

Learning Outcome

This machine project on *Developing a Small Scale Database System* provides a venue for students to achieve the following learning outcomes:

- LO1. Design conceptual, logical and physical data models to represent the information requirements of the organization used in the Field Work activity;
- LO2. Formulate SQL queries to provide access to data (creation, update, manipulation, retrieval) to support the information requirements of the organization;
- LO3. Design and implement a small-scale database application based on the business processes identified from the Field Work activity to support the data access requirements defined in LO2.

Description

Team compositions from the Field Work activity will be retained. The teams will design and implement the features of a prototype database application based on the requirements that they have gathered and documented in their respective Field Work reports. Throughout the software development project, the team will iteratively design data models to represent various levels of the data and information requirements of the organization, including the associated business rules. The team will also develop a database application to provide various users of the organization an end-to-end¹ access to the database using well-formulated SQL query statements and based on the identified business processes.

Milestones and Deliverables

The following deliverables must be submitted on the stated schedule:

A. Revised Requirements Specification

The Requirements Specification document is a revision of the Field Work report and contains the following:

- Forms for data access by various user groups
 - The forms are screenshots of your database application's user interface, designed using your target language. You can use any of the following languages to develop your INTRODB database application - C, Java, C#, Visual Basic, and Python.
 - Data access includes facilities for creating new records, updating and viewing existing records, as well as for viewing the generated detailed and summary reports.
- Conceptual Data Model

The Requirements Specification document is a type-written, single-spaced report printed on short-bond paper. It contains the title page and the following sections:

1. Introduction

In paragraph form, provide an overview of the database application that you will be developing. This is based on your Field Work but should be written from your own point of view. The following serves as your checklist for the items that should be present in this chapter:

- What database application are you developing?
- What features will your application provide? What business processes are these features supporting? What classes of users will use these features?
- What data can your users access from your application?

¹ **End-to-end access** means the data created by one user type should be visible and can be manipulated by another user type. For example, data from orders made by customers should be visible (e.g., via reports) to company managers (administrators). Similarly, data about products created by administrators should be made available to customers.

2. Data Requirements of the <XYZ System>

In this chapter, the conceptual model of the data requirements for your application is presented, along with the description of the important software features (relevant to an online sales/reservation app) to be implemented.

2.1 Conceptual Data Model

Draw the conceptual data model.

Describe each entity type and the attributes it contains (including optional, composite, derived and key attributes; and attribute domain if applicable). Use a table format to summarize the description of each entity type (e.g., *Table 1. Attributes of <Entity Type>*).

Describe the business rules, which are *statements from policies, processes or events in the organization that define or constrain its data.* These include the following (refer to the lecture slides for more details):

- The attribute domain, if applicable, for an entity type
- The entity integrity and key constraints of an entity type
- The relationships between entity types, including primary key - foreign key
- The multiplicity constraints between entity types

Below are some example business rules which may impact your DB design and/or application design.

- Domain Constraint:
 - A password must have at least 6 alphanumeric characters;
 - An employee can be one of three types, namely *salaried*, *hourly*, and *commissioned*.
- Structural Constraint:
 - An employee can belong to only one department.
 - An employee may be assigned to work on one or more projects.
- Business Process:
 - A password must be changed every 30 days (this business rule needs the date when the password was last changed to be stored in the DB);
 - A user cannot reuse his last three passwords (thus, your DB will need a password history table);
 - A user is given three attempts to login to the system (this business rule needs the number of failed attempts and a status field indicating if a user has been locked out to be stored in the DB);
 - A user can have three challenge questions (this business rule needs the challenge questions and the corresponding answers for each user account to be stored in the DB).

Provide relevant examples (entity and relationship instances) to clarify your discussion as necessary.

2.2 Software Features

Describe in detail your own software features that will support the business process and data requirements. All data entry (to add and update records in the database tables) and view screens (to display query results) for all classes of users (customers and administrators) must be presented.

Forms for displaying reports are also discussed in this section. For each report, (a) define its purpose and users; (b) describe the information that will be generated; (c) show a sample format (title, contents, summary); and (d) discuss variants (e.g., *transactions report can be generated daily, weekly, monthly, annually; and filtered by location or product*). There should be at least five (5) reports - 4 are OLTP and 1 is an OLAP. Each OLTP report must have two (2) or more variants. The OLAP report should provide comparative values across two attributes (e.g., date vs location, or product vs date).

Note the following:

- Follow the format stated in the Field Work specs to describe your features and reports, but you must present your own user interface design (screenshots from your running software).
- Your user interface should be neat and should show evidence of careful analysis in designing forms that will allow users access to relevant data and information while ensuring data integrity constraints are observed (through proper DB design and correct choice of UI elements).
- Grading will be based on user interfaces that support the user tasks / business processes.
- Reports will be graded based on their content and relevance to the organization.
- Graphics and colorful interfaces will not merit extra grades.

B. Logical and Physical Data Models and Queries

The second set of deliverables includes the logical data models and the query statements.

1. Derive the relational model.
2. For each data entry and view form in your software that provides access to the database (add, update, retrieval), show the query statement. Explain the query as needed.
3. For each report to be generated by your database application, present and explain the corresponding query statements that will be used. Query statements for variants of a report should also be discussed.

Deliverables:

- Preliminary Conceptual Model
 - Submit the printed copy on **November 10, 2017**, first 15 minutes of class.
- Requirements Specification document
 - Include the revised Conceptual Model (Appendix A).
 - Include the Relational Model (Appendix B).
 - Include the Queries for your different User Entry, Views and Reports (Appendix C).
 - Submit the printed copy on **November 27, 2017**, first 15 minutes of class.
 - Submit a copy of the document in Canvas

Be prepared to present your Conceptual Model and running software prototype (data entry forms and views for a complete business flow) in class.

Penalties for the late submission of deliverables A and/or B.

- For each late submission of required deliverables, 10 points will be deducted from the final project grade. Incomplete submission will also merit the same deductions.
- No late submission will be accepted beyond 3 days after the stated due date.

C. Final Software

Each team will present its database application on **December 6, 2017**. The actual schedule will be provided by your respective INTRODB teachers.

Deliverables. On **December 7, 2017**, ALL groups must submit the following in Canvas:

- The application code and executable file
- The database scripts (or the database itself) needed to re-construct and populate the database
 - Prior to submission, clean your database. Remove all your test data. Then populate each table with at least 5 rows of valid and meaningful data.

***** No submission = No software presentation *****

During the software presentation:

- Each group must bring its own laptop, with the corresponding tools (IDE, compiler, MySQL server) already installed and ready for use.
- The application code / software will be installed and the database will be generated on the actual demo schedule, using the contents from the submitted thumb drive.
- All members must be present during the demo and must be ready to show evidence of involvement and understanding of the database system project.

- The team must prepare and use a test script during the presentation. The test script is a step-by-step guide simulating an actual business process that involves access to data in the database using the features provided by the application.
 - A student may receive a grade of 0 if he fails to show up for the presentation, late for presentation, or cannot answer the questions.
-

Format of the Title Page (for Requirements Specification)

<Name of your Database Application>

A Requirements Specification Document
for the course on
Introduction to Databases
(INTRODB)

Submitted by

<lastname, firstname> of all group members, in alphabetical order

<Teacher's Name>
Teacher

November 28, 2016

Criteria for Grading

The Machine Project comprises 22 points of your final grade for INTRODB. Your Machine Project has 100 points subdivided into the following components:

Milestone 1: Requirements Specification	22pts
Milestone 2: Logical Model and Queries	18pts
Milestone 3: Database Application and Demo	60pts

Individual grading may be applied for members who are not actively contributing to the fulfillment of the project requirements.