CSC 8500

Research Project

Matthew Ingersole

Table of Contents

Introduction

The process of buying a product can be as simple as walking into a supermarket and buying a carton of milk. For products of higher value the process can be much more complex, such as buying an investment or a house

A purchase of higher value typically involves multiple stages before the transaction is complete. Because of the higher value and the complexity of the process, these transactions generally involve a facilitator on the sales end, AKA the Salesman.

A salesman's income is directly related to the amount of business he generates. The salesman must perform duties in respect to all areas of the sales process, but a good salesman's main going concern should always be the amount of business coming in the door. This constant flow of transactions entering and exiting the sales process is known as the 'pipeline'.

Each and every salesman has a pipeline, it is all the transactions or potential future transactions that the salesman is overseeing that have not yet completed. And it is the management of this pipeline that divides the best salesmen from the rest.
During my time as a Financial Advisor I witnessed first hand the value of pipeline management and designed a complex excel database to manage my own pipeline. The system was a success for myself and my colleagues but could have been far better with a relational design.

For my CSC 8500 research project I have taken my old excel system and transformed it into a functional relational database. I have designed the database in consultation with ex-colleagues and have imported 12 months of data from 2009-2010 for testing purposes.

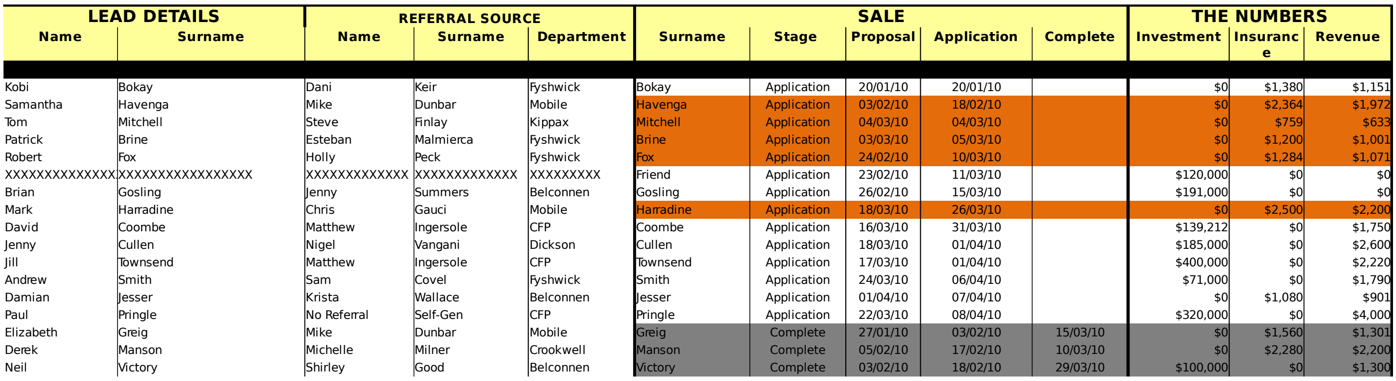
1 - System Definition

The following transcripts are from my discussions with Alex Mcadam, a Financial Advisor based in Kiama, NSW on 1 May 2015:

Defining the scope and boundary

Sale Process - "A sale starts with a client who may be interested in a product or service, known as a 'lead' or a 'referral'. Leads are generated by a referral partner and/or referral department or neither, in which case it is self-generated. At some point I must make a sales proposal of some kind to the client (eg. investment of $100,000 with a fee of $2000). The client must decide whether to proceed with the proposal. If the client wishes to proceed, the sale will go through a few stages (eg. proposal -> application -> complete) and could be declined at any time if the client changes their mind. Finally the sale completes, and the commission is paid to me."

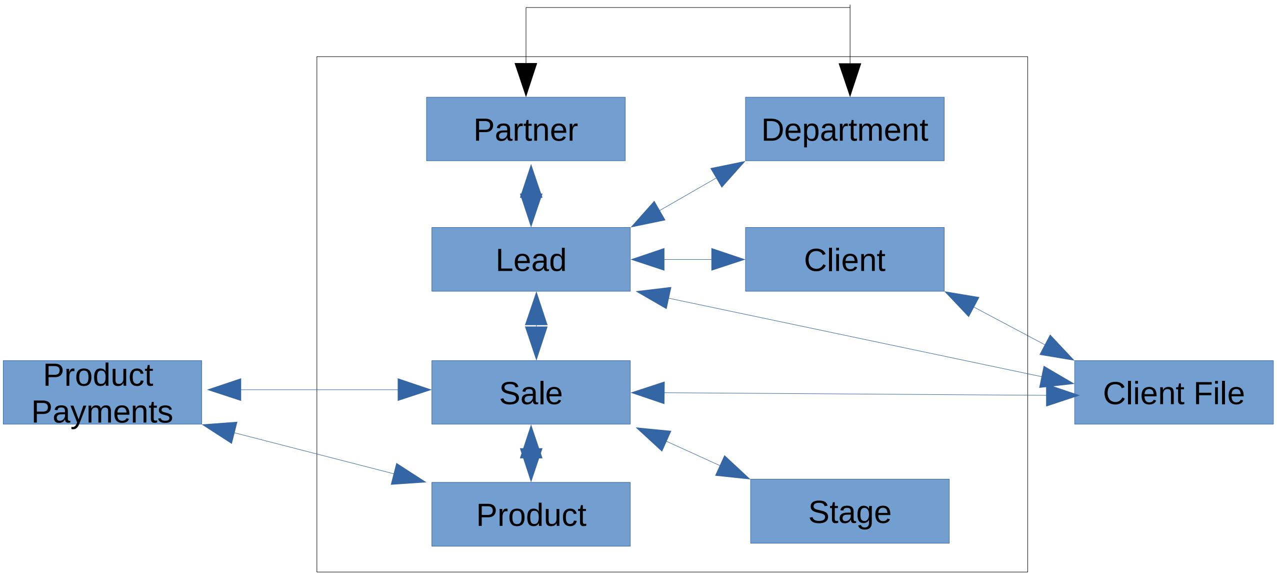
Boundaries - "I need the system to be as lightweight and simple as possible. I already have access to large CRM systems that can do this for me but it requires too much data input and slows me down. I only want to manage my pipeline. I do not want detailed information on every client. I do not care which department employs which partner. I do not want the database to be linked to the product payments system (commissions paid for completed sales)"



LEAD DETAILS		REFERRAL SOURCE			SALE					THE NUMBERS		
Name	Surname	Name	Surname	Department	Surname	Stage	Proposal	Application	Complete	Investment	Insurance	Revenue
Kobi	Bokay	Dani	Keir	Fyshwick	Bokay	Application	20/01/10	20/01/10		$0	$1,380	$1,151
Samantha	Havenga	Mike	Dunbar	Mobile	Havenga	Application	03/02/10	18/02/10		$0	$2,364	$1,972
Tom	Mitchell	Steve	Finlay	Kippax	Mitchell	Application	04/03/10	04/03/10		$0	$759	$633
Patrick	Brine	Esteban	Malmierca	Fyshwick	Brine	Application	03/03/10	05/03/10		$0	$1,200	$1,001
Robert	Fox	Holly	Peck	Fyshwick	Fox	Application	24/02/10	10/03/10		$0	$1,284	$1,071
XXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXX	XXXXXXXXXXXXX	XXXXXXXXXXXXX	XXXXXXXXX	Friend	Application	23/02/10	11/03/10		$120,000	$0	$0
Brian	Gosling	Jenny	Summers	Belconnen	Gosling	Application	26/02/10	15/03/10		$191,000	$0	$0
Mark	Harradine	Chris	Gauci	Mobile	Harradine	Application	18/03/10	26/03/10		$0	$2,500	$2,200
David	Coombe	Matthew	Ingersole	CFP	Coombe	Application	16/03/10	31/03/10		$139,212	$0	$1,750
Jenny	Cullen	Nigel	Vangani	Dickson	Cullen	Application	18/03/10	01/04/10		$185,000	$0	$2,600
Jill	Townsend	Matthew	Ingersole	CFP	Townsend	Application	17/03/10	01/04/10		$400,000	$0	$2,220
Andrew	Smith	Sam	Covel	Fyshwick	Smith	Application	24/03/10	06/04/10		$71,000	$0	$1,790
Damian	Jesser	Krista	Wallace	Belconnen	Jesser	Application	01/04/10	07/04/10		$0	$1,080	$901
Paul	Pringle	No Referral	Self-Gen	CFP	Pringle	Application	22/03/10	08/04/10		$320,000	$0	$4,000
Elizabeth	Greig	Mike	Dunbar	Mobile	Greig	Complete	27/01/10	03/02/10	15/03/10	$0	$1,560	$1,301
Derek	Manson	Michelle	Milner	Crookwell	Manson	Complete	05/02/10	17/02/10	10/03/10	$0	$2,280	$2,200
Neil	Victory	Shirley	Good	Belconnen	Victory	Complete	03/02/10	18/02/10	29/03/10	$100,000	$0	$1,300

A snapshot of the current excel system

The Pipeline database system will be a simple lead and sale management system. As such it will not be designed to keep detailed information on clients. The system will not be concerned with the employment relationship between partners and departments. The system will not be concerned with the actual commission payments from product providers to the salesman. The system will only be designed to manage new leads through the sale process until completion.



Systems boundary diagram

Identifying the major user views

User views - "I currently use an excel system to manage my pipeline. I am the only user. I would like the new system to allow my assistant to view information. She can use this information to help me schedule appointments and schedule me action items. I would like the new system to allow my referral partners and departments to send me leads. They should only be able to create leads and view their existing leads in the system.

The database system is for use by a Financial Advisor, his assistant/s, his referral department/s and his referral partner/s. The Financial Advisor should have full access to the database. The Assistant should have limited viewing access to the database. Departments and partners should be able to create new leads and view existing leads that they have created.

Data	Access Type	Financial Advisor	Assisstant	Partner	Department
Department	Maintain	x			
	Query	x	x		
Partner	Maintain	x			
	Query	x	x		
Lead	Maintain	x			
	Query	x	x		
Department Lead	Maintain	x			x
	Query	x	x		x
Partner Lead	Maintain	x		x	
	Query	x	x	x	
Sale	Maintain	x			
	Query	x	x		
Product	Maintain	x			
	Query	x	x		
Stage	Maintain	x			
	Query	x	x		
SaleStage	Maintain	x			
	Query	x	x		

Major user views

2 - Requirements Collection and Analysis

The following transcipts are from discussions with Alex McAdam, a Financial Advisor based in Kiama, NSW.

Gathering information on user views

"I need full access to the database, after all it is my business that we are managing here. My assistant only needs to query the database for diary and scheduling purposes. I would like my referral partners and departments to be able to create new leads and view their current leads in the system."

Managing the user views

Below is a coss-reference of user views with the main types of data used by each.

	Financial Advisor	Assistant	Partner	Department
Partner	x	x		
Department	x	x		
Lead	x	x	x	x
Client	x	x		
Sale	x	x		
Product	x	x		
Stage	x	x		
SaleStage	x	x		

The centralized approach will be used to merge all user requirements, as all user views are overlapping.

Gathering information on the system requirements

Main transactions - "The creation of leads and sales and the movement of those sales through the various sales stages will be the primary transactions. I will rarely create new referral partners, referral departments, products and/or stages."

Historical Data - "I will keep all sales data in the system for the financial year. Once the financial year is finished I will delete all past sales and start fresh for the new financial year."

Leads - "A lead can be generated from a referral department and/or a referral partner or neither. A lead is sent on a date. A lead is sent for one client only. A lead can become zero or multiple sales. I would like to keep information on the date of the lead and who (if anyone) sent it to me. I would also like a comment of some sort so I know what the client would like to discuss (ie. wants to discuss investments). If a lead does not move to a sale I want it deleted from the system. I want to know which leads have not yet been made a sale proposal, so I know who to contact for a first appointment."

Clients - " A client is from one and only one lead. To keep the system fast and simple, I would like to only keep minimal information on the client. Name, Surname and a contact email or phone is sufficient."

Referral Department - "I only want to keep information on the name of each referral department. I would like the ability to search all leads and sales that were generated by a department for training and feedback puposes."

Referral Partners - "I only want to keep information on the name and surname of each referral partner. I would like the ability to search all leads and sales that were generated by a partner for training and feedback puposes."

Sales - "A sale has a value and a commission. A sale can come from only one lead. Each sale is of one product. Each sale goes through multiple stages but the latest stage is always the current stage. I need to keep information on the product being sold and the value and commission of the sale. I want to know which stage current sales are in so I can action them accordingly."

Product - "A product can be sold in multiple sales. I only need to keep information on the name of the product."

Stage - "A stage can contain contain multiple sales. I need to keep information on the name of the stage. The stages are as follows "Proposal", "Application", "Complete", "Declined". A sale always starts at "Proposal". A sale must move through "Application" to get to "Complete". A sale could go to "Declined at anytime."

SaleStage - "I would like to keep historical information on when each sale entered each stage in the process. I will use this information to determine how long my sales process takes and use it for forecasting purposes."

Other - "I would like to know my total sales figures, so I can determine my bonus and progress toward sales targets."

System Requirements

Initial Database Size

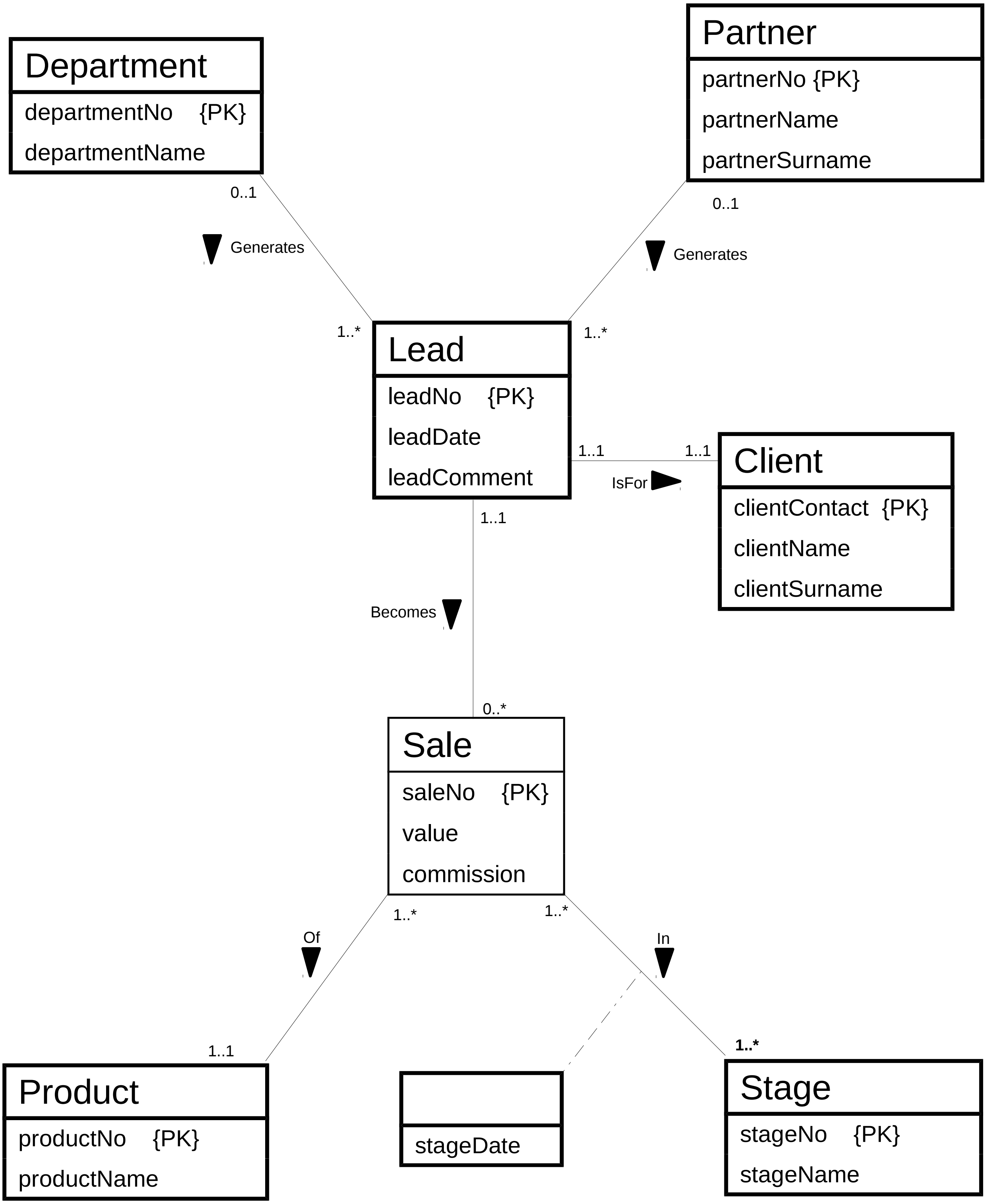
1) There are approximately 15 Referral Departments.
2) There are approximately 50 Referral Partners.
3) There are 4 sale Stages.
4) There are 2 Products offered.

Database rate of growth

1) Approximately 10 Leads/Clients will be added each week.
2) Approximately 50% of Leads will eventually result in a Sale, 5 per week.
3) Once a Lead does not result in a Sale, the Lead will be deleted from the database.
4) A Sale will take approximately 2 weeks and could move through up to 3 stages in this period.
5) Approximately 1 new Referral Department will be added every six months.
6) Approximately 1 new Referral Partner will be added every month.
7) It is not expected that new Stages and Products will be added.

3 - Database Design

3.1 - Conceptual Database Design using ER Modelling



3.2 - Logical Database Design

Partner

Partner has a one-to-many relationship with Lead. As Partner is on the "one side" of the relationship, Partner is designated as the parent entity. The primary key from Partner is posted to Lead but Partner itself remains unchanged.

Partner(partnerNo, partnerName, partnerSurname)

Partner = {partnerNo:D(partnerNo), partnerName:D(partnerName), partnerSurname:D(partnerSurname)}

Where

D(partnerNo) = {partnerNo|partnerNo is an INT AND MaxLength = 11}
D(partnerName) = {partnerName|partnerName is a VARCHAR AND MaxLength = 30}
D(partnerSurname) = {partnerSurname|partnerSurname is a VARCHAR AND MaxLength = 30}

Department

Department has a one-to-many relationship with Lead. As department is on the "one side" of the relationship, Department is designated as the parent entity. The primary key from Department is posted to Lead but Department itself remains unchanged.

Department(departmentNo, departmentName)

Department = {departmentNo:D(departmentNo), departmentName:D(departmentName)}

Where

D(departmentNo) = {departmentNo|departmentNo is an INT AND MaxLength = 11}
D(departmentName) = {departmentName|departmentName is a VARCHAR AND MaxLength = 30}

Client

Client has a one-to-one relationship with Lead. Because the relationship has mandatory participation on both sides we should combine the entities involved into one relation and choose one of the primary keys to be the primary key of the new relation while the other primary key will serve as an alternate key. In this case we will merge Client into the Lead entity and use Lead's primary key as the new primary key, Client's primary key (clientContact) will become an alternate key in Lead.

Lead

Lead has a one-to-one relationship with Client. As mentioned above Client will be merged into Lead and Client's primary key (clientContact) will become an alternate key for Lead.

Lead has a one-to-many relationship with both Department and Partner. In both relationships Lead is the "many" side and is therefore the child entity. We must post the primary keys of both Partner and Department into Lead to act as Foreign Keys.

Lead has a one-to-many relationship with Sale. In this relationship Lead is the "one" side and is therefore the parent entity. The primary key of Lead is posted into Sale but Lead remains unchanged.

Lead(leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, departmentNo)

Lead = {leadNo:D(leadNo), leadDate:D(leadDate), leadComment:D(leadComment), clientContact:D(clientContact), clientName:D(clientName), clientSurname:D(clientSurname), partnerNo:D(partnerNo), departmentNo:D(departmentNo)}

Where

D(leadNo) = {leadNo|leadNo is an INT AND MaxLength = 11}
D(leadDate) = {leadDate|leadDate is a TIMESTAMP}
D(leadComment) = {leadComment|leadComment is a MEDIUMTEXT AND MaxLength = 16,777,215}
D(clientContact)={clientContact|clientContact is a VARCHAR AND MaxLength = 30}
D(clientName)={clientName|clientName is a VARCHAR AND MaxLength = 30}
D(clientSurname)={clientSurname|clientSurname is a VARCHAR AND MaxLength = 30}
D(partnerNo) = {partnerNo|partnerNo is an INT AND MaxLength = 11}
D(departmentNo) = {departmentNo|departmentNo is an INT AND MaxLength = 11}

Sale

Sale has a one-to-many relationship with Lead. In this relationship Sale is on the "many" side of the relationship and is therefore the child entity. As such we must post a copy of Lead's primary key (leadNo) to Sale to act as a foreign key.

Sale has a one-to-many relationship with Product. In this relationship Sale is on the many side of the relationship and is therefor the child entity. As such we must post a copy of Product's primary key (productNo) to Sale to act as a foreign key.

Sale has a many-to-many relationship with Stage. We must create a new relation (SaleStage) to represent this relationship. We post a copy of Sale's primary key (saleNo) to the new relation to act as a foreign key and, in this case, part of the composite primary key.

Sale(saleNo, value, commission, productNo, leadNo)

Sale = {saleNo:D(saleNo), value:D(value), commission:D(commission), productNo:D(productNo), leadNo:(leadNo)}

Where

D(saleNo) = {saleNo|saleNo is an INT AND MaxLength = 11}
D(value) = {value|value is an INT AND MaxLength = 11}
D(commission) = {commission|commission is an INT AND MaxLength = 11}
D(productNo) = {productNo|productNo is an INT AND MaxLength = 11}
D(leadNo) = {leadNo|leadNo is an INT AND MaxLength = 11}

Product

Product has a one-to-many relationship with Sale. In this relationship Product is on the "one" side of the relationship and is therefore the parent entity. As such we must post a copy of Product's primary key (productNo) to Sale to act as a foreign key. The Product entity remains unchanged.

Product(productNo, productName)

Product = {productNo:D(productNo), productName:D(productName)}

Where

D(productNo) = {productNo|productNo is an INT AND MaxLength = 11}
D(productName) = {productName|productName is a VARCHAR AND MaxLength = 30}

Stage

Stage has a many-to-many relationship with Sale. We must create a new relation (SaleStage) to represent this relationship. We post a copy of Stage's primary key (stageNo) to the new relation to act as a foreign key and, in this case, part of the composite primary key. The entity Stage remains unchanged.

Stage(stageNo, stageName)

Stage = {stageNo:D(stageNo), stageName:D(stageName)}

Where

D(stageNo) = {stageNo|stageNo is an INT AND MaxLength = 11}
D(stageName) = {stageName|stageName is a VARCHAR AND MaxLength = 30}

SaleStage

There is a many-to-many relationship between Sale and Stage. We must create a new relation (SaleStage) to represent this relatiionship. As mentioned above we will post the primary keys of both Sale (saleNo) and Stage(stageNo) into the new relation to act as foreign keys and, together, the primary key. We will also post all attributes of the relationship between Sale and Stage (stageDate) into the new relation.

SaleStage(saleNo, stageNo, stageDate)

SaleStage = {saleNo:D(saleNo), stageNo:D(stageNo), stageDate:D(stageDate)}

Where

D(saleNo) = {saleNo|saleNo is an INT AND MaxLength = 11}
D(stageNo) = {stageNo|stageNo is an INT AND MaxLength = 11}
D(stageDate) = {stageDate|stageDate is a TIMESTAMP}

3.3 - Normalization

1NF

For 1NF we begin with one large table.

Pipeline (leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName, saleNo, value, commission, stageNo, stageDate, stageName, productNo, productName)

Functional Dependancies

We examine the data of the relation in 1NF to determine the functional dependancies. By maximising the right-hand-side and minimizing the left-hand-side of the functional dependancies we arrive with the following full functional dependancies:

partnerNo ® partnerName, partnerSurname

departmentNo ® departmentName

leadNo ® leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName

clientContact ® leadNo, leadDate, leadComment, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName

saleNo ® value, commission, productNo, leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName, productNo, productName

stageNo ® stageName

productNo ® productName

saleNo, stageNo ® stageDate, stageName, value, commission, productNo, leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName, productNo, productName

We will now try and determine the candidate keys, primary key and secondary key(if any) of the relation.

Step 1

By testing the determinants of the FD's we are able to find the following super key:

saleNo, stageNo+ = (stageNo, saleNo, leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName, value, commission, stageDate, stageName, productNo, productName)

After performing (partnerNo+, departmentNo+, leadNo+, clientContact+, saleNo+, stageNo+, productNo+) we find no other superkeys.

Step 2

Based on the determinants, the following are definately superkeys:

(Pipeline - partnerNo+) ⋃ partnerNo = saleNo, stageNo, ...

(Pipeline - departmentNo+) ⋃ departmentNo = saleNo, stageNo, ...

(Pipeline - leadNo+) ⋃ leadNo = saleNo, stageNo, ...

(Pipeline - clientContact+) ⋃ clientContact = saleNo, stageNo, ...

(Pipeline - saleNo+) ⋃ saleNo = saleNo, stageNo, ...

(Pipeline - stageNo+) ⋃ stageNo = saleNo, stageNo, ...

(Pipeline - productNo+) ⋃ productNo = saleNo, stageNo, ...

Step 3

For each superkey in Step 2, {saleNo, stageNo} is also a superkey. No subset of {saleNo, stageNo} is a superkey, therefore {saleNo, stageNo} is a candidate key. As no other candidate keys exist, {saleNo, stageNo} is the primary key and no secondary key exists for the relation.

2NF

To move to 2NF we must ensure that all non-primary key attributes are fully functionally dependant on the primary key of the relation.

stageName is not fully functionally dependant on saleNo, stageNo. It is only fully dependant on stageNo, therefore we must remove the partially dependant attribute by placing it in a new relation along with a copy of its determinant (stageNo). The new relations appear as follows:

> Stage (stageNo, stageName)
>
> SaleStage (stageNo, saleNo, leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName, value, commission, stageDate, productNo, productName) {FK stageNo}

leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName, value, commission, productNo and productName are not fully functionally dependant on saleNo, stageNo. They are only functionally dependant on saleNo. Therefore we must remove all the partially dependant attributes by placing them in a new relation along with a copy of their determinant (saleNo). The new relations appear as follows:

> SaleStage (saleNo, stageNo, stageDate) {FK stageNo, saleNo}
>
> Sale (saleNo, leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName, value, commission, productNo, productName)

We arrive at 2NF with the following three tables:

> Stage (stageNo, stageName)
>
> SaleStage (saleNo, stageNo, stageDate){FK stageNo, saleNo}
>
> Sale (saleNo, value, commission, productNo, productName, leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName)

3NF

To move to 3NF we must ensure that no non-primary key attributes are transitively dependant on the primary key.

productName is transitively dependant on saleNo through productNo. Therefore we must remove the transitively dependant attribute by placing the attribute in a new relation along with a copy of its determinant(productNo). The relations will appear as follows:

> Sale (saleNo, value, commission, productNo, leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName){FK productNo}

Product (productNo, productName)

leadDate, leadComment, clientcontact, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName are transitively dependant on saleNo through either leadNo or clientContact. Therefore we must remove the transitively dependant attributes by placing the attributes in a new relation along with a copy of their determinant(leadNo). It should be noted that clientContact is also a determinant and this attribute will act as secondary key to the new relation. The relations will appear as follows:

Sale (saleNo, value, commission, productNo, leadNo){FK leadNo}

Lead (leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, partnerName, partnerSurname, departmentNo, departmentName){2K clientContact}

partnerName, partnerSurname are transitively dependant on leadNo through partnerNo. Therefore we must remove the transitively dependant attributes by placing the attributes in a new relation along with a copy of their determinant(partnerNo). The relations will appear as follows:

Lead (leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, departmentNo, departmentName){FK partnerNo}{2K clientContact}

Partner (partnerNo, partnerName, partnerSurname)

departmentName is transitively dependant on leadNo through departmentNo. Therefore we must remove the transitively dependant attribute by placing the attribute in a new relation along with a copy of its determinant(departmentNo). The relations will appear as follows:

Lead (leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, departmentNo){FK partnerNo, departmentNo}{2K clientContact}

Department(departmentNo, departmentName)

We arrive at 3NF with the following seven tables:

Stage (stageNo, stageName)

SaleStage (saleNo, stageNo, stageDate){FK saleNo, stageNo}

Product (productNo, productName)

Sale (saleNo, value, commission, productNo, leadNo){FK leadNo}

Partner (partnerNo, partnerName, partnerSurname)

Department(departmentNo, departmentName)

Lead (leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, departmentNo){FK partnerNo, departmentNo}{2K clientContact}

BCNF

We must check that the functional dependancies of all relations are also candidate keys. In this case they are and the seven relations are already in BCNF.

4 - Database Queries

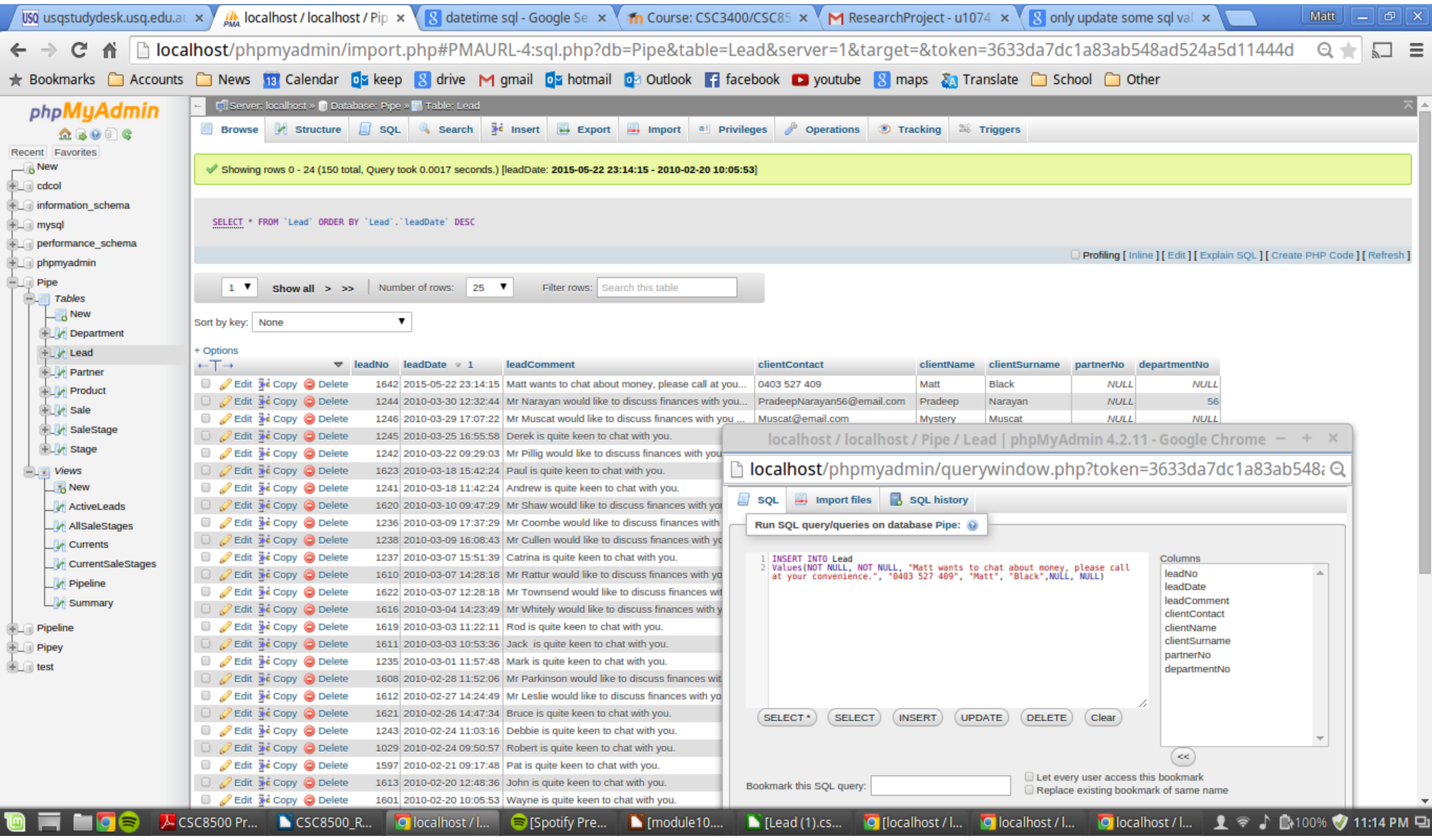
For additional queries, please refer to the Views within the Pipe database or the various .ini files within the Pipe application folder.

1 - Update Lead

SQL

INSERT INTO Lead
Values (NOT NULL, NOT NULL, "Matt wants to chat about money, please call at your convenience.", "0403 527 409", "Matt", "Black",NULL, NULL);

This will create a new lead tuple in the Lead table. The "NOT NULL, NOT NULL" for attributes 1 and 2 will allow for the Auto-increment INT for leadNo and timestamp for leadDate to be automatically created. The "NULL, NULL" in the final attributes indicate that there was no referral partner or referral department for this lead. This operation would be performed each time a new lead enters the database. All values would change based on the user input except the leadNo and leadDate attributes which are automatically created by the database.



✓ Showing rows 0 - 24 (150 total, Query took 0.0017 seconds.) [leadDate: **2015-05-22 23:14:15 - 2010-02-20 10:05:53**]

```
SELECT * FROM `Lead` ORDER BY `Lead`.`leadDate` DESC
```

leadNo	leadDate	leadComment	clientContact	clientName	clientSurname	partnerNo	departmentNo
1642	2015-05-22 23:14:15	Matt wants to chat about money, please call at you...	0403 527 409	Matt	Black	NULL	NULL
1244	2010-03-30 12:32:44	Mr Narayan would like to discuss finances with you...	PradeepNarayan56@email.com	Pradeep	Narayan	NULL	56
1246	2010-03-29 17:07:22	Mr Muscat would like to discuss finances with you ...	Muscat@email.com	Mystery	Muscat	NULL	NULL
1245	2010-03-25 16:55:58	Derek is quite keen to chat with you.					
1242	2010-03-22 09:29:03	Mr Pillig would like to discuss finances with you					
1623	2010-03-18 15:42:24	Paul is quite keen to chat with you.					
1241	2010-03-18 11:42:24	Andrew is quite keen to chat with you.					
1620	2010-03-10 09:47:29	Mr Shaw would like to discuss finances with yo					
1236	2010-03-09 17:37:29	Mr Coombe would like to discuss finances with					
1238	2010-03-09 16:08:43	Mr Cullen would like to discuss finances with yo					
1237	2010-03-07 15:51:39	Catrina is quite keen to chat with you.					
1610	2010-03-07 14:28:18	Mr Rattur would like to discuss finances with yo					
1622	2010-03-07 12:28:18	Mr Townsend would like to discuss finances wit					
1616	2010-03-04 14:23:49	Mr Whitely would like to discuss finances with y					
1619	2010-03-03 11:22:11	Rod is quite keen to chat with you.					
1611	2010-03-03 10:53:36	Jack is quite keen to chat with you.					
1235	2010-03-01 11:57:48	Mark is quite keen to chat with you.					
1608	2010-02-28 11:52:06	Mr Parkinson would like to discuss finances wit					
1612	2010-02-27 14:24:49	Mr Leslie would like to discuss finances with yo					
1621	2010-02-26 14:47:34	Bruce is quite keen to chat with you.					
1243	2010-02-24 11:03:16	Debbie is quite keen to chat with you.					
1029	2010-02-24 09:50:57	Robert is quite keen to chat with you.					
1597	2010-02-21 09:17:48	Pat is quite keen to chat with you.					
1613	2010-02-20 12:48:36	John is quite keen to chat with you.					
1601	2010-02-20 10:05:53	Wayne is quite keen to chat with you.					

localhost / localhost / Pipe / Lead | phpMyAdmin 4.2.11 - Google Chrome

Run SQL query/queries on database Pipe:

```
INSERT INTO Lead
Values(NOT NULL, NOT NULL, "Matt wants to chat about money, please call at your convenience.", "0403 527 409", "Matt", "Black",NULL, NULL)
```

Columns: leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, departmentNo

Screenshot of the output from the above SQL statement

2 - Currents View

SQL

```
CREATE VIEW Currents AS
SELECT saleNo, MAX(stageNo) as stageNo
FROM SaleStage
GROUP BY saleNo;
```

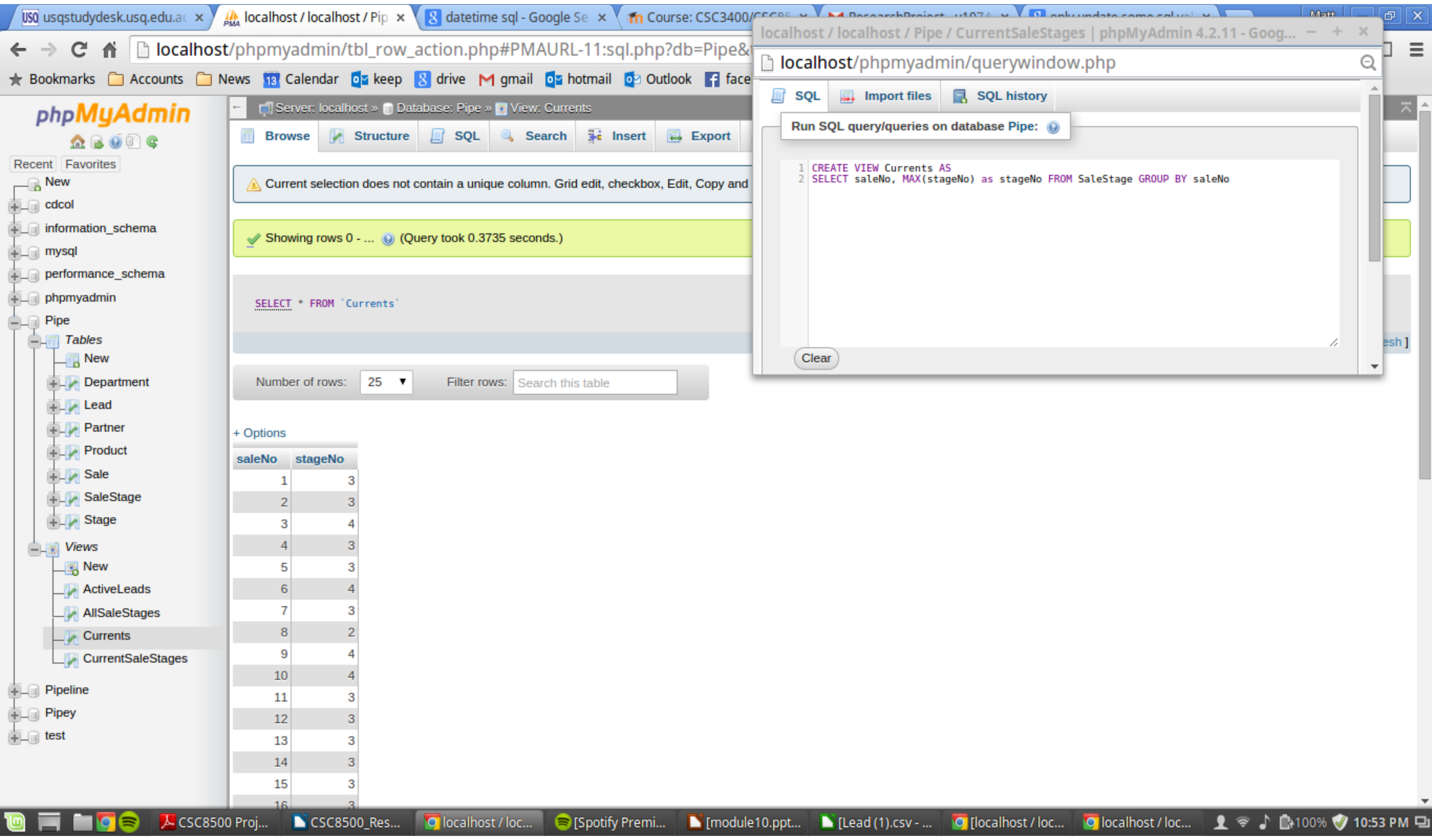
This statement creates a view from the select clause. The purpose of this statement is to indicate what is the highest stageNo for each saleNo. This is important as a sale can only move forward in stage, by knowing the highest stageNo we can know the current stage of each sale.

Relational Algebra

$\rho_{Currents}(saleNo, stageNo)_{saleNo\ \ MAX\ stageNo}(SaleStage)$

Relational Calculus

{S1| ((∃S1)(SaleStage)((~∃S2)(SaleStage(S2))(S2.stageNo > S1.stageNo))))



⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and

✓ Showing rows 0 - ... (Query took 0.3735 seconds.)

```
SELECT * FROM `Currents`
```

Number of rows: 25 Filter rows: Search this table

```
CREATE VIEW Currents AS
SELECT saleNo, MAX(stageNo) as stageNo FROM SaleStage GROUP BY saleNo
```

saleNo	stageNo
1	3
2	3
3	4
4	3
5	3
6	4
7	3
8	2
9	4
10	4
11	3
12	3
13	3
14	3
15	3
16	3

localhost/phpmyadmin/tbl_row_action.php#PMAURL-11:sql.php?db=Pipe&
localhost/phpmyadmin/querywindow.php
Server: localhost » Database: Pipe » View: Currents
Browse
Structure
SQL
Search
Insert
Export
Import files
SQL history
Run SQL query/queries on database Pipe:
Clear
+ Options
Pipe
Tables
Department
Lead
Partner
Product
Sale
SaleStage
Stage
Views
ActiveLeads
AllSaleStages
Currents
CurrentSaleStages

Screenshot of the output from the above SQL statement

3 - Select all Lead tuples for a department

From the following two statements we are able to view all tuples from the table Lead whose departmentNo is 48. This is useful as we may want to know which leads have come from a single department.

a)

SQL

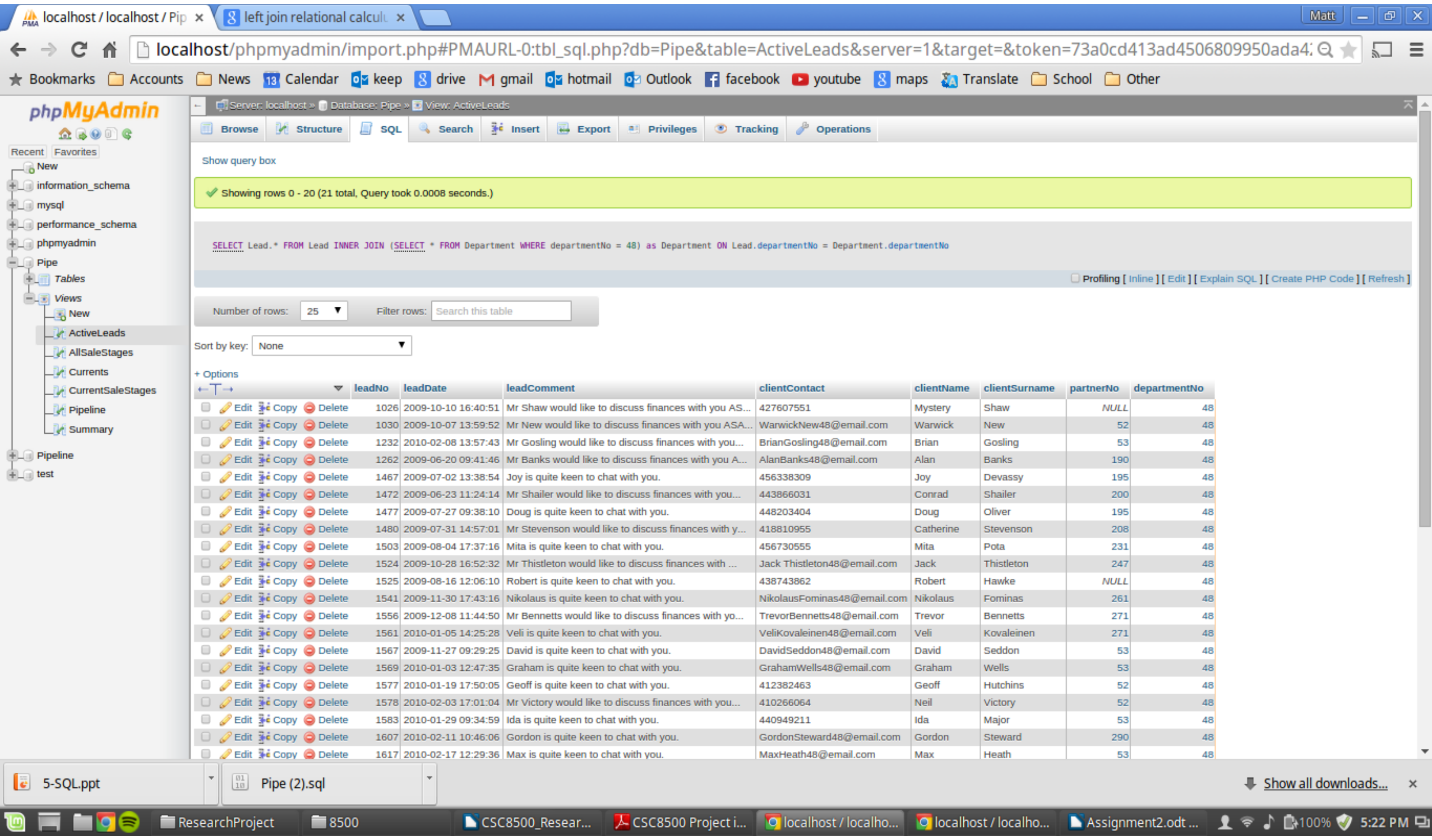
```
SELECT Lead.*
FROM Lead
INNER JOIN (SELECT *
            FROM Department
            WHERE departmentNo = 48) as Department
ON Lead.departmentNo = Department.departmentNo
```

Relational Algebra

$\pi_{\text{leadNo, leadDate, leadComment, clientName, clientSurname, partnerNo, departmentNo}}(\text{Lead} \bowtie_{\text{Lead.departmentNo=d.departmentNo}}(\sigma_{\text{departmentNo=48}}(\text{Department})))$

Relational Calculus

{L | Lead(L) ($_{\exists}$D) (Department(D) (L.departmentNo = D.departmentNo) D.departmentNo = 48)}



Showing rows 0 - 20 (21 total, Query took 0.0008 seconds.)

```
SELECT Lead.* FROM Lead INNER JOIN (SELECT * FROM Department WHERE departmentNo = 48) as Department ON Lead.departmentNo = Department.departmentNo
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

leadNo	leadDate	leadComment	clientContact	clientName	clientSurname	partnerNo	departmentNo
1026	2009-10-10 16:40:51	Mr Shaw would like to discuss finances with you AS...	427607551	Mystery	Shaw	NULL	48
1030	2009-10-07 13:59:52	Mr New would like to discuss finances with you ASA...	WarwickNew48@email.com	Warwick	New	52	48
1232	2010-02-08 13:57:43	Mr Gosling would like to discuss finances with you...	BrianGosling48@email.com	Brian	Gosling	53	48
1262	2009-06-20 09:41:46	Mr Banks would like to discuss finances with you A...	AlanBanks48@email.com	Alan	Banks	190	48
1467	2009-07-02 13:38:54	Joy is quite keen to chat with you.	456338309	Joy	Devassy	195	48
1472	2009-06-23 11:24:14	Mr Shailer would like to discuss finances with you...	443866031	Conrad	Shailer	200	48
1477	2009-07-27 09:38:10	Doug is quite keen to chat with you.	448203404	Doug	Oliver	195	48
1480	2009-07-31 14:57:01	Mr Stevenson would like to discuss finances with y...	418810955	Catherine	Stevenson	208	48
1503	2009-08-04 17:37:16	Mita is quite keen to chat with you.	456730555	Mita	Pota	231	48
1524	2009-10-28 16:52:32	Mr Thistleton would like to discuss finances with ...	Jack Thistleton48@email.com	Jack	Thistleton	247	48
1525	2009-08-16 12:06:10	Robert is quite keen to chat with you.	438743862	Robert	Hawke	NULL	48
1541	2009-11-30 17:43:16	Nikolaus is quite keen to chat with you.	NikolausFominas48@email.com	Nikolaus	Fominas	261	48
1556	2009-12-08 11:44:50	Mr Bennetts would like to discuss finances with yo...	TrevorBennetts48@email.com	Trevor	Bennetts	271	48
1561	2010-01-05 14:25:28	Veli is quite keen to chat with you.	VeliKovaleinen48@email.com	Veli	Kovaleinen	271	48
1567	2009-11-27 09:29:25	David is quite keen to chat with you.	DavidSeddon48@email.com	David	Seddon	53	48
1569	2010-01-03 12:47:35	Graham is quite keen to chat with you.	GrahamWells48@email.com	Graham	Wells	53	48
1577	2010-01-19 17:50:05	Geoff is quite keen to chat with you.	412382463	Geoff	Hutchins	52	48
1578	2010-02-03 17:01:04	Mr Victory would like to discuss finances with you...	410266064	Neil	Victory	52	48
1583	2010-01-29 09:34:59	Ida is quite keen to chat with you.	440949211	Ida	Major	53	48
1607	2010-02-11 10:46:06	Gordon is quite keen to chat with you.	GordonSteward48@email.com	Gordon	Steward	290	48
1617	2010-02-17 12:29:36	Max is quite keen to chat with you.	MaxHeath48@email.com	Max	Heath	53	48

Screenshot of the output from the above SQL statement

b)

SQL

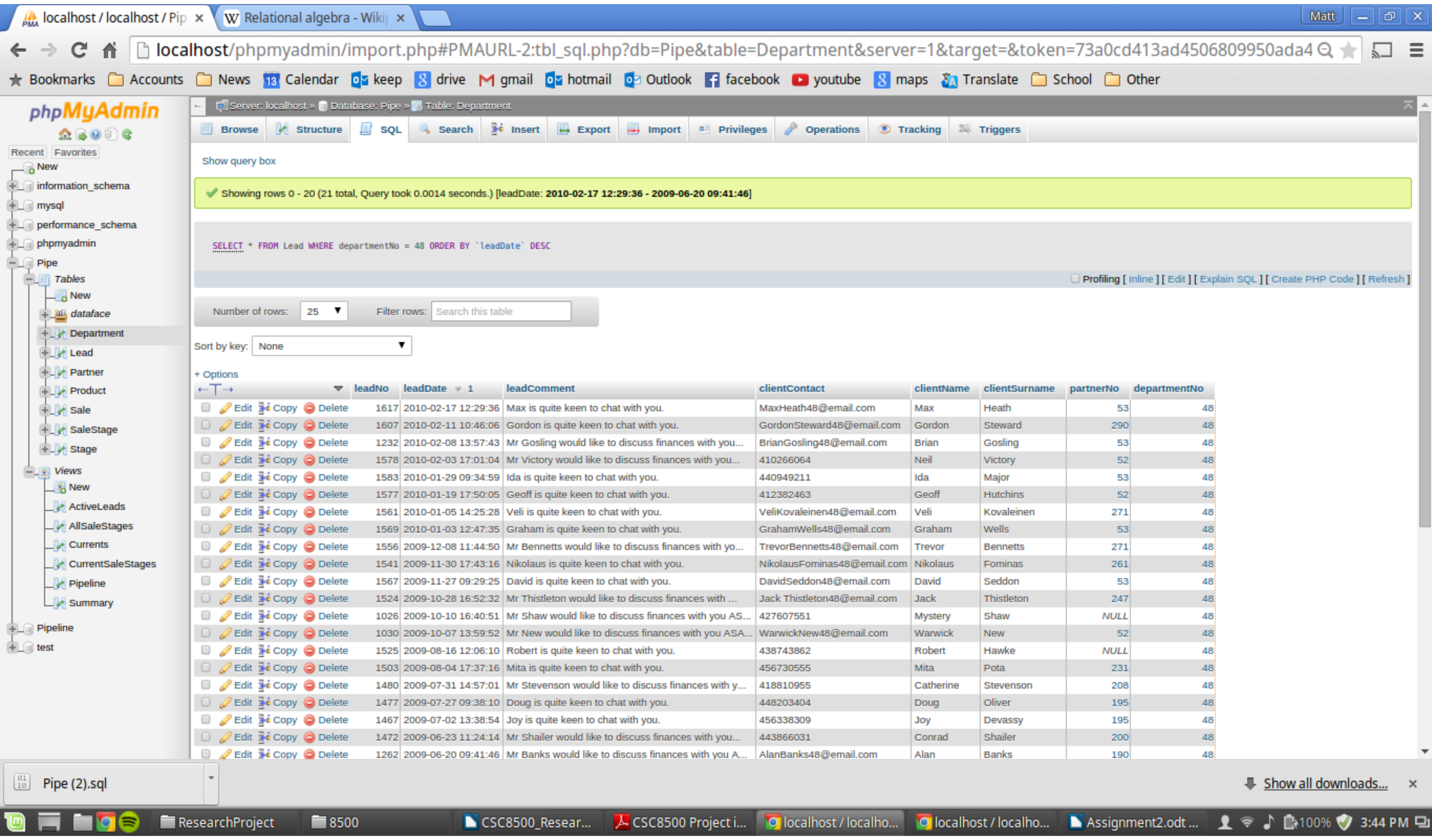
```
SELECT *
FROM Lead
WHERE departmentNo = 48;
```

Relational Algebra

$\sigma_{departmentNo=48}(Lead)$

Relational Calculus

{L | Lead(L) L.departmentNo = 48}



Show query box

✓ Showing rows 0 - 20 (21 total, Query took 0.0014 seconds.) [leadDate: **2010-02-17 12:29:36 - 2009-06-20 09:41:46**]

```
SELECT * FROM Lead WHERE departmentNo = 48 ORDER BY `leadDate` DESC
```

Number of rows: 25 Filter rows: Search this table

Sort by key: None

leadNo	leadDate	leadComment	clientContact	clientName	clientSurname	partnerNo	departmentNo
1617	2010-02-17 12:29:36	Max is quite keen to chat with you.	MaxHeath48@email.com	Max	Heath	53	48
1607	2010-02-11 10:46:06	Gordon is quite keen to chat with you.	GordonSteward48@email.com	Gordon	Steward	290	48
1232	2010-02-08 13:57:43	Mr Gosling would like to discuss finances with you...	BrianGosling48@email.com	Brian	Gosling	53	48
1578	2010-02-03 17:01:04	Mr Victory would like to discuss finances with you...	410266064	Neil	Victory	52	48
1583	2010-01-29 09:34:59	Ida is quite keen to chat with you.	440949211	Ida	Major	53	48
1577	2010-01-19 17:50:05	Geoff is quite keen to chat with you.	412382463	Geoff	Hutchins	52	48
1561	2010-01-05 14:25:28	Veli is quite keen to chat with you.	VeliKovaleinen48@email.com	Veli	Kovaleinen	271	48
1569	2010-01-03 12:47:35	Graham is quite keen to chat with you.	GrahamWells48@email.com	Graham	Wells	53	48
1556	2009-12-08 11:44:50	Mr Bennetts would like to discuss finances with yo...	TrevorBennetts48@email.com	Trevor	Bennetts	271	48
1541	2009-11-30 17:43:16	Nikolaus is quite keen to chat with you.	NikolausFominas48@email.com	Nikolaus	Fominas	261	48
1567	2009-11-27 09:29:25	David is quite keen to chat with you.	DavidSeddon48@email.com	David	Seddon	53	48
1524	2009-10-28 16:52:32	Mr Thistleton would like to discuss finances with ...	Jack Thistleton48@email.com	Jack	Thistleton	247	48
1026	2009-10-10 16:40:51	Mr Shaw would like to discuss finances with you AS...	427607551	Mystery	Shaw	NULL	48
1030	2009-10-07 13:59:52	Mr New would like to discuss finances with you ASA...	WarwickNew48@email.com	Warwick	New	52	48
1525	2009-08-16 12:06:10	Robert is quite keen to chat with you.	438743862	Robert	Hawke	NULL	48
1503	2009-08-04 17:37:16	Mita is quite keen to chat with you.	456730555	Mita	Pota	231	48
1480	2009-07-31 14:57:01	Mr Stevenson would like to discuss finances with y...	418810955	Catherine	Stevenson	208	48
1477	2009-07-27 09:38:10	Doug is quite keen to chat with you.	448203404	Doug	Oliver	195	48
1467	2009-07-02 13:38:54	Joy is quite keen to chat with you.	456338309	Joy	Devassy	195	48
1472	2009-06-23 11:24:14	Mr Shailer would like to discuss finances with you...	443866031	Conrad	Shailer	200	48
1262	2009-06-20 09:41:46	Mr Banks would like to discuss finances with you A...	AlanBanks48@email.com	Alan	Banks	190	48

Screenshot of the output from the above SQL statement

4 - ActiveLeads View

SQL

CREATE VIEW ActiveLeads AS
SELECT * FROM Lead l
WHERE l.leadNo NOT IN(SELECT leadNo FROM Sale)

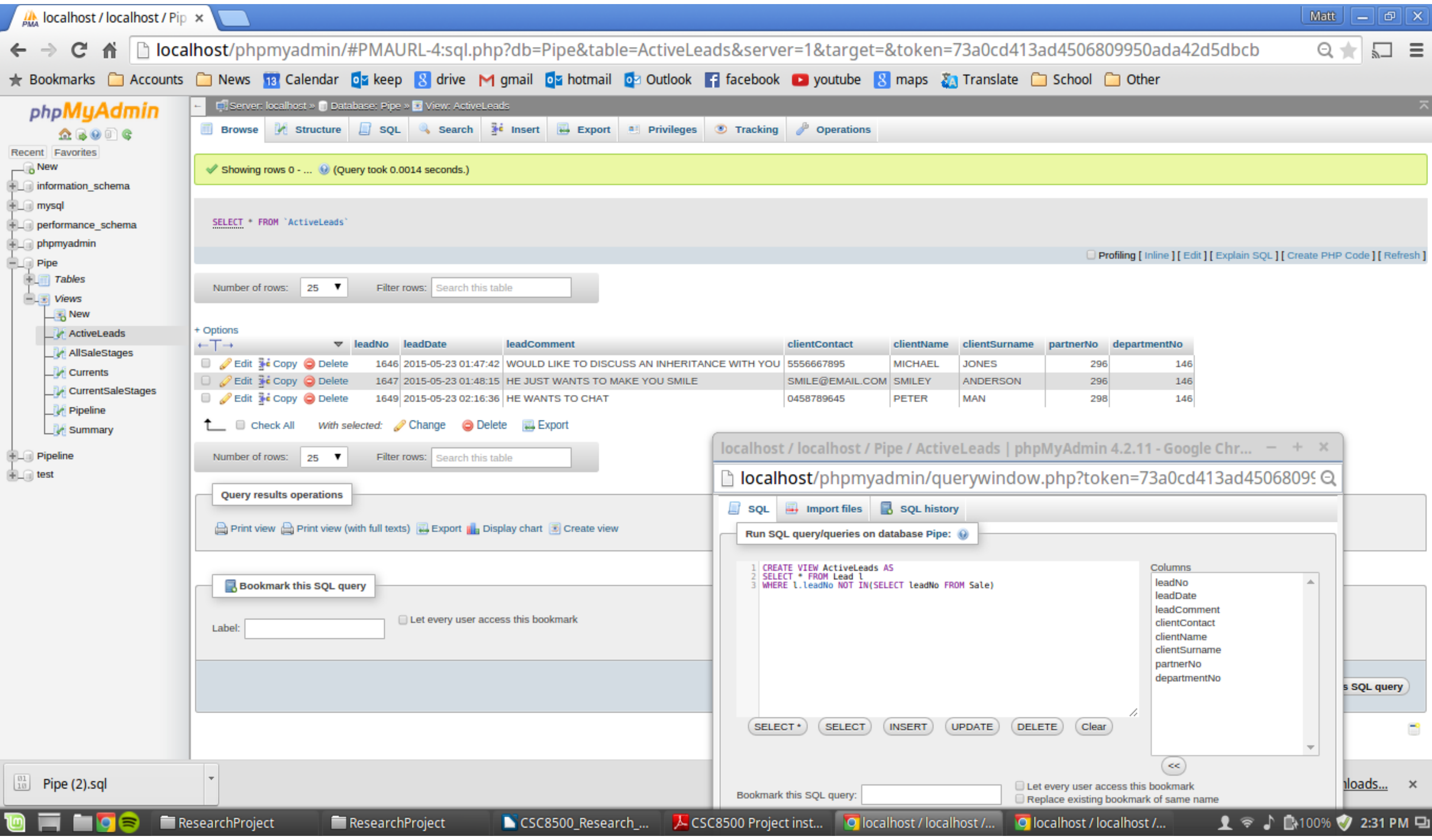
This view is very useful as it allows us to see which leads have not yet resulted in a sale. We want to know which leads have not been made a sales proposal yet so we can book an appointment to actually make the sale proposal. This view is the main way for a user to know who to contact for initial sales proposals.

Relational Algebra

$\pi_{\text{leadNo}}(\text{Lead}) - \pi_{\text{leadNo}}(\text{Sale})$

Relational Calculus

{L | Lead(L) ((~∃S) (Sale(S) (S.leadNo = L.leadNo)))}



Server: localhost » Database: Pipe » View: ActiveLeads

✓ Showing rows 0 - ... (Query took 0.0014 seconds.)

```
SELECT * FROM `ActiveLeads`
```

Number of rows: 25 Filter rows: Search this table

leadNo	leadDate	leadComment	clientContact	clientName	clientSurname	partnerNo	departmentNo
1646	2015-05-23 01:47:42	WOULD LIKE TO DISCUSS AN INHERITANCE WITH YOU	5556667895	MICHAEL	JONES	296	146
1647	2015-05-23 01:48:15	HE JUST WANTS TO MAKE YOU SMILE	SMILE@EMAIL.COM	SMILEY	ANDERSON	296	146
1649	2015-05-23 02:16:36	HE WANTS TO CHAT	0458789645	PETER	MAN	298	146

localhost / localhost / Pipe / ActiveLeads | phpMyAdmin 4.2.11 - Google Chr...

Run SQL query/queries on database Pipe:

```
CREATE VIEW ActiveLeads AS
SELECT * FROM Lead l
WHERE l.leadNo NOT IN(SELECT leadNo FROM Sale)
```

Columns: leadNo, leadDate, leadComment, clientContact, clientName, clientSurname, partnerNo, departmentNo

Screenshot of the output from the above SQL statement

5 - Select with multiple left joins

SQL

```
SELECT l.*, p.partnerName as partnerName, p.partnerSurname, d.departmentName
FROM Lead l
LEFT JOIN (SELECT partnerNo, partnerName, partnerSurname
                    FROM Partner) as p
ON l.partnerNo = p.partnerNo
LEFT JOIN (SELECT departmentNo, departmentName
                    FROM Department) as d
ON l.departmentNo = d.departmentNo
```

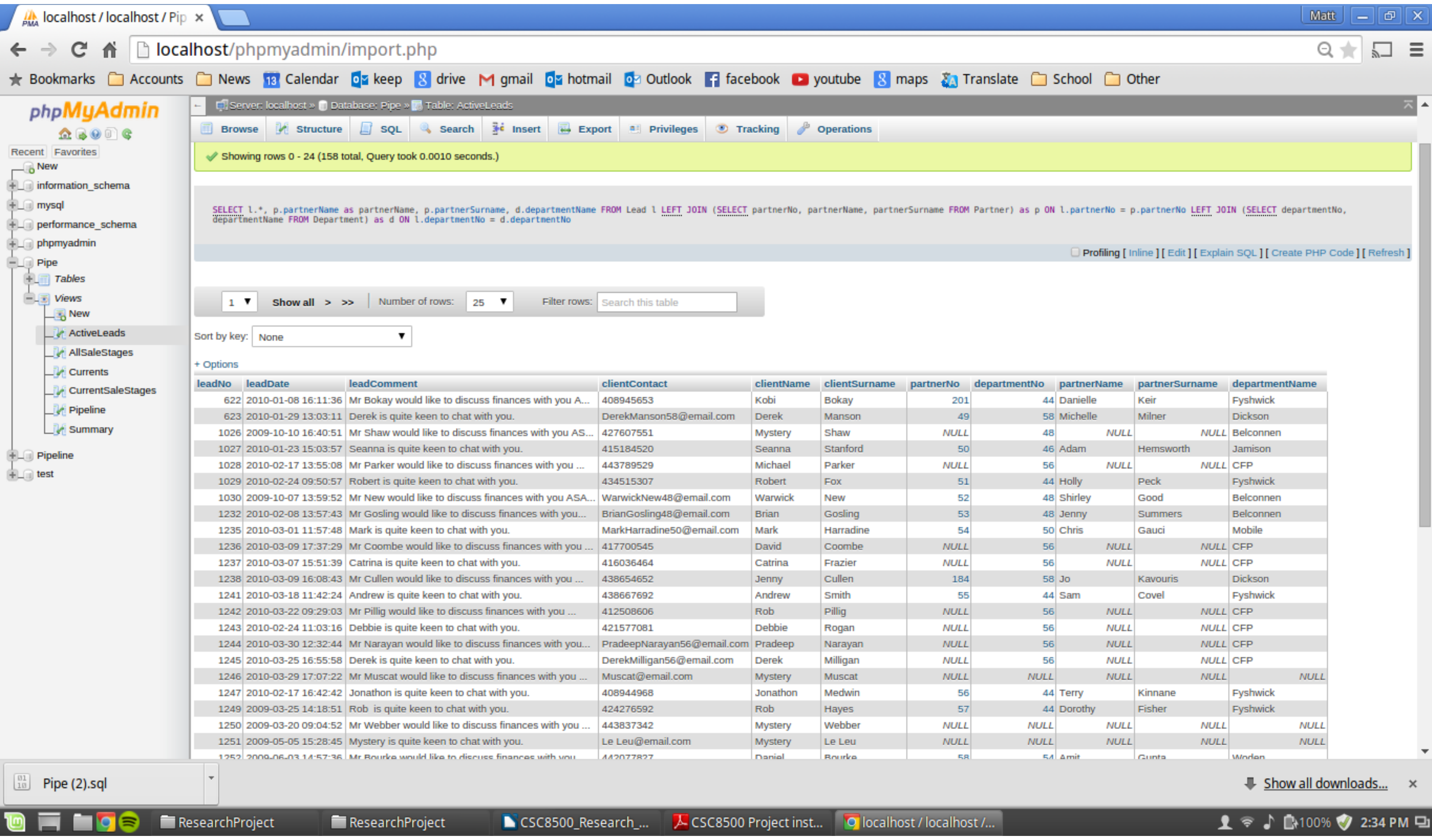
This select clause is useful as it allows us to see the partnerName, partnerSurname and departmentName(if they exist) for each lead. This is important as we want to visualize who it is who sent the lead (for training and feedback purposes). This is easier for the user to view than only seeing partnerNo and departmentNo (which we would see in Lead).

Relational Algebra

$((\pi_{\text{leadNo, leadDate, leadComment, clientName, clientSurname, partnerNo, departmentNo}}(\text{Lead}))⟕(\pi_{\text{partnerNo, partnerName, partnerSurname}}(\text{Partner})))⟕\pi_{\text{departmentNo, departmentName}}(\text{Department})$

Relational Calculus

{L, P.partnerName, P.partnerSurname, D.departmentName | Lead(L) (($\exists$P) (Partner(P) (P.partnerNo = L.partnerNo)) (($\exists$D) (Department(D) (D.departmentNo = L.departmentNo))}


localhost / localhost / Pip
localhost/phpmyadmin/import.php
Bookmarks Accounts News Calendar keep drive gmail hotmail Outlook facebook youtube maps Translate School Other
phpMyAdmin
Recent Favorites
New
information_schema
mysql
performance_schema
phpmyadmin
Pipe
Tables
Views
New
ActiveLeads
AllSaleStages
Currents
CurrentSaleStages
Pipeline
Summary
Pipeline
test
Server: localhost » Database: Pipe » Table: ActiveLeads
Browse Structure SQL Search Insert Export Privileges Tracking Operations
Profiling [Inline] [Edit] [Explain SQL] [Create PHP Code] [Refresh]
1 Show all > >> Number of rows: 25 Filter rows: Search this table
Sort by key: None
+ Options
Pipe (2).sql
Show all downloads...
ResearchProject ResearchProject CSC8500_Research_... CSC8500 Project inst... localhost / localhost /...
100% 2:34 PM
Matt

Showing rows 0 - 24 (158 total, Query took 0.0010 seconds.)

```
SELECT l.*, p.partnerName as partnerName, p.partnerSurname, d.departmentName FROM Lead l LEFT JOIN (SELECT partnerNo, partnerName, partnerSurname FROM Partner) as p ON l.partnerNo = p.partnerNo LEFT JOIN (SELECT departmentNo, departmentName FROM Department) as d ON l.departmentNo = d.departmentNo
```

leadNo	leadDate	leadComment	clientContact	clientName	clientSurname	partnerNo	departmentNo	partnerName	partnerSurname	departmentName
622	2010-01-08 16:11:36	Mr Bokay would like to discuss finances with you A...	408945653	Kobi	Bokay	201	44	Danielle	Keir	Fyshwick
623	2010-01-29 13:03:11	Derek is quite keen to chat with you.	DerekManson58@email.com	Derek	Manson	49	58	Michelle	Milner	Dickson
1026	2009-10-10 16:40:51	Mr Shaw would like to discuss finances with you AS...	427607551	Mystery	Shaw	NULL	48	NULL	NULL	Belconnen
1027	2010-01-23 15:03:57	Seanna is quite keen to chat with you.	415184520	Seanna	Stanford	50	46	Adam	Hemsworth	Jamison
1028	2010-02-17 13:55:08	Mr Parker would like to discuss finances with you ...	443789529	Michael	Parker	NULL	56	NULL	NULL	CFP
1029	2010-02-24 09:50:57	Robert is quite keen to chat with you.	434515307	Robert	Fox	51	44	Holly	Peck	Fyshwick
1030	2009-10-07 13:59:52	Mr New would like to discuss finances with you ASA...	WarwickNew48@email.com	Warwick	New	52	48	Shirley	Good	Belconnen
1232	2010-02-08 13:57:43	Mr Gosling would like to discuss finances with you...	BrianGosling48@email.com	Brian	Gosling	53	48	Jenny	Summers	Belconnen
1235	2010-03-01 11:57:48	Mark is quite keen to chat with you.	MarkHarradine50@email.com	Mark	Harradine	54	50	Chris	Gauci	Mobile
1236	2010-03-09 17:37:29	Mr Coombe would like to discuss finances with you ...	417700545	David	Coombe	NULL	56	NULL	NULL	CFP
1237	2010-03-07 15:51:39	Catrina is quite keen to chat with you.	416036464	Catrina	Frazier	NULL	56	NULL	NULL	CFP
1238	2010-03-09 16:08:43	Mr Cullen would like to discuss finances with you ...	438654652	Jenny	Cullen	184	58	Jo	Kavouris	Dickson
1241	2010-03-18 11:42:24	Andrew is quite keen to chat with you.	438667692	Andrew	Smith	55	44	Sam	Covel	Fyshwick
1242	2010-03-22 09:29:03	Mr Pillig would like to discuss finances with you ...	412508606	Rob	Pillig	NULL	56	NULL	NULL	CFP
1243	2010-02-24 11:03:16	Debbie is quite keen to chat with you.	421577081	Debbie	Rogan	NULL	56	NULL	NULL	CFP
1244	2010-03-30 12:32:44	Mr Narayan would like to discuss finances with you...	PradeepNarayan56@email.com	Pradeep	Narayan	NULL	56	NULL	NULL	CFP
1245	2010-03-25 16:55:58	Derek is quite keen to chat with you.	DerekMilligan56@email.com	Derek	Milligan	NULL	56	NULL	NULL	CFP
1246	2010-03-29 17:07:22	Mr Muscat would like to discuss finances with you ...	Muscat@email.com	Mystery	Muscat	NULL	NULL	NULL	NULL	NULL
1247	2010-02-17 16:42:42	Jonathon is quite keen to chat with you.	408944968	Jonathon	Medwin	56	44	Terry	Kinnane	Fyshwick
1249	2009-03-25 14:18:51	Rob is quite keen to chat with you.	424276592	Rob	Hayes	57	44	Dorothy	Fisher	Fyshwick
1250	2009-03-20 09:04:52	Mr Webber would like to discuss finances with you ...	443837342	Mystery	Webber	NULL	NULL	NULL	NULL	NULL
1251	2009-05-05 15:28:45	Mystery is quite keen to chat with you.	Le Leu@email.com	Mystery	Le Leu	NULL	NULL	NULL	NULL	NULL
1252	2009-06-03 14:57:36	Mr Bourke would like to discuss finances with you...	442077827	Daniel	Bourke	58	54	Amit	Gupta	Woden

Screenshot of the output from the above SQL statement

5 - Implementation of Database System and Video Demonstration

Video Presentation

Watch the demo video here -> https://www.youtube.com/watch?v=nmizF2yYfKo

Installation Instructions

1. Installation video here -> youtube.com/watch?v=mRMSIl87LPc&feature=youtu.be
2. Open the Pipe_Install folder
3. Import the Pipe.sql file to a new MySql database named "Pipe"
4. Copy the folders "Pipe" and "xataface" to the htdocs folder in your xampp lampp folder ("opt -> lampp-> -> htdocs" for Linux users)
5. Ensure that the Pipe folder has full read and write access
6. In browser go to -----> localhost/xataface/installer.php
7. Select "Install a pre-built application"
8. Choose the Pipe.tar file within the Pipe_Install folder and proceed
9. Enter database name as "Pipe" and your MYSQL credentials (should be "root" and "") and proceed
10. Select Install on server
11. Enter host as "localhost"
12. Enter Path as "/Pipe/"
13. Enter your pc username and password
14. Click Test FTP connection, if successful proceed to next step
15. In browser go to localhost/Pipe/
16. Enjoy the Pipeline database system