

# Project in Database Design I (1DL301)

Fall 2022, Period 2

The project gives you the opportunity to practice a relational database design process from A to Z, from reading the customer specification, which might be redundant, vague or even inconsistent; to conceptual modeling using ER diagrams; to creating, populating and querying the database in an RDBMS; to writing applications accessing and modifying the stored data.

Your group has been asked to help with a project for AltOnline AB, a new Swedish company which, inspired by the tremendous success of Amazon.com, Inc., wants to become a market leader in online sales in Sweden. The project is rather large and your part is to design and implement a database system for its online store.

## Customer specification and requirements

### Structure of the store departments

AltOnline is planning to sell a huge variety of products. To make the navigation as easy and user-friendly as possible, AltOnline wishes to have a hierarchical structure (i.e., a tree) of departments. The following is an example of such a structure:

*Electronics*  
*Electronics / Computers and tablets*  
*Electronics / Computers and tablets / Desktops (leaf<sup>1</sup>)*  
*Electronics / Computers and tablets / Laptops (leaf)*  
*Electronics / Computers and tablets / Tablets (leaf)*  
*Electronics / Computers and tablets / Accessories*  
*Electronics / Computers and tablets / Accessories / For desktops (leaf)*  
*Electronics / Computers and tablets / Accessories / For laptops (leaf)*  
*Electronics / Computers and tablets / Accessories / For tablets (leaf)*  
*Electronics / TV and video*  
*Electronics / TV and video / TVs (leaf)*  
*Electronics / TV and video / Projectors (leaf)*  
*etc.*

Each product in the store will belong to exactly one leaf department.

The managers at AltOnline have specified how the pages are going to look like (rather than what they want to store in the database).

### Department page

A department page will show:

- Logo of the store
- Breadcrumbs (the path to the department, e.g., *Home / Electronics / Computer and tablets*)
- Title of the department

---

<sup>1</sup>Leaf is a department which does not have any child departments.

- Description of the department

If the department is not a leaf, we want to show a list of all child departments, including the following for each of them:

- Title of the child department
- Short description
- Link to the child department's page

If the department is a leaf, we want to show a list of its products, including the following for each product:

- Title of the product
- Short description
- Current retail price (with the value added tax)
- *Add to basket* button (disabled if the product is out of stock)
- Link to the product page

## **Product page**

A product page will show:

- Logo of the store
- Breadcrumbs
- Title of the product
- Description
- Retail price without and with the value added tax
- Value added tax in percents
- Stock quantity (how many items are in stock)
- *Add to basket* button (disabled if the product is out of stock)
- A list of all keyword-related products (products that share at least one of the keywords) showing the following for each of them:
  - Title
  - Short description
  - Current retail price
  - Link to the product page
- User reviews (see below)
- Average rating of the product (see below)

The product might be on sale; in that case we want to show also the price before the discount and the discount as a percentage. The discount is always registered in percents by store managers.

## Homepage

The homepage (the front page) of the store will show

- Welcome text (this text is going to change often, so it must be stored in the database)
- A list of the top level departments, including the following for each of them:
  - Title
  - Short description
  - Link to the department page
- A table of featured products, selected by the store administrators, including the following for each of them:
  - Title
  - Short description
  - Current retail price
  - Link to the products page

*Hint: Create a special department at the root of the hierarchy and use it to store the welcome text as its description.*

## Users

Users who wish to buy or review some products must register by filling a sign-up form with the following fields:

- Swedish personal identity number (personnr.),
- Name
- Address
- Email address
- Phone number
- Password (entered twice)
- Checkbox to give a permission to send the newsletter

Users will use their email address and the password to sign in.

## Reviews

Registered users can rate a product by giving it a certain number of stars – 1 star being a really bad product and 5 stars being a really good product. In addition, the user can accompany the rating with a longer review (plain text).

## Orders

All orders must be stored in the database, including

- Order date
- Status of the order: new, open, dispatched
- Date of the last change of the order
- Payment reference
- Tracking number (if available)

Payments are not going to be handled by AltOnline but by its partner, we only need to store the reference provided by their system, which is a 64 characters long string.

Orders are always going to be sent to the address registered for the customer.

Contents of the basket is stored in the customer's web browser, so we don't need to store it in the database.

*Hint: Think carefully about how / where you are going to store information about the bought products, their quantities and prices. Prices might change and the products might even be removed from the store. (There are several ways how to do this.)*

## Tasks

This is a group project which means that each member has to contribute to the solution. It is recommended that you work together on all tasks. You are responsible to take contact with other group members in your group and organize your work on the project. Do not wait with the project work until the first lab, start as soon as possible.

1. Make an ER model of the database for AltOnline. You can draw the diagram by hand and submit a legible scan/picture or use any software you are comfortable with. A simple online tool you can use is [ERDPlus.com](https://erdplus.com)<sup>2</sup>.

This task includes finding out what you need to store in the database. Make your own assumptions (and write them down) if necessary. You can also ask your assistant to play the role of a manager at AltOnline and answer your questions about things which are not clear.

2. Normalise your ER model up to the 3rd Normal Form (3NF). Make sure you show the state of each normal form up to the 3NF (i.e. show 1NF, 2NF). For each relation indicate the the primary and foreign keys. If the derived tables from your ER-model are already normalized (in 3NF), we still want to see the three normal forms (keys and functional dependencies).
3. Write the SQL commands to create the tables. (Do not use any automatic tools for this, the purpose of the task is to learn / practice how to do it manually.)

Remember to specify the primary key for each table as well as all foreign keys.

---

<sup>2</sup>Logged in users can save their work on the server, but we strongly recommend to back up your work often by exporting the diagram (and saving it on your own computer). You can use ERDPlus as an unregistered user as well, but if you accidentally click on a link in the top of the page or close the window, there is no confirmation dialog and you will loose your work.

4. Create at least 2 top-level departments, each having at least 3 child departments, and at least 10 products in the store. Make some of the products featured on the homepage. Create at least 2 users and add their reviews for the same product. Create one order for one of the users.

Use SQL commands. (Tip: You don't need to write the real text for descriptions and reviews – use e.g., [lipsum.com.](https://lipsum.com/))

5. Write SQL queries to get:

- Welcome text for the homepage
- List of the top level departments with fields needed for the homepage
- List of the featured products with fields needed for the homepage
- Given a product, list all keyword-related products
- Given an department, list of all its products (title, short description, current retail price) with their average rating
- List of all products on sale sorted by the discount percentage (starting with the biggest discount)

Optional challenges:

- List 10 best-selling products (in last 30 days)
  - Write an SQL query as complex as possible (but still doing something useful).
6. Analyze and optimize the performance of these queries<sup>3</sup>. Identify and choose at least one SQL query which can run faster by adding an index. Write SQL statement(s) for adding this index (these indices). In your report, include the output of the EXPLAIN command before and after adding the index (indices) as well.
  7. Create a Java or a Python program which connects to the database, asks the user for a department ID (i.e., the value of the primary key) and lists all its products (outputting the ID, the title and the retail price after the discount) if the given department is a leaf department, otherwise lists all its child departments (outputting the ID and the title).

Create another program which asks for a product ID, shows the current discount and allows the user to change it.

(You can use another programming language as well, but remember that the assistants might not be able to help you if you have problems related to the programming language or a MySQL connector.)

## Labs

Labs will be used to help and guide you but you are expected to work on the project outside of the classroom as well. The presence at labs is not mandatory but recommended. Before each lab we will send out a link to Doodle where you can register for one of the dates.

- Lab 1: Conceptual modelling (ER)
- Lab 2: Normalization. Also will be dedicated to practicing SQL queries
- Lab 3: Creating tables, populating the database, SQL queries, indices
- Lab 4: Creating the programs

---

<sup>3</sup>Since your database will be populated with only a few entries, the execution time will not be a reliable metric of performance. Instead, monitor the number of rows that the database examines as a measure of your queries performance.

## Database server

We have already created an empty database for each group to use on our university MySQL server. Instructions on how to connect to the database using MySQL Workbench can be found [on the course page on Studium](#).

## Software

If you are not going to work on university lab computers, you will need to install following software on your computers:

- [MySQL Workbench](#) or another MySQL client
- If you are going to use Java:  
[Java Development Kit](#), [MySQL Connector/J](#) (JDBC driver for MySQL) and possibly an IDE, e.g., NetBeans or Eclipse
- If you are going to use Python:  
[Anaconda \(Python 3.x version\)](#) or another Python 3.x distribution

## Deliverables and submission instructions

The final report must include the following parts:

- A. Title page including the name of the group, and the names and email addresses of all its members
- B. ER diagram
- C. Normalization
- D. SQL code to generate the tables and populate the tables (Task 3 and Task 4)
- E. The diagram generated by MySQL Workbench's Reverse Engineer
- F. SQL queries as described in Task 5
- G. SQL to create indices (as described in Task 6)
- H. Source code of the programs (as specified in Task 7)

**The preliminary and final version of the report must be submitted via the Studium in the PDF format after each of the milestones:**

- Milestone 1: The preliminary report including parts A – B
- Milestone 2: The preliminary report including parts A – C
- Milestone 3: The preliminary report including parts A – G
- Milestone 4: Final version

Deadlines for each of these milestones will be published on Studium.

After the submission of each Milestone you will receive feedback from the assistant. The assistant will also check your database. More details will be published later.

## **Optional additional challenges (if you are up to them)**

If you wish you can make the project more challenging, you can add one or several extra features to it:

- Add support for products belonging to several departments (but one of them must be the "main" department).
- Add support for order history (how and when the order status changed).
- Extra challenging: AltOnline changed its mind and wants the non-leaf department pages to show also all the products which belong to any of descendant departments (i. e. children, grand children, etc.). What do you need to change in your database to be able to find these products in an efficient way?
- Feel free to add any feature you want.