

Final Project

Description

The final project, which is mandatory, will consist in an actual implementation of a database solution. It will cover most of the subject syllabus, and will consist of the following:

- A relational database
- Its logical design
- A CRUD application that will connect to the relational database
- A NoSQL implementation of at least part of the relational database
- The corresponding documentation, including the logical data model

The students are free to choose whichever tools they consider appropriate. The recommendations are Microsoft SQL Server as relational database, MongoDB as document database, and Neo4j as graph database, since these were the tools used in class. However, the students may choose any other database of the same family as long as all the requirements can be fulfilled. For the CRUD application, they are free to choose whatever development stack, programming language(s), and even application type (standalone/web) they deem appropriate.

The following elements from the subject syllabus will be implemented in the final project:

- An Entity/Relationship Diagram
- A relational database, including its tables, primary and foreign keys, indexes, constraints and referential integrity checks
- Test data
- Transactions
- Stored procedures
- An audit solution involving the use of triggers
- A data warehouse implementation with temporal tables
- Definition of users and privileges
- A CRUD application covering some of the desired functionality
- A NoSQL database representing at least part of the relational database. The students will choose whether they prefer to implement a document database or a graph database.

Final Project Definition

The project will consist in the implementation of a web shop.

Relational database

The database will store products, of which it will be necessary to register their name, description, unit price and existing stock. When a user registers a purchase, an invoice is generated. The database will store the date the invoice is issued, its applicable tax, the total monetary amount, the credit card with which the user paid, and the list of products purchased, including the quantity and the unit price of each product at the time of purchase. Credit card is the only payment method acceptable. A user can register as many credit cards as s/he desires, but it is mandatory to register at least one in order to be registered as a user. For each card, its IBAN code, cardholder name, expiry month and year, CCV, and total monetary amount purchased with that card so far will be stored. For the user, it will be necessary to register his/her name and surname, address, zip code, city, phone number, email, and the total monetary amount paid so far. Only Danish residents with a Danish phone number will be allowed to register in the application.

In order to provide useful information to the application's users, and because of the company's will to store experimental data, product ratings will also be managed. Users will be able to rate any product from 1 to 5, as well as, optionally, to include a comment. In order to avoid calculations, the average rating of each product will be stored with it.

Test data

At the time of delivery, the relational database will contain, at least, the following:

- 20 products
- 15 users from, at least, 3 different cities
- An average of 2 credit cards per user
- An average of 3 purchase invoices per user, so that:
 - At least two users never purchased anything
 - At least 3 users purchased with different credit cards
 - At least 3 credit cards were never used
 - Each purchase includes an average of 3 products
- Correctly calculated total monetary amounts per card and per user

- An average of 3 ratings per product, so that:
 - At least 5 products do not have ratings
 - At least a third of the ratings include comments
 - At least 5 users have never rated

Constraints

- Users and/or cards with a purchased amount higher than zero cannot be deleted
- Products that have been purchased cannot be deleted

Transactions

The following transactions will be implemented:

- Invoice insert. The corresponding product stocks and total monetary amounts should be updated
- Rating insert. The corresponding product's average rating should be updated

The students will decide whether they will implement the transactions in the application or as a stored procedure.

Audit solution

Users and cards will be audited in separate audit tables. The information to include will be:

- An auto-incremental numeric primary key
- All the fields from each table duplicated:
 - One field will include the value before the insert, update, or delete statement was executed
 - The other field will include the value after the statement was executed
- The type of statement that was executed (insert, update, or delete)
- The precise time at which the statement was executed
- The ID and name of the DBMS or operating system user that executed the statement
- The ID and name of the host where the statement was executed

The auditing solution will be implemented by coding the appropriate triggers

Data warehouse

A data warehousing solution will be implemented by the use of temporal tables. It will cover products and ratings, and it will allow to query what was a product's description, price, rating, stock, and user comments at a determined point in the past.

In order to test the proper operation of the data warehouse, at least 5 products should have a history of changes in the aforementioned attributes from the time the data warehouse was implemented, involving an average of 3 changes per product.

Security

Users will be defined only at database level (not at server level). There will be, at least:

- A user with full database admin privileges
- A user with read-only privileges
- A user with restricted reading privileges, which will be unable to see invoice-related information

None of the users will have server level privileges.

CRUD application

A CRUD application (create/read/update/delete) with a minimum front-end will manage purchases from a user point of view. The only functionality to implement will be:

- Log in
- Purchases. A user will be able to choose products and quantities, be informed of the total price, choose a credit card to pay with, and finish the purchase
- Rating. A user will be able to rate any product and, optionally, include a comment

The application will comply with the following:

- All database queries will be prepared in order to avoid SQL injection
- There will be at least one call to a stored procedure
- There will be a multiple search option which will search products by several fields, also implementing approximate character string search (LIKE) by at least one of them

NoSQL database

A NoSQL database will be implemented to cover all products and user ratings. The students will choose which type of NoSQL database to implement:

Document database

A collection or series of collections will be implemented to register users (including their cities), products (including their descriptions and average ratings) and ratings (including numeric ratings and user comments).

Graph database

A graph will be implemented to register users (including their cities), products and user numeric ratings per product.

Final Project Delivery

The final project will be uploaded individually to Wiseflow. It will consist of a report and a series of artifacts.

Final Project Artifacts

- Relational database scripts (one or several), including:
 - Database creation, including tables, keys, indexes, constraints, and referential integrity checks
 - Load of test data
 - Data warehouse implementation
 - Stored procedures
 - Triggers
 - Creation of users and privileges
- The source code of the CRUD application. It can be attached or included as a link to an external public code repository (e.g.: Github, Bitbucket), in which case the content of the repository must not change since the final project's delivery date
- For the NoSQL database chosen:
 - For the document database, either an exported JSON file or series of JSON files
 - For the graph database, the sequence of node and relationship creation statements
- A brief installation procedure that specifies how to organise the code and import the databases in a virgin test environment with full operational capabilities

Final Project Report

The report will feature the following structure:

1. **Cover page**, including:
 - Title
 - Full name and date of birth of all students in the group
 - Group name (in this case: **SD19i**)
 - Date of delivery
2. **Table of contents** (paginated index)
3. **Problem description**
4. **Content**, divided in the following sections:
 - 4.1. **Logical data model**, including:
 - 4.1.1. Entity/Relationship Model
 - 4.1.2. Explanation of non-obvious design decisions
 - 4.1.3. Explanation of non-obvious normalisation and denormalisation decisions

- 4.2. **Tools.** Explanation of choices for RDBMS and version, NoSQL model, database and version, and programming languages and versions
- 4.3. **Physical data model**, including:
 - 4.3.1. Explanation of general and relevant specific choices for data types
 - 4.3.2. Explanation of general and relevant specific choices for primary and foreign keys
 - 4.3.3. Explanation of general and relevant specific choices for indexes
 - 4.3.4. Explanation of general and relevant specific choices for constraints and referential integrity checks
- 4.4. **Transactions.** Explanation of the structure and implementation of transactions, specifying whether the transactions have been implemented in a stored procedure or in the application's programming language, and justifying the decision
- 4.5. **Auditing.** Explanation of the audit structure implemented with special attention to the function of triggers
- 4.6. **Data warehouse.** Explanation of the implementation for the data warehouse
- 4.7. **Security.** Explanation of non-obvious choices for users and privileges
- 4.8. **NoSQL design.** Explanation of the design adopted for the chosen NoSQL database
5. **Conclusions**
6. **Bibliography**, including all sources referenced in the report
7. **Annexes**, if necessary

It will have the following maximum number of pages:

- 2 students 25 pages
- 3 students 30 pages
- 4 students 35 pages

All pages will include their page number, starting with the Introduction. The cover page, table of contents, bibliography, and annexes do not count for the total number of pages.

One page should not contain more than 2400 characters (including spaces and notes). Each figure counts as 800 characters.

**This document is subject to change during the following weeks.
Check constantly for updated versions and, eventually, for the final version.**