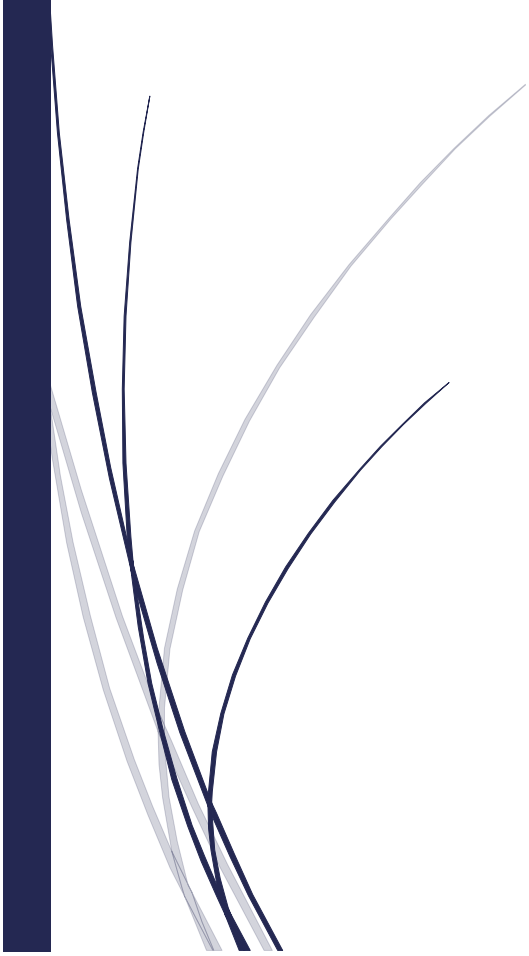
2017F Edition

Labs

V 1.7



Daniel Gaudreault
ALGONQUIN COLLEGE

Introduction to the Job lab approach

This course uses an “On-the-job” approach to lab work. Every task is planned to build up in a consistent “environment”. The premise is that you have been hired part-time by a company called ThinkCube. You will be working the IT department as a junior database developer. They are a company in the midst of rapid growth and individual effort is considered a primary attribute. A few key things to remember: When you see the word supervisor, it means your lab instructor. Each Lab of work is 1 lab.

Lab 1 – Installing your environment

Submission

At the end of the lab, approach your supervisor to demonstrate a working installation of all the software.

Tasks

1. Download applicable files from blackboards' Labs section.
    a. thinkcube.zip
    b. toaddatamodeler_XXX.msi
    select 32bit (X86) or 64bit (X64) version as applicable. Most modern PCs would is the 64bit edition.

Getting TOAD data modeler.

PLEASE NOTE, THAT THE SOFTWARE WILL RUN IN DEMO MODE FOR 30 DAYS, SO DON'T PANIC IF THERE IS A DELAY IN GETTING YOUR KEY. IF YOU DON"T HAVE YOU ID HANDY, STILL INSTALL TOAD AND DO THE REGISTRATION LATER.

2. First create an account at:

    http://www.toadworld.com/training/p/toad-world-academic-program

    You should be able to use any email address you want for this.

3. After registering and activating your account, go to:

    http://www.toadworld.com/academicprogram/Registration.aspx

    Fill in registration, using your Algonquin email address and mark yourself off as a student. You will then need proof of you being a student. According to their website valid forms of proof include:

    - Register with school provided email address

    OR scanned copy of (*choose one*):

    - Student ID card
    - Transcript
    - Tuition bill or statement

4. After getting your confirmation (this may take up to a day), generate a key for TOAD data modeler by going to:

    http://www.toadworld.com/AcademicProgram/Default.aspx

    Click get key, fill in the appropriate info. Download the software attached to this lab and use the key to activate.

Installing PostGreSQL.

5. Download the version 9.4.X or 9.5.x of PostGreSQL from
    http://www.enterprisedb.com/products-services-training/pgdownload#windows.
    If this link does not work, copy and paste the URL.
    This means DO NOT DOWNLOAD version 9.6. If unsure, check with your lab supervisor.

6. Run the installer accepting all defaults. Make sure to take note of what you set the administrative password to.
7. You do not need run the Stack Builder. If you launch the stack builder, cancel out of it.
8. Unzip the **thinkcube.zip** file, taking note of where you are extracting it to.
9. Launch pgAdmin III. It was installed as part of PostGreSQL.
10. Double click the entry for localhost.
11. Expand the Database tree leaf
12. Right click and select New Database
13. Give it the name of "thinkcube" (without the quote marks) and clicks OK.
14. Now right click on the newly created database. (thinkcube. It will have a red X on it at first.)
15. Select Restore.
16. Select the thinkcube.backup file that was extracted in step 8. Click Restore. Once the process is complete, close the window. This process may take several minutes depending on your computers' specifications.
17. Expand the tree as follows: thinkcube->schema->public->tables. You should now have several tables listed there. (11)
18. Right click on employees, select View data->View all rows. If you see records, close that window.
19. Go demo to the lab supervisor.

Lab 2 – Conceptual Diagramming

Introduction

The purpose of this lab is to help you develop your conceptual diagramming skills. You can use paper, Visio (Chen Database Notation), or https://erdplus.com.

Submission

You will submit a diagram exported as a PNG. Filename should be <Lastname>-<firstname>-lab2.png

Tasks

1. You will do this by diagramming a variety of small diagrams that will eventually build up to a full diagram.
    a. Simple Entity – draw a simply entity for each of the following items.
        i. A company
        ii. A office
        iii. An employee
    b. Entity with Attributes – add the following attributes to the diagrams you created above
        i. An employee has
            1. A name
            2. A date of birth
            3. A social insurance number
            4. A salary
        ii. An office has
            1. An address
            2. City
            3. Province
            4. Postal code
            5. Office spaces
        iii. A company has
            1. a name
            2. a telephone number
    c. Relations – Create relationships between the following
        i. A company has many offices
        ii. A company has many employees
        iii. An office can have many employees
    d. Associative entity – create an associative entity
        i. Create a new entity to contain skills
            1. Skills have a name
        ii. An employee can have multiple skills
        iii. Skill can be assignment to multiple employees

Lab 3 – Diagramming 1

Introduction

You will learn how to use diagramming software and then create a diagram from a written specification. You will be evaluated based on the accuracy of the final model, the relationships between objects and how well you followed the specified naming conventions.

To complete this lab, you will need to have Toad Data Modeler installed and have completed **Lab 2 –** successfully. If you have any problems with the actual procedures, refer to the previous lab.

Submission

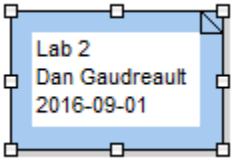
You will submit a diagram exported as a PNG. Filename should be <Lastname>-<firstname>-lab4.png

Tasks

Part 1

1. Load Toad Data Modeler & continue evaluation.
2. Click on File->New and select Project.
3. Select "Toad Data Modeler Project" and click Next.
4. As the name of the project, enter Labs and click Create
5. Click File->New and select Model (or clicks the new model icon).
6. Enter Lab 2 as the model name.
7. Double-click PostGreSQL 9.4 or PostGreSQL 9.5 (Doesn't really make a difference at this point). You will be prompted to enable that database platform. Click Yes.

8. In the tool strip, click the note icon  and then click somewhere on the workspace. Doube click the newly created note object.
    a. In the name field, enter "Diagram Label"
    b. In the text box enter the following:
       Lab 2
       <You Name>
       <Current Date>
    c. Click "Apply"
    d. Move the text box to the top left of the workspace.



9. Now to create a reference table. In the tool strip, click the new entity icon  . This can also be found under Objects->New menu or pressing CTRL+E
    a. Click somewhere on the diagram to create a new entity. Double click the new entity.
    b. In the Caption field, enter "genres". Note that the Name field updates to match it.
    c. Select the Attributes tab if it isn't already selected.
    d. Click on the Add button at the bottom of the window. This will create a new column/field. Now click Edit. Select "Do not show next time" and click yes on the dialog that pops up.
        i. In the Caption field, enter "id". Note that the Name field updates to match it.
        ii. In the data type field select "bigserial"
        iii. Check PRIMARY KEY checkbox
        iv. Click "OK+Add"

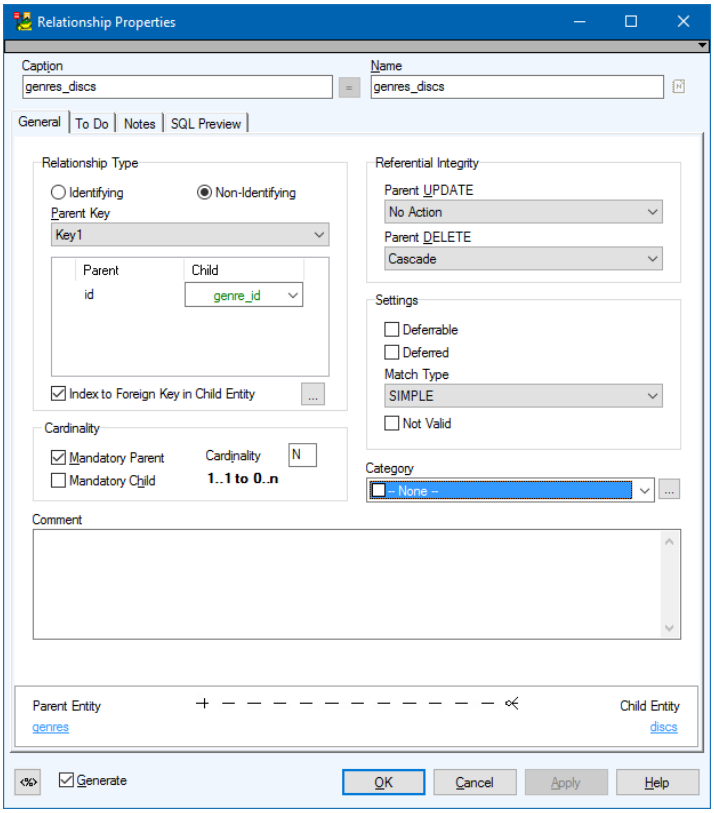
e. Create another field
   i. In the caption field, enter “name”
   ii. In the data type field select “character varying”
   iii. In the length field, enter 50
   iv. Check the NOT NULL checkbox
   v. Click “OK”

f. Click “OK”
   You should now have a table on your diagram that contains 2 fields.

10. Now to create a second reference table called types. Redo #9 except name the table “types”.
11. Create a third table. This is a regular table to contain detailed information.
    a. Redo #9, except the table is named “discs”. You can skip step “f” as you will be adding a few more fields.
    b. If you clicked the final “OK” in step 4, double click the discs table to bring up its properties.
    c. Add the following fields with their appropriate datatypes

Field name	Data type
artist	character varying(50) (Select “Character Varying(x)” and enter 50 for the “Length”)
release_date	date
synopsis	text

    d. Click “OK”. To close the diagram,
       You should now have 3 tables on your diagram
12. Now to create a relationship. In the tool strip, click the new “Non-identifying relationship” icon  . Again, this can also be found under the Objects->New menu.
    a. Click on the “genres” table and then click on the “discs” table.
    b. Edit the newly created field in the discs table to be called genre_id. (Double click the table and then double click the newly created “id” field that has a green key next to it, “OK” your way out of the dialogs)
    c. Double click the newly created relationship. This will open the Relationship Properties dialog.
    d. Change the caption to genres_discs.
    e. In the cardinality section,Check the Mandatory Parent checkbox.
    f. In the Referential Integrity section, change the “Parent DELETE” to “Cascade”.
    g. Click “OK”



13. Redo #12, this time connect “types” and “discs”. Make the caption “types_discs”.

14. Arrange the tables to a more compact layout by dragging the tables around.
15. Click the Model->Verify menu item. Ensure all tables are selected and click Save & Verify. Click Close.
    A verification log should have shown up at the bottom of the main window. Look for any errors.

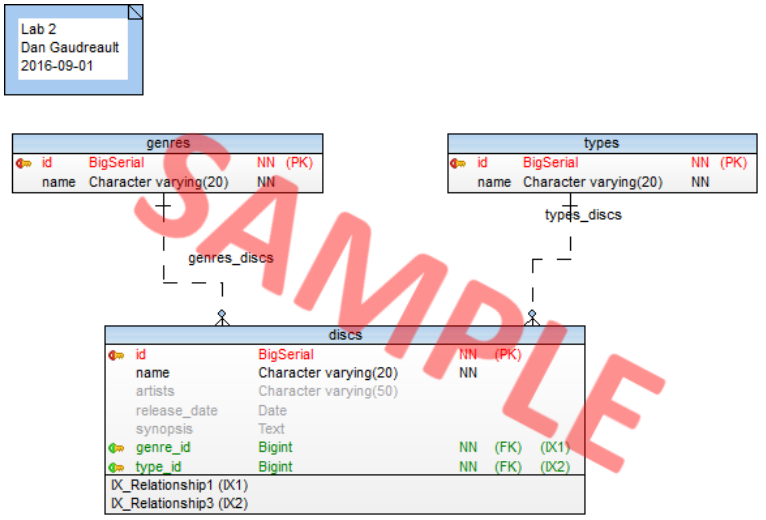


Figure 1 – Lab 3 sample

Part 2

16. Create the following reference tables. A reference table has normally 2 columns. A primary key and a descriptor column (called name or description).
    a. Products
    b. Versions
    c. Employees
17. Create a table whose name is derived from its' purpose. It is a table that tracks the combination of Products and Versions. It should contain 4 fields.
    a. The Primary Key
    b. An "Active" field that is a Boolean
    c. A foreign key that references Products
    d. A foreign key that references Versions
18. Create a table whose name is derived from its' purpose. It is a table that tracks the products a company has available for sale. It should contain the following fields
    a. The Primary Key
    b. A product name (normally this would be "name")
    c. An "Active" field that is a Boolean, default value of true
    d. A foreign key that references Product Versions
    e. Description (This can hold a lot of text)
    f. Suggested retail price (normally known as sell price or msrp). This is probably a numeric that contains 7 digits of precision with 2 reserved for decimal scale.
    g. Maximum discount. This is probably a numeric that contains 5 digits of precision with 2 reserved for decimal scale.
    h. Minimum sell quantity. This should probably be an integer of some sort.
19. Create a table that has to do with orders. It should have the following fields:
    a. The primary key
    b. An order timestamp, default value of "now()"
    c. A timestamp that contains when the order was shipped
    d. A foreign key that references Employees
20. Create a table that has to do with the products in an order. This would normally be called a derivative of "Order Lines". It should have the following fields:
    a. The primary key
    b. A foreign key that references order
    c. A foreign key that references Sellable Products

d. Quantity field. This is probably an integer.
e. A sell price field. This is probably a numeric that contains 7 digits with 2 reserved for decimal precision. It also should have a check constraint of greater than 0.
f. Extended Price. This is probably a numeric that contains 8 digits with 2 reserved for decimal precision.

21. Export this diagram as a PNG. Make sure the filename follows the naming rules for this lab. Attach to the submission area for this lab.

Lab 4 – Normalization

Introduction

Normalization is an important concept in database design. This lab will help you develop your normalization skills.

Submission

Write up the requested items and submit a Word document. Plain text, Older Word versions and RTF files are also acceptable. OpenOffice and LibreOffice documents will result in an instant 0.

You will submit a single file. Filename should be <Lastname>-<firstname>-lab3.docx. It will contain the normalization steps for tasks 2 and 4 as well as the diagrams for tasks 3 and 5.

Tasks

Normalization – Exercises & Answers

staffNo	dentistName	patientNo	patientName	appointment		surgeryNo
				date	time	
S1011	Tony Smith	P100	Gillian White	12-Aug-03	10.00	S10
S1011	Tony Smith	P105	Jill Bell	13-Aug-03	12.00	S15
S1024	Helen Pearson	P108	Ian MacKay	12-Sept-03	10.00	S10
S1024	Helen Pearson	P108	Ian MacKay	14-Sept-03	10.00	S10
S1032	Robin Plevin	P105	Jill Bell	14-Oct-03	16.30	S15
S1032	Robin Plevin	P110	John Walker	15-Oct-03	18.00	S13

Figure 1: Details of patient dental appointments.

1. The table shown in Figure 1 is susceptible to update anomalies. Provide examples of insertion, deletion, and modification anomalies.
2. Fill out the process of normalizing the table shown in Figure 1 to 3NF. State any assumptions you make about the data shown in this table.
3. Create a conceptual diagram based on task 2.

UNF:

DentalOffice [(staffNo,dentistName), (patientNo,patnientName), appointment,surgeryNo]

1NF:

2NF:

3NF:

4. Based on the following "User View", normalize up to 3NF. UNF has been completed for you.
5. Create a conceptual diagram based on the 3NF view of task 3.

Timberbridge
Manufacturing
Supplier Order
Detail Report

Order Number	Order Date	Supp. Number	Supp. Name	Part Number	Part Desc	Qty Ordered	Quoted Price
12489	9/2/2016	124	Fastnall	SC12	1/2" Screw	1100	89.95
12491	9/2/2016	311	Butt-r-us	BT04	1"Butt Jnt	500	440
				BT98	2"Butt Jnt	250	399.99
				EPX1	Butt Epoxy	5	19.96
12494	9/4/2016	124	Fastnall	SC34	3/4" Screw	2500	279.96
12495	9/4/2016	256	Stapleton	STP12	1/2" Frn. Stp.	3200	404.96
12498	9/5/2016	124	Fastnall	TC3	Funiture Tacks	1200	150.96
				SC12	1/2" Screw	1000	79.96

****** ******** End of report ********

UNF:
Order [OrderNo, Orderdate, SuppNo, SuppName, (PartNo, PartDesc, QtyOrd, Price)]

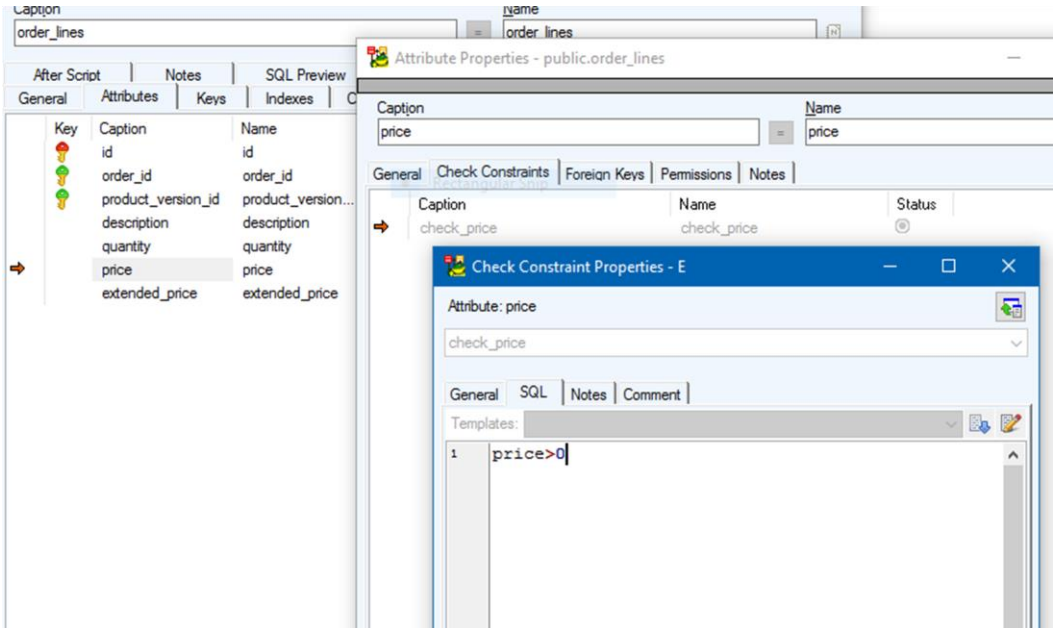
1NF:

2NF:

3NF:

Tips & Tricks

TIP: CREATING A CHECK CONSTRAINT IS SIMILAR TO CREATING AN ATTRIBUTE, JUST ONE LAYER DEEPER. DOUBLE CLICK ON THE ATTRIBUTE THAT REQUIRES A CHECK CONSTRAINT. CLICK ON ADD. ENTRER THE CHECK CONSTRAINT ON THE SQL TAB. IT IS A FREE TEXT FIELD.



Lab 5 – Diagramming 2

Introduction

This lab will continue the concepts of diagramming as seen in **Lab 3 – Diagramming 1**. This lab will see you making changes to an existing diagram and synchronizing them to the server.

Submission

You will submit 2 files:

1. A diagram exported as a PNG. Filename should be <Lastname>-<firstname>-lab5.png
2. The update processing script as a text file: Filename should be <Lastname>-<firstname>-lab5.sql

Tasks

1. Load Toad Data Modeler. Open the thinkcube.txp file that was downloaded during **Lab 1 – Installing your environment.**
2. Save your file with a new name. I would suggest lab 4 or Lab 4 just to keep things nice and tidy.
3. Add the following fields. The table, field name and default values are listed
    a. versions: active boolean default true
    b. product_versions: active boolean default true
    c. product_types: active boolean default true
4. Create a reference table called email_log_types. See **Lab 3 – Diagramming 1** on how to create a reference table. Sample values woud include “Opt out activity” and “Campaign sent”. (You don’t need to create these, I am only putting them in as an example)
5. Create a table with the following properties:
    a. Name: customer_email_log
    b. Primary key:id
    c. Fields:
        i. log_date, timestamp with default of now()
        ii. log_note, text, NOT NULL
    d. A reference from customers
    e. A reference from email_log_types
6. Export a PNG named <Lastname>-<firstname>-lab4.png. For instructions on how to do this see **Lab 2 –**
7. Click on the File-Reverse Engineering->Connections menu item.
8. Click the new connection icon in the tool strip. It should be the first one to the left.
9. Give the connection a name. i.e. “Thinkcube local” and click Next.
10. Select PostGreSQL 9.4 or 9.5 in the list of data sources and click Next.
11. Accept “Native Connection” and click Next.
12. Enter the following values in the connection information dialog:
    a. Hostname: localhost
    b. Username: postgres
    c. Password: Whatever you entered in Lab 1 (I told you to remember it…)
    d. Check off Save Password
    e. Database name: thinkcube and click Next.
    f. Click Next and Finish.

13. Select the newly created connection and then click the test connection icon. It looks like a little plug with a blue checkmark (Third from the left). If it didn't work, check your settings and try again. If you get stuck, come see the lab monitor.
14. Close the window.
15. Click on the Model->Generate Change Script->Run menu item.
16. Ensure that "Store Connection" radio button is selected and select the connection you just created in the drop down and click Next.
17. Accept all defaults clicking on Next until you hit the "Review" table.
18. Select a logical folder for your output file. Remember to name it properly according to the lab instructions.
19. Click Finish.
20. Now launch pgAdmin III.
21. Open the connection to your local host and double click the thinkcube database.
22. Click the SQL icon in the tool strip.
23. Open the file that was just created on step 19.
24. Click the run icon (Little green arrow head).

Lab 6 – SQL 1 – DDL and DML

Introduction

Often, you may need to create database structures manually. This lab will help prime your knowledge of manually creating database objects. Since creating, updating and deleting data are core actions to any database server, this lab will also see you maintaining your data in the thinkcube database.

Submission

Make sure to save all commands from each step. Each step can only be run once successfully.

You will submit a single file. Filename should be <Lastname>-<firstname>-lab6.sql

Tasks

1. Load pgAdmin III. This was installed in **Lab 1 – Installing your environment**
2. Click on the local server connection. It may be named something like: PostgreSQL 9.4(localhost:5432)
3. Click on "Databases"
4. Click on thinkcube
5. Click on the SQL icon in the toolbar. It looks like a magnifying glass with the letters SQL in it. This will open the SQL editor.
6. Type in following commands

```
CREATE TABLE order_logs (
    id BIGSERIAL NOT NULL PRIMARY KEY,
    log_date timestamp default now(),
    log_note text,
    order_id int8 references orders(id)
);
```

7. Click the Run icon (Green sideways triangle)
8. Create another table. This one is a reference table called order_statuses. For a reminder of what the fields should look like in a reference table, see Lab 2 –
9. Alter the customers table and add a phone number column.
10. Alter the customers table to have a Boolean field called opt_out.
11. Alter the table orders and add a field called order_status_id. Datatype is int8 and it should reference id field from the table order_statuses.
12. Insert the following values into the order status table that was created in Lab 6:
    a. New
    b. Processing
    c. Shipped
    d. Invoiced
    e. Paid

    You will need to create 5 separate **INSERT** statements.

Tips & Tricks

TIP: YOU SHOULD DO YOUR ALTERS AS INDIVIDUAL TASKS TO HELP SIMPLIFY DEBUGGING.

IF YOU ARE HAVING PROBLEMS WITH STEP 11, LOOK AT STEP 6 FOR A HINT ON HOW TO CREATE A REFERENCE.

13. Run the following command to ensure that everything that needs to be in that table is present:
    SELECT * FROM order_statuses;
14. **Insert** these values into the email log types table that was created in Lab 6:
    a. Subscribe
    b. Unsubscribe
    c. Campaign Mailer

    You will need to create 3 separate INSERT statements.
15. Run the following command to ensure that everything that needs to be in that table is present:
    SELECT * FROM email_log_types;
16. **Update** all customers to have a value of false in the opt_out column
17. Run the following command to ensure that everything that was to be updated in that table actually happened:
    SELECT * FROM customers;
18. **Update** all orders to have a status of "New". Run step 13 if you don't remember the id of the new order status.
19. Run the following command to ensure that everything that was to be updated in that table actually happened:
    SELECT * FROM orders;

Lab 7 – SQL 2 – Querying your database

Introduction

This lab will have you creating queries with single and multiple criteria. You will be using the thinkcube database for this lab.

Submission

Make sure to save all commands from each step.

You will submit a single file. Filename should be <Lastname>-<firstname>-lab7.sql

Tasks

1. Select all customers.
2. Select all customers that live in the city of “Fairbanks”.
3. Select all customers who have an email address ending in “.edu”.
4. Select all customers that live in the city of “Fairbanks” who have an email address ending in “.edu”
5. Select all customers who have a name that contains “dan” anywhere.
6. Select all orders placed before November 1, 2015.
7. Select all order lines for order id 39357.
8. Select all order lines for order ids 41200 through 41323 inclusively
9. Select all order lines for order ids 41200 through 41323 exclusively.
10. Select all employees who work in sales. (you need to do 2 queries to achieve this, unless you are prepared to create a JOIN).

Lab 8 – SQL 3 – Aggregates, JOINS and Subqueries

Introduction

Summarizing data and running complex queries are tasks that are very important to the professional database developer. This lab will help you develop these skills. You will be using the thinkcube database for this lab.

You MUST restore the tradeshow_leads.backup file into the thinkcube database before attempting tasks 16-18. To restore, right click on the thinkcube database and select Restore… browse to the tradeshow_leads.backup file you downloaded from the lab section and hit the "Restore" button. Once complete, click "Done"

Submission

Make sure to save all commands from each step.

You will submit a single file. Filename should be <Lastname>-<firstname>-lab8.sql

Tasks

1. Select all customers from Canada. This must be done as a JOIN
2. Select all orders for customers between ids 4563 and 5678. Display the customer name, customer city, order id and order date. Order the results from oldest to newest based on order date.
3. Redo #2, this time only for customers in Canada.
4. Redo #3, also display the name of the province in addition to the fields already defined.
5. Calculate the average price in the order lines for product version id 198
6. Calculate the maximum price in the order lines for product version id 198
7. Calculate to the order total for order id 34567. Display the order id, the order date, the count of order lines and the sum of the extended price.
8. Calculate the average selling price for any product in the product name of Groovecom.
9. Redo #8 as a subquery.
10. Calculate the average, minimum and maximum number of order lines per order. I just want the average,minimum and maximums of all orders combined. This will required a derived table that calculates the number of order lines grouped by order.
11. Calculate the average number of orders per customer in Ontario.
12. Calculate the average sum of the extended price per customer for the first 500 customers.
13. Retrieve all columns from both products and product versions using a regular join
14. Redo #13, this time use a left join. What is different this time?
15. Redo #14, this time limit only to products in the products table that are not in the product versions table.
16. Using a union operator, retrieve a list of email address from both the customers and tradeshow_leads tables.
17. Using an except operator, identify all customers in the table tradeshow_leads that are not in the customers table using the e-mail field
18. Using an intersect operator, identify all customers in the table tradeshow_leads that are in the customers table using the e-mail field.

Lab 9 – SQL 4 – Views, Indexes and Transactions

Introduction

Advanced concepts such as views and transactions are always important and can be a challenge to understand. You will be using the thinkcube database for this lab.

Submission

Make sure to save all commands and or results as applicable from each step.

You will submit a single file. Filename should be <Lastname>-<firstname>-lab9.sql

Tasks

1. Run the following command and take note of the execution time (Bottom right of the Query Window):
   select count(*) from order_lines join orders on orders.id=order_lines.order_id
2. Create an index on the order_id column of the order_lines table.
3. Re-run the command for task 1. What is the new execution time?
4. Run the following command:
   select * from orders where order_date >='2015-01-01' and order_date <='2015-03-01';
5. Create an index on order_date on the orders table.
6. Re-run the command for task 4. What is the new execution time?
7. Create a view that lists customer email and the state/province name.
8. Update the view from 7 to exclude empty email addresses.
9. Update the view from 8 to include the country.
10. On the query window, click File->New Window. This will establish a second connection to your database server. We will refer to these as SQL1 and SQL2. Make sure to not get confused as to which window to use.
11. In SQL2, count the number of rows in customers.
12. In SQL1, type and the following commands:
    BEGIN;
    Insert into customers (name, email,state_province_id,country_id) values ('test','test@test.com',1,1);
13. Again in SQL2, count the rows in the customers table? What happened?
14. In SQL1, run the following command:
    COMMMIT;
15. Again in SQL2, count the rows in the customers table? What happened?
16. Redo steps 12 and 13.
17. In SQL1, run the following command:
    ROLLBACK;
18. Again in SQL2, count the rows in the customers table? What happened?

Lab 10 – Creating Functions, Rules and Triggers

Introduction

A base principle in all programming involves creating reusable pieces of code. Although databases are not a runtime environment, they do have the ability to have custom code created and stored that can be reused as needed. You will be using the thinkcube database for this lab.

Submission

Make sure to save all commands from each step.

You will submit a single file. Filename should be <Lastname>-<firstname>-lab9.sql

Tasks

1. Create a function that will generate a random phone number. This number must be formatted standard Canadian (999) 999-9999. So far I have seen 7 different approaches that work. You may need to research some of these depending on your approach:
    a. RANDOM()
    b. TRUNC()
    c. SUBSTRING()
    d. CAST()
    e. How to concatenate
2. Create a function that will strip out all characters except for numbers. You will need to research how to use:
    a. regexp_replace()
3. Create a new field on the customer table called clean_phone. It should be a varchar(15).
4. Create a new trigger on insert and update that will populate the clean_phone field using the function you created in task 2.
5. Run the following query:
   ```
   update customers set phone = <THE FUNCTION YOU CREATED IN TASK 1>
   ```
6. Create a trigger that logs changes to the opt-out field on customers into the table you created task 5 of
7. **Lab 5 – Diagramming** 2. This trigger is to log the date and time of the change, and it should create a note indicating what state the opt-out field became. Something like “customer opted out” if the opt-out field. The field email_log_type_id should be set to whatever the id of the “Opt out activity” row in the email_log_types table.