Storing and Managing Data Module 06-32245 Semester project

Felipe Orihuela-Espina

Created: 9th September 2022

Last modified: 18th September 2022

Version: v0.1

Document version

Date	Doc. Version	Author	Comments
9-Sep-2022	v0.1	FOE	First draft completed

Contents

1 About the course research project

This document provides a description of the semester project for the course Storing and Managing Data (Module 06-32245) for the term Autum 2022. The project is a small *hands-on* task that allows the student to implement some of the methods learnt during the course. The **deadline for submission** of the course project is 16:00h of **1st of December 2022**.

> ☞ Please, read these instructions carefully, and **follow the guides strictly**. Failing to do so may result in your project not being evaluated with the consequent loss in marking.

The marks obtained for this project are counted towards the final marking of the course. The semester project will be assessed on a scale of 1 to 10, and the project mark will count 25% of the total course marking. Course projects are developed outside lecture times.

2 Project description

In this project you will be building a small data vault with minimal features to store and mine medical imaging data. Some exemplary data will be provided to you as further explained in Sect. 6, but you need to take care of implementing its functionality.

A **data vault** [4] is all at once, a database architecture (inc. Relational and NoSQL), model (e.g. guaranteeing performance and scalability), methodology (e.g. using Agile) and implementation (e.g. pattern based) for developing data warehouses. A **data warehouse** *is* a database; but a database specifically designed to make data analytics fast and easy. Further details about data vaults are given in Sect. 4.

The project is loosely inspired in NeuroVault (`https://neurovault.org/`) whose goal is to store, share, visualize, and decode neuroimages [2]. However note, that NeuroVault does not follow **data vault 2.0** [4] and has a different implementation.

> ☞ This is a **research** project. You will not be given the solution nor a recipe to it. In fact, there is not a single optimal solution. Each solution has pros- and cons- and you will be expected to justify your decisions in the report.

3 Project submission

3.1 What to submit?

Upon conclusion of the project, prepare a folder named `SMD2022_Project` with the following subfolders:

1. **report**: A subfolder called `report` containing the report file. This should contain the `.pdf` as well as the original document. If it was developed in LaTeX, then the `.tex`, `.bib` and all figures, must be included here. The lecturer must be able to compile the `.pdf` in Windows using a regular Miktek distribution. It is was developed in Word, then the `.doc`/`.docx` outght to be included here. If you use OpenOffice or other alternative, you need to convert it to MS office format (as well as, of course, export it to `.pdf`). Further instructions to prepare the report are given below.
2. **code**: A subfolder called `code` with ***all*** source code ready for execution. Step-by-step instructions on how to execute the code should be provided as part of the technical documentation (see folder documentation below). Code must be developed in SQL and Python.
   - SQL code should be provided in `.sql` files and have to be executable using the `\i` command in PostgreSQL. Your database should be called `smdvault`. The code to create the database and its associated tables, types, etc should be placed in a file named `dataVault.sql`.
   - Python code must be richly documented. In-code documentation will be generated using documentation tool Sphynx[1].

   > ☞ The code must be provided with *all* required libraries, and must run smoothly off-the-shell on a Windows 11 machine by the indicated deadline. Your code must NOT require any manual changes by the assesor. It is your responsibility to ensure this is the case. In this sense, *it is strongly advisable to arrange for a presubmission and test session with the lecturer or one of the PGTA*. This is not compulsory, but again, strongly advisable.

3. **results**: A subfolder called `results` with ***all*** result files. All intermediate files generated by the code, as well as all final results files should be in this folder. You may include here tables, queries, databases (in `.csv` or `.xlsx` format), figures (in `.gif`, `.tif`, `.png` or `.jpg` format), as well as additional documents (in `.pdf` or `.docx` format).
4. **documentation**: A subfolder called `doc` with ***all*** documentation files. Include here all additional technical documentation. Note that this is different from the report. At the very least, this should contain:
   - A `.txt` file encoded in UTF-8 named `README.txt` with step-by-step instructions on how to install and execute your code.
   - The compiled output of the in-code documentation in Sphyinx, e.g. `.html`, `.css`, etc
   - A `ERmodel.pdf` showing the full ER model of your data vault enterprise layer.
   - One or more files showing Unified Modeling Language (UML) documentation of your staging and information mart layers.

   All these subfolders must be present even if empty.

   All **material** (the report and supplementary files) should be zipped into a **Windows 11 friendly** `.zip` file and uploaded to Canvas by the above deadline.

> ☞ Make sure you pack your submission exactly as indicated here. Respect filenames and folders names as well as folder structure and organization as requested in these instructions. If you are using Mac OS X, Linux or any other operating system beware of different file encodings. It is NOT the responsibility of the lecturers nor the PGTAs to repackage your submission.

[1] `https://www.sphinx-doc.org/en/master/`

Submission of videos are strictly prohibited and cannot be accepted of proof that your code is running since we cannot guarantee its authenticity. Besides, it is circumstantial if your code runs in your machine or that of your friends. It ought to run in our machines! (which is yet another reason why a presubmission check is so critical).

3.2 About the report

The **report** is the single most important part of the submission; far more important than the code itself. The emphasis of the report ought to be in showing understanding of the data vault model. The exemplary medical imaging domain must be clearly detached from the abstract concepts of the data vault model. For instance, it is utterly circumstancial that you have a hub for this or that medical imaging concept, the critical thing is that you understand what a hub is in a data vault and why and where you opt to create one.

The report must be in `.pdf` format and be named:

SURNAME_NAME_STUDENTID_Report.pdf

where SURNAME, NAME and STUDENTID must be substituted for your own family name, first name and student ID number. The file will be placed in the `report` subfolder, and be accompanied by the originals `.tex/.bib` files or `doc/docx`.

The report will be *styled* to look like a short conference paper. The maximum length of the report must be **4 pgs maximum** excluding the references in the IEEE conference paper style; a template is provided and font size and margins ought to be respected. The report must be written in English. The content of the report must follow this structure:

1. Title set to: ''Implementation and verification of a data vault"
2. Subtitle set to: ''Storing and Managing Data (Module 06-32245), Autumn 2022 "
3. Author name and surnames, student ID and institutional e-mail, and course name and period
4. Abstract (Up to 200 words)
5. Introduction
6. State of the art of data on vaults
7. Methods
8. Results.
9. Discussion
10. Conclusions
11. Acknowledgements (Only if required. It may be absent)
12. References.

All the sections above (except acknowledgements if not required) must be included and exactly in this order.

It is suggetsed that you make the abstract structured. An structured abstract is one organized in the following elements: *Background* (2-3 sentences recommended) describing the problem, *Aim* (1 sentence) either in the form of goal, or research question or sumetimes even a hypothesis, *Methods* (3-4 sentences) succintly describing the rationale of the solution, *Results* (2-3 sentences; 1 per finding) summarizing the most important findings, *Conclusions* (1-2 sentences) contextualizing the results and stating the impact of the work carried out. Note that each of these elements are later discussed in detail in the body of the report.

The results must be fully replicable and reproducible from the contents of the technical report. It is therefore suggested that the results section is organised by either verification and validation tests carried out or by features added to the vault.Describe every test carried out in detail.

Think of the report *as if* you were writing a conference paper. Analogous to what happen in conference papers, the research must be reproducible from the main paper alone, in this case, your report, with supplementary material being the complement that allows the reviewer (in this case the lecturer) to verify what it is stated in the paper i.e. the report.

It is strongly recommended that you use LaTeX. A **template** has been made available for you in the Canvas platform that already contains the correct formatting.

☞ Plagiarism is a serious offence and will not be tolerated. Your report will be scanned with TurnItIn for potential plagiarism. Always reference all your sources adequately. Be particularly careful with copying and pasteing from the web. In as much as possible, rephrase ideas with your own words and give credit to where credit is due.

3.3 Project requirements

You will be assessed in Windows 11 64-bit Intel-based machines using PostgreSQL 14.5. SQL code will be tested using the `\i` command. Assume that a user `smd` with password `smd2022` already exists and that has the right permissions to create, populate and query the database. You do NOT need to include the code for creating the user or grant the permissions in your code, but only for creating the database, connecting to it and to create the tables and attributes within it.

- Follow the course project submission rules explained in this document strictly.
- Languages: SQL and Python
- Data: Provided by the lecturer. Links to the datasets are available in the Canvas platform.
- Naming convention and rules of capitalization: Both across SQL names for tables and attributes names and Python for classes, methods and variables, the adopted naming convention will be a varian of the `CamelCase`, where by the first letter of the composing words can be either lower or capital case, but all other intermediate letters will be lower case; see some examples below. Snake style will strictly NOT be allowed.
  Examples:
  - CamelCase - Valid
  - camelcase - Valid
  - Camelcase - Valid
  - camelCase - Valid
  - CamElCasE - Not valid
  - Camel_Case - Not valid
  - Camel-Case - Not valid
  - camel_case - Not valid
  - camel Case - Not valid

You can make use of any software tool or library available. But if you do so, make sure that:

1. You submit the software Windows installation files, and instructions to install (in the `README.txt` file)
2. During reporting, make sure that you provide the lecturer with a step-by-step foolproof guidance on how to replicate your results. Even if the lecturers and PGTAs are familiar with the softwares and architectures we are NOT fortune tellers or mind-readers! We cannot guess where or how have you been clicking around! Nor we do know of the myriad of intermediate undocumented tests that your carried out.

3.4 Project assessment

The marking goes from 0 to 10 but 12 points are on offer. If you go over 10, your final marking will be cap at 10. The final marks that you will receive will depend on the quality and number of features that you incorporate to your vault. Quality matters! That is you might get partial markings i.e. just because you have a staging layer does not mean you will get all 6 marks.

- Minimal features (Enterprise layer) (6 pts)
  - An Enterprise layer for data storing and management. The vault is capable of holding information about fNIRS neuroimaging data coming from different studies. Each study may involve any number of participants, sessions, treatments, groups, etc Further description of the data are given below. This is the barebones of the data vault and forms the content of the `dataVault.sql` file.
- Intermediate features (Staging and information mart layers) (4 pts)
  - A Staging layer in SQL and/or python capable of reading the data provided and transform it to populate the data vault. A full three substages extract, transform and load (ETL) process ought to be implemented. The code corresponding to the staging layer ought to be in files `staging.sql` and/or `staging.py`.
  - An information mart layer for basic data analytics. The vault will be capable of rudimentary database querying, aggregation and visualization.
  - A browser based intuititive GUI for querying (i.e. for the information mart layer only).
- Advanced features (2 pts) For instance, but not limited to:
  - Advanced aspects for staging and information mart layers, as well as business rules application [4].

- Full implementation of the Data warehouse layer with the data vault model including extensions such as the Metrics Vault, the Business Vault or the Operational Vault [4].
- *Thorough* technical documentation. Note the emphasis in thorough; low quality documentation won't be given additional marks. Technical documentation is NOT a user guide, but adequate documentation of classes, methods and parameters formed of both in-code (SQL or Sphynx based) as well as off-line (ER, UML) documentation.
- Some niceties; User profiling and authentication, extensive meta-data, historical updates and versioning, protection against SQL injection, advanced data analytics and visualization, etc.

It is very unlikely that you will be able to implement all of the features listed above, and further do so correctly in such a short time as it is a term. Consider it an open ended project where the further you get, the higher your score. However, please note that the project will be better marked if those features of your vault are thoroughly tested and verified (higher quality), over the quantity of features. You will have to prioritize and almost certainly compromise. Make sure that you clearly justify every decision in your report.

3.5 Submission checklist

A simple checklist before submitting:

- `[ ]` My files and folders are named strictly as requested.
- `[ ]` If first language is not English, make sure no trace of your original language remains e.g. in the report, code, comments, figures, etc.
- `[ ]` Do NOT submit unrequired material. Remove all unrequired material from the folders before submitting.
- `[ ]` If using an operating system other than Windows 11, make sure that the files can be opened in a Windows 11 machine.
- `[ ]` The `.pdf` of the report as well as the original document is in the `report` subfolder.
- `[ ]` The `dataVault.sql` with the code to generate the data vault database is in the `code` subfolder.
- `[ ]` Your data vault database is called `smdvault`.
- `[ ]` The `staging.sql` and/or `staging.py` with the code for the staging layer is in the `code` subfolder.
- `[ ]` The `README.txt` with the instructions to install and run the code is in the `doc` subfolder.
- `[ ]` The `ERmodel.pdf` with the ER model of your data vault enterprise layer in the `doc` subfolder.
- `[ ]` I have NOT included videos in the submission folders.

4 Data vaults

The original Linstedt's book on data vaults [4] is available in Springer site using your institutional account:

`https://www.sciencedirect.com/book/9780128025109/building-a-scalable-data-warehouse-with-data-vault-2-0?dl=book`

A **data vault** [4] is all at once, a database architecture (inc. Relational and NoSQL), model (e.g. guaranteeing performance and scalability), methodology (e.g. using Agile) and implementation (e.g. pattern based) for developing data warehouses. A **data warehouse** *is* a database; but a database specifically designed to make data analytics fast and easy. This is in contrast to classical databases which have traditionally emphasize transactions. But don't get fooled! They *are* relational databases afterall. The difference with the traditional databases as we shall see is how the tables are organized (see below).

Data warehouses integrate transaction data, often, in a so called online analytical processing (OLAP) databases. Data warehouses exist as layers on top of other databases; usually online transaction processing (OLTP) databases. Specifically, the **data vault** architecture is based on 3 layers; staging, enterprise and information delivery layers.

- **Staging layer**: The staging layer is the input layer. It is the layer in charge of receiving the data and transforming it to be written in the data base in the enterprise layer. The layer is oriented to implement the extract, transform, load (ETL) process. Extract means reading available input e.g. in the form of files. Transform refers to decomposing the input data into nuggets ready for inserting them in the database. For instance, one single input file may contain data which in the database is scattered through several tables. Hence, the transform stage will take care of splitting each piece of data and reorganise it to a format compatible with the database architecture. Finally, the load stage takes the transformed data and inserts it into the data vault. During the ETL process the data is tagged with the input timestamp as well as the business key identifying who is inserting the data.

- **Enterprise layer** a.k.a. **Data warehouse layer**: The enterprise layer is the one responsible for storing all the data. This is the relational database underpinning the data vault warehouse. Contrary to regular relational databases, in the data vault model, data are never deleted or overwritten. Instead, as new data arrives, new timestampted entries are added. One can get the most up-to-date data simply filtering the most recent entry for each data item. Users of the data vault do not interact directly with the enterprise layer. Instead they enter data using the staging layer and the extract data using the information delivery layer. This layer is highly sophisticated and contains several vaults (separated into basic and advanced features in the assesment).
- **Information delivery layer**: The information delivery layer is the output layer. This layer queries the enterprise layer as a manner of services for users. Different services are implemented as the so called information marts. In a coarse oversimplification an information mart is a small collection of closely related queries to answer some specific question using data from the database.

4.1 The enterprise layer

Perhaps the most distinctive feature of the data vault model is not so much its layered architecture but how the relational tables are organized. In the classical relational databases, there is often 1 table for each concept or entities, and perhaps other supporting tables for the relations. But if you consider an entity and its impelementing table, the table contains the attributes of such entity as columns of the table. Further, there is no default mechanism to trace the origin of the data. This may have some disadvantages for traceability, efficiency, etc. However, these problems cannot be blamed on the relational model but instead should be blamed on the *design* of the tables themselves. The data vault model keep all the benefit of the relational databases, but they redesign the tables so that they are efficient for querying as columnar databases plus they massively increase the flexibility of the architecture (this later often one of the advantages mentioned for alternative non-relational models, but the data vault model makes such difference disappear). Of course there is a price to pay, which in this case it is the lose of normalization and the ever growing storing space needed, but if querying overwhelmingly dominates the transactions e.g. as in contrast to input transactions, then this model advantages generally outbalance its compomises.

In order to design the tables, the data vault uses three major concepts; namely **hubs**, **satellites** and **links**. Let's start by the first two; hubs and relations. Imagine that you have a certain entity with a collection of attributes. The most common solution is to implement such entity as a single table. However, in the data vault that table is exploded in 1 hub and multiple satellites. The hub is the central table to identify the entity and still keeps those attribute that cannot change or change ever so rarely. However, for all the other attributes prone to change, a satellite table is created for each attribute.

Regardless of whether a table is a hub, a satellite or a relations, all tables in a data vault has 3 specific attributes:

- A **timestamp** declaring *when* the data was inserted.
- A **record source** declaring *who* the data was inserted. Here, who is a *business key* identifiy either an individual or other actor responsible for introducing data into a database. Business keys are coded using a hash keys.
- A **sequence number** akin to the unique identifier for each object that represent regular primary keys. However becuase in data vaults data is not deleted but instead new entries are added, in the data vaults tables the primary key is not only formed by the sequence number, but also by the timestamp and record source.

In a naive example, imagine that you have an entity dog with the attributes breed, name, owner and age. In a regular database you will create a single table DOG with these three attributes and perhaps and ID.

- `DOG (dogID, breed, name, owner, age)`

In contrast, in a data vault you will create 1 hub DOG with only those attributes that do not change or do it hardly ever e.g. breed, and then separated satellites for those attributes that change often e.g. name, owner and age;

- `HubDog (dogSequence, timestamp, source, breed)`
- `SatDogName (dogSequence, timestamp, source, name)`
- `SatDogOwner (dogSequence, timestamp, source, owner)`
- `SatDogAge (dogSequence, timestamp, source, age)`

Needless to say that the `dogSequence` in the satellites are foreign keys referencing the `dogSequence` in the hub.

Finally, data vault links are just like regular relations. Most often they will join hubs, but strictly they can link also satellites as well as other relations.

4.2 The information layer

The data vault model does no prescribe a single way of querying the enterprise layer. Notwithstanding, a popular submodel for querying data vault is called the *dimensional model*, first introduced to data warehouses in 1997 by Kimball[2]. The realization of the dimensional model in relational databases is call **star schema**, bacause conceptually it forms a star-like linked collection of tables and views. The idea of the dimensional model and its star schema implementation counterpart is simple; when querying the database, do not only retrieve the main query but also all the related concepts i.e. dimensions, associated with it.

For instance, imagine that you query a data vaults about pets, and you want to retrieve the information of the most recent visit of the pet to the vet. Rather than just retrieve the pet identifier, the visit date and the vet identifier, you also retrieve in collateral views the information about the pet e.g. name, species, age, etc, as well as the information about the vet, e.g. address. In this manner, if the user is interested in "expanding" its query dynamically, it can do so without need to formulated another query. More specifically, rather than just retrieving the core of the star referred to as the *fact table*;

- `FACTLASTVISIT (petID, vetID, date)`

you also retrieve the dimension tables that form the arms of the star;

- `DIMPET (petID, petName, species, petAge)`
- `DIMVET (vetID, vetName, address, popularScore, ...)`

The fact table has two different types of columns: first, the foreign key references to dimension tables (pet and vet), and second, the measure values, i.e. the facts, themselves, in this case the date of the last visit.

The information in the dimension tables may or may not be rendered to the user by default; at the end of the day the user did not query for it. It is more like a pre-loading of its most likely follow-up request of information. For instance, the owner of the pet was obviously interested in the date of the last visit in the first place -that was his query after all, but then upon realizing that his pet need to go again on a regular check, he need to recall the address of the vet. So he interacts with his quey e.g. perhaps a mouse click, to expand the information on the vet. Besides the mechanics, the critical thing about the dimension tables is that they provide the semantics to the fact table. Consider a case of the fact table:

- `(283ho38o32, 78qwbrk2qw, 17-Sep-2022)`

The hash keys for the pet and the vet are not very "meaningful" to the user, but the pet "Sloppy (cat)" and the vet "Jane Wisdom" of Solihull do. This information is contained in the dimension tables. So perhaps a partial view of the dimensional information is a more "useful" answer to the query;

- `(Sloppy (cat), Jane Wisdom (Solihull), 17-Sep-2022)`

Note how not all the dimensional information has to be displayed e.g. the pet age. Often there will be a manner in which the rest of the dimensional information is made available e.g. a click on "Sloppy" may provide further information about its age.

In this project we will constraint ourselves with star schema querying for our information mart layer. Although in principle the model is more generic allowing for so called multiple start, we again will constraint ourselves to the simplest case where each information mart is composed of a single star schema. Notwithstanding, the information mart layer itself will be composed of several information marts.

[2] Note how the dimensional model predates the data vault model.

5 Project architecture

As part of this project, the data vault's enterprise layer will have the architecture shown in Fig. 1. The model has been created using the free option of Draw.io[3]. Note how this architecture is minimal and you will need to complement e.g. adding more satellites as appropriate, adding explicit support for a stimulus boxcar (rather than implicit), etc.

The section 6 described the input data that you will have to read in your staging layer. Finally, in your information layer you will need *at least* have to provide support for the following queries;

1. Individual plotting of the time course of light raw intensity at some wavelength, HbO2 or HbR for some channel.
2. Grand averaged per channel timecourse pictorial representation of the HbO2 and HbR for a group with dispersion regions.
3. A listing of experiments in the database accompanied by the list of factors and treatments given
4. For a certain experiment choose at run time, retrieve the groups and the list of experimental units.
5. Given an individual observation whether of light raw intensity at some wavelength, HbO2 or HbR retrive all available metadata.
6. A boxplot comparing the distribution of either HbO2 or HbR concentrations for two intervals of time for a subject.
7. A boxplot comparing the distribution of either HbO2 or HbR concentrations for two intervals of time for a group.

These above are a minimum set of queries, but in order to get full marks you need to come up with some original queries of your own *interesting for the domain* i.e. trivial queries are not considered. For instance but not limited, if you add support for the stimulus boxcar a boscar based comparison of stimulus periods vs rest periods for both subjects and groups. Another example, if you add support for the probe set up i.e. channel locations, a topographical representation of average values for a certain interval for both subjects and groups on top of a schematical head or a brain atlas.

6 Data description

You will be provided with two *real* deidentified (and reduced) datasets that you can use for testing and exemplification. The two datasets were collected for research purposes by one of the course lecturers (FOE) and his group. Both datasets are now considered to have surpassed the lifecycle associated to their original research goal.

> ☞ Every effort has been made to remove all personally identifiable information from these datasets. Further, they are sufficiently old and intentionally incomplete to eliminate potential privacy and liability problems. Notwithstanding, should there remain any data that can lead to the identification of any participant, we request that you let us know immediately.

The datasets are sufficiently different to illustrate the capabilities of your vault, e.g. they were acquired with different machines (hence the original raw files are different), they have different number of sessions and different temporal course for the sessions, etc.

> ☞ IMPORTANT: The goal of the semester project is NOT the analysis of these datasets per se, but the development of the data vault to store manage them. For the purposes of project marking, you are NOT required to give evidence of any experimental differences on these datasets. However you are suggested associated goals e.g showing such differences, as a way to guide your implementation of the data vault and exemplify its usage.

6.1 A bit of experimental jargon

The formalities of experimental design and analysis are beyond the scope of this project. It is covered by a subfield of statistics and you can find tens of books on the topic. For the purposes of this project you do not need to know formal experimental design and analysis. Notwithstanding, since the data you are going to be using comes from scientific experiments it might be convenient to have some informal notions.

[3] `https://app.diagrams.net/`

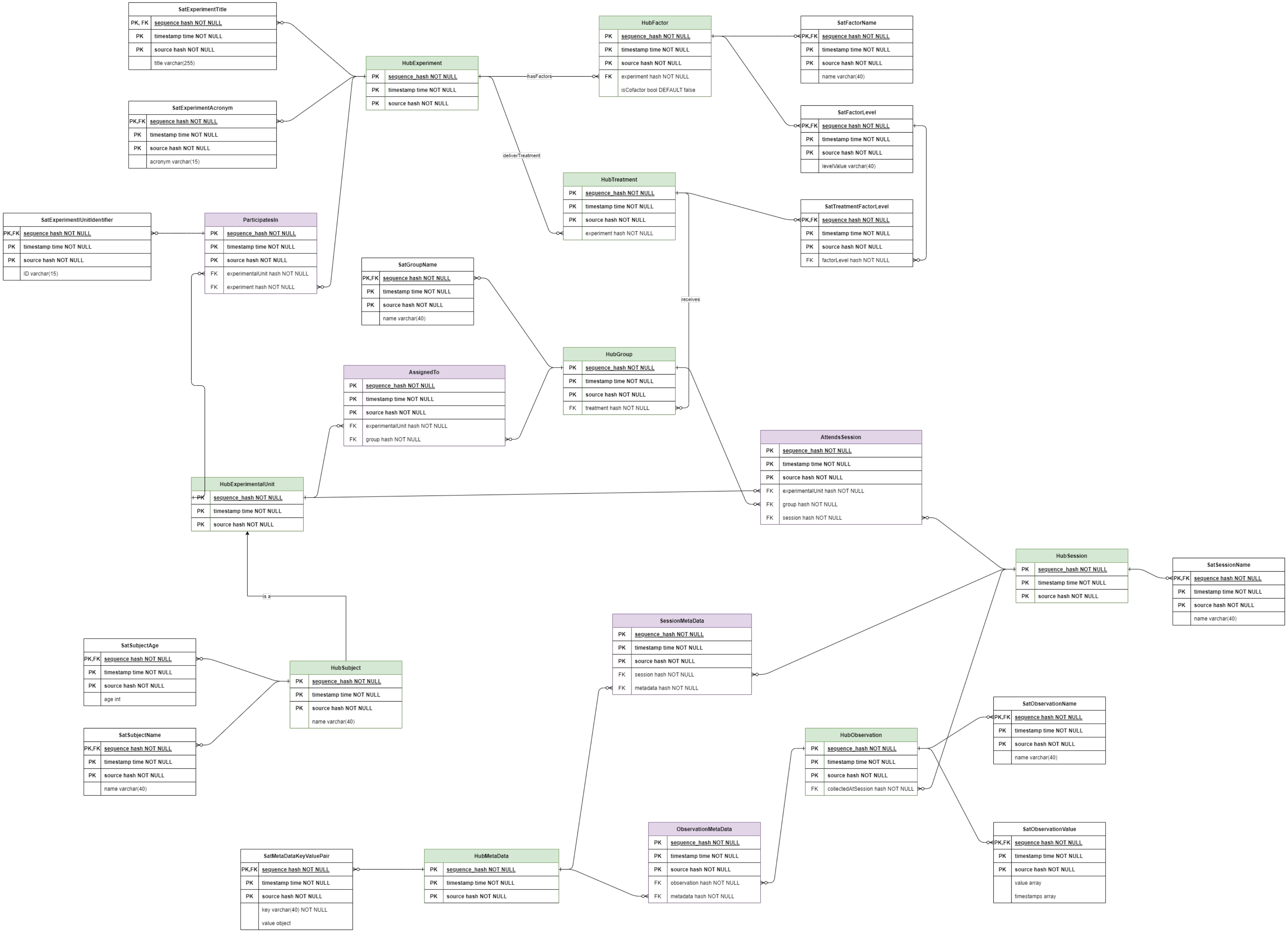
SatExperimentTitle
PK, FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
title varchar(255)
SatExperimentAcronym
PK,FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
acronym varchar(15)
HubExperiment
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
hasFactors
HubFactor
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
FK experiment hash NOT NULL
isCofactor bool DEFAULT false
SatFactorName
PK,FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
name varchar(40)
SatFactorLevel
PK,FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
levelValue varchar(40)
deliverTreatment
HubTreatment
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
experiment hash NOT NULL
SatTreatmentFactorLevel
PK,FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
FK factorLevel hash NOT NULL
SatExperimentIUnitIdentifier
PK,FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
ID varchar(15)
ParticipatesIn
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
FK experimentalUnit hash NOT NULL
FK experiment hash NOT NULL
SatGroupName
PK,FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
name varchar(40)
receives
HubGroup
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
FK treatment hash NOT NULL
AssignedTo
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
FK experimentalUnit hash NOT NULL
FK group hash NOT NULL
AttendsSession
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
FK experimentalUnit hash NOT NULL
FK group hash NOT NULL
FK session hash NOT NULL
HubExperimentalUnit
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
HubSession
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
SatSessionName
PK,FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
name varchar(40)
is a
SessionMetaData
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
FK session hash NOT NULL
FK metadata hash NOT NULL
SatSubjectAge
PK,FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
age int
HubSubject
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
name varchar(40)
SatSubjectName
PK,FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
name varchar(40)
SatObservationName
PK,FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
name varchar(40)
HubObservation
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
FK collectedAtSession hash NOT NULL
ObservationMetaData
PK sequence_hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
FK observation hash NOT NULL
FK metadata hash NOT NULL
SatObservationValue
PK,FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
value array
timestamps array
SatMetaDataKeyValuePair
PK,FK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL
key varchar(40) NOT NULL
value object
HubMetaData
PK sequence hash NOT NULL
PK timestamp time NOT NULL
PK source hash NOT NULL

Figure 1: ER model of the data vault's enterprise layer. .

- A **study** is the observation of some phenomenon; physical, cultural, etc for the purposes of extracting or generating knowledge. There are two major types of studies; observational and intervnetional. In observational studies, the observer (i.e. the researcher) does not make any explicit effort to alter the phenomena e.g. a social scientist passively observing how people behave when entering a pub. In interventional studies, the observer makes explicit efforts alter the phenomena e.g. a chemist testing different concentrations of a solute.
- An (scientific) **experiment** is a very specific type of interventional study where two experimental mechanisms are present; control and randomization. Control means that there is a baseline state of the phenomenon against which the intervention or interventions are compared e.g. a pharmaceutical study with a placebo. Randomization means that the administration of the treatments does not follow a predefined order. A **trial** is just another word for experiment.
- Since an experiment is interventional by definition, the observer ought to change something (intervention **factors**) whilst keeping all other things either constant (**co-factors**). Anything that can affect the outcome of the experiment is not intervened nor maintained constant is referred to as **confounders**. For instance, in an experiment to test how the temperature affects the flavour of coffee, the experimenter changed the temperature (factor), but always used the same type of coffee beam, roasting procedure (co-factors), however did not control for the material or the type of the coffee maker (confounders).
- Factors take different values e.g. different temperatures while making the coffee, or each dose of a new drug. Each tested value is called **level**. If an experiment has more than one factor, a specific combination of levels across the factors is called a **treatment**. Fro example, an agricultural engineer wants to know the effect of a fertilizer on strawberries. He modifies the dose (low and high) as well as time of administration (early vs late). Then there are 4 possible treatments; e.g. (low,early), (low, late), (high, early), (high, late). Depending on the experiment, it is not compulsory that all combinations are tested. But if all combinations are tested, the experiment is to be full factorial.
- The intervention represented by the different treatments are administered one at a time to different individual instances of the phenomenon. Each individual instance is called an **experimental unit**. The whole set of experimental units are called the **universe**. Whether finite or infinite, not all potential experiemental units are probed. The collection of experimental units that are actually tested from the universe is called the **cohort**. For instance, for our agricultural engineer the experimental unit are each strawberry plant, the universe are all the straberry plants that exist (or will exist), and the cohort is the handful of plants that he has in his greenhouse. Or in another example, in a medical trial, a new surgical intervention is being tested. Each patient is an experimental unit (when the experimental unit is a human, the term experimental unit is more changed for the term **subject** or participant or volunteer), the world population is the universe and the cohort is the few hundreds subjects participating in the study.
- Within the cohort, all the experimental units that receive the same treatment are referred to as a **group**, e.g. the control and intervention groups. An experimental unit may be (randomly) assigned to one or more groups yielding different experimental designs. Two popular designs are the so called **between-subjects**, where the experimental units are given at most 1 treatment (e.g. no overlap between groups), and **within-subject** where every experimental units is given *all* treatments.

6.2 About the application domain

The datasets that you are given with this project are two neuroimaging datasets acquired using Functional Near Infrared Spectroscopy (fNIRS). fNIRS is a biomedical imaging modality capitalizing on light absorption [9, 7]. It works as follows; Infrared light at several wavelengths (most commonly 2[4] but some devices operate with more than 2) is shined on the scalp by means of light sources or photoemitters. e.g. LED or lasers, and then upon exiting the head tissues, the light that has not been absorbed by the tissues is captured using photodetectors. The lowering of the light intensity (attenuation) depends on the composition of the head tissues, and importantly a major contributor to such attenuation is the concentration of hemoglobin in its both species; oxygenated and reduced. Therefore, the light collected at the detector conveys information about tissue oxygenation in the form of relative concentrations of oxy- and deoxy-haemoglobin. Moreover, since such concentrations of hemoglobin are in part dependent on the brain activity (i.e. the neurons to activate require oxygen), fNIRS can be used to monitor brain activity indirectly.

[4] Both datasets in this projects use only 2 wavelengths.

In a little bit more detail, an fNIRS device consists of a number of emitters and detectors. These, both emitters and detectors are arranged by the experiment in different preselected locations at the scalp by the researcher depending on the region of the brain that he wishes to monitor, but nonetheless, it does so following a small number of rules;

- In order to monitor some brain region, a light emitter has to be paired with a light detector. This is called a *channel*. For optical reasons, the most common separation between a light emitter and its coupled detector is approximately 3 cm in a human adult.
- With a careful arrangement of the available light emitters and detectors, every emitter may be coupled to more than one detector and viceversa; each pair is a channel.
- Channels are positioned according to specific standard locations on the head. There exist several positioning systems with the most popular being the 10-20 international system [3].
- In every channel, all the wavelengths available in the device are operated. That is, every channel records as many signals as wavelengths. Suppose that you have an arrange of 12 channels and 2 wavelengths, then the fNIRS neuroimage will consist of 24 signals recorded in time. Moreover, because the experimenter knows “where” the channels (the mid point between an emitter-detector pair) are located (e.g. according to the 10-20 system), then he can form an image in the classical sense where every channel is akin of an image pixel at some coordinates, and any non probed intermediate locations are simply interpolated.

Most fNIRS devices allow the researcher to save the data in two major forms;

- As light intensities (at each operating wavelength)
- As Hb concentrations (both oxy- and deoxy-)

If the data is saved as light intensities, it needs to be converted to Hb concentrations to be useful. This conversion process is known as reconstruction. There are several models for reconstructing Hb data from light intensities, with the most common one being the modified Beer-Lamber law (MBLL) [1].

> ☞ For this project you are already given both forms of data; raw intensities, and Hb reconstructed data. Therefore, you do NOT need to understand the MBLL nor need to know how it operates; you just need to know that the Hb data comes from the raw light intensities data. Notwithstanding, for practical purposes of this project, you can ignore it all together!

6.3 Dataset 1: Visuomotor functional connectivity

~~The dataset was collected only for the purposes of validating processing and analysis algorithms for fNIRS neuroimages. The dataset was collected by FOE during his time at Imperial College. The original dataset consisted of data collected from 10 subjects, but for this project you are only given 6 of those.~~

You are provided with the fNIRS raw ligth intensity data (those with substring `_MES_` in the filename) as well as the oxy-hemoglobin (HbO2) and deoxy-hemoglobin (HbR) data.

Dataset associated goals

The original aims of the dataset were to serve as a baseline for validation of data analysis algorithms.

Experimental Protocol

Each subject performed four different exercises as indicated in Table 1.

		Motor Stimulus	
		Not Present	Present
Visual Stimulus	Not Present	Rest	Motor Only
	Present	Visual Only	VisuoMotor Task

Table 1: Experimental stimuli.

The exercises were conducted in a classical block design paradigm. During each exercise, the stimulus task was repeated 5 times in fixed length blocks with interblock rest periods in the following order: 30

seconds initial rest, 5 episodes of 20 seconds task, followed by 30 seconds rest. The Resting state also follows this particular setup, although no stimulus were provided. The execution of the four exercises were done as separated recordings but all four were acquired during a single experimental session. The execution of the different tasks were randomised to avoid possible biasing due to the order of execution, and at least 2 minutes were left between exercises, to leave the brain time enough to fully recover from previous exercise.

Stimulus

As the goal of the original experiment was not to establish new hypotheis about brain function but rather testing new algorithms, both visual and motor stimulus were very well known tasks for which the expected response can be easily found in literature.

The visual stimulus was a checkerboard projected in a screen, which is known to lead to activation in the visual cortex [8]. Visual Stimulus consists of the projection of a blinking checkerboard alternating at 2 [s^{-1}] in contrast into a computer screen, with rest consisting on a black screen.

The motor task was a finger thumb opposition which is known to trigger activation in the motor cortex [5]. No prior training for uniformity of tapping was used.

The participants remained seated for the duration of the experiment, and were be asked to keep their heads as still as possible. During intertrial periods, subjects were be asked to keep their eyes closed. Rest consists of eyes close (plus screen switched off) and hands lay still on a table (or on top their knees).

Probe Positioning

The functional brain behaviour as represented by its haemodynamics for each subject was monitored using a Near InfraRed Spectroscopy device (NIRS) (Hitachi ETG 4000 OTS) at 10Hz.

Probes were positioned according to the UI 10/10 system [3]. In the following manner:

- Left Motor Cortex (right hand fingertapping): Central row middle probe was positioned in C3 which can be found 30% from the left preauricular point (LPA-T9) on the coronal central reference curve.
- Visual Occipital Cortex (visual stimulus): Central row middle probe was positioned in Oz, which can be found 10% from inion in the sagital central reference curve.

See Figure 2 for the approximate probe positioning and channel locations.

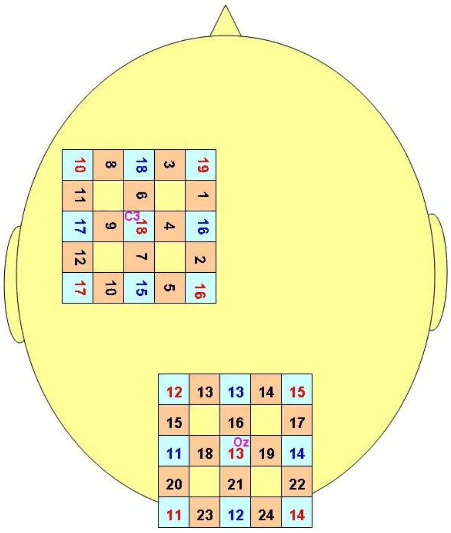

Figure 2: Dataset 1. Approximate probe positioning and channel locations.

Files description

In this dataset, each subject comes with 3 files in `.csv` format:

- `*_MES_Probe1.csv` - This is the raw light intensity data at the 2 wavelengths. You will find that the file has two parts. The first part is meta data, and the second part contains the data itself. In the metadata part the information is structured as key-value pairs. In the data part the information is structured as a large array for which the rows are samples in time (check the meta data for the sampling rate), and the columns correspond to the 2 wavelengths per channel.

- `*_HBA_Oxy.csv` / `*_HBA_Deoxy.csv` - These are the HbO2 and HbR reconstructed data. Reconstruction has been carried out with the MBLL from the corresponding `*_MES_Probe1.csv` data. Again, you will find that these files have two parts. The first part is meta data, and the second part contains the data itself. In the metadata part the information is structured as key-value pairs. In the data part the information is structured as a large array for which the rows are samples in time (check the meta data for the sampling rate), and the columns correspond to the change in Hb concentration per channel.

All of the 3 files above are text files and therefore they can be opened with any text editor (e.g. Notepad or Notepad++ in Windows, Braces in Mac, gEdit in Linux, etc), or spreadsheet (e.g. MS Office Excel) or even word processors (e.g. MS Office Word). The original files from the HITACHI device are bigger than the ones you are begin provided. They have been simplified for the purposes of this project.

6.4 Dataset 2: Pre-autism dataset

The dataset was collected for the purpose of studying biomedical markers of preautistic disorders. The dataset was originally collected at the Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) in Mexico by MRC, a student of FOE for her thesis.

The original dataset is a multimodal dataset with both fNIRS and electroencephalography (EEG) records. As aforedescribed, fNIRS is an optical neuroimaging modality to monitor brain hemodynamics. EEG is also a neuroimaging modality but senses the electrical activity of the firing neurons. The behaviour of the EEG and fNIRS recordings are linked by the neurovascular coupling, a physiological phenomena ruling the oxygen intake by the neurons which is poorly understood and non-linear neither in space or time. For this project you are NOT given the EEG data, but only the fNIRS data. The original dataset also collected the Autism-Spectrum Quotient Test (AQ) which has been concealed here intentionally. Also, the original dataset consists of 43 subjects; here you are only given 10.

You are provided with the fNIRS data, both the raw light intensities and the Hb reconstructed data.

Dataset associated goals

The original goal was to study the differences on brain hemodynamics among of the preautistic groups; narrow, medium and broad [6]. The groups were determined using the AQ scores.

Experimental Protocol

In addition to completing the AQ, each subject performed two different conversational tasks: first without (session 1 -normal conversation) and then with syllabic stress (session 2 -stressed conversation). The participants were to speak to the interviewer sitting in front of them during the task. The conversation task period consisted of 6 cycles of 30-s conversations. During data collection, the ordering of the stimulation was randomized, but here we have already sorted them for you.

For activation tasks, participant and opposite talker, sat opposite each other in comfortable chairs with their eyes open and had to engage in a food conversation. Before beginning and after finishing the conversation tasks, participant and interviewer were separated by a door that separated two rooms of $\sim$ 1.5 m × 1.5 m, that is, each one was in different rooms (Fig. 3). The door was used to avoid introducing any stimuli other than conversations.

In order for the interviewer to be aware of the session times, he was supported by a monitor. To start the conversation tasks, the interviewer opened the door once the screen showed the phrase 'Experiment begins' (3 s before the conversation tasks). At the end of the conversation tasks, the interviewer said goodbye in 5 s and then closed the door.

Stimulus

A conversation consisted of the interviewer's turn ('listen 1), and the participant's turn (speaking). The only difference between both sessions, is that the stressed conversation condition that included syllables

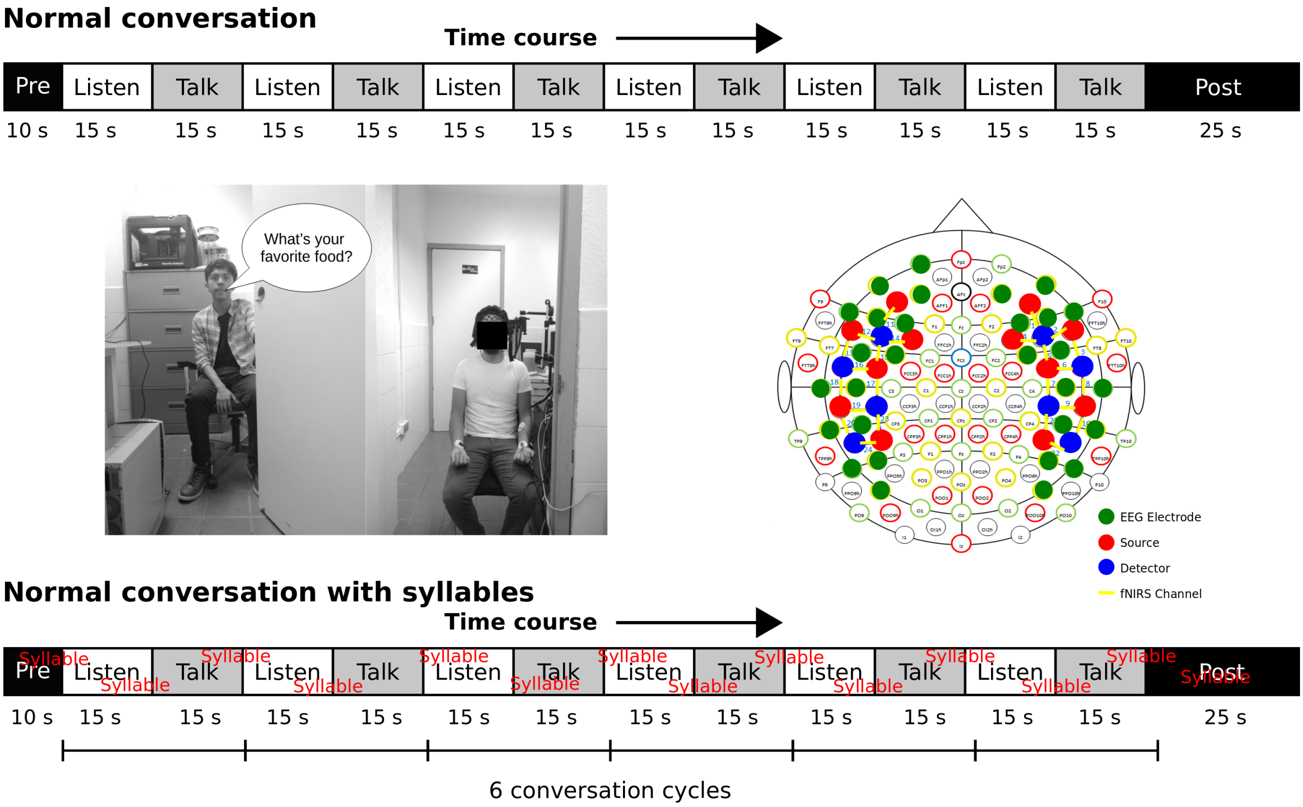

Figure 3: Dataset 2 - Preautism Experimental Setting. Left: Task procedures and setting. Right: Probes positioning. Top and Bottom: Sessions schematic timecourses.

such as ‘na’, ‘ka’, ‘ta’, or ‘a’ that the subjects were instructed to mention sporadically throughout the duration of the session. A typical initial question was ‘what is your favorite food?’, thereafter the conversations were developing. Some were more specific, for example, regarding the participant’s diet, and other questions were more typical regarding day-to-day meals.

Probe Positioning

Subjects were fitted with a cap containing the fNIRS and EEG probes according to the measured coronal (ear-to-ear) and sagittal (forehead-to-back-of-head) distances. The center of the probe set (Cz) was placed at half the distance of the aforementioned measurements, which was considered the estimated midpoint of the motor cortex. During both tasks, simultaneous measurements of EEG, fNIRS were acquired. Before data collection, the lasers were heated at least 30 minutes before. Relative changes in HbO2 and HHb were measured using a NIRScout (NIRx, Germany). Absorption was measured at 2 wavelengths of near-infrared light (760 and 850 nm) and a sampling frequency of 5.2 Hz. The NIRx system is multiplexed in time, so the sampling frequency is determined by the number of light sources and the lighting pattern used. From these, the HbO_2 and HHb were reconstructed using the modified Beer-Lambert law. A probe holder (16 × 8) was used for NIRS with 12 sources and 8 light detectors forming 24 optical channels. The distance between the source-detector optode pairs was approximately 3 cm. To reduce ambient light, dark caps were placed on each optode that also served to keep the optodes well coupled to the scalp.

At the same time, electrical activity was recorded using a 29-channel EEG machine g.HIamp (g.tec, Austria) from standard Fp1, AF7, AF3, AF4, AF8, F7, F5, F3, F4, F6, F8, FC5, FC3, FC4, FC6, C5, C6, T7, T8, TP7, CP5, CP6, TP8, P7, P5, P6, P8, PO7, PO8 positions, using the international method 10-20. After electrode placement, gel was applied to each and the skin surface was lightly scraped to reduce impedance. The impedance at the electrodes was confirmed to be less than 10 kΩ. The electrodes were placed in the regions: Broca Area (fNIRS-EEG), superior temporal sulcus (STS) (fNIRS-EEG) and Fusiforme Gyrus (EEG).

Files description

In this dataset, each subject comes with the following files:

- `.hdr` - Header file with meta-data only.
- `.dat` - The Hb reconstructed data. The information is structured as a large array for which the rows are samples in time (check the meta data for the sampling rate in the `.hdr` file), and the columns correspond to the changes in HbO2 and HbR concentrations per channel.
- `.wl1` - Raw intensities at wavelength 1. The information is structured as a large array for which the rows are samples in time (check the meta data for the sampling rate in the `.hdr` file), and the columns correspond to the raw intensities at wavelength 1 per channel.
- `.wl2` - Raw intensities at wavelength 2. The information is structured as a large array for which the rows are samples in time (check the meta data for the sampling rate in the `.hdr` file), and the columns correspond to the raw intensities at wavelength 1 per channel.
- `.evt` - List de eventons (timeline)

All of the files above are text files and therefore they can be opened with any text editor (e.g. Notepad or Notepad++ in Windows, Braces in Mac, gEdit in Linux, etc), or spreadsheet (e.g. MS Office Excel) or even word processors (e.g. MS Office Word). The original files from the NIRScout device are bigger than the ones you are begin provided. They have been simplified for the purposes of this project. Further, it also contained other files omitted here for simplicity.

7 Final remarks

- This is a *research* project. That is although full marks are guaranteed with knowledge acquired in the lectures and strictly bounded byt eh module's syllabus, you will be in a much better position to guarantee thoese marks, if you go above and beyond. Moreover, things are not always explicit. The lectures are not cooking recipes to solve problems. Instead they afford you the minimum knowledge to address them, but more importantly, they empower you with the capacity to learn on your own. Make sure you use this capacity here; thinking is not a luxury but a demand!
- Do not assume that just because something works in your computer it means that it is correct, specially in postgreSQL which is machine dependent. Extensive testing is required. This is not something specific of this project nor a detriment of your learning process; this is a factual reality of programming. That is why companies spend large amounts of money in beta testers, and even then they still sometimes got it wrong. Of course, you do NOT have to spend money, but you have to be aware that one hit (e.g. correct running in your machine under some marginal testing) is a guarantee that your code works elsewhere under different conditions. Here, you are given as much as possible information about the conditions where you are going to be evaluated. These should suffice but still you are strongly encourage to book a presubmission check session with your assigned evaluator (this will be assigned later in the course).
- The project is about data vaults, a specific architecture for relational databases. It is not about neuroscience, nor neuroimages, nor experiments, factors, and treatments. This latter is just the application domain. The concepts that are central to databases are transversal to whether yur database is about banking and financing, a library, or why not, scientific knowledge. It is extremely important that you understand this and abstract yourself from the application domain. The sooner you do that, the sooner you will realize that understanding the domain of application is utterly irrelevant. Just like in algebra (the abstraction) the arithmetics of adding 7+3 does not change depending on whether the 7 and the 3 represents pounds, books, goats, or spaceships. You do not need to understand what an elephant is to know how to add 7+3 elephants. Therefore **you do NOT need to understand the application domain**. In databases, tables, relations, primary and foreign keys, etc or in the data vault hubs, satellite and links are concepts that not transcend the application domain, in this case neuroscience and neuroimages. Ergo, **you do NOT need to understand neuroscience nor neuroimages to solve the project**.
- Your assessment is far more dependent on the report demonstrating that you have mastered the knowledge than on having correct code.

Bibliography

[1] Mark Cope, David T. Delpy, E. O. R. Reynolds, Susan Wray, J. S. Wyatt, and P. Zee van der. Methods of quantitating cerebral near infrared spectroscopy data. *Advances in Experimental Medicine and Biology*, 222:183–189, 1988.

[2] Krzysztof J Gorgolewski, Gael Varoquaux, Gabriel Rivera, Yannick Schwarz, Satrajit S Ghosh, Camille Maumet, Vanessa V Sochat, Thomas E Nichols, Russell A Poldrack, Jean-Baptiste Poline, et al. Neurovault. org: a web-based repository for collecting and sharing unthresholded statistical maps of the human brain. *Frontiers in neuroinformatics*, 9:8, 2015.

[3] Valer Jurcak, Daisuke Tsuzuki, and Ippeita Dan. 10/20, 10/10 and 10/5 system revisited: Their validity as a relative head-surface-based positioning systems. *NeuroImage*, 34:1600–1611, 2007.

[4] Dan Linstedt and Michael Olschimke. *Building a Scalable Data Warehouse with Data Vault 2.0*. Morgan Kaufmann, 2015.

[5] Michael M. Plichta, Martin J. Herrmann, Christina G. Baehne, Ann-Christine Ehlis, and Andreas J. Fallgatter. Brain activation in the visual and the motor cortex assessed with event-related functional near infrared spectroscopy (fnirs): are the results reproducible?, 2005.

[6] Michelle Rojas-Cisneros, Felipe Orihuela-Espina, and Samuel Montero-Hernández. Analysis in the broader, medium, and narrow autism phenotypes using fnirs. In *2021 OSA Biophotonics Congress: Optics in the Life Sciences*, 2021.

[7] Felix Scholkmann, Stefan Kleiser, Andreas Jaakko Metz, Raphael Zimmermann, Juan Mata Pavia, Ursula Wolf, and Martin Wolf. A review on continuous wave functional near-infrared spectroscopy and imaging instrumentation and methodology. *Neuroimage*, 85:6–27, 2014.

[8] Matthias L. Schroeter, Markus M. Bücheler, Karsten Müller, Kamil Uludag, Hellmuth Obrig, Gabriele Lohmann, Marc Tittgemeyer, Arno Villringer, and D. Yves Cramon von. Towards a standard analysis for functional near-infrared imaging. *NeuroImage*, 21:283–290, 2004.

[9] Arno Villringer and Britton Chance. Non-invasive optical spectroscopy and imaging of human brain function. *Trends in Neuroscience*, 20(10):435–442, 1997.