Sample Technical Document: Automated Backup & Monitoring Script for Student Records Database

Author: Samuel Mulandu

Date: 2025-11-29

Purpose: Demonstrate ability to create scripts for database maintenance, backup, and monitoring.

1. Overview

This document describes a sample approach for automating backups, monitoring data integrity, and performing basic maintenance on a small PostgreSQL database containing student records. The database stores approximately 5,000 records with fields for student ID, name, email, and course enrollment.

2. Architecture

- Database: PostgreSQL
- Backup Tool: pg_dump
- Automation: Bash script scheduled via cron
- Monitoring: Script checks for missing or duplicate records and sends a log report
- Storage: Backups saved to a secure server folder with timestamped filenames

3. Sample Bash Script

```
#!/bin/bash
# Automated Backup and Monitoring Script for Student Records Database

# Variables
DB_NAME="student_records"
DB_USER="db_user"
BACKUP_DIR="/home/user/db_backups"
LOG_FILE="/home/user/db_backups/backup_log.txt"
DATE=$(date +%F_%H-%M-%S)

# Step 1: Backup Database
pg_dump -U $DB_USER $DB_NAME >
$BACKUP_DIR/${DB_NAME}_backup_$DATE.sql

# Step 2: Verify Backup
if [ -f "$BACKUP_DIR/${DB_NAME}_backup_$DATE.sql" ]; then
```

```
    echo "$DATE: Backup successful" >> $LOG_FILE
else
    echo "$DATE: Backup failed" >> $LOG_FILE
fi

# Step 3: Data Quality Check (Simple Example)
DUPLICATES=$(psql -U $DB_USER -d $DB_NAME -c "SELECT student_id,
COUNT(*) FROM students GROUP BY student_id HAVING COUNT(*) > 1;")
if [ -z "$DUPLICATES" ]; then
    echo "$DATE: No duplicate student IDs found" >> $LOG_FILE
else
    echo "$DATE: Duplicate student IDs found" >> $LOG_FILE
fi
```

4. Summary

This sample demonstrates:

- Ability to automate backups
- Basic monitoring of data integrity
- Structured logging and documentation
- Understanding of database operations relevant to maintenance tasks