Indepth DQL documentation - <https://dgraph.io/docs/query-language/>

**DGraph Installation and setup(tutorial):**

* Install docker
* Pull latest image: docker pull dgraph/dgraph:latest
* Pull standalone image: docker pull dgraph/standalone:latest
* Verify: docker images
* For mac ensure to use correct file path is used (check docker resources - file sharing in settings): docker run --name <CONTAINER\_NAME> -d -p "8080:8080" -p "9080:9080" -v <DGRAPH\_DATA\_PATH>:/dgraph dgraph/standalone:latest
* NOTE: Select a name <CONTAINER\_NAME> for you Docker container and create a directory <GRAPH\_DATA\_PATH> that will hold the Dgraph data on your local file system.
* Launch ratel UI: docker run --name ratel -d -p "8000:8000" dgraph/ratel:latest
* In the UI connect to the localhost:8080 using the latest Ratel version

**DGraph Installation for production:**<https://dgraph.io/docs/deploy/installation/single-host-setup/>

**Dgraph Schema (copy paste):**

To create a new node = <newnode\_name>:value\_type .

To create a new edge = <newedge\_name>:[uid]

To create a new double edge = <newedge\_name>:[uid] @reverse .

#Nodes

<acceptance\_criteria>: [string] .

<app\_name>: string .

<common\_bugs>: [string] .

<domain>: string .

<feature\_name>: string .

<platform\_name>: string .

<quality>: string .

<region>: string .

<software\_type>: string .

<subdomain\_name>: string .

<user\_story>: string .

#Edges

<subdomains>: [uid] @reverse .

<user\_stories>: [uid] @reverse .

<software\_types>: [uid] @reverse .

<regions>: [uid] @reverse .

<requirements>: uid @reverse .

<functional>: [uid] @reverse .

<non\_functional>: [uid] @reverse .

<apps>: [uid] @reverse .

<platforms>: [uid] @reverse .

#Default schema (do not modify or delete)

<dgraph.drop.op>: string .

<dgraph.graphql.p\_query>: string @index(sha256) .

<dgraph.graphql.schema>: string .

<dgraph.graphql.xid>: string @index(exact) @upsert .

type <dgraph.graphql> {

dgraph.graphql.schema

dgraph.graphql.xid

}

type <dgraph.graphql.persisted\_query> {

dgraph.graphql.p\_query

}

**Mutation example:**

{

"set": [

{

"uid": "\_:domain1",

"domain": "Banking",

"subdomains": [

{

"uid": "\_:subdomain1",

"subdomain\_name": "Retail Banking",

"regions": [

{

"uid": "\_:region1",

"region": "India",

"apps": [

{

"uid": "\_:app1",

"app\_name": "SBI YONO",

"platforms": [

{

"uid": "\_:platform1",

"platform\_name": "Mobile",

"software\_types": [

{

"uid": "\_:software1",

"software\_type": "Frontend Development",

"requirements": {

"functional": [

{

"uid": "\_:feature1",

"feature\_name": "User Authentication",

"user\_stories": [

{

"uid": "\_:us1",

"quality": "Good",

"user\_story": "As a retail customer in India, I want to securely log into the SBI YONO app using biometric authentication (fingerprint or face recognition) on my mobile device so that I can access my bank account without remembering or entering a password.",

"acceptance\_criteria": [

"The biometric authentication setup must be completed successfully during initial login.",

"The app should authenticate the user within 2 seconds after biometric input.",

"If biometric authentication fails, the app should prompt for a PIN as a fallback method.",

"The system must lock the account after five consecutive failed authentication attempts."

],

"common\_bugs": [

"Users unable to log in due to server issues or outdated credentials."

]

},

{

"uid": "\_:us2",

"quality": "Bad",

"user\_story": "As a user, I want biometric authentication enabled in the SBI YONO app, so I can log in securely.",

"acceptance\_criteria": [

"The biometric setup should work.",

"Biometric authentication must allow access."

],

"common\_bugs": [

"Potential security gaps due to incomplete implementation."

]

}

]

},

{

"uid": "\_:feature2",

"feature\_name": "Fund Transfers",

"user\_stories": [

{

"uid": "\_:us3",

"quality": "Good",

"user\_story": "As a user, I want to transfer funds using NEFT, IMPS, and UPI so that I can send money securely and efficiently.",

"acceptance\_criteria": [

"Users must be able to select the desired transfer method.",

"Transaction confirmation should be displayed after successful transfer.",

"Failed transactions must provide a clear error message."

],

"common\_bugs": [

"Delayed transactions, failure in cross-bank transfers."

]

}

]

},

{

"uid": "\_:feature3",

"feature\_name": "Account Overview",

"user\_stories": [

{

"uid": "\_:us4",

"quality": "Good",

"user\_story": "As a user, I want to view my transaction history and account balance on a dashboard so that I can track my spending.",

"acceptance\_criteria": [

"The dashboard must load within 2 seconds.",

"Transaction history should include all transactions from the last 6 months.",

"Filters for specific dates must function correctly."

],

"common\_bugs": [

"Users unable to view recent transactions due to syncing errors."

]

}

]

},

{

"uid": "\_:feature4",

"feature\_name": "Bill Payments",

"user\_stories": [

{

"uid": "\_:us5",

"quality": "Good",

"user\_story": "As a user, I want to pay utility bills directly from the app so that I can avoid late fees.",

"acceptance\_criteria": [

"Bill payment confirmation must be sent via SMS and email.",

"Payment methods must include UPI, credit/debit card, and net banking.",

"Failed payments must offer retry options."

],

"common\_bugs": [

"Failed payments leading to service disconnection or late fees."

]

}

]

},

{

"uid": "\_:feature5",

"feature\_name": "Investment Services",

"user\_stories": [

{

"uid": "\_:us6",

"quality": "Good",

"user\_story": "As a user, I want to invest in mutual funds and fixed deposits via the app so that I can grow my savings.",

"acceptance\_criteria": [

"Investment confirmations must display immediately after transaction.",

"Portfolio details must be updated in real-time.",

"Error messages must be clear if transactions fail."

],

"common\_bugs": [

"Incorrect transaction records or delay in fund allocation."

]

}

]

},

{

"uid": "\_:feature6",

"feature\_name": "Loan Services",

"user\_stories": [

{

"uid": "\_:us7",

"quality": "Good",

"user\_story": "As a user, I want to apply for pre-approved loans and view EMIs so that I can manage my finances effectively.",

"acceptance\_criteria": [

"Loan application status must be visible in real-time.",

"EMI schedules must be automatically updated.",

"Clear error messages must display if loan application fails."

],

"common\_bugs": [

"Application rejection due to eligibility miscalculation."

]

}

]

},

{

"uid": "\_:feature7",

"feature\_name": "Agri Solutions",

"user\_stories": [

{

"uid": "\_:us8",

"quality": "Good",

"user\_story": "As a farmer, I want real-time updates on agricultural market prices so that I can make informed financial decisions.",

"acceptance\_criteria": [

"Real-time market price updates must refresh every hour.",

"Advisory services must include personalized recommendations.",

"Error messages must be displayed for failed updates."

],

"common\_bugs": [

"Lack of real-time updates for agricultural market prices."

]

}

]

}

],

"non\_functional": [

{

"uid": "\_:nfeature1",

"feature\_name": "UI Responsiveness",

"user\_stories": [

{

"uid": "\_:nus1",

"quality": "Good",

"user\_story": "As a user, I want the app to work seamlessly on any device size so that I can access it on both phones and tablets.",

"acceptance\_criteria": [

"UI should automatically adjust to screen size.",

"No layout distortions on devices with resolutions as low as 480x800."

],

"common\_bugs": [

"Lagging or distorted layout on lower-end devices."

]

}

]

},

{

"uid": "\_:nfeature2",

"feature\_name": "Performance",

"user\_stories": [

{

"uid": "\_:nus2",

"quality": "Good",

"user\_story": "As a user, I want the app to load quickly so that I can complete transactions without delays.",

"acceptance\_criteria": [

"All app features must respond within 2 seconds.",

"Transaction confirmation must be instant."

],

"common\_bugs": [

"App freezing during heavy traffic hours or on weak network connections."

]

}

]

},

{

"uid": "\_:nfeature3",

"feature\_name": "Security",

"user\_stories": [

{

"uid": "\_:nus3",

"quality": "Good",

"user\_story": "As a user, I want my data to be encrypted so that I can ensure the safety of my personal information.",

"acceptance\_criteria": [

"All sensitive data must be encrypted using AES-256.",

"2FA must be required for all high-value transactions."

],

"common\_bugs": [

"Potential data breaches if not updated regularly."

]

}

]

},

{

"uid": "\_:nfeature4",

"feature\_name": "Scalability",

"user\_stories": [

{

"uid": "\_:nus4",

"quality": "Good",

"user\_story": "As a user, I want the app to remain functional during peak traffic times so that I can access it without interruptions.",

"acceptance\_criteria": [

"App must support 10,000 simultaneous users without lag.",

"Database scaling must be automatic during peak hours."

],

"common\_bugs": [

"App crashes during peak usage times (e.g., during festival sales)."

]

}

]

},

{

"uid": "\_:nfeature5",

"feature\_name": "Accessibility",

"user\_stories": [

{

"uid": "\_:nus5",

"quality": "Good",

"user\_story": "As a visually impaired user, I want screen reader compatibility so that I can navigate the app easily.",

"acceptance\_criteria": [

"Screen readers must read all text accurately.",

"Colorblind-friendly options must be available."

],

"common\_bugs": [

"Text unreadable or not optimized for colorblind users."

]

}

]

}

]

}

}

]

}

]

}

]

}

]

}

]

}

]

}

**Query Example**:

1. View all data -   
   {

allDomains(func: has(domain)) {

uid

domain

subdomains {

uid

subdomain\_name

regions {

uid

region

apps {

uid

app\_name

platforms {

uid

platform\_name

software\_types {

uid

software\_type

requirements {

functional {

feature\_name

user\_stories {

user\_story

quality

common\_bugs

acceptance\_criteria

}

}

non\_functional {

feature\_name

user\_stories {

user\_story

quality

common\_bugs

acceptance\_criteria

}

}

}

}

}

}

}

}

}

}

1. Adding data

mutation

{

"Functional": {

"uid": "0x123",

"functional\_user\_stories": [

{"uid": "0x456"}

]

}

}

query

{

functional(func: has(functional\_user\_stories)) {

uid

functional\_user\_stories {

user\_story

}

}

}

**MongoDB Schema:**

{

"Domain": "string",

"Subdomains": [

{

"Subdomain Name": "string",

"Regions": [

{

"Region Name": "string",

"Platforms": [

{

"Platform Name": "string",

"Software Types": [

{

"Software Type Name": "string",

"Requirements": {

"Functional": [

{

"Feature Name": "string",

"User Stories": [

{

“\_id”: “”

"Quality": "string",

"User Story": "string",

"Acceptance Criteria": ["string"],

"Common Bugs": ["string"],

"App Names": ["string"]

}

]

}

],

"Non-Functional": [

{

"Feature Name": "string",

"User Stories": [

{

“\_id”: “”

"Quality": "string",

"User Story": "string",

"Acceptance Criteria": ["string"],

"Common Bugs": ["string"],

"App Names": ["string"]

}

]

}

]

}

}

]

}

]

}

]

}

]

}

**Problems + solutions:**

<https://github.com/dgraph-io/pydgraph?tab=readme-ov-file#install> - important documentation read into upsert.

<acceptance\_criteria>: [string] .

<app\_name>: string .

<apps>: [uid] @reverse .

<common\_bugs>: [string] .

<dgraph.drop.op>: string .

<dgraph.graphql.p\_query>: string @index(sha256) .

<dgraph.graphql.schema>: string .

<dgraph.graphql.xid>: string @index(exact) @upsert .

<domain>: string @index(exact) .

<domains>: [uid] @reverse .

<domains\_root>: [uid] @reverse .

<feature\_name>: string .

<functional>: [uid] @reverse .

<non\_functional>: [uid] @reverse .

<platform\_name>: string .

<platforms>: [uid] @reverse .

<quality>: string .

<region>: string .

<regions>: [uid] @reverse .

<requirements>: uid @reverse .

<root>: string @index(exact) .

<software\_type>: string .

<software\_types>: [uid] @reverse .

<subdomain\_name>: string .

<subdomains>: [uid] @reverse .

<user\_stories>: [uid] @reverse .

<user\_story>: string .

type <dgraph.graphql> {

dgraph.graphql.schema

dgraph.graphql.xid

}

type <dgraph.graphql.persisted\_query> {

dgraph.graphql.p\_query

}

{

"set": [

{

"uid": "\_:root",

"root": "Root",

"domains\_root": [

{

"uid": "\_:domain1",

"domain": "Banking",

"subdomains": [

{

"uid": "\_:subdomain1",

"subdomain\_name": "Retail Banking",

"regions": [

{

"uid": "\_:region1",

"region": "India",

"apps": [

{

"uid": "\_:app1",

"app\_name": "SBI YONO",

"acceptance\_criteria": [

"App should support biometric authentication for secure login.",

"App should support balance check, fund transfer, and bill payments."

],

"common\_bugs": [

"App crashes during balance check.",

"Slow performance during fund transfer."

],

"platforms": [

{

"uid": "\_:platform1",

"platform\_name": "Mobile",

"software\_types": [

{

"uid": "\_:software1",

"software\_type": "Frontend"

}

]

}

]

}

]

}

]

}

]

},

{

"uid": "\_:domain2",

"domain": "Healthcare",

"subdomains": [

{

"uid": "\_:subdomain2",

"subdomain\_name": "Patient Management",

"regions": [

{

"uid": "\_:region2",

"region": "USA",

"apps": [

{

"uid": "\_:app2",

"app\_name": "HealthCare Tracker",

"acceptance\_criteria": [

"App should track patient health data.",

"App should provide real-time notifications for critical health changes."

],

"common\_bugs": [

"Notification delay issues.",

"UI bugs in health data visualization."

],

"platforms": [

{

"uid": "\_:platform2",

"platform\_name": "Web",

"software\_types": [

{

"uid": "\_:software2",

"software\_type": "Backend"

}

]

}

]

}

]

}

]

}

]

},

{

"uid": "\_:domain3",

"domain": "Education",

"subdomains": [

{

"uid": "\_:subdomain3",

"subdomain\_name": "Online Courses",

"regions": [

{

"uid": "\_:region3",

"region": "Europe",

"apps": [

{

"uid": "\_:app3",

"app\_name": "EduLearn",

"acceptance\_criteria": [

"App should support interactive video lessons.",

"App should track student progress and provide quizzes."

],

"common\_bugs": [

"Video buffering issues.",

"Incorrect quiz results."

],

"platforms": [

{

"uid": "\_:platform3",

"platform\_name": "Mobile",

"software\_types": [

{

"uid": "\_:software3",

"software\_type": "Middleware"

}

]

}

]

}

]

}

]

}

]

}

]

}

]

}