

Cho phép Security Admin chọn dữ liệu nào  
được đưa vào chính sách backup.

```
CREATE TABLE backup_data_type (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL, -- Ví dụ: User Data, System Config
    description TEXT,
    is_active BOOLEAN DEFAULT TRUE
);
```

Cho phép chọn nơi lưu bản sao lưu.

```
CREATE TABLE backup_storage (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL, -- VD: Local Server, AWS S3, Google Drive, gdrive, 'ftp',
    storage_type ENUM('local','s3','gdrive','ftp'),
    endpoint VARCHAR(255) NULL, -- địa chỉ hoặc bucket
    access_key VARCHAR(255) NULL,
    secret_key VARCHAR(255) NULL,
    encryption_enabled BOOLEAN DEFAULT FALSE,
    is_active BOOLEAN DEFAULT TRUE
);
```

Bảng này để ghi lại mọi lần backup  
được chạy (thành công/thất bại).

```
CREATE TABLE backup_log (
    id INT PRIMARY KEY AUTO_INCREMENT,
    policy_id INT,
    backup_file_path VARCHAR(255),
    storage_id INT,
    status ENUM('pending','running','success','failed') DEFAULT 'pending',
    started_at DATETIME,
    finished_at DATETIME,
    message TEXT,
    FOREIGN KEY (policy_id) REFERENCES backup_policy(id),
    FOREIGN KEY (storage_id) REFERENCES backup_storage(id)
);
```

Ghi lại các lần restore để theo dõi  
RTT thực tế.

```
CREATE TABLE restore_log (
    id INT PRIMARY KEY AUTO_INCREMENT,
    backup_id INT, -- bản backup được khôi phục
    restored_by INT,
    status ENUM('running','success','failed'),
    started_at DATETIME,
    finished_at DATETIME,
    message TEXT,
    FOREIGN KEY (backup_id) REFERENCES backup_log(id)
);
```

Ghi nhận và cảnh báo khi backup thất bại  
hoặc dung lượng vượt giới hạn.

```
CREATE TABLE backup_alert (
    alert_id INT PRIMARY KEY AUTO_INCREMENT,
    policy_id INT NOT NULL,
    backup_log_id INT NULL,
    restore_log_id INT NULL,
    storage_id INT NULL,
    alert_type ENUM('failure','storage_full','restore_failed','verification_failed') NOT NULL,
    message TEXT NOT NULL,
    severity ENUM('info','warning','critical') DEFAULT 'warning',
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    resolved BOOLEAN DEFAULT FALSE,
    resolved_at DATETIME NULL,
    resolved_by NVARCHAR(100) NULL,
    CONSTRAINT fk_alert_policy FOREIGN KEY (policy_id)
        REFERENCES backup_policy(policy_id),
    CONSTRAINT fk_alert_backup FOREIGN KEY (backup_log_id),
        REFERENCES backup_log(backup_log_id),
    CONSTRAINT fk_alert_restore FOREIGN KEY (restore_log_id),
        REFERENCES restore_log	restore_log_id),
    CONSTRAINT fk_alert_storage FOREIGN KEY (storage_id)
        REFERENCES backup_storage(storage_id)
);
```

Chứa các chính sách backup do security  
admin tạo.

```
CREATE TABLE backup_policy (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    frequency ENUM('daily','weekly','monthly','custom') DEFAULT 'daily',
    custom_cron VARCHAR(100) NULL, -- biểu thức cron nếu chọn 'custom'
    retention_days INT DEFAULT 30, -- số ngày lưu trữ
    auto_delete_old BOOLEAN DEFAULT TRUE, -- tự xoá bản cũ
    rto_minutes INT DEFAULT 60, -- mục tiêu RTO (thời gian khôi phục)
    created_by INT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

Một chính sách có thể áp dụng cho nhiều  
loại dữ liệu.

```
CREATE TABLE backup_policy_data (
    policy_id INT,
    data_type_id INT,
    PRIMARY KEY (policy_id, data_type_id),
    FOREIGN KEY (policy_id) REFERENCES backup_policy(id),
    FOREIGN KEY (data_type_id) REFERENCES backup_data_type(id)
);
```

Bảng này để lưu liên kết chính sách  
với nơi lưu trữ

```
CREATE TABLE backup_policy_storage (
    policy_id INT,
    storage_id INT,
    PRIMARY KEY (policy_id, storage_id),
    FOREIGN KEY (policy_id) REFERENCES backup_policy(id),
    FOREIGN KEY (storage_id) REFERENCES backup_storage(id)
);
```

Dùng cho các bài kiểm tra định kỳ  
khả năng khôi phục.

```
CREATE TABLE backup_verification (
    id INT PRIMARY KEY AUTO_INCREMENT,
    policy_id INT,
    verification_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    status ENUM('passed','failed'),
    verified_by INT,
    notes TEXT,
    FOREIGN KEY (policy_id) REFERENCES backup_policy(id)
);
```

backup\_policy

backup\_storage

backup\_policy

backup\_log

backup\_policy\_data

backup\_policy\_storage

backup\_verification