Project 2021

The project consists of one large assignment, which is described below. **You are expected to work on the project assignment throughout the course. Each time a module is completed, the tasks performed during that module can be completed also on the project.**

1. Project Description, the *Soundgood Music School*

The purpose is to facilitate information handling and business transactions for the Soundgood music school company, by developing a database which handles all the school's data and also an application that can handle some of the transactions.

1.1 Business Overview

Soundgood sells music lessons to students who want to learn to play an instrument. When someone wants to attend the school, they apply by submitting contact details, which instrument they want to learn, and their present skill. If there is room, the student is offered a place, which can be accepted or rejected. There is no concept like 'course' or sets of lessons. Instead, students continue to take lessons as long as they wish. Students pay per lesson and instructors are payed per given lesson.

1.2 Detailed Descriptions

**Lesson**

There are individual lessons and group lessons. A group lesson has a specified number of places (which may vary), and is not given unless a minimum number of students enroll. A lesson is given for a particular instrument and a particular level. There are three levels, beginner, intermediate and advanced. Besides lessons for a particular instrument, there are also ensembles, where students playing different instruments participate at the same lesson. Ensembles have a specific target genre (e.g., punk rock, gospel band), and there is a maximum and minimum number of students also for ensembles.

Group lessons and ensembles are given at scheduled time slots. Individual lessons do not have a fixed schedule, but are rather to be seen as appointments, like for example an appointment with a dentist. Administrative staff must be able to make bookings, it must therefore be possible to understand which instructor is available when, and for which instruments.

There is no concept like 'course' or sets of lessons, a student who has been offered a place, and accepted, continue to take lessons as long as desired, and can either book one lesson at the time or book many lessons during a longer time period.

**Student**

A student can take any number of lessons, for any number of instruments. Person number, name, age, address, contact details and contact details for parents must be stored for each student. It must also be possible to see which students are siblings, since there is a discount for siblings.

**Instructor**

An instructor can be assigned to group lessons and ensembles, and can also be available to give individual lessons during specified time periods. An instructor can teach a specified set of instruments, and may also be able to teach ensembles. Person number, name, age, address and contact details must be stored for each instructor.

**Student Enrollment**

When someone wants to attend the school, they apply by submitting personal data (like name and contact details), which instrument (or ensemble) they apply for, and how skilled they are at playing that instrument. If there is a place available in the desired kind of lesson, the student is offered to participate.

**Student Payment**

Students are charged monthly for all lessons taken during the previous month. Currently, there is one price for beginner and intermediate levels, and another price for the advanced level. Also, there are different prices for individual and group lessons. There is also a discount for siblings, if two or more siblings have taken lessons during the same month, they all get a certain percentage discount. Soundgood wants to have a high level of flexibility to change not just prices, but also pricing scheme. They might, for example, not always have the same price for beginner and intermediate lessons.

**Instructor Payment**

There are no instructors with fixed monthly salaries, instead they are payed monthly for all lessons given during the previous month. Instructor payments depend on the same things as student fees (see above), namely level of lesson and whether a given lesson was a group or individual lesson. Instructor payments are not affected by sibling discounts.

**Renting Instruments**

Soundgood offers students the ability to rent instruments to be delivered at their home. There is a wide selection of instruments, wind, string etc., supporting different brands and in different quantities in stock at the soundgood music school. Each student can rent up to two specific instruments at any given period, the renting happens with a lease up to 12 month period. Students can list and search current instruments and rent them if they don't exceed their two-instrument quota. Instruments are rented per month. The fee is payed the same way lessons are payed, each month students are charged for the instruments that where rented the previous month.

1.3 Requirements on the Soundgood Music School Application

The database must store all data described above, in sections 1.1 and 1.2, but no other data. There will also be an application providing a user interface which can be used by administrative staff to manage student enrollments, instrument rentals, bookings and payments. In addition, the database will also be used to retrieve reports and statistics of all possible kinds, but a user interface is not required for that purpose. It will instead be done by manually querying the database.

The database will not be used for any financial purpose like bookkeeping, taxes or bank contacts. What is written above regarding student fees and instructor payments is only about calculating what sum shall be payed to or by who, and sending that information to Soundgood's financial system.

2. Grading

There are four tasks, which are described below. Each task gives max ten points, and is divided into a mandatory part (5p) and a higher grade part (5p). To pass the project all mandatory tasks must be passed. The higher grade tasks are optional, and contribute to the final grade as specified in the [Course Layout](https://canvas.kth.se/courses/27118/pages/course-layout). No partial score is given, either a part of a task is accepted and gives five points, or it is not accepted and does not give any points. A task is passed when both oral and written reports are passed. Reporting is explained in detail below.

3. Tasks

You are encouraged to collaborate and discuss with as many other students as you wish when doing the project, group discussions always give a better result than individual work. **You are however allowed to create the actual solution in groups of max two students. That is, your solution is allowed to be identical to the solution of at most one other student. You are also allowed to work alone, but that is not recommended. The written report is individual, and may not be written together with any other student.**

3.1. Task 1, Conceptual Model

**How To Prepare**

Before solving this task you have to understand the [conceptual model lecture](https://canvas.kth.se/courses/27118/pages/conceptual-model) (only recorded, not given live). Note that the conceptual model lecture page contains several videos, which together lasts about three hours.

**When To Solve**

Your are recommended to start working on this project task as soon as possible at course start, but remember to first watch the recorded lecture on the [conceptual model](https://canvas.kth.se/courses/27118/pages/conceptual-model). The [seminar](https://canvas.kth.se/courses/27118/pages/seminars) where this task is discussed is held Nov 12.

**Intended Learning Outcomes**

* Model needs for information based on an organizational description and convert the model to a functioning database.

**Mandatory Part**

Create a conceptual model for the Soundgood music school database. The conceptual model must cover the entire description of the Soundgood music school company in section one above. The diagram must be made either in UML or in one of all possible crow foot notations (for example IE notation). Below follows guidelines for what shall be written in the report.

* In the *Method* chapter of your report, explain the procedure you followed to create the conceptual model. You shall mention all steps that are covered in the videos on conceptual modeling. If you did not perform a particular step, explain why the result was better (or at least not worse) without that step. *Do not explain the result of each step*, only explain the steps themselves.
* In the *Result* chapter of your report, show and briefly explain your conceptual model.
* In the *Discussion* chapter of your report, evaluate your conceptual model. Suggested assessment criteria are listed below, you do not have to cover them all. These same criteria will also be used to grade your project report.
  + Are naming conventions followed? Are all names sufficiently explaining?
  + Is the notation (UML or crow foot) correctly followed?
  + Is there a reasonable number of entities? Is some important entity missing?
  + Are there attributes for all data that shall be stored? Is cardinality specified for all attributes?
  + Are all relevant relations specified? Do all relations have cardinality at both ends and name at least at one end?
  + Are all business rules and constraints that are not visible in the diagram explained in plain text?

3.2. Task 2, Logical and Physical Model

**How To Prepare**

Before solving this task you have to understand the lectures on [normalisation](https://canvas.kth.se/courses/27118/pages/normalisation" \o "Normalisation) (given live and recorded) and on [logical and physical models](https://canvas.kth.se/courses/27118/pages/logical-and-physical-models) (only recorded, not given live).

**When To Solve**

Your are recommended to start working on this project task after having solved task 1, attended the lecture on [normalisation](https://canvas.kth.se/courses/27118/pages/normalisation" \o "Normalisation) which is given Nov 15, and watched the recorded lecture on [logical and physical models](https://canvas.kth.se/courses/27118/pages/logical-and-physical-models). The [seminar](https://canvas.kth.se/courses/27118/pages/seminars) where this task is discussed is held Nov 26.

**Intended Learning Outcomes**

* Model needs for information based on an organizational description and convert the model to a functioning database.
* Use relational databases and query languages.

**Mandatory Part**

Translate the conceptual model from task one into the same kind of model that was created in the lecture on logical and physical models, that is a logical model with enough physical aspects to enable creating a database. You are allowed to change the conceptual model if you discover flaws, but only change if it is necessary. The model that is created must cover the entire description of the Soundgood music school company in section one above. The diagram must be made in a crow foot notation (for example IE notation). Also create a database based on the model, the database must be exactly as described in the model. If you discover flaws when creating the database, and want to change it, also the model must be changed. You are advised to also look at tasks three and four already now, to get an understanding of how your database will be used. Below follows guidelines for what shall be written in the report.

* In the *Method* chapter of your report, explain the procedure you followed to create the model. You do not have to mention all eleven steps covered in the videos on on logical and physical models, but it must be clear how you proceeded. *Do not explain the result of each step of your procedure*, only explain the steps themselves.
* In the *Result* chapter of your report, show and briefly explain your model. Also include a link to a git repository where you have stored SQL scripts that create your database. There shall be one script that creates the database, and another script that inserts data. You may explain the SQL in the scripts if you wish, and think it clarifies the model, but it is not required to write such an explanation.
* In the *Discussion* chapter of your report, evaluate your model. Suggested assessment criteria are listed below, you do not have to cover them all. These same criteria will also be used to grade your project report.
  + Are naming conventions followed? Are all names sufficiently explaining?
  + Is the crow foot notation correctly followed?
  + Is the model in 3NF? If not, is there a good reason why not?
  + Are all tables relevant? Is some table missing?
  + Are there columns for all data that shall be stored? Are all relevant column constraints and foreign key constraints specified? Can all column types be motivated?
  + Can the choice of primary keys be motivated?
  + Are all relevant relations correctly specified?
  + Is it possible to perform all tasks listed in section one, Project Description, above?
  + Are all business rules and constraints that are not visible in the diagram explained in plain text?

3. 3. Task 3, SQL

**How To Prepare**

Before solving this task you have to understand the lecture on [SQL](https://canvas.kth.se/courses/27118/pages/sql-the-structured-query-language) (given live and recorded).

**When To Solve**

Your are recommended to start working on this project task after having solved task 2, and attended the lecture on [SQL](https://canvas.kth.se/courses/27118/pages/sql-the-structured-query-language), which is given Nov 11. The [seminar](https://canvas.kth.se/courses/27118/pages/seminars" \o "Seminars)where this task is discussed is held Dec 7.

**Intended Learning Outcomes**

* Use relational databases and query languages.
* Describe and explain basic concepts, principles and theories in the area of data/databases/data storage and in information administration and database design

**Mandatory Part**

The goal here is to create OLAP, Online Analytical Processing, queries and views. Such queries serve to analyze the business and to create reports. You also have to make sure the database contains sufficient data to check that all queries work as intended. If needed, update the script that inserts data, created in task two. You are also allowed to change the database you created in task two if needed. The queries that shall be created are explained below, only OLAP queries will be created here. The OLTP (Online Transaction Processing) queries used by the business itself, which in the case of Soundgood is to rent out instruments, register taken and given lessons, etc, will be created in task 4, together with the program executing them.

**The following queries will be executed manually, to generate analysis reports.**

* Show the number of lessons given per month during a specified year. It shall be possible to retrieve the total number of lessons per month (just one number per month) and the specific number of individual lessons, group lessons and ensembles (three numbers per month). This query is expected to be performed a few times per week.
* The same as above, but retrieve the average number of lessons per month during the entire year, instead of the total for each month.
* List all instructors who has given more than a specific number of lessons during the current month. Sum all lessons, independent of type, and sort the result by the number of given lessons. This query will be used to find instructors risking to work too much, and will be executed daily.

**The following queries will be performed programmatically, and the results will be displayed on Soundgood's web page. You only have to create the queries, not the web page.**

* List all ensembles held during the next week, sorted by music genre and weekday. For each ensemble tell whether it's full booked, has 1-2 seats left or has more seats left. Hint: you might want to use a CASE statement in your query to produce the desired output.

**Below is specified what shall be written in the report.**

* In the Method chapter of your report, mention which DBMS you use, which tool is used to develop SQL queries, and how you have verified that your SQL queries work as intended. You shall just tell which method you used for testing the queries, not explain each test in detail.
* In the Result chapter of your report, include a link to a git repository where you have stored a script with all queries. *Also explain each query and show that all queries work as intended by****including the output of each query***. The git repository must also contain the scripts that create the database and insert data. It shall be possible to test your solution by executing first the script that creates the database, then the script that inserts data, and finally any of the queries created in this task.
* In the Discussion chapter of your report, for each query motivate why you did or did not create a view or a materialized view.

3.4. Task 4, Programmatic Access

**How To Prepare**

Before solving this task you have to understand the lectures on [transactions](https://canvas.kth.se/courses/27118/pages/transactions) (given live and recorded) and on [Database Applications](https://canvas.kth.se/courses/27118/pages/database-applications) (only recorded, not given live).

**When To Solve**

Your are recommended to start working on this project task after having solved task 3, attended the lecture on [transactions](https://canvas.kth.se/courses/27118/pages/transactions) which is given Nov 29, and watched the recorded lecture on [Database Applications](https://canvas.kth.se/courses/27118/pages/database-applications). The [seminar](https://canvas.kth.se/courses/27118/pages/seminars) where this task is discussed is held Dec 17.

**Intended Learning Outcomes**

* Describe how a program can access a database and write such a program.

**Mandatory Part**

The assignment is to develop part of Soundgood's web site. You are however only required to develop a very limited set of functionalities, namely what is used when instruments are rented. Also, since focus here is on database access, you are not required to develop the web interface, but a command line user interface is sufficient. You are allowed to reuse as much code as you wish from all classes in the view layer of the JDBC bank example at the page [Database Applications](https://canvas.kth.se/courses/27118/pages/database-applications), but all code included in your program is your responsibility. You are not allowed to blame any deficiency in your application on the bank program. Your program shall be stored in a public git repository, for example on GitHub. *The program is required to handle ACID transactions properly*, and shall have the functionality listed below.  You also have to make sure the database contains sufficient data to check that all queries work as intended. If needed, update the script that inserts data, created in task two. You are also allowed to change the database you created in task two if needed.

* **List instruments** It shall be possible to list all instruments of a certain kind (guitar, saxophone, etc) that are available to rent. Instruments which are already rented shall not be included in the listing. The listing shall show brand and price for each listed instrument.
* **Rent instrument** It shall be possible to specify which student is renting the instrument, and which instrument is being rented. Since different instruments of the same kind might have different prices, it must be possible to specify exactly which particular instrument to rent, not just any instrument of the desired kind. Remember that a student is not allowed to rent more than two instruments at the same time, your program must check that this limit is not exceeded.
* **Terminate rental** It shall be possible to terminate an ongoing rental. You are free to decide how the user specifies which rental to terminate. You are not allowed to delete all information about a terminated rental from the database. Instead, the database must still contain all information about the rental, but also show that the rental has been terminated.

Below follows guidelines for what shall be written in the report.

* In the *Method* chapter of your report, explain how you proceeded and reasoned  
  when writing the program. *Do not explain the result of each step you took*, only explain the steps themselves.
* In the *Result* chapter of your report, briefly explain the program and in particular explain ACID transaction handling. Include links to your git repository, and make sure the repository is public. **Also include a**  
  **printout of a sample run**. The git repository must also contain the scripts that create the database and insert data. It shall be possible to test your solution by executing first the script that creates the database, then the script that inserts data, and finally execute your program.
* In the *Discussion* chapter of your report, evaluate your program. Suggested assessment criteria are listed below, you do not have to cover them all. These same criteria will also be used to grade your project report.
  + Are naming conventions followed? Are all names sufficiently explaining?
  + Is transaction management correct? Are there ACID transactions, which are committed or rolled back correctly?
  + Is the program working as expected and does it meet all functional requirements listed in this task?

4. How to Get Help

Questions about the project can be asked these ways:

* At tutorials, tutorials exist only to provide help with project tasks. You are welcome to ask about anything related to project work at tutorials.
* At seminars. The purpose of seminars is to give feedback to solutions, but you can also ask about things that are unclear.
* In [Piazza (Länkar till en externa sida.)](http://piazza.com/kth.se/fall2020/iv1351ht201/home). Do not send direct emails to teachers, instead post questions in Piazza since then all students can be helped by the answers. Mind that posts can be anonymous to peer students in Piazza.

5. How to Report

Each solved task shall be reported both orally and in writing. A task is passed only when both reports have been accepted. You are encouraged to collaborate and discuss with as many other students as you wish when doing the project, group discussions always give a better result than individual work. **You are however allowed to create the actual solution in groups of max two students. That is, your solution is allowed to be identical to the solution of at most one other student. You are also allowed to work alone, but that is not recommended. The written report is individual, and may not be written together with any other student.**