Consider Only all Blue tables are chosen after whole team review.

***USERS\_SQL***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | mobile\_ number | Display Name | Display Picture | User Status  Status is a predefined key word f sql | Verification Code | Verified | Created At  (For the admin panel to control users) |
| Auto generated | Primary key | Required | Nullable | Required  Default: Available | Nullable | Required  Default: false | Required  Default: Date.now |
| SERIAL | VARCHAR(20) | VARCHAR(100) | TEXT | TEXT | UUID | BOOLEAN | TIMESTAMP WITH TIME ZONE |

Sedki & Mostafa’s Comments:

1. **unseen counter**: this entry will allow us to get how many msgs the given user didn't see instead of iterating over all msgs to get it
2. we need to **status** one for {online, offline}, and the second one for the **text status**.

Omar, Rodaina & Noura’s Comments:

1. Unseen Counter: should be implemented by aggregating on the messages because each message contains its status, and the counter should be synchronized with the actual messages status, that’s an unreliable shortcut, so no changes required to the current model.

2. Online Status: always implemented using Sockets for real time update when the user opens the client (mobile app) it fires an event to the backend and the backend broadcasts this event to the rest of the clients so they update his online status whenever someone opens a conversation view with him, and same before he closes the app, database entries should never include real time attributes, thus no change is required to the current model.

Sedki comments:

1. Unseen counter: this counter should be updated regularly, so iterating over all messages just to update the counter once will be computationally very expensive, the proposed way of updating this counter is that for every message to arrive while the user is offline (arrive at the server but not consumed by the offline user) we increment this counter by 1(thus we need the old counter value). This way we actually updating 1 record for each new message.
2. Actually I’m convinced.

***CHATS - SQL***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Mobile Number | Is Archived | Is Deleted | Created At |
| Primary key | Foreign key | Required  Default: FALSE | Required  Default: FALSE | Required  Default:  Date.now |
| SERIAL | VARCHAR(20) | BOOLEAN | BOOLEAN | TIMESTAMP WITH TIME ZONE |

***GROUP\_CHATS - SQL***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Mobile Number | Is Archived | Is Admin | Is Deleted | Display picture | Group name |
| Primary key | Foreign key | Required | Required | Required |  |  |

Omar, Rodaina & Noura’s Comments:

GROUP\_CHATS: should be splitted into two tables, one containing the general info of the group that is the same for all the users (ID, Display Picture, Group Name, Created By, Created At) and another containing the rest and a foreign key to the group chat id from the general table.

GROUP\_CHATS Added Columns:

Created By: the creator of the group.

Created At: Group creation date.

Ex:

***GROUP\_CHATS - SQL***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Group Name | Display Picture | Created By | Created At |
| Primary key | Required | Nullable | Foreign Key | Required  Default: Date.now |
| SERIAL | VARCHAR(100) | TEXT | VARCHAR(20) | TIMESTAMP WITH TIMEZONE |

***GROUP\_CHAT\_MEMBERS - SQL***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Group Chat ID | Mobile Number | Is Archived | Is Admin | Join At | Left At |
| Primary key | Foreign key | Foreign key | Required  Default: FALSE | Required  Default: FALSE | Required | Nullable |
| SERIAL | INT | VARCHAR(20) | BOOLEAN | BOOLEAN | TIMESTAMP WITH TIME ZONE | TIMESTAMP WITH TIME ZONE |

Sedki comments:

1. Since we don’t have a local db at the client side, just server side dbs, so some situations will diverse from the actual whatsapp behavior, for example when a user join a group he shouldn’t see the previous messages before joining, just the newer ones. In our schema we imply that all members of a group have the same history of messages which is not true  
   Proposed solution: add Join timestamp column.
2. Left group and Left at Index can be joined in a single column Left timestamp which can be null if the user didn’t leave the group yet, actually this way is more consistent giving the prev. point.

***BLOCKED - SQL***

|  |  |  |
| --- | --- | --- |
| ID | Blocker Mobile Number | Blocked Mobile Number |
| Auto generated | Foreign key | Foreign key |
| SERIAL | VARCHAR(20) | VARCHAR(20) |

***REPORTED - SQL***

|  |  |  |
| --- | --- | --- |
| ID | Reporter Mobile Number | Reported Mobile Number |
| Auto generated | Foreign key | Foreign key |
| SERIAL | VARCHAR(20) | VARCHAR(20) |

***MESSAGES - Mongo***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Chat ID | Sender Mobile Number | Type  Array | Timestamp | Geographical Location | Data | Status | Replying to message | payload |
| Auto generated | Foreign Key | Foreign Key | Required  1- Text  2- Voice note  3- Photo  4- Gif  5- Video  6- Contact  7- Location  8- Document  9- Tagged  10-Reply | Auto generated | Nullable | Required |  |  |  |

***GROUP\_MESSAGES - Mongo***

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Group Chat ID | Sender Mobile Number | Type  Array | Timestamp | Geographical Location | Data | Received by  array | Read by array | Tagged in array | Replying to message | payload |
| Auto generated | Foreign Key | Foreign Key | Required  1- Text  2- Voice note  3- Photo  4- Gif  5- Video  6- Contact  7- Location  8- Document  9- Tagged  10-Reply | Auto generated | Nullable | Required | If count = 0 then message status is sent  If count less than the entire number of participants then message status is received finally status is seen |  |  |  |  |

***STORIES - MongoDB***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Mobile Number | Data | Duration | Type | Expiration Date | Viewed by array |
| Auto generated | Foreign key | Required | Required | Required | Required |  |