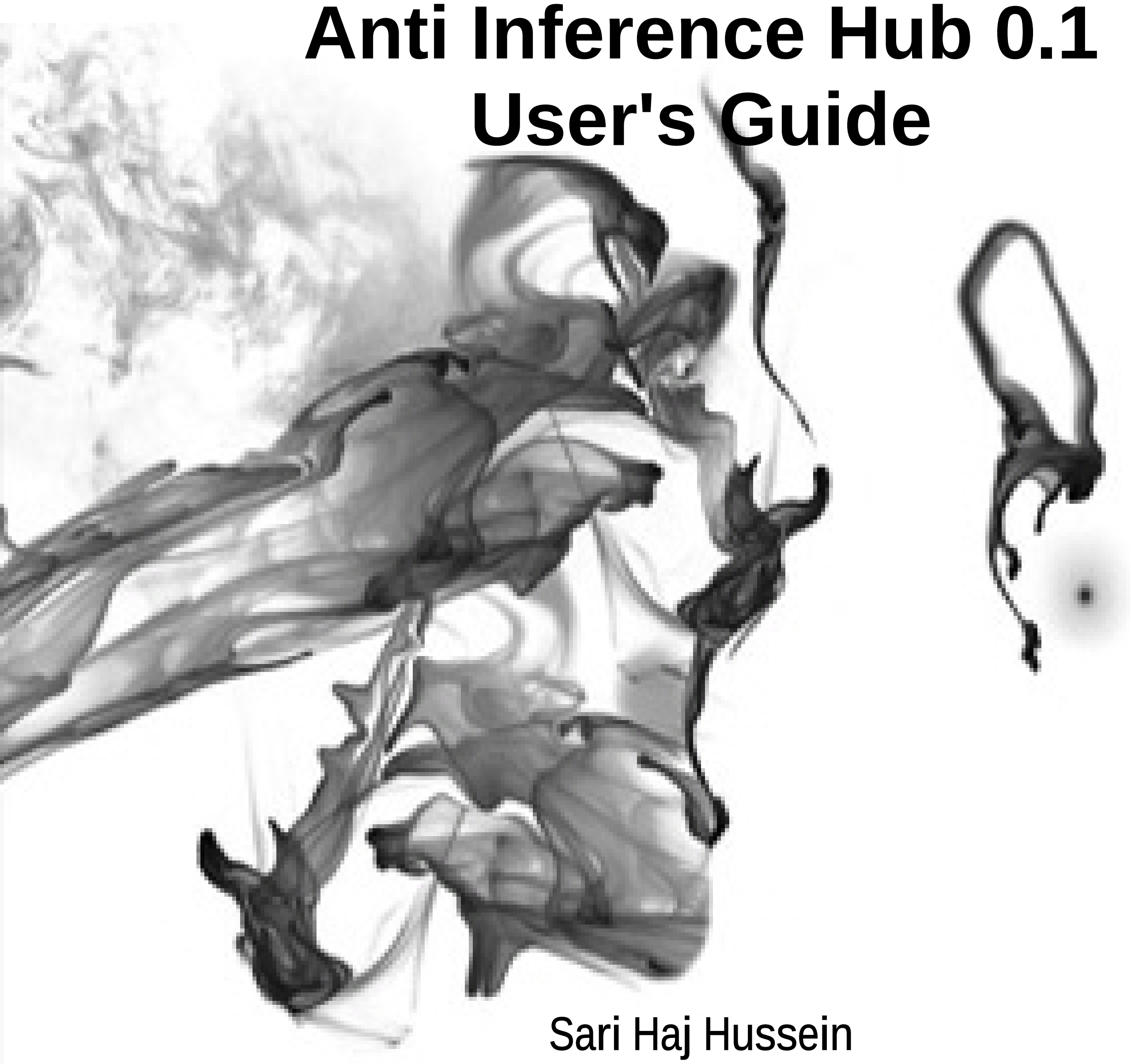
Anti Inference Hub 0.1
User's Guide



Sari Haj Hussein
Chalmers University of Technology

Table of Contents

1. Introduction 1
1.1. What is Anti Inference Hub? 1
1.2. Some Intended Purposes 1
1.3. Features 1
1.4. Open Source Software 1
1.5. What Anti Inference Hub is Not 2
1.6. Supported Operating Systems 2
1.7. Where to Get Anti Inference Hub? 2
1.8. Development and Maintenance of Anti Inference Hub 2
1.9. Reporting Problems and Getting Help 2
2. Installing Anti Inference Hub 3
2.1. Introduction 3
2.2. Obtaining the Binary Package 3
2.3. Installing the Binary Package Under Windows 3
2.4. Installing the Binary Package Under Linux 3
2.5. Uninstalling Anti Inference Hub 3
3. User Interface 4
3.1. Introduction 4
3.2. Starting Anti Inference Hub 4
3.3. The Hub Window 4
3.3.1. The Menu for the Hub Window 4
3.3.1.1. The File Menu for the Hub Window 5
3.3.1.2. The Help Menu for the Hub Window 6
3.3.2. The Activity Log Panel 6
3.3.3. The Buttons Panel for the Hub Window 7
3.4. The Hub Client Window 7
3.4.1. The Menu for the Hub Client Window 8
3.4.1.1. The File Menu for the Hub Client Window 8
3.4.1.2. The Help Menu for the Hub Client Window 9
3.4.2. The Hub Host Name Panel 10
3.4.3. The Input to Hub Panel 10
3.4.4. The Output from Hub Panel 11
3.4.5. The Buttons Panel for the Hub Client Window 11
4. Connecting with the Database 12
4.1. Introduction 12
4.2. The Setup Database Connection Box 12
5. Setting Inference Channels 13
5.1. Introduction 13
5.2. The Setup Inference Channels Box 13
5.3. The Add Channel Box 14
6. Initializing Keys 15
6.1. Introduction 15
6.2. The Initialize Keys Box 15
7. Adjusting Super Clients 17
7.1. Introduction 17
7.2. The Adjust Super Clients Box 17
7.3. The Add Super Client Box 18
8. The Inference Problem: Demystified 19

8.1. Introduction 19
8.2. Specific Inference Problems 19
8.2.1. Inference from Queries Based on Sensitive Data 19
8.2.2. Statistical Databases 19
8.2.3. Inference from Data Combined with Metadata 20
8.2.3.1. Key Integrity 20
8.2.3.2. Functional and Multivalued Dependencies 21
8.2.3.3. Value Constraints 21
8.2.3.4. Classification Constraints 21
8.3. General Characterizations of the Inference Problem 21
8.4. Tools and Techniques for Dealing with Inference Channels 22
8.5. References 22
9. Technology behind Anti Inference Hub 24
10. The Sample Database 25
10.1. Introduction 25
10.2. Location of the Sample Database 25
10.3. Creating the Sample Database 26
10.4. Our Test Environment 26
11. Anti Inference Hub in Action 27
11.1. Introduction 27
11.2. The Inference Attempt to Address 27
11.3. Thwarting the Inference Attempt 27
A. Anti Inference Hub License 33
B. Anti Inference Hub Code Disclaimer 34

Chapter 1. Introduction

1.1. What is Anti Inference Hub?

Anti Inference Hub is the first dynamic query processing engine that defends against the Inference Problem in Multilevel Databases by integrating smoothly with common DBMSs (Oracle, PostgreSQL, and MySQL), and monitoring queries submitted by end users.

1.2. Some Intended Purposes

Here are some example usages of Anti Inference Hub:

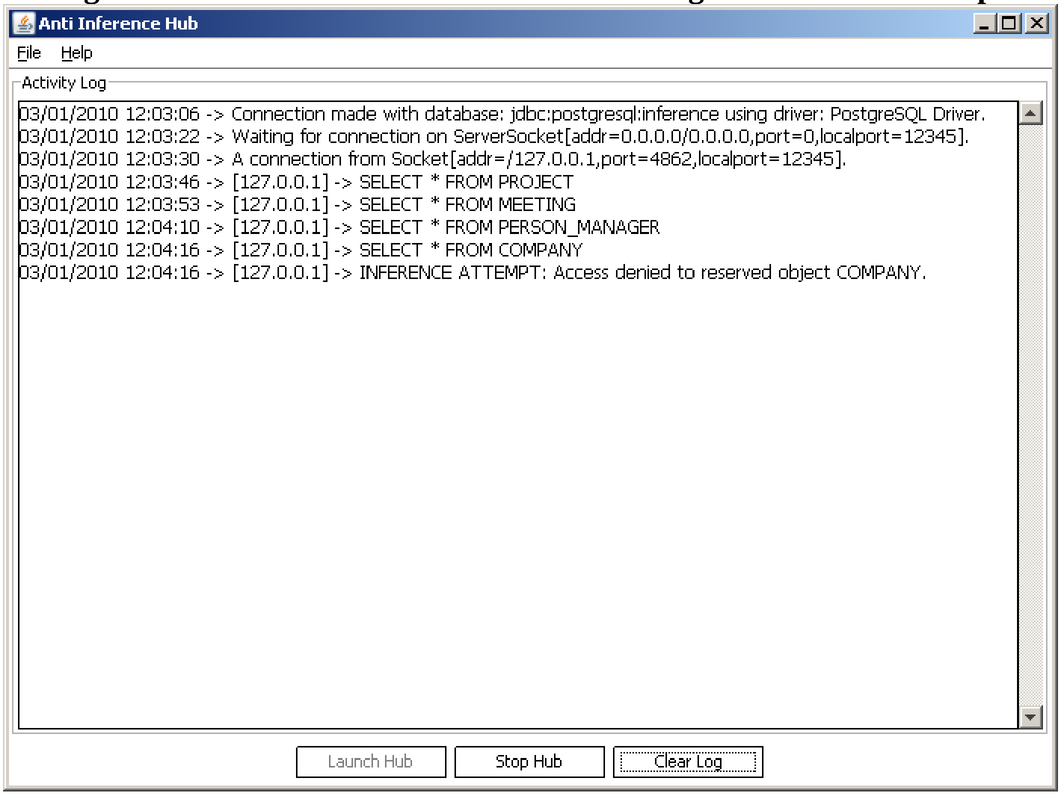
- Database designers use it to estimate the security of their database design.
- Database administrators use it to protect sensitive data stored in the database.
- Network security engineers use it to examine security problems, and identify attackers.
- People use it to learn more about the Inference Problem, and Database Security in whole.

1.3. Features

Following are some of the many features provided by Anti Inference Hub:

- Available for any operating system with Java installed.
- Integrates easily with Oracle, PotgreSQL, and MySQL.
- Fast query processing and analyzing.
- Results for safe queries are provided in a platform independent XML format.

Figure 1.1. Anti Inference Hub after Thwarting an Inference Attempt



```
03/01/2010 12:03:06 -> Connection made with database: jdbc:postgresql:inference using driver: PostgreSQL Driver.
03/01/2010 12:03:22 -> Waiting for connection on ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=12345].
03/01/2010 12:03:30 -> A connection from Socket[addr=/127.0.0.1,port=4862,localport=12345].
03/01/2010 12:03:46 -> [127.0.0.1] -> SELECT * FROM PROJECT
03/01/2010 12:03:53 -> [127.0.0.1] -> SELECT * FROM MEETING
03/01/2010 12:04:10 -> [127.0.0.1] -> SELECT * FROM PERSON_MANAGER
03/01/2010 12:04:16 -> [127.0.0.1] -> SELECT * FROM COMPANY
03/01/2010 12:04:16 -> [127.0.0.1] -> INFERENCE ATTEMPT: Access denied to reserved object COMPANY.
```

1.4. Open Source Software

Anti Inference Hub is an open source software project, and is released under the GNU General Public License (GPL). You can freely use Anti Inference Hub on any number of computers you like, without worrying about license keys or fees or such. In addition, all source code is freely available under the GPL. Because of that, it is very easy for people to contribute to the project.

1.5. What Anti Inference Hub is Not

- Anti Inference Hub is not an intrusion prevention system (IPS). It only prevents inference attacks.
- Anti Inference Hub does not interfere with network traffic.

1.6. Supported Operating Systems

Anti Inference Hub should run on any operating system with Java installed.

1.7. Where to Get Anti Inference Hub?

You can get the latest copy of Anti Inference Hub from its website: https://github.com/angyjoe/aih.

1.8. Development and Maintenance of Anti Inference Hub

Anti Inference Hub was developed by Sari Haj Hussein. Ongoing development and maintenance of Anti Inference Hub are also handled by Sari Haj Hussein. Anti Inference Hub is an open source software project, and is released under the GNU General Public License (GPL). All source code is freely available under the GPL. You are welcome to modify Anti Inference Hub to suit your own needs, and it would be appreciated if you contribute your improvements back to us. The Anti Inference Hub source code is available from Anti Inference Hub website: https://github.com/angyjoe/aih.

1.9. Reporting Problems and Getting Help

If you have problems, or need help with Anti Inference Hub, please post your questions to Anti Inference Hub mailing list at https://lists.sourceforge.net/lists/listinfo/aih-list.

Chapter 2. Installing Anti Inference Hub

2.1. Introduction

To use Anti Inference Hub, you must obtain a binary package for your operating system, then install it into its final destinations.

2.2. Obtaining the Binary Package

You can obtain the binary package from Anti Inference Hub website: https://github.com/angyjoe/aih.

2.3. Installing the Binary Package Under Windows

Unpack Anti Inference Hub binary package into a directory of choice.

2.4. Installing the Binary Package Under Linux

Same as in Windows, unpack Anti Inference Hub binary package into a directory of choice.

2.5. Uninstalling Anti Inference Hub

Remove the directory where you unpacked Anti Inference Hub binary package.

Chapter 3. User Interface

3.1. Introduction

By now you have installed Anti Inference Hub and are most likely keen to get started exploring the user interface.

3.2. Starting Anti Inference Hub

To start Anti Inference Hub under Windows, use:

- The file RunHub.bat to start the Hub.
- The file RunHubClient.bat to start the Hub Client.

And to start Anti Inference Hub under Linux, use:

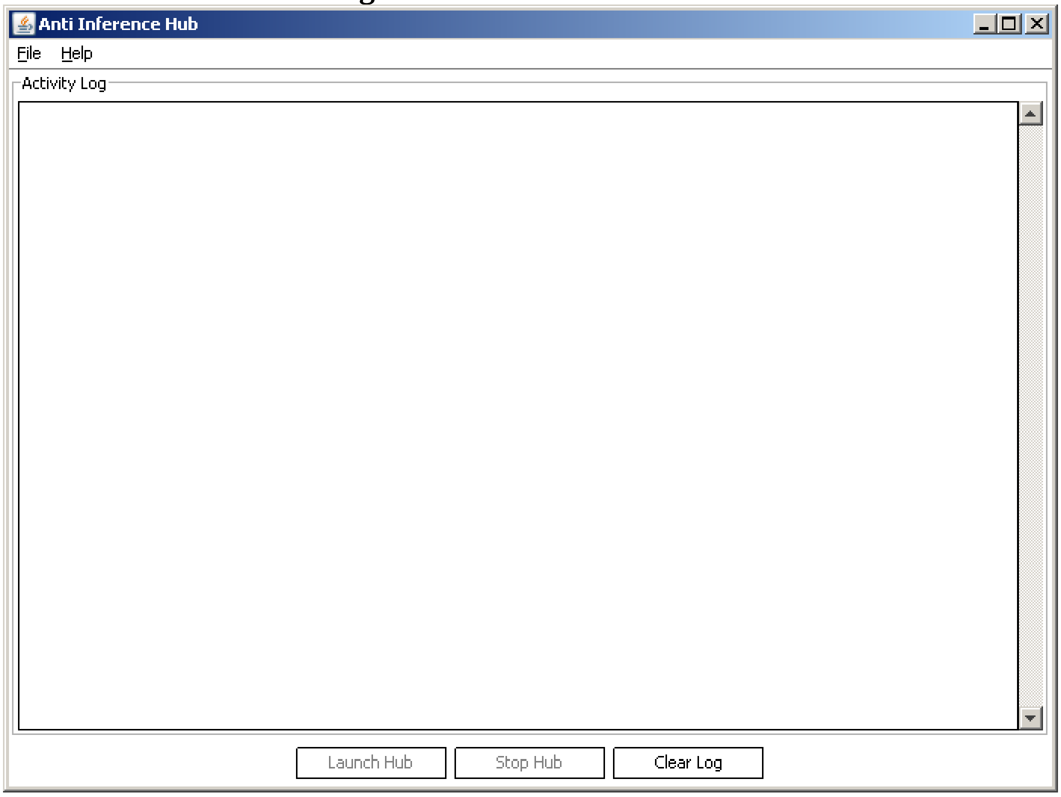
- The file RunHub.sh to start the Hub.
- The file RunHubClient.sh to start the Hub Client.

All of these files are included in Anti Inference Hub binary package.

3.3. The Hub Window

Let's look at the Hub Window.

Figure 3.1. The Hub Window



The Hub Window consists of parts that are commonly known from many other GUI programs.

1. The Menu that is used to configure the Hub.
2. The Activity Log Panel that displays a summary of the activities done by the Hub.
3. The Buttons Panel that controls whether the Hub is on or off, and clears the log.

3.3.1. The Menu for the Hub Window

The Menu for the Hub window sits on top of the window. Menu items will be grayed out if the corresponding feature is not available. For example, you cannot setup inference channels before setting up a database connection.

Figure 3.2. The Menu for the Hub Window

The Menu contains the following items:

File This menu contains items to setup database connection, inference channels, initialize keys, adjust super clients, and exit the program.

Help This menu contains items to help the user, and the usual about dialog.

3.3.1.1. The File Menu for the Hub Window

Figure 3.3. The File Menu for the Hub window

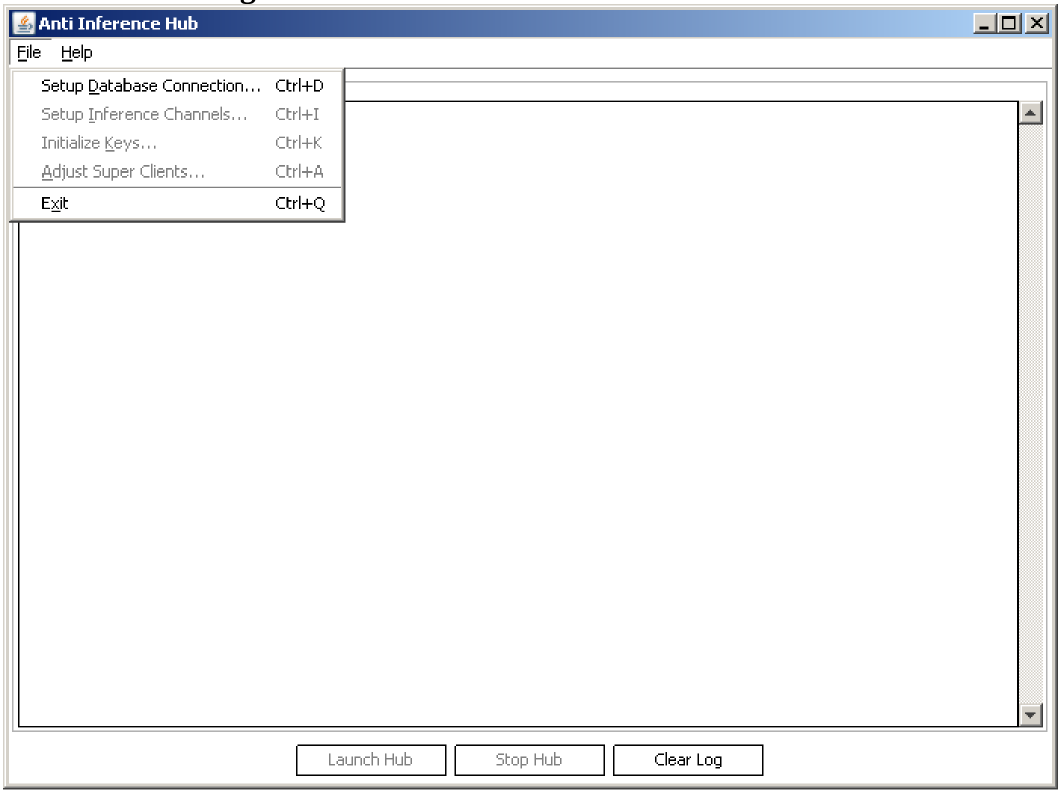

Anti Inference Hub
File
Help
Setup Database Connection… Ctrl+D
Setup Inference Channels… Ctrl+I
Initialize Keys… Ctrl+K
Adjust Super Clients… Ctrl+A
Exit Ctrl+Q
Launch Hub
Stop Hub
Clear Log

Table 3.1. Items in the File Menu for the Hub window

Menu Item	Accelerator	Description
Setup Database Connection…	Ctrl+D	This menu item brings up the Setup Database Connection Box that allows you to setup connection with supported DBMSs (Oracle, PostgreSQL, and MySQL).
Setup Inference Channels…	Ctrl+I	This menu item brings up the Setup Inference Channels Box that allows you to view, add, and remove inference channels for the database.

Initialize Keys…	Ctrl+K	This menu item brings up the Initialize Keys Box that allows you to view, initialize, and remove keys for inference channels objects.
Adjust Super Clients…	Ctrl+A	This menu item brings up the Adjust Super Clients Box that allows you to view, add, and remove super clients connecting with the Hub.
Exit	Ctrl+Q	This menu item quits the Hub.

3.3.1.2. The Help Menu for the Hub Window

Figure 3.4. The Help Menu for the Hub window

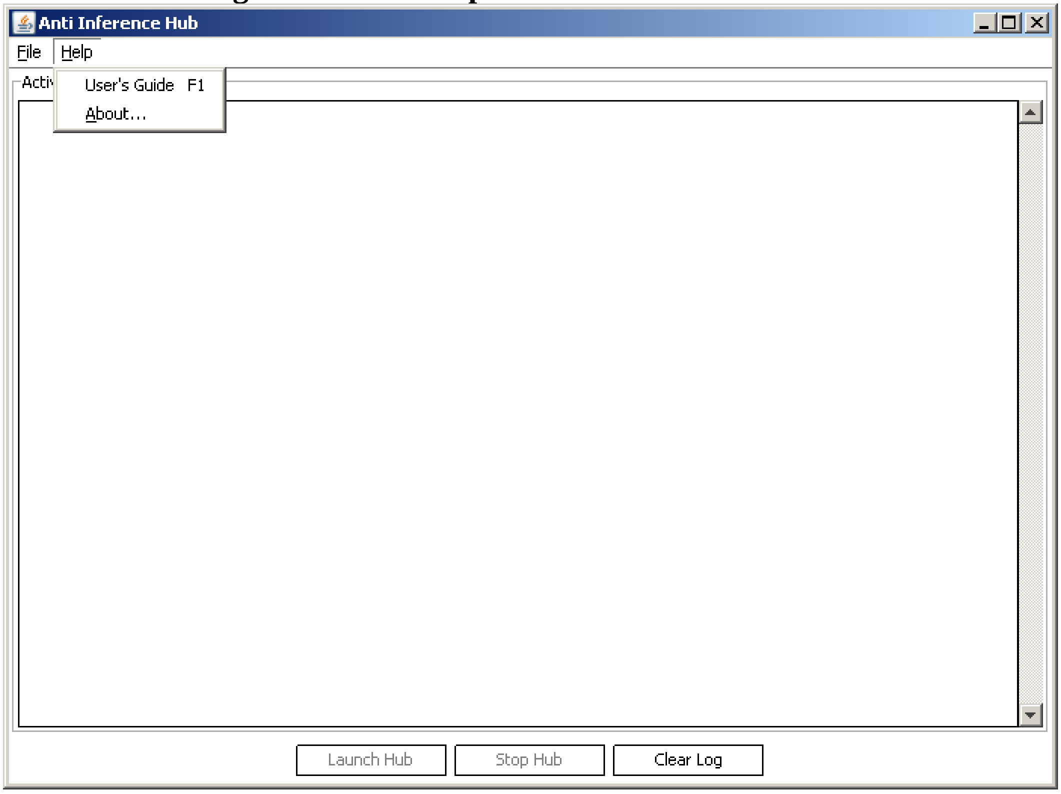


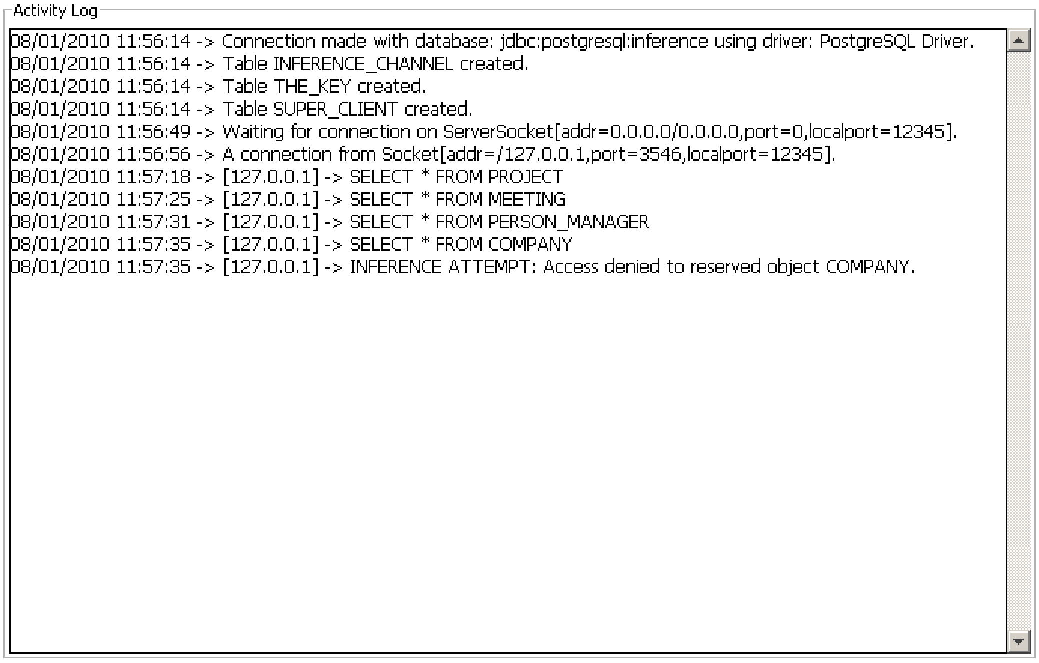
Table 3.2. Items in the Help Menu for the Hub window

Menu Item	Accelerator	Description
User's Guide	F1	This menu item opens the user's guide.
About…		This menu item brings up the About Box that provides some information on Anti Inference Hub, such as license, and third party packages used.

3.3.2. The Activity Log Panel

The Activity Log Panel displays a summary of the activities done by the Hub.

Figure 3.5. The Activity Log Panel



```
08/01/2010 11:56:14 -> Connection made with database: jdbc:postgresql:inference using driver: PostgreSQL Driver.
08/01/2010 11:56:14 -> Table INFERENCE_CHANNEL created.
08/01/2010 11:56:14 -> Table THE_KEY created.
08/01/2010 11:56:14 -> Table SUPER_CLIENT created.
08/01/2010 11:56:49 -> Waiting for connection on ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=12345].
08/01/2010 11:56:56 -> A connection from Socket[addr=/127.0.0.1,port=3546,localport=12345].
08/01/2010 11:57:18 -> [127.0.0.1] -> SELECT * FROM PROJECT
08/01/2010 11:57:25 -> [127.0.0.1] -> SELECT * FROM MEETING
08/01/2010 11:57:31 -> [127.0.0.1] -> SELECT * FROM PERSON_MANAGER
08/01/2010 11:57:35 -> [127.0.0.1] -> SELECT * FROM COMPANY
08/01/2010 11:57:35 -> [127.0.0.1] -> INFERENCE ATTEMPT: Access denied to reserved object COMPANY.
```

The activities logged in the Activity Log Panel are dated and timed, and they can be one of the following:

1. Establishing a successful connection with the DBMS.
2. Successful creation of statistical tables needed for query processing. The tables are INFERENCE_CHANNEL, THE_KEY, and SUPER_CLIENT.
3. Listening for connection requests from clients on a default port (which is 12345).
4. Queries received by the Hub, and clients who sent them.
5. Inference attempts thwarted by the Hub, and clients who initiated the attacks.
6. Inference attempts permitted by the Hub, and super clients who were allowed to infer.

3.3.3. The Buttons Panel for the Hub Window

Figure 3.6. The Buttons Panel for the Hub Window

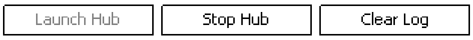


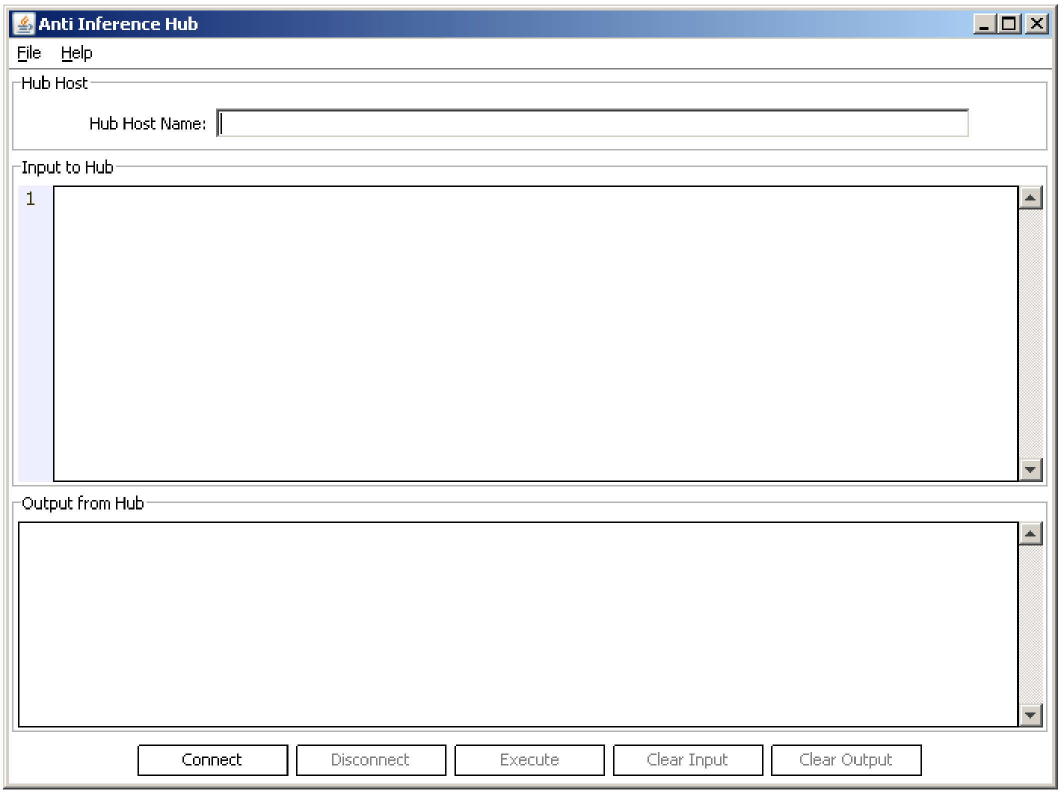
Table 3.3. Buttons in the Buttons Panel for the Hub Window

Button	Description
Launch Hub	This button is used to launch the Hub so that it listens for connection requests from clients.
Stop Hub	This button is used to stop the Hub so that no more connections are allowed.
Clear Log	This button is used to clear the content of the Activity Log Panel.

3.4. The Hub Client Window

Let's look at the Hub Client Window.

Figure 3.7. The Hub Client Window



The Hub Client Window consists of parts that are commonly known from many other GUI programs.

1. The Menu that has fewer functionalities than that of the Hub Window.
2. The Hub Host Name Panel that is used to enter the host name of the Hub the Hub Client will connect to.
3. The Input to Hub Panel that is used to enter SQL queries.
4. The Output from Hub Panel that is used to display Hub responses to SQL queries.
5. The Buttons Panel that controls whether the Hub Client is connected to the Hub or not, executes queries, and clears input/output.

3.4.1. The Menu for the Hub Client Window

The menu for the Hub Client Window sits on top of the window.

Figure 3.8. The Menu for the Hub Client Window

The Menu contains the following items:

File This menu contains an item to exit the program.
Help This menu contains items to help the user, and the usual about dialog.

3.4.1.1. The File Menu for the Hub Client Window

Figure 3.9. The File menu for the Hub Client Window

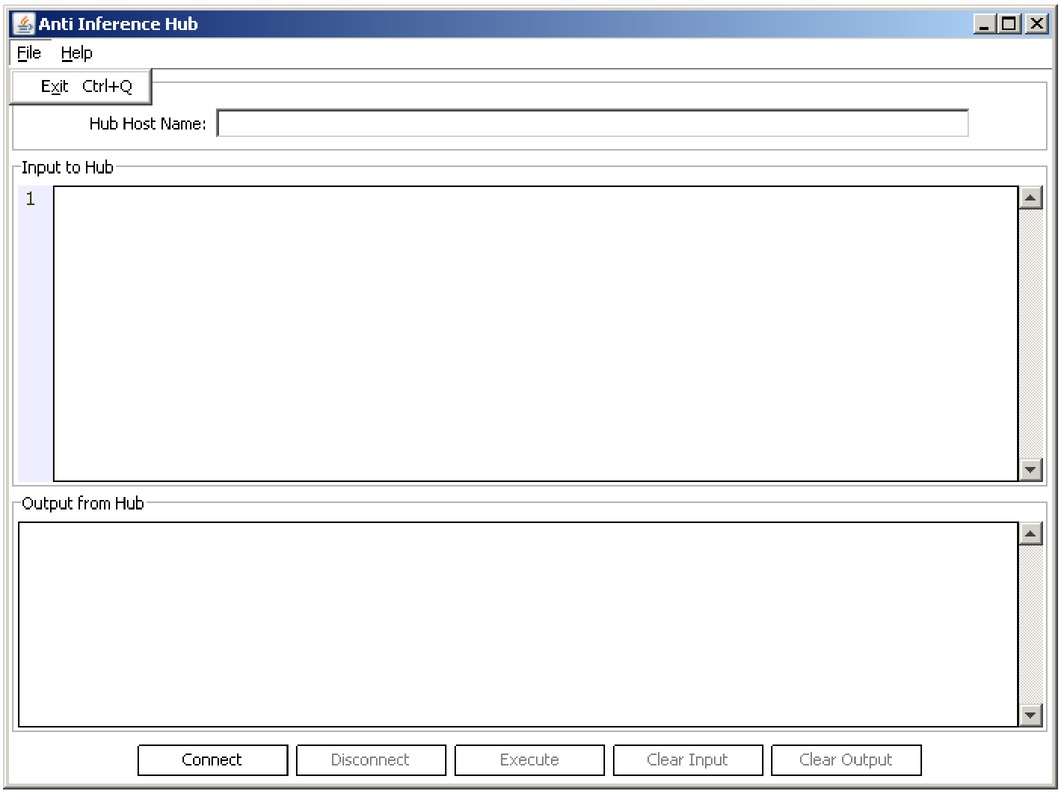


Table 3.4. Items in the File menu for the Hub Client Window

Menu Item	Accelerator	Description
Exit	Ctrl+Q	This menu item quits the Hub Client.

3.4.1.2. The Help Menu for the Hub Client Window

Figure 3.10. The Help Menu for the Hub Client window

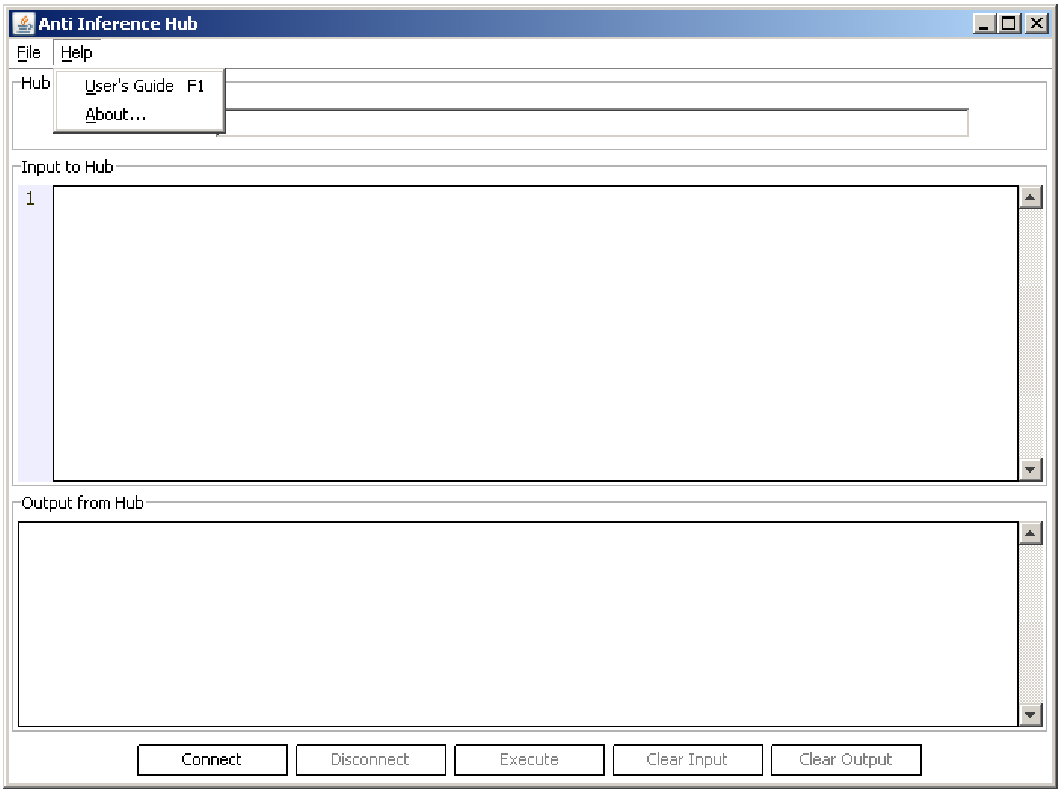


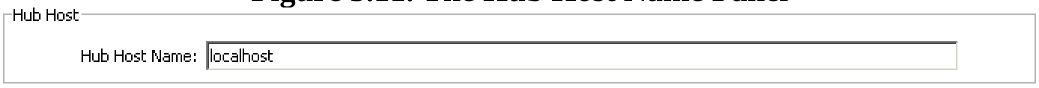
Table 3.5. Items in the Help Menu for the Hub Client window

Menu Item	Accelerator	Description
User's Guide	F1	This menu item opens the user's guide.
About…		This menu item brings up the About Box that provides some information on Anti Inference Hub, such as license, and third party packages used.

3.4.2. The Hub Host Name Panel

The Hub Host Name Panel is used to enter the host name of the Hub the Hub Client will connect to. The host name can be either a machine name or a textual representation of an IP address.

Figure 3.11. The Hub Host Name Panel

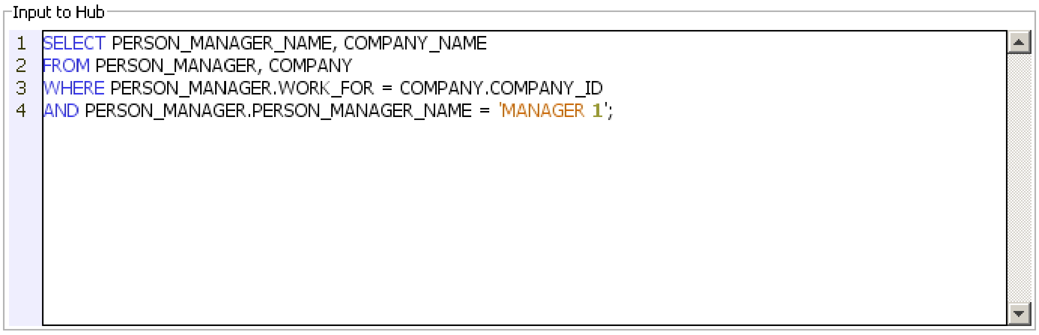

Hub Host

Hub Host Name: localhost

3.4.3. The Input to Hub Panel

The Input to Hub Panel is used to enter SQL queries. It comes with an out-of-the-box SQL syntax highlighter.

Figure 3.12. The Input to Hub Panel

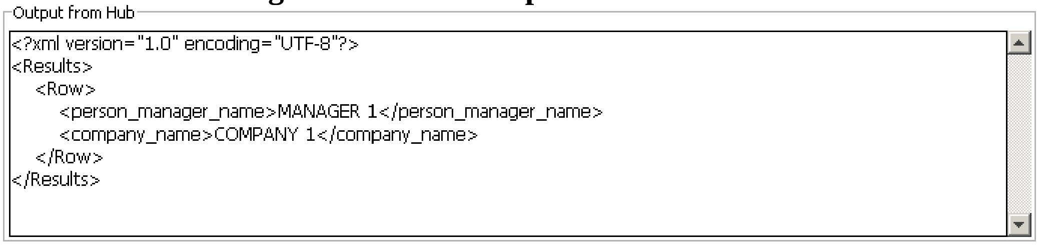


```sql
SELECT PERSON_MANAGER_NAME, COMPANY_NAME
FROM PERSON_MANAGER, COMPANY
WHERE PERSON_MANAGER.WORK_FOR = COMPANY.COMPANY_ID
AND PERSON_MANAGER.PERSON_MANAGER_NAME = 'MANAGER 1';
```

3.4.4. The Output from Hub Panel

The Output from Hub Panel is used to display Hub responses to SQL queries. Results for safe SQL queries are provided in a platform independent XML format, whereas, notifications of blocked queries, and other error messages are provided in a text format.

Figure 3.13. The Output from Hub Panel



```xml
<?xml version="1.0" encoding="UTF-8"?>
<Results>
    <Row>
        <person_manager_name>MANAGER 1</person_manager_name>
        <company_name>COMPANY 1</company_name>
    </Row>
</Results>
```

3.4.5. The Buttons Panel for the Hub Client Window

Figure 3.14. The Button Panel for the Hub Client Window

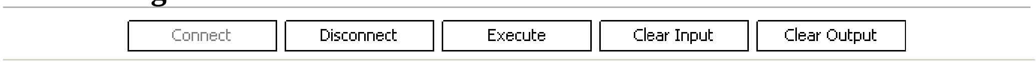


Table 3.6. Buttons in the Buttons Panel for the Hub Client Window

Button	Description
Connect	This button is used to connect the Hub Client to the Hub specified in the Hub Host Name field.
Disconnect	This button is used to disconnect the Hub Client from the Hub.
Execute	This button is used to send the content (SQL query) of the Input to Hub Panel to the Hub to be executed.
Clear Input	This button is used to clear the content of the Input to Hub Panel.
Clear Output	This button is used to clear the content of the Output from Hub Panel.

Chapter 4. Connecting with the Database

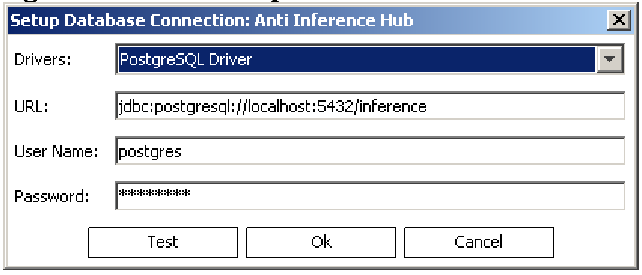
4.1. Introduction

Connecting the Hub with the database is the first step you should do to get it to work. Anti Inference Hub can protect databases designed using Oracle, PostgreSQL, and MySQL.

4.2. The Setup Database Connection Box

The Setup Database Connection Box is accessible through the Setup Database Connection menu item in the Hub Window, or the accelerator Ctrl+D.

Figure 4.1. The Setup Database Connection Box



Setup Database Connection: Anti Inference Hub

Drivers: PostgreSQL Driver

URL: jdbc:postgresql://localhost:5432/inference

User Name: postgres

Password: *********

Table 4.1. Components in the Setup Database Connection Box

Component	Description
Drivers Combo Box	This combo box is used to specify the JDBC driver the Hub will use to connect to the database. Depending on the DBMS you are using, this will be Oracle Thin Driver, or PostgreSQL Driver, or MySQL Driver.
URL Text Field	This field is used to specify the URL of the database the Hub will connect to. The URL depends on the driver specified in the Drivers Combo Box.
User Name Text Field	This field is used to specify the user name under which the Hub will connect to the database.
Password Text Field	This field is used to specify the password under which the Hub will connect to the database.
Test Button	This button is used to test the connection between the Hub and the database.
OK Button	This button is used to save the connection between the Hub and the database for ongoing usage by the Hub. It also closes the Setup Database Connection Box.
Cancel Button	This button is used to close the Setup Database Connection Box without saving any thing.

Chapter 5. Setting Inference Channels

5.1. Introduction

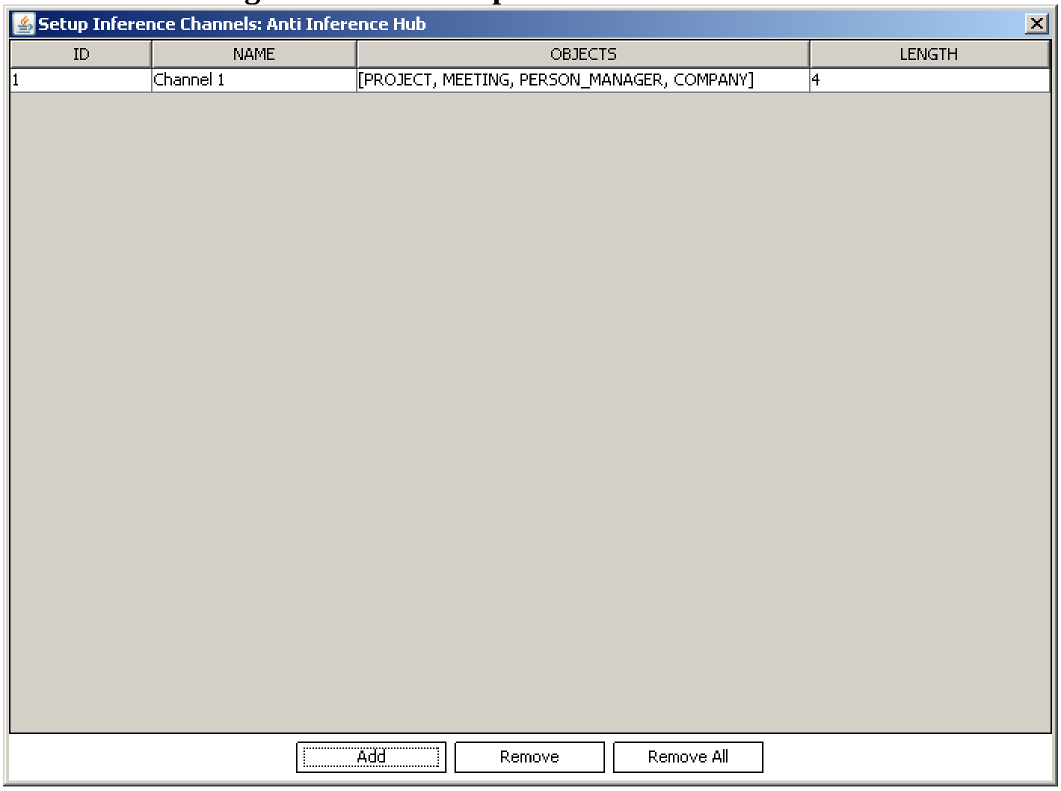
You should manually locate inference channels in the database so that the Hub can distinguish safe queries from unsafe ones. You should thoroughly consider your database design, and try to be as accurate as possible when doing this, otherwise, the Hub may block responses to safe queries!

Over time, many algorithms for locating inference channels were proposed by researches; however, all of them tend to generate inaccurate and unsatisfying results, therefore, Anti Inference Hub does not implement any of these algorithms, rather, it transfers the task of locating inference channels to the database designer (or the database security specialist). When a sound algorithm for locating inference channels has been proposed in the scientific community, we will make it available in Anti Inference Hub. For further information about this issue, please refer to Section 8.4. Tools and Techniques for Dealing with Inference Channels.

5.2. The Setup Inference Channels Box

The Setup Inference Channels Box is accessible through the Setup Inference Channels menu item in the Hub Window, or the accelerator Ctrl+I.

Figure 5.1. The Setup Inference Channels Box



ID	NAME	OBJECTS	LENGTH
1	Channel 1	[PROJECT, MEETING, PERSON_MANAGER, COMPANY]	4

Table 5.1. Components in the Setup Inference Channels Box

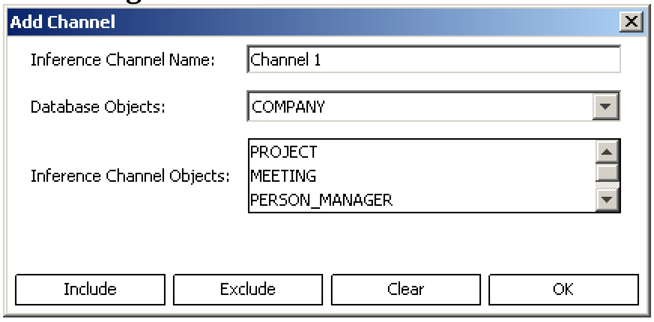
Component	Description
Inference Channels Table	This table lists currently located inference channels in the database. Every inference channel is characterized by its ID, name, objects contained in it, and its length.
Add Button	This button is used to locate a new inference channel in the database. When clicked, it brings

	up the Add Channel Box.
Remove Button	This button is used to remove the selected inference channel in the Inference Channels Table.
Remove All Button	This button is used to remove all inference channels in the Inference Channels Table whether selected or not.

5.3. The Add Channel Box

The Add Channel Box is used to locate a new inference channel in the database.

Figure 5.2. The Add Channel Box



Inference Channel Name: Channel 1

Database Objects: COMPANY

Inference Channel Objects: PROJECT
MEETING
PERSON_MANAGER

Table 5.2. Components in the Add Channel Box

Component	Description
Inference Channel Name Text Field	This field is used to specify the name of the new inference channel. A name must be specified.
Database Objects Combo Box	This combo box is used to specify a database object to include in the new inference channel. One object at least must be included.
Inference Channel Objects List	This list displays the current selection of database objects that will be included in the new inference channel.
Include Button	This button is used to include the selected database object from Database Objects Combo Box in the Inference Channel Objects List.
Exclude Button	This button is used to exclude the selected database objects in the Inference Channel Objects List. Note that more than one database object can be selected in the Inference Channel Objects List.
Clear Button	This button is used to clear the content of the Inference Channel Objects List.
OK Button	This button is used to save the new inference channel in the database. It also closes the Add Channel Box.

Chapter 6. Initializing Keys

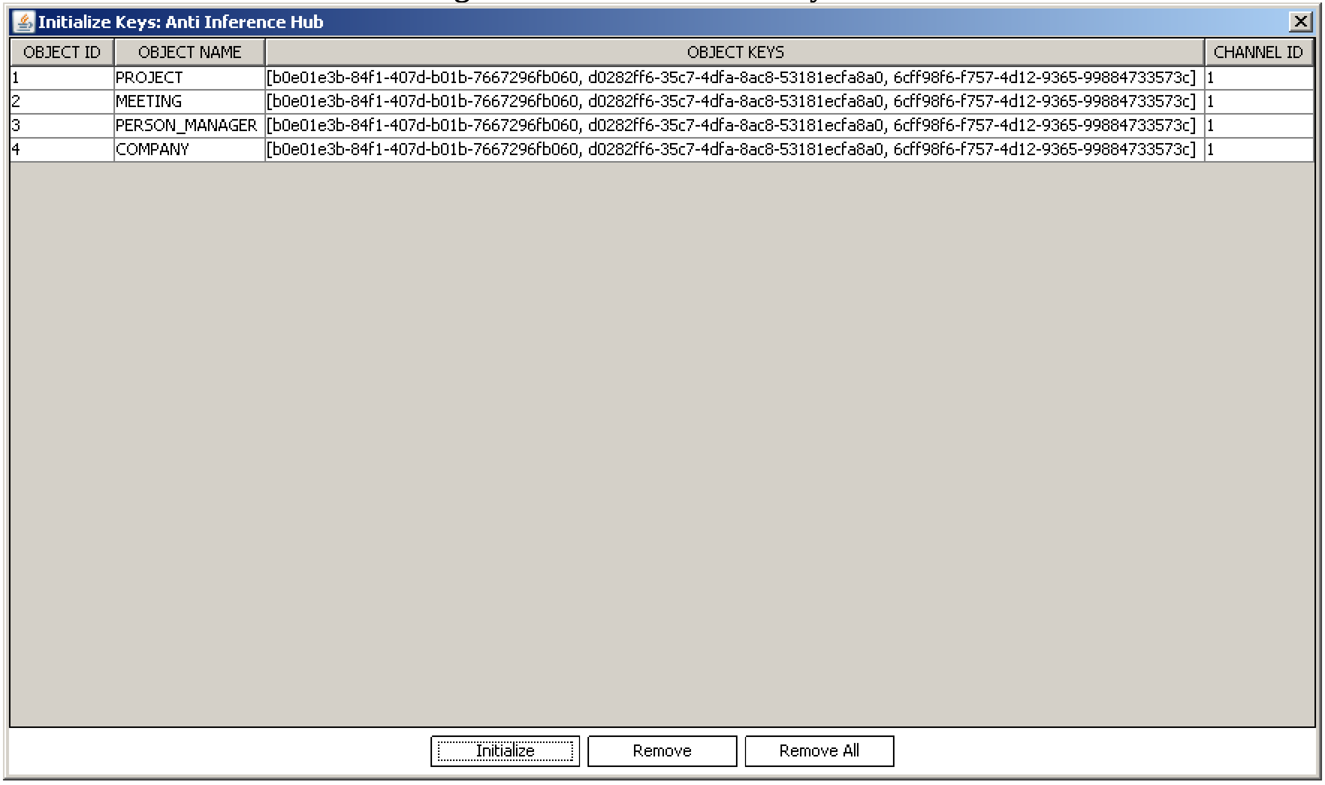
6.1. Introduction

Initializing keys for inference channels objects is an automatic process that you should consider right after locating inference channels in the database. For further information about the purpose of the keys, please refer to Chapter 9. Technology behind Anti Inference Hub.

6.2. The Initialize Keys Box

The Initialize Keys Box is accessible through the Initialize Keys menu item in the Hub Window, or the accelerator Ctrl+K.

Figure 6.1. The Initialize Keys Box



OBJECT ID	OBJECT NAME	OBJECT KEYS	CHANNEL ID
1	PROJECT	[b0e01e3b-84f1-407d-b01b-7667296fb060, d0282ff6-35c7-4dfa-8ac8-53181ecfa8a0, 6cff98f6-f757-4d12-9365-99884733573c]	1
2	MEETING	[b0e01e3b-84f1-407d-b01b-7667296fb060, d0282ff6-35c7-4dfa-8ac8-53181ecfa8a0, 6cff98f6-f757-4d12-9365-99884733573c]	1
3	PERSON_MANAGER	[b0e01e3b-84f1-407d-b01b-7667296fb060, d0282ff6-35c7-4dfa-8ac8-53181ecfa8a0, 6cff98f6-f757-4d12-9365-99884733573c]	1
4	COMPANY	[b0e01e3b-84f1-407d-b01b-7667296fb060, d0282ff6-35c7-4dfa-8ac8-53181ecfa8a0, 6cff98f6-f757-4d12-9365-99884733573c]	1

Table 6.1. Components in the Initialize Keys Box

Component	Description
Keys Table	This table lists all objects in all inference channels along with their keys. Information in this table is characterized by object ID, object name, object keys, and inference channel ID the object belongs to.
Initialize Button	This button is used to initialize keys for all objects in all inference channels in the database.
Remove Button	This button is used to remove the selected association between an object and its keys in the Keys Table. Note though that objects are not removed from inference channels.
Remove All Button	This button is used to remove all associations between objects and their keys in the Keys

	Table whether selected or not. Note again that objects are not removed from inference channels.

Chapter 7. Adjusting Super Clients

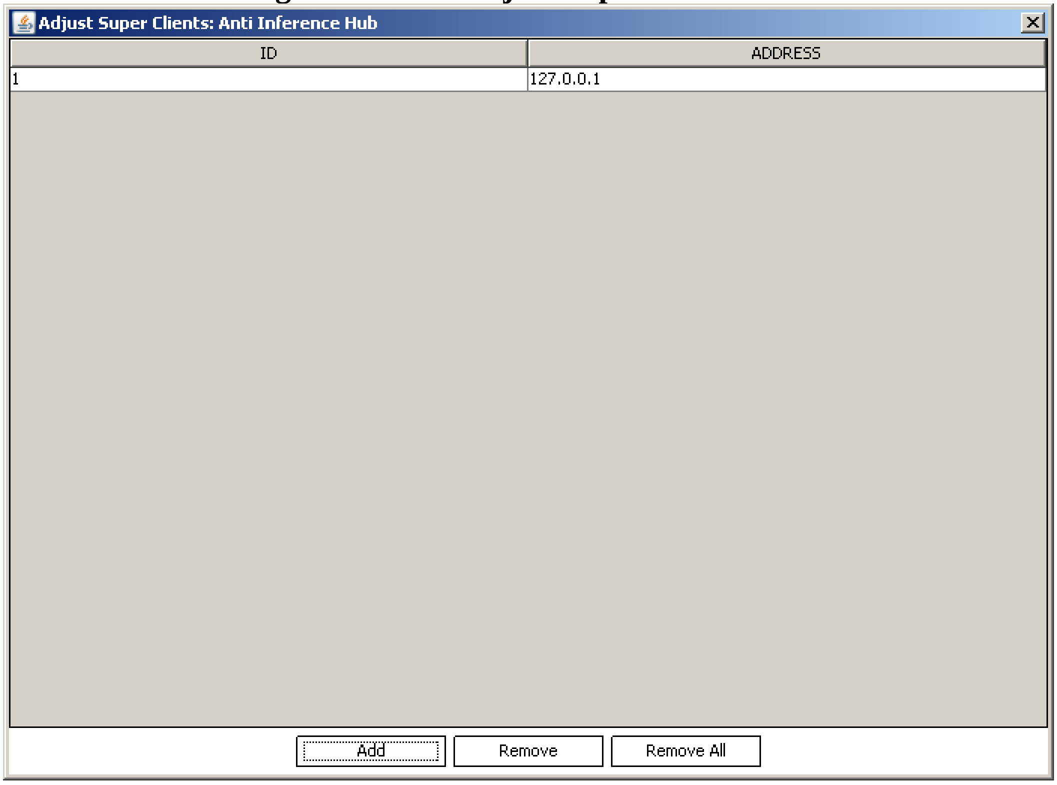
7.1. Introduction

Adjusting super clients is an optional step that you may consider before launching the Hub. A super client is a Hub Client that is allowed to infer, or in other words, a super client is a client that is considered database-friendly by the Hub! There is a very sound rationale behind allowing super clients in Anti Inference Hub. Read more about this by referring to Chapter 9. Technology Behind Anti Inference Hub.

7.2. The Adjust Super Clients Box

The Adjust Super Clients Box is accessible through the Adjust Super Clients menu item in the Hub Window, or the accelerator Ctrl+A.

Figure 7.1. The Adjust Super Clients Box



ID	ADDRESS
1	127.0.0.1

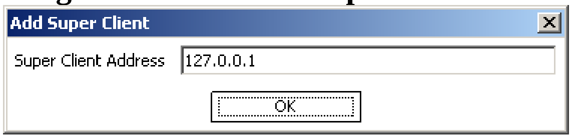
Table 7.1. Components in the Adjust Super Clients Box

Component	Description
Super Clients Table	This table lists all super clients. Every super client is characterized by its ID, and the textual representation of its IP address.
Add Button	This button is used to adjust a new super client. When clicked, it brings up the Add Super Client Box.
Remove Button	This button is used to remove the selected super client from the Super Clients Table.
Remove All Button	This button is used to remove all super clients in the Super Clients Table whether selected or not.

7.3. The Add Super Client Box

The Add Super Client Box is used to adjust a new super client.

Figure 7.2. The Add Super Client Box



Super Client Address: 127.0.0.1

Table 7.2. Components in the Add Super Client Box

Component	Description
Super Client Address Field	This field is used to specify the textual representation of the IP address of the new super client. This field has an installed IPv4 address filter, therefore, any attempt by the user to fill it with an invalid IP address will cause the field to revert to its default value which is 127.0.0.1.
OK Button	This button is used to save the new super client. It also closes the Add Super Client Box.

Chapter 8. The Inference Problem: Demystified

8.1. Introduction

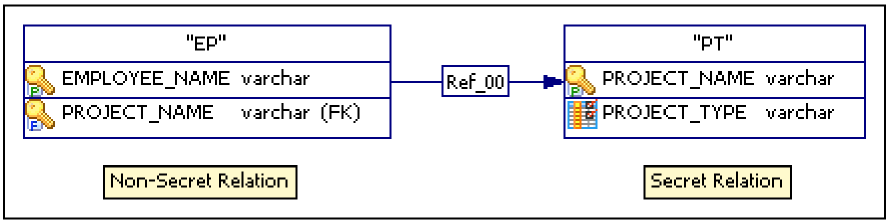
An inference channel in a database is a means by which one can infer data classified at a high level from data classified at a low level. The inference problem is the problem of detecting and removing inference channels. It is clear that inference problems are of vital interest to the designers and users of secure databases. Yet so far inference problems in multilevel databases have not been studied very deeply. This is partly due to the difficulty of the problem. We have no way of controlling what data is learned outside of the database, and our abilities to predict it will be limited. Thus even the best model can give us only an approximate idea of how safe a database is from illegal inferences. This fact should always be kept in mind when dealing with inference problems.

8.2. Specific Inference Problems

In this section we describe a number of inference channels that have been discovered in the course of database security research.

8.2.1. Inference from Queries Based on Sensitive Data

Suppose we have the following schema in a database:



And suppose a low user makes the following SQL query:

```
SELECT EP.EMPLOYEE-NAME
FROM EP, PT
WHERE EP.PROJECT_NAME = PT.PROJECT_NAME;
```

This query is evaluated by taking the natural join of the two relations EP and PT along PROJECT_NAME, and then projecting along EMPLOYEE_NAME, therefore, we have an inference channel, even though only the non-secret data (employee names) is being returned to the low user. The following SQL query represents a similar problem:

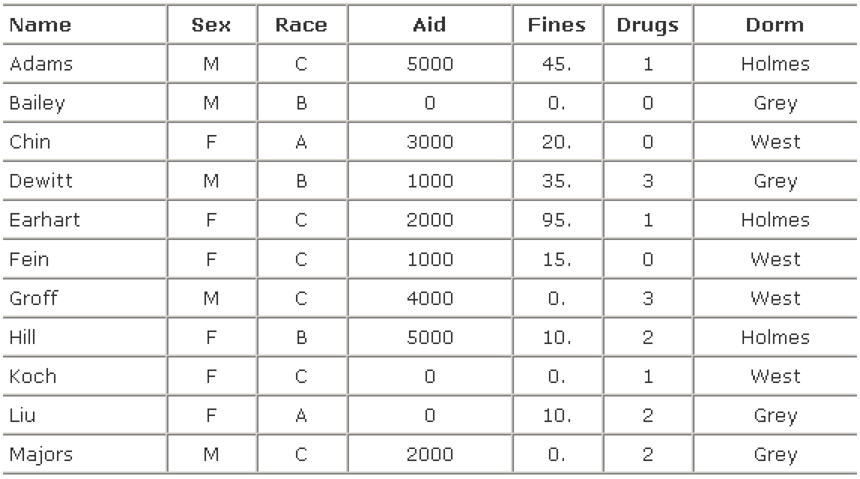
```
SELECT EP.EMPLOYEE_NAME
FROM EP, PT
WHERE EP.PROJECT_NAME = PT.PROJECT_NAME
AND PT.PROJECT-TYPE = 'SDI';
```

If we examine the above SQL queries carefully, we quickly observe that even though the data returned to the user has a low classification, the data that is required to evaluate the query has a higher classification. Thus, the inference channels arise from the fact that these queries are conditioned on data that are supposed to be invisible to the user.

Inferences of this type are easy to eliminate. The system can either modify the user query such that the query involves only the authorized data or simply abort the query.

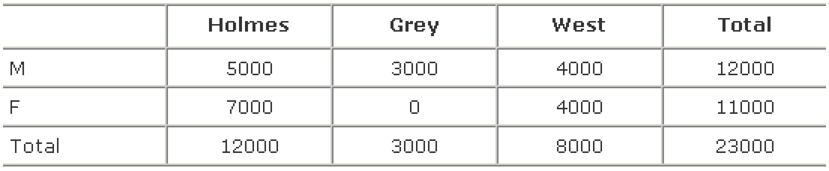
8.2.2. Statistical Databases

The problem of statistical database security is the problem of answering queries about statistics on data, such as mean, median, standard deviation, and so on, without releasing the data itself. The threat against statistical database security is that an attacker may be able to find out statistical information about individuals by posing queries on aggregate statistics over a period of time and performing arithmetic operations on the answers received, using his own information about the size and nature of the sets of individuals involved. For example, suppose the following sample database:



Name	Sex	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45.	1	Holmes
Bailey	M	B	0	0.	0	Grey
Chin	F	A	3000	20.	0	West
Dewitt	M	B	1000	35.	3	Grey
Earhart	F	C	2000	95.	1	Holmes
Fein	F	C	1000	15.	0	West
Groff	M	C	4000	0.	3	West
Hill	F	B	5000	10.	2	Holmes
Koch	F	C	0	0.	1	West
Liu	F	A	0	10.	2	Grey
Majors	M	C	2000	0.	2	Grey

It might seem safe to report student aid total by sex and dorm. Such a report is shown in the following table:



	Holmes	Grey	West	Total
M	5000	3000	4000	12000
F	7000	0	4000	11000
Total	12000	3000	8000	23000

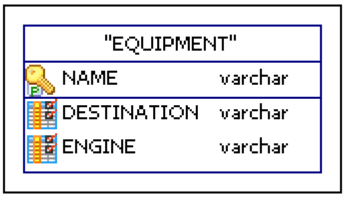
This seemingly innocent report reveals that no female living in Grey is receiving financial aid. Thus, we can infer that any female living in Grey (such as Liu) is certainly not receiving financial aid.

8.2.3. Inference from Data Combined with Metadata

We will consider some kinds of metadata used in databases, and show how it can be used to assist in making inferences.

8.2.3.1. Key Integrity

Suppose we have the following schema in a database:

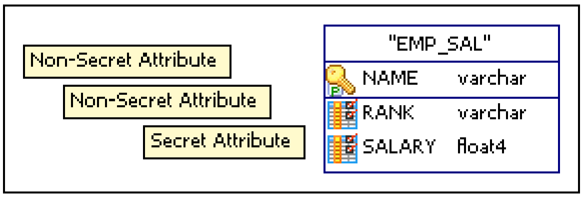


"EQUIPMENT"	
NAME	varchar
DESTINATION	varchar
ENGINE	varchar

And the following tuple in the relation (Wombat, Persian Gulf, Nuclear). Suppose a low user wants to insert the tuple (Wombat, Norfolk, Nuclear). If we choose to preserve key integrity, we must either delete the secret tuple or reject this insertion. We have an integrity problem if we delete the secret tuple (since it can cause data inserted by a high user to be deleted by a low user). If we reject the insertion, then the low user can derive an inference. It turns out that this problem can be eliminated using Polyinstantiation, in which case both tuples are allowed to exist.

8.2.3.2. Functional and Multivalued Dependencies

Suppose we have the following schema in a database:



"EMP_SAL"	
NAME	varchar
RANK	varchar
SALARY	float4

Suppose every employee is aware of the constraint that all employees having identical ranks have the same salaries. Given this scenario, an employee who is not permitted to have access to secret data can easily determine employee salaries, which are secret. If we examine the above example carefully, we see that it contains an inference channel because the functional dependency RANK → SALARY is not properly reflected in the classification levels of attributes RANK and SALARY. The way to avoid the problem in cases such as this is to raise the classification of the attribute RANK from non-secret to secret.

8.2.3.3. Value Constraints

Suppose that an attribute A is non-secret while attribute B is secret. Suppose the database enforces the constraint $A + B \leq 20$, which is made available to low users. The value of B does not affect the value of A directly, but it does determine the set of possible values A can take. Thus we have an inference channel. The usual solution to this problem is to allow such constraints to be defined only over a single security level. If a constraint is defined over several levels, then it is necessary to partition it into several single-level constraints. Thus, for example, the constraint $A + B \leq 20$ can be partitioned into $A \leq 10$ and $B \leq 10$.

8.2.3.4. Classification Constraints

Suppose the following integrity constraints apply to a database containing facts about ancient Iraqis:
Every ancient Iraqi is a Sumerian, Elamite, Assyrian, or a Babylonian.
All Sumerian and Assyrian are peaceable.
All Elamite and Babylonian are violent.
Saddam does not wish it to be known that he is peaceable! Bush tries to find out about Saddam from a system that refuses to answer whenever the answer, together with the integrity constraints, would imply the secret:
Bush: Is Saddam a Sumerian?
System: I will not tell you.
Bush: Is he an Elamite?
System: No.
Bush: Is he an Assyrian?
System: No.
Bush: Is he a Babylonian, then?
System: I will not tell you.
Bush, knowing that it is his disposition that Saddam is trying to conceal, notes that if he were not a Sumerian, then the system could have answered "No" to the first question without revealing the secret. Thus, Saddam must be a peaceable Sumerian.

8.3. General Characterizations of the Inference Problem

Probably the earliest formal characterization of the inference problem in databases is that of Goguen and Meseguer [GOGU84]. Consider a database in which each data item is given an access class, and suppose that the set of access classes is partially ordered. Define the relation → as follows: Given data items x and y, we say x → y if it is possible to infer y from x. The relation → is reflexive and transitive. A set S is said to be inferentially closed if whenever x is in S and x → y holds, then y belongs to S as well. Now, for an access class L, let E(L) denote the set consisting of all possible responses that are classified at access class less than or equal to L. There is an inference channel if E(L) is not inferentially closed.

In a refinement of Goguen and Meseguer’s definition, Denning and Morgenstern [DENN86] derive the inference relation from classical information theory. Given two data items x and y, let H(y) denote the uncertainty of y, and let $H_x(y)$ denote the uncertainty of y given x (where uncertainty is defined in the usual information-theoretic way). Then, the reduction in uncertainty of y given x is defined as follows:

$$INFER(x \to y) = \frac{H(y) - H_x(y)}{H(y)}$$

The value of INFER(x → y) is between 0 and 1. If the value is 0, then it is impossible to infer any information about y from x. If the value is between 0 and 1, then y becomes somewhat more likely given x. If the value is 1, then y can be inferred given x. Denning and Morgenstern pointed out serious drawbacks of their definition:

1. In most cases it is difficult, if not impossible, to determine the value of $H_x(y)$.
2. The definition does not take into account the computational complexity that is required to draw the inference.

8.4. Tools and Techniques for Dealing with Inference Channels

In this section, we discuss the various techniques and tools that have been proposed for locating inference channels and preventing their exploitation once they have been found. Basically two kinds of techniques have been proposed for locating and eliminating inference channels. One is to use semantic data modeling techniques to detect inference channels in the database design, and then to redesign the database so that these channels no longer exist. The other is to evaluate database transactions (involving either reads or updates, or both) to determine whether they lead to illegal inferences. If they do, the query is either disallowed or reclassified at the higher level.

For the first technique, we refer the reader to [HINK88a], [SMIT90], [SMIT90a], and [LUNT89a]. And for the second technique, we refer the reader to [MAZU88], [STAC90], [THUR87], [STON74], and [HAIG90]. When you consider these research works, you soon realize that more work needs to be done to understand the relative advantages and disadvantages of the two techniques.

8.5. References

- [DENN86] Denning, Dorothy E., and Matthew Morgenstern, "Military Database Technology Study: AI Techniques for Security and Reliability," SRI tech. report, Aug. 1986.
- [DENN86a] Denning, Dorothy E., "A Preliminary Note on the Inference Problem in Multilevel Database Management Systems," Proc. Nat’l Computer Security Center Invitational Workshop on Database Security, June 1986.
- [GOGU84] Goguen, Joseph A., and José Meseguer, "Unwinding and Inference Control," Proc. Symp. Security and Privacy, Apr. 1984, pp. 75-86.
- [HAIG90] Haigh, J.T., R.C. O’Brien, P.D. Stachour, and D.L. Toups, "The LDV Approach to Database Security," in Database Security III: Status and Prospects, D.I. Spooner and C. Landwehr, eds., North-Holland, Amsterdam, 1990.
- [HINK88a] Hinke, Thomas H., "Inference Aggregation Detection in Database Management Systems," Proc. IEEE Symp. Research in Security and Privacy, Apr. 1988, pp. 96-106.
- [LUNT89a] Lunt, T.F., |Aggregation and Inference: Facts and Fallacies," Proc. IEEE Symp. Research in Security and Privacy, May 1989, pp. 102-109.

- [MAZU88] Mazumdar, S., D. Stemple, and T. Sheard, "Resolving the Tension between Integrity and Security Using a Theorem Prover," Proc. ACM Int’l Conf. Management of Data, ACM, New York, 1988, pp. 233-242.
- [MEAD88a] Meadows, Catherine, and Sushil Jajodia, "Integrity versus Security in Multi-Level Secure Databases," in Database Security: Status and Prospects, C. Landwehr, ed., North-Holland, Amsterdam, 1988, pp. 89-101.
- [SICH83] Sicherman, G.L., W. de Jonge, and R.P. van de Riet, "Answering Queries without Revealing Secrets," ACM Trans. Database Systems, Mar. 1983, Vol. 8, No. 1, pp. 41-59.
- [SMIT90] Smith, Gary W., "Modeling Security-Relevant Data Semantics," Proc. IEEE Symp. Research in Security and Privacy, May 1990, pp. 384-391.
- [SMIT90a] Smith, Gary W., The Modeling and Representation of Security Semantics for Database Applications, doctoral dissertation, George Mason Univ., Fairfax, Va., 1990.
- [STAC90] Stachour, Paul D., and Bhavani Thuraisingham, "Design of LDV: A Multilevel Secure Relational Database Management System," IEEE Trans. Knowledge and Data Eng., Vol. 2, No. 2, June 1990, pp. 190-209.
- [STON74] Stonebraker, M., "Implementation of Integrity Constraints and Views by Query Modification," ACM Nat’l Conf. Proc., 1974, pp. 180-186.
- [SU86] Su, Tzong-An, Inferences in Databases, doctoral dissertation, Case Western Reserve Univ., Cleveland, Ohio, 1986.
- [SU87] Su, Tzong-An, and Gultekin Ozsoyoglu, "Data Dependencies and Inference Control in Multilevel Relational Database Systems," Proc. Symp. Security and Privacy, Apr. 1987, pp. 202-211.
- [SU90] Su, Tzong-An, and Gultekin Ozsoyoglu, "Multivalued Dependency Inferences in Multilevel Relational Database Systems," Database Security III: Status and Prospects, D.L. Spooner and C. Landwehr, eds., North-Holland, Amsterdam, 1990, pp. 293-300.
- [THUR87] Thuraisingham, Bhavani M., "Security Checking in Relational Database Management Systems Augmented with Inference Engines," Computers and Security, Vol. 6, 1987, pp. 479-492.
- Sushil Jajodia, and Catherine Meadows, "Inference Problems in Multilevel Secure Database Management Systems," in "Information Security: An Integrated Collection of Essays," IEEE Computer Society Press, Los Alamitos, CA USA, 1995.
- Charles P. Pfleeger, and Shari Lawrence Pfleeger, "Security in Computing, Fourth Edition," Prentice Hall, October 13, 2006.

Chapter 9. Technology behind Anti Inference Hub

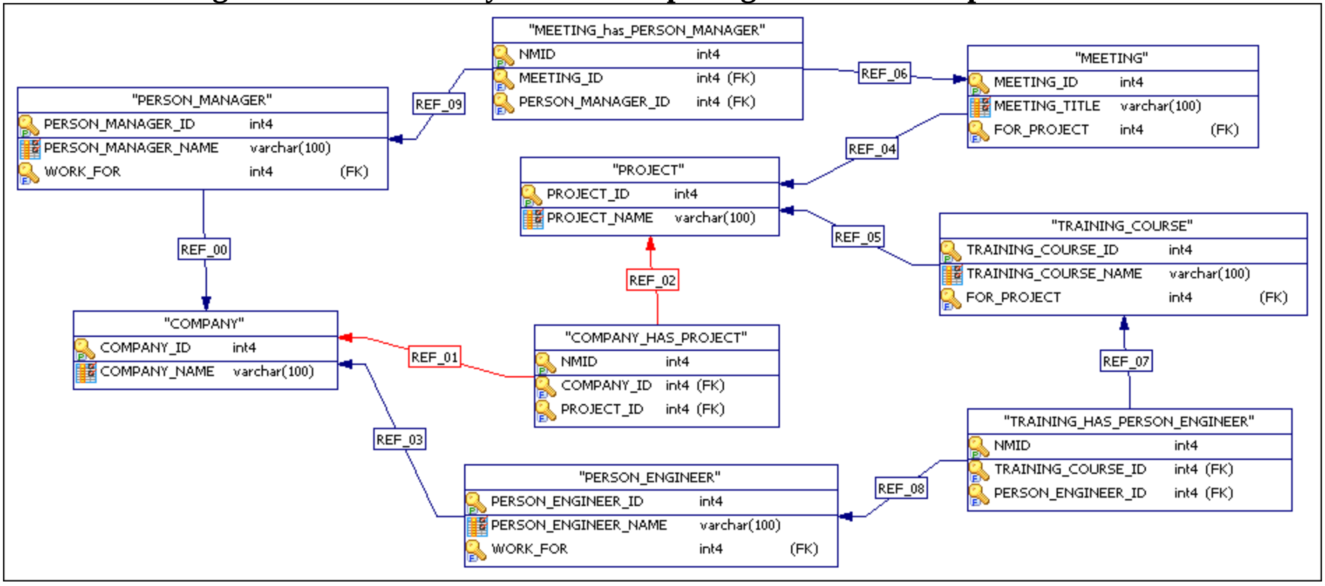
Anti Inference Hub is based on a dynamic query processing technology described in the following paper: X. Chen, R. Wei, "A Dynamic Method for Handling the Inference Problem in Multilevel Secure Databases," Proc. International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume I - Volume 01, 2005, pp. 751-756.

Chapter 10. The Sample Database

10.1. Introduction

Anti Inference Hub comes with a sample database that may be used for testing and understanding purposes.

Figure 10.1. The Entity-Relationship Diagram of the Sample Database



10.2. Location of the Sample Database

The sample database is included in the directory database in Anti Inference Hub binary package.

Table 10.1. Content of the database Directory

File	Description
inferenceoracle.sql	This file contains SQL script needed to create the sample database with an Oracle DBMS.
inferencepostgresqlmysql.sql	This file contains SQL script needed to create the sample database with a PostgreSQL DBMS, or a MySQL DBMS.
createoracledb.bat	This file executes SQL script in the file inferenceoracle.sql in accordance with an Oracle DBMS running under Windows.
createoracledb.sh	This file executes SQL script in the file inferenceoracle.sql in accordance with an Oracle DBMS running under Linux.
createpostgresqldb.bat	This file executes SQL script in the file inferencepostgresqlmysql.sql in accordance with a PostgreSQL DBMS running under Windows.
createpostgresqldb.sh	This file executes SQL script in the file inferencepostgresqlmysql.sql in accordance with a PostgreSQL DBMS running under Linux.
createmysqldb.bat	This file executes SQL script in the file inferencepostgresqlmysql.sql in accordance with a MySQL DBMS running under Windows.

createmysqldb.sh	This file executes SQL script in the file inferencepostgresqlmysql.sql in accordance with a MySQL DBMS running under Linux.
entity-relationship diagram.pdd	This file is the entity-relationship diagram of the sample database built using MicroOLAP Database Designer for PostgreSQL.
entity-relationship diagram.bmp	This file is the entity-relationship diagram of the sample database exported as a bitmap image.

10.3. Creating the Sample Database

To create the sample database, first install a DBMS of choice under an operating system of choice, then navigate to the database directory, and execute the file corresponding to your platform.

10.4. Our Test Environment

The sample database was successfully tested with the following:

- Oracle Database 10g Release 2 (10.2.0.1.0) under Windows XP Service Pack 2 and under Linux Ubuntu.
- PostgreSQL 8.4.2-1 under Windows XP Service Pack 2 and under Linux Ubuntu.
- MySQL 5.1.42 under Windows XP Service Pack 2 and under Linux Ubuntu.

Chapter 11. Anti Inference Hub in Action

11.1. Introduction

In this chapter, we put Anti Inference Hub in action using the sample database created in Chapter 10.

11.2. The Inference Attempt to Address

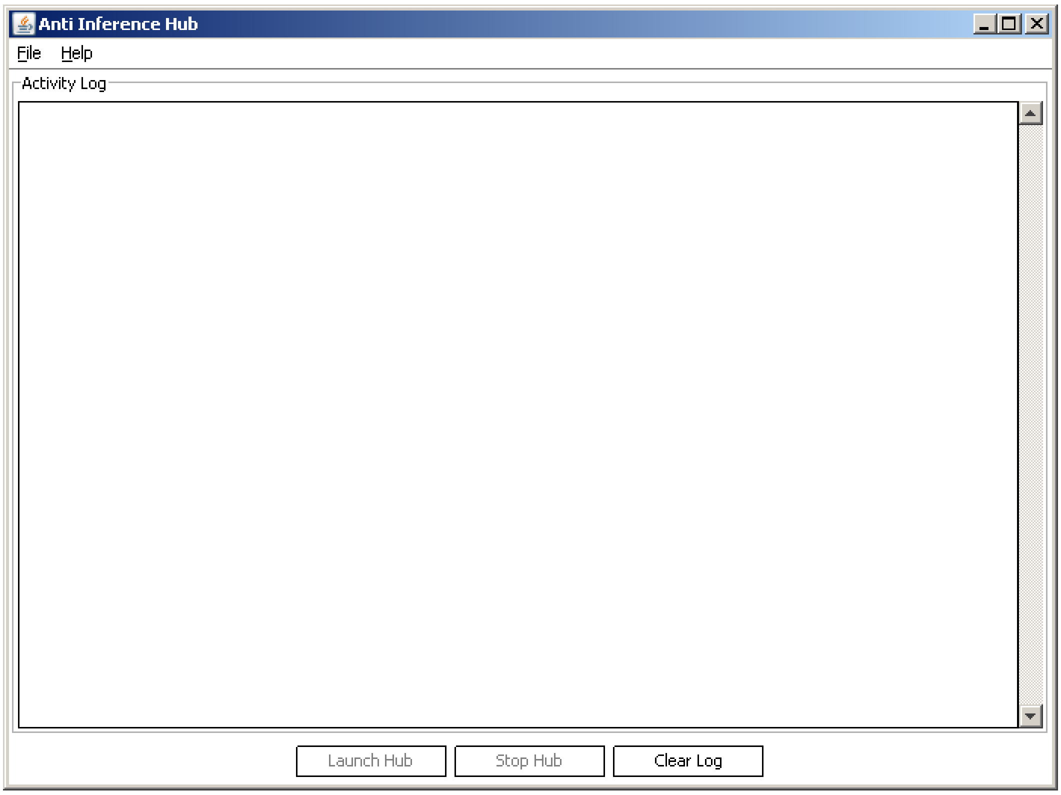
In the sample database, suppose that a low user is able to know the following by executing queries against the database:

- For which COMPANY a PERSON_MANAGER works.
- PERSON_MANAGER attending a MEETING.
- MEETING on a PROJECT.

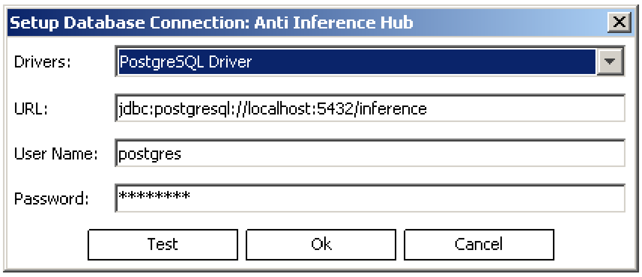
If that was true, then the low user can immediately infer the COMPANY supporting the PROJECT. In the following section, we show how Anti Inference Hub can be used to thwart this inference attempt.

11.3. Thwarting the Inference Attempt

Start the Hub as described in Section 3.2. Starting Anti Inference Hub.



Press Ctrl+D, and establish the connection with the DBMS you are using.


Setup Database Connection: Anti Inference Hub

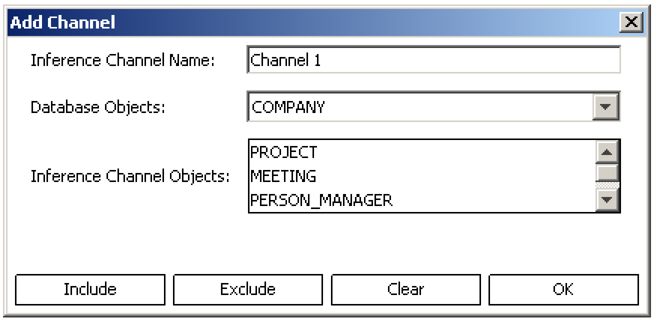
Drivers: PostgreSQL Driver

URL: jdbc:postgresql://localhost:5432/inference

User Name: postgres

Password: *********

Press Ctrl+I, then press the button Add in the box that shows to setup an inference channel named "Channel 1", of length 4, containing the following objects: PROJECT, MEETING, PERSON_MANAGER, and COMPANY.

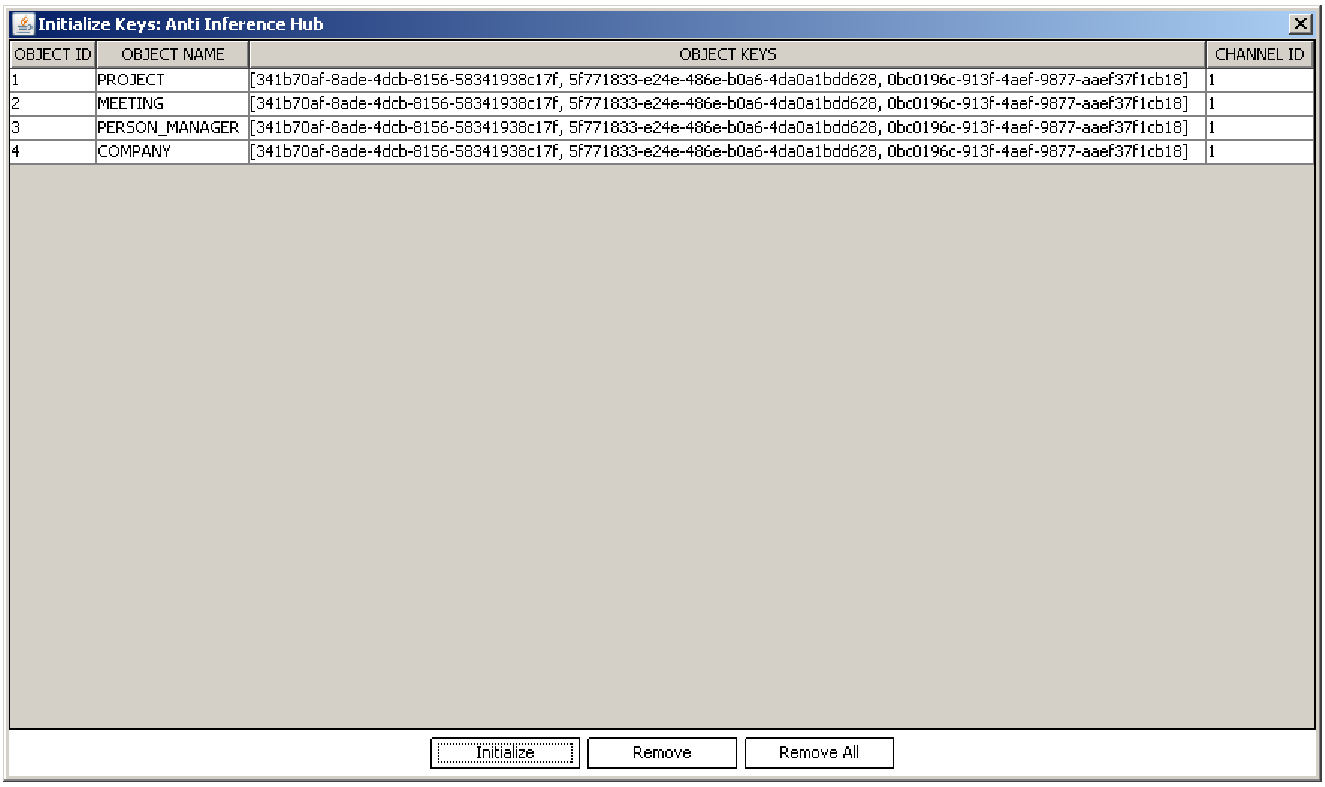


Inference Channel Name: Channel 1

Database Objects: COMPANY

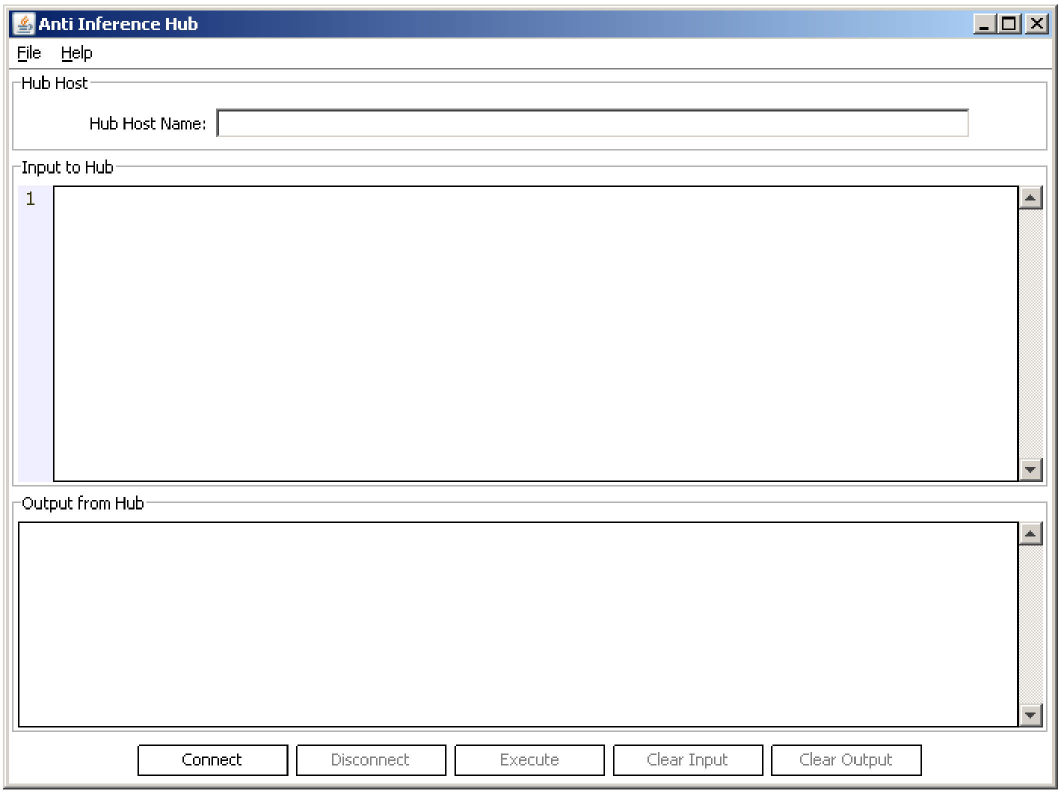
Inference Channel Objects: PROJECT, MEETING, PERSON_MANAGER

Close the Setup Inference Channels Box, and press Ctrl+K, then press the button Initialize in the box that shows to initialize "Channel 1" objects keys.



OBJECT ID	OBJECT NAME	OBJECT KEYS	CHANNEL ID
1	PROJECT	[341b70af-8ade-4dcb-8156-58341938c17f, 5f771833-e24e-486e-b0a6-4da0a1bdd628, 0bc0196c-913f-4aef-9877-aaef37f1cb18]	1
2	MEETING	[341b70af-8ade-4dcb-8156-58341938c17f, 5f771833-e24e-486e-b0a6-4da0a1bdd628, 0bc0196c-913f-4aef-9877-aaef37f1cb18]	1
3	PERSON_MANAGER	[341b70af-8ade-4dcb-8156-58341938c17f, 5f771833-e24e-486e-b0a6-4da0a1bdd628, 0bc0196c-913f-4aef-9877-aaef37f1cb18]	1
4	COMPANY	[341b70af-8ade-4dcb-8156-58341938c17f, 5f771833-e24e-486e-b0a6-4da0a1bdd628, 0bc0196c-913f-4aef-9877-aaef37f1cb18]	1

Close the Initialize Keys Box, then press the button "Launch Hub" to launch the Hub. Now, start the Hub Client as described in Section 3.2. Starting Anti Inference Hub.



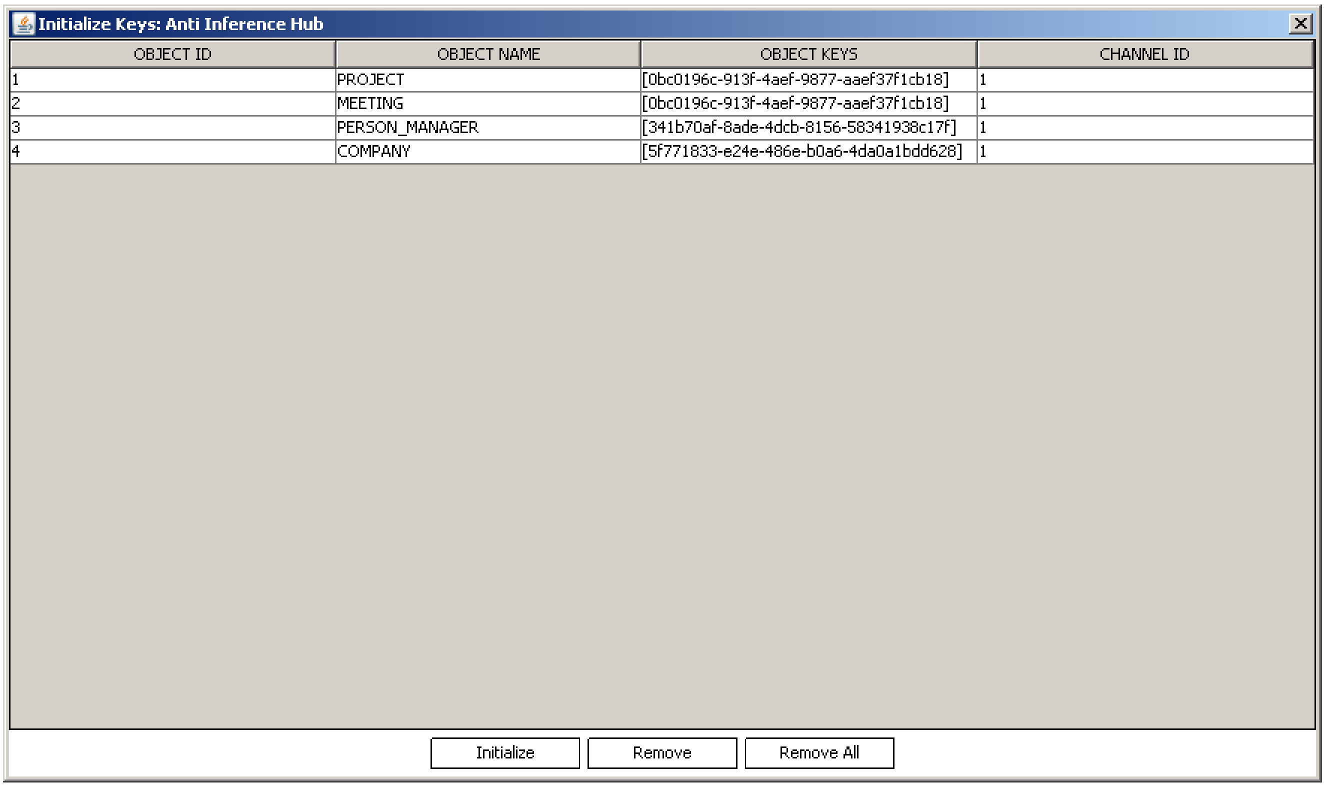
Enter the appropriate Hub Host Name in its field, and connect to the Hub by pressing the Connect button. Act as a low user, and execute the following query in the Hub Client:

```
SELECT PERSON_MANAGER_NAME, COMPANY_NAME
FROM PERSON_MANAGER, COMPANY
WHERE PERSON_MANAGER.WORK_FOR = COMPANY.COMPANY_ID
AND PERSON_MANAGER.PERSON_MANAGER_NAME = 'MANAGER 1';
```

In this query, you are trying to know for which COMPANY a PERSON_MANAGER works. You should receive the following XML output from the Hub:

```
<?xml version="1.0" encoding="UTF-8"?>
<Results>
    <Row>
        <person_manager_name>MANAGER 1</person_manager_name>
        <company_name>COMPANY 1</company_name>
    </Row>
</Results>
```

Which tells you that "MANAGER 1" works for "COMPANY 1". Note the change to "Channel 1" objects keys by opening the Initialize Keys Box in the Hub.



OBJECT ID	OBJECT NAME	OBJECT KEYS	CHANNEL ID
1	PROJECT	[0bc0196c-913f-4aef-9877-aaef37f1cb18]	1
2	MEETING	[0bc0196c-913f-4aef-9877-aaef37f1cb18]	1
3	PERSON_MANAGER	[341b70af-8ade-4dcb-8156-58341938c17f]	1
4	COMPANY	[5f771833-e24e-486e-b0a6-4da0a1bdd628]	1

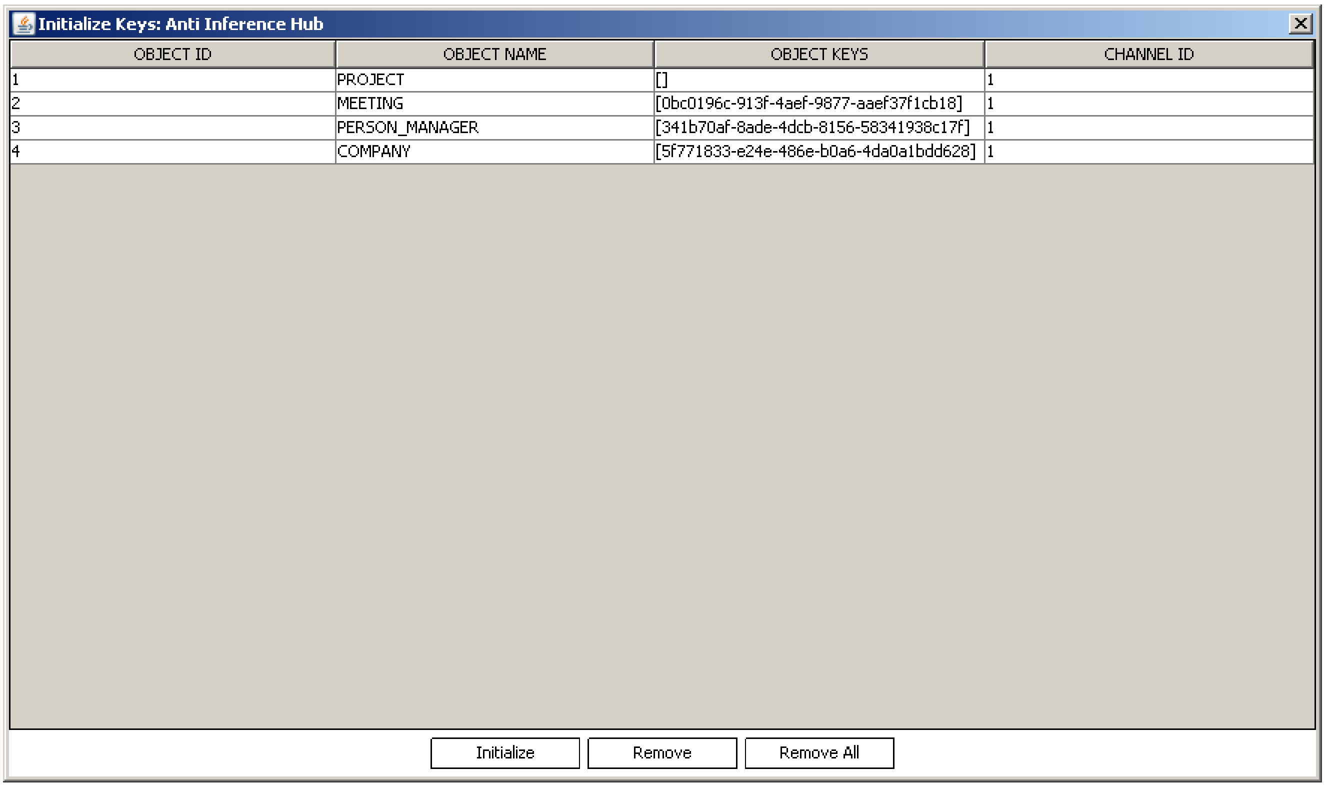
Now, execute the following query in the Hub Client:

```
SELECT MEETING_TITLE, PERSON_MANAGER_NAME
FROM MEETING, MEETING_HAS_PERSON_MANAGER, PERSON_MANAGER
WHERE MEETING.MEETING_ID = MEETING_HAS_PERSON_MANAGER.MEETING_ID
AND PERSON_MANAGER.PERSON_MANAGER_ID =
MEETING_HAS_PERSON_MANAGER.PERSON_MANAGER_ID
AND MEETING.MEETING_TITLE = 'MEETING 1';
```

In this query, you are trying to know the PERSON_MANAGER attending a MEETING. You should receive the following XML output from the Hub:

```
<?xml version="1.0" encoding="UTF-8"?>
<Results>
    <Row>
        <meeting_title>MEETING 1</meeting_title>
        <person_manager_name>MANAGER 1</person_manager_name>
    </Row>
</Results>
```

Which tells you that "MANAGER 1" attends "MEETING 1". Note the change to "Channel 1" objects keys by opening the Initialize Keys Box in the Hub.



Initialize Keys: Anti Inference Hub

OBJECT ID	OBJECT NAME	OBJECT KEYS	CHANNEL ID
1	PROJECT	[]	1
2	MEETING	[0bc0196c-913f-4aef-9877-aaef37f1cb18]	1
3	PERSON_MANAGER	[341b70af-8ade-4dcb-8156-58341938c17f]	1
4	COMPANY	[5f771833-e24e-486e-b0a6-4da0a1bdd628]	1

Note that PROJECT object is now a reserved object in "Channel 1" because it has an empty key set. Now, execute the following query in the Hub Client:

```
SELECT MEETING_TITLE, PROJECT_NAME
FROM MEETING, PROJECT
WHERE MEETING.FOR_PROJECT = PROJECT.PROJECT_ID
AND PROJECT.PROJECT_NAME = 'PROJECT 1';
```

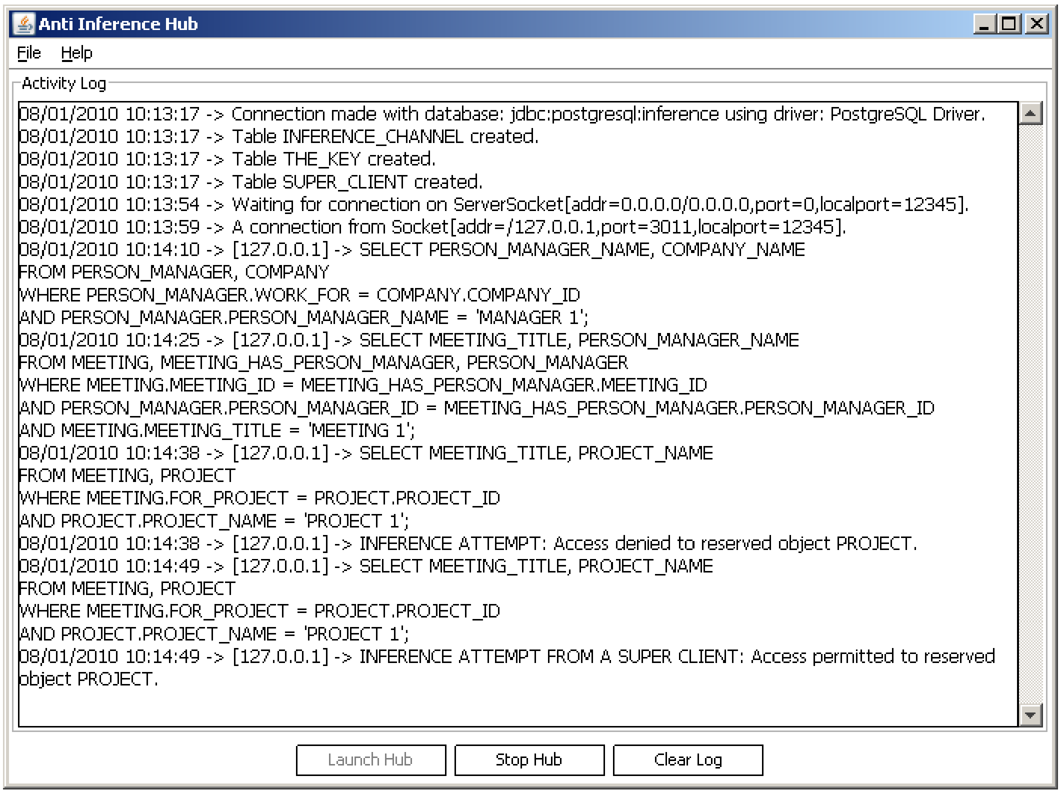
In this query, you are trying to know the MEETING on a PROJECT. If you received a response to this query (actually the response will tell you that "MEETING 1" is on project "PROJECT 1"), then you can immediately infer that "COMPANY 1" is supporting "PROJECT 1"; an inference you base on the results for queries you executed so far. Therefore, result for this query should be blocked. Indeed, it is! The output from the Hub will be:

```
INFERENCE ATTEMPT: Access denied to reserved object PROJECT.
```

Why don't you adjust yourself as a super client and re-execute the same query! Press Ctrl+A, then press the button Add in the box that shows to adjust yourself as a super client, then re-execute the same query. You should receive the following XML output from the Hub:

```
INFERENCE ATTEMPT FROM A SUPER CLIENT: Access permitted to reserved
object PROJECT.
<?xml version="1.0" encoding="UTF-8"?>
<Results>
    <Row>
        <meeting_title>MEETING 1</meeting_title>
        <project_name>PROJECT 1</project_name>
    </Row>
</Results>
```

Thus, as a super client, the Hub allows you to infer. Please note that Anti Inference Hub does not take any IP spoofing attacks into consideration. Securing a network against such attacks falls beyond the purpose of Anti Inference Hub. You may take a look at the Hub log which should be as follow:



```
08/01/2010 10:13:17 -> Connection made with database: jdbc:postgresql:inference using driver: PostgreSQL Driver.
08/01/2010 10:13:17 -> Table INFERENCE_CHANNEL created.
08/01/2010 10:13:17 -> Table THE_KEY created.
08/01/2010 10:13:17 -> Table SUPER_CLIENT created.
08/01/2010 10:13:54 -> Waiting for connection on ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=12345].
08/01/2010 10:13:59 -> A connection from Socket[addr=/127.0.0.1,port=3011,localport=12345].
08/01/2010 10:14:10 -> [127.0.0.1] -> SELECT PERSON_MANAGER_NAME, COMPANY_NAME
FROM PERSON_MANAGER, COMPANY
WHERE PERSON_MANAGER.WORK_FOR = COMPANY.COMPANY_ID
AND PERSON_MANAGER.PERSON_MANAGER_NAME = 'MANAGER 1';
08/01/2010 10:14:25 -> [127.0.0.1] -> SELECT MEETING_TITLE, PERSON_MANAGER_NAME
FROM MEETING, MEETING_HAS_PERSON_MANAGER, PERSON_MANAGER
WHERE MEETING.MEETING_ID = MEETING_HAS_PERSON_MANAGER.MEETING_ID
AND PERSON_MANAGER.PERSON_MANAGER_ID = MEETING_HAS_PERSON_MANAGER.PERSON_MANAGER_ID
AND MEETING.MEETING_TITLE = 'MEETING 1';
08/01/2010 10:14:38 -> [127.0.0.1] -> SELECT MEETING_TITLE, PROJECT_NAME
FROM MEETING, PROJECT
WHERE MEETING.FOR_PROJECT = PROJECT.PROJECT_ID
AND PROJECT.PROJECT_NAME = 'PROJECT 1';
08/01/2010 10:14:38 -> [127.0.0.1] -> INFERENCE ATTEMPT: Access denied to reserved object PROJECT.
08/01/2010 10:14:49 -> [127.0.0.1] -> SELECT MEETING_TITLE, PROJECT_NAME
FROM MEETING, PROJECT
WHERE MEETING.FOR_PROJECT = PROJECT.PROJECT_ID
AND PROJECT.PROJECT_NAME = 'PROJECT 1';
08/01/2010 10:14:49 -> [127.0.0.1] -> INFERENCE ATTEMPT FROM A SUPER CLIENT: Access permitted to reserved
object PROJECT.
```

When done, disconnect from the Hub, and stop it.

Appendix A. Anti Inference Hub License

The Apache License explains all the things that you are allowed to do with Anti Inference Hub code and documentation.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Appendix B. Anti Inference Hub Code Disclaimer

The author of this software code has used his best efforts in preparing the code. These efforts include the development, research, testing, and optimization of the theories and programs to determine their effectiveness. This software code is not designed or intended for use in the design, construction, operation or maintenance of any nuclear facility. Author disclaims any express or implied warranty of fitness for such uses. The author makes no warranty of any kind, expressed or implied, with regard to this software code or to the documentation accompanying it. In no event shall the author be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption whatsoever) arising out of, the furnishing, performance, or use of this software code, even if advised of the possibilities of such damages.