# Database-related

| | |
|---|---|
| Category | Notes |
| Created | @October 4, 2023 1:24 PM |
| Parent item | Own Notes |
| Status | Archived |
| Updated | @December 11, 2023 11:32 AM |

## Migrate db

Sometimes, even in developer mode, changes we made are not immediately reflected in the database. To make sure it does, run:

```
bench --site [site-name] migrate
```

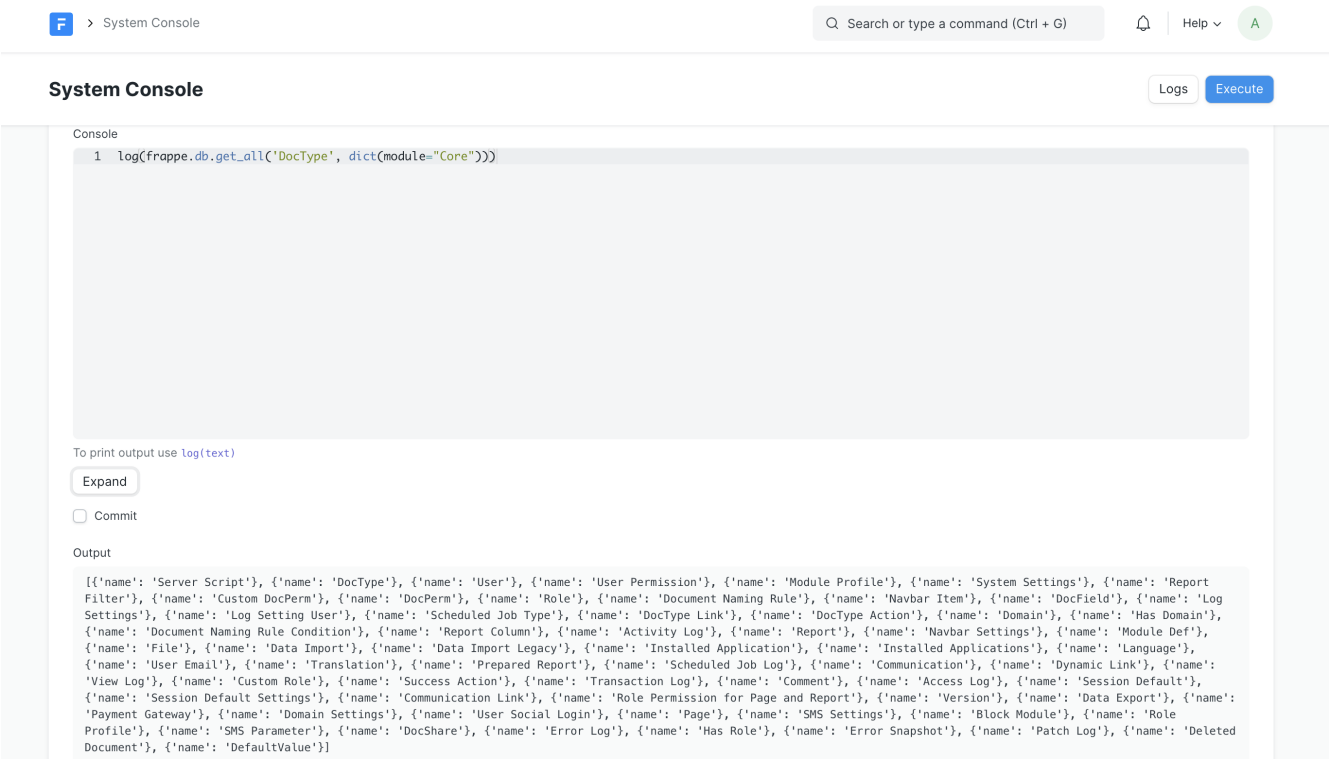
```
bench --site [site-name] clear-cache
```

## Query through Desk

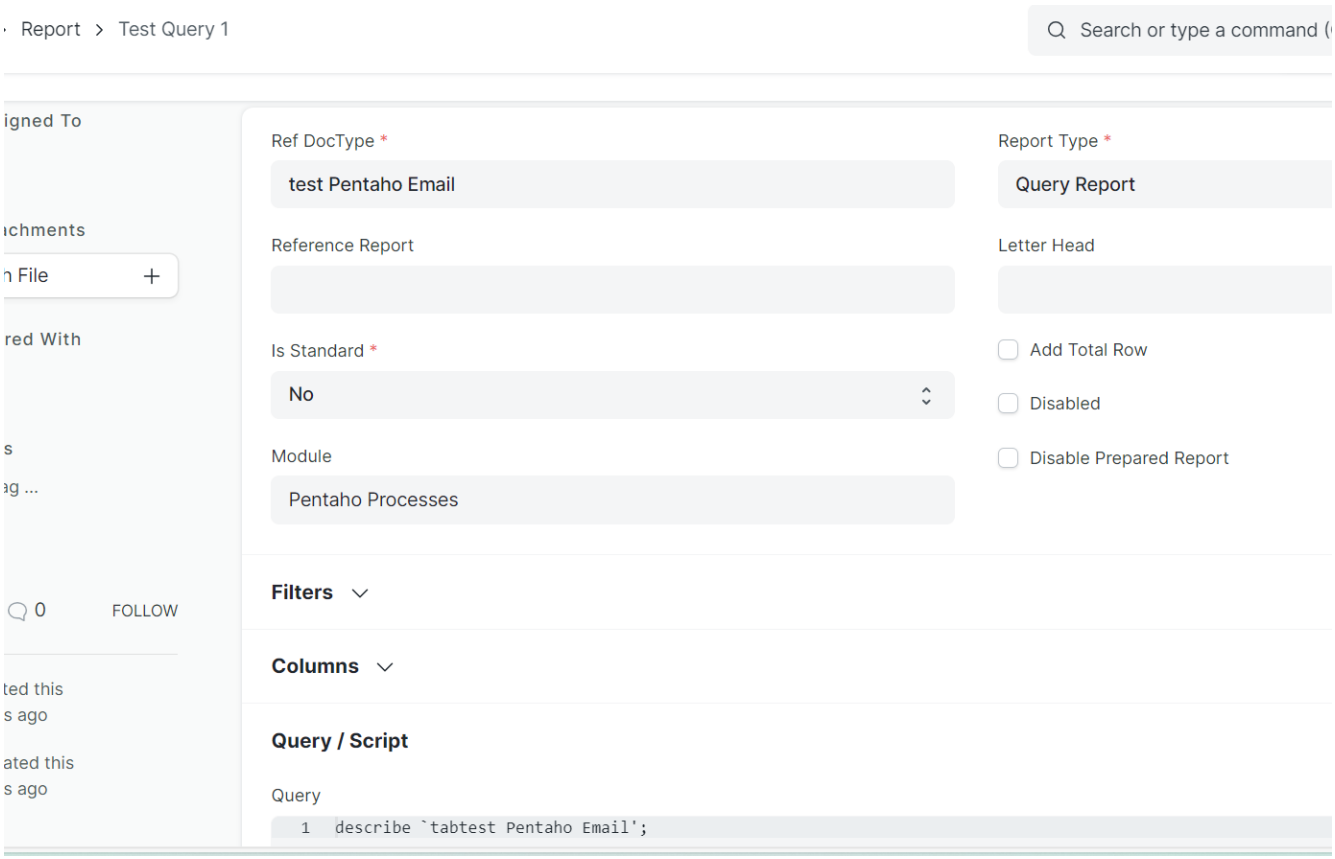
There are several ways, but it is limited to read-only operations (e.g. SELECT).

1. Using System Console

   System Console (frappeframework.com)




2. Using Query Report




3. Database Access (Need Frappe Cloud $50+)

   Database Access (frappecloud.com)

## ▼ Query through DBeaver

If your VirtualBox VM is set to use a "Bridged Adapter", it means that the VM will get an IP address from the same network as your host machine (Windows 11). Here's how you can connect DBeaver to MariaDB in this scenario:

### Step 1: Find VM's IP Address

1. **Get IP Address:**
   - Start your Debian VM.
   - Open a terminal and use the following command to find its IP address:

     ```
     ip a | grep inet | grep global
     ```

   - Note down the IP address assigned to your VM.

### Step 2: Configure MariaDB on Debian

1. **Allow External Connections:**
   - Edit the MariaDB configuration file.

     ```
     sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
     ```

   - Change the `bind-address` directive to `0.0.0.0` or comment it out with `#`.
   - Save and exit the editor.
2. **Restart MariaDB:**

   ```
   sudo systemctl restart mariadb
   ```

3. **Create a Database User:**

- Log into MariaDB.

```
mysql -u root -p
```

- Create a user and grant privileges. Replace `your_username`, `your_password`, and `your_database` accordingly. Check the `site_config.json` of the site to get the database name.

```
CREATE USER 'your_username'@'%' IDENTIFIED BY 'your_password';
GRANT ALL PRIVILEGES ON your_database.* TO 'your_username'@'%';
FLUSH PRIVILEGES;
EXIT;
```

▼ ***Server ERPNext credentials***

```
CREATE USER 'tcs-user'@'%' IDENTIFIED BY 'Ruby1G/29A5';
GRANT ALL PRIVILEGES ON covalminipc_db.* TO 'tcs-user'@'%';
FLUSH PRIVILEGES;
EXIT;
```

### Step 3: Configure Firewall on Debian (if applicable)

Ensure that the firewall allows connections to the MariaDB port (default 3306).

```
sudo ufw allow 3306/tcp
sudo ufw reload
```

### Step 4: Connect DBeaver to MariaDB

1. **Install DBeaver:**
   - Download and install DBeaver from the official website.
2. **Create a New Connection:**
   - Open DBeaver.
   - Click on "New Database Connection" (or navigate to "Database" > "New Database Connection").
   - Choose "MariaDB" as the database type and click "Next".
3. **Configure Connection Settings:**
   - **Host:** Use the IP address of the VM you noted down earlier.
   - **Port:** Use the default MariaDB port (3306).
   - **Username & Password:** Use the credentials of the user you created in MariaDB.
   - Click "Test Connection" to ensure everything is set up correctly, and then "Finish" to complete the setup.

### Step 5: Interact with the Database

- You should now see your MariaDB connection in the "Database Navigator" pane in DBeaver.
- You can interact with your database – create tables, insert data, run queries, etc.

### Notes:

- Ensure that your VM is running when trying to connect via DBeaver.
- If you encounter issues, double-check each step, especially user privileges, firewall settings, and network configurations.
- Always ensure that your database is secure, especially when allowing external connections. Use strong passwords and consider additional security measures.

## ▼ Changing Database Name

▼ Before creating site

Yes, changing the `db_name` is safest before the site is in use to avoid any potential issues with data consistency,

user access, or application functionality. If you're setting up a new Frappe site and wish to specify a particular database name, you can do so during the site creation process.

### Creating a New Frappe Site with a Specific Database Name:

1. **Create a New Site:**
   Use the
   `bench new-site` command and specify the `-db-name` parameter to set a custom database name.

   ```
   bench new-site [site-name] --db-name [custom_db_name]
   ```

   Replace `[site-name]` with the desired name for your new site and `[custom_db_name]` with your preferred database name.
2. **Enter MySQL Root Password:**
   You'll be prompted to enter the MySQL root password to create the new database.
3. **Set Admin Password:**
   You'll also be prompted to set the Administrator password for the new Frappe site.
4. **Complete Setup:**
   Once the site is created, complete the setup wizard in the web interface to configure your new site.

### Notes:

- Ensure that the custom database name adheres to MySQL naming conventions and does not conflict with existing database names.
- The database user and password will be generated automatically and stored in the `site_config.json` file in the site's directory.
- If you're using a version control system, remember to exclude sensitive information (like `site_config.json`) from being tracked to protect your credentials.

### Additional Tips:

- **Backup:** Even though it's a new site, it's a good practice to take a backup after setup and keep it safe.
- **Documentation:** Document the database name and its purpose for future reference.
- **Validation:** Validate the site functionality and ensure that the database is being utilized as expected.

By following this process, you can create a new Frappe site with a specific, meaningful database name from the outset, avoiding the need to change it later when the site is in use. Always ensure to follow best practices for database management and Frappe/ERPNext administration to maintain the integrity and security of your data and applications.

▼ After creating site

Changing the database name of a Frappe site (or any application) can be risky and is generally not recommended once the site is in use, as it can cause issues with connectivity and data integrity if not done carefully. However, if you decide to proceed with this, make sure to take all necessary precautions.

### Precautions:

1. **Backup:** Always create a backup of your database before making any changes.
2. **Downtime:** Ensure that you can afford some downtime, as the site will be unavailable during the process.
3. **Documentation:** Make sure to document all steps and changes for future reference.
4. **Validation:** Validate all functionalities and data after making the change.

### Steps to Change `db_name` in Frappe:

(***Very not recommended***)

1. **Backup Database:**

   ```
   bench --site [site-name] backup
   ```

2. **Rename Database:**
   - Access your MySQL/MariaDB shell:

     ```
     mysql -u [username] -p
     ```

   - Rename the database:

     ```
     RENAME DATABASE `_762565ef8d152680` TO `new_db_name`;
     ```

3. **Update Site Config:**
   - Edit the `site_config.json` file located in `frappe-bench/sites/[site-name]`:

     ```
     {
       "db_name": "new_db_name",
       "db_password": "existing_password",
       ...
     }
     ```

4. **Migrate and Check:**
   - Run migration to ensure everything is in order:

     ```
     bench --site [site-name] migrate
     ```

   - Check the site functionality thoroughly.

### Alternative: Using Aliases in DBeaver

If you want to avoid the risk of changing the database name, you can use aliases in your database management tool (like DBeaver) to refer to your databases with a more meaningful name.

### In DBeaver:

- **Create a New Connection:** Set up a new connection to your MySQL/MariaDB server.
- **Edit Connection:** Right-click on the connection and select "Edit Connection".
- **Set Alias:** In the "General" tab, set an alias for your connection that is meaningful to you.

This way, you can refer to your databases with meaningful names in DBeaver without changing the actual database names and risking issues with your Frappe sites.

### Note:

Always ensure that you thoroughly test all functionalities after making changes to database configurations, especially in a staging environment before applying changes to production.

# ▼ Backup & Restore DB

## ▼ Normal backup & restore

Certainly! Backing up and restoring data is crucial for any system, especially for ERP systems like ERPNext running on the Frappe framework. Here's a step-by-step guide for both backing up and restoring data on your Frappe site:

### Backing Up Data on Your Frappe Site:

1. **Login to the Frappe Bench:**
   - Navigate to your Frappe bench directory using the terminal.

     ```
     cd path_to_your_bench
     ```

2. **Backup Command:**
   - Execute the backup command:

     ```
     bench backup
     # or
     bench --site [site-name] backup --with-files
     ```

   - This will create a backup of both your database and files associated with your site.
3. **Locate Backup Files:**
   - The backup files will be saved in the `sites/[your_site_name]/private/backups` directory. You'll find two files:
     - A `.sql` file (database backup)
     - A `.tar` file (file backup)

## Restoring Data on Your Current Site:

1. **Navigate to Bench Directory:**
   - As before, navigate to your Frappe bench directory.
2. **Restore Command:**
   - Use the restore command with the path to your backup:

     ```
     bench --site [your_site_name] restore path_to_your_backup.sql
     ```

3. **Migrate After Restore:**
   - After restoring, you should run the migrate command to ensure everything is in sync:

     ```
     bench --site [your_site_name] migrate
     ```

## Restoring Data on a Hypothetical New Site:

1. **Setup a New Frappe Environment:**
   - If you haven't already, set up a new Frappe environment on the server or machine where you want to restore the site.
2. **Create a New Site:**
   - Use the `bench` command to create a new site:

     ```
     bench new-site new-site-name
     ```

3. **Restore the Backup:**
   - Navigate to the directory where your backup files are located.
   - Use the `bench` command to restore the database and files:

     ```
     bench --site new-site-name restore path-to-your-database-backup.sql.gz
     ```

4. **Restore Site Config:**
   - If you've backed up the site's configuration (`site_config_backup.json`), you should copy its contents to the `site_config.json` file of the new site.
5. **Migrate the Database:**
   - After restoring, you'll need to migrate the database to ensure that the schema is updated:

     ```
     bench --site new-site-name migrate
     ```

6. **Restore Public and Private Files:**

- Extract the contents of `files.tar` (public files) to the `./sites/new-site-name/public/files/` directory.
- Extract the contents of `private-files.tar` (private files) to the `./sites/new-site-name/private/files/` directory.

7. **Rebuild the Website:**
   - This step ensures that all static files are generated and the website is rendered correctly:

```
bench --site new-site-name build
```

8. **Start the Bench:**
   - If everything looks good, you can start the bench using:

```
bench start
```

9. **Verify the Restoration:**
   - Access the new site using a web browser to ensure that the data, configurations, and files have been restored correctly.

**Note:** Always ensure you have a backup of your data before performing any restore operations. It's also a good practice to test the restore process in a staging environment before applying it to a production environment.

## ▼ Bench Partial Restore

The `bench partial-restore` command is a specialized tool in the Frappe framework that allows for selective restoration of specific tables or sections of your database from a backup. Here are some examples and scenarios where this feature can be effectively utilized:

### Example 1: Restoring a Specific Table

Imagine you've made significant changes to the "Sales Invoice" table in your ERPNext instance, but later realized that these changes were not beneficial. Instead of restoring the entire database, you can use the `bench partial-restore` command to only restore the "Sales Invoice" table from a backup taken before the changes.

**Command**:

```
bench --site your_site_name partial-restore /path/to/backup/file.sql --only Sales_Invoice
```

### Example 2: Excluding Tables from Restoration

Let's say you've taken a full backup of your database, but you don't want to restore certain tables like "Logs" or "Error Messages" because they contain a lot of transient data that isn't necessary for your current needs.

**Command**:

```
bench --site your_site_name partial-restore /path/to/backup/file.sql --exclude Logs,Error_Messages
```

### Example 3: Migrating Specific Data to a New Site

If you're setting up a new site and only want to migrate specific tables, like "Customers" and "Suppliers", from an existing site, you can use the `partial-restore` command to achieve this.

**Command**:

```
bench --site new_site_name partial-restore /path/to/backup/file.sql --only Customers,Suppliers
```

### Benefits:

1. **Flexibility**: Allows for targeted restoration, ensuring that only the desired data is restored.

2. **Efficiency**: Faster than a full restore, especially when dealing with large databases.
3. **Safety**: Reduces the risk of unintentionally overwriting or losing data from other tables.

### Challenges:

1. **Dependencies**: Some tables might have dependencies on others. Restoring them in isolation might lead to inconsistencies or errors.
2. **Version Mismatches**: If the backup is from an older version of ERPNext or Frappe, there might be schema differences that can cause issues during restoration.

In all scenarios, it's crucial to have a clear understanding of the data structure and dependencies to ensure a smooth restoration process. Always test the restoration in a staging environment before applying it to a production site.

## ▼ Backup & restore on a different bench

If the site you're restoring is on a different bench, the process becomes a bit more involved, especially if the benches are on different servers or machines. Here's a step-by-step guide to handle this scenario:

1. **Transfer Backup Files:**
   - First, ensure that the backup files (database, public files, private files, and site config) are available on the machine where the new bench is located. You can use tools like `scp` (for Linux) or file transfer methods to move these files.
2. **Clone Custom Apps to the New Bench:**
   - If the custom apps are hosted on a version control system like Git, clone each of the repositories into the `apps` directory of your new bench.

     ```
     git clone repository-url
     ```

   - If they aren't in a version control system, you'll need to manually transfer the app directories from the old bench's `apps` folder to the new bench's `apps` folder.
3. **Create a New Site on the New Bench:**
   - Use the `bench` command to create a new site:

     ```
     bench new-site new-site-name
     ```

4. **Install the Custom Apps on the New Site:**
   - For each custom app, use the `bench` command to install it on the new site:

     ```
     bench --site new-site-name install-app app-name
     ```

5. **Restore the Backup:**
   - Navigate to the directory where your backup files are located on the new machine.
   - Use the `bench` command to restore the database:

     ```
     bench --site new-site-name restore path-to-your-database-backup.sql.gz
     ```

6. **Restore Site Config (if needed):**
   - If you've backed up the site's configuration (`site_config_backup.json`), you should copy its contents to the `site_config.json` file of the new site.
7. **Migrate the Database:**
   - After restoring, run the migrate command to ensure that all the database schemas are updated and in sync with the apps:

     ```
     bench --site new-site-name migrate
     ```

8. **Restore Public and Private Files:**
    - Extract the contents of `files.tar` (public files) to the `./sites/new-site-name/public/files/` directory.
    - Extract the contents of `private-files.tar` (private files) to the `./sites/new-site-name/private/files/` directory.
9. **Start the Bench:**
    - If everything looks good, you can start the bench using:

```
bench start
```

10. **Verify the Restoration:**
    - Access the new site using a web browser to ensure that the data, configurations, and files have been restored correctly.

**Note:** If the benches are on different versions or have different configurations, you might encounter issues. It's always a good idea to ensure that both benches are on the same version of Frappe and have similar configurations to avoid potential problems.