Systems and Database Administration

Assignment

TU Dublin

TU-857/4



Configuring a production Database

# Overview

In this assignment, you are required to set up and configure a production-ready database for the scenario assigned to you (see section below on determining your assigned scenario). You will be required to create the tables and database objects specified, configure the database and produce a report outlining the decisions you have taken.

The report is split into four sections, described below. Each section should open with a discussion aimed at stakeholders describing the motivation behind your decisions and end with a step-by-step guide, aimed at database administrators, describing the steps they need to take to implement the functionality you have chosen. Your final submission will include the report, the database configuration files (a zip of the root database directory will suffice) and a single script which will generate the finished database. Screenshots and visual aids are encouraged where they help to illustrate the points made.

# Determining your Assigned Scenario

Your scenario is assigned to you based on the **last digit** of your student number. Check the last digit of your student number against the table below to determine your scenario. Please ensure you submit the correct scenario as no marks will be awarded if the wrong scenario is chosen. Please include your student number, and the table below (with the scenario you have submitted highlighted in bold) at the top of your report.

|  |  |
| --- | --- |
| Final Digit of Student Number | Scenario |
| 0, 1, 2, 3, 4 | Stockbroker |
| 5, 6, 7, 8, 9 | Human Resources |

# Creating the Database

You will need to create a new database for this assignment. This can be done by creating a new directory (I suggest placing this in the /home/postgres directory) and running the *initdb* command. See lab 1 for a refresher on how to do this. Note that whenever you go to start or stop the postgres server you will need to specify the new directory using the -D argument for any *pg\_ctl* commands

# Report Sections

Your report should be split into 4 sections, each clearly labelled, as described below.

## 1. Security (20%)

Discuss the major risks and challenges posed to maintaining the security of the database. Describe the measures that have been taken to combat these. Outline the policies and procedures which should be put in place after the database has been deployed to maintain the security of the system. Include a step-by-step guide to implement the security policies you have chosen.

## 2. Auditing (20%)

Discuss the role of auditing within the database. Discuss the objectives of the auditing policy and how the database will be configured to support these objectives. Describe the auditing options available, state which option(s) you have chosen and justify your choice. Include a step-by-step guide to implement the auditing steps you have taken.

### 3. Performance Optimisation (30%)

Identify the most likely performance bottlenecks and potential performance issues in your database. Discuss the options available to optimise performance. State which option(s) you have chosen and justify your choice, clearly stating the benefits and drawbacks of your choice. Include a step-by-step guide to implement the performance optimisation steps you have chosen.

## 4. Backup / Recovery / Availability Policy (30%)

Discuss the objectives of the backup / recovery policy in your database. Discuss the options available, state which option(s) you have chosen and justify your choice, clearly stating the associated benefits and drawbacks. Describe the configuration steps which have been taken in order to support this policy.

Discuss what measures should be taken to increase the availability of the database and prevent downtime. You do not need to provide code or configuration for availability measures, but can instead outline the steps the business should take to ensure it.

# Scenarios

## Stockbroker

The stockbroker database will be used to back up a trading application. This database includes the following tables

**Companies**: This table holds details of the listed companies in which shares can be bought. The table includes the company name, an internal ID, and the current annual revenue of the company (in dollars). This table will hold upwards of 2000 records.

**Traders:** This table holds details of the traders working for the stockbroker company. This includes a staff ID, name, and date the company was joined

**Portfolios:** This table holds details of the different portfolios managed by the stockbroker company. This table includes the portfolio name and ID

**Prices**: This table holds details of the historical prices of each company in the companies table. Each evening, the closing price of each company in the companies table is written to the historical prices table. This table will occasionally be used to generate reports and provide large batches of data for machine learning algorithms. Columns include companyID, date and value (in dollars)

**Positions:** This table holds records of the total amount of stock held at each company. This table holds the company ID, portfolio ID and current stock held. At the end of each trading day, all traders will submit the trades they have made. These trades need to be processed ASAP to make sure they are lodged in time. Each trade will update the current stock held; with a buy increasing the value and a sell decreasing it.

This database contains two different types of user.

**Traders** have the ability to make trades using company funds. All traders work from the office, so for security reasons, remote working is disallowed.

**Customers** invest in stock and have read-only access to the current positions for any portfolios they have invested in. Customers are allowed to access the database directly to pull the data for use in their own applications.

## Human Resources

The Human Resources (HR) database will be used by an employee management application for a very large U.S. grocery chain (+1,000,000 employees). The database includes the following tables

**Employees**: This table holds details of the employees working for the company. This includes a staff ID, name, date the company was joined, a position, social security number and current salary

**Stores:** This table holds details of the outlets belonging to the company. This table includes a store ID, location, and employee count.

**Payments**: This table holds details of payments to be made to employees. This includes the employee ID, the payment type (salary/bonus), the employee’s bank account number and a status (paid/pending)

**Performance Reviews:** This table holds details of each employee’s quarterly review. The table includes the employee ID, the text of the review (typically 2 – 3 A4 pages worth of text), a review rating, and any associated bonus outcome. This data is held for reporting purposes, and must be removed after 5 years for privacy.

The database contains 2 different types of users.

**HR Staff** who access the database from HQ in the global headquarters in Bentonville, Arkansas. HR staff are responsible for populating the performance reviews table and adding payments to the payments table

**Employees** who access the database from their location of work. All stores are located on a company-wide VPN. Employees have read-only access to their own records, payments and performance reviews.