ProtonDB User Documentation

Welcome to ProtonDB — a lightweight document-based database designed for developers and technical users who prefer simplicity and performance. This documentation covers all the available commands, their use cases, and expected outputs, and explains the basic structure of queries.

Query Structure

Every query in ProtonDB follows the format:

<object>.<operation>(<argument>)

Where:
- **object:** the target entity (e.g., db, collection, user defined collection, profile)
- **operation**: the action to perform (e.g., create, drop, insert, update)
- **argument**: optional data or parameters to use for the operation

Database Commands

1. db.create()

Description: Creates a new database

Example: db.create("users")

Expected Output: Database 'users' created

2. db.drop()

Description: Deletes an existing database

Example: db.drop("users")

Expected Output: Database 'users' dropped

3. db.use()

Description: Switches of the current database context

Example: db.use("users")

Expected Output: Switched to database: users

4. db.list()

Description: Lists of all available databases

Example: database.list()

Expected Output:

users
logs

Collection Commands

1. collection.create()

Description: Creates a new collection

Example: collection.create("customers")

Expected Output: Collection 'customers' created

2. collection.drop()

Description: Deletes a collection

Example: collection.drop("customers")

Expected Output: Collection 'customers' dropped

3. collection.list()

Description:

list() - Lists all collections in current database
list(<database>) - Lists all collections in specified database

Example: collection.list()

Expected Output:

customers
orders

Document Commands

Example Collection: "document"

1. <collection >.insert()

Description:

insert(<document>) - Inserts a document
insert([<document1>, <document2>, ...]) - Inserts an array of document

Example: document.insert({"name": "John"})

Expected Output: Inserted 1

2. <collection>.print()

Description:

print() – Displays of all documents
print(<condition>) – Displays of documents based on specified condition

Example: document.print()

Expected Output:

```
{
        "name": "John"
}
```

3. <collection>.update()

Description: Updates matching documents

update(<action>, <data>) - Updates documents with the data and action
update(<action>, <data>, <condition>) - Updates documents with the data and action based on the specified condition.

Example: document. update (add, "name":"John", age = 30)

Expected Output: Document updated 1

4. <collection>.remove()

Description:

remove() – Removes all documents
remove(<condition>) – Removes documents based on specified condition

Example: document.remove(name = John)

Expected Output: Document removed 1

Flags:

action	- add | drop | alter
data	- {"key": value}
condition	- key <operator> value
operators	- (< | <= | > | >= | = | !=)

Note: data is {"key"} for update(drop, data, condition)

Profile Commands

1. profile.create()

Description: Creates a new user profile

Example: profile.create("admin420","1234","admin")

Expected Output: Profile admin 'admin' created

2. profile.delete()

Description: Deletes a profile

Example: profile.delete("admin420")

Expected Output: Profile 'admin' deleted

3. profile.grant()

Description:

grant(<username>) - Grants access to current database
grant(<username>, <database>) – Grants access to specified Database

Example: profile.grant("user1")

Expected Output: Access granted to logs for user1

4. profile.revoke()

Description:

revoke(<username>) - Revoke access to current database
revoke(<username>, <database>) – Revoke access to specified Database

Example: profile.revoke("user1","logs")

Expected Output: Access revoked from logs for user1

5. profile.list()

Description: Lists of all existing profiles

Example: profile.list()

Expected Output:

Admin admin 2025-07-01T19:43:36.3325257Z
user1 user 2025-07-01T19:43:36.3325257Z

ProtonDB Shell

Connection

Host — The IP address or domain name of the server running ProtonDB.
Default: "127.0.0.1" (localhost)

Port — The port on which the ProtonDB server is listening.
Default: 9090

Username — The name of the user profile to authenticate with. Required for accessing user-specific privileges

Password — The password associated with the username. It's used to validate and generate a secure checksum for login.

Input

- **Single line** – collection.create("Student")
- **Multi-line** – Student.insert({
  "name": "John",
  "age": 15,
  "class": "B"
  })

  Note: Input is only confirmed when ended with ")"

Commands

1. Help

Description: Displays of database help message

Command: --help, :h

2. Version

Description: Displays of ProtonDB version

Command: --version, :v

3. Quit

Description: Close and quit ProtonDB

Command: quit, :q

4. Clear Screen

Description: Clears console screen

Command: cls