4.1 MySQL

Адміністрування MySQL охоплює широкий спектр практичних завдань, які забезпечують працездатність, безпеку та оптимальну швидкодію системи. Приклади можна згрупувати за основними напрямами роботи адміністратора.

По-перше, управління користувачами та правами доступу. У MySQL адміністратор створює облікові записи, задає для них паролі, а також визначає рівні доступу до баз і таблиць. Наприклад:

```
CREATE USER 'student'@'localhost' IDENTIFIED BY 'StrongPass123!';
GRANT SELECT, INSERT ON university.* TO 'student'@'localhost';
```

Це створює користувача та призначає йому права лише на читання і додавання даних у таблицях бази `university`.

По-друге, резервне копіювання та відновлення баз даних. Для цього зазвичай використовують утиліту `mysqldump` або системні інструменти. Приклад створення резервної копії:

```
mysqldump -u root -p company > company_backup.sql
```

А відновлення здійснюється так:

```
mysql -u root -p company < company_backup.sql
```

Третій напрям — оптимізація продуктивності. Адміністратор може створювати індекси, які прискорюють пошук у таблицях:

```
CREATE INDEX idx_lastname ON employees(last_name);
```

Також відстежується ефективність запитів за допомогою ключового слова `EXPLAIN`.

Четвертим прикладом є моніторинг і контроль. Виконується перегляд активних з’єднань та процесів:

```
SHOW PROCESSLIST;
```

Або перегляд статусу сервера:

```
SHOW STATUS LIKE 'Threads_connected';
```

П’ятий приклад стосується адміністрування схеми бази даних. Наприклад, додавання нової таблиці:

```
CREATE TABLE logs (
    id INT AUTO_INCREMENT PRIMARY KEY,
    event_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    description TEXT
);
```

І нарешті, важливим завданням є оновлення СУБД та налаштування конфігурації у файлі `my.cnf`, де задають параметри розміру буферів, шляхи зберігання журналів та інші системні параметри.

Отже, адміністрування MySQL включає роботу з користувачами, резервне копіювання, оптимізацію продуктивності, моніторинг, управління схемами та конфігурацію сервера.

Ось узагальнена таблиця з прикладами основних задач адміністрування MySQL та відповідних команд:

Напрям адміністрування	**Приклад завдання**	**Команда / інструкція**
Управління користувачами та доступом	Створити нового користувача з паролем	`CREATE USER 'student'@'localhost' IDENTIFIED BY 'StrongPass123!';`
	Надати права на роботу з базою `university`	`GRANT SELECT, INSERT ON university.* TO 'student'@'localhost';`
	Змінити пароль користувача	`ALTER USER 'student'@'localhost' IDENTIFIED BY 'NewStrongPass456!';`
Резервне копіювання та відновлення	Створити резервну копію бази `company`	`mysqldump -u root -p company > company_backup.sql`

	Відновити базу з резервної копії	`mysql -u root -p company < company_backup.sql`
Оптимізація продуктивності	Створити індекс для прискорення пошуку	`CREATE INDEX idx_lastname ON employees(last_name);`
	Перевірити план виконання запиту	`EXPLAIN SELECT * FROM employees WHERE last_name='Smith';`
Моніторинг та контроль	Переглянути активні процеси	`SHOW PROCESSLIST;`
	Перевірити кількість підключень	`SHOW STATUS LIKE 'Threads_connected';`
Управління схемами БД	Створити нову таблицю `logs`	`sql CREATE TABLE logs (  id INT AUTO_INCREMENT PRIMARY KEY,  event_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  description TEXT );`
Конфігурація сервера	Налаштувати параметри у файлі `my.cnf`	Напр., `innodb_buffer_pool_size=1G`
Безпека та аудит	Переглянути список користувачів і привілеїв	`SELECT user, host, authentication_string FROM mysql.user;`
	Вимкнути доступ для анонімних користувачів	`DROP USER ''@'localhost';`
	Увімкнути журналування запитів для аудиту	У `my.cnf`: `general_log=ON`, `general_log_file=/var/log/mysql/general.log`

Переглянути останні помилки	`SHOW ERRORS;`
Перевірити SSL-підключення клієнтів	`SHOW VARIABLES LIKE 'have_ssl';`

Це дає повну картину адміністративних завдань у MySQL — від управління користувачами до аудиту та безпеки.

«Best practices» для безпечного адміністрування MySQL

1. Використовувати складні паролі для всіх облікових записів та регулярно їх змінювати.
2. Видалити або деактивувати анонімні облікові записи, що створюються за замовчуванням.
3. Обмежувати привілеї користувачів за принципом «мінімально необхідних прав» (least privilege).
4. Забороняти підключення користувача `root` віддалено, дозволяти лише локальний доступ.
5. Використовувати SSL/TLS для шифрування з’єднань між клієнтами та сервером.
6. Регулярно оновлювати MySQL та застосовувати патчі безпеки.
7. Увімкнути журналування запитів (general log, error log, slow query log) для моніторингу та аудиту.
8. Налаштувати резервне копіювання з перевіркою працездатності відновлення.
9. Використовувати індекси та оптимізувати запити, щоб уникати перевантаження сервера.
10. Захищати файл конфігурації `my.cnf` від несанкціонованого доступу, встановлюючи обмежені права доступу.
11. Використовувати міжмережеві екрани (firewall) для обмеження доступу до порту MySQL лише з довірених IP-адрес.

12. Періодично переглядати список користувачів і їхні привілеї в системній таблиці `mysql.user`.

13. Виконувати моніторинг продуктивності та вчасно виявляти підозрілу активність.

Такий чекліст допомагає систематизувати рутинні та критично важливі завдання адміністратора бази даних.