# Title: Calendar suggestion Redis structure

For emails, one suggestion card

For  Leads/Contacts , suggestion card <= emails associated to Lead

**Redis Key structure -**

**Email Conversation**

Key\_name:Acc\_id:email\_conversation\_id= {

   “conversation\_id” 123,

   ‘email\_id’ : 8,

  “start\_date\_time” : timestamp,

  “end\_date\_time: timestamp,

  “timezone”: “”,

  “entities”: [

 {

entity\_type: “Lead”,

entity\_id: 123,

Is\_primary: true

 },

{

entity\_type: “Contact”,

entity\_id: 124,

Is\_primary: false

 },

{

entity\_type: “User”,

entity\_id: 5,

Is\_primary: false

 }

  ],

  “expiry\_date\_time”: timestamp(utc)

}

**Lead**

Key\_name:Acc\_id:Lead:Lead\_id= {

email\_ids: timestamp(start\_time),

email\_ids: timestamp(start\_time)

}

Payload sent by ML team -

**CREATE/Update** - will happen in sidekiq worker

Create -

1. Create conversation key
2. Get primary entities, foreach :
   1. Check if lead\_key exist, else create.
   2. If lead\_key exist
      1. Lead\_key\_expiry >= conv\_key\_expiry
         1. Add key in lead hash
      2. Lead\_key\_expiry < conv\_key\_expiry
         1. Update expiry
         2. Add key in lead hash

Update:

1. update\_payload
2. Get primary entities, foreach :
   1. Check if lead\_key exist, else create.
   2. If lead\_key exist
      1. Lead\_key\_expiry >= conv\_key\_expiry
         1. Add key in lead hash
      2. Lead\_key\_expiry < conv\_key\_expiry
         1. Update expiry
         2. Add key in lead hash

Cleanup: (will be done at the time of loading suggestion/expiry)

1. Conv\_key will be deleted by expiry
2. Lead\_key will be deleted by expiry
3. On loading suggestion for leads -

Check if key exists

1. If key not exist, do zrem
2. If exist, process

1. On status change, delete lead\_key. // not needed will be handled by expiry time.

**Helpful redis commands -**

**$redis\_freddy**.zadd(**'**myzset**'**, **5**, **'**one**'**)

=> **true**

[5] pry(main)> **$redis\_freddy**.zadd(**'**myzset**'**, **4**, **'**two**'**)

=> **true**

[6] pry(main)> **$redis\_freddy**.zadd(**'**myzset**'**, **3**, **'**three**'**)

=> **true**

[7] pry(main)> **$redis\_freddy**.zadd(**'**myzset**'**, **6**, **'**six**'**)

=> **true**

[8] pry(main)> **$redis\_freddy**.zrangebyscore(**'**myzset**'**, **'**2**'**, **'**+inf**'**)

=> [**"**three**"**, **"**two**"**, **"**one**"**, **"**six**"**]

[9] pry(main)> **$redis\_freddy**.zrangebyscore(**'**myzset**'**, **'**4**'**, **'**+inf**'**)

=> [**"**two**"**, **"**one**"**, **"**six**"**]

[10] pry(main)> **$redis\_freddy**.zrangebyscore(**'**myzset**'**, **'**(4**'**, **'**+inf**'**)

=> [**"**one**"**, **"**six**"**]

[11] pry(main)> **$redis\_freddy**.zrangebyscore(**'**myzset**'**, **'**9**'**, **'**+inf**'**)

=> []

[12] pry(main)> **$redis\_freddy**.zrangebyscore(**'**myzset**'**, **'**0**'**, **'**+inf**'**)

=> [**"**three**"**, **"**two**"**, **"**one**"**, **"**six**"**]

[13] pry(main)> **$redis\_freddy**.zadd(**'**myzsetstr**'**, **'**5**'**, **'**five**'**)

=> **true**

[14] pry(main)> **$redis\_freddy**.zadd(**'**myzsetstr**'**, **'**4**'**, **'**four**'**)

=> **true**

[15] pry(main)> **$redis\_freddy**.zadd(**'**myzsetstr**'**, **'**1**'**, **'**one**'**)

=> **true**

[16] pry(main)> **$redis\_freddy**.zadd(**'**myzsetstr**'**, **'**3**'**, **'**three**'**)

=> **true**

[17] pry(main)> **$redis\_freddy**.zrangebyscore(**'**myzset**'**, **'**2**'**, **'**+inf**'**)

=> [**"**three**"**, **"**two**"**, **"**one**"**, **"**six**"**]

[18] pry(main)> **$redis\_freddy**.zrangebyscore(**'**myzsetstr**'**, **'**2**'**, **'**+inf**'**)

=> [**"**three**"**, **"**four**"**, **"**five**"**]

[19] pry(main)> **$redis\_freddy**.zrangebyscore(**'**myzsetstr**'**, **'**4**'**, **'**+inf**'**)

=> [**"**four**"**, **"**five**"**]

[20] pry(main)> **$redis\_freddy**.zrangebyscore(**'**myzsetstr**'**, **'**2**'**, **'**4**'**)

=> [**"**three**"**, **"**four**"**]

[21] pry(main)> **$redis\_freddy**.zrangebyscore(**'**myzsetstr**'**, **'**1**'**, **'**4**'**)

=> [**"**one**"**, **"**three**"**, **"**four**"**]

[22] pry(main)> **$redis\_freddy**.zrem(**'**myzsetstr**'**, [**'**one**'**, **'**four**'**, **'**five**'**])

=> **3**

[23] pry(main)> **$redis\_freddy**.zrangebyscore(**'**myzsetstr**'**, **'**1**'**, **'**4**'**)

=> [**"**three**"**]

[24] pry(main)> **$redis\_freddy**.zrem(**'**myzsetstr**'**, [**'**three**'**, **'**two**'**])

=> **1**

[25] pry(main)> **$redis\_freddy**.exists(**'**myzsetstr**'**)

=> **false**

[26] pry(main)> **$redis\_freddy**.exists(**'**myzset**'**)

=> **true**

[27] pry(main)> **$redis\_freddy**.zrem(**'**myzsetstr**'**, [**'**three**'**, **'**two**'**])

=> **0**

[28] pry(main)> **$redis\_freddy**.exists(**'**myzsetstr**'**)

=> **false**

[29] pry(main)> **$redis\_freddy**.zadd(**'**myzsetstr**'**, **'**1**'**, **'**one**'**)

=> **true**

[30] pry(main)> **$redis\_freddy**.exists(**'**myzsetstr**'**)

=> **true**

[31] pry(main)> **$redis\_freddy**.zrem(**'**myzsetstr**'**, [**'**three**'**, **'**two**'**])

=> **0**

[32] pry(main)> **$redis\_freddy**.exists(**'**myzsetstr**'**)

=> **true**

[33] pry(main)> **$redis\_freddy**.zrem(**'**myzsetstr**'**, [**'**one**'**, **'**four**'**, **'**five**'**])

=> **1**

[34] pry(main)> **$redis\_freddy**.exists(**'**myzsetstr**'**)

=> **false**

[35] pry(main)>