DBeaver: SQL Queries: -

1. *For finding license count from database for Foundation summary page:*

**select**

**distinct**(**trim**(**lower**(**unnest**(sd.licenses)))) **as** license\_count

**from**

snyk\_dependency sd

**inner** **join** project\_repository pr **on**

sd.repository\_id = pr.repository\_id

**inner** **join** project p **on**

p.project\_id = pr.project\_id

**inner** **join** vd\_project vp **on**

vp.project\_id::text = p.external\_id::text

**where**

vp.status = 'published'

/\*and pr.enable = true\*/

**and** p.foundation\_id = 'a092M00001GkhBlQAJ'

1. *For finding language count from database for Foundation summary page:*

**select**

**distinct** **trim**(**lower**(**language**))

**from**

repository\_language rl

**inner** **join** project\_repository pr **on**

rl.repository\_id = pr.repository\_id

**inner** **join** project p **on**

p.project\_id = pr.project\_id

**inner** **join** vd\_project vp **on**

vp.project\_id::text = p.external\_id::text

**where**

vp.status = 'published'

**and** pr.**enable** = **true**

**and** p.foundation\_id = 'a0941000002wBz4AAE'

1. *For finding dependency count from database for Foundation summary page:*

**select**

**count**(\*)

**from**

snyk\_dependency sd

**inner** **join** project\_repository pr **on**

sd.repository\_id = pr.repository\_id

**inner** **join** project p **on**

p.project\_id = pr.project\_id

**inner** **join** vd\_project vp **on**

vp.project\_id::text = p.external\_id::text

**where**

vp.status = 'published'

/\*and pr.enable = true\*/

**and** p.foundation\_id = 'a09410000182dD2AAI'

1. *For Issue count from database as per timestamp:*

**select**

p.project\_id,

**count** (\*)

**from**

project p

**join** vd\_project vd **on** p.external\_id::TEXT = vd.project\_id::TEXT

**Join** project\_repository prepo **ON** p.project\_id::TEXT=prepo.project\_id::TEXT

**join** repository repo **on** prepo.repository\_id::TEXT=repo.repository\_id::text

**join** snyk\_issue sissue **on** sissue.repository\_id::TEXT=repo.repository\_id::TEXT

**where**

vd."name" = 'Zowe dev'

**and** prepo.enable = **true**

**and** sissue.status = 'open'

**and** sissue.applicable = **true**

**and** sissue.created\_at >= date\_part('epoch'::text, **timestamp** '2021-01-01T00:00:00+00:00')

**and** sissue.created\_at <= date\_part('epoch'::text, **timestamp** '2021-01-31T00:00:00+00:00')

**group** **by** p.project\_id

1. *For Issue count from database:*

**select**

p.project\_id,

**count**(\*)

**from**

project p

**join** vd\_project vd **on** p.external\_id::TEXT = vd.project\_id::TEXT

**Join** project\_repository prepo **ON** p.project\_id::TEXT=prepo.project\_id::TEXT

**join** repository repo **on** prepo.repository\_id::TEXT=repo.repository\_id::text

**join** snyk\_issue sissue **on** sissue.repository\_id::TEXT=repo.repository\_id::TEXT

**where**

vd."name" = 'CoMPAS'

**and** prepo.enable = **true**

**group** **by** p.project\_id

1. *To get project\_id and the total open issue count*

**select**

p.project\_id,

**count**(\*)

**from**

project p

**join** vd\_project vd **on** p.external\_id::TEXT = vd.project\_id::TEXT

**Join** project\_repository prepo **ON** p.project\_id::TEXT=prepo.project\_id::TEXT

**join** repository repo **on** prepo.repository\_id::TEXT=repo.repository\_id::text

**join** snyk\_issue sissue **on** sissue.repository\_id::TEXT=repo.repository\_id::TEXT

**where**

vd."name" = 'CoMPAS'

**and** prepo.enable = **true**

**and** sissue.status = 'open'

**group** **by** p.project\_id

1. *To get details of any project.*

**select**

vd.project\_id,

p.project\_id ,

prepo.repository\_id,

vd."name",

repo.url,

sissue.status,

sissue.severity,

sissue.is\_fixable,

sissue.updated\_at,

sissue.created\_at

**from**

project p

**join** vd\_project vd **on** p.external\_id::TEXT = vd.project\_id::TEXT

**Join** project\_repository prepo **ON** p.project\_id::TEXT=prepo.project\_id::TEXT

**join** repository repo **on** prepo.repository\_id::TEXT=repo.repository\_id::text

**join** snyk\_issue sissue **on** sissue.repository\_id::TEXT=repo.repository\_id::TEXT

**where**

p.sfdc\_project\_id <> ''

**and** prepo."enable" ='true'

**and** sissue.created\_at >= date\_part('epoch'::text, **timestamp** '2020-05-02T00:00:00+00:00')

**and** sissue.created\_at <= date\_part('epoch'::text, **timestamp** '2020-05-11T00:00:00+00:00')

**and** vd.project\_id ='84be6ace-80e7-4cc7-b2a7-a88163a52d0f'

1. *Pending projects list.*

**select** vp.name,foundation\_id

**from** project p

**join** vd\_project vp **on** vp.project\_id::text = p.external\_id::text

**where** vp.status= 'pending'

**order** **by** vp.name

1. *Repository git\_sha value.*

**select** repository\_id, url, github\_sha\_value **from** repository **where** "name" **like** 'Repo2Duplicate'

1. *To get details of any repository.*

**select** \* **from** repository **where** name = 'Repo\_1'

1. *Contributor details for that project.*

/\*select \* from vd\_project vp where name = 'VDSKTest101'\*/

**select** \* **from** vd\_contributor vc **where** project\_id = 'ece1bd7c-8fce-4e7c-b0b2-33b8a421da17'

1. *Create table(Do not use this)*

**CREATE** **TABLE** public.repository\_statistics1 (

repository\_id uuid,

total\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

total\_open\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

total\_open\_fixable\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

total\_fixed\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

high\_open\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

high\_open\_fixable\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

high\_fixed\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

medium\_open\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

medium\_open\_fixable\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

medium\_fixed\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

low\_open\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

low\_open\_fixable\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

low\_fixed\_issues **integer** **NOT** **NULL** **DEFAULT** 0,

total\_dependencies **integer** **NOT** **NULL** **DEFAULT** 0,

last\_scanned\_date **bigint**,

last\_scanned\_status text,

error text,

**primary** **KEY**(repository\_id)

);

1. *Project status and all details.*

**SELECT** \* **from** snyk\_issue si

**join** repository r **on** si.repository\_id = si.repository\_id

**join** project\_repository pr **on** pr.repository\_id = r.repository\_id

**join** project p **on** pr.project\_id = p.project\_id

**join** vd\_project vp **on** p.external\_id::text = vp.project\_id::text

**where**

vp.name = 'CoMPAS'

/\*fixed issues, total issues\*/

1. *Repository status and all details.*

**select**

\*

**from**

snyk\_issue si

**where**

si.repository\_id = '3b9fc3fc-5ce7-4156-8712-ff5b288858e3'

**and** status = 'open'

1. *To get Most Active in fixing vulnerabilities*

**select** vp.**name**, sub.\* **from** vd\_project vp

**join** (

**select**

vd.project\_id , **count** **as** cnt

**from**

project p

**join** vd\_project vd **on** p.external\_id::**TEXT** = vd.project\_id::**TEXT**

**Join** project\_repository prepo **ON** p.project\_id::**TEXT**=prepo.project\_id::**TEXT**

**join** repository repo **on** prepo.repository\_id::**TEXT**=repo.repository\_id::**text**

**join** snyk\_issue sissue **on** sissue.repository\_id::**TEXT**=repo.repository\_id::**TEXT**

**where**

prepo.**enable** = **true**

**and** vd.status = 'published'

**and** sissue.status = 'fixed'

--and sissue.is\_fixable = true

**group** **by** vd.project\_id

**order** **by** **count** **desc** ) sub **on** vp.project\_id = sub.project\_id

**order** **by** sub.cnt **desc**

1. *Reverse query on the vulnerabilities that impact the most projects*

**select**

cwe\_identifiers, cve\_identifiers/\*, count\*/

**from**

project p

**join** vd\_project vd **on** p.external\_id::**TEXT** = vd.project\_id::**TEXT**

**Join** project\_repository prepo **ON** p.project\_id::**TEXT**=prepo.project\_id::**TEXT**

**join** repository repo **on** prepo.repository\_id::**TEXT**=repo.repository\_id::**text**

**join** snyk\_issue sissue **on** sissue.repository\_id::**TEXT**=repo.repository\_id::**TEXT**

**where**

prepo.**enable** = **true**

**and** vd.status = 'published'

**and** sissue.status = 'open'

**and** sissue.is\_fixable = **true**

**group** **by** sissue.cwe\_identifiers, sissue.cve\_identifiers

/\*order by count desc\*/

1. *To get organization data*

**select**

**count**(\*)

**from**

project p

**join** organization\_project op **on** op.project\_id::**text** = p.project\_id::**text**

**join** github\_organization go2 **on** go2.organization\_id::**TEXT** = op.organization\_id::**TEXT**

**join** vd\_project vd **on** p.external\_id::**TEXT** = vd.project\_id::**TEXT**

**Join** project\_repository prepo **ON** p.project\_id::**TEXT**=prepo.project\_id::**TEXT**

**join** repository repo **on** prepo.repository\_id::**TEXT**=repo.repository\_id::**text**

**join** snyk\_issue sissue **on** sissue.repository\_id::**TEXT**=repo.repository\_id::**TEXT**

**where**

vd.project\_id = 'f6d7b138-4860-4fca-ae49-98e12646cfc5'

**and** prepo.**enable** = **true**

**and** sissue.status = 'open'

**and** sissue.applicable = **true**

**and** go2.organization\_name = 'hyperledger-archives'