Project Two: README

An outlined proposal for a real-world database application facilitating the governance of select agent research by universities and governmental agencies.

Cristopher D Benge
cb3704@columbia.edu

Chisom J Amalunweze
jca2158@columbia.edu

December 1, 2021

*Submitted in partial fulfillment of the requirements for the
assignment of COMS W4111 - Introduction to Databases
to the
Faculty of Graduate Studies
of the Fu Foundation School of Engineering & Applied Science
at Columbia University in the City of New York*

Publication submitted to the graduate studies faculty of Columbia University in the City of New York for evaluation and consideration: *Alexandros Biliris, PhD*

This document was typeset using LaTeX, with a mixture of `classicthesis` developed by André Miede. The bibliography was processed by Biblatex. Robert Slimbach's Minion Pro acts as both the text and display typeface. Sans-serif text is typeset in Slimbach and Carol Twombly's Myriad Pro; monospaced text uses Jim Lyle's `Bitstream Vera Mono`.

CONTENTS

Project Two Submission 4
1.1 Outline of Schema Changes 4
Interesting Queries 5
2.1 Query #1 5
2.1.1 SQL Query 5
2.1.2 Results 5
2.2 Query #2 6
2.2.1 SQL Query 6
2.2.2 Results 6
2.3 Query #3 7
2.3.1 SQL Query 7
2.3.2 Results 7
Appendix 8
3.1 Schema Creation Script 8

PROJECT TWO SUBMISSION

Project Two builds on the schema, data, and queries that we submitted in Project One. The name of our PostgreSQL account for this submission is **cb3704**. The details of our team and database account is listed below:

Team Member Name	Columbia UNI	Columbia Email
Cristopher D Benge	cb3704	*cb3704@columbia.edu*
Chisom J Amalunweze	jca2158	*jca2158@columbia.edu*

1.1 Outline of Schema Changes

The following changes have been implemented in the second phase of our group project. The full script of all object creation and table population is provided in the appendix section of this document:

- ADMINISTRATOR_COMPOSITE table created and populated.
  - Contains a **composite type** column named *adminName*.
  - Leverages new custom type **fullname** added to the schema.
  - This table serves as a replacement for the existing **Administrator** table from Project 1.
  - 10 rows populated.
- LABORATORY_WITH_ARRAY table created and populated.
  - Utilizes the **array** data type for column *inspectionSchedule*
  - Utilizes the **tsvector (document)** data type for column *inspectionCertifications*
  - This table serves as a replacement for the existing **Laboratory** table from Project 1.
  - 10 rows populated.
- LABORATORY_AUDIT table created.
  - Audit table that captures change data and metadata after inserts on **Laboratory_with_array** table.
  - Automatically populated 10 rows after inserts on **Laboratory_with_array**.
- FUNC_LABORATORY_AUDIT function created.
  - Function is called by **trg_insert_Laboratory** trigger.
  - Populates the **Laboratory_Audit** table.
- TRG_INSERT_LABORATORY trigger (AFTER INSERT) created.
  - Executes function **func_Laboratory_audit** on table **Laboratory_with_array** after insert operations.
- FULLNAME type added.
  - Custom type created with *firstName* and *lastName* text values.
  - Used by new table **Administrator_Composite**.

INTERESTING QUERIES

2.1 Query #1

As an inspector, I perform on-site laboratory inspections that are either scheduled or surprise and doucment the inspection summary. I would only want to view the list of laboratories that have not had a SUCCESSFUL inspection in at least the last 12 months of type AD-HOC or ROUTINE and limit the list to only those labotatories that are scheduled to be operating in September (ARRAY searching) – **NOTE**: *use of the ARRAY data type in filtering.*

2.1.1 SQL Query

```
with ins as (
        select  li.laboratoryId, max(li.scheduledDate) as scheduleDate
          from  laboratoryInspection li
         where  li.inspectionType in ('ad-hoc', 'routine')
           and  li.inspectionOutcome = 'successful'
         group  by li.laboratoryId
        having  max(li.scheduledDate) >= (now() - INTERVAL '1 YEAR')
)

select  l.laboratoryId, l.safetyLevel, e.entityName, e.contactFirstName,
        e.contactLastName, e.contactPhoneNumber, e.contactEmailAddress
  from  Laboratory_with_array l
        join ins on l.laboratoryId = ins.laboratoryId
        join ResearchEntity e on l.managingEntityId = e.entityId
 where  'September' = any (l.inspectionSchedule);
```

2.1.2 Results

laboratoryId	safetyLevel	entityName	contactFirstName	contactLastName
8	0	Colorado State University	Andrew	Fogarty
9	2	University of Pennsylvania	Siduo	Jiang
2	2	The University of Texas at Austin	Mary	Curie
3	2	University of Massachusetts Amherst	Mark	Knoffler

contactPhoneNumber	contactEmailAddress
970-491-6444	csurams@colostate.edu
215-898-6236	andrew.maksym@upenn.edu
512-471-8871	irb@austin.utexas.edu
413-545-0111	rescomp@research.umass.edu

2.2 Query #2

As a faciltator, I need to review reported incidents (theft, loss, spill) but I want to review reports only from those labs that are certified to handle Safety level 1 AND hazardous materials. – **NOTE**: *use of text search of document data type via tsvector and to_tsquery syntax.*

2.2.1 SQL Query

```
select  l.laboratoryId, e.entityName, i.incidentReportedDate, i.incidentOccurredDate,
        i.threatLevel, i.incidentType, i.incidentSummary, i.investigationOpenDate,
        i.investigationClosedDate
  from  Incident i
        join Laboratory_with_array l on i.laboratoryId = l.laboratoryId
        join ResearchEntity e on l.managingEntityId = e.entityId
 where  coalesce(i.investigationStatus, 'open') = 'open'
   and  l.inspectionCertifications @@ to_tsquery('HazardousMaterials & Safety1');
```

2.2.2 Results

13 results returned, limiting to top 5...

labId	entityName	reportDate	...	summary	openDate	closedDate
3	University of Massachusetts Amherst	2021-11-05	...	all good!	2021-11-06	NULL
3	University of Massachusetts Amherst	2021-11-10	...	big spill	2021-11-12	NULL
4	University of South Carolina	2021-02-25	...	incident	2021-03-01	NULL
3	University of Massachusetts Amherst	2021-11-20	...	incident	2021-11-24	NULL
7	Cornell University	2021-03-24	...	incident	2021-03-25	NULL
...	...	...	...	...	...	...

2.3 Query #3

As an administrator with the last name of ’Turing’, I would like to see all of the experiments from the research universities that I have approved in the system, and sort the experiments by the start date of the experiment (ascending order). – **NOTE**: *use of filtering by the custom type [implemented in Project 2] in the WHERE clause.*

2.3.1 SQL Query

```
select  re.entityName, a.adminName, e.experimentStatus, e.experimentStartDate,
        e.experimentClosedDate
  from  ResearchEntity re
        join Administrator_composite a
            on re.approvedByAdministratorId = a.administratorId
        join Experiment e on re.entityId = e.entityId
 where  (adminName).lastName = 'Turing'
 order  by experimentStartDate asc;
```

2.3.2 Results

11 results returned, limiting to top 5...

entityName	adminName	status	startDate	closedDate
University of Massachusetts Amherst	(Alan,Turing)	closed	1977-10-09	2021-01-22
Drexel University	(Alan,Turing)	closed	1977-10-09	2021-01-22
Drexel University	(Alan,Turing)	design	2021-05-10	NULL
University of Massachusetts Amherst	(Alan,Turing)	design	2021-05-10	NULL
Drexel University	(Alan,Turing)	initiated	2021-10-24	NULL
...	...	...	...	...

APPENDIX

3.1 Schema Creation Script

```
/*
SQL Schema for Project 2, COMS W4111 - Introduction to Databases
Dr. Alexandros Biliris, Section V03 (CVN)
Columbia University, Fall 2021

By : Cristopher Benge (cb3704@columbia.edu)
Chisom Jachimike Amaluweze (jca2158@columbia.edu)
*/

--- *******************************************
--- CREATE TABLES, FUNCTIONS, TRIGGERS
--- *******************************************

drop table if exists Laboratory_audit;
drop table if exists Administrator_Composite;
drop table if exists Laboratory_with_array;
drop type if exists fullname;

create type fullname as (
firstName text,
lastName text);

create table Administrator_Composite (
administratorId int not null generated always as identity,
adminName fullname,
primary key (administratorId));

create table Laboratory_with_array (
laboratoryId int not null generated always as identity,
safetyLevel int not null check (safetyLevel between 0 and 3),
managingEntityId int not null,
managedSinceDate date not null default CURRENT_DATE,
inspectionSchedule text array[4],
inspectionCertifications tsvector,
primary key (laboratoryId),
foreign key (managingEntityId) references ResearchEntity (entityId));

create table Laboratory_audit (
laboratoryId int not null,
safetyLevel int not null,
managingEntityId int not null,
managedSinceDate date not null,
inspectionSchedule text array[4],
inspectionCertifications tsvector,
UserName name,
AddedTime date);

create or replace function func_Laboratory_audit()
returns trigger as
$$
begin
insert into Laboratory_audit (laboratoryId, safetyLevel, managingEntityId,
managedSinceDate, inspectionSchedule, inspectionCertifications, UserName, AddedTime)
values(NEW.laboratoryId, NEW.safetyLevel, new.managingEntityId, NEW.managedSinceDate,
NEW.inspectionSchedule, NEW.inspectionCertifications, current_user, current_date);

return new;
end;
$$
language 'plpgsql';

create trigger trg_insert_Laboratory
after insert on Laboratory_with_array
for each row
execute procedure func_Laboratory_audit();

--- *******************************************
--- Insert New Data
--- *******************************************

insert into Administrator_composite (adminName)
values(row('Charles','Babbage')),
(row('Ada','Lovelace')),(row('Alan', 'Turing')),(row('Edgar', 'Codd')),
(row('Jim', 'Gray')),(row('Christopher','Date')),(row('Ralph', 'Kimball')),
(row('Bill', 'Inmon')),(row('Ken','Henderson')),(row('Larry','Ellison'));

insert  into Laboratory_with_array (safetyLevel, managingEntityId, managedSinceDate,
inspectionSchedule, inspectionCertifications)
select  floor(random() * 3) as safetyLevel, e.entityId,
(select now() - '1 years'::interval * round(random() * 2)) as managedSince,
'{"January", "March", "June", "September" }',
to_tsvector('Safety1 Safety2 HazardousMaterials TentedPrograms HazMatApproved')
from  ResearchEntity e;
```