

Design Concepts for NeuroCore and Neuroscience Databases

Jeffrey S. Grethe,¹ Jonas Mureika,² and Edriss N. Merchant²

¹*Center for Cognitive Neuroscience, Dartmouth College, Hanover, New Hampshire*

²*University of Southern California Brain Project,
University of Southern California, Los Angeles, California*

3.2.1 Design Concepts for Neuroscience Databases

In Chapter 3.1, we gave a review of our design of NeuroCore to provide extendible databases for repositories of experimental neuroscience data. In this chapter, we offer a more formal presentation of NeuroCore. In designing the NeuroCore database, four specific design concepts were taken into account:

1. Due to the heterogeneity in the data and the methods used to collect the data in a neuroscience laboratory, the database should be easily modified to meet a specific laboratory's need while remaining compatible with other databases.
2. In order for a researcher to fully integrate these new data repositories into their own laboratory, the database should be an experimental tool as well as a data storage, retrieval, and analysis system.
3. Due to the diversity of the neuroscience community, there will not be a single monolithic database to store all neuroscience data. This is in contrast to the molecular biology community, where a few monolithic databases are being used to store data concerning genetic sequences.
4. Like the molecular biology community, however, any tools developed should foster intra- as well as inter-laboratory collaborations.

These four design principles lay out the specific requirements that any neuroscience database framework

must fulfill. The following sections give an overview of the major components of the NeuroCore database and how they relate to these design criteria.

3.2.2 The Three Main Components of the NeuroCore Database

In order for the database to be easily modified to meet a specific laboratory's need, a core database schema, with a standard core structure, was created that can be extended as each researcher sees fit (see Fig. 1). By exploiting the object-oriented properties of new object-relational database technologies (i.e., the definition of new data types and the ability to construct inheritance hierarchies), we were able to create a database schema that is easily modified without requiring major changes to its core structure. The NeuroCore database consists of four primary components:

1. Core database
2. Database extension interface
3. Database federation interface
4. New datatypes for the database created as datablades (see Appendix A2 for more information on the Neuroscience Time-Series Datablade)

Each of these three components is discussed briefly in the following sections.

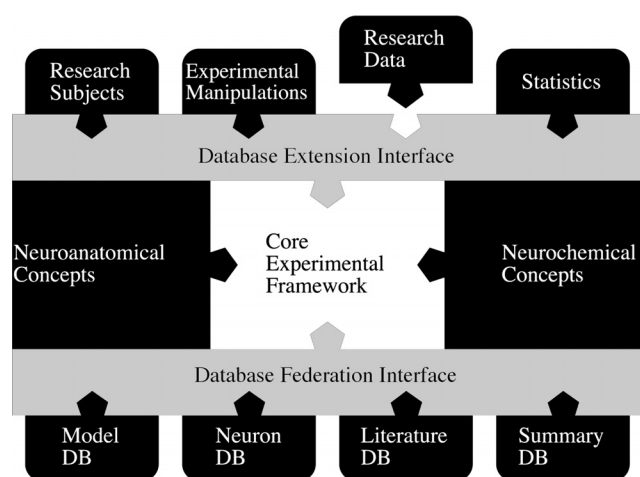


Figure 1 Overview of the NeuroCore database. The core database consists of a description of the experimental framework as well as constructs to define the neuroanatomical and neurochemical concepts needed in defining a neuroscience experiment. In order for individual laboratories to be able to store their own data, an extension interface is attached to the core schema to allow users to extend the database as they see fit.

Core Database Schema

The core schema of NeuroCore consists of five basic components:

- Core experimental framework
- Neuroanatomical concepts
- Neurochemical concepts
- Database federation interface
- Database extension interface

The following sections will give an overview of each of these areas of the NeuroCore database. For a more detailed description of each area, including a description of the database tables used in the implementation of the database, please visit the USCBP website at <http://www-hbp.usc.edu>.

CORE EXPERIMENTAL FRAMEWORK

The core experimental framework in NeuroCore contains the database tables that store the critical information needed for storing an experiment. The most important information is the hierarchical description of what constitutes an experiment. Attached to this hierarchical definition of an experiment are mechanisms to store the protocols involved in the experiment. Also included in this core framework are the description of the research subjects involved in the experiment as well as the description of the researchers involved in the experiment.

NEUROANATOMICAL CONCEPTS

The neuroanatomical section of the NeuroCore database contains the basic neuroanatomical information needed to provide a framework for storing specific

neuroanatomical data as well as providing the proper anatomical framework to store any neuroscience experiment within NeuroCore. The storage of neuroanatomical concepts within the NeuroCore database can be accomplished in three areas:

- Information regarding single cells and their associated cell types that have been stored in the database
- The storage of multiple anatomical dictionaries that can be used to reference atlas locations within the database
- The storage of graphical anatomical atlases

NEUROCHEMICAL CONCEPTS

The neurochemical section of the NeuroCore database contains the basic neurochemical information needed to provide a framework for storing neurochemical data within NeuroCore. The NeuroCore database allows for the storage of chemical dictionaries that can be used to reference chemicals from within the database.

DATABASE FEDERATION INTERFACE

The federation interface enables each NeuroCore database to be able to share information with other databases. To this end, three basic considerations were taken into account:

- The core database structure needs to contain information concerning the basic terminology used in structuring and storing the data (e.g., neuroanatomical and neurochemical dictionaries).
- All data stored in the database should contain information regarding the unit of measurement on which the data are based (e.g., cm vs. mm).
- Support for various levels of security is necessary so that researcher's can store and share their data securely.

Database Extension Interface

The database extension interface provides mechanisms whereby researchers can augment their database to store necessary information regarding their particular experiments. The core database contains the database tables necessary to describe the core information concerning any neuroscience experiment (e.g., the experiment table, which contains information on the name of the experiment, the type of the experiment, and the researchers involved in the experiment). However, the core database also contains "parent" super-tables that allow researchers to attach their own table definitions as "child" sub-tables to the core database in an inheritance hierarchy. Recall the definition of inheritance in Chapter 1.2 (Introduction to Databases) and the description of inheritance and its relation to NeuroCore in Chapter 3.1. This allows the database to be extended so that it can store relevant information concerning the research

subjects (e.g., the *researchSubject* table allows researchers to attach their own tables as sub-tables to this table to describe the subjects used in their experiments). A standard behavioral neuroscience laboratory might have a table describing animal subjects and another describing human subjects used in their laboratory, the experimental data collected, the experimental protocols used, and any annotations or statistics (metadata) normally included in the experiments. This allows the researcher to add data specific to their laboratory in a fairly simple manner without the need to change the core database structure. Examples of specific extensions to the core database were given in Chapter 3.1, *Repositories for the Storage of Experimental Neuroscience Data*, whereas some interfaces designed to help users work with these databases are described in Chapter 3.3, *NeuroCore On-Line Notebook*.

Neuroscience Time-Series Datablade

The last component of the NeuroCore database is the Neuroscience Time-Series Datablade (see Appendix A2 for further details). Time-series data cannot normally be stored efficiently with standard commercial databases as they have no standard mechanisms for storing and manipulating physiological time series (most commercial databases would need to store all the individual data points from a time-series recording as individual entries in a table). In order to allow for the efficient storage of such data, a new datatype was created to store neurophysiological time-series data. This new datatype, created in InformixTM as a datablade, stores the actual time-series data as an opaque data type (internally stored as a large binary object in the database) that can be manipulated in the database.

3.2.3 Detailed Description of the NeuroCore Database

Overview of Table Descriptions

In developing the core database structure, various aspects of what constitutes a neuroscience experiment were considered. Not only does the core schema need to incorporate the generalities of an experiment, it must also allow the user to capture information regarding neuroanatomy, neurochemistry, and neurophysiology in a structured format. Each of the following sections describes an aspect of NeuroCore necessary to provide a structure that will allow any user to extend the core database structure so that the extended database can capture *any* neuroscience experiment.

The NeuroCore database consists of objects, concepts, or entities that need to be stored for a neuroscientific experiment and the relations that exist between these objects. The objects in the database are implemented as tables which consist of one or more columns that define

what constitutes the object described by the table. Each instance of an object, a single data entry, is entered as a row (tuple) in one of these tables. The tables themselves can be categorized as follows.

CORE TABLE

The core tables in the schema are all the tables that allow for the storage of data related to specific concepts that are part of the core framework. One example of such a core table is the table that stores information regarding experiments that are stored in the database.

SET BACKING TABLE

The set backing tables in the schema allow a many-to-many relationship to be defined between data across tables. For instance, to associate the person table with the laboratory table, we first realize that many people can be associated within a laboratory and similarly many laboratories can be associated with a particular person. To implement this situation, a set backing table is used to hold the multiple entries (see Chapter 1.2 for more information on set backing tables).

DICTIONARY TABLE

Defines the terminology that is needed to build a structured framework for the storage of neuroscience data. For example, the names of neuroanatomical structures or neurochemical compounds that need to be defined are stored in dictionary tables so they can be referenced from other tables.

EXTENDIBLE PARENT TABLE AND EXTENSION TABLE

The extendible parent and extension tables comprise the framework that allows for individual users to tailor the database to suit their own needs. The extendible parent table is a super table that allows users to attach their own extension tables to the core database structure. For example, all tables that define what research data are to be stored in the database are extension tables and are children of (attached to) a single extendible parent table.

The relations between the objects implemented in the database schema can be classified into two general categories, as follows.

Inheritance Relations This relation describes the parent/child (super-table/sub-table) relations in the database that form the object-oriented structure (inheritance hierarchy) of the database. These relations are the necessary link between the extension tables and the extendible parent tables that form the basis for adding extensions to the database.

Foreign Key-Primary Key Relations The foreign key-primary key relation is a standard construct that allows one database entry in a specified table to relate to one or more entries in another table (e.g., a set of email

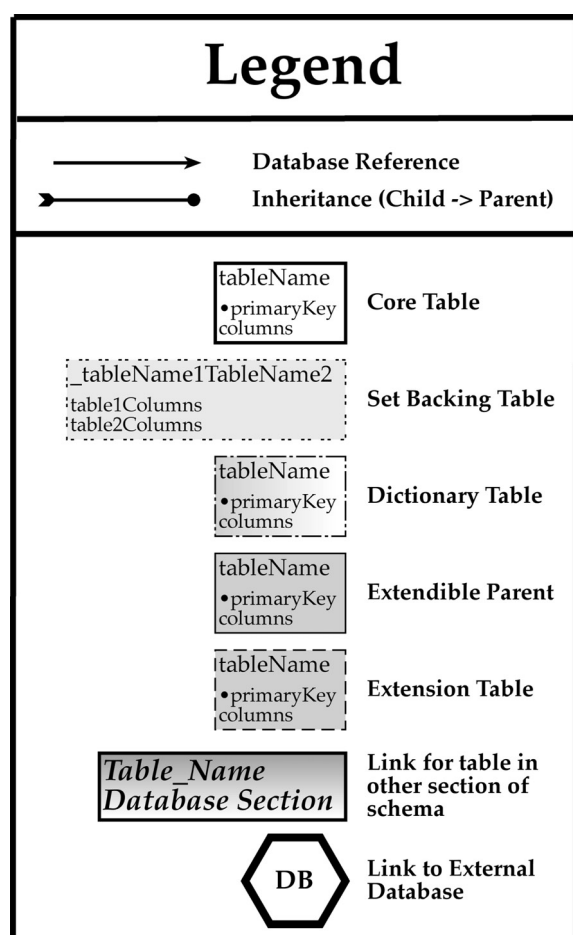


Figure 2 Legend for all schema diagrams. These diagrams represent the database schema by displaying the actual tables and their contents with the relations (primary key-foreign key or inheritance) that exist between the tables.

addresses is associated with one person; a research subject has only one species designation). Relations that map one entry to one or more entries (1-1 or 1-n) are captured by the definition of the objects themselves. However, relations that may map a set of entries to another set of entries (n-m) are defined through the addition of *set backing tables*. Each aspect of the core described in the following sections is accompanied by a schema diagram (see legend in Fig. 2) which shows all the tables and relations involved in that particular aspect of the core database.

Top-Level of Core Database

All tables in the core database schema are created as sub-tables (children) of a top-level database object (*redObject*; see Fig. 4). This allows all tables in the core schema to have certain necessary or global information defined for them at a higher level. Each section of the core schema falls under its own super-table (parent) which is a child of the *redObject* table. The *redObject* table itself just defines that each tuple will have a

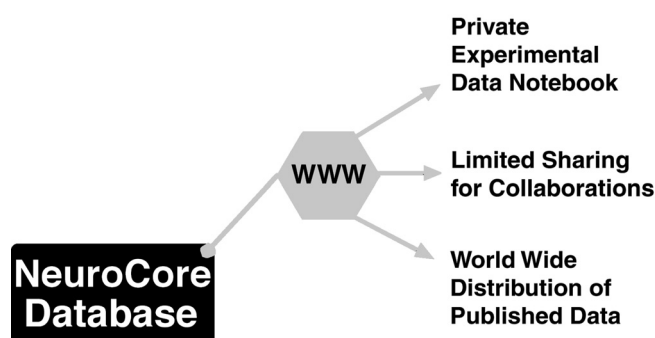


Figure 3 Various levels of access taken into consideration in the design of NeuroCore. Access to the database can be limited to individual researchers or can be granted to allow either a set of collaborators or the general public to view the data.

uniqueID. Three tables are then used to partition the database into the following categories:

- *databaseMetadata*: all the necessary information regarding the tables, columns, relations in the NeuroCore database
- *contactDB*: all the information regarding researchers and laboratories that needs to be stored in the database
- *databaseCore*: the parent for all the experiment related tables in the database. This section of the database can further be partitioned into the following categories:
 - *experimentCore*: contains a hierarchical description of the experiment and the framework for storing the protocols used within this hierarchy and the research subjects used in the experiments
 - *neuroChemCore*: defines the necessary neurochemical constructs needed by the database
 - *neuroAnatCore*: defines the necessary neuroanatomical constructs needed by the database
 - *coreExtension*: the interface layer where users can add extensions to tailor the database to meet their own needs

This hierarchical structure with a single root object is also useful in another regard; one can associate a global relation from any tuple in the database to a specified table, as the uniqueID defined by the *redObject* table is unique across all tables in the NeuroCore database. For example, we may want to be able to associate a literature reference found in an external article repository (AR) to specific tuples in the database through the use of the *redReference* table. This is important because a literature reference will relate to a specific set of tuples in the database (e.g., authors, animal subjects, research data, etc.) which are distributed across many different tables. Each of the tuples can also be related to different sets of references. It is important to remember that the NeuroCore database is not designed to store all neuroinformatics data (e.g., published articles, neural models, summaries and annotations). NeuroCore, therefore,

needs to be able to refer to external federated databases that contain this information. Article repositories are a prime example of this concept. Many scientific journals being published today are also being published in an electronic format (e.g., the *Journal of Neuroscience* published by the Society for Neuroscience is now available electronically through the society's Web site); therefore, they will need to be able to be referenced from within NeuroCore and will also need to be able to reference data stored within NeuroCore (see Chapter 3.3 for more information regarding the interaction between article and data repositories).

Another very important aspect for any experimental database is the issue of database security. For a database to be widely accepted in the neuroscience community, researchers must be able to decide who can retrieve, view, and analyze their data. Unlike standard database security constraints where one can set access privileges for entire tables or even, in some cases, specific columns found in certain tables, a scientific database needs to be able to define access privileges at a much finer scale, on the level of individual tuple entries. Sharing of data in a scientific community can occur at three distinct levels (Figure 3).

PRIVATE

Private indicates data that the researcher does not want made public to anyone. This would include data that have been recently collected or are currently being analyzed and have not yet been published. Information stored as private would form a researcher's personal experimental database. This is the default access privilege for all newly entered tuples, and the only access privileges given to these tuples would be to the owner.

COLLABORATORY

In certain instances, it will be useful for the researcher to share information with a select group of individuals and/or other groups. This method of sharing requires that a specific set of researchers be given access to a limited set of tuples throughout the database.

PUBLIC

Once data have been formally published, they can be made available to all interested individuals. In order to share data at these various levels, a tuple-level security system (i.e., security at the level of individual data entries) has been implemented where each tuple in the database can be given numerous security conditions (Fig. 5). There are three access privileges for a tuple that can be granted to any user:

- Select: allows the user to view the information stored in that tuple
- Update: allows the user to update information in the tuple
- Delete: allows the user to delete the entire tuple

Each grantor who receives any of these privileges can also be given the permission to grant these privileges to other users. When allowing users to grant access to certain individuals and groups (collections of individuals), one also needs to be able to explicitly revoke privileges to a particular user. The *redSecurity* table stores all information regarding tuple access permissions (tupleAuth of each user stored in the *databaseUser* table). In addition to being able to define individual users, one can define groups of users (a group is also a database user) through the *_userGroup* table. Each user is also assigned to a specific *userClass* and *userStatus*. Each user's personal information is stored in the contact portion of the database (Fig. 7) and is referenced through the *_userPerson* table. A complete security log is also maintained as part of the *updateLog* table.

For each table created as part of the NeuroCore database, certain information needs to be stored so that all tuples and their relations with other tables can be fully described (Fig. 6). Including the release information stored in the *schemaVersion* table, information is stored that defines the functionality of all tables within a NeuroCore database and all columns within these tables (*databaseTables* and *databaseColumns*). The general information contained in these tables allows users to query this information to discover the description and purpose of each table along with its contents. Along with the general description of each table column, each column's unit of measurement is referenced (with the *measurementTypes*, *measureSystems*, and *measurementUnits* tables that are referenced through the *_dbColumnUnits* table), which can aid in the conversion of data from one format to another format. This becomes very important when one begins dealing with multiple databases residing in a federation. In order to share data from these databases, we must know what units of measurement have been used to store the data and in what format the data have been stored (this is especially important for image data, which are stored in a specific format and also have specific dimensions).

The last piece of information that must be stored in order to fully describe any NeuroCore database are the relations that exist between tables in the database. All relations in the database are described in the *databaseRelation* table (each set of columns in a relation is referenced through the *relationColumns* table, whereas specific relation types are defined by the *relationTypes* table). By storing all the information concerning the tables and their relations, one is able to completely reconstruct the database from the information contained in the metadata tables. One is then able to create tools that can build browsing, query, and input interfaces dynamically based on the information contained in the metadata. This affords users of NeuroCore databases a simple mechanism through which they can federate their databases. This also makes examining and extending a NeuroCore database easier, as one knows what columns

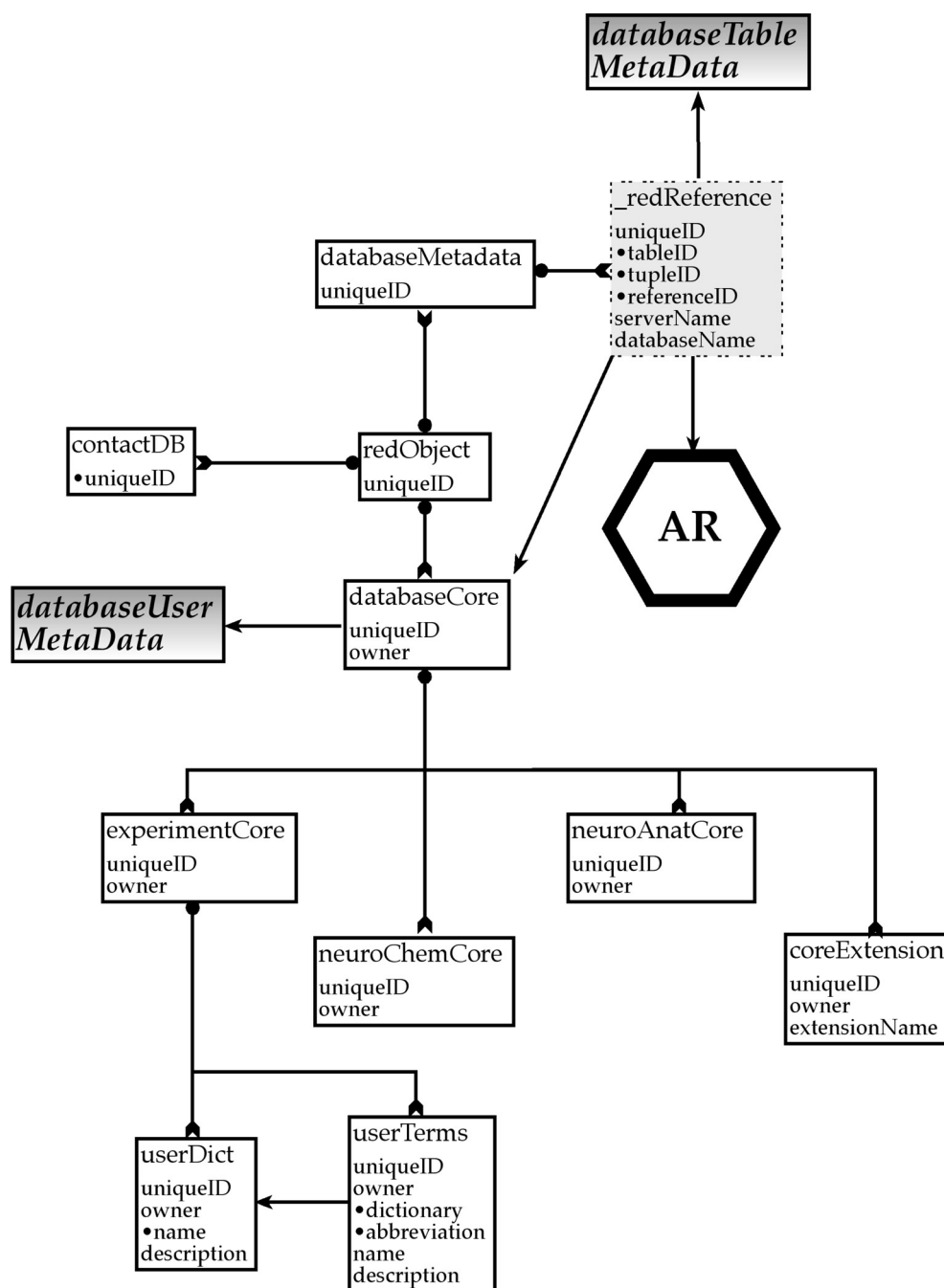


Figure 4 Top-level schema diagram for the NeuroCore database. The NeuroCore database can be subdivided into three main sections: database metadata, contact database (core) and the experiment core which contains information about the generalities of an experiment, neuroanatomical and neurochemical information as well as all the extensions added by the user.

a table contains and its data type and unit of measurement and this table's relationships to the rest of the database.

NeuroCore Schema for Contact Information

The contact portion of the core schema stores all the relevant information regarding the people and labs involved in the experiments being stored, as well as all

of the users of the database (Fig. 7). Each researcher and the laboratory(s) with which they are involved are stored within the *person* and *lab* tables, respectively. The information concerning which laboratories are associated with each researcher is stored in the *_labPerson* table. The contact information (email address, mailing address, and phone number) for each researcher is stored in the *netAddress*, *address*, and *phone* tables, respectively. Each *contact* (being a *lab* or a *person*) can have multiple

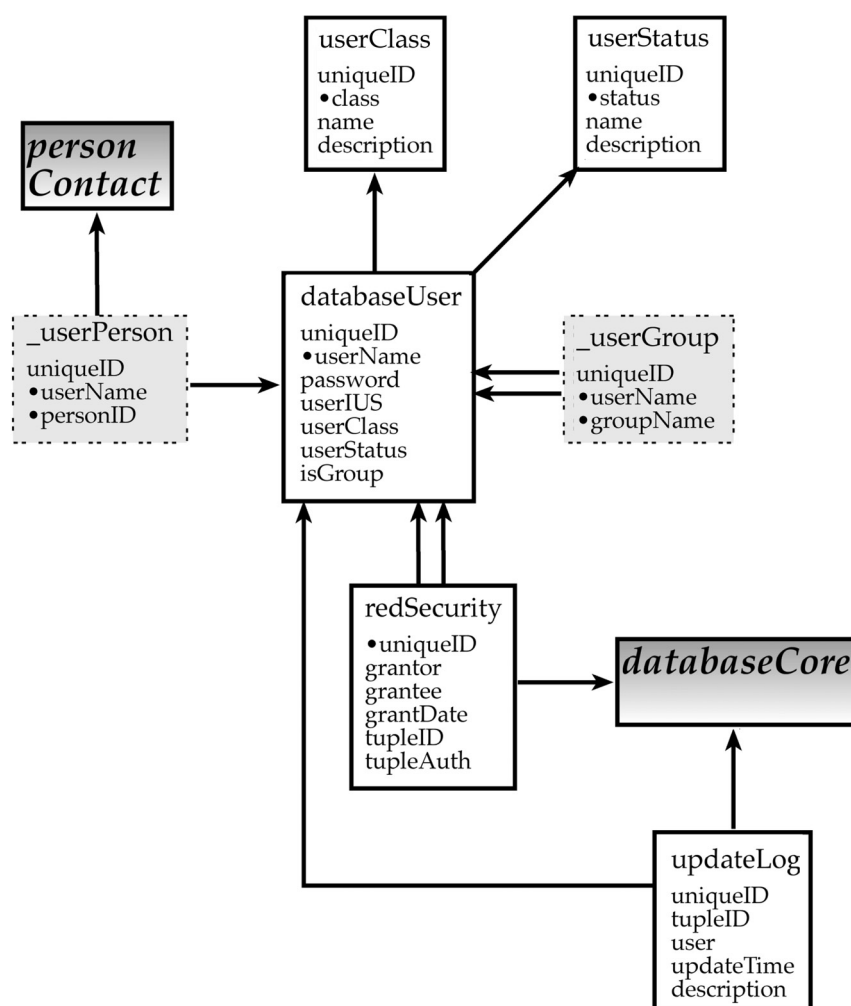


Figure 5 Schema diagram for security in the NeuroCore database. All tables depicted are sub-tables of the *databaseMetadata* table (Fig. 4). For an experimental database, each tuple in the database needs to be able to have security permissions assigned to it.

addresses, phone numbers, and electronic addresses associated with it. Relations between the contacts and tables in the *experimentCore* are handled by the *_orLogPerson*, *_hypPerson*, and *_expPerson* tables.

NeuroCore Schema for a Scientific Experiment

Each experiment that is described in a NeuroCore database consists of three distinct components:

- Hierarchical description of the experiment structure
- Description of the research subjects involved in the experiment
- Framework for describing the protocol used to generate the data

The following three sections describe each of these distinct components in more detail.

HIERARCHICAL DESCRIPTION OF EXPERIMENT

Fig. 8 depicts the experimental portion of the core database. To begin storing the definition of an experi-

ment, one first enters general descriptive information in the *experiment* table. Any researchers involved in the experiment are also entered in the *_expPerson* table. If one also wishes to describe any hypotheses being tested by this experiment, one enters similar information in the *hypothesis* and *_hypPerson* tables, respectively. All hypotheses related to a specific experiment are entered in the *_expHypothesis* table. Researchers may then also associate various keywords located in the *userTerms* table (see also Figure 4) with their experiment through the *_expKeyword* table. The complete structure of the experiment is then stored in a hierarchical fashion.

Each experiment is divided into a hierarchical collection of experimental components (*expComponent* table), which can be further subdivided into experimental segments (*expSegment* table). An experimental component, for example, would be a recording session from a typical behavioral experiment. An experimental segment, therefore, would be an individual trial that was part of that session. However, in order to accommodate neuroanatomical, neurochemical, and neurophysiological data in

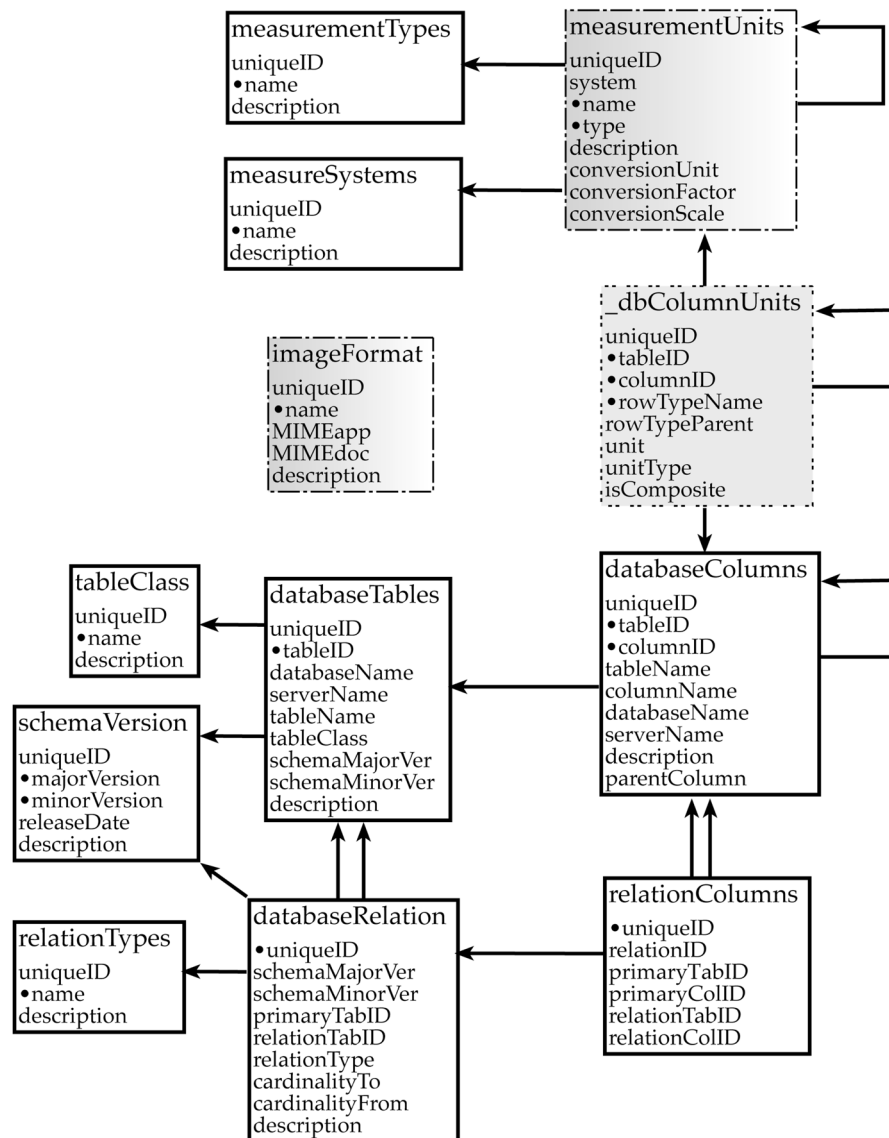


Figure 6 Schema for the metadata associated with all portions of the NeuroCore database. All tables depicted are sub-tables of the *databaseMetadata* table (Fig. 4). This information allows the complete structure and function of any NeuroCore database to be reconstructed. This includes information concerning all the tables and table columns, as well as descriptions and specifications of these columns. Also included is a description of all the relations between columns.

the same database, two parallel hierarchical descriptions have been implemented. First, a generic structure for any collection of data was implemented through the *expDataSet* and *expDataSegment* tables. These tables can store any number of unordered datasets and their respective data segments. For example, a neuroanatomical experiment may consist of a collection of stained histological sections from a specific brain where the researcher may be using multiple staining techniques (e.g., cell body stains, fiber stains, etc.). In this instance, each histological series using a specific stain can be considered an experimental data set; however, each of these data sets actually consists of a number of individual histological sections which can be considered the experimental data segments that constitute the various data sets.

Neurophysiological experiments, on the other hand, consist of an ordered collection of components where the data related to any one component may be dependent on previous components. Therefore, the *expSession* and *expTrial* tables have been implemented in order to define an ordered hierarchical description of an experiment. In addition to the standard experimental components and segments that can be defined, one can also group an ordered set of experimental segments, or trials, into experimental blocks through the *expBlock* table (each trial is related to a block through the *_blockTrial* table). Once the hierarchical description of an experiment has been defined, the experimental protocol associated with the experiment must then be defined.

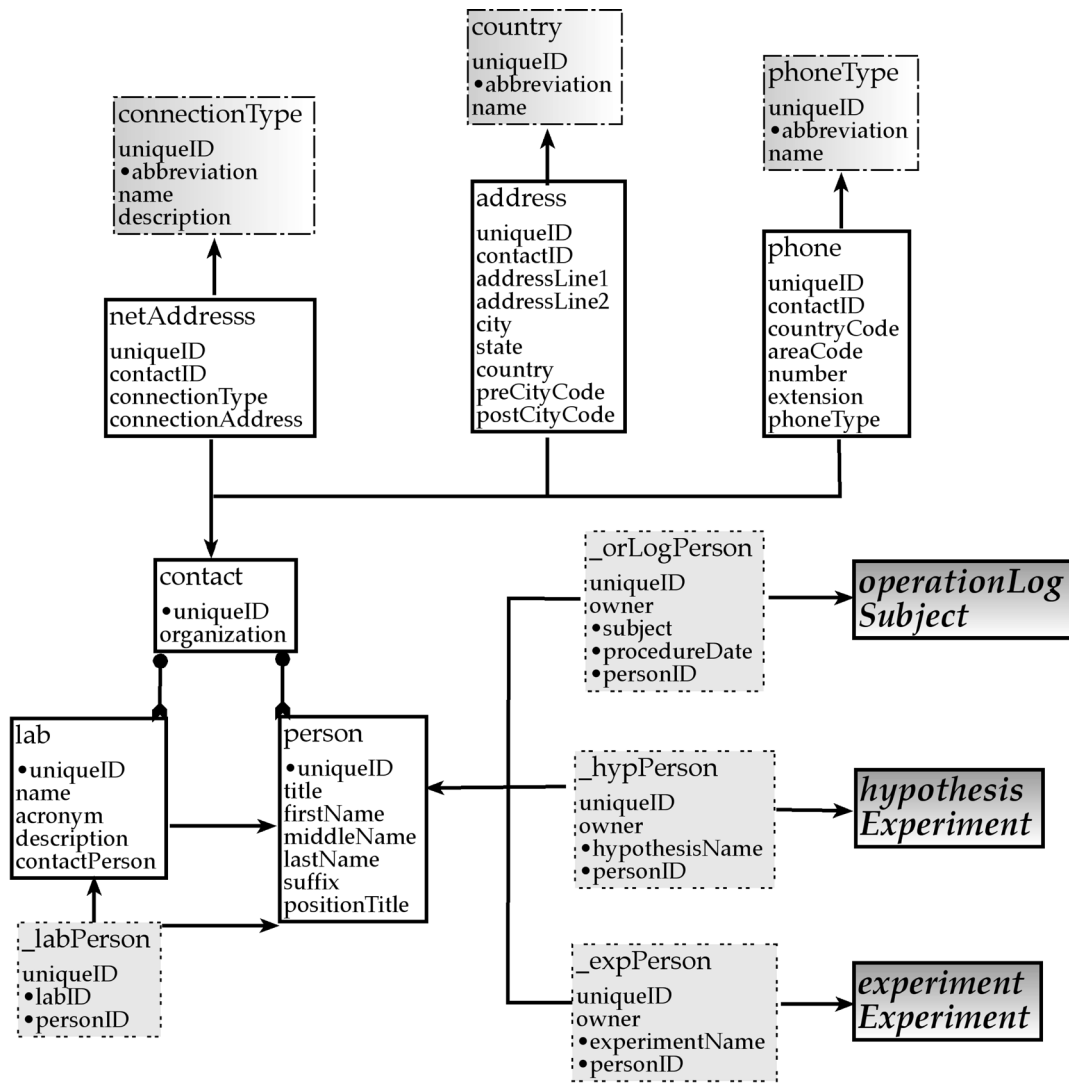


Figure 7 Schema for the contact portion of the NeuroCore core database. All tables depicted are sub-tables of the *contactDB* table (Fig. 4). All information regarding the researchers and their laboratories are stored in these tables.

FRAMEWORK FOR THE DESCRIPTION OF PROTOCOLS

Probably the most important aspect of storing an experiment in the NeuroCore database is the storage and description of the experimental protocols involved (Fig. 9). All protocols in the database are referenced through the *protocol* table and are identified by an ID and version number. In order to track these protocols, the complete ancestry (*protocolAncestry*) and history (*protDefaultHistory*) of the default version for a protocol are stored. This allows researchers to keep a history of the versions for a specific protocol with which they may be working. A protocol may also be named (*namedProtocol*) so that referencing a protocol is more natural.

Each protocol consists of a set of associated manipulations defined by the *_protocolManip* table where each manipulation is stored as a tuple in an extension table attached under the manipulation table (see Figure 14). Protocols stored in a NeuroCore database can

consist of two very different types of collections of manipulations:

1. A set of ordered manipulations where each manipulation can be followed in a sequence by another manipulation without any regard to the exact timing of the specific manipulations. The ordering of these manipulations is handled by the *_protocolManip* table, where one is able to define where each manipulation starts in the sequence (*manipStart*) and end in the sequence (*manipEnd*).
2. A set of manipulations that occur at a specific time relative to the beginning of the experimental segment. This is very common in many behavioral and neurophysiological experiments. The timing of each manipulation relative to the beginning of the experimental segment is stored in the manipulation extension tables themselves.

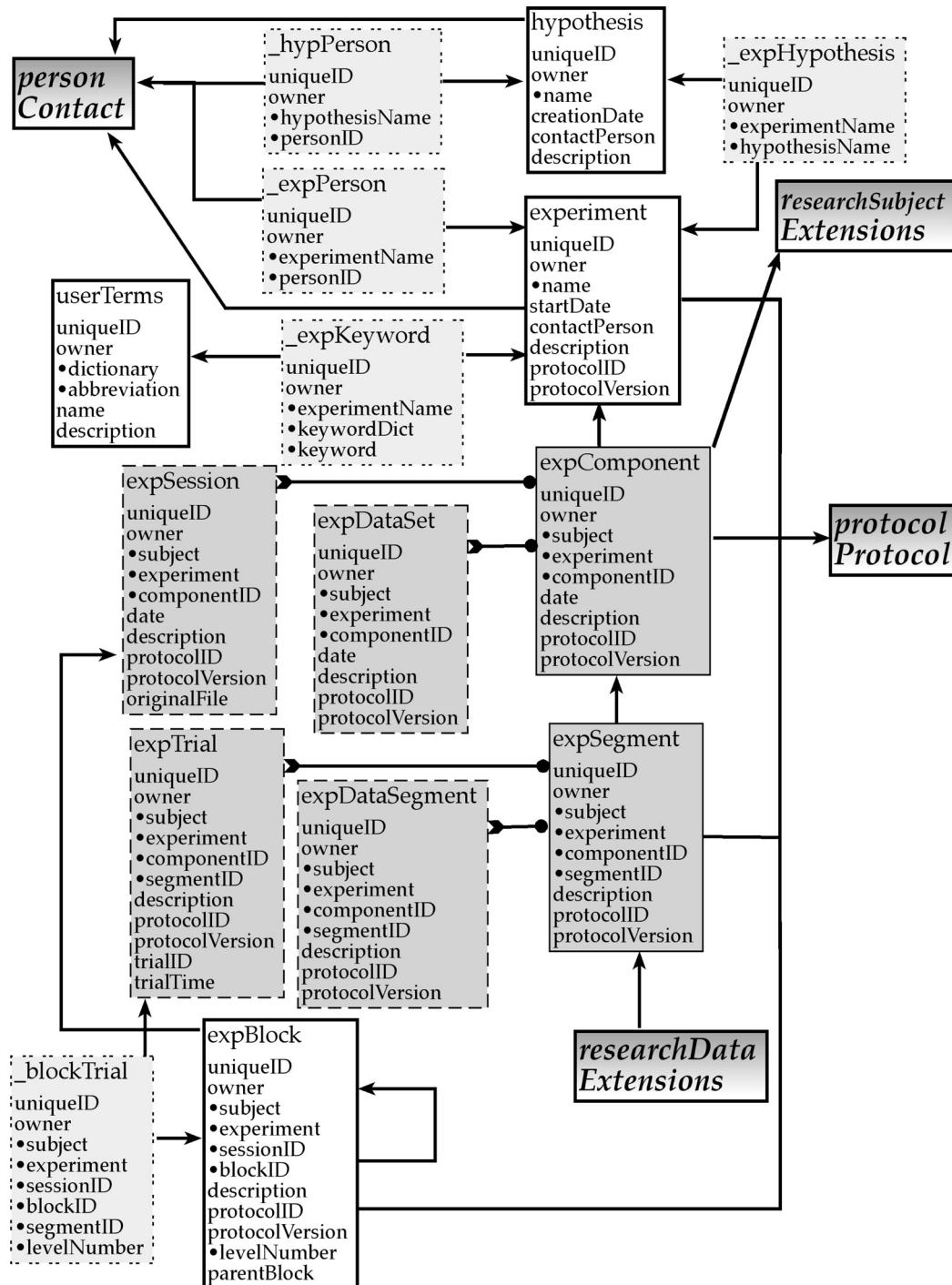


Figure 8 Experimental portion of the core schema. All tables depicted are sub-tables of the *experimentCore* table (Figure 4). Each experiment consists of a hierarchical collection of experimental components and segments which can be grouped into blocks.

The first collection of manipulations defined describes the standard process-oriented scientific data model that allows one to define a sequential collection of processes and the transitions between them. This protocol framework is the typical model used to describe experimental protocols for the genome databases developed by the Human Genome Project and other molecular biology databases (Chen and Markowitz, 1995; Lee *et al.*, 1993;

Pratt and Cohen, 1992). However, this framework is not sufficient to describe the protocols involved in neuroscientific experiments. In order to be able to store neuroscientific protocols, provisions must also be made for the second class of protocols defined above, where the set of manipulations is specifically timed. One interesting aspect of neuroscientific experiments is that they can be described by both types of

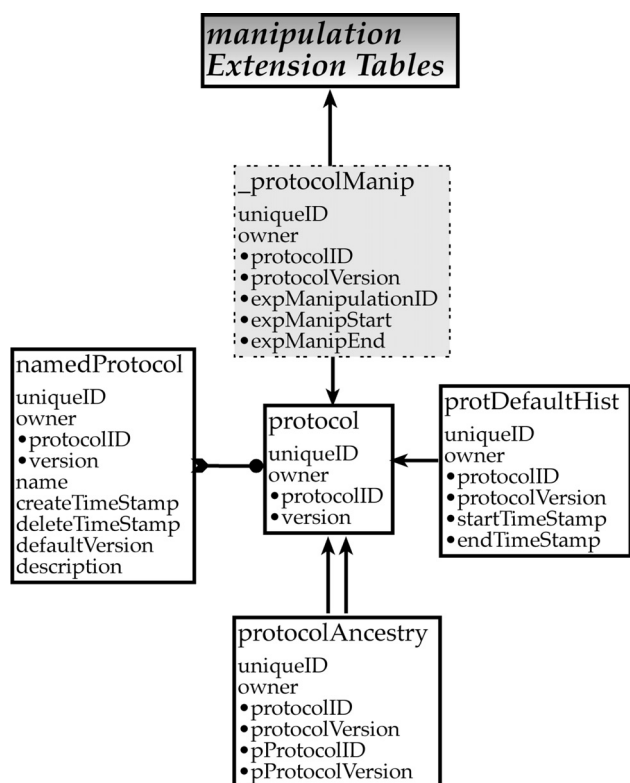


Figure 9 Section of NeuroCore database that provides the framework for the storage of protocols in the core database. All tables depicted are sub-tables of the *experimentCore* table (Figure 4). Each protocol consists of a set of manipulations that are grouped and/or ordered in time.

protocols described previously. By using the implemented framework, the NeuroCore core database is able to store both types of protocols together in a single framework.

As mentioned in the previous section, each experiment in NeuroCore is defined as a hierarchical collection of components and segments. In order to be able to store complete information regarding the experimental protocol, each entry in the experiment, component, block, or segment can have a protocol associated with it (segments are the only objects that are required to have a protocol). This allows for the researcher to store protocols at varying levels of the experimental hierarchy. This can be useful, for example, in defining the experimental conditions vs. the experimental manipulations used to collect the actual data. The experimental conditions, which may remain constant across a complete experiment, can be defined via a protocol referenced in the experiment table. This differs from the experimental manipulations, which can be referenced from the experimental component and segment tables.

SUBJECT SECTION

The last aspect of defining an experiment in a NeuroCore database is the definition of the research subjects

associated with it (Fig. 10). Each research subject in the database is stored as a tuple in an extension of the *researchSubject* table. One standard extension to this table is the *animalSubject* table that defines animal research subjects of a particular species, as defined by the *animalSpecies* table. Each subject defined in NeuroCore can be associated with one or more experiments through the *_subjExperiment* table, allowing subjects to be represented in multiple experiments. A group of experimental subjects can also be classified as belonging to a research group (e.g., control animals, sham lesion animals, lidocaine infusion animals, etc., as defined in the *resGroupType* table), which is defined through the *researchGroup* table. Finally, the procedures performed on all research subjects are tracked by the *operationLog* table, which notes the procedures performed and the researchers performing those procedures through the *orProcedure*, *orProcedureType*, and *_orLogPerson* tables.

Reference Sections

One of the important aspects in the design of the NeuroCore database was inclusion of the framework necessary to describe neuroscientific experiments. One key component of this framework is the definition of the proper terminology. Two important classes of terminology are vital for the neurosciences, namely, the definition of neurochemical and neuroanatomical terminology. These two terminology classes are outlined in the following sections.

NEUROCHEMICAL REFERENCE SECTION

Many neuroscience experiments require the use of various chemicals, drugs, molecular markers, etc. To be able to reference such information, the necessary dictionary tables needed to be added to the NeuroCore framework (Fig. 11). The neurochemical dictionary consists of a dictionary of chemicals and chemical compounds stored in the *chemicalDict* table. All chemical compounds found in the dictionary are also related to the type of compound they are (e.g., protein, neurotransmitter, etc.) and stored in the *chemTypeDict* table, which is referenced through the *_chemDictType* table.

NEUROANATOMICAL REFERENCE SECTION

The neuroanatomical section of the core database contains information pertaining to anatomical terminology and specific anatomical objects (Fig. 12), as well as anatomical atlases (Fig. 13). All anatomical terminology is stored in the sub-tables of the *anatomicalTerms* table. Provisions have been made to store terminology regarding cell groups (*cellGroups*), fiber tracts (*fiberTracts*), ventricles (*ventricles*), and superficial structures (*superficial*), such as sulci and gyri, from various anatomical dictionaries and atlases defined in the *anatomicalDict* table. Terminology regarding cell types (e.g., Purkinje cell, granule cell) and cell processes (e.g., axon, soma)

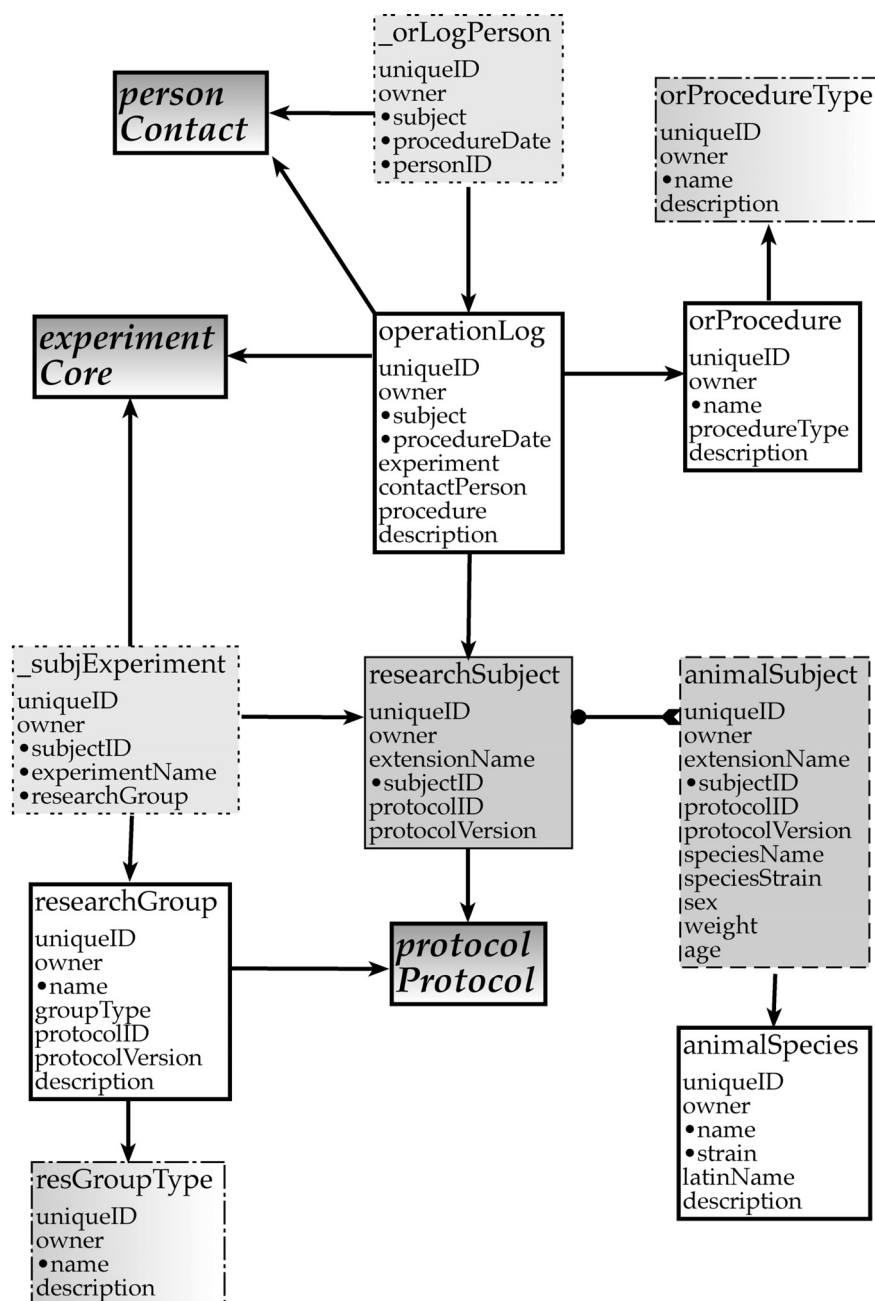


Figure 10 Schema pertaining to the research subjects being used in an experiment. All tables depicted are sub-tables of the *experimentCore* table (Figure 4). Each research subject in an experiment can belong to a specified research group.

can be defined in the *cellType* and *cellProcess* tables, respectively. Specific neuroanatomical information regarding individual cells and their anatomical locations can also be stored in the *cell* table. Each cell (i.e., single unit being recorded from, anatomical reconstruction of a single cell, etc.) also is located in a specific coordinate system (*coordSystem*).

In addition to these anatomical descriptors, researchers can store and access various data associated with published anatomical atlases. Neuroanatomical atlases (all located under the parent table *atlas*) can either be

classified as two-dimensional, defined by the *atlas2D* table, or as three-dimensional, defined by the *atlas3D* table. All atlases are defined within a specific coordinate system (*coordSystem* table). Two-dimensional atlases can further be categorized as published atlases, via the *published2DAtlas* table, and histological series, via the *histoSeries2D* table. Each published atlas contains both atlas drawings (stored in the *atlasDrawing* table) and their associated atlas photographs. Histological series, on the other hand, consist just of a sequence of section images. All section images and atlas photographs

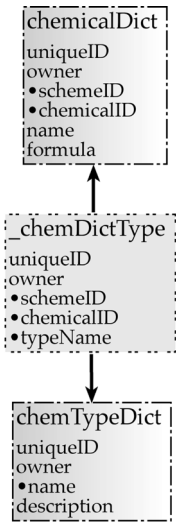


Figure 11 Neurochemistry portion of the NeuroCore database. All tables depicted are sub-tables of the *neuroChemCore* table (Figure 4). These tables define the chemical dictionary for all neurochemical referenced from within NeuroCore.

are stored in the *atlasPhoto* table. In addition, each atlas drawing has associated with it the various anatomical regions found in that particular section. These regions are defined through the *_drawingAnatTerm* table.

Extension Interface for the NeuroCore Database

The previous sections have described the essential components of the NeuroCore database that describe the framework for all neuroscientific experiments. However, this framework does not define the specific attributes regarding the research data, experimental protocols, and research subjects used in any neuroscience experiment. In order to completely store a particular experiment, the core database needs to be extended so that the details of a particular experiment can be stored. The database has been designed so that the core schema can be extended into five different areas (Fig. 14):

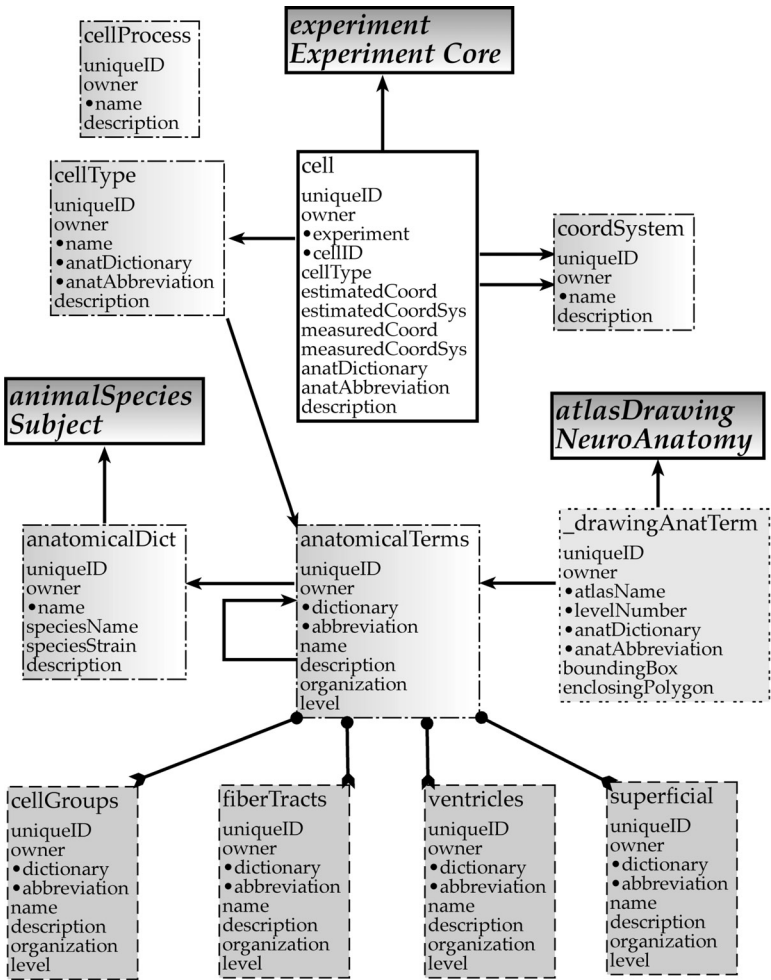


Figure 12 Neuroanatomical portion of the core database dealing with anatomical terms and objects. All tables depicted are sub-tables of the *neuroAnatCore* table (Fig. 4). All anatomical terms are stored with reference to an anatomical dictionary.

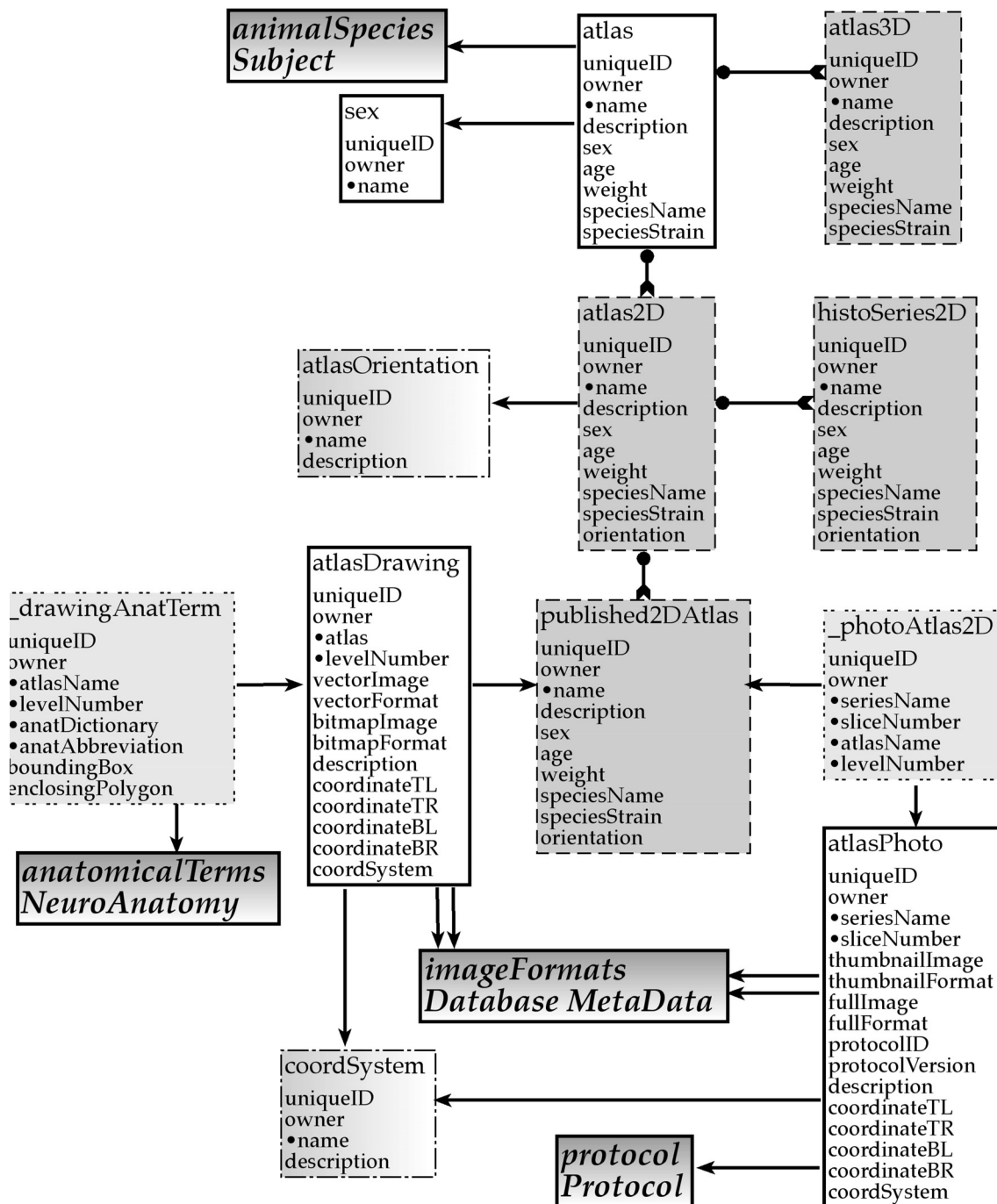


Figure 13 Neuroanatomical portion of the core database dealing with anatomical atlases. All tables depicted are sub-tables of the *neuroAnatCore* table (Fig. 4). Each atlas can reference the anatomical objects contained within it.

1. The research subject being used in an experiment through extensions to the *researchSubject* table. A standard *animalSubject* extension is already included.
2. The research data being collected in a particular experiment through extensions to the *researchData* table.
3. The actual manipulations which make up the protocols being used in a specific experiment through extensions to the *expManipulation* table.
4. The equipment being used in the experiment through extensions to the *expApparatus* table. The *_dataApparatus* and *_manipApparatus* tables relate this equipment to the specific data collected using the equipment.
5. The metadata (statistics, annotations, etc.) associated with each experiment through extensions to the *expMetadata* table. The *_dataMetaData* relates each piece of metadata to its original underlying data, thereby allowing researchers to generate new metadata on the original data sets.

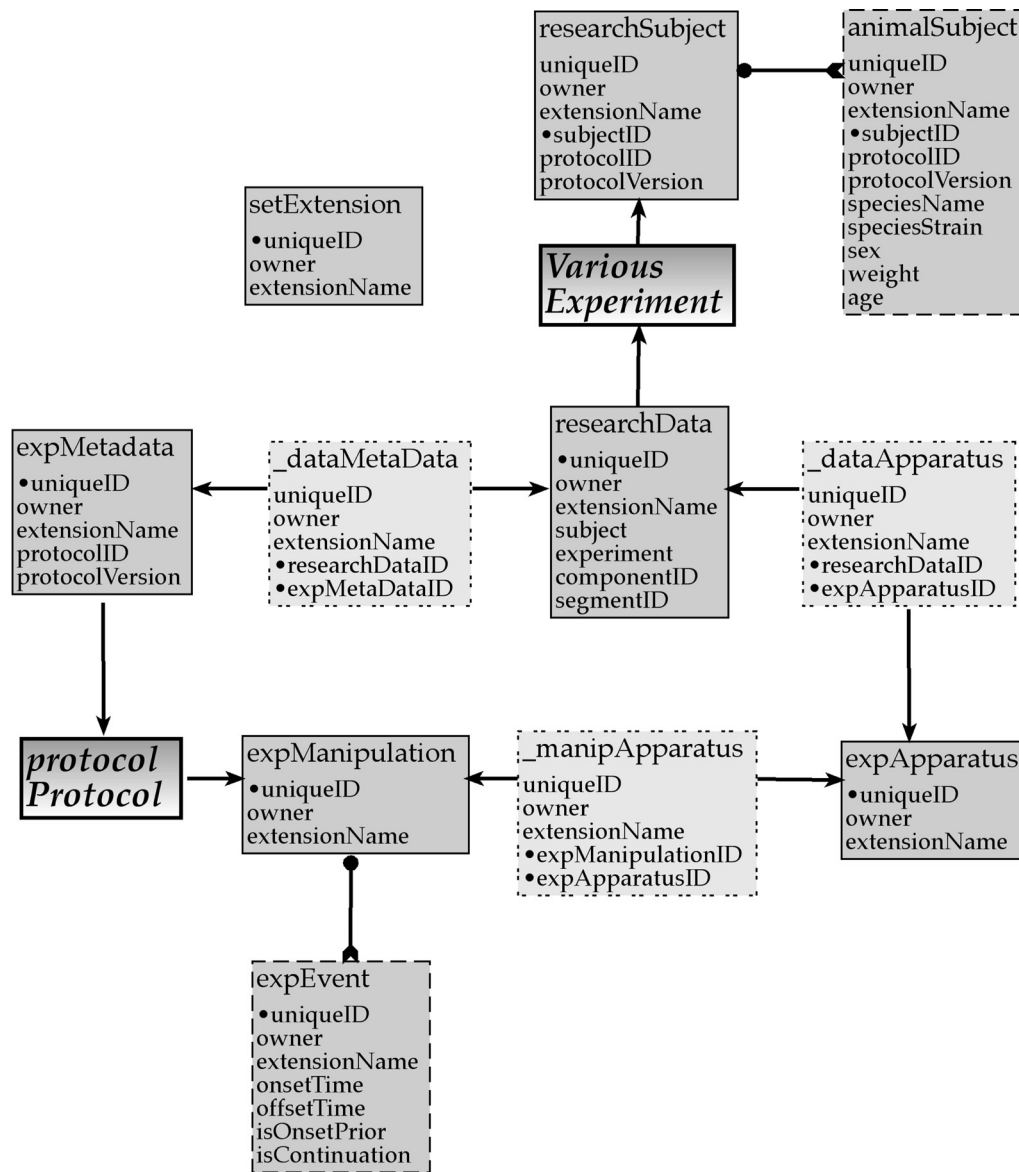


Figure 14 Areas of the core database where a user may extend the core database schema. All tables depicted are sub-tables of the *coreExtension* table (Fig. 4).

One very important standard extension included with NeuroCore is the *timeSeriesData* table. This table allows for the storage of time-series data through the use of a newly defined database object or database type. This special datatype is briefly described in the following section (a more complete description can be found in Appendix A2). A discussion of how a NeuroCore database may be extended for various experiments can be found in Chapter 3.1.

Neuroscience Time-Series Datatype

The Time-Series Datatype is constructed as a data-blade in Informix by using Informix's opaque type. The opaque data type is a fundamental data type that a user can define. (A fundamental data type is atomic. It cannot

be broken into smaller pieces, and it can serve as the building block for other data types.) When the user defines an opaque data type, the data type system of the database server is extended. The new opaque data type can be used in the same way as any built-in data type that the database server provides. To define the opaque data type to the database server, the user must provide the following information:

- A data structure that serves as the internal storage of the opaque data type
- Support functions that allow the database server to interact with this internal structure
- Optional additional routines that can be called by other support functions or by end users to operate on the opaque data type

For further information on implementation of the Time-Series datablade, please see Appendix A2.

Conclusion

The previous two chapters have provided an overview of the core functionality of the NeuroCore database. The core framework that has been introduced here is not able to store a complete neuroscientific experiment, as it lacks the extensions that allow one to fully describe the experimental preparation and the types of research data being collected. Case studies in extending the core for specific protocols were presented in Chapter 3.1. For the latest release information and

further documentation of the NeuroCore database, please visit the University of Southern California Brain Project Web site at <http://www-hbp.usc.edu/Projects/NeuroCore.htm>.

References

- Chen, I. A., and Markowitz, V. M. (1995). An overview of the object-protocol model (OPM) and the OPM data measurement tools. *Information Systems* **20**(5), 393–417.
- Lee, A. J., Rundensteiner, E. A., Thomas, S., and Lafortune, S. (1993). An information model for genome map representation and assembly. 2nd International Conference on Information and Knowledge Management.
- Pratt, J. M., and Cohen, M. (1992). A process-oriented scientific database model. *ACM SIGMOD Record*. **21**, 3.