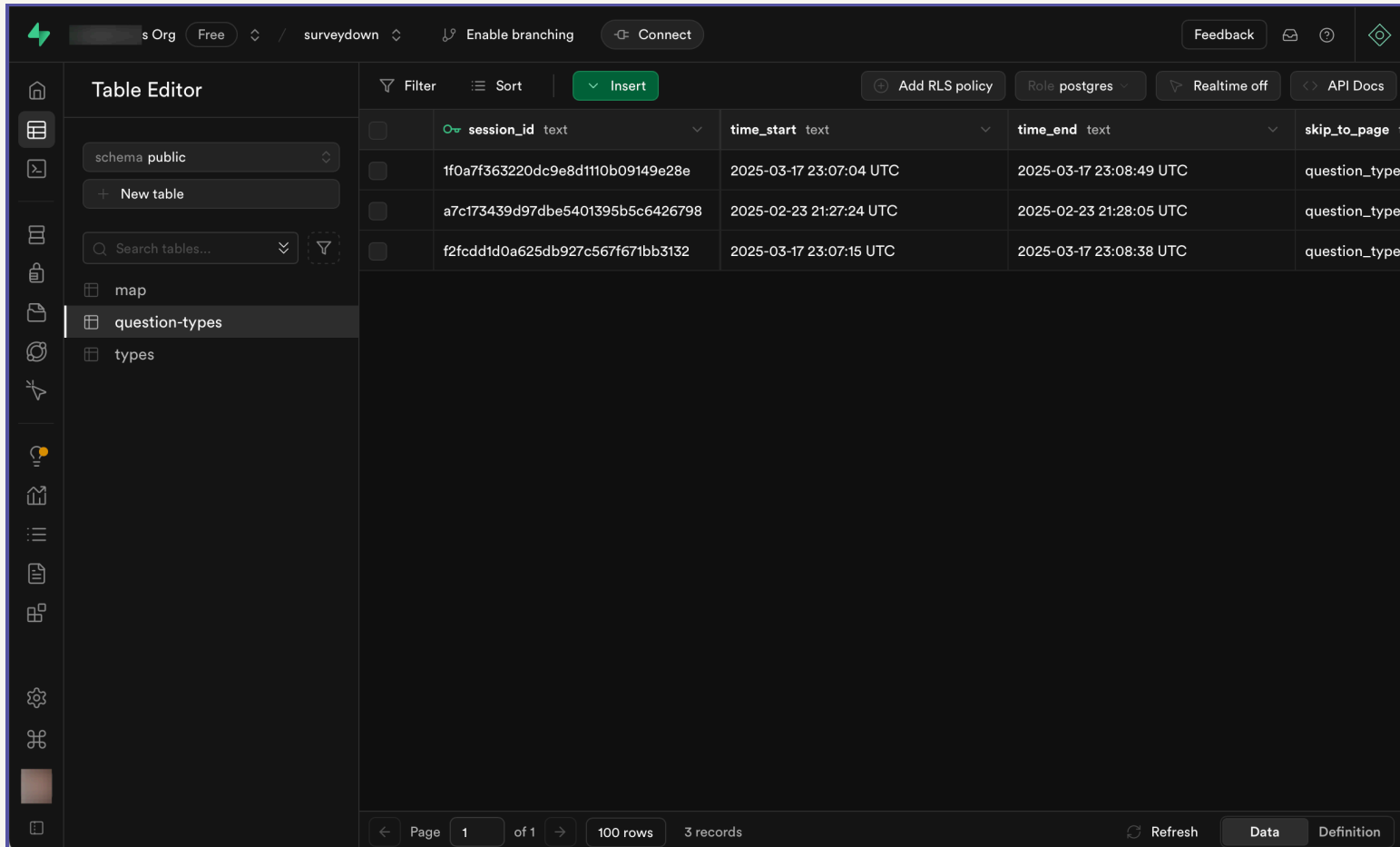
```
postgresql://postgres.hwumdvjnymbrgeemvrhq:[YOUR-PASSWORD]@
```

View parameters

- IPv4 compatible
Session pooler connections are IPv4 proxied for free
- Only use on a IPv4 network
Use Direct Connection if connecting via an IPv6 network

Credentials

Data Management in Supabase



The screenshot shows the Supabase Table Editor interface. On the left, a sidebar contains icons for database management and a list of tables: 'map', 'question-types', and 'types'. The 'question-types' table is selected. The main area displays a table with the following columns: 'session_id' (text), 'time_start' (text), 'time_end' (text), and 'skip_to_page' (text). The table contains three records. The top record has a session_id of '1f0a7f363220dc9e8d1110b09149e28e' and timestamps from '2025-03-17 23:07:04 UTC' to '2025-03-17 23:08:49 UTC'. The bottom record has a session_id of 'f2fcdd1d0a625db927c567f671bb3132' and timestamps from '2025-03-17 23:07:15 UTC' to '2025-03-17 23:08:38 UTC'. The interface includes a top navigation bar with 's Org', 'Free', 'surveydown', 'Enable branching', and 'Connect' buttons. A bottom status bar shows 'Page 1 of 1', '100 rows', '3 records', and 'Refresh' button.

session_id	time_start	time_end	skip_to_page
1f0a7f363220dc9e8d1110b09149e28e	2025-03-17 23:07:04 UTC	2025-03-17 23:08:49 UTC	question_type:
a7c173439d97dbe5401395b5c6426798	2025-02-23 21:27:24 UTC	2025-02-23 21:28:05 UTC	question_type:
f2fcdd1d0a625db927c567f671bb3132	2025-03-17 23:07:15 UTC	2025-03-17 23:08:38 UTC	question_type:

Survey Data

Data Management in surveydown Dashboard

Survey Dashboard

Dashboard

DB Config

Current Database Settings

Host:

Port:

6543

Database:

postgres

User:

GSS Mode:

disable

Default Table:

question-types

Update Database Connection

Host:

aws-0-us-east-1.pooler.supabase.com

Port:

6543

Database Name:

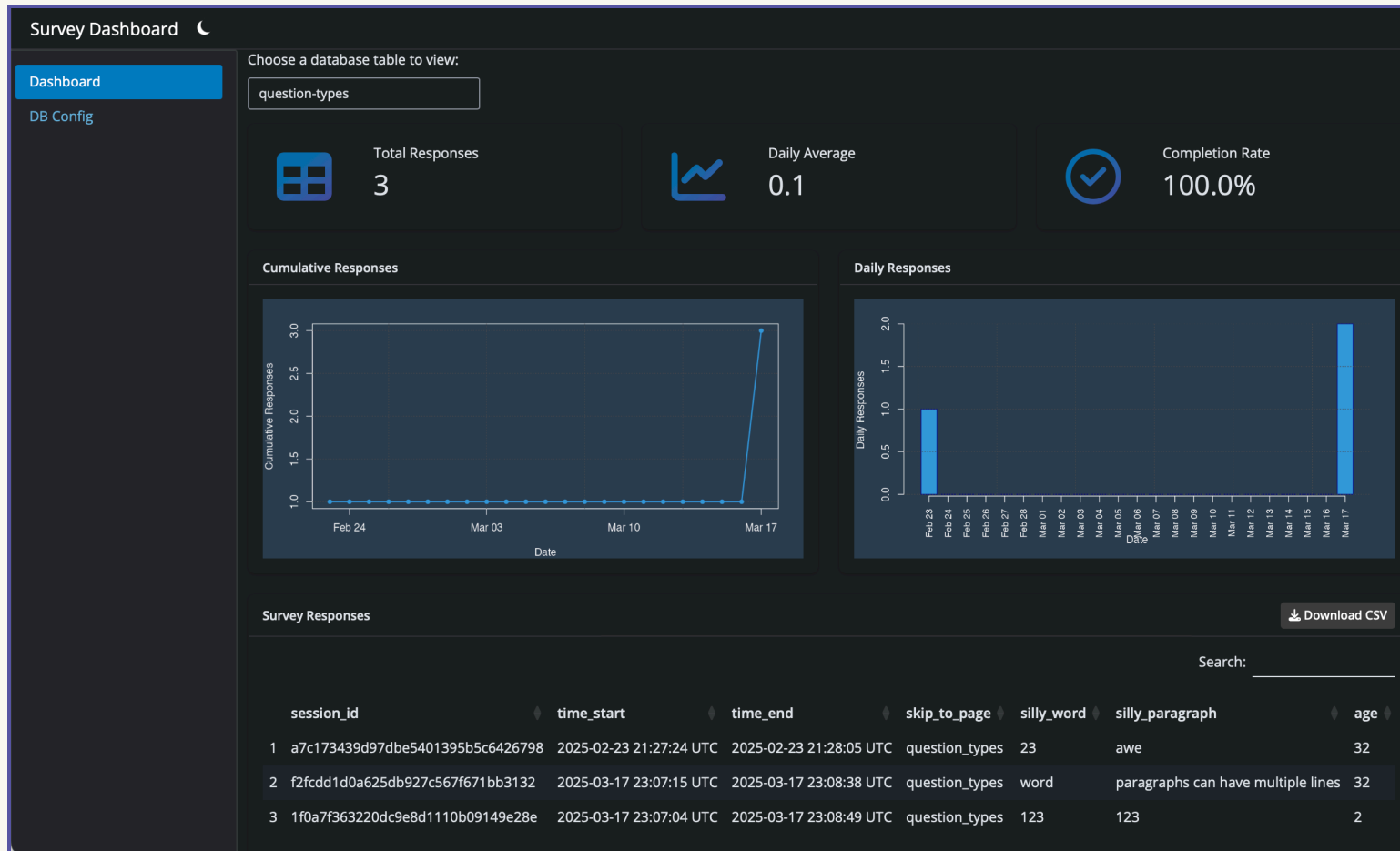
postgres

User:

postgres.hwumdvjnymbrgeemvrhq

Credentials

Data Management in surveydown Dashboard



Survey Data

Future Work

GUI for defining surveys

More question types

...and more!

You can contribute!

Check out surveydown.org for more!