Simon Fraser University

CMPT 354

Summer 2022

Group Project - Implementation of a Relational Database

Project Title:	Police Database (PoliceDB)
Project Milestone:	Milestone 4(a) - Implementation

#	Student Name	Student Number	Email Address
1	**KAVI BAKSHI**	**301380038**	**kbakshi@sfu.ca**
2	**SANSEERAT VIRK**	**301369321**	**ssv3@sfu.ca**
3	**DYLAN FENG**	**301422343**	**ddf3@sfu.ca**

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above.

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by Simon Fraser University**.**

A short description of the final project, and what it accomplished

The project is the admin view connected to the police database which itself can be connected to other different types of views. Another type of view can be a police view which may not have access to all features of the admin view.
Our admin view consists of retrieving the different tables. It allows clients to update, delete certain information and perform other queries required by milestone 4/5, and then view the updated tables.
We have selected a subset of tables from the database to be accessed by the Admins and limited amount of operations a Admin can perform on the database in order to satisfy the milestone 4 requirements (Different types of Queries; INSERT, UPDATE, DELETE, DIVISION, etc.)

A description of how your final schema differed from the schema you turned in.

The final schema grew bigger through the milestones as we needed to comply with the milestone requirements.
Tables needed to be modified or split into two. To normalize into BCNF form, the police officer table got split into two. The product of this split was the driver license table. The driver license is a prerequisite to enter a unique badge number into the system to ensure two badge numbers don't belong to the same person. We can guarantee a person is unique since the government enforces a unique Driver License among people. The driver license has a one to one correspondence with the badge number

The address in crime location was reduced to coordinates of a crime since they capture all the attributes of an address. This is useful for processing addresses in different applications as it enforces a set standard. Whereas a one line of address requires string processing on the application side which is expensive.

List changes that were made

1) Complaint (Complaint no., Person, Phone no., Description) changed to Complaint (Complaint no., Phone no., Description) since persons name is not necessarily required.
2) Evidence (Evidence ID, Type, Description) changed to Evidence (Evidence ID, Description, **crime_id**) to allow for crime_id to be foreign key thus allowing for connection between evidence and crime, and type of evidence no longer required since we have description.
3) Oversees (**Badge number, Crime ID**, **Government ID**, Case no.) changed to Oversees (**Badge number, Crime ID**, **Government ID**, Case no., description) to allow for reason for overseeing a certain case
4) Checks In/Out_Equipment (Equipment ID,Type, **Badge no,**, Date, Time) changed to Checks In/Out_Equipment (Equipment ID, **Badge no,**, Date, Time) so there is no type involved, as information not required
5) Accesses (**Badge no.**, **Aisle no.**, **Shelf no.**, **Position**, Time Accessed, Date Accessed) to evidence_accesses (**Evidence ID, detective_badge,** Date Accessed), there is no need to have badge no. aisle no. etc since the evidence ID suffices and is related to those attributes in evidence_storage table.
6) LockedUp (**Suspect ID**, **Cell no.,** Start Date, Duration) changed to LockedUp (**Suspect ID**, **Cell no.,** Start Date, Release Date) since duration can be calculated from release date - start date.

Github link: https://github.com/sanvirk99/Relational_Database_App

SQL Queries and Screenshots

Insert Query:

Inserting into police officer table

PostgreSQL Query:

```
INSERT INTO public.driver_licence(
      driver_l, name, birth_date, height_cm, eye_colour, address)
      VALUES ('${req.body.licence}',
        '${req.body.name}',
        '${req.body.birth_date}',
        '${req.body.height}',
        '${req.body.eye_colour}',
        '${req.body.address}');


INSERT INTO public.police_officer
    (badge, driver_l,duty_name,rank_name)
    VALUES
    (nextval('badge_sequence'), '${req.body.licence}','${req.body.duty_name}','${req.body.duty_rank}');
```

Before tables:

police_officer

GO TO HOME PAGE

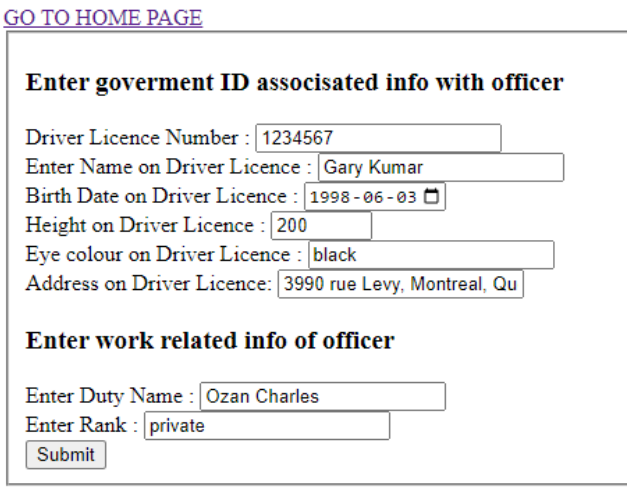
badge	driver_l	duty_name	rank_name
123456	1111111	Ronald Tyson	Constable
123457	2222222	Emyr Johnston	Captain
123458	3333333	Arianne Walker	Lieutenant
123459	4444444	Amy Rutledge	Officer
123460	5555555	Dave Talley	Sergeant

driver_licence

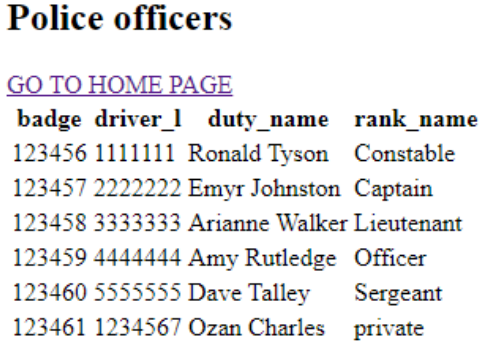
GO TO HOME PAGE

driver_l	name	birth_date	height_cm	eye_colour	address
1111111	William Zheng	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	black	3881 Granville St, Halifax, Nova Scotia, B3K 5M1
2222222	Richard Wu	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	black	731 Findlay Creek Road, Grasmere, British Columbia, V0B 1R0
3333333	Thomas Kumar	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	gray	2707 Blanshard, Victoria, British Columbia, V8W 2H9
4444444	Charles Wang	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	brown	723 Merivale Road, Ottawa, Ontario, K2G 3K2
5555555	Daniel Liu	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	light brown	3202 Main St, Rosetown, Saskatchewan(SK), S0L 2V0

Insert Form from GUI:



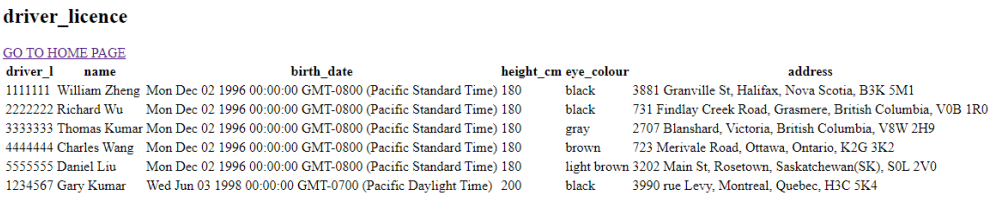
Resulting Tables:



Police officers

GO TO HOME PAGE

badge	driver_l	duty_name	rank_name
123456	1111111	Ronald Tyson	Constable
123457	2222222	Emyr Johnston	Captain
123458	3333333	Arianne Walker	Lieutenant
123459	4444444	Amy Rutledge	Officer
123460	5555555	Dave Talley	Sergeant
123461	1234567	Ozan Charles	private



driver_licence

GO TO HOME PAGE

driver_l	name	birth_date	height_cm	eye_colour	address
1111111	William Zheng	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	black	3881 Granville St, Halifax, Nova Scotia, B3K 5M1
2222222	Richard Wu	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	black	731 Findlay Creek Road, Grasmere, British Columbia, V0B 1R0
3333333	Thomas Kumar	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	gray	2707 Blanshard, Victoria, British Columbia, V8W 2H9
4444444	Charles Wang	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	brown	723 Merivale Road, Ottawa, Ontario, K2G 3K2
5555555	Daniel Liu	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	light brown	3202 Main St, Rosetown, Saskatchewan(SK), S0L 2V0
1234567	Gary Kumar	Wed Jun 03 1998 00:00:00 GMT-0700 (Pacific Daylight Time)	200	black	3990 rue Levy, Montreal, Quebec, H3C 5K4

Delete CASCADE Operation Query

1) Delete government officer, cascades to delete row in oversees

PostgreSQL Query:

```
DELETE FROM goverment_officer
    WHERE goverment_id='${req.params.id}';
```

Data in tables:

Goverment officer (deleting of

GO TO HOME PAGE

goverment_id	name	.....
1234	Bobby Kirk	Delete
1235	Jagga Virk	Delete
1236	Gaggu GILL	Delete
1237	Sidhu Moosa	Delete
1238	Raj Kirk	Delete

Goverment officer

GO TO HOME PAGE

goverment_id	crime_id	badge	description
1234	1	123456	Dispatch Operator Overseeing
1235	2	123457	Dispatch Operator Overseeing
1236	3	123458	Dispatch Operator Overseeing
1237	4	123459	Dispatch Operator Overseeing
1238	5	123460	Dispatch Operator Overseeing

Result of deleting government officer with ID: 1237:

Goverment officer (deleting officer also deletes asocciation in oversee table)

GO TO HOME PAGE

goverment_id	name	.....
1234	Bobby Kirk	Delete
1235	Jagga Virk	Delete
1236	Gaggu GILL	Delete
1238	Raj Kirk	Delete

Goverment officer

GO TO HOME PAGE

goverment_id	crime_id	badge	description
1234	1	123456	Dispatch Operator Overseeing
1235	2	123457	Dispatch Operator Overseeing
1236	3	123458	Dispatch Operator Overseeing
1238	5	123460	Dispatch Operator Overseeing

2) Delete Complaint from complaint table, cascades to delete complaint from office_review table

PostgreSQL Query:

```
DELETE FROM public.complaint
    WHERE complaint_id = '${req.params.id}';
```

Data in tables:

List of complaints (deleting officer also deletes asocciation in officer review table)

GO TO HOME PAGE

complaint_id	contact_info	description	.....
1	6045823342	loud neighbour	Delete
2	7782935267	gunshots heard	Delete
3	6042682935	illegal fireworks	Delete
4	7782350964	hit and run	Delete
5	7789230398	distripution of illegal materials	Delete

Reviewed complaints by office worker police specified with id along with time

GO TO HOME PAGE

badge	complaint_id	time
123458	1	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
123458	2	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
123458	3	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)

Result of deleting complaint with complaint_id = 3

List of complaints (deleting officer also deletes asocciation in officer review table)

GO TO HOME PAGE

complaint_id	contact_info	description	.....
1	6045823342	loud neighbour	Delete
2	7782935267	gunshots heard	Delete
4	7782350964	hit and run	Delete
5	7789230398	distripution of illegal materials	Delete

Reviewed complaints by office worker police specified with id along with time

GO TO HOME PAGE

badge	complaint_id	time
123458	1	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
123458	2	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)

Update operation Query

1) Changing any attribute of any row in table

PostgreSQL Query:

```
UPDATE public.suspect
    SET  name='${req.body.name}', birth_date='${req.body.birth_date}', height_cm='${req.body.height}',
    eye_colour='${req.body.eye_colour}', address='${req.body.address}'
    WHERE suspect_id=${req.params.id};
```

Data in tables:

List of all suspects

GO TO HOME PAGE

suspect_id	name	birth_date	height_cm	eye_colour	address	.....
1	James Smith	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	brown	1215 Reserve St, Castleton, Ontario, K0K 1M0	Update
2	Robert Johnson	Sun Aug 24 1997 00:00:00 GMT-0700 (Pacific Daylight Time)	170	black	3466 Islington Ave, Toronto, Ontario, M9V 2X5	Update
3	John White	Wed May 11 1994 00:00:00 GMT-0700 (Pacific Daylight Time)	185	brown	2845 3rd Avenue, Lloydminster, Alberta, T1J 3Y2	Update
4	David Brown	Thu Aug 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	190	grey	519 avenue Royale, Quebec, Quebec, G1E 2L3	Update
5	Joseph Lee	Thu Aug 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	200	black	4706 Burdett Avenue, Victoria, British Columbia, V8R 5A7	Update

Update form:

Abort update

change any information you want to update

name : David Brown
Birth Date : 2000-07-24
Height in CM : 190
Eye colour on Driver Licence : blue
Address on Driver Licence: 519 avenue Royale, Quebec
Submit

Result:

List of all suspects

GO TO HOME PAGE

suspect_id	name	birth_date	height_cm	eye_colour	address	.....
1	James Smith	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	brown	1215 Reserve St, Castleton, Ontario, K0K 1M0	Update
2	Robert Johnson	Sun Aug 24 1997 00:00:00 GMT-0700 (Pacific Daylight Time)	170	black	3466 Islington Ave, Toronto, Ontario, M9V 2X5	Update
3	John White	Wed May 11 1994 00:00:00 GMT-0700 (Pacific Daylight Time)	185	brown	2845 3rd Avenue, Lloydminster, Alberta, T1J 3Y2	Update
5	Joseph Lee	Thu Aug 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	200	black	4706 Burdett Avenue, Victoria, British Columbia, V8R 5A7	Update
4	David Brown	Mon Jul 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	190	blue	519 avenue Royale, Quebec, Quebec, G1E 2L3	Update

Selection Query

1) Select police officers with badge number greater than 123458

PostgreSQL Query:

```
SELECT badge, duty_name AS officer_name
FROM public.police_officer
WHERE badge > '123458';
```

Data in tables:

police_officer

GO TO HOME PAGE

badge	driver_l	duty_name	rank_name
123456	1111111	Ronald Tyson	Constable
123457	2222222	Emyr Johnston	Captain
123458	3333333	Arianne Walker	Lieutenant
123459	4444444	Amy Rutledge	Officer
123460	5555555	Dave Talley	Sergeant
123461	1234567	Ozan Charles	private

Result:

Select police officers with badge number greater than 123458

GO TO HOME PAGE

badge	officer_name
123459	Amy Rutledge
123460	Dave Talley
123461	Ozan Charles

2) Select prison cells with max capacity less than 10

PostgreSQL Query:

```
SELECT *
FROM public.prison
WHERE max_capacity < '10';
```

Data in tables:

Prison

GO TO HOME PAGE

cell	max_capacity	current_capacity
4	3	0
5	6	0
2	11	2
1	10	2
3	14	1

Result:

Select prison cells with max capacity less than 10

GO TO HOME PAGE

cell	max_capacity	current_capacity
4	3	0
5	6	0

Projection Query

1) All the types of equipment in the database

PostgreSQL Query:

```
SELECT equipment.type
FROM public.equipment;
```

Data in tables:

Equipment

GO TO HOME PAGE

equipment_id	type
1	ammunition
2	protective
3	electronic
4	unclassified
5	transportation

Result:

All the types of equipment in the database

GO TO HOME PAGE

type
ammunition
protective
electronic
unclassified
transportation

2) Names of all the police officers

PostgreSQL Query:

```
SELECT duty_name AS officer_name
FROM public.police_officer;
```

Data in tables:

police_officer

GO TO HOME PAGE

badge	driver_l	duty_name	rank_name
123456	1111111	Ronald Tyson	Constable
123457	2222222	Emyr Johnston	Captain
123458	3333333	Arianne Walker	Lieutenant
123459	4444444	Amy Rutledge	Officer
123460	5555555	Dave Talley	Sergeant
123461	1234567	Ozan Charles	private

Result:

Names of all the police officers

GO TO HOME PAGE

officer_name
Ronald Tyson
Emyr Johnston
Arianne Walker
Amy Rutledge
Dave Talley
Ozan Charles

Join Query

1) Names of all the suspects that are locked up in Cell no. 1
(Join the Suspects and Locked Up table to find the name(s) of the suspect(s) in cell no. 1

PostgreSQL Query:

```
SELECT suspect.name
FROM suspect, locked_up
WHERE suspect.suspect_id = locked_up.suspect_id AND locked_up.locked_in_cell = 1;
```

Data in tables:

Locked_up

GO TO HOME PAGE

suspect_id	locked_in_cell	start_date	release_date
1	1	Fri Jul 01 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Sat Jul 02 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
2	2	Fri Jul 01 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Tue Jul 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
3	2	Sat Jul 02 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Mon Jul 04 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
4	1	Sun Jul 10 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Thu Jul 28 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
5	3	Sun Jun 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Tue Jul 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)

Suspect

GO TO HOME PAGE

suspect_id	name	birth_date	height_cm	eye_colour	address
1	James Smith	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	brown	1215 Reserve St, Castleton, Ontario, K0K 1M0
2	Robert Johnson	Sun Aug 24 1997 00:00:00 GMT-0700 (Pacific Daylight Time)	170	black	3466 Islington Ave, Toronto, Ontario, M9V 2X5
3	John White	Wed May 11 1994 00:00:00 GMT-0700 (Pacific Daylight Time)	185	brown	2845 3rd Avenue, Lloydminster, Alberta, T1J 3Y2
4	David Brown	Thu Aug 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	190	grey	519 avenue Royale, Quebec, Quebec, G1E 2L3
5	Joseph Lee	Thu Aug 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	200	black	4706 Burdett Avenue, Victoria, British Columbia, V8R 5A7

Result:

name
James Smith
David Brown

2) Birthdates of all the suspects that are locked up in Cell no. 2
(Join the Suspects and Locked Up table to find the Birth date(s) of the suspect(s) in cell no. 2

PostgreSQL Query:

```
SELECT suspect.birth_date
FROM suspect, locked_up
WHERE suspect.suspect_id = locked_up.suspect_id AND locked_up.locked_in_cell = 2;
```

Data in tables:

Locked_up

GO TO HOME PAGE

suspect_id	locked_in_cell	start_date	release_date
1	1	Fri Jul 01 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Sat Jul 02 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
2	2	Fri Jul 01 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Tue Jul 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
3	2	Sat Jul 02 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Mon Jul 04 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
4	1	Sun Jul 10 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Thu Jul 28 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
5	3	Sun Jun 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Tue Jul 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)

Result:

birth_date
Sun Aug 24 1997 00:00:00 GMT-0700 (Pacific Daylight Time)
Wed May 11 1994 00:00:00 GMT-0700 (Pacific Daylight Time)

Aggregate Query

1) Finding the number of suspects locked in a cell

PostgreSQL Query:

```
SELECT COUNT(*) AS total_suspects
    FROM public.locked_up;
'
```

Data in tables:

Locked_up

GO TO HOME PAGE

suspect_id	locked_in_cell	start_date	release_date
1	1	Fri Jul 01 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Sat Jul 02 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
2	2	Fri Jul 01 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Tue Jul 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
3	2	Sat Jul 02 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Mon Jul 04 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
4	1	Sun Jul 10 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Thu Jul 28 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
5	3	Sun Jun 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Tue Jul 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)

Result:

total_suspects
5

2) Find the longest duration a suspect is locked up for

PostgreSQL Query:

```
SELECT MAX(DATE_PART('day', release_date::timestamp - start_date::timestamp)) AS MaxDaysLockedUp
    FROM public.locked_up;
```

Data in tables:

Locked_up

GO TO HOME PAGE

suspect_id	locked_in_cell	start_date	release_date
1	1	Fri Jul 01 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Sat Jul 02 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
2	2	Fri Jul 01 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Tue Jul 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
3	2	Sat Jul 02 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Mon Jul 04 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
4	1	Sun Jul 10 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Thu Jul 28 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
5	3	Sun Jun 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Tue Jul 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)

Result:

What is the longest duration that a suspect is locked up for?

GO TO HOME PAGE

max_days_locked_up
30

3) Number of crimes that are committed and logged into the database

PostgreSQL Query:

```
SELECT COUNT(*) AS total_crimes
    FROM public.crime;
```

Data in tables:

crime

GO TO HOME PAGE

crime_id	description	cordinates
1	grand theft	49.181004 -122.802752
2	hit and run	49.181004 -122.802757
3	robbery from building	49.181004 -122.802786
4	illegal fire	49.181004 -122.802782
5	violence on property	49.181004 -122.802756

Result:

Number of crimes that are commited and logged into the database

GO TO HOME PAGE

total_crimes
5

Nested Aggregation with group-by

1) Show all suspects that have a locked up duration which is greater than the lowest/minimum locked up duration of suspects

PostgreSQL Query:

```
SELECT suspect.suspect_id, name, locked_in_cell, birth_date,
DATE_PART('day', release_date::timestamp - start_date::timestamp) AS DaysLockedUp
FROM public.suspect, public.locked_up
WHERE (suspect.suspect_id = locked_up.suspect_id) AND (suspect.suspect_id IN
                (SELECT suspect_id
                FROM public.locked_up
                GROUP BY suspect_id
                HAVING DATE_PART('day', release_date::timestamp - start_date::timestamp)
                > (SELECT MIN(DATE_PART('day', release_date::timestamp - start_date::timestamp))
                    FROM public.locked_up)));
```

Data in tables:

Suspect

GO TO HOME PAGE

suspect_id	name	birth_date	height_cm	eye_colour	address
1	James Smith	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	brown	1215 Reserve St, Castleton, Ontario, K0K 1M0
2	Robert Johnson	Sun Aug 24 1997 00:00:00 GMT-0700 (Pacific Daylight Time)	170	black	3466 Islington Ave, Toronto, Ontario, M9V 2X5
3	John White	Wed May 11 1994 00:00:00 GMT-0700 (Pacific Daylight Time)	185	brown	2845 3rd Avenue, Lloydminster, Alberta, T1J 3Y2
4	David Brown	Thu Aug 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	190	grey	519 avenue Royale, Quebec, Quebec, G1E 2L3
5	Joseph Lee	Thu Aug 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	200	black	4706 Burdett Avenue, Victoria, British Columbia, V8R 5A7

Locked_up

GO TO HOME PAGE

suspect_id	locked_in_cell	start_date	release_date
1	1	Fri Jul 01 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Sat Jul 02 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
2	2	Fri Jul 01 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Tue Jul 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
3	2	Sat Jul 02 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Mon Jul 04 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
4	1	Sun Jul 10 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Thu Jul 28 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
5	3	Sun Jun 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Tue Jul 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)

Result:

Show all suspects that have a locked up duration which is greater than the lowest/minimum locked up duration of suspects

GO TO HOME PAGE

suspect_id	name	locked_in_cell	birth_date	dayslockedup
2	Robert Johnson	2	Sun Aug 24 1997 00:00:00 GMT-0700 (Pacific Daylight Time)	4
3	John White	2	Wed May 11 1994 00:00:00 GMT-0700 (Pacific Daylight Time)	2
4	David Brown	1	Mon Jul 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	18
5	Joseph Lee	3	Thu Aug 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	30

2) Show all suspects that have a locked up duration which is greater than the average locked up duration of suspects.

PostgreSQL Query:

```
SELECT suspect.suspect_id, name, locked_in_cell, birth_date,
DATE_PART('day', release_date::timestamp - start_date::timestamp) AS DaysLockedUp
FROM public.suspect, public.locked_up
WHERE (suspect.suspect_id = locked_up.suspect_id) AND (suspect.suspect_id IN
                (SELECT suspect_id
                FROM public.locked_up
                GROUP BY suspect_id
                HAVING DATE_PART('day', release_date::timestamp - start_date::timestamp)
                 > (SELECT AVG(DATE_PART('day', release_date::timestamp - start_date::timestamp))
                    FROM public.locked_up)));
```

Data in tables:

Suspect

GO TO HOME PAGE

suspect_id	name	birth_date	height_cm	eye_colour	address
1	James Smith	Mon Dec 02 1996 00:00:00 GMT-0800 (Pacific Standard Time)	180	brown	1215 Reserve St, Castleton, Ontario, K0K 1M0
2	Robert Johnson	Sun Aug 24 1997 00:00:00 GMT-0700 (Pacific Daylight Time)	170	black	3466 Islington Ave, Toronto, Ontario, M9V 2X5
3	John White	Wed May 11 1994 00:00:00 GMT-0700 (Pacific Daylight Time)	185	brown	2845 3rd Avenue, Lloydminster, Alberta, T1J 3Y2
4	David Brown	Thu Aug 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	190	grey	519 avenue Royale, Quebec, Quebec, G1E 2L3
5	Joseph Lee	Thu Aug 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	200	black	4706 Burdett Avenue, Victoria, British Columbia, V8R 5A7

Locked_up

GO TO HOME PAGE

suspect_id	locked_in_cell	start_date	release_date
1	1	Fri Jul 01 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Sat Jul 02 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
2	2	Fri Jul 01 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Tue Jul 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
3	2	Sat Jul 02 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Mon Jul 04 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
4	1	Sun Jul 10 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Thu Jul 28 2022 00:00:00 GMT-0700 (Pacific Daylight Time)
5	3	Sun Jun 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)	Tue Jul 05 2022 00:00:00 GMT-0700 (Pacific Daylight Time)

Result:

Show all suspects that have a locked up duration which is greater than the average locked up duration of suspects

GO TO HOME PAGE

suspect_id	name	locked_in_cell	birth_date	dayslockedup
4	David Brown	1	Mon Jul 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	18
5	Joseph Lee	3	Thu Aug 24 2000 00:00:00 GMT-0700 (Pacific Daylight Time)	30

Division Query

1) Find the Checkout Officer(s) badge no. and name who have checked out all the different equipment.

   PostgreSQL Query:

```
SELECT DISTINCT checkout_officer as badge_no, duty_name as officer_name
FROM public.equipment_checkout, public.police_officer
WHERE equipment_checkout.checkout_officer = police_officer.badge AND checkout_officer NOT IN (SELECT checkout_officer
                              FROM ((SELECT checkout_officer, equipment_id
                                    FROM (SELECT equipment.equipment_id
                                          FROM public.equipment) as equip
                                    cross join (SELECT DISTINCT checkout_officer
                                                FROM public.equipment_checkout) as officer)
                                   EXCEPT
                                   (SELECT checkout_officer, equipment_id
                                    FROM public.equipment_checkout)) as checkedout);
```

Data in tables:

Equipment

GO TO HOME PAGE

equipment_id	type
1	ammunition
2	protective
3	electronic
4	unclassified
5	transportation

police_officer

GO TO HOME PAGE

badge	driver_l	duty_name	rank_name
123456	1111111	Ronald Tyson	Constable
123457	2222222	Emyr Johnston	Captain
123458	3333333	Arianne Walker	Lieutenant
123459	4444444	Amy Rutledge	Officer
123460	5555555	Dave Talley	Sergeant

equipment_checkout

GO TO HOME PAGE

equipment_id	checkout_officer	time
1	123456	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
2	123456	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
3	123457	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
4	123458	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
5	123459	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
3	123456	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
4	123456	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
5	123456	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
4	123459	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
3	123459	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
2	123459	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
1	123459	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)

Result:

Find the Checkout Officer(s) badge no. and name who have checked out all the different equipments

GO TO HOME PAGE

badge_no	officer_name
123456	Ronald Tyson
123459	Amy Rutledge

2) Finding the police officers who have investigated all crime

PostgreSQL Query:

```
select police.badge, police.duty_name from police_officer police where not exists
        (select * from crime where not exists
        (select investigates.crime_id from investigates where
        investigates.crime_id = crime.crime_id and investigates.officer = police.badge))
```

Data in tables:

police_officer

GO TO HOME PAGE

badge	driver_l	duty_name	rank_name
123456	1111111	Ronald Tyson	Constable
123457	2222222	Emyr Johnston	Captain
123458	3333333	Arianne Walker	Lieutenant
123459	4444444	Amy Rutledge	Officer
123460	5555555	Dave Talley	Sergeant

crime

GO TO HOME PAGE

crime_id	description	cordinates
1	grand theft	49.181004 -122.802752
2	hit and run	49.181004 -122.802757
3	robbery from building	49.181004 -122.802786
4	illegal fire	49.181004 -122.802782
5	violence on property	49.181004 -122.802756

investigates

GO TO HOME PAGE

crime_id	officer	time
1	123456	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
2	123456	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
3	123456	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
4	123456	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
5	123456	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
2	123457	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
3	123458	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
4	123459	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
1	123460	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
2	123460	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
3	123460	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
4	123460	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)
5	123460	Mon Aug 01 2022 11:35:59 GMT-0700 (Pacific Daylight Time)

Result:

Find the Officer(s) badge no. and name who have investigated all crime

GO TO HOME PAGE

badge	duty_name
123456	Ronald Tyson
123460	Dave Talley

Trigger Function and Query:

1) Function that checks if driver license length is within range and height less than 999

Creating Trigger Function:

```
CREATE FUNCTION check_driverlicense_values()
    RETURNS TRIGGER
AS $$
BEGIN
    IF length(NEW.driver_l) < 7 THEN
        RAISE EXCEPTION 'The Driver license number cannot be less than 7 numbers';
    END IF;
    IF length(CAST ((NEW.height_cm) AS TEXT)) > 3 THEN
        RAISE EXCEPTION 'The Height cannot be more than 999cm, person too tall';
    END IF;
    RETURN NEW;
END;
$$
LANGUAGE plpgsql;

--Creates trigger that executes function when inserting or updating
CREATE TRIGGER driverlicense_check
    BEFORE INSERT OR UPDATE
ON public.driver_licence
FOR EACH ROW
    EXECUTE PROCEDURE check_driverlicense_values();
```

PostgreSQL Query and results

Query

```
--Query that showcases the height error trigger. Height should be <999 cm
INSERT INTO public.driver_licence(
    driver_l, name, birth_date, height_cm, eye_colour, address)
    VALUES  ('9999999','Kavi Bakshi', '2000-06-08','1601','brown','456, Ash ST, British Columbia(BC), V5X 2C1');
```

Result

```
ERROR:  The Height cannot be more than 999cm, person too tall
CONTEXT:  PL/pgSQL function check_driverlicense_values() line 7 at RAISE
SQL state: P0001
```

Query

```
--Query that showcases the Driver license number error, driver license number should be >= length 7
INSERT INTO public.driver_licence(
    driver_l, name, birth_date, height_cm, eye_colour, address)
    VALUES  ('67','Kavi Bakshi', '2000-06-08','160','brown','456, Ash ST, British Columbia(BC), V5X 2C1');
```

Result

```
ERROR:  The Driver license number cannot be less than 7 numbers
CONTEXT:  PL/pgSQL function check_driverlicense_values() line 4 at RAISE
SQL state: P0001
```

2) Function that updates the prison tables current_capacity attribute according to locked_up table

Creating Trigger Function:

```
--Function that updates the prison tables current_capacity attribute according to locked_up table
CREATE FUNCTION update_prison_capacity()
    RETURNS TRIGGER
AS $$
BEGIN
    CASE TG_OP
    WHEN 'INSERT' THEN
        UPDATE public.prison AS prison
        SET current_capacity = current_capacity + 1
        WHERE prison.cell = NEW.locked_in_cell;
    WHEN 'DELETE' THEN
    UPDATE public.prison AS prison
    SET current_capacity = current_capacity - 1
    WHERE prison.cell = OLD.locked_in_cell
    AND    prison.current_capacity > 0;
    ELSE
      RAISE EXCEPTION 'Unexpected prison_count: "%". Error occurred, check function update_prison_capacity() ', TG_OP;
    END CASE;
    RETURN NULL;
END;
$$
LANGUAGE plpgsql;

--Creates trigger that executes function when inserting or deleting locked_up table
CREATE TRIGGER prison_capacity
    AFTER INSERT OR DELETE ON public.locked_up
FOR EACH ROW
    EXECUTE PROCEDURE update_prison_capacity();
```

Tables Before:

Locked_up Table

	suspect_id [PK] integer	locked_in_cell smallint	start_date date	release_date date
1	1	1	2022-07-01	2022-07-02
2	2	2	2022-07-01	2022-07-05
3	3	2	2022-07-02	2022-07-04
4	4	1	2022-07-10	2022-07-28
5	5	3	2022-06-05	2022-07-05

Prison Table

	cell [PK] integer	max_capacity smallint	current_capacity smallint
1	1	10	2
2	2	11	2
3	3	14	1
4	4	1	0
5	5	6	0

Tables after deleting row number 3 (suspect_id = 3) from locked_up table:

Locked_up Table

	suspect_id [PK] integer	locked_in_cell smallint	start_date date	release_date date
1	1	1	2022-07-01	2022-07-02
2	2	2	2022-07-01	2022-07-05
3	4	1	2022-07-10	2022-07-28
4	5	3	2022-06-05	2022-07-05

Prison Table

	cell [PK] integer	max_capacity smallint	current_capacity smallint
1	1	10	2
2	2	11	1
3	3	14	1
4	4	1	0
5	5	6	0