

The SchemaWind

An open-source project for developers to use working schema of various fields of technological domains as well as for contributing to resolving the rising demands of models, database structures, and schema in the back-end applications.



By Praabindh's Org

SchemaWind defines how data is organized within a relational database and includes logical constraints such as table names, fields, data types, and the relationships between these entities.

License

License [GPL v3](#)

Important Project Links

Project Live Demo Link

<https://praabindh.github.io/SchemaWind/>

Project GitHub Link

<https://github.com/praabindh/SchemaWind>

Project Documentation Link

<https://github.com/praabindh/SchemaWind/blob/main/README.md>

Features

- Full Schema Toggle Control
- Continuous Integration And Deployment
- Dynamic Representation
- Interactive Environment
- Free To Use And Contribute
- Developmental Workflow
- Let For Personal And Commercial Usage
- Live Previews
- Fullscreen Mode
- Cross Platform

- Zero Maintenance Downtime
- Direct File Download
- Open-Source Platform

Web Snapshots

SchemaWind | Home

The SchemaWind By Praabindh's Org

An open-source project for developers to use working schema of various fields of technological domains as well as for contributing to resolving the rising demands of models, database, and schema structures.

[GitHub Profile](#) [Schema File](#)

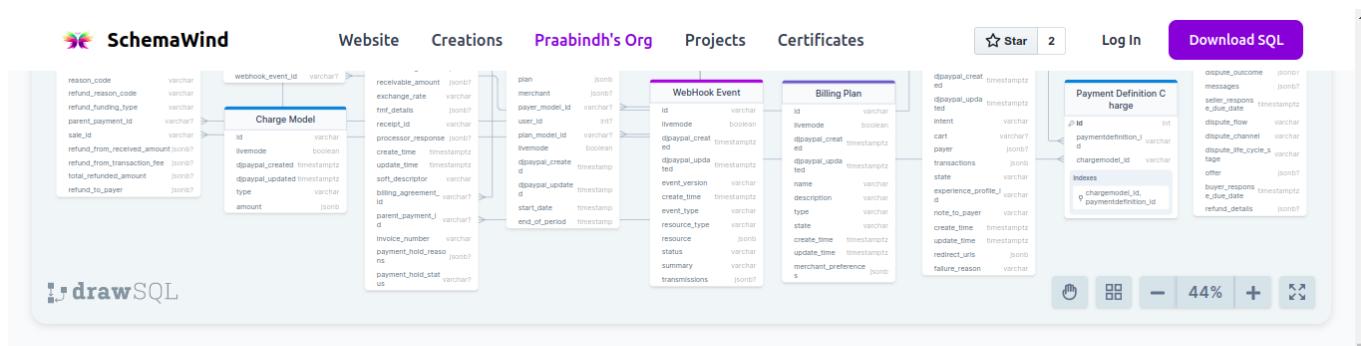
Social media icons: Netflix, Twitter, Instagram, Spotify, Discord, WhatsApp

Paypal's Schematic Portrayal By Praabindh's Org

Detailed schema diagram showing relationships between various tables:

- Refund**: Contains columns like id, livemode, dpaypal_created, dpaypal_updated, amount, state, reason, invoice_number, description, create_time, update_time, reason_code, refund_reason_code, refund_funding_type, parent_payment_id, sale_id, refund_from_received_amount, refund_to_refund, refund_to_transaction_fee, total_refunded_amount, refund_to_payer.
- WebHook Event Trigger**: Contains columns like id, remote_ip, headers, body, valid, processed, error, webhook_version, created, updated, webhook_event_id.
- Sales**: Contains columns like id, livemode, dpaypal_created, dpaypal_updated, amount, state, reason_code, protection_eligibility, protection_eligibility_type, clearing_time, transaction_fee, receivable_amount, exchange_rate, merchant, plan, processor_response, user_id, plan_model_id, plan_mode_id, dpaypal_create, dpaypal_update, start_date, end_of_period.
- Billing Agreement**: Contains columns like id, name, state, agreement_details, plan, shipping_address, override_merchant_preferences, override_charge_mode, merchant, payer_model_id, user_id, user_id, dpaypal_create, dpaypal_update, timestamp.
- Prepared Billing Agreement**: Contains columns like id, name, state, data, executed_at, canceled_at, updated, executed_agreement, user_id, user_id, time_created.
- Payer**: Contains columns like id, first_name, last_name, email, shipping_address, created, dpaypal_create, dpaypal_update, timestamp, user_id, time_created.
- Billing Plan Payment Definitions**: Contains columns like id, billingplan_id, paymentdefinition_id, indexes.
- WebHook Event**: Contains columns like id, livemode, dpaypal_create, dpaypal_updated, intent, payer, transactions, event_version, event_type, resource_type, resource, status, create_time, update_time, redirect_urls, failure_reason.
- Billing Plan**: Contains columns like id, livemode, dpaypal_create, dpaypal_updated, intent, cart, payer, transactions, event_version, event_type, resource_type, resource, status, create_time, update_time, redirect_urls, failure_reason.
- Payment Definition**: Contains columns like id, paymentdefinition_id, type, frequency_interval, frequency, cycles, amount.
- Payment Definition Charge**: Contains columns like id, paymentdefinition_id, chargechannel_id, chargemodel_id, indexes.
- Dispute**: Contains columns like dispute_id, seller_id, buyer_id, dispute_outcome, messages, seller_response, e_duty_time, dispute_flow, dispute_channel, dispute_life_cycle_s, tags, offer, buyer_response, e_duty_date, refund_details.

[drawSQL](#)



Encompass Dynamic Schemas

The Schema is a critical component for development of any application

Database schema design helps organize data into separate entities, making it easier to share a single schema within another database.

[Explore SchemaWind →](#)

[About The Developer →](#)



Deployed At

Zero Maintenance Downtime



Developers

Trusted Users Around The World



Open-Source

Free To Use And Contribute



Commercial

Personal And Commercial Usage

"SchemaWind is just incredible. It contains many predesigned schemas and table models of more remarkable technological giants. SchemaWind would be the perfect reference for your next backend application."

Praabindh Pradeep | CEO | Praabindh's Org



© 2023 SchemaWind. All Rights Reserved. Built With ❤️ By Praabindh Pradeep.



SchemaWind is an innovative open-source project designed to address the growing demands of developers in the field of back-end application development. It aims to provide a comprehensive collection of working schemas for various technological domains, enabling developers to efficiently organize and structure data within relational databases.

At its core, SchemaWind focuses on defining how data is organized and stored within a relational database. It encompasses logical constraints such as table names, fields, data types, and the relationships between these entities. By offering a standardized schema framework, the project empowers developers to streamline their database design process and enhance the overall efficiency and scalability of their applications.

SchemaWind | Netflix Schema

 **SchemaWind**

Website Creations Praabindh's Org Projects Certificates Star 2 Log In Download SQL

Netflix's Schema By Praabindh.

In 2020 Netflix deployed their first CockroachDB cluster in production. Today they have 100 production clusters and 150+ test clusters in use.

[GitHub Profile](#) [Schema File](#)







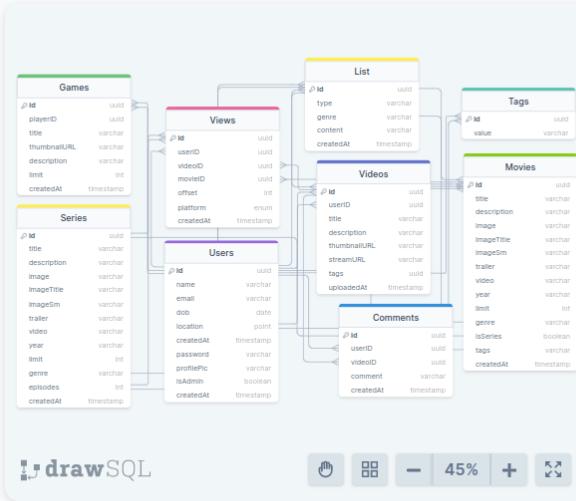



Netflix Schema Using DrawSQL

Netflix database schema portrays the “blueprint” of a database which describes how the data may relate to other tables or other data models.

- Continuous Integration And Deployment
- Developmental Workflow
- Dynamic Representation

Netflix adopted JanusGraph + cassandra + elasticsearch as their graph database infrastructure. The use of the graph database is in their digital asset management.



The project aims to cater to a wide range of technological domains, ensuring that developers have access to working schemas for different industries and applications. This includes domains such as e-commerce, finance, healthcare, social networking, and more. By providing domain-specific schemas, SchemaWind

enables developers to leverage established best practices and industry standards, saving time and effort in designing and implementing database structures.

One of the key aspects of SchemaWind is its collaborative nature. The project encourages developers to contribute their own schema designs and share their expertise with the community.

SchemaWind | Twitter Schema

The screenshot shows the SchemaWind website interface. At the top, there's a navigation bar with links for "Website", "Creations", "Praabindh's Org", "Projects", and "Certificates". There are also buttons for "Star" (with a count of 2), "Log In", and "Download SQL". The main title is "Twitter's Schema By Praabindh." Below the title, a subtitle reads: "Gizzard is Twitter's distributed data storage framework built on top of MySQL (InnoDB). InnoDB was chosen because it doesn't corrupt data. Gizzard us just a datastore." There are two buttons at the bottom left: "GitHub Profile" and "Schema File". To the right of the text is a colorful 3D illustration of a target with a bullseye, surrounded by social media icons like a megaphone, a Twitter bird, and a speech bubble. Below the illustration are six large icons representing various platforms: Netflix, Twitter, Instagram, Spotify, Discord, and WhatsApp.

The screenshot shows a detailed database schema diagram for Twitter using the DrawSQL tool. The diagram consists of several tables connected by relationships:

- Users**: Contains fields like id, name, email, password, created_at, updated_at, remember_token, and slug.
- Draft Scheduled Tweets**: Contains user_id, draft_tweet_id, and scheduled_tweet_id.
- Replies**: Contains reply_count, tweet_id, user_id, reply_content, mention_id, like, retweet, reply, views, and created_at.
- Tweets**: Contains id, content, user_id, created_at, updated_at, and category.
- Relationships**: Contains follower_id, followed_id, created_at, and updated_at.
- Liked By**: Contains tweet_id, user_id, and created_at.
- Login Activity**: Contains login_count, user_id, and login_time.
- UserAccount**: Contains id, name, email, password, birthdate, birthmonth, birthyear, phonenum, profilepicture_url, bio, website, location, verified, created_at, and two_auth.
- Retweet**: Contains tweet_id, user_id, and created_at.
- Trending**: Contains tweets_id, hashtag, and timestamp.
- Posts**: Contains tweet_id, tweet_user_id, tweet_content, like, retweet, reply, views, and created_at.
- User Languages**: Contains id, language, and varname.
- Quote Tweet**: Contains tweet_id, user_id, and created_at.
- Languages**: Contains lang_id, lang_name, and varname.
- Twittercircle**: Contains count, user_id, and friend_id.

The DrawSQL interface includes a toolbar at the bottom with various icons for zooming and navigating the diagram.

SchemaWind | Instagram Schema

 **SchemaWind**

Website Creations Praabindh's Org Projects Certificates Star 2 Log In Download SQL

Instagram's Schema By Praabindh.

PostgreSQL is the primary database of the application, it stores most of the data of the platform such as user data, photos, tags, meta-tags, etc.

[GitHub Profile](#) [Schema File](#)

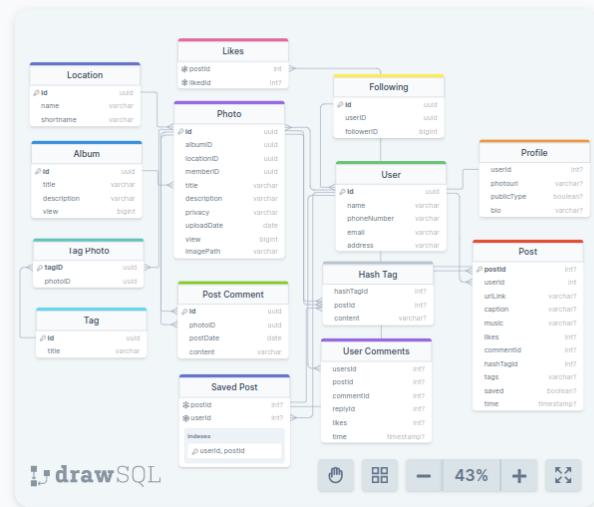



Instagram Schema Using DrawSQL

Instagram database schema portrays the “blueprint” of a database which describes how the data may relate to other tables or other data models.

- ✓ Continuous Integration And Deployment
- ✓ Developmental Workflow
- ✓ Dynamic Representation

Instagram mainly uses two backend database systems: PostgreSQL and Cassandra. Both PostgreSQL and Cassandra have mature replication frameworks that work well as a globally consistent data store. Global data neatly maps to data stored in these servers.



© 2023 SchemaWind. All Rights Reserved. Built With ❤️ By Praabindh Pradeep.



Our Project fosters a collaborative environment where developers can learn from each other, discuss best practices, and collectively work towards resolving the evolving demands of models, database structures, and schema in back-end applications. SchemaWind encourages community engagement through forums, discussions, and issue tracking, facilitating knowledge exchange and problem-solving.

SchemaWind | Spotify Schema

 **SchemaWind**

Website Creations Praabindh's Org Projects Certificates Star 2 Log In Download SQL

Spotify's Schema By Praabindh.

Spotify's storage unit is Cassandra database, consists of over 100 Cassandra clusters, each containing a nested storage system within itself.

[GitHub Profile](#) [Schema File](#)










Spotify Schema Using DrawSQL

Spotify database schema portrays the “blueprint” of a database which describes how the data may relate to other tables or other data models.

- Continuous Integration And Deployment
- Developmental Workflow
- Dynamic Representation

Historically, Spotify has been a heavy user of PostgreSQL. Since the very first versions of the Spotify backend it has been our go-to relational database and the default choice for persistent storage – when it works.



The project provides a user-friendly platform where developers can access and contribute to the repository of schemas. It offers documentation, tutorials, and guidelines to assist developers in understanding and implementing the schemas effectively. By leveraging SchemaWind, developers can accelerate their development process by utilizing pre-defined schemas and benefiting from community-driven contributions.

SchemaWind | Discord Schema

 SchemaWind

Website Creations Praabindh's Org Projects Certificates Star 2 Log In Download SQL

Discord's Schema By Praabindh.

Cassandra was the only database that fulfilled all of Discord's requirements, as they can add nodes to scale it and it can tolerate a loss of nodes without any impact on the application.

[GitHub Profile](#) [Schema File](#)




Discord Schema Using DrawSQL

Discord database schema portrays the “blueprint” of a database which describes how the data may relate to other tables or other data models.

- Continuous Integration And Deployment
- Developmental Workflow
- Dynamic Representation

Discord deals with 4 billion messages sent through the platform per day by its millions of users. The company runs a set of NoSQL database clusters (powered by ScyllaDB) but its real-time nature means that the databases need to respond to queries as quickly as possible.



© 2023 SchemaWind. All Rights Reserved. Built With ❤️ By Praabindh Pradeep.



Our open-source initiative aims to improve the efficiency, consistency, and maintainability of back-end applications by providing a rich collection of working schemas and fostering a collaborative environment for developers worldwide. Join SchemaWind today and be part of the thriving community shaping the future of back-end application development!

SchemaWind | WhatsApp Schema

 SchemaWind

Website Creations Praabindh's Org Projects Certificates Star 2 Log In Download SQL

WhatsApp's Schema By Praabindh.

WhatsApp uses an Erlang-based, distributed DBMS (Database Management System) called Mnesia. This DBMS provides benefits that many traditional databases don't such as: Real-time key/value lookup.

[GitHub Profile](#) [Schema File](#)

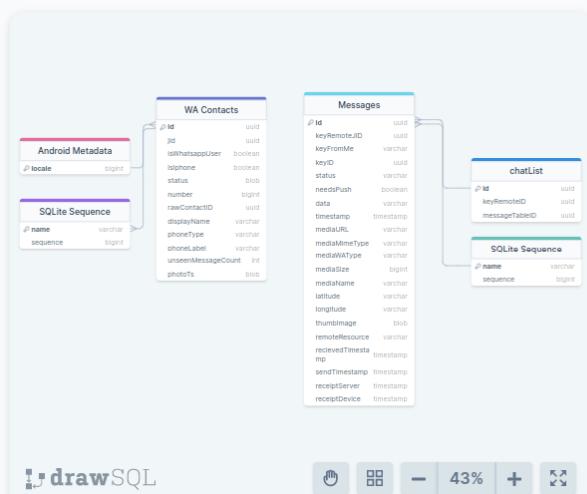



WhatsApp Schema Using DrawSQL

WhatsApp database schema portrays the "blueprint" of a database which describes how the data may relate to other tables or other data models.

- Continuous Integration And Deployment
- Developmental Workflow
- Dynamic Representation

Web socket handler from a sender gets in touch with the message service, and the messages will be stored in Cassandra. The message service then communicates with Kafka. Those messages are saved in Kafka with an instruction to send to the group.



The diagram illustrates the WhatsApp schema using DrawSQL. It shows the following tables and their relationships:

- WA Contacts**: Contains columns for ID, locale, jid, isWhatsAppUser, isPhone, status, number, rawContactID, displayName, phoneType, phoneLabel, unreadMessageCount, and photoTs.
- Android Metadata**: Contains columns for ID and locale.
- SQLite Sequence**: Contains columns for name and sequence.
- Messages**: Contains columns for ID, keyRemoteID, keyFromMe, keyID, status, needsPush, date, timestamp, mediaURL, mediaMimeType, mediaAltType, mediaSize, mediaName, latitude, longitude, thumbnail, remoteResource, receivedTimestamp, sendTimestamp, receiptServer, receiptDevice, and receiptTime.
- chatList**: Contains columns for ID, keyRemoteID, messageTableID, and sequence.



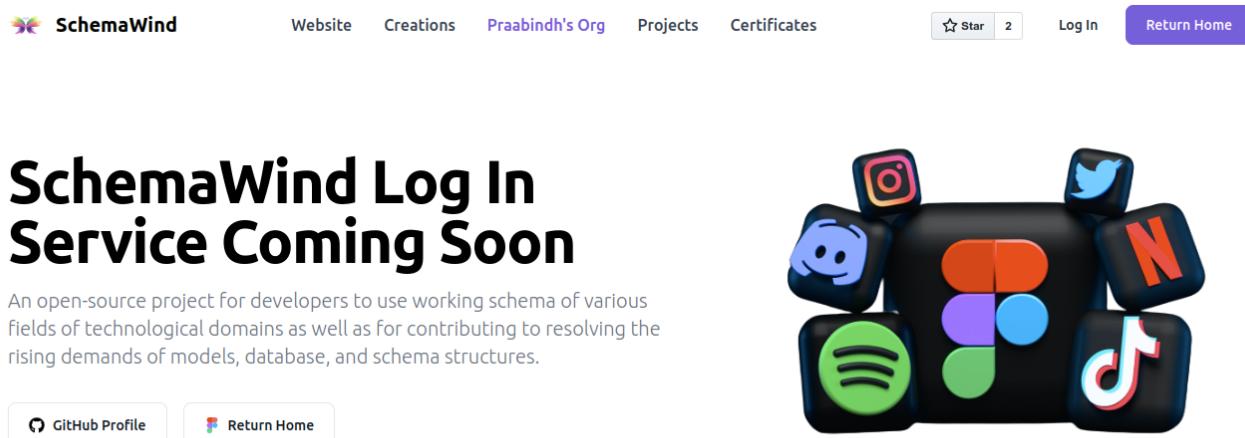
© 2023 SchemaWind. All Rights Reserved. Built With ❤️ By Praabindh Pradeep.



SchemaWind acts as a vital tool for developers, offering a comprehensive repository of working schemas, promoting community collaboration, and contributing to the resolution of the growing demands for models, database structures, and schemas in the back-end application development process. SchemaWind is a comprehensive open-source project that provides developers with ready-to-use schemas, fosters community

collaboration, promotes best practices in database design, and facilitates efficient development of back-end applications across various technological domains.

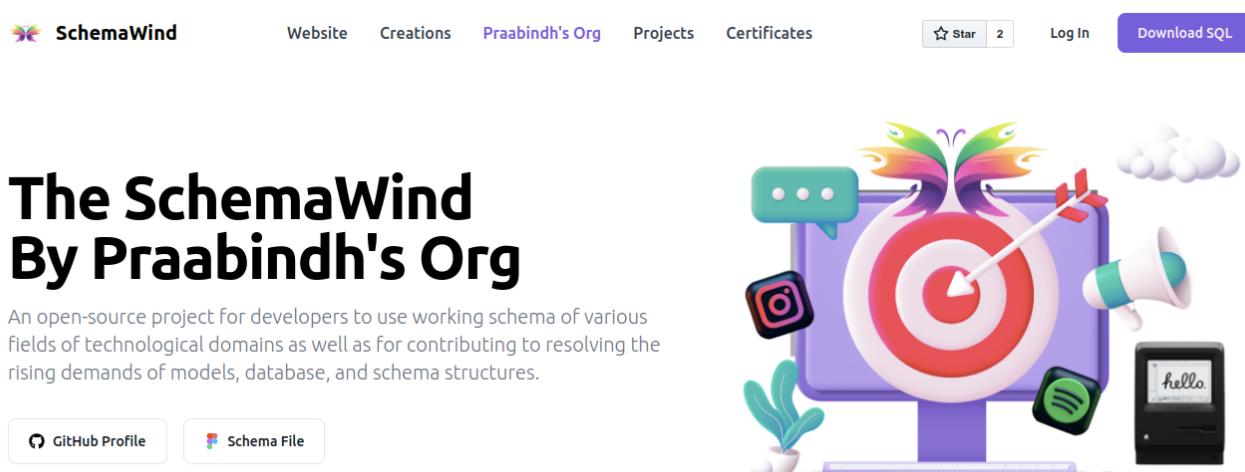
SchemaWind | Login



The page features a large title "SchemaWind Log In Service Coming Soon" in bold black font. Below it is a subtitle: "An open-source project for developers to use working schema of various fields of technological domains as well as for contributing to resolving the rising demands of models, database, and schema structures." At the bottom left are two buttons: "GitHub Profile" and "Return Home". On the right is a graphic of various social media and technology icons (Instagram, Twitter, Spotify, etc.) arranged in a cluster.



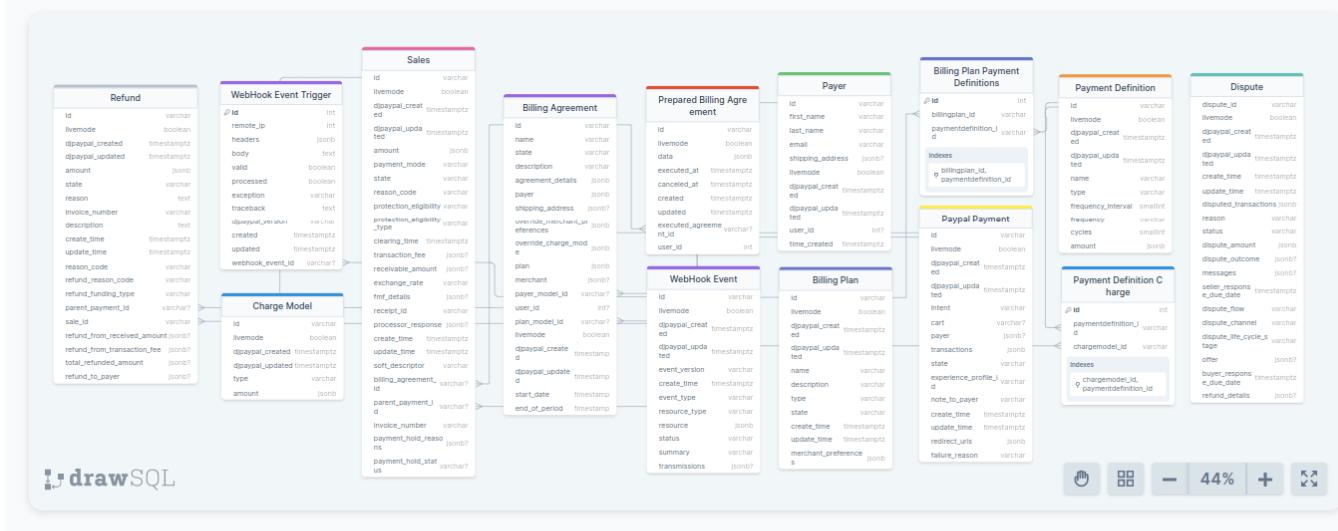
SchemaWind | Home Complete



The page features a large title "The SchemaWind By Praabindh's Org" in bold black font. Below it is a subtitle: "An open-source project for developers to use working schema of various fields of technological domains as well as for contributing to resolving the rising demands of models, database, and schema structures." At the bottom left are two buttons: "GitHub Profile" and "Schema File". On the right is a colorful 3D-style illustration of a target with a bullseye, surrounded by a megaphone, a smartphone displaying "Hello", and various social media icons.



Paypal's Schematic Portrayal By Praabindh's Org



Encompass Dynamic Schemas

The Schema is a critical component for development of any application

Database schema design helps organize data into separate entities, making it easier to share a single schema within another database.

[Explore SchemaWind →](#)

[About The Developer →](#)



Deployed At

Zero Maintenance Downtime



Developers

Trusted Users Around The World



Open-Source

Free To Use And Contribute



Commercial

Personal And Commercial Usage

"SchemaWind is just incredible. It contains many predesigned schemas and table models of more remarkable technological giants. SchemaWind would be the perfect reference for your next backend application."

Praabindh Pradeep | CEO | Praabindh's Org

SchemaWind

© 2023 SchemaWind. All Rights Reserved. Built With ❤️ By Praabindh Pradeep.



Authors

- Praabindh Pradeep - [@praabindhp](#)
- Praabindh P - [@praabindh](#)

Built During The Internship Phase Of Product Engineer In [Codingmart Technologies](#), Coimbatore, Tamil Nadu, India.

FAQ

Is This Project Open-Source?

Yes, this project is 100% open-source and can be used for personal as well as commercial usage.

Can I Contribute To This Project?

Yes, contributions can be added to the project by users who has a verified SchemaWind account.

How Can I Download The .SQL File I Want From The Project?

You should go to the respective schema page of SchemaWind, and at the top right corner, you can see a Download SQL button. When clicked the SQL file starts downloading.

How Can I Start Contributing?

You should signup with SchemaWind, and verify your identity, and once your account gets verified, you could start contributing to the project.

Will I Get The Credits For Contributing?

Of course, verified users would get full credits for the verified schema contributed to the project if any are not found with any plagiarism.